

SIMATIC

Automatisierungssystem S7-200

Systemhandbuch

Dieses Handbuch hat die Bestellnummer:

6ES7298-8FA01-8AH0

Vorwort, Inhaltsverzeichnis	
Einführung in die Micro-SPS S7-200	1
Installieren einer Micro-SPS S7-200	2
Installieren und Bedienen der Software STEP 7-Micro/WIN	3
Eingeben eines Programmierbeispiels	4
Erweiterte Funktionen in STEP 7-Micro/WIN	5
Grundlegendes zum Programmieren einer S7-200 CPU	6
Speicher der CPU: Datentypen und Adressierungsarten	7
Steuerung über Ein- und Ausgänge	8
Kommunikation im Netz mit einer S7-200 CPU	9
Operationssatz	10
Anhänge	
Technische Daten	A
Berechnungstabelle Strombilanz	B
Fehlermeldungen	C
Sondermerker	D
Einsetzen von STEP 7-Micro/WIN mit STEP 7 und STEP 7-Micro/DOS	E
Ausführungszeiten von AWL-Operationen	F
S7-200 Bestellnummern	G
S7-200 Fehlerbehebung	H
Index	

Sicherheitstechnische Hinweise

Dieses Handbuch enthält Hinweise, die Sie zu Ihrer persönlichen Sicherheit sowie zur Vermeidung von Sachschäden beachten müssen. Die Hinweise sind durch ein Warndreieck hervorgehoben und je nach Gefährdungsgrad folgendermaßen dargestellt:



Gefahr

bedeutet, daß Tod, schwere Körperverletzung oder erheblicher Sachschaden eintreten **werden**, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.



Warnung

bedeutet, daß Tod, schwere Körperverletzung oder erheblicher Sachschaden eintreten **können**, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.



Vorsicht

bedeutet, daß eine leichte Körperverletzung oder ein Sachschaden eintreten können, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.

Hinweis

ist eine wichtige Information über das Produkt, die Handhabung des Produktes oder den jeweiligen Teil der Dokumentation, auf den besonders aufmerksam gemacht werden soll.

Qualifiziertes Personal

Inbetriebsetzung und Betrieb eines Gerätes dürfen nur von **qualifiziertem Personal** vorgenommen werden. Qualifiziertes Personal im Sinne der sicherheitstechnischen Hinweise dieses Handbuchs sind Personen, die die Berechtigung haben, Geräte, Systeme und Stromkreise gemäß den Standards der Sicherheitstechnik in Betrieb zu nehmen, zu erden und zu kennzeichnen.

Bestimmungsgemäßer Gebrauch

Beachten Sie folgendes:



Warnung

Das Gerät darf nur für die im Katalog und in der technischen Beschreibung vorgesehenen Einsatzfälle und nur in Verbindung mit von Siemens empfohlenen bzw. zugelassenen Fremdgeräten und -Komponenten verwendet werden.

Der einwandfreie und sichere Betrieb des Produktes setzt sachgemäßen Transport, sachgemäße Lagerung, Aufstellung und Montage sowie sorgfältige Bedienung und Instandhaltung voraus.

Marken

SIMATIC® SIMATIC NET® und SIMATIC HMI® 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 können.

Copyright © Siemens AG 1998 All rights reserved

Weitergabe sowie Vervielfältigung dieser Unterlage, Verwertung und Mitteilung ihres Inhalts ist nicht gestattet, soweit nicht ausdrücklich zugestanden. Zuwiderhandlungen verpflichten zu Schadenersatz. Alle Rechte vorbehalten, insbesondere für den Fall der Patenterteilung oder GM-Eintragung.

Siemens AG
Bereich Automatisierungstechnik
Geschäftsgebiet Industrie-Automatisierungssysteme
Postfach 4848, D- 90327 Nürnberg

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, und notwendige Korrekturen sind in den nachfolgenden Auflagen enthalten. Für Verbesserungsvorschläge sind wir dankbar.

© Siemens AG 1998
Technische Änderungen bleiben vorbehalten.

Vorwort

Zweck des Handbuchs

Die Familie S7-200 umfaßt verschiedene Kleinststeuerungen (Micro-SPS), mit denen Sie eine breite Palette von Automatisierungsaufgaben lösen können. Durch das kompakte Design, die Möglichkeit der Erweiterung, den günstigen Preis und einen leistungsstarken Befehlssatz eignet sich die Familie S7-200 hervorragend für kleinere Steuerungsanwendungen. Die zahlreichen Möglichkeiten an Größen und Spannungsversorgungen bei der Wahl der CPU sowie die vielfältigen Programmiermöglichkeiten bieten Ihnen extrem hohe Flexibilität beim Umsetzen Ihrer Automatisierungslösungen.

Diesem Handbuch entnehmen Sie Informationen zum Installieren und Programmieren von S7-200 Automatisierungssystemen, einschließlich folgender Themen:

- Einbauen und Verdrahten der S7-200 CPU und der Erweiterungsmodule und Installieren der Software STEP 7-Micro/WIN
- Entwerfen und Eingeben eines Programms
- Funktionen der CPU, Datentypen und Adressierungsarten, Zyklus der CPU, Paßwortschutz und Kommunikation im Netz

Dieses Handbuch umfaßt außerdem Beschreibungen der Operationen und Beispiele zum Programmieren, typische Ausführungszeiten der Operationen sowie Datenblätter für die S7-200 Geräte.

Leserkreis

Dieses Handbuch wendet sich an Ingenieure, Programmierer und Wartungspersonal mit allgemeinen Kenntnissen über Automatisierungssysteme und Bedien- und Beobachtungssysteme.

Umfang des Handbuchs

Die Informationen in diesem Handbuch beziehen sich insbesondere auf die folgenden Produkte:

- Folgende S7-200 CPUs: CPU 212 Release 1.01, CPU 214 Release 1.01, CPU 215 Release 1.02 und CPU 216 Release 1.02
- Version 2.1 der Programmiersoftware STEP 7-Micro/WIN:
 - STEP 7-Micro/WIN 16 für Windows 3.1x (16 Bit)
 - STEP 7-Micro/WIN 32 für Windows 95 und Windows NT (32 Bit)

Approbationen

Die Produktreihe SIMATIC S7-200 erfüllt folgende Normen und Richtlinien:

- EG-Richtlinie 73/23/EEC zu Niederspannungen
- EG-Richtlinie 89/336/EWG zur elektromagnetischen Verträglichkeit
- Underwriters Laboratories, Inc.: UL 508 registriert (Industrial Control Equipment)
- Canadian Standards Association: CSA C22.2 Nr. 142, gepr. (Process Control Equipment)
- Factory Mutual Research: FM Klasse I, Kategorie 2, Gefahrenbereichgruppen A, B, C und D, T4A
- VDE 0160: Elektronische Geräte zur Verwendung in Starkstromanlagen

Anhang A entnehmen Sie, welche CPUs welchen Normen entsprechen.

Weitere Informationen

In folgenden Handbüchern finden Sie ausführliche Informationen zu bestimmten Themen:

- *Systemhandbuch Dezentrales Peripheriegerät ET 200*: In diesem Handbuch wird beschrieben, wie die Produkte ET 200 für die dezentrale Peripherie eingesetzt werden.
- PROFIBUS-Richtlinie (Process Field Bus) EN 50170: Diese Richtlinie beschreibt das Standardprotokoll für die DP-Kommunikationsfähigkeit der S7-200.
- *Textdisplay TD 200 Benutzerhandbuch*: In diesem Handbuch wird beschrieben, wie Sie das TD 200 installieren und es mit einem Automatisierungssystem S7-200 zusammen einsetzen.

Verwendung des Handbuchs

Wenn Sie zum ersten Mal mit einem Automatisierungssystem S7-200 arbeiten, sollten Sie das komplette Handbuch lesen. Haben Sie bereits Erfahrung im Umgang mit Bedien- und Beobachtungsgeräten, entnehmen Sie dem Inhaltsverzeichnis und dem Index, an welchen Stellen Sie bestimmte Informationen finden.

Das Handbuch umfaßt folgende Themenbereiche:

- "Einführung in die Micro-SPS S7-200" (Kapitel 1) bietet einen Überblick über einige Funktionalitäten der S7-200.
- "Installieren einer Micro-SPS S7-200" (Kapitel 2) bietet Vorgehensweisen, Abmessungen und grundlegende Richtlinien zum Installieren der S7-200 CPU und der Erweiterungsmodule.
- "Installieren und Bedienen der Software STEP 7-Micro/WIN" (Kapitel 3) beschreibt die Installation der Programmiersoftware. Außerdem werden die Funktionen der Software erläutert.
- "Eingeben eines Programmierbeispiels" (Kapitel 4) gibt Ihnen Anweisungen zum Eingeben eines Programmierbeispiels mit der Software STEP 7-Micro/WIN.
- "Erweiterte Funktionen in STEP 7-Micro/WIN" (Kapitel 5) beschreibt, wie Sie mit dem TD 200-Assistenten und dem Operations-Assistenten der S7-200 arbeiten und andere neue Funktionen in STEP 7-Micro/WIN einsetzen.
- "Grundlegendes zum Programmieren einer S7-200 CPU" (Kapitel 6), "Speicher der CPU: Datentypen und Adressierungsarten" (Kapitel 7) und "Steuerung der Ein- und Ausgänge" (Kapitel 8) bieten Informationen zur Datenverarbeitung in der S7-200 CPU und zur Ausführung Ihres Programms.
- "Kommunikation im Netz mit einer S7-200 CPU" (Kapitel 9) erläutert, wie Sie Ihre S7-200 CPU an verschiedene Arten von Netzen anschließen können.
- "Operationssatz" (Kapitel 10) erläutert die Operationen, die Sie für die S7-200 CPUs verwenden können, und liefert außerdem Beispiele zu den Operationen.

Zusätzliche Informationen (wie Datenblätter zu den Geräten, Beschreibungen zu den Fehlercodes, die Ausführungszeiten der Operationen und Informationen zur Fehlerbehebung) werden in den Anhängen aufgeführt.

Weitere Unterstützung

Haben Sie technische Fragen oder benötigen Sie Informationen zu Schulungen bzw. zur Bestellung dieses Produkts, wenden Sie sich bitte an Ihre Siemens-Vertretung.

Unter folgender Adresse im Internet erhalten Sie neben Informationen zu Produkten und Dienstleistungen von Siemens auch technische Unterstützung, Anwendungshinweise und Antworten auf häufig gestellte Fragen (FAQs):

<http://www.ad.siemens.de>

Inhaltsverzeichnis

1	Einführung in die Micro-SPS S7-200	1-1
1.1	Funktionen der verschiedenen Micro-SPS S7-200	1-2
1.2	Hauptkomponenten einer Micro-SPS S7-200	1-4
2	Installieren einer Micro-SPS S7-200	2-1
2.1	Vorbereitungen für die Montage	2-2
2.2	Ein- und Ausbauen einer Micro-SPS S7-200	2-5
2.3	Installieren der Feldverdrahtung	2-8
2.4	Schutzbeschaltungen	2-13
2.5	Spannungsversorgung	2-15
3	Installieren und Bedienen der Software STEP 7-Micro/WIN	3-1
3.1	Installieren der Software STEP 7-Micro/WIN	3-2
3.2	Einrichten der Kommunikationshardware mit STEP 7-Micro/WIN	3-4
3.3	Einrichten der Kommunikation mit einer S7-200 CPU	3-7
3.4	Einrichten der Voreinstellungen in STEP 7-Micro/WIN	3-25
3.5	Erstellen und Speichern eines Projekts	3-26
3.6	Erstellen eines Programms	3-27
3.7	Erstellen eines Datenbausteins	3-32
3.8	Arbeiten mit der Statustabelle	3-34
3.9	Arbeiten mit symbolischer Adressierung	3-36
4	Eingeben eines Programmierbeispiels	4-1
4.1	Erstellen eines Programms für eine Beispielanwendung	4-2
4.2	Anlegen eines Projekts	4-6
4.3	Erstellen einer Symboltabelle	4-8
4.4	Eingeben des Programms im Kontaktplan	4-10
4.5	Erstellen einer Statustabelle	4-14
4.6	Laden und Überwachen des Programmierbeispiels	4-15
5	Erweiterte Funktionen in STEP 7-Micro/WIN	5-1
5.1	Konfigurieren des Textdisplay TD 200 mit dem TD 200-Assistenten	5-2
5.2	Arbeiten mit dem Operations-Assistenten der S7-200	5-12
5.3	Arbeiten mit dem Analogeingabefilter-Assistent	5-14
5.4	Erstellen einer Liste der Querverweise	5-17

5.5	Erstellen einer Liste der verwendeten Elemente	5-18
5.6	Arbeiten mit den Funktionen Suchen und Ersetzen	5-19
5.7	Dokumentieren Ihres Programms	5-21
5.8	Drucken Ihres Programms	5-23
6	Grundlegendes zum Programmieren einer S7-200 CPU	6-1
6.1	Richtlinien für das Entwerfen einer Automatisierungslösung mit einer Micro-SPS	6-2
6.2	S7-200 Programme	6-4
6.3	S7-200 Programmiersprachen	6-5
6.4	Grundlegende Elemente zum Entwerfen eines Programms	6-8
6.5	Der Zyklus der CPU	6-10
6.6	Einstellen der Betriebsart für die CPU	6-13
6.7	Einrichten eines Paßworts für die CPU	6-14
6.8	Testen und Überwachen Ihres Programms	6-16
6.9	Fehlerbehebung bei der S7-200 CPU	6-19
7	Speicher der CPU: Datentypen und Adressierungsarten	7-1
7.1	Direkte Adressierung der Speicherbereiche in der CPU	7-2
7.2	Indirekte Adressierung der Speicherbereiche in der CPU	7-9
7.3	Datenhaltung in der S7-200 CPU	7-11
7.4	Nullspannungsfestes Speichern von Daten mit Ihrem Programm	7-16
7.5	Speichern Ihres Programms im Speichermodul	7-17
8	Steuerung über Ein- und Ausgänge	8-1
8.1	Integrierte und erweiterte Ein- und Ausgänge	8-2
8.2	Konfigurieren von Eingabefiltern zur Rauschunterdrückung	8-5
8.3	Konfigurieren der Signalzustände der Ausgänge	8-6
8.4	Schnelle Ein- und Ausgänge	8-7
8.5	Analogpotentiometer	8-8
9	Kommunikation im Netz mit einer S7-200 CPU	9-1
9.1	Kommunikationsfähigkeiten der S7-200 CPU	9-2
9.2	Komponenten für die Kommunikation im Netz	9-6
9.3	Datenkommunikation über PC/PPI-Kabel	9-9
9.4	Datenkommunikation über MPI- oder CP-Baugruppe	9-13
9.5	DP-Standardkommunikation (dezentrale Peripherie)	9-15
9.6	Leistungsfähigkeit des Netzes	9-28
10	Operationssatz	10-1
10.1	Gültige Bereiche für die S7-200 CPUs	10-2
10.2	Operationen mit Kontakten	10-4
10.3	Operationen mit Vergleichskontakten	10-7
10.4	Operationen mit Ausgängen	10-10

10.5	Operationen mit Zeiten, Zählern, schnellen Zählern, schnellen Ausgängen und Impulsausgaben	10-13
10.6	Arithmetische Operationen und Operationen für den PID-Regler	10-50
10.7	Operationen zum Inkrementieren und Dekrementieren	10-66
10.8	Übertragungs- und Tabellenoperationen	10-68
10.9	Schiebe- und Rotieroperationen	10-78
10.10	Operationen für die Programmsteuerung	10-84
10.11	Stackoperationen	10-99
10.12	Verknüpfungsoperationen	10-102
10.13	Umwandlungsoperationen	10-108
10.14	Interrupt- und Kommunikationsoperationen	10-114
A	Technische Daten	A-1
A.1	Allgemeine technische Daten	A-3
A.2	CPU 212, DC-Versorgung, DC-Eingänge, DC-Ausgänge	A-6
A.3	CPU 212, AC-Versorgung, DC-Eingänge, Relaisausgänge	A-8
A.4	CPU 212, 24-V-AC-Versorgung, DC-Eingänge, Relaisausgänge	A-10
A.5	CPU 212, AC-Versorgung, AC-Eingänge, AC-Ausgänge	A-12
A.6	CPU 212 AC-Versorgung, DC-Eingänge (stromliefernd), Relaisausgänge	A-14
A.7	CPU 212, AC-Versorgung, 24-V-AC-Eingänge, AC-Ausgänge	A-16
A.8	CPU 212, AC-Versorgung, AC-Eingänge, Relaisausgänge	A-18
A.9	CPU 214, DC-Versorgung, DC-Eingänge, DC-Ausgänge	A-20
A.10	CPU 214, AC-Versorgung, DC-Eingänge, Relaisausgänge	A-22
A.11	CPU 214, AC-Versorgung, AC-Eingänge, AC-Ausgänge	A-24
A.12	CPU AC-Versorgung, DC-Eingänge (stromliefernd), Relaisausgänge	A-26
A.13	CPU 214, AC-Versorgung, 24-V-AC-Eingänge, AC-Ausgänge	A-28
A.14	CPU 214, AC-Versorgung, AC-Eingänge, Relaisausgänge	A-30
A.15	CPU 215, DC-Versorgung, DC-Eingänge, DC-Ausgänge	A-32
A.16	CPU 215, AC-Versorgung, DC-Eingänge, Relaisausgänge	A-34
A.17	CPU 216, DC-Versorgung, DC-Eingänge, DC-Ausgänge	A-36
A.18	CPU 216, AC-Versorgung, DC-Eingänge, Relaisausgänge	A-38
A.19	Erweiterungsmodul EM 221 Digitaleingabe 8 x 24 V DC	A-40
A.20	Erweiterungsmodul EM 221, Digitaleingabe 8 x 120 V AC	A-41
A.21	Erweiterungsmodul EM 221 Digitaleingabe (stromliefernd) 8 x 24 V DC	A-42
A.22	Erweiterungsmodul EM 221, Digitaleingabe 8 x 24 V AC	A-43
A.23	Erweiterungsmodul EM 222, Digitalausgabe 8 x 24 V DC	A-44
A.24	Erweiterungsmodul EM 222, Digitalausgabe 8 x Relais	A-45
A.25	Erweiterungsmodul EM 222, Digitalausgabe 8 x 120/230 V AC	A-46
A.26	Erweiterungsmodul EM 223 Digitalein-/Digitalausgabe 4 x 24-V-DC-Eingang/4 x 24-V-DC-Ausgang	A-48

A.27	Erweiterungsmodul EM 223, Digitalein-/Digitalausgabe 8 x 24-V-DC-Eingang/8 x 24-V-DC-Ausgang	A-50
A.28	Erweiterungsmodul EM 223, Digitalein-/Digitalausgabe 16 x 24-V-DC-Eingang/16 x 24-V-DC-Ausgang	A-52
A.29	Erweiterungsmodul EM 223, Digitalein-/Digitalausgabe 4 x 24-V-DC-Eingang/4 x Relaisausgang	A-54
A.30	Erweiterungsmodul EM 223, Digitalein-/Digitalausgabe 4 x 120-V-AC-Eingang/4 x 120/230-V-AC-Ausgang	A-55
A.31	Erweiterungsmodul EM 223, Digitalein-/Digitalausgabe 8 x 24-V-DC-Eingang/8 x Relaisausgang	A-56
A.32	Erweiterungsmodul EM 223, Digitalein-/Digitalausgabe 16 x 24-V-DC-Eingang/16 x Relaisausgang	A-58
A.33	Erweiterungsmodul EM 231, Analogeingabe AE 3 x 12 Bit	A-60
A.34	Erweiterungsmodul EM 232, Analogausgabe AA 2 x 12 Bit	A-66
A.35	Erweiterungsmodul EM 235, Analogeing-/Analogausgabe AE 3 / AA 1 x 12 Bit ...	A-69
A.36	Speichermodul 8 K x 8	A-78
A.37	Speichermodul 16 K x 8	A-79
A.38	Batteriemodul	A-80
A.39	Steckleitung für Erweiterungsmodule	A-81
A.40	PC/PPI-Kabel	A-82
A.41	DC-Eingangssimulator für die CPU 212	A-84
A.42	DC-Eingangssimulator für die CPU 214	A-85
A.43	DC-Eingangssimulator für die CPU 215/216	A-86
B	Berechnungstabelle Strombilanz	B-1
C	Fehlermeldungen	C-1
C.1	Fehlercodes und Meldungen von schweren Fehlern	C-2
C.2	Fehler zur Laufzeit	C-3
C.3	Verletzungen der Übersetzungsregeln	C-4
D	Sondermerker	D-1
E	Einsetzen von STEP 7-Micro/WIN mit STEP 7 und STEP 7-Micro/DOS	E-1
E.1	Arbeiten mit STEP 7-Micro/WIN und STEP 7	E-2
E.2	Importieren von Dateien aus STEP 7-Micro/DOS	E-4
F	Ausführungszeiten von AWL-Operationen	F-1
G	S7-200 Bestellnummern	G-1
H	S7-200 Fehlerbehebung	H-1
	Index	Index-1

Einführung in die Micro-SPS S7-200

Die Familie S7-200 umfaßt verschiedene Kleinsteuerungen (Micro-SPS), mit denen Sie eine breite Palette von Automatisierungsaufgaben lösen können. Bild 1-1 zeigt eine Micro-SPS S7-200. Durch das kompakte Design, die Möglichkeit der Erweiterung, den günstigen Preis und einen leistungsstarken Befehlssatz eignet sich die Familie S7-200 hervorragend für kleinere Steuerungsanwendungen. Die zahlreichen Möglichkeiten an Größen und Spannungsversorgungen bei der Wahl der CPU bieten Ihnen extrem hohe Flexibilität beim Umsetzen Ihrer Automatisierungslösungen.

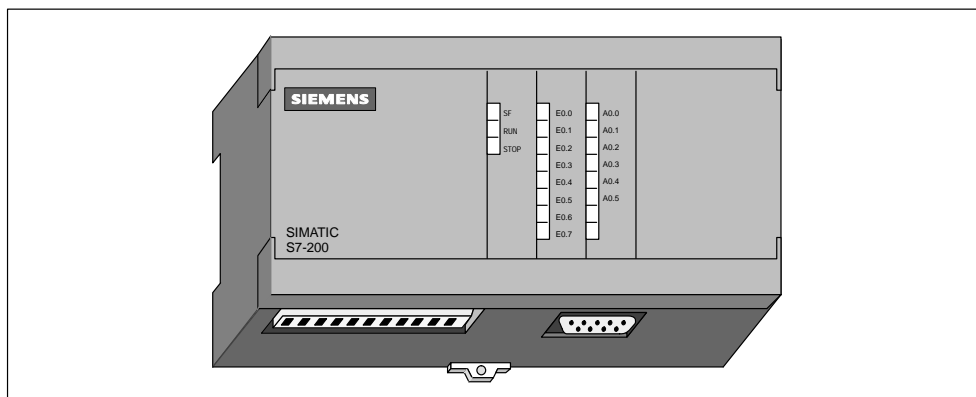


Bild 1-1 Micro-SPS S7-200

Kapitelübersicht

Abschnitt	Beschreibung	Seite
1.1	Funktionen der verschiedenen Micro-SPS S7-200	1-2
1.2	Hauptkomponenten einer Micro-SPS S7-200	1-4

1.1 Funktionen der verschiedenen Micro-SPS S7-200

Anforderungen an die Ausrüstung

Bild 1-2 zeigt den grundlegenden Aufbau eines Automatisierungssystems mit einer Micro-SPS S7-200. Das System umfaßt ein Zentralgerät S7-200 (CPU), einen Personal Computer, die Programmiersoftware STEP 7-Micro/WIN und ein Kommunikationskabel.

Damit Sie einen Personal Computer (PC) verwenden können, benötigen Sie eine der folgenden Ausrüstungen:

- ein PC/PPI-Kabel
- einen Kommunikationsprozessor (CP) und MPI-Kabel
- eine MPI-Baugruppe. Das Kommunikationskabel ist im Lieferumfang der MPI-Baugruppe enthalten.

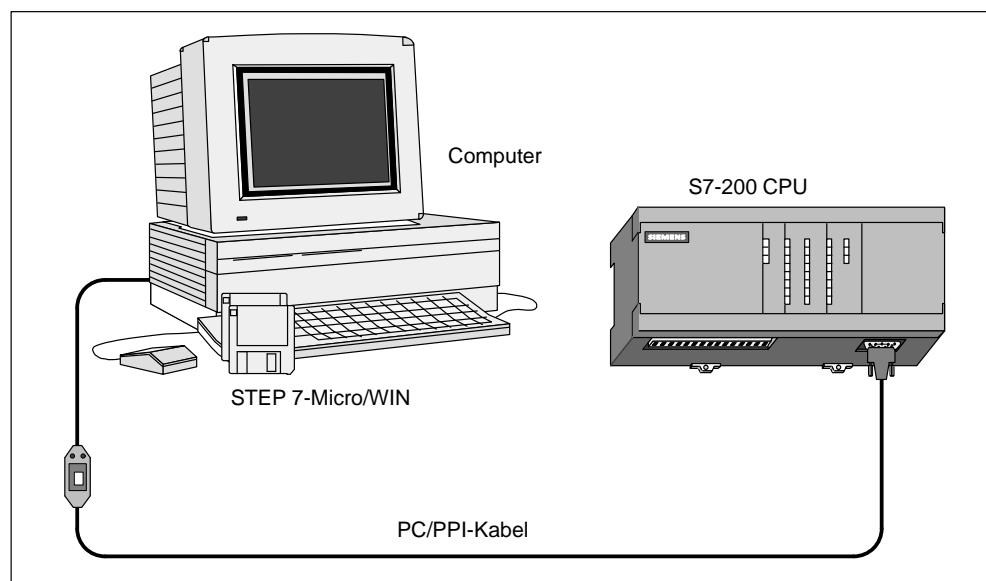


Bild 1-2 Komponenten eines Automatisierungssystems mit einer Micro-SPS S7-200

Übersicht über den Leistungsumfang der S7-200 CPUs

Die Familie S7-200 umfaßt eine Reihe verschiedener CPUs. Durch diese Vielzahl der CPUs wird eine Bandbreite von Funktionalitäten zur Verfügung gestellt, die den Entwurf kostengünstiger Automatisierungslösungen gestattet. Tabelle 1-1 bietet einen Überblick über die wesentlichen Funktionalitäten der einzelnen S7-200 CPUs.

Tabelle 1-1 Überblick über die S7-200 CPUs

Funktionalität	CPU 212	CPU 214	CPU 215	CPU 216
Physikalische Größe	160 mm x 80 mm x 62 mm	197 mm x 80 mm x 62 mm	218 mm x 80 mm x 62 mm	218 mm x 80 mm x 62 mm
Speicher				
Programm (EEPROM)	512 Wörter	2 K Wörter	4 K Wörter	4 K Wörter
Anwenderdaten	512 Wörter	2 K Wörter	2,5 K Wörter	2,5 K Wörter
Interne Merker	128	256	256	256
Speichermodul	keine	Ja (EEPROM)	Ja (EEPROM)	Ja (EEPROM)
Batteriemodul (optional)	keine	typ. 200 Tage	typ. 200 Tage	typ. 200 Tage
Hochleistungskondensator	typ. 50 Std.	typ. 190 Std.	typ. 190 Std.	typ. 190 Std.
Ein- und Ausgänge (E/A)				
Integrierte E/A	8 DE/6 DA	14 DE/10 DA	14 DE/10 DA	24 DE/16 DA
Anz. Erweiterungsmodule (max.)	2 Module	7 Module	7 Module	7 Module
Prozeßabbild E/A	64 DE/64 DA	64 DE/64 DA	64 DE/64 DA	64 DE/64 DA
Analoge E/A (Erweiterung)	16 AE/16 AA	16 AE/16 AA	16 AE/16 AA	16 AE/16 AA
Eingabefilter	Nein	Ja	Ja	Ja
Operationen				
Boolesche Ausführungszeiten	1,2 µs/Operation	0,8 µs/Operation	0,8 µs/Operation	0,8 µs/Operation
Zeiten/Zähler	64/64	128/128	256/256	256/256
Schleifen mit FOR/NEXT	Nein	Ja	Ja	Ja
Festpunktarithmetik	Ja	Ja	Ja	Ja
Gleitpunktarithmetik	Nein	Ja	Ja	Ja
PID-Regler	Nein	Nein	Ja	Ja
Zusätzliche Funktionalitäten				
Schneller Zähler	1 SW	1 SW, 2 HW	1 SW, 2 HW	1 SW, 2 HW
Analogpotentiometer	1	2	2	2
Impulsausgänge	keine	2	2	2
Kommunikationsinterrupts	1 Senden/ 1 Empfangen	1 Senden/ 1 Empfangen	1 Senden/ 2 Empfangen	2 Senden/ 4 Empfangen
Zeitgesteuerte Interrupts	1	2	2	2
Hardware-Interrupteingänge	1	4	4	4
Echtzeituhr	keine	Ja	Ja	Ja
Kommunikation				
Anzahl Schnittstellen:	1 (RS-485)	1 (RS-485)	2 (RS-485)	2 (RS-485)
Unterstützte Protokolle Schnittstelle 0: Schnittstelle 1:	PPI, frei progr. -/- -/-	PPI, frei progr. -/- -/-	PPI, frei prog. Komm., MPI DP, MPI	PPI, frei prog. Komm., MPI PPI, frei prog. Komm., MPI
Punkt-zu-Punkt	Nur Slave	Ja	Ja	Ja

1.2 Hauptkomponenten einer Micro-SPS S7-200

Eine Micro-SPS S7-200 besteht aus einer S7-200 CPU und optionalen Erweiterungsmodulen.

Zentralgerät S7-200

Die Zentralbaugruppe S7-200 ist ein kompaktes Gerät und besteht aus einer Zentraleinheit (CPU), der Spannungsversorgung und digitalen Ein- und Ausgängen.

- Die CPU bearbeitet das Programm und speichert die Daten für die Automatisierungslösung bzw. den Prozeß.
- Die Spannungsversorgung liefert den Strom für das Zentralgerät und alle angeschlossenen Erweiterungsmodule.
- Über die Ein- und Ausgänge wird das System gesteuert. Die Eingänge überwachen die Signale der Feldgeräte (z. B. Sensoren und Schalter) und die Ausgänge steuern Pumpen, Motoren oder andere Geräte in Ihrem Prozeß.
- Über die Kommunikationsschnittstelle können Sie ein Programmiergerät oder andere Geräte an die CPU anschließen. Einige S7-200 CPUs verfügen über zwei Kommunikationsschnittstellen.
- Die Statusanzeigen liefern Ihnen visuelle Informationen über die Betriebsart der CPU (RUN oder STOP), den aktuellen Signalzustand der integrierten Ein- und Ausgänge und eventuelle Systemfehler.

Die Bilder 1-3, 1-4 und 1-5 zeigen die verschiedenen S7-200 CPUs.

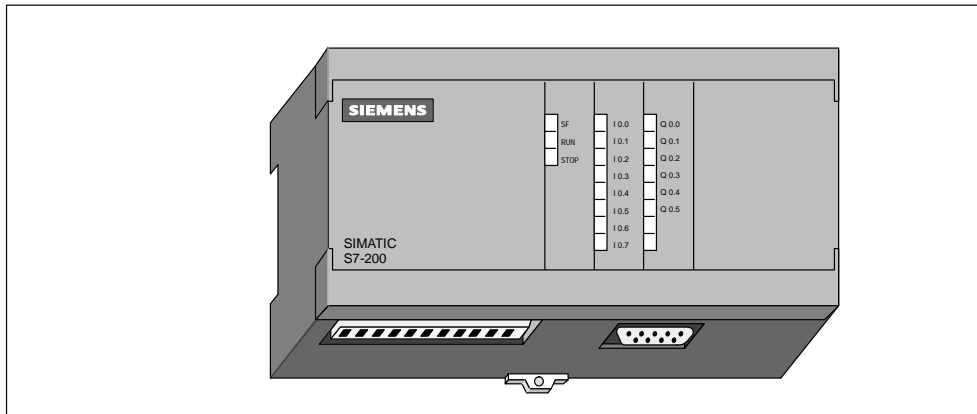


Bild 1-3 CPU S7-212

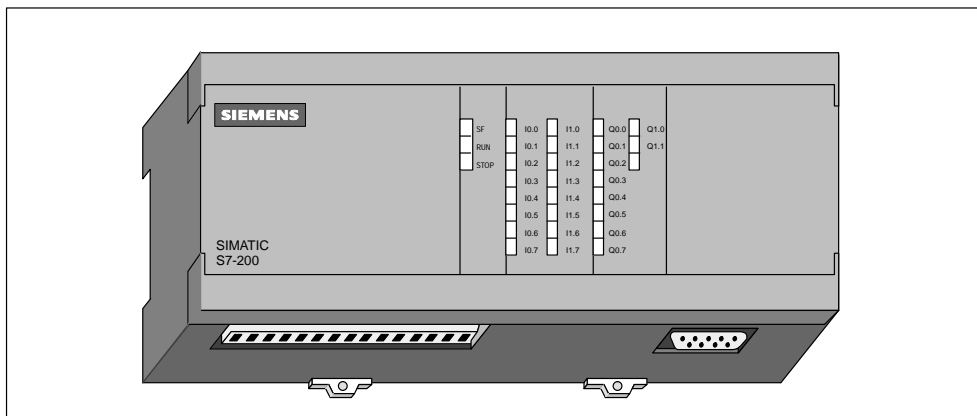


Bild 1-4 CPU S7-214

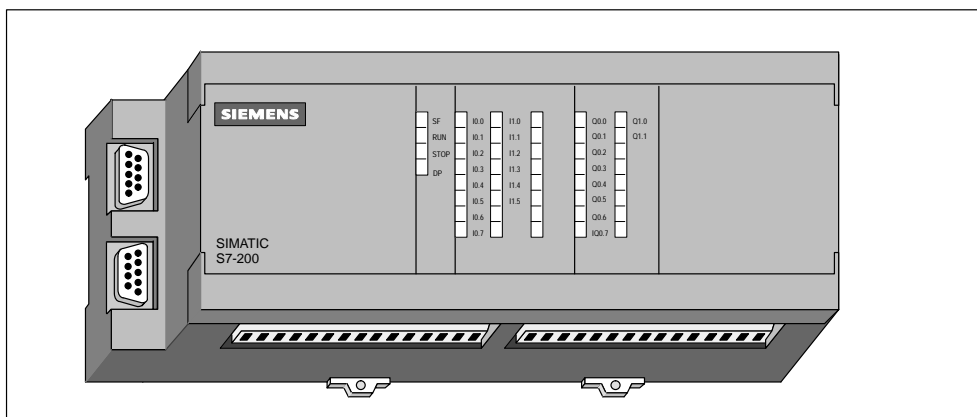


Bild 1-5 CPU S7-215 und CPU S7-216

Erweiterungsmodul

Das Zentralgerät S7-200 verfügt über eine bestimmte Anzahl integrierter Ein- und Ausgänge. Mittels eines Erweiterungsmoduls können sie zusätzliche Ein- und Ausgänge bereitstellen. Bild 1-6 zeigt ein Erweiterungsmodul und den im Lieferumfang enthaltenen Busverbinder, mit dem Sie das Erweiterungsmodul an das Zentralgerät anschließen.

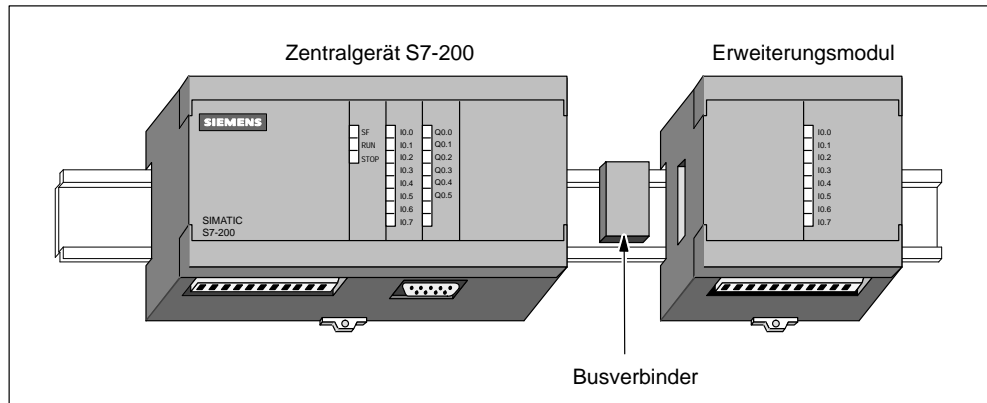


Bild 1-6 CPU mit einem Erweiterungsmodul

2

Installieren einer Micro-SPS S7-200

Die Geräte der Familie S7-200 wurden so ausgelegt, daß sie einfach zu installieren sind. Mittels der Bohrungen können Sie die Module in eine Schalttafel einbauen. Sie können die Module aber auch mit den dafür vorgesehenen Rasthaken auf einer Standard-Hutschiene (DIN) montieren. Die kompakte Größe der S7-200 macht eine effiziente Platzausnutzung möglich.

In diesem Kapitel erfahren Sie, wie Sie Ihr Automatisierungssystem S7-200 installieren und verdrahten.

Kapitelübersicht

Abschnitt	Beschreibung	Seite
2.1	Vorbereitungen für die Montage	2-2
2.2	Ein- und Ausbauen einer Micro-SPS S7-200	2-5
2.3	Installieren der Feldverdrahtung	2-8
2.4	Schutzbeschaltungen	2-13
2.5	Spannungsversorgung	2-15

2.1 Vorbereitungen für die Montage

Anordnung der Geräte

Sie können die Geräte der S7-200 in eine Schalttafel einbauen oder auf einer Hutschiene montieren. Sie können die Module sowohl horizontal als auch vertikal anordnen. Eine Steckleitung für Erweiterungsmodule bietet Ihnen bei der Montage zusätzliche Flexibilität. Bild 2-1 zeigt die typischen Geräteanordnungen für Schalttafel und Hutschiene.

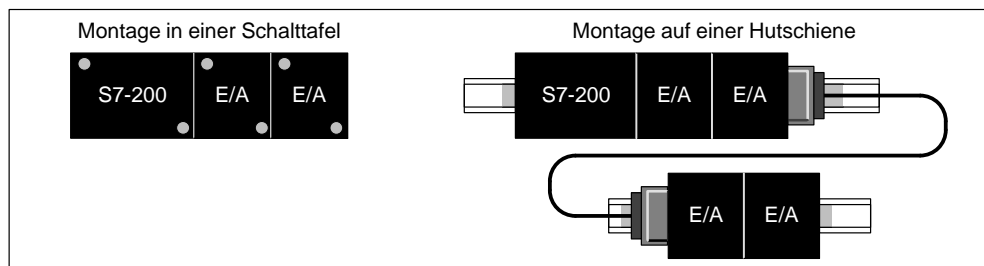


Bild 2-1 Geräteanordnungen für die Montage

Platzbedarf bei der Montage einer S7-200

Beachten Sie beim Entwerfen Ihrer Geräteanordnung die folgenden Richtlinien:

- Die S7-200 CPU und die Erweiterungsmodule sind auf natürliche Wärmeabfuhr durch Konvektion ausgerichtet. Lassen Sie deshalb oberhalb und unterhalb der Geräte jeweils mindestens 25 mm Platz, um die Wärmeabfuhr zu gewährleisten (siehe Bild 2-2). Lange Betriebszeiten bei maximaler Umgebungstemperatur und maximaler Last verkürzen die Lebensdauer der elektronischen Gerätekomponenten.
- Bei vertikalem Einbau muß die Ausgangslast wegen Wärmebedingungen verringert werden. Den Datenblättern in Anhang A entnehmen Sie genaue Informationen zu Ihrer CPU. Montieren Sie die CPU und die Module auf einer Hutschiene, empfehlen wir Ihnen, Erdungsklemmen einzusetzen.
- Wenn Sie die S7-200 horizontal oder vertikal in eine Schalttafel einbauen, benötigen Sie eine Einbautiefe von mindestens 75 mm (siehe Bild 2-2).
- Möchten Sie zusätzliche Erweiterungsmodule vertikal oder horizontal einbauen, müssen Sie auf jeder Seite des Geräts mindestens 25 mm Platz zum Ein- und Ausbauen der Module lassen. Diesen Platz benötigen Sie, um den Busverbinder zu stecken bzw. zu ziehen.
- Bauen Sie Ihre Geräte so auf, daß noch ausreichend Platz für die Verdrahtung der Ein- und Ausgänge und für die Anschlüsse der Kommunikationskabel vorhanden ist.

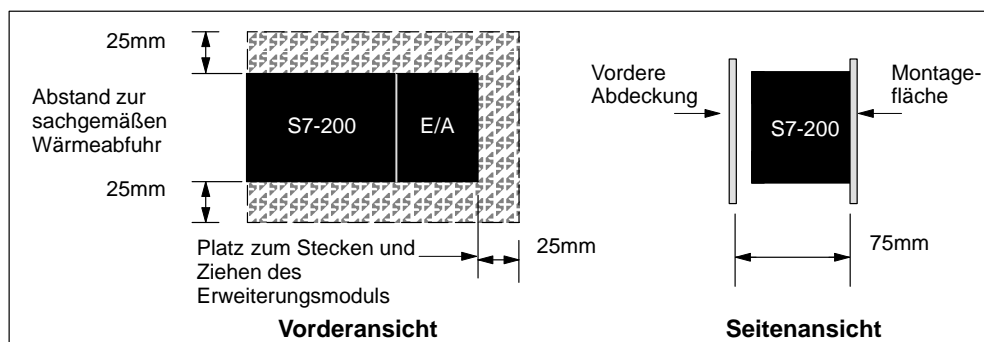


Bild 2-2 Platzbedarf bei der Montage einer S7-200 CPU

Anforderungen an die Hutschiene

Die S7-200 CPUs und die Erweiterungsmodule können auf einer Hutschiene (DIN EN 50 022) montiert werden. Bild 2-3 zeigt die Abmessungen einer Standard-Hutschiene.

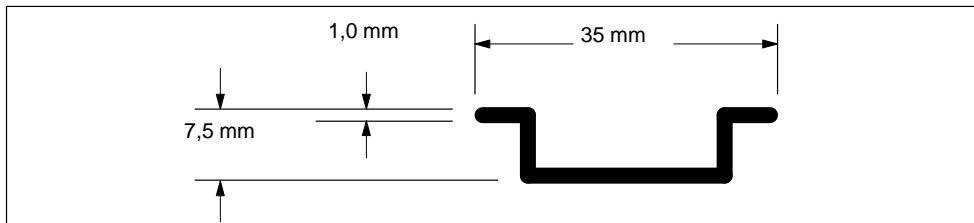


Bild 2-3 Abmessungen einer Standard-Hutschiene

Abmessungen für den Einbau in eine Schalttafel

Die S7-200 CPUs und die Erweiterungsmodule sind mit Bohrungen versehen, die den Einbau in eine Schalttafel erleichtern. Die Bilder 2-4 bis 2-8 zeigen die Abmessungen für die verschiedenen Geräte der S7-200 bei Montage in einer Schalttafel.

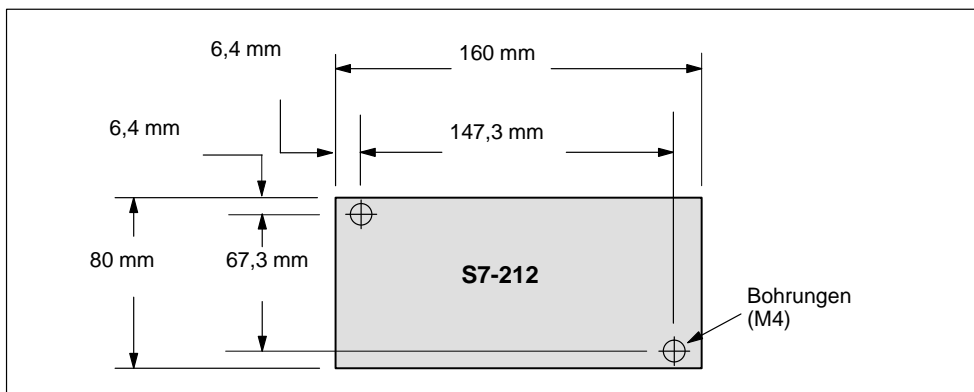


Bild 2-4 Abmessungen für die Montage der CPU S7-212

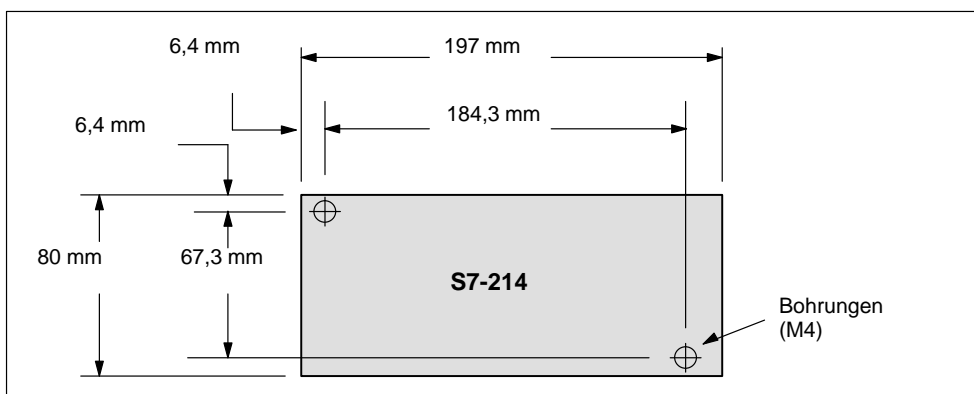


Bild 2-5 Abmessungen für die Montage der CPU S7-214

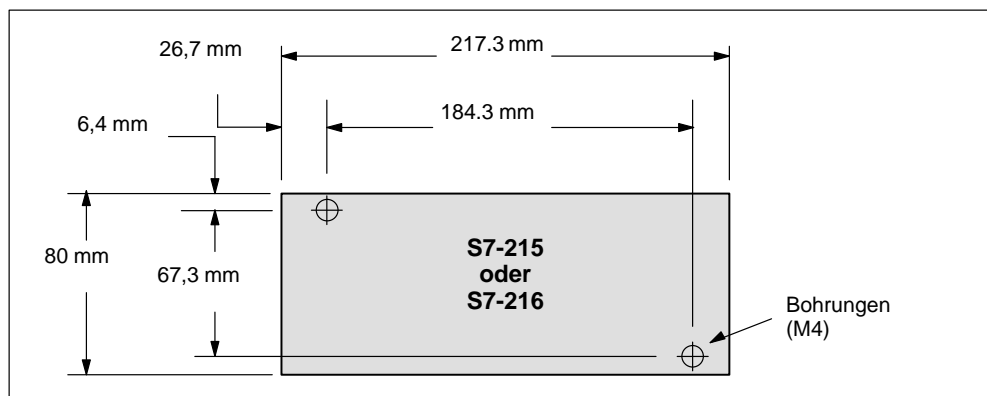


Bild 2-6 Abmessungen für die Montage der CPUs S7-215 und S7-216

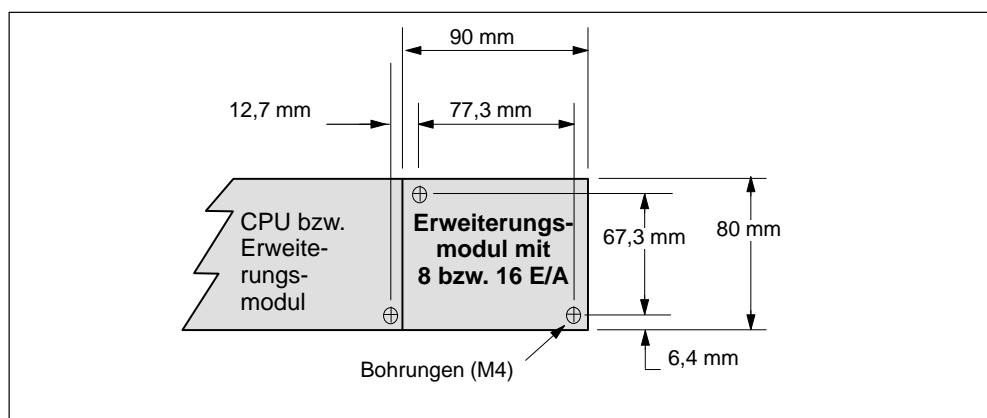


Bild 2-7 Abmessungen für die Montage eines Erweiterungsmoduls mit 8 bzw. 16 Ein- und Ausgängen

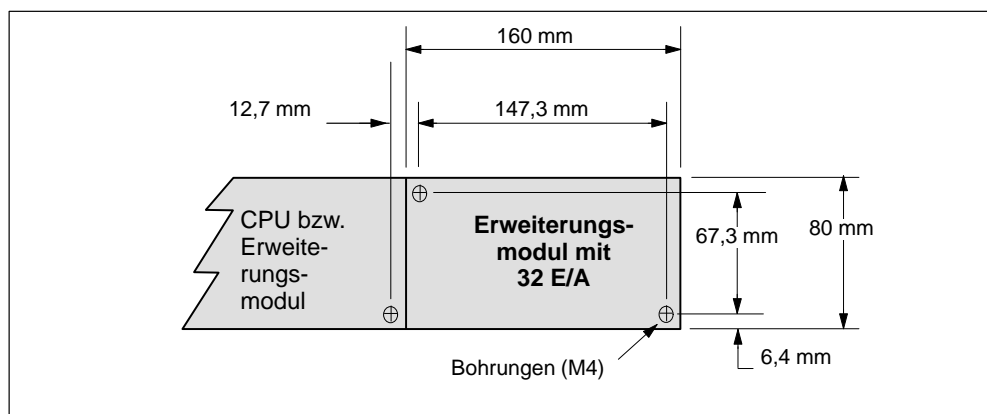


Bild 2-8 Abmessungen für die Montage eines Erweiterungsmoduls mit 32 Ein- und Ausgängen

2.2 Ein- und Ausbauen einer Micro-SPS S7-200

Montage einer Micro-SPS S7-200 in einer Schalttafel



Warnung

Wenn Sie versuchen, ein S7-200 Modul oder andere Geräte in eingeschaltetem Zustand ein- oder auszubauen, können Sie einen elektrischen Schlag bekommen.

Ist die Spannungsversorgung für die S7-200 und alle angeschlossenen Geräte während des Ein- bzw. Ausbaus von Geräten nicht abgeschaltet, so kann dies zu tödlichen oder schweren Verletzungen und/oder Sachschaden führen.

Treffen Sie alle notwendigen Sicherheitsvorkehrungen und vergewissern Sie sich, daß vor der Installation eines Geräts die Spannungsversorgung für die Geräte der S7-200 abgeschaltet ist.

Gehen Sie zum Installieren eines S7-200 Zentralgeräts folgendermaßen vor:

1. Bringen Sie die Bohrungen für Schrauben der Größe DIN M4 in der Schalttafel an. Richten Sie sich hierzu nach den Abmessungen und Hinweisen, die in Abschnitt 2.1 für den Einbau in eine Schalttafel angegeben sind.
2. Schrauben Sie die Geräte der S7-200 in der Schalttafel fest. Verwenden Sie hierzu Schrauben der Größe DIN M4.

Gehen Sie zum Installieren eines Erweiterungsmoduls folgendermaßen vor:

1. Brechen Sie die Abdeckung des Buserweiterungsanschlusses aus dem Gehäuse des Erweiterungsmoduls heraus. Führen Sie hierzu einen Schraubendreher in den Spalt zwischen Abdeckung und Gehäuse ein und hebeln Sie ihn vorsichtig nach oben. Achten Sie darauf, daß Sie sämtliche Kunststoffreste entfernen und das Gerät nicht beschädigen. Bild 2-9 zeigt, wie Sie den Schraubendreher ansetzen müssen.
2. Stecken Sie den Busverbinder in den Busanschluß des Geräts und vergewissern Sie sich, daß der Verbinder richtig einrastet.
3. Stellen Sie sicher, daß das Modul richtig ausgerichtet ist. Wenn Sie eine Steckleitung für Erweiterungsmodule verwenden, stecken Sie das Kabel so in das Modul, daß die Oberseite des Kabels nach vorne zeigt.
4. Verbinden Sie nun Erweiterungsmodul und Busverbinder, indem Sie das Erweiterungsmodul auf den Busverbinder schieben, bis es einrastet.

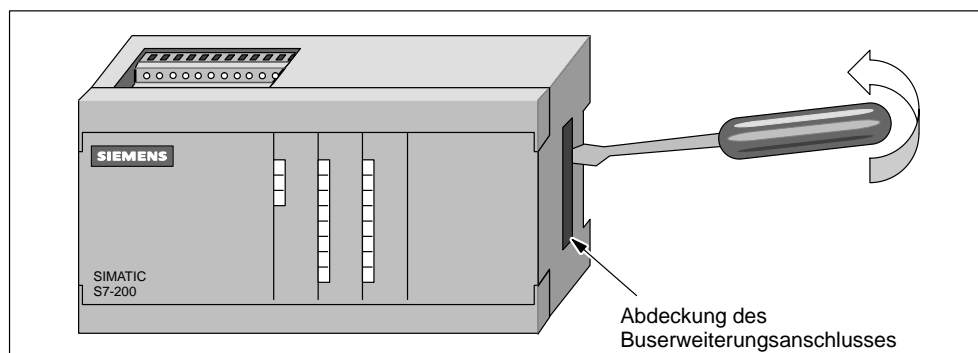


Bild 2-9 Herausbrechen der Abdeckung des Buserweiterungsanschlusses an einer S7-200 CPU

Montage einer Micro-SPS S7-200 auf einer Hutschiene



Warnung

Wenn Sie versuchen, ein S7-200 Modul oder andere Geräte in eingeschaltetem Zustand ein- oder auszubauen, können Sie einen elektrischen Schlag bekommen.

Ist die Spannungsversorgung für die S7-200 und alle angeschlossenen Geräte während des Ein- bzw. Ausbaus von Geräten nicht abgeschaltet, so kann dies zu tödlichen oder schweren Verletzungen und/oder Sachschaden führen.

Treffen Sie alle notwendigen Sicherheitsvorkehrungen und vergewissern Sie sich, daß vor der Installation eines Geräts die Spannungsversorgung für die Geräte der S7-200 abgeschaltet ist.

Gehen Sie zum Montieren einer S7-200 CPU folgendermaßen vor:

1. Verschrauben Sie die Hutschiene im Abstand von jeweils 75 mm mit der Schalttafel.
2. Öffnen Sie den Rasthaken (an der Unterseite des Geräts) und hängen Sie das Gerät mit der Rückseite auf der Hutschiene ein.
3. Schließen Sie den Rasthaken und achten Sie darauf, daß der Haken richtig einrastet und das Gerät sicher auf der Schiene befestigt ist.

Hinweis

In Umgebungen, in denen starke Schwingungen auftreten, oder bei vertikalem Einbau der Geräte kann es erforderlich sein, die Geräte auf der Hutschiene mit Erdungsklemmen vor dem Verrutschen zu sichern.

Gehen Sie zum Installieren eines Erweiterungsmoduls folgendermaßen vor:

1. Brechen Sie die Abdeckung des Buserweiterungsanschlusses aus dem Gehäuse des Erweiterungsmoduls heraus. Führen Sie hierzu einen Schraubendreher in den Spalt zwischen Abdeckung und Gehäuse ein und hebeln Sie ihn vorsichtig nach oben. Achten Sie darauf, daß Sie sämtliche Kunststoffreste entfernen und das Gerät nicht beschädigen. Bild 2-9 zeigt, wie Sie den Schraubendreher ansetzen müssen.
2. Stecken Sie den Busverbinder in den Busanschluß des Geräts und vergewissern Sie sich, daß der Verbinder richtig einrastet.
3. Stellen Sie sicher, daß das Modul richtig ausgerichtet ist. Wenn Sie eine Steckleitung für Erweiterungsmodule verwenden, stecken Sie das Kabel so in das Modul, daß die Oberseite des Kabels nach vorne zeigt.
4. Öffnen Sie den Rasthaken und hängen Sie das Erweiterungsmodul mit der Rückseite auf der Hutschiene ein. Schieben Sie das Erweiterungsmodul auf den Busverbinder, bis das Modul einrastet.
5. Schließen Sie den Rasthaken, um das Erweiterungsmodul auf der Schiene zu befestigen. Achten Sie darauf, daß der Haken richtig einrastet und das Gerät sicher auf der Schiene befestigt ist.

Ausbauen eines S7-200 Moduls



Warnung

Wenn Sie versuchen, ein S7-200 Modul oder andere Geräte in eingeschaltetem Zustand ein- oder auszubauen, können Sie einen elektrischen Schlag bekommen.

Ist die Spannungsversorgung für die S7-200 und alle angeschlossenen Geräte während des Ein- bzw. Ausbaus von Geräten nicht abgeschaltet, so kann dies zu tödlichen oder schweren Verletzungen und/oder Sachschaden führen.

Treffen Sie alle notwendigen Sicherheitsvorkehrungen und vergewissern Sie sich, daß vor dem Ein- bzw. Ausbau eines Geräts die Spannungsversorgung für die Geräte der S7-200 abgeschaltet ist.

Zum Ausbauen der S7-200 CPU oder eines Erweiterungsmoduls der S7-200 gehen Sie folgendermaßen vor:

1. Lösen Sie alle Kabel und Leitungen von dem Gerät, das Sie ausbauen möchten. Befindet sich das Gerät in der Mitte des Aufbaus, so müssen Sie zunächst die Geräte rechts und links davon um mindestens 25 mm verschieben, damit der Busverbinder herausgezogen werden kann (siehe Bild 2-10).
2. Lösen Sie die Befestigungsschrauben bzw. öffnen Sie den Rasthaken und verschieben Sie das Gerät um mindestens 25 mm, so daß der Busverbinder aus dem Busanschluß herausgezogen wird. Sie müssen die Busverbinder auf beiden Seiten des Geräts herausziehen.
3. Nehmen Sie das Gerät aus der Schalttafel bzw. von der Hutschiene und installieren Sie ein neues Modul.



Warnung

Wenn Sie ein falsches Gerät installieren, kann das Programm der Micro-SPS S7-200 unvorhersehbare Auswirkungen haben.

Wird ein Erweiterungsmodul durch eine andere Variante ersetzt oder nicht richtig ausgerichtet, so kann dies zu tödlichen oder schweren Verletzungen und/oder Sachschaden führen.

Ersetzen Sie daher ein Erweiterungsmodul immer durch die gleiche Variante und richten Sie es korrekt aus. Wenn Sie eine Steckleitung für Erweiterungsmodule verwenden, stecken Sie das Kabel so in das Modul, daß die Oberseite des Kabels nach vorne zeigt.

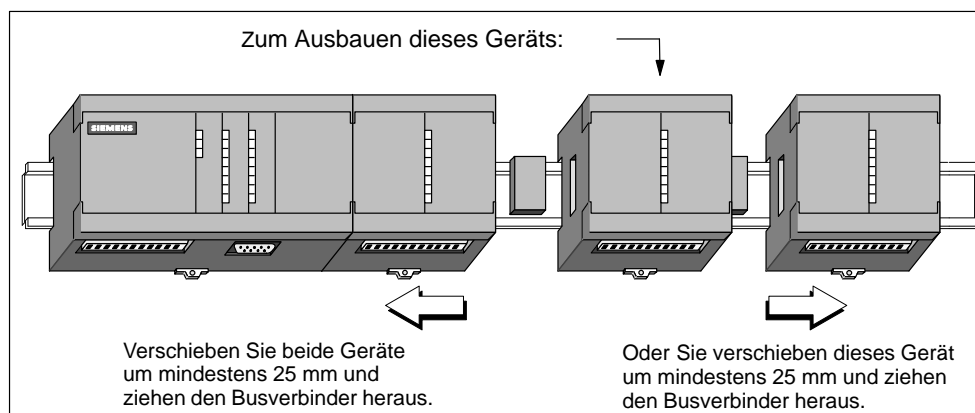


Bild 2-10 Ausbauen eines Erweiterungsmoduls

2.3 Installieren der Feldverdrahtung



Warnung

Wenn Sie versuchen, ein S7-200 Modul oder andere Geräte in eingeschaltetem Zustand ein- oder auszubauen, können Sie einen elektrischen Schlag bekommen.

Ist die Spannungsversorgung für die S7-200 und alle angeschlossenen Geräte während des Ein- bzw. Ausbaus von Geräten nicht abgeschaltet, so kann dies zu tödlichen oder schweren Verletzungen und/oder Sachschaden führen.

Treffen Sie alle notwendigen Sicherheitsvorkehrungen und vergewissern Sie sich, daß vor der Verdrahtung die Spannungsversorgung für die Geräte der S7-200 abgeschaltet ist.

Allgemeine Richtlinien

Die folgenden Richtlinien sollten Sie bei Installation und Verdrahtung Ihrer Micro-SPS S7-200 beachten:

- Stellen Sie sicher, daß Sie bei der Verdrahtung der Micro-SPS S7-200 alle geltenden und verbindlichen Normen befolgen. Beachten Sie bei der Installation und beim Betrieb der Geräte die entsprechenden nationalen und regionalen Vorschriften. Erfragen Sie bei den Behörden vor Ort die Normen und Vorschriften, die für Ihren speziellen Fall zu befolgen sind.
- Verwenden Sie Leitungen mit dem angemessenen Querschnitt für die jeweilige Stromstärke. Die Geräte der S7-200 können Sie mit Leitungen mit einem Querschnitt zwischen 1,50 mm² und 0,50 mm² verdrahten.
- Ziehen Sie die Anschlußklemmen nicht zu fest an. Maximales Drehmoment: 0,56 Nm.
- Verlegen Sie Leitungen immer so kurz wie möglich (maximal 500 m geschirmt oder 300 m ungeschirmt). Leitungen sollten Sie paarweise verlegen: ein Neutral- oder Nulleiter zusammen mit einem Phasenleiter oder einer Signalleitung.
- Trennen Sie Wechselstromverdrahtung und Hochspannungs-Gleichstromverdrahtung mit schnellen Schaltfolgen von Niederspannungs-Signalverdrahtung.
- Kennzeichnen Sie alle Leitungen und führen Sie sie zu dem Gerät, das angeschlossen werden soll. Vergewissern Sie sich, daß die Leitungen die erforderliche Zugentlastung haben. Ausführliche Informationen zum Kennzeichnen der Anschlüsse entnehmen Sie den Datenblättern in Anhang A.
- Versehen Sie blitzschlaggefährdete Leitungen mit einem geeigneten Überspannungsschutz.
- Schließen Sie eine externe Spannungsversorgung nicht parallel zu einem DC-Ausgang an eine Ausgangslast an. Dadurch kann am Ausgang Rückwärtsstrom entstehen, sofern Sie den Aufbau nicht mit einer Diode oder einer ähnlichen Sperre versehen.



Warnung

Steuerungen können bei unsicheren Betriebszuständen ausfallen und dadurch den unkontrollierten Betrieb der gesteuerten Geräte verursachen.

Solche gefährlichen Ereignisse können zu tödlichen oder schweren Verletzungen und/oder Sachschaden führen.

Sorgen Sie daher für eine NOT-AUS-Funktion, elektromechanische oder andere redundante Sicherheitseinrichtungen, die von Ihrem Automatisierungssystem unabhängig sind.

Richtlinien für Erdung und Bezugsspannung galvanisch getrennter Stromkreise

Beachten Sie folgende Hinweise zur Erdung und zur Festlegung der Bezugsspannung von galvanisch getrennten Stromkreisen:

- Für jeden Stromkreis im Aufbau ist der Bezugspunkt (0 Volt) festzulegen, ferner die Punkte, an denen Stromkreise mit möglicherweise unterschiedlichen Bezugsspannungen sich treffen können. Solche Verbindungen können zu unerwünschtem Stromfluß und infolgedessen zu Fehlfunktionen oder Schäden an Schaltungen führen. Eine häufige Ursache für unterschiedliche Bezugsspannungen sind weit voneinander entfernte Erdungspunkte. Werden Geräte mit weit voneinander entfernten Erdungspunkten an ein Kommunikations- oder Sensorkabel angeschlossen, können unerwartete Ströme durch den von Kabel und Erdung gebildeten Stromkreis fließen. Selbst bei kurzen Entfernungen können die Lastströme großer Elektromaschinen Unterschiede im Erdungspotential verursachen oder durch elektromagnetische Induktion unerwünschte Ströme direkt induzieren. Spannungsversorgungen mit nicht gegeneinander ausgewogenen Bezugsspannungen können schädliche Ströme zwischen ihren jeweiligen Stromkreisen erzeugen.
- Die S7-200 Produkte arbeiten mit gruppenweiser Potentialtrennung, wodurch das Entstehen unerwünschter Ströme in Ihrer Anlage verhindert wird. Beim Planen Ihres Aufbaus müssen Sie darauf achten, wo die Potentialgrenzen liegen und wo keine vorhanden sind. Achten Sie auch auf die Potentialgrenzen von Spannungsversorgungen und anderen Geräten sowie auf die Lage der Bezugspunkte aller betroffenen Spannungsversorgungen.
- Wählen Sie die Erdungsbezugspunkte und Potentialgrenzen so, daß überflüssige Stromkreisschleifen unterbrochen und damit als unerwünschte Induktionsquellen ausgeschaltet werden. Berücksichtigen Sie auch vorübergehende Verbindungen, die neue Stromkreisreferenzen errichten, z. B. das Anschließen eines Programmiergeräts an die CPU.
- Beachten Sie beim Auswählen von Erdungspunkten die entsprechenden Sicherheitsvorschriften, und stellen Sie die einwandfreie Funktion von stromkreisunterbrechenden Schutzeinrichtungen sicher.

Die folgenden Beschreibungen stellen eine Einführung in die allgemeinen Potentialtrennungseigenschaften der Familie S7-200 dar, doch manche dieser Merkmale können bei bestimmten Produkten unterschiedlich ausgeprägt sein. Sie entnehmen den Datenblättern in Anhang A die geltenden technischen Daten zu jedem Produkt und zusätzliche Angaben, welche Stromkreise mit Potentialgrenzen versehen sind. Die Bemessungsdaten der Potentialgrenzen sind ebenfalls angegeben. Potentialgrenzen mit einer Bemessungsgrenze von unter 1500 V AC sind lediglich für die funktionsbezogene Potentialtrennung ausgelegt, sie dürfen nicht als Sicherheitsgrenzen betrachtet und genutzt werden.

- Die Referenzspannung für die Logik der CPU ist gleich der DC-Geberversorgung M.
- Bei einer CPU mit Gleichspannungsversorgung ist die Referenzspannung für die Logik der CPU gleich der Eingangsspannung M.
- Die Referenzspannung für die Kommunikationsschnittstellen der CPU ist gleich der Referenzspannung für die Logik der CPU (gilt nicht für die DP-Schnittstelle).
- Analogeingänge und Analogausgänge sind von der Logik der CPU nicht galvanisch getrennt. Analogeingänge sind völdifferential ausgeführt, um die Gleichtaktunterdrückung bei Niederspannungen sicherzustellen.
- Die Logik der CPU ist bis 100 V DC galvanisch von Erde getrennt.
- Digitale DC-Eingänge und DC-Ausgänge sind bis 500 V AC galvanisch von der Logik der CPU getrennt.
- Digitale E/A-Gruppen sind bis 500 V AC galvanisch voneinander getrennt.
- Relaisausgänge, AC-Ausgänge und AC-Eingänge sind bis 1500 V AC galvanisch von der Logik der CPU getrennt.
- Gruppen aus AC- und Relaisausgängen sind bis 1500 V AC galvanisch voneinander getrennt.
- Phasen- und Neutralleiter der AC-Spannungsversorgung sind bis 1500 V AC galvanisch von Erde, der Logik der CPU und sämtlichen Erweiterungsmodulen getrennt.

Optionaler Klemmenblock für die Feldverdrahtung

Bei dem optionalen Klemmenblock (Bild 2-11) kann die Verdrahtung beim Aus- und Wiedereinbau der S7-200 angeschlossen bleiben. Die Bestellnummer für den Klemmenblock entnehmen Sie Anhang G.

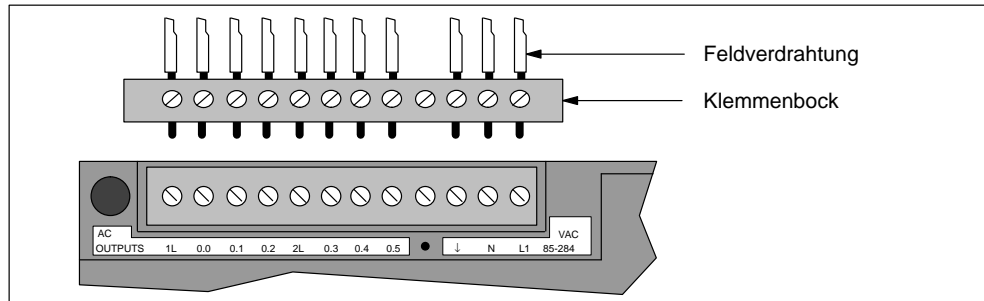


Bild 2-11 Steckbarer Klemmenblock für die Feldverdrahtung

Installationsrichtlinien für AC-Aufbau

Im folgenden sind allgemeine Installationsrichtlinien für AC-Aufbau aufgeführt. Die Richtlinien können Sie in Bild 2-12 nachvollziehen.

- Installieren Sie einen Einzeltrennschalter (1), der die Stromzufuhr zur CPU sowie zu allen Eingangs- und Ausgangstromkreisen (Laststromkreisen) unterbrechen kann.
- Installieren Sie Überstromschutzvorrichtungen für die Stromversorgung der CPU (2), für die Ausgänge und für die Eingänge. Zusätzlich können Sie jeden Ausgang einzeln absichern. Ein externer Überstromschutz für die Eingänge ist nicht nötig, wenn Sie die Geberversorgung (24 V DC) (3) der Micro-SPS einsetzen. Diese Geberversorgung ist gegen Kurzschluß geschützt.
- Verbinden Sie alle Erdungsklemmen der S7-200 mit der nächsten verfügbaren Erdung (4), um die höchstmögliche Störfestigkeit zu erreichen. Es wird empfohlen, alle Erdungsklemmen einzeln anzuschließen. Verwenden Sie hierzu Leitungen mit einem Querschnitt von 1,5 mm².
- Die Geberversorgung des Zentralgeräts kann die Eingänge (5) des Zentralgeräts, die DC-Erweiterungseingänge (6) sowie die Erweiterungs-Relaispulen (7) speisen. Diese Geberversorgung ist gegen Kurzschluß geschützt.

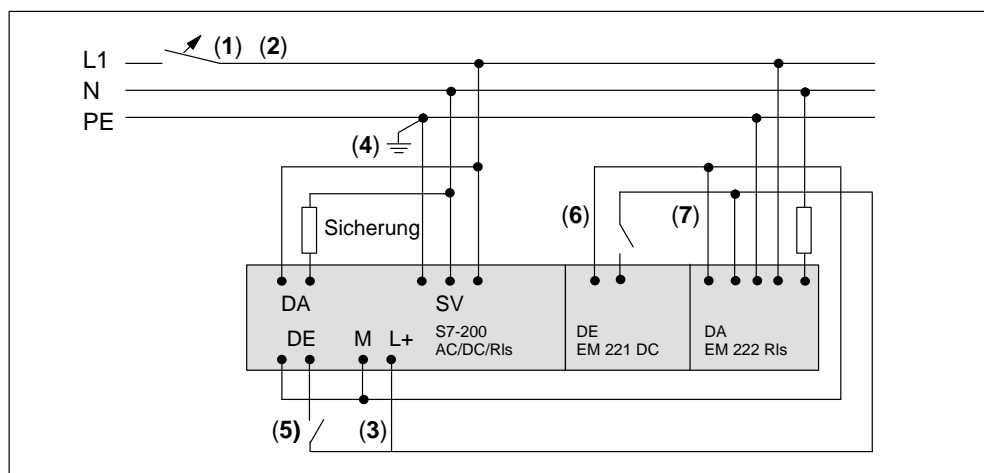


Bild 2-12 120/230 V AC: Einzeltrennschalter als Schutz für die CPU und die Laststromkreise

Installationsrichtlinien für DC-Aufbau

Im folgenden sind allgemeine Installationsrichtlinien für DC-Aufbau aufgeführt. Die Richtlinien können Sie in Bild 2-13 nachvollziehen.

- Installieren Sie einen Einzeltrennschalter (1), der die Stromzufuhr zur CPU sowie zu allen Eingangs- und Ausgangsstromkreisen (Laststromkreisen) unterbrechen kann.
- Installieren Sie Überstromschutzvorrichtungen für die Stromversorgung der CPU (2), für die Ausgänge (3) und Eingänge (4). Zusätzlich können Sie jeden Ausgang einzeln absichern. Ein externer Überstromschutz für die Eingänge ist nicht nötig, wenn Sie die Geberversorgung (24 V DC) der Micro-SPS einsetzen. Diese Geberversorgung ist intern strombegrenzt.
- Stellen Sie sicher, daß die DC-Versorgung ausreichend stoßspannungsfest ist, damit die Spannung auch während plötzlich auftretender Lastschwankungen konstant gehalten wird. Eventuell ist eine externe kapazitive Last (5) nötig.
- Versehen Sie ungeerdete DC-Versorgungen mit einem Widerstand und einem Kondensator (6), die von der Spannungsquelle bis zur Erdung parallel geschaltet sind. Der Widerstand sorgt für einen Kriechstromweg, der statische Aufladungen verhindert. Der Kondensator nimmt hochfrequente Störströme auf. Typische Werte sind 1 MΩ und 4700 pF. Sie können auch ein geerdetes DC-System aufbauen, indem Sie die DC-Versorgung erden (7).
- Verbinden Sie alle Erdungsklemmen der S7-200 mit der nächsten verfügbaren Erdung (8), um die höchstmögliche Störfestigkeit zu erreichen. Es wird empfohlen, alle Erdungsklemmen einzeln anzuschließen. Verwenden Sie hierzu Leitungen mit einem Querschnitt von 1,5 mm².
- Legen Sie 24-V-Gleichstrom nur aus Stromquellen an, die sichere elektrische Trennung von 120/230-V-Wechselstrom und ähnlichen Gefahrenquellen bieten.

Sichere elektrische Trennung wird beispielsweise in folgenden Normen definiert:

- PELV (protected extra low voltage) gemäß EN 60204-1
- Klasse 2 oder Stromkreis mit begrenzter Spannung/Strom gemäß UL 508

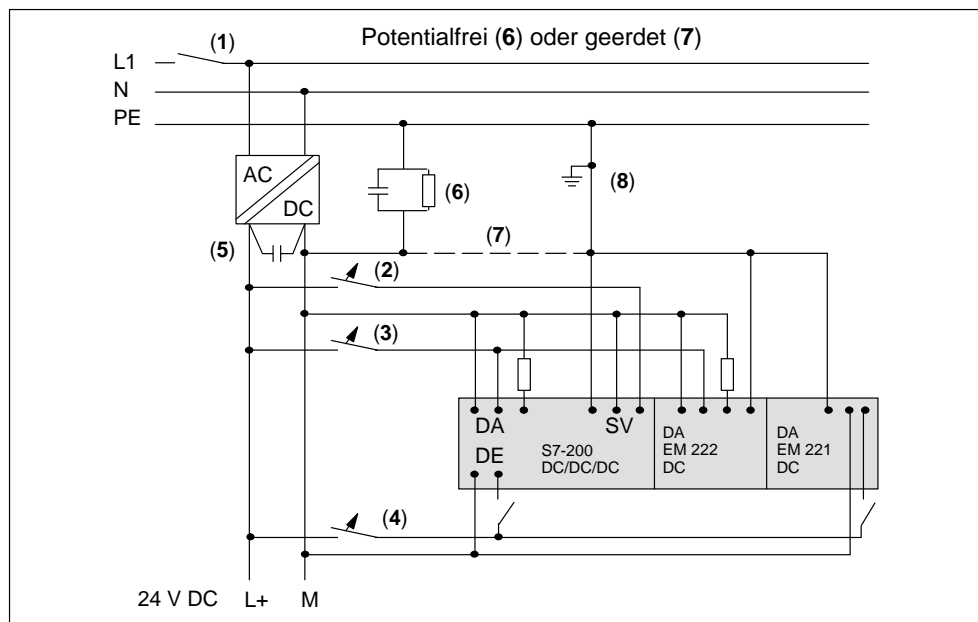


Bild 2-13 Isolierter DC-Aufbau

Installationsrichtlinien für "amerikanischen" Aufbau

Im folgenden sind allgemeine Installationsrichtlinien für "amerikanischen" Aufbau aufgeführt, wobei mehrere AC-Spannungen vorhanden sind. Die Richtlinien können Sie in Bild 2-14 nachvollziehen.

- Installieren Sie einen Einzeltrennschalter (1), der die Stromzufuhr zur CPU sowie zu allen Eingangs- und Ausgangstromkreisen (Laststromkreisen) unterbrechen kann.
- Installieren Sie Überstromschutzvorrichtungen für die Stromversorgung der CPU (2), für die Ausgänge (3) und Eingänge (4). Zusätzlich können Sie jeden Ausgang einzeln absichern.
- An die AC-Versorgung schließen Sie die Stromversorgung der CPU, die AC-Ausgangslastspannungen und die relaisgesteuerten Lastspannungen entweder einphasig an den Neutralleiter (5) oder zweiphasig (6) an.
- Verbinden Sie alle Erdungsklemmen der S7-200 mit der nächsten verfügbaren Erdung (7), um die höchstmögliche Störfestigkeit zu erreichen. Es wird empfohlen, alle Erdungsklemmen einzeln anzuschließen. Verwenden Sie hierzu Leitungen mit einem Querschnitt von 1,5 mm².



Vorsicht

In einem Aufbau mit 230-V-AC-Nennspannung überschreiten zweiphasige Spannungen die Nennspannung der Stromversorgung, der Eingänge und der Ausgänge der S7-200.

Eine zu hohe Spannung kann den Ausfall der S7-200 und der angeschlossenen Geräte verursachen.

Verwenden Sie keine zweiphasigen Anschlüsse, wenn die Nennspannung der S7-200 überschritten wird.

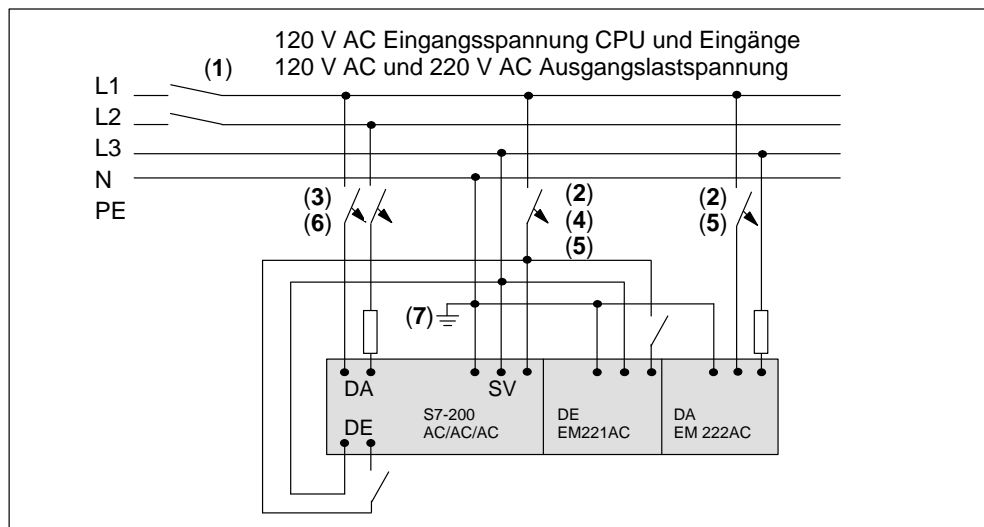


Bild 2-14 AC-Aufbau

2.4 Schutzbeschaltungen

Allgemeine Richtlinien

Versehen Sie induktive Lastspannungen mit Schutzbeschaltungen, die den Spannungsanstieg beim Ausschalten begrenzen. Beachten Sie beim Aufbau einer geeigneten Schutzbeschaltung die folgenden Richtlinien. Die Effektivität eines Aufbaus hängt von der jeweiligen Anwendung ab und muß immer für den Einzelfall geprüft werden. Vergewissern Sie sich deshalb, daß alle Komponenten für die jeweilige Anwendung bemessen sind.

Richtlinien für Schutzbeschaltungen für Transistorausgänge

In den DC-Transistorausgängen der S7-200 sind Zener-Dioden eingebaut, die sich für viele Aufbauarten eignen. Verwenden Sie externe Entstördioden entweder für große oder häufig geschaltete induktive Lastspannungen, um dadurch eine Überspannung in den internen Dioden zu verhindern. Die Bilder 2-15 und 2-16 zeigen einige typische Anwendungen für Transistorausgänge.

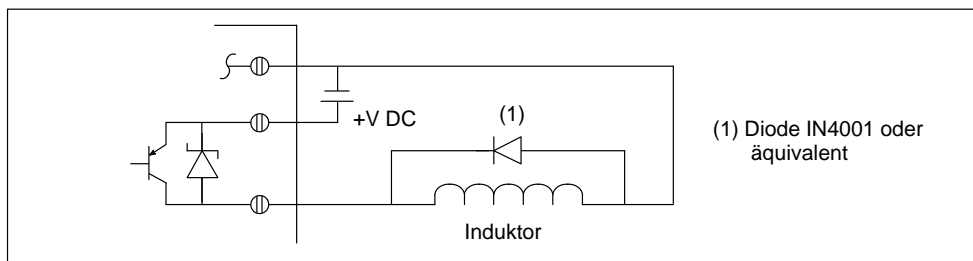


Bild 2-15 Überspannungsschutz: Diode

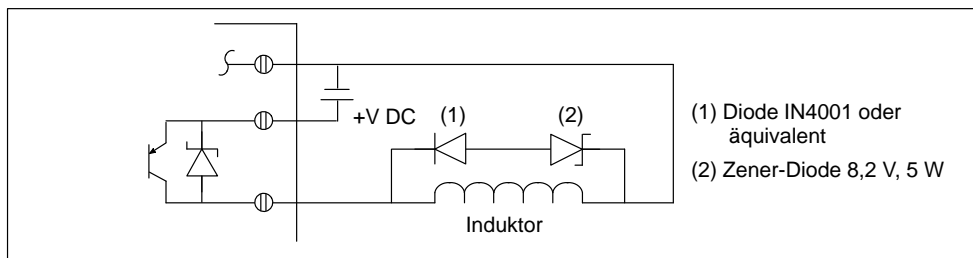


Bild 2-16 Überspannungsschutz: Zener-Diode

Schutzbeschaltungen für DC-Relaisausgänge

Bei Gleichstromverbrauchern mit niedriger Spannung (30 V) sollten Widerstände oder Kondensatoren parallel zu den Verbrauchern als Überspannungs-Schutzelemente eingesetzt werden (siehe Bild 2-17).

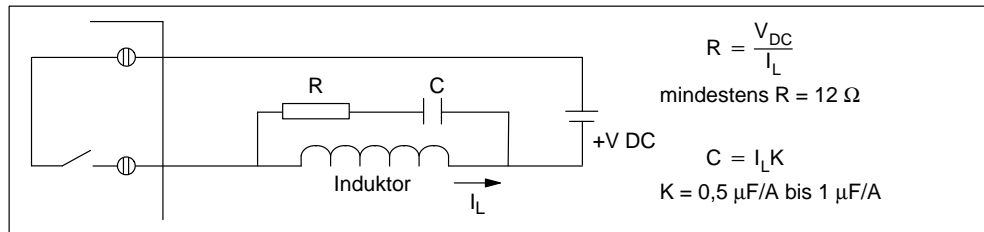


Bild 2-17 Widerstand/Kondensator, parallel zum Verbraucher geschaltet

Sie können bei Gleichstromverbrauchern auch Entstörungsdioden verwenden (siehe Bilder 2-15 und 2-16). Bei Zener-Dioden, die in Sperrichtung geschaltet sind, ist eine Schwellenspannung bis zu 36 V zulässig.

Ausgangs-Schutzbeschaltungen bei Wechselspannung

Wenn Sie mit einem Relais oder einem Wechselspannungsausgang Lastspannungen von 115 V/230 V AC schalten, müssen Sie die Widerstände oder Kondensatoren parallel zu den Relaiskontakten bzw. parallel zu den AC-Ausgängen anbringen (siehe Bild 2-18). Sie können auch einen Metalloxid-Varistor (MOV) einsetzen, um Spitzenspannungen zu begrenzen. Achten Sie darauf, daß die Arbeitsspannung des Varistors mindestens 20% höher ist als die Nennspannung.

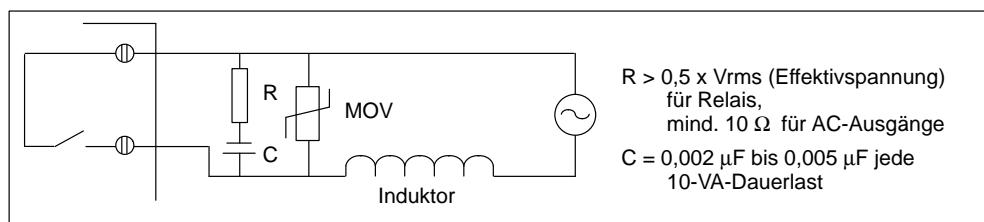


Bild 2-18 AC-Lastspannung parallel zum Relais bzw. zu den AC-Ausgängen

Der Kondensator verursacht dabei einen zusätzlichen Ableitstrom am offenen Schalter. Achten Sie darauf, daß der Ableitstrom $I = 2 \times 3,14 \times f \times C \times V_{rms}$ (Effektivspannung) für Ihre Anwendung akzeptabel ist.

Beispiel: Ein NEMA-Schütz der Größe 2 verzeichnet einen Einschaltstromstoß an der Spule von 183 VA und eine Dauerbelastung von 17 VA. Bei 230-V-Wechselspannung beträgt der Einschaltstromstoß $183 \text{ VA}/230 \text{ V} = 0,8 \text{ A}$, was innerhalb der Schaltleistung der Relaiskontakte von 2 A liegt.

Widerstand = $0,5 \times 230 = 115 \Omega$; wählen Sie 140Ω als Standardwert.

Kondensator = $(17 \text{ VA}/10) \times 0,005 = 0,0085 \mu F$; wählen Sie den Wert $0,01 \mu F$.

Ableitstrom = $2 \times 3,14 \times 50 \times 0,01 \times 10^{-6} \times 230 = 0,72 \text{ mA}$ Effektivwert.

2.5 Spannungsversorgung

Die Zentralgeräte der S7-200 besitzen eine interne Stromversorgung, die neben dem Zentralgerät die Erweiterungsmodule und andere 24-V-DC-Verbraucher speist. Mit Hilfe der folgenden Informationen können Sie berechnen, wieviel Leistung bzw. Strom das Zentralgerät für Ihre Konfiguration zur Verfügung stellen kann.

Leistungsbedarf

Jede S7-200 CPU liefert Gleichstrom von 5 V und 24 V:

- Jede CPU besitzt eine 24-V-DC-Geberversorgung, die die integrierten Eingänge und die Relaispulen der Erweiterungsmodule mit 24-V-Gleichstrom versorgt. Ist der Leistungsbedarf der CPU an 24-V-Gleichstrom höher als die interne Stromversorgung liefern kann, können Sie zusätzlich eine externe 24-V-DC-Versorgung anschließen, die die Erweiterungsmodule speist.
- Die CPU verfügt außerdem über 5-V-Gleichstrom für eventuell angeschlossene Erweiterungsmodule. Ist der Leistungsbedarf der Erweiterungsmodule an 5-V-Gleichstrom höher, als die interne Stromversorgung liefern kann, müssen Sie auf Erweiterungsmodule verzichten, so daß die Strombilanz der S7-200 wieder eingehalten wird.



Warnung

Wenn Sie parallel zu der DC-Geberversorgung der S7-200 eine externe 24-V-DC-Spannungsquelle anschließen, kann es sein, daß die beiden Spannungsquellen sich beim Aufbauen der geeigneten Ausgangsspannung gegenseitig beeinträchtigen.

Als Folge kann sich die Lebensdauer verkürzen bzw. eine oder beide Spannungsquellen können sofort ausfallen. Daraus resultiert ein unvorhersehbarer Betrieb des Automatisierungssystems, der zu tödlichen oder schweren Verletzungen und/oder Sachschaden führen kann.

Die DC-Geberversorgung der S7-200 und eine externe Spannungsquelle müssen die Spannung an unterschiedlichen Punkten liefern, wobei maximal eine Verbindung zwischen den beiden Spannungsquellen bestehen darf.

Die Datenblätter in Anhang A bieten Informationen zu der verfügbaren Leistung der einzelnen CPUs und zum Leistungsbedarf der Erweiterungsmodule.

Beispiel für die Berechnung einer Strombilanz

Tabelle 2-1 zeigt ein Beispiel für die Berechnung des Leistungsbedarfs einer Micro-SPS S7-200 mit folgenden Geräten:

- CPU 214 DC/DC/DC
- Drei Erweiterungsmodule EM 221 Digitaleingabe 8 x 24 V DC
- Zwei Erweiterungsmodule EM 222 Digitalausgabe 8 x Relais

In diesem Beispiel liefert die CPU genügend 5-V-Gleichstrom für die Erweiterungsmodule. Die CPU benötigt jedoch eine externe 24-V-DC-Versorgung. (Die Ein- und Ausgänge haben einen Leistungsbedarf von 448 mA an 24-V-Gleichstrom, die CPU liefert aber nur 280 mA.) In Anhang B finden Sie eine leere Tabelle für Ihre eigenen Berechnungen.

Tabelle 2-1 Beispiel für die Berechnung der Strombilanz einer S7-200

Leistung der CPU	5 VDC	24 V DC
<i>CPU 214 DC/DC/DC</i>	<i>660 mA</i>	<i>280mA</i>

minus

Strombedarf des Systems	5 VDC	24 V DC
<i>CPU 214 DC/DC/DC</i>	<i>Zentralgerät</i>	<i>14 Eingänge x 7 mA = 98 mA</i>
<i>Drei Erweiterungsmodule EM 221</i>	<i>3 x 60 mA = 180 mA</i>	<i>3 x 60 mA = 180 mA</i>
<i>Zwei Erweiterungsmodule EM 222</i>	<i>2 x 80 mA = 160 mA</i>	<i>2 x 85 mA = 170 mA</i>
Gesamtbedarf	<i>340 mA</i>	<i>448 mA</i>

gleich

Strombilanz	5 VDC	24 V DC
Gesamt-Strombilanz	<i>320 mA</i>	<i>[168 mA]</i>

Installieren und Bedienen der Software STEP 7-Micro/WIN

3

Dieses Handbuch beschreibt die Version 2.1 der Software STEP 7-Micro/WIN. Vorherige Versionen der Software weisen zum Teil andere Funktionsweisen auf.

STEP 7-Micro/WIN ist eine Software, die unter Windows läuft, und zwar sowohl unter Windows 3.1 (16-Bit-Umgebung, deshalb STEP 7-Micro/WIN 16) als auch unter Windows 95 und Windows NT (32-Bit-Umgebungen, deshalb STEP 7-Micro/WIN 32). Zum Arbeiten mit STEP 7-Micro/WIN empfehlen wir folgende Hardware:

- Empfohlen: Personal Computer (PC) mit Prozessor 80586 oder höher und 16 MB RAM-Speicher oder Siemens Programmiergerät (z.B. PG 740); Mindestausstattung Computer: Prozessor 80486 mit 8 MB Ram-Speicher
- Eine der folgenden Komponenten:
 - PC/PPI-Kabel zum Anschließen an die Kommunikationsschnittstelle (am PC: COM1 oder COM2)
 - Kommunikationsprozessor (CP) und MPI-Kabel
 - MPI-Baugruppe (ein Kommunikationskabel ist im Lieferumfang der MPI-Baugruppe enthalten)
- VGA-Monitor bzw. beliebiger von Microsoft Windows unterstützter Monitor
- Mindestens 50 MB freier Speicherplatz auf der Festplatte
- Microsoft Windows 3.1, Windows for Workgroups 3.11, Windows 95 oder Windows NT 4.0 oder höher
- Optional, aber empfohlen: eine von Microsoft Windows unterstützte Maus

STEP 7-Micro/WIN bietet Ihnen umfassende Online-Hilfe. Wählen Sie einen der Befehle im Menü **Hilfe** oder drücken Sie F1, um Hilfe zu aktuellen Fragestellungen zu erhalten.

Kapitelübersicht

Abschnitt	Beschreibung	Seite
3.1	Installieren der Software STEP 7-Micro/WIN	3-2
3.2	Einrichten der Kommunikationshardware mit STEP 7-Micro/WIN	3-4
3.3	Einrichten der Kommunikation mit einer S7-200 CPU	3-7
3.4	Einrichten der Voreinstellungen in STEP 7-Micro/WIN	3-25
3.5	Erstellen und Speichern eines Projekts	3-26
3.6	Erstellen eines Programms	3-27
3.7	Erstellen eines Datenbausteins	3-32
3.8	Arbeiten mit der Statustabelle	3-34
3.9	Arbeiten mit symbolischer Adressierung	3-36

3.1 Installieren der Software STEP 7-Micro/WIN

Vor der Installation

Führen Sie die folgenden Schritte aus, bevor Sie STEP 7-Micro/WIN installieren:

- Haben Sie bereits eine vorherige Version von STEP 7-Micro/WIN auf Ihrem Computer installiert, sichern Sie alle Projekte aus STEP 7-Micro/WIN auf Diskette.
- Schließen Sie alle Anwendungen, einschließlich der Symbolleiste von Microsoft Office.

Es kann sein, daß Sie aufgefordert werden, Ihren Computer neu zu starten.

Installation unter Windows 3.1

Läuft auf Ihrem Rechner Windows 3.1 (Windows for Workgroups 3.11), gehen Sie zum Installieren der Software STEP 7-Micro/WIN 16 folgendermaßen vor:

1. Schieben Sie die erste Diskette in das Diskettenlaufwerk Ihres Computers (im allgemeinen Laufwerk A: oder B:).
2. Wählen Sie im Programm-Manager den Menübefehl **Datei ► Ausführen...**
3. Im Dialogfeld "Ausführen" geben Sie **a:\setup** ein und bestätigen mit "OK" bzw. drücken die Eingabetaste. Daraufhin wird die Installation gestartet.
4. Folgen Sie den Anweisungen auf dem Bildschirm bis zum Ende des Installationsvorgangs.

Installation unter Windows 95 oder Windows NT 4.0

Läuft auf Ihrem Rechner Windows 95 oder Windows NT 4.0, gehen Sie zum Installieren der Software STEP 7-Micro/WIN 32 folgendermaßen vor:

1. Schieben Sie die erste Diskette in das Diskettenlaufwerk Ihres Computers (im allgemeinen Laufwerk A: oder B:).
2. Klicken Sie auf die Schaltfläche Start, um das Startmenü in Windows 95 zu öffnen.
3. Klicken Sie auf den Menübefehl **Ausführen....**
4. Im Dialogfeld "Ausführen" geben Sie **a:\setup** ein und bestätigen mit "OK" bzw. drücken die Eingabetaste. Daraufhin wird die Installation gestartet.
5. Folgen Sie den Anweisungen auf dem Bildschirm bis zum Ende des Installationsvorgangs.
6. Am Ende des Installationsvorgangs wird das Dialogfeld zum Installieren und Deinstallieren von Baugruppen automatisch angezeigt (siehe Bild 3-1). Sie können die Hardware, die Ihr Rechner zur Kommunikation benötigt, sofort installieren (siehe Abschnitt 3.2), Sie können sie aber auch später noch installieren (siehe Abschnitt 3.3).

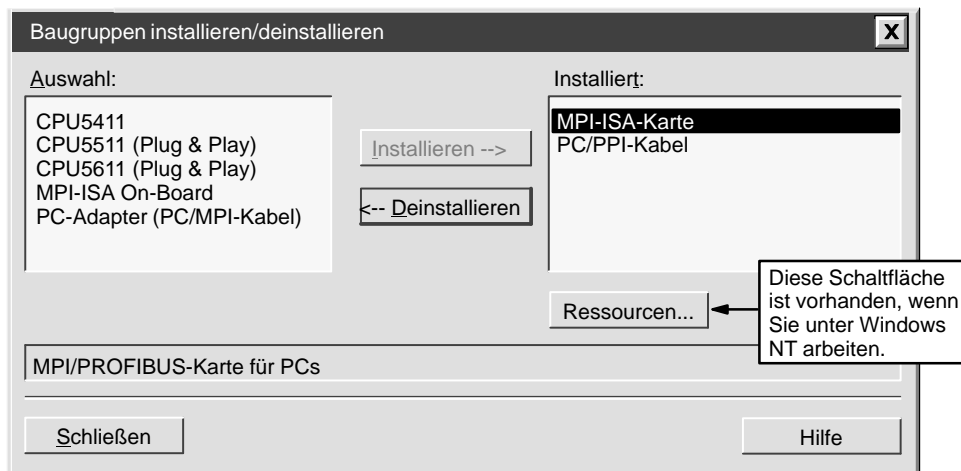


Bild 3-1 Dialogfeld zum Installieren und Deinstallieren von Baugruppen

Fehler während der Installation

Während der Installation können folgende Fehler auftreten:

- Nicht genügend Speicher: Sie benötigen mindestens 50 MB freien Speicherplatz auf Ihrer Festplatte.
- Fehlerhafte Diskette: Vergewissern Sie sich, daß die Diskette fehlerhaft ist. Wenden Sie sich an Ihre Siemens-Vertretung.
- Bedienerfehler: Installieren Sie die Software erneut und beachten Sie die Installationsanweisungen.
- Sie haben nicht alle anderen geöffneten Anwendungen geschlossen (einschließlich der Symbolleiste von Microsoft Office).

Lesen Sie die Datei READMEx.TXT, die sich auf den Installationsdisketten befindet. Diese Datei enthält aktuelle Informationen zu STEP 7-Micro/WIN. (Anstelle des x bedeutet A=Deutsch, B=Englisch, C=Französisch, D=Spanisch, E=Italienisch).

3.2 Einrichten der Kommunikationshardware mit STEP 7-Micro/WIN

Allgemeine Informationen zum Installieren und Deinstallieren der Kommunikationshardware

Wenn Sie unter Windows 95 oder Windows NT 4.0 arbeiten, dann wird das Dialogfeld zum Installieren und Deinstallieren von Baugruppen automatisch nach der Installation der Software aufgerufen (siehe Bild 3-1). Wenn Sie unter Windows 3.1 arbeiten, gehen Sie folgendermaßen vor:

1. Wählen Sie den Menübefehl **Einrichten ► Kommunikation...**. Daraufhin wird das Dialogfeld "Kommunikation" angezeigt.
2. Wählen Sie die Schaltfläche "PG/PC-Schnittstelle...". Das Dialogfeld "PG/PC-Schnittstelle einstellen" wird aufgerufen.
3. Wählen Sie die Schaltfläche "Installieren...". Daraufhin erscheint das Dialogfeld "Baugruppen installieren/deinstallieren" (siehe Bild 3-1).

Richten Sie sich bei der Installation von Kommunikationshardware nach folgenden Kriterien:

- Ihrem Betriebssystem (Windows 3.1, Windows 95 oder Windows NT 4.0)
- Der Hardware, die Sie einsetzen, z.B.:
 - PC mit PC/PPI-Kabel
 - PC bzw. SIMATIC Programmiergerät mit MPI-Schnittstelle Kommunikationsprozessor (CP)
 - CPU 212, CPU 214, CPU 215, CPU 216
 - Modem
- Die verwendete Baudrate

Tabelle 3-1 zeigt die möglichen Hardware-Konfigurationen und Baudraten, die von STEP 7-Micro/WIN unterstützt werden, und zwar je nach der von Ihnen eingesetzten CPU. Ausführliche Informationen zum Einrichten der Kommunikation entnehmen Sie dem Abschnitt 3.3.

Tabelle 3-1 Von STEP 7-Micro/WIN unterstützte Hardware-Konfigurationen

CPU-Typ	Version von STEP 7-Micro/WIN	Unterstützte Hardware	Unterstützte Baudraten	Betriebssystem	Parametrierung
CPU 212, CPU 214, CPU 216 CPU 215 Schnittstelle 0	STEP 7-Micro/WIN 16	PC/PPI-Kabel, MPI-ISA-Karte	9600 Baud oder 19.200 Baud	Windows 3.1	PPI, PPI-Multi-Master
				Windows 95 oder Windows NT	PPI
	STEP 7-Micro/WIN 32	PC/PPI-Kabel, MPI-ISA-Karte, MPI-ISA-Karte (On-Board), CP 5411, CP 5511, CP 5611	9600 Baud oder 19.200 Baud	Windows 95 oder Windows NT	PPI, PPI-Multi-Master
CPU 215 Schnittstelle 1 (DP-Schnittstelle)	STEP 7-Micro/WIN 16	Nicht unterstützt	Nicht unterstützt	Windows 3.1 Windows 95 oder Windows NT	Nicht unterstützt
	STEP 7-Micro/WIN 32	MPI-ISA-Karte, MPI-ISA-Karte (On-Board), CP 5411, CP 5511, CP 5611	9600 Baud bis 12 MBaud	Windows 95 oder Windows NT	MPI

Hinweis

STEP 7-Micro/WIN 16 unterstützt nicht die Parametrierung für Multi-Master-Betrieb unter Windows 95 und Windows NT 4.0.

Folgende Hardware-Konfigurationen sind möglich:

- CPU 212, CPU 214, CPU 216, CPU 215 (Schnittstelle 0)
 - PC/PPI-Kabel (PPI), 9600 Baud oder 19.200 Baud
 - MPI-Baugruppe (PPI), 9600 Baud oder 19.200 Baud
- CPU 215 (Schnittstelle 1, d.h. DP-Schnittstelle)
 - MPI-Baugruppe (MPI), 9600 Baud bis 12 Mbaud

Hinweis

STEP 7-Micro/WIN 16 unterstützt keine Kommunikation über die Schnittstelle 1 der CPU 215.

Die Auswahl für die MPI-Baugruppe ist unterschiedlich für STEP 7-Micro/WIN 16 und STEP 7-Micro/WIN 32.

Auf der linken Seite des Dialogfelds "Baugruppen installieren/deinstallieren" sehen Sie eine Liste der Hardware, die Sie noch nicht installiert haben (siehe Bild 3-1). Auf der rechten Seite des Dialogfelds sehen Sie eine Liste der Hardware, die Sie installiert haben. Wenn Sie mit dem Betriebssystem Windows NT 4.0 arbeiten, enthält das Dialogfeld eine Schaltfläche "Ressourcen" unterhalb der Liste der installierten Hardware.

Zum Installieren von Hardware gehen Sie folgendermaßen vor:

1. Markieren Sie in dem Textfeld "Auswahl" die Hardware, die Ihnen zur Verfügung steht und die Sie installieren möchten. Eine Beschreibung Ihrer Auswahl wird im unteren Bereich des Dialogfelds angezeigt.
2. Wählen Sie die Schaltfläche "Installieren -->".

Zum Deinstallieren von Hardware gehen Sie folgendermaßen vor:

1. Wählen Sie in dem Textfeld rechts bereits installierte Hardware aus.
2. Wählen Sie die Schaltfläche "<-- Deinstallieren".

Haben Sie die gewünschte Hardware installiert bzw. deinstalliert, klicken Sie auf die Schaltfläche "Schließen". Daraufhin wird das Dialogfeld geschlossen und wieder das Dialogfeld "PG/PC-Schnittstelle einstellen" angezeigt. Die installierte Hardware wird jetzt in dem Feld "Benutzte Baugruppenparametrierung" aufgeführt (siehe Bild 3-7).

Ausführliche Informationen zum Einrichten der Kommunikation für Ihre Konfiguration entnehmen Sie dem Abschnitt 3.3.

Besondere Informationen zur Installation von Hardware unter Windows NT

Unter Windows NT gehen Sie zum Installieren von Hardware-Baugruppen geringfügig anders vor als beim Installieren von Hardware-Baugruppen unter Windows 95. Auch wenn sich die Baugruppen für beide Betriebssysteme nicht unterscheiden, erfordert die Installation unter Windows NT ausführlichere Kenntnis der Hardware, die Sie installieren möchten. Windows 95 versucht automatisch, Systemressourcen einzurichten. Unter Windows NT geschieht dies nicht. Windows NT bietet Ihnen nur Standardwerte. Diese Werte entsprechen eventuell der Hardware-Konfiguration. Sie können die Parameter allerdings ganz leicht ändern, so daß sie den erforderlichen Einstellungen für das System entsprechen.

Nachdem Sie die Hardware installiert haben, markieren Sie sie in dem Textfeld "Installiert" und wählen anschließend die Schaltfläche "Ressourcen". Daraufhin wird das Dialogfeld "Ressourcen" angezeigt (siehe Bild 3-2). Im Dialogfeld "Ressourcen" können Sie die Systemeinstellungen für die tatsächliche Hardware, die Sie installiert haben, ändern. Ist die Schaltfläche nicht verfügbar (bzw. grau), brauchen Sie keine weiteren Einstellungen vorzunehmen.

Es kann erforderlich sein, daß Sie mit Hilfe des Handbuchs zu Ihrer Hardware die Einstellungen der einzelnen Parameter, die in dem Dialogfeld aufgeführt werden, ermitteln. Es kann sein, daß Sie zunächst verschiedene Interrupts ausprobieren müssen, bevor Sie die Kommunikation fehlerfrei aufbauen können.

Ausführliche Informationen zum Einrichten der Kommunikation für Ihre Konfiguration entnehmen Sie dem Abschnitt 3.3.

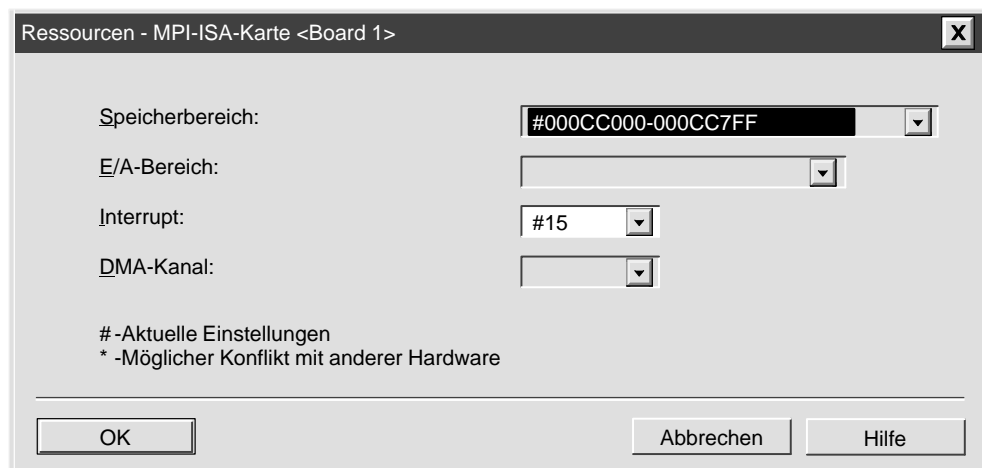


Bild 3-2 Dialogfeld "Ressourcen" unter Windows NT

3.3 Einrichten der Kommunikation mit einer S7-200 CPU

Sie können die S7-200 CPUs in einer Vielzahl von Konfigurationen anordnen, die alle die Kommunikation im Netz unterstützen. Sie können die Software STEP 7-Micro/WIN auf einem Personal Computer (PC) installieren, auf dem eines der Betriebssysteme Windows 3.1x, Windows 95 oder Windows NT läuft. Sie können STEP 7-Micro/WIN aber auch auf einem SIMATIC Programmiergerät (z.B. einem PG 740) installieren. Den PC oder das Programmiergerät setzen Sie in den folgenden Kommunikationskonfigurationen als Master ein:

- Ein einziger Master ist mit einem oder mehreren Slaves verbunden (siehe Bild 3-3).
- Ein Master ist mit einem oder mehreren Slaves und einem oder mehreren Masters verbunden (siehe Bilder 3-4 und 3-5).
- Eine CPU 215 wird als Modul in der dezentralen Peripherie eingesetzt und gehört zu einem Automatisierungssystem S7-300 oder S7-400 oder zu einem anderen PROFIBUS-Master (siehe Bild 3-13).
- Ein einziger Master ist mit einem oder mehreren Slaves verbunden. Dieser Master ist über 11-Bit-Modems mit einer S7-200 CPU, die als Slave eingesetzt ist, oder mit einem Netz aus mehreren S7-200 CPUs, die als Slaves eingesetzt sind, verbunden (siehe Bild 3-14).

Anschließen Ihres Computers an die S7-200 CPU über das PC/PPI-Kabel

Bild 3-3 zeigt Ihnen eine typische Konfiguration zum Verbinden Ihres Personal Computers und Ihrer CPU mit einem PC/PPI-Kabel. Zum Einrichten der Kommunikation zwischen den Komponenten gehen Sie folgendermaßen vor:

1. Stellen Sie mit den DIP-Schaltern am PC/PPI-Kabel eine Baudrate ein.
2. Schließen Sie die Schnittstelle RS-232 des PC/PPI-Kabels, die mit der Kennzeichnung PC versehen ist, an die Kommunikationsschnittstelle Ihres Computers, entweder COM1 oder COM2, an. Drehen Sie die Verbindungsschrauben fest.
3. Schließen Sie die zweite Schnittstelle des PC/PPI-Kabels (RS-485) an die Kommunikationsschnittstelle Ihrer CPU an. Drehen Sie die Verbindungsschrauben fest.

Die technischen Daten des PC/PPI-Kabels entnehmen Sie dem Abschnitt A.40. Die Bestellnummer des Kabels entnehmen Sie Anhang G.

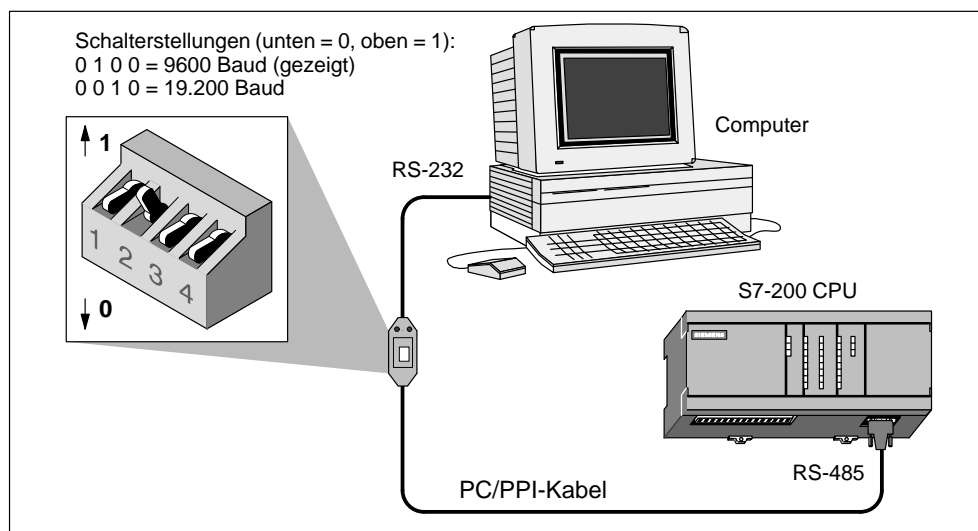


Bild 3-3 Kommunizieren mit einer CPU im PPI-Modus

Bild 3-4 zeigt eine Konfiguration, in der ein Personal Computer mit mehreren S7-200 CPUs verbunden ist. STEP 7-Micro/WIN ist so ausgelegt, daß immer nur mit einer S7-200 CPU kommuniziert werden kann. Sie können jedoch auf jede CPU im Netz zugreifen. Bei den CPUs in Bild 3-4 kann es sich um Slaves als auch um Master handeln. Das TD 200 ist ein Master. Ausführliche Informationen zur Kommunikation im Netz entnehmen Sie dem Kapitel 9.

Hinweis

Nur STEP 7-Micro/WIN 16 unter Windows 3.1 und STEP 7-Micro/WIN 32 unterstützen mehrere Master bei Verwendung des PC/PPI-Kabels. STEP 7-Micro/DOS unterstützt den Multi-Master-Modus nicht.

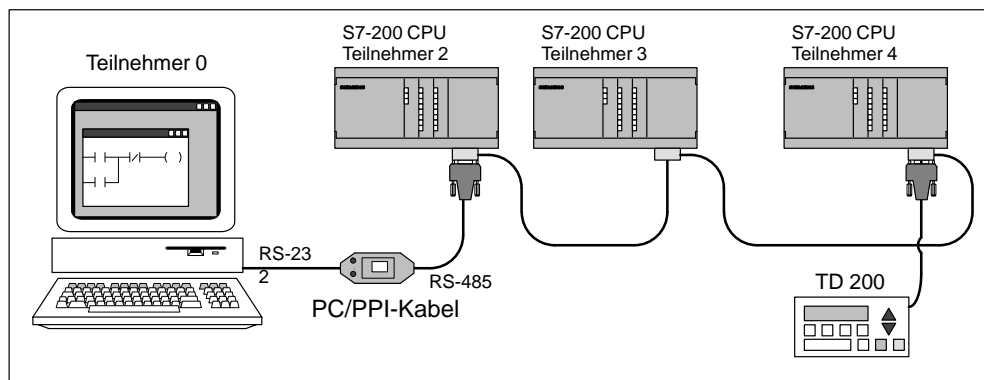


Bild 3-4 Kommunizieren mit mehreren S7-200 CPUs über PC/PPI-Kabel

Anschließen Ihres Computers an die S7-200 CPU über MPI-Baugruppe oder CP

Sie können STEP 7-Micro/WIN zusammen mit einer MPI-Baugruppe oder einem Kommunikationsprozessor (CP) einsetzen. Beide Baugruppen verfügen über eine RS-485-Schnittstelle zum Anschließen an das Netz über MPI-Kabel. STEP 7-Micro/WIN 32 (die 32-Bit-Version) unterstützt die MPI-Baugruppenparametrierung für ein MPI-Netz. STEP 7-Micro/WIN 16 (16-Bit-Version) unterstützt diese Parametrierung nicht. Nachdem Sie die MPI-Kommunikation eingerichtet haben, können Sie STEP 7-Micro/WIN mit einem Netz verbinden, in dem andere Master vorhanden sind. Jeder Master muß eine eindeutige Adresse haben. Bild 3-5 zeigt ein Beispiel für ein MPI-Netz mit Master- und Slave-Geräten. Ausführliche Informationen zur Kommunikation im Netz entnehmen Sie dem Kapitel 9. Informationen zur MPI-Baugruppe und den verschiedenen CPs entnehmen Sie dem Abschnitt 9.4. Anhang G führt sämtliche Bestellnummern auf.

Hinweis

Wenn Sie die PPI-Parametrierung verwenden, unterstützt STEP 7-Micro/WIN nicht den Fall, daß zwei Anwendungen gleichzeitig mit derselben MPI-Baugruppe oder demselben CP arbeiten. Schließen Sie die andere Anwendung, bevor Sie STEP 7-Micro/WIN über die MPI-Baugruppe oder den CP mit dem Netz verbinden.

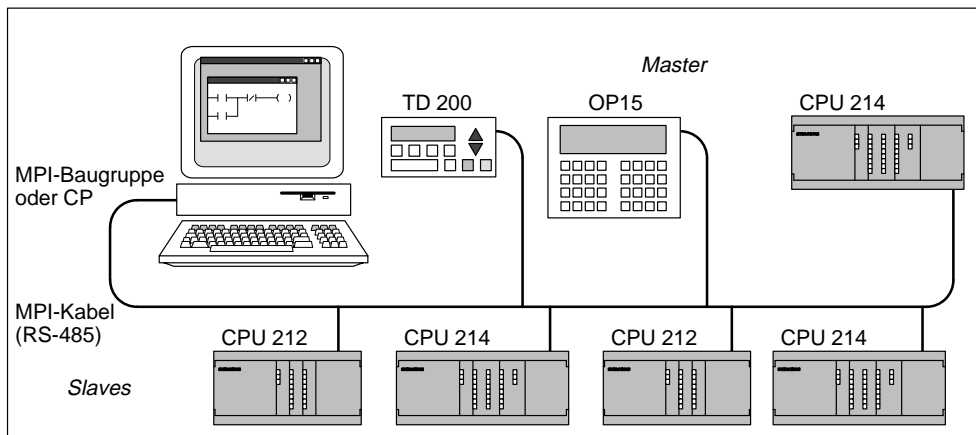


Bild 3-5 Beispiel für den Einsatz einer MPI-Baugruppe bzw. eines Kommunikationsprozessors mit Master- und Slave-Geräten

Wo beginne ich, die Kommunikation einzurichten?

Je nach dem Betriebssystem, mit dem Sie arbeiten, können Sie die Kommunikation folgendermaßen einrichten:

- Unter Windows 3.1
Nur in STEP 7-Micro/WIN 16
- Unter Windows 95 und Windows NT 4.0
 - Im letzten Schritt des Installationsvorgangs (siehe Abschnitt 3.1)
 - Über das Symbol "PG/PC-Schnittstelle einstellen" in der Systemsteuerung unter Windows
 - In STEP 7-Micro/WIN 32

Einrichten der Kommunikation in STEP 7-Micro/WIN

STEP 7-Micro/WIN bietet Ihnen das Dialogfeld "Kommunikation", in dem Sie Ihre Kommunikation einrichten können (siehe Bild 3-6). Zum Aufrufen dieses Dialogfelds können Sie folgendermaßen vorgehen:

- Wählen Sie den Menübefehl **Einrichten ► Kommunikation...**
- Erstellen Sie ein neues Projekt und klicken Sie auf die Schaltfläche "Kommunikation..." im Dialogfeld "CPU-Typ".
- Bei geöffnetem Projekt wählen Sie den Menübefehl **CPU ► CPU-Typ...** und klicken auf die Schaltfläche "Kommunikation..." im Dialogfeld "CPU-Typ".

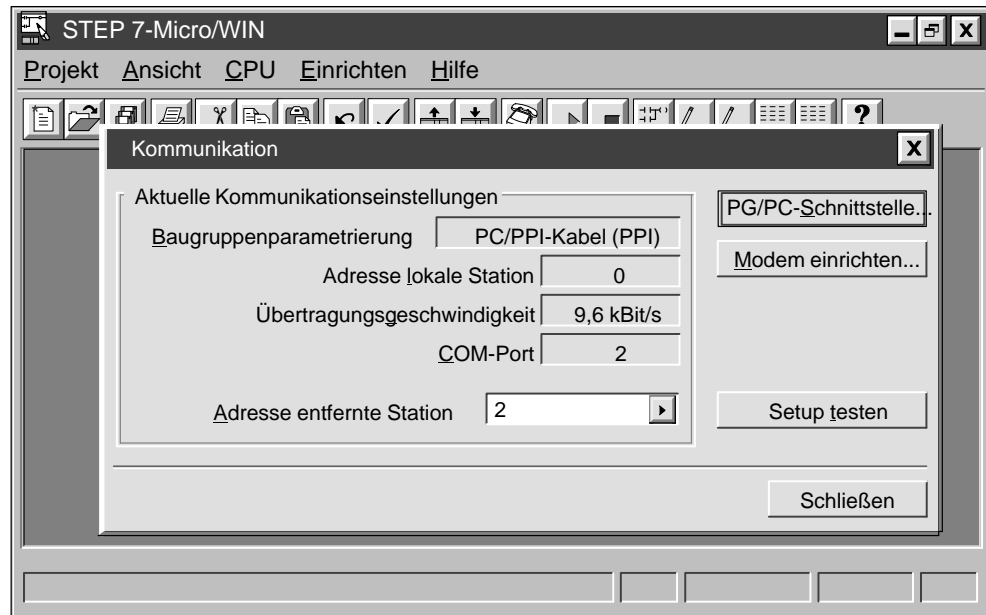


Bild 3-6 Einrichten der Kommunikation zwischen Programmiergerät bzw. PC und CPU

Haben Sie das Dialogfeld "Kommunikation" aufgerufen, wählen Sie die Schaltfläche "PG/PC-Schnittstelle...". Das Dialogfeld "PG/PC-Schnittstelle einstellen" wird aufgerufen (siehe Bild 3-7).

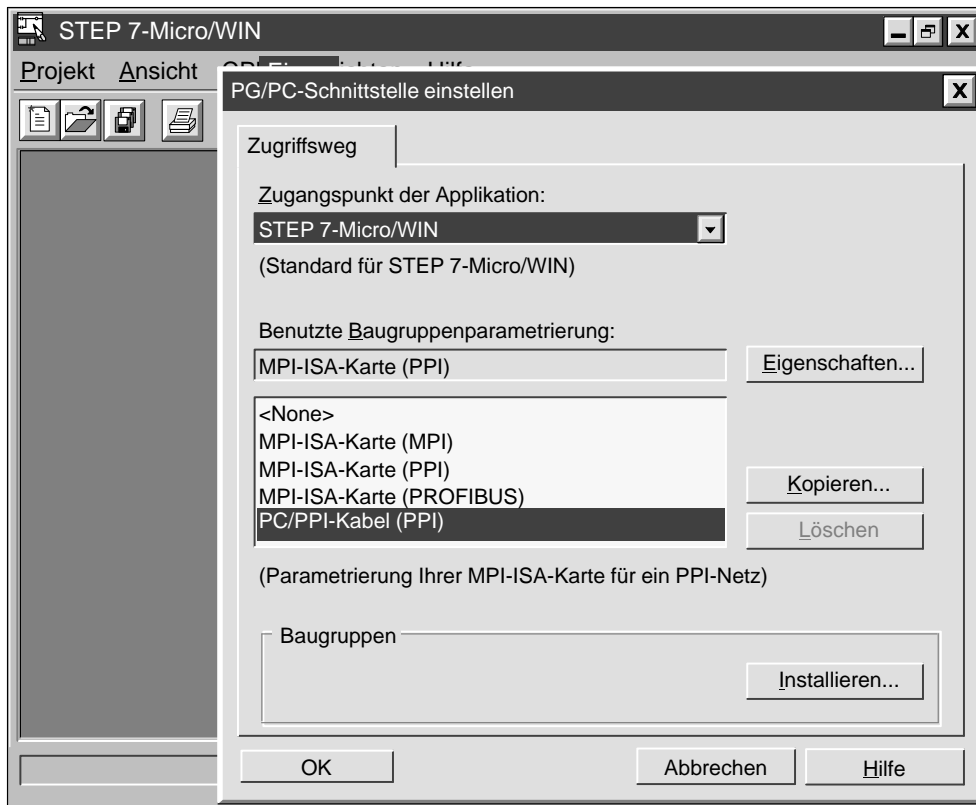


Bild 3-7 Einstellungen im Dialogfeld "PG/PC-Schnittstelle einstellen"

Einrichten der Kommunikation in der Systemsteuerung von Windows

Arbeiten Sie mit einem der Betriebssysteme Windows 95 oder Windows NT 4.0, dann können Sie die Kommunikation auch in der Systemsteuerung einrichten. Wählen Sie das Symbol "PG/PC-Schnittstelle einstellen" in der Systemsteuerung von Windows (siehe Bild 3-8).

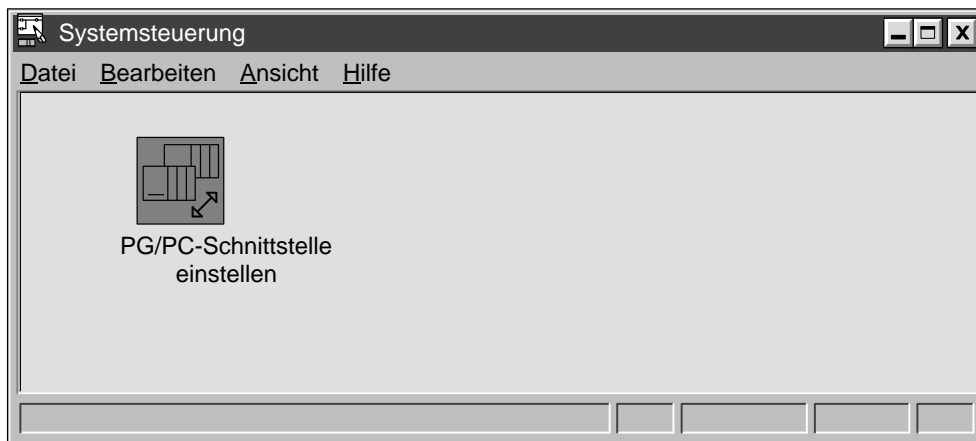


Bild 3-8 Symbol "PG/PC-Schnittstelle einstellen" in der Systemsteuerung

Einrichten der Kommunikation während der Installation

Wenn Sie unter Windows 95 oder Windows NT 4.0 arbeiten, dann wird das Dialogfeld zum Installieren und Deinstallieren von Baugruppen automatisch nach der Installation der Software STEP 7-Micro/WIN aufgerufen. Sie können die Kommunikation sofort oder später einrichten.

Auswählen und Einrichten der richtigen Baugruppenparametrierung

Befinden Sie sich im Dialogfeld "PG/PC-Schnittstelle einstellen" (siehe Bild 3-7), stellen Sie im Register "Zugriffsweg" im Textfeld "Zugangspunkt der Applikation" die Software "STEP 7-Micro/WIN" ein. Dieses Dialogfeld wird von verschiedenen Anwendungen, z.B. STEP 7 und WinCC, gemeinsam genutzt. Deshalb müssen Sie dem Programm mitteilen, für welche Anwendung Sie die Parameter einstellen möchten.

Haben Sie "STEP 7-Micro/WIN" angegeben und Ihre Hardware installiert, müssen Sie die tatsächlichen Eigenschaften zur Kommunikation mit Ihrer Hardware einstellen. Als erstes geben Sie das Protokoll an, das Sie für Ihr Netz verwenden möchten. Mit Hilfe von Tabelle 3-1 oder dem Kapitel 9 können Sie ermitteln, welche Protokolle von Ihrer CPU unterstützt werden und welches Sie am besten für Ihre Konfiguration einsetzen. In den meisten Fällen kann das PPI-Protokoll für alle CPUs verwendet werden, Ausnahme ist die Hochgeschwindigkeitsschnittstelle (DP-Schnittstelle) der CPU 215. Für diese Schnittstelle müssen Sie das MPI-Protokoll einsetzen.

Nachdem Sie sich entschieden haben, welches Protokoll Sie verwenden möchten, können Sie im Dialogfeld "PG/PC-Schnittstelle einstellen" im Feld "Benutzte Baugruppenparametrierung" die richtigen Einstellungen auswählen. In diesem Textfeld wird die von Ihnen installierte Hardware und in Klammern das entsprechende Protokoll angegeben. Beispiel: Sie möchten in einer einfachen Konfiguration über das PC/PPI-Kabel mit einer CPU 214 kommunizieren. In diesem Fall wählen Sie "PC/PPI-Kabel (PPI)." In einem anderen Fall möchten Sie über die Hochgeschwindigkeitsschnittstelle (DP-Schnittstelle) einer CPU 215 mit Hilfe einer einfachen MPI-ISA-Karte, die Sie in Ihrem Computer installiert haben, kommunizieren. In diesem Fall wählen Sie "MPI-ISA-Karte (MPI)."

Nachdem Sie die passende Baugruppenparametrierung angegeben haben, müssen Sie die einzelnen Parameter für die aktuelle Konfiguration einrichten. Wählen Sie hierzu im Dialogfeld "PG/PC-Schnittstelle einstellen" die Schaltfläche "Eigenschaften...". Daraufhin wird ein weiteres Dialogfeld aufgerufen, das sich nach den von Ihnen gewählten Parametern richtet. In den folgenden Abschnitten werden die verschiedenen Dialogfelder ausführlich beschrieben.

Möchten Sie eine Baugruppenparametrierung löschen, gehen Sie folgendermaßen vor:

1. Befinden Sie sich im Dialogfeld "PG/PC-Schnittstelle einstellen" (siehe Bild 3-7), stellen Sie im Register "Zugriffsweg" im Textfeld "Zugangspunkt der Applikation" die Software "STEP 7-Micro/WIN" ein.
2. Stellen Sie sicher, daß Ihre Hardware bereits installiert ist (siehe Abschnitt 3.2).
3. Geben Sie das Protokoll an, mit dem Sie arbeiten möchten.
4. Wählen Sie im Dialogfeld "PG/PC-Schnittstelle einstellen" die entsprechenden Einstellungen im Textfeld "Benutzte Baugruppenparametrierung" aus.
5. Wählen Sie im Dialogfeld "PG/PC-Schnittstelle einstellen" die Schaltfläche "Eigenschaften...".

Ab hier nehmen Sie alle weiteren Einstellungen entsprechend der von Ihnen gewählten Baugruppenparametrierung vor.

Einrichten der Parameter für das PC/PPI-Kabel (PPI)

In diesem Abschnitt wird erläutert, wie Sie zum Einrichten der PPI-Parameter für die folgenden Betriebssysteme und die folgende Hardware vorgehen:

- Windows 3.1: PC/PPI-Kabel
- Windows 95 oder Windows NT 4.0: PC/PPI-Kabel

Verwenden Sie ein PC/PPI-Kabel, und Sie klicken im Dialogfeld "PG/PC-Schnittstelle einstellen" auf die Schaltfläche "Eigenschaften", dann werden die Eigenschaften für das PC/PPI-Kabel (PPI) angezeigt (siehe Bild 3-9).

Gehen Sie folgendermaßen vor:

1. Geben Sie im Register "PPI-Netz" eine lokale Teilnehmeradresse an. Dies ist die Adresse für STEP 7-Micro/WIN im Netz des Automatisierungssystems.
2. Geben Sie einen Wert für Timeout an. Dieser Wert gibt an, wie lange die Kommunikationstreiber versuchen sollen, eine Kommunikation aufzubauen. Der standardmäßig eingestellte Wert dürfte ausreichend sein.
3. Geben Sie an, ob Sie möchten, daß STEP 7-Micro/WIN in einem Netz mit mehreren Mastern kommuniziert. Ausführliche Informationen hierzu entnehmen Sie Kapitel 9. Sie können das Kontrollkästchen "Multi Master Netzwerk" aktiviert lassen, sofern Sie nicht mit einem Modem arbeiten. Ist dies der Fall, deaktivieren Sie das Kontrollkästchen, da STEP 7-Micro/WIN diese Funktionalität nicht unterstützt.
4. Stellen Sie für die Übertragungsgeschwindigkeit die Baudrate ein, mit der STEP 7-Micro/WIN im Netz kommunizieren soll. Ausführliche Informationen zu gültigen Baudraten für die verschiedenen CPUs entnehmen Sie Kapitel 9, Tabelle 9-1.
5. Geben Sie die höchste Teilnehmeradresse an. Hierbei handelt es sich um die Adresse, über die hinaus STEP 7-Micro/WIN nicht weiter nach anderen Mastern im Netz sucht.

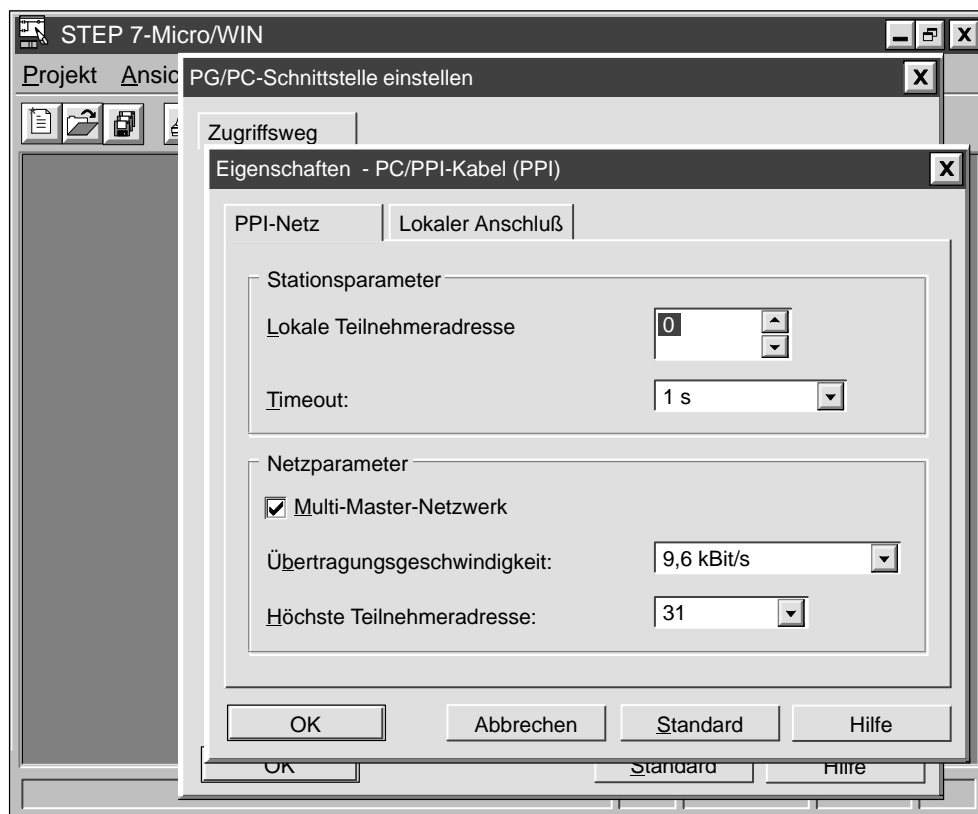


Bild 3-9 Dialogfeld "Eigenschaften - PC/PPI-Kabel (PPI)", Register "PPI-Netz"

6. Öffnen Sie das Register "Lokaler Anschluß" (siehe Bild 3-10).
7. Im Register "Lokaler Anschluß" geben Sie den COM-Port an, an den das PC/PPI-Kabel angeschlossen ist. Arbeiten Sie mit einem Modem, geben Sie den COM-Port an, an den das Modem angeschlossen ist, und aktivieren das Kontrollkästchen "Modemverbindung".
8. Klicken Sie auf "OK", um das Dialogfeld "PG/PC-Schnittstelle einstellen" zu verlassen.



Bild 3-10 Dialogfeld "Eigenschaften - PC/PPI-Kabel (PPI)", Register "Lokaler Anschluß"

Einrichten der Parameter für die MPI-Baugruppe (PPI)

In diesem Abschnitt wird erläutert, wie Sie zum Einrichten der PPI-Parameter für die folgenden Betriebssysteme und die folgende Hardware vorgehen:

- Windows 3.1: MPI-ISA-Karte (einschließlich der in SIMATIC Programmiergeräten)
- Windows 95 oder Windows NT 4.0:
 - MPI-ISA-Karte
 - MPI-ISA-Karte, On-Board (MPI-Baugruppen für SIMATIC Programmiergeräte)
 - CP 5411
 - CP 5511
 - CP 5611

Verwenden Sie eine der für das PPI-Protokoll aufgeführten MPI-Baugruppen oder CPs (Kommunikationsprozessoren), und Sie klicken im Dialogfeld "PG/PC-Schnittstelle einstellen" auf die Schaltfläche "Eigenschaften", dann werden die Eigenschaften für die installierte Baugruppe angezeigt (siehe Bild 3-11).

Gehen Sie folgendermaßen vor:

1. Geben Sie im Register "PPI-Netz" eine lokale Teilnehmeradresse an. Dies ist die Adresse für STEP 7-Micro/WIN im Netz des Automatisierungssystems.
2. Geben Sie einen Wert für Timeout an. Dieser Wert gibt an, wie lange die Kommunikationstreiber versuchen sollen, eine Kommunikation aufzubauen. Der standardmäßig eingestellte Wert dürfte ausreichend sein.
3. Geben Sie an, ob Sie möchten, daß STEP 7-Micro/WIN in einem Netz mit mehreren Mastern kommuniziert. Ausführliche Informationen hierzu entnehmen Sie Kapitel 9. Sie brauchen das Kontrollkästchen "Multi Master Netzwerk" nicht zu deaktivieren.
4. Stellen Sie für die Übertragungsgeschwindigkeit die Baudrate ein, mit der STEP 7-Micro/WIN im Netz kommunizieren soll. Ausführliche Informationen zu gültigen Baudraten für die verschiedenen CPUs entnehmen Sie Kapitel 9, Tabelle 9-1.
5. Geben Sie die höchste Teilnehmeradresse an. Hierbei handelt es sich um die Adresse, über die hinaus STEP 7-Micro/WIN nicht weiter nach anderen Mastern im Netz sucht.
6. Klicken Sie auf "OK", um das Dialogfeld "PG/PC-Schnittstelle einstellen" zu verlassen.

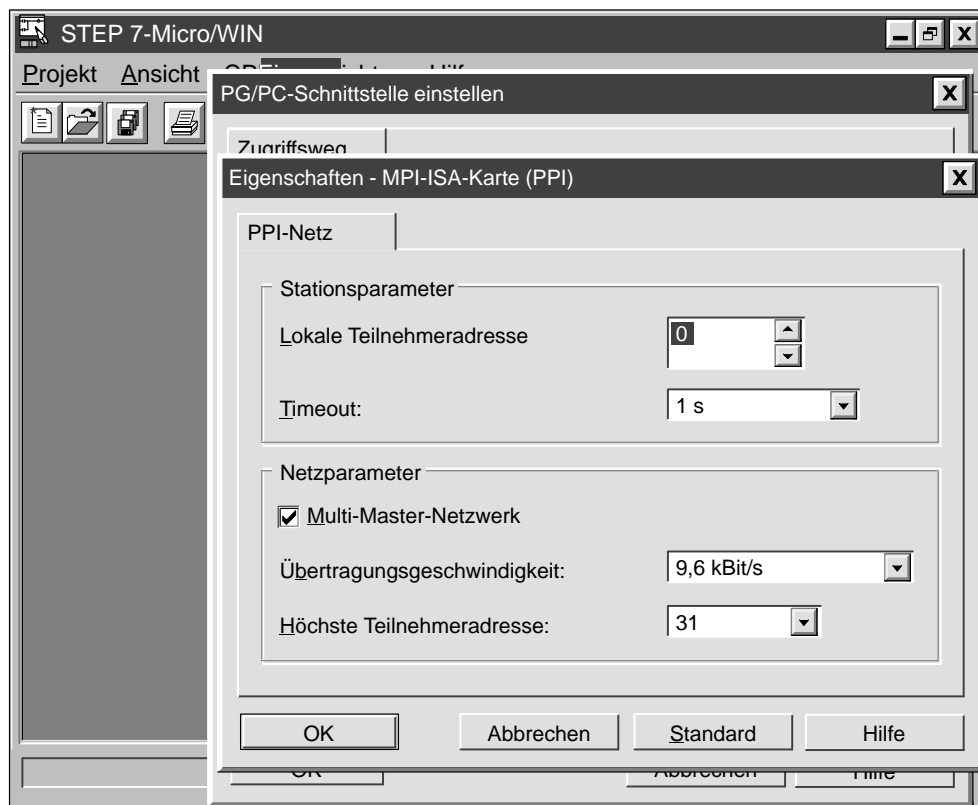


Bild 3-11 Dialogfeld "Eigenschaften - MPI-ISA-Karte (PPI)", Register "PPI-Netz"

Einrichten der Parameter für die MPI-Baugruppe (MPI)

In diesem Abschnitt wird erläutert, wie Sie zum Einrichten der MPI-Parameter für die folgenden Betriebssysteme und die folgende Hardware vorgehen:

- Windows 3.1: MPI-ISA-Karte (einschließlich der in SIMATIC Programmiergeräten)
- Windows 95 oder Windows NT 4.0:
 - MPI-ISA-Karte
 - MPI-ISA-Karte, On-Board (MPI-Baugruppen für SIMATIC Programmiergeräte)
 - CP 5411
 - CP 5511
 - CP 5611

Verwenden Sie eine der für das MPI-Protokoll aufgeführten MPI-Baugruppen oder CPs (Kommunikationsprozessoren), und Sie klicken im Dialogfeld "PG/PC-Schnittstelle einstellen" auf die Schaltfläche "Eigenschaften", dann werden die Eigenschaften für die installierte Baugruppe angezeigt (siehe Bild 3-12).

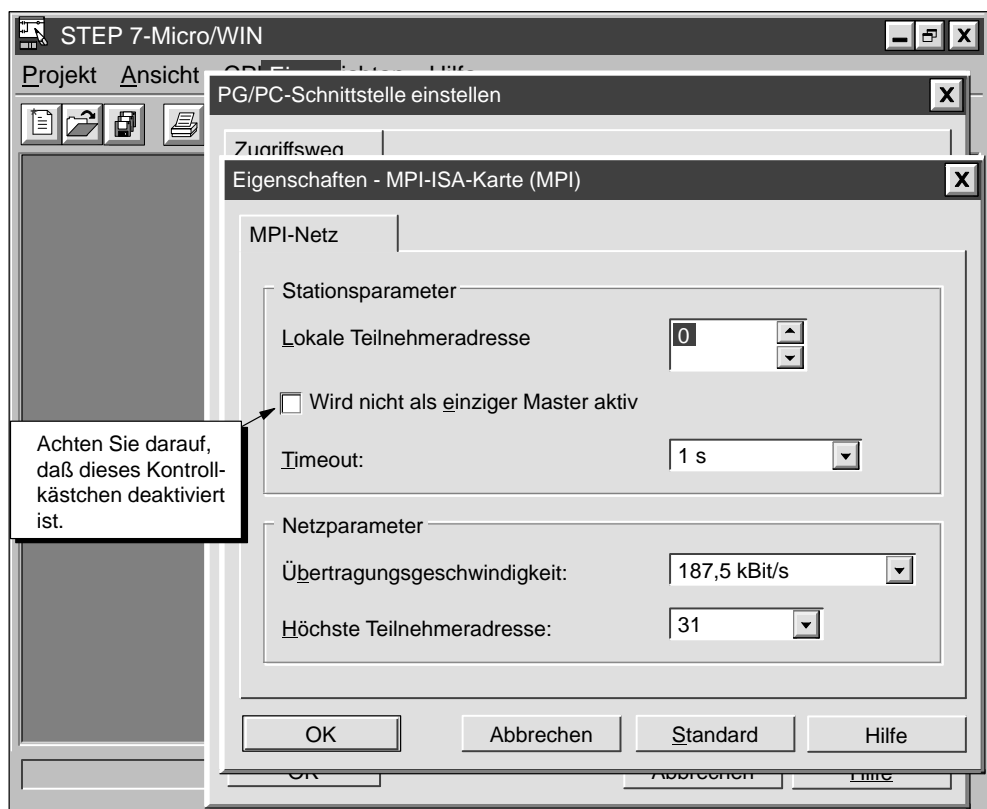


Bild 3-12 Dialogfeld "Eigenschaften - MPI-ISA-Karte (MPI)", Register "MPI-Netz"

Gehen Sie folgendermaßen vor:

1. Geben Sie im Register "MPI-Netz" eine lokale Teilnehmeradresse an. Dies ist die Adresse für STEP 7-Micro/WIN im Netz des Automatisierungssystems.
2. Achten Sie darauf, daß das Kontrollkästchen "Wird als einziger Master aktiv" nicht aktiviert ist, ganz gleich, wieviele Master in Ihrem Netz vorhanden sind. Ist das Kontrollkästchen mit einem Haken markiert, klicken Sie es an, um die Funktion zu deaktivieren. Achten Sie darauf, daß Sie das Kommunikationskabel zwischen dem Programmiergerät und der CPU anschließen, bevor Sie die Kommunikation initiieren. Rufen Sie die Kommunikation auf, bevor Sie das Programmiergerät an das Netz der CPU einschließlich einem oder mehreren Mastern angeschlossen haben, wird die Kommunikation unterbrochen und das Netz wird erneut initialisiert.
3. Geben Sie einen Wert für Timeout an. Dieser Wert gibt an, wie lange die Kommunikationstreiber versuchen sollen, eine Kommunikation aufzubauen. Der standardmäßig eingestellte Wert dürfte ausreichend sein.
4. Stellen Sie für die Übertragungsgeschwindigkeit die Baudrate ein, mit der STEP 7-Micro/WIN im Netz kommunizieren soll. Da Sie wahrscheinlich über die DP-Schnittstelle der CPU 215 kommunizieren, können Sie eine beliebige Übertragungsgeschwindigkeit bis maximal 12 Mbaud angeben. Ausführliche Informationen zu gültigen Baudraten für die verschiedenen CPUs entnehmen Sie Kapitel 9, Tabelle 9-1.
5. Geben Sie die höchste Teilnehmeradresse an. Hierbei handelt es sich um die Adresse, über die hinaus STEP 7-Micro/WIN nicht weiter nach anderen Mastern im Netz sucht.
6. Klicken Sie auf "OK", um das Dialogfeld "PG/PC-Schnittstelle einstellen" zu verlassen.

Fehlerbehebung beim Einrichten der MPI-Kommunikation für 16-Bit-Anwendungen

Die Option "MPI-Baugruppe" aktiviert die MPI-Treiber in der Konfigurationsdatei S7DPMPI.INI, die während der Installation von STEP 7-Micro/WIN in dem Windows-Verzeichnis abgelegt wurde.

Tritt ein Interruptfehler auf, müssen Sie für die MPI-Baugruppe einen freien Hardware-Interrupt (IRQ) einrichten. Bei dem voreingestellten Interrupt handelt es sich um IRQ 5. Im Feld "IRQ" können Sie die Nummer des Interrupts angeben, der von der MPI-Baugruppe verwendet werden soll. Tritt ein Interruptfehler auf, bedeutet dies, daß IRQ 5 bereits belegt ist. Zum Einstellen eines anderen Hardware-Interrupts gehen Sie folgendermaßen vor:

1. Wählen Sie den Menübefehl **Einrichten ► Kommunikation...** Daraufhin wird das Dialogfeld "Kommunikation" angezeigt. Geben Sie bei den Optionen für den Hardware-Interrupt einen neuen Wert an.
2. Bestätigen Sie Ihre Eingaben mit "OK" oder der Eingabetaste. Die Software ändert nun automatisch die Datei S7DPMPI.INI und meldet Ihnen, wenn Sie die Anwendung verlassen müssen.
3. Starten Sie STEP 7-Micro/WIN neu und wählen Sie wieder die Option MPI.

Hinweis

Die S7-200 CPUs mit mehr als einer Kommunikationsschnittstelle verfügen über folgende voreingestellte Adressen:

- CPU 215 Schnittstelle 0: 2
 Schnittstelle 1: 126
 - CPU 216 Schnittstelle 0: 2
 Schnittstelle 1: 2
-

Fehlerbehebung beim Einrichten der MPI-Kommunikation unter Windows NT 4.0

Das Einrichten der MPI-Baugruppe unter Windows NT 4.0 ist nicht ganz einfach. Haben Sie Probleme beim Einrichten (vorausgesetzt, Sie haben die MPI-Baugruppe bereits als Hardware installiert), gehen Sie folgendermaßen vor:

1. Stellen Sie sicher, daß die MPI-Baugruppe nicht defekt ist. Hierzu gibt es verschiedene Methoden: Sie können die Baugruppe unter Windows 95 prüfen und sie unter STEP 7-Micro/WIN Version 2.0 testen.
2. Prüfen Sie anhand der DIP-Schalter auf der MPI-Baugruppe wieviel Speicherplatz Sie für die Baugruppe reservieren müssen (siehe Tabelle 3-2).
3. Prüfen Sie, welche Ressourcen Windows NT für die Baugruppe reserviert hat und stellen Sie sicher, daß die reservierten Ressourcen der Schalterstellung entsprechen. Gehen Sie hierzu folgendermaßen vor:
 - a. Rufen Sie das Dialogfeld "PG/PC-Schnittstelle einstellen" auf.
 - b. Wählen Sie die Schaltfläche "Installieren...".
 - c. Markieren Sie im Feld "Installiert" die MPI-Karte.
 - d. Wählen Sie die Schaltfläche "Ressourcen". Diese Schaltfläche ist nur unter Windows NT verfügbar.
4. Ist die Zuordnung der Ressourcen korrekt und Ihre Baugruppe funktioniert trotzdem nicht, versuchen Sie, den der Baugruppe zugeordneten Hardware-Interrupt zu ändern. Es kann sein, daß ein Konflikt mit anderer Hardware vorliegt. Diese Änderung können Sie im Dialogfeld "Ressourcen" eingeben.
5. Haben Sie bereits alle Interrupts getestet und die Baugruppe funktioniert immer noch nicht, müssen Sie die DIP-Schalter auf der Baugruppe so ändern, daß eine andere Adresse eingestellt wird. Wiederholen Sie Schritt 3. Wiederholen Sie Schritt 4.
6. Haben Sie die hier beschriebenen Vorgehensweisen ausgeführt, doch Ihre Baugruppe funktioniert immer noch nicht, sind wahrscheinlich alle Ihre Ressourcen von anderer Hardware belegt. Sie können versuchen, ein Gerät Ihrer übrigen Hardware zu entfernen bzw. zu deaktivieren (z.B. eine Soundkarte), um so einige Ressourcen verfügbar zu machen. Beginnen Sie dann erneut mit Schritt 2.
7. Bleiben alle diese Versuche erfolglos, setzen Sie andere Kommunikationstreiber ein.

Die im Lieferumfang der MPI-Baugruppe enthaltene Dokumentation erläutert die möglichen Hardware-Konflikte ausführlich.

Tabelle 3-2 Erforderlicher Speicherplatz für eine MPI-Baugruppe

Schalter 1	Schalter 2	Schalter 3	Speicher
EIN	EIN	EIN	#000C8000-000C87FF
EIN	EIN	AUS	#000C9000-000C97FF
EIN	AUS	EIN	#000CC000-000CC7FF
EIN	AUS	AUS	#000D0000-000D07FF
AUS	EIN	EIN	#000D1000-000D17FF
AUS	EIN	AUS	#000DC000-000DC7FF
AUS	AUS	EIN	#000E1000-000E17FF

Anschließen einer CPU 215 als dezentrale Peripherie

Sie können die CPU 215 an ein PROFIBUS-Netz anschließen, in dem die CPU 215 als dezentrale Peripherie für ein Automatisierungssystem S7-300 oder S7-400 oder für einen anderen PROFIBUS-Master dient (siehe Bild 3-13).

Die CPU 215 verfügt über eine Schnittstelle auf der CPU, die mit "DP" gekennzeichnet ist. Über diese DP-Schnittstelle verbinden Sie Ihre CPU 215 als Modul der dezentralen Peripherie mit einem PROFIBUS-Netz.

Die einzige Einstellung, die Sie für die CPU 215 vornehmen müssen, damit Sie sie als PROFIBUS-Slave einsetzen können, ist die Teilnehmeradresse der DP-Schnittstelle der CPU. Diese Adresse muß der Adresse in der Konfiguration des Masters entsprechen. Der Master konfiguriert die CPU 215. Ausführliche Informationen zur Kommunikation mit der dezentralen Peripherie (DP) entnehmen Sie dem Abschnitt 9.5.

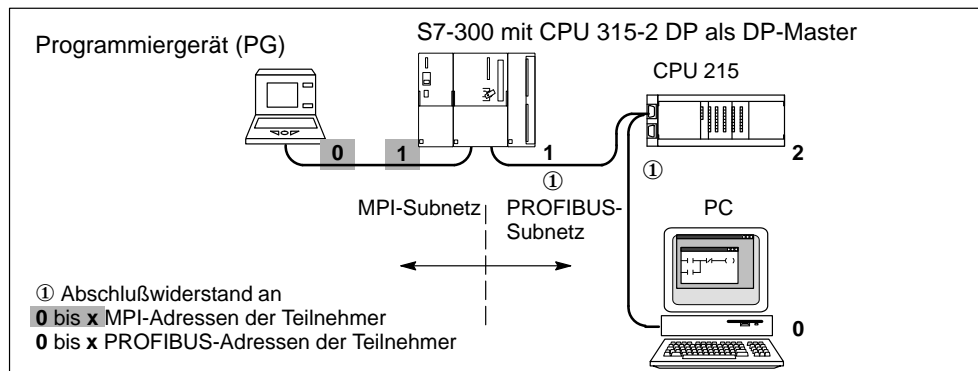


Bild 3-13 CPU 215 in einem PROFIBUS-Subnetz mit MPI-Subnetz

Verbinden einer S7-200 CPU mit einem STEP 7-Micro/WIN-Master per Modem

Wenn Sie STEP 7-Micro/WIN auf einem PC unter Windows 3.1x, Windows 95 oder Windows NT oder auf einem SIMATIC Programmiergerät (z.B. PG 740) als einzigen Master installiert haben, können Sie zu folgenden Geräten der S7-200 eine Verbindung per Modem aufbauen:

- zu einer S7-200 CPU als Slave
- zu mehreren S7-200 CPUs als Slaves im Netz

Je nachdem, ob Sie die Verbindung nur zu einer S7-200 CPU oder zu einem Netz aus S7-200 CPUs herstellen möchten, benötigen Sie folgende Kabel und Adapter (siehe Bild 3-14):

- Ein an beiden Enden RS-232-fähiges Kabel zum Verbinden von PC bzw. SIMATIC Programmiergerät und 11-Bit-Modem (voll duplex), das an die Telefonleitung angeschlossen ist
- Einen Null-Modem-Adapter, über den Sie das Modem am anderen Ende der Telefonleitung an ein PC/PPI-Kabel anschließen
- Ein PC/PPI-Kabel, mit dem Sie den Null-Modem-Adapter an eine der folgenden Schnittstellen anschließen:
 - an die Kommunikationsschnittstelle der S7-200 CPU (siehe Bild 3-14)
 - an einen Siemens Programmierschnittstellenstecker in einem PROFIBUS-Netz (siehe Bild 9-3)

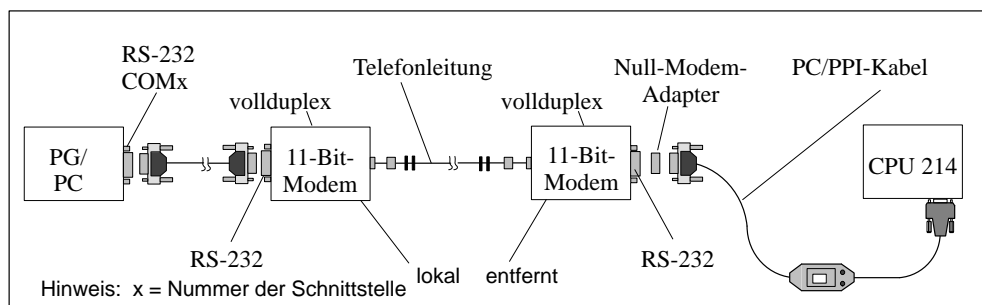


Bild 3-14 Kommunikation mit der S7-200 über 11-Bit-Modems

Da in einer solchen Konfiguration immer nur ein Master aktiv sein darf, gibt es kein Token Passing. Die Konfiguration unterstützt nur das PPI-Protokoll. Damit über die PPI-Schnittstelle kommuniziert werden kann, muß das Modem für das Automatisierungssystem S7-200 mit 11-Bit-Zeichenfolgen arbeiten. Das Automatisierungssystem S7-200 benötigt ein Startbit, acht Datenbits, ein Paritätsbit (gerade Parität), ein Stoppbit, asynchrone Kommunikation und eine Übertragungsgeschwindigkeit von 9600 Baud für die PPI-Kommunikation. Viele Modems unterstützen dieses Datenformat nicht. Das Modem benötigt die in Tabelle 3-3 aufgeführten Einstellungen.

Bild 3-15 zeigt die Anschlußbelegung des Null-Modem-Adapters. Ausführliche Informationen zur Kommunikation im Netz mit Hilfe des PC/PPI-Kabels entnehmen Sie dem Kapitel 9.

Tabelle 3-3 Erforderliche Modem-Einstellungen

Datenformat in Bits	Übertragungsgeschwindigkeit zwischen Modem und PC	Übertragungsgeschwindigkeit in der Leitung	Andere Funktionen
8 Datenbits	9600 Baud	9600 Baud	DTR-Signal ignorieren
1 Startbit			Keine Flußkontrolle
1 Stoppbit			
1 Paritätsbit (gerade)			

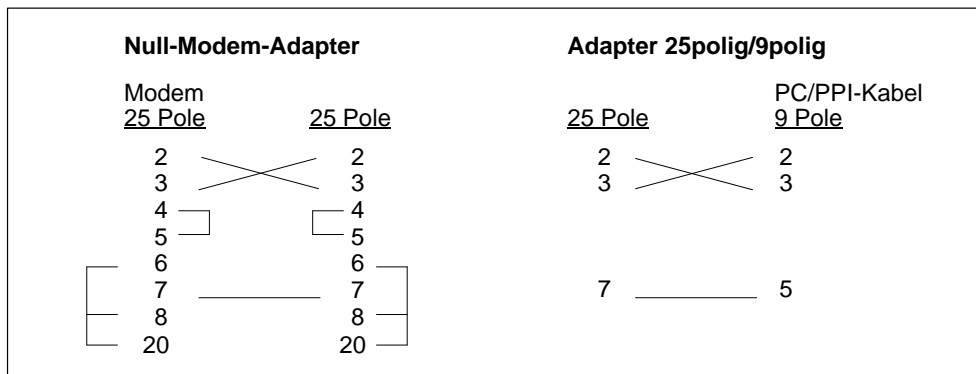


Bild 3-15 Anschlußbelegung eines Null-Modem-Adapters

Einrichten der Kommunikationsparameter bei Verwendung von Modems

Zum Einrichten der Parameter für die Kommunikation zwischen Programmiergerät bzw. PC und der CPU mit Hilfe von Modems verwenden Sie die Baugruppenparametrierung für das PC/PPI-Kabel. Ansonsten können Sie das Modem nicht einrichten. Stellen Sie sicher, das Sie die Funktion zum Einrichten des Modems aufrufen können und gehen Sie anschließend zum Einrichten der Parameter folgendermaßen vor:

Hinweis

Die im folgenden beschriebene Konfiguration bezieht sich auf das Modem Multi Tech MultiModemZDX MT1932ZDX. Arbeiten Sie mit einer anderen Art von Modem, wählen Sie im Dialogfeld "Modem einrichten" im Feld "Ausgewähltes Modem" die Option "Anwenderdefiniert". Bei Ihrem Modem muß es sich um ein 11-Bit-Modem handeln, das mit einer Geschwindigkeit von 9600 Baud kommunizieren kann. Nehmen Sie beim Eingeben der Parameter in den Registern des Dialogfelds "Modem einrichten" das Handbuch Ihres Modems zu Hilfe.

1. Wählen Sie den Menübefehl **Einrichten ► Kommunikation....**

Wird im Dialogfeld "Kommunikation" im Feld "Baugruppenparametrierung" die Option "PC/PPI-Kabel (PPI)" angezeigt, dann wählen Sie die Schaltfläche "PG/PC-Schnittstelle..." und fahren mit Schritt 3. fort.

Wird als Baugruppenparametrierung nicht "PC/PPI-Kabel (PPI)" angezeigt, dann wählen Sie die Schaltfläche "PG/PC-Schnittstelle einstellen..." und fahren mit Schritt 2. fort.

- Im Register "Zugriffsweg" wählen Sie in dem Feld "Benutzte Baugruppenparametrierung" die Option "PC/PPI-Kabel (PPI)". Steht Ihnen diese Option nicht zur Verfügung, müssen Sie das PC/PPI-Kabel installieren. Informationen hierzu entnehmen Sie Abschnitt 3.1.
- Wählen Sie die Schaltfläche "Eigenschaften". Daraufhin wird das Dialogfeld "Eigenschaften – PC/PPI-Kabel (PPI)" angezeigt.
- Öffnen Sie im Dialogfeld "Eigenschaften – PC/PPI-Kabel (PPI)" das Register "Lokaler Anschluß".
- Stellen Sie sicher, daß das Kontrollkästchen "Modemverbindung" aktiviert ist. Ist das Kontrollkästchen nicht durch einen Haken gekennzeichnet, klicken Sie es an und aktivieren es dadurch.
- Klicken Sie auf die Schaltfläche "OK". Daraufhin wird wieder das Register "Zugriffsweg" angezeigt.
- Klicken Sie auf die Schaltfläche "OK". Daraufhin wird wieder das Dialogfeld "Kommunikation" angezeigt.

8. Wählen Sie die Schaltfläche "Modem einrichten...". Daraufhin wird das Dialogfeld "Modem einrichten" angezeigt. (Sie können die Schaltfläche "Modem einrichten" auch im Dialogfeld "Verbinden" wählen, das Sie über den Menübefehl **Einrichten ► Modem verbinden...** aufrufen.)

Im Dialogfeld "Modem einrichten" bietet Ihnen das Register "Allgemeines" die Anforderungen an das Modem (11-Bit-Zeichenfolge) und führt die Hardware auf, die Sie benötigen. Bild 3-14 zeigt diese Hardware-Komponenten.

9. Öffnen Sie das Register "Konfiguration lokales Modem" (siehe Bild 3-16).
10. Wählen Sie im Register "Konfiguration lokales Modem" im Feld "Ausgewähltes Modem" das Modem Multi Tech MultiModemZDX MT1932ZDX.

Die einzigen anderen Felder in diesem Register, die Sie noch ändern können, sind die Telefonnummer und der Wert für Timeout. Der Timeout-Wert gibt an, wie lange das lokale Modem versuchen soll, zu dem entfernten Modem eine Verbindung herzustellen. Läuft die in dem Feld "Timeout" in Sekunden angegebene Zeit ab, bevor die Verbindung aufgebaut ist, mißlingt der Versuch, eine Verbindung herzustellen.

11. Möchten Sie die Konfiguration Ihres lokalen Modems testen, wählen Sie die Schaltfläche "Modem testen", während das Modem lokal an Ihren Rechner (PG bzw. PC) angeschlossen ist.
12. Trennen Sie die Verbindung zum lokalen Modem und schließen Sie das entfernte Modem lokal an Ihren Rechner (PG bzw. PC) an.

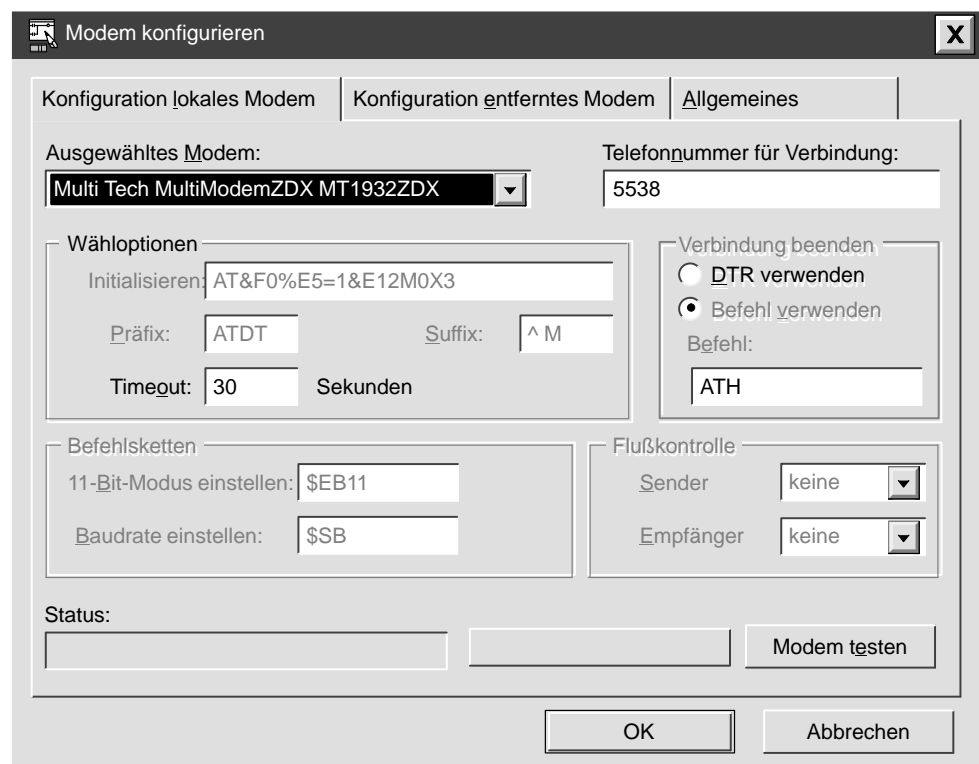


Bild 3-16 Register "Konfiguration lokales Modem" im Dialogfeld "Modem einrichten"

13. Öffnen Sie das Register "Konfiguration entferntes Modem" (siehe Bild 3-17).
14. Wählen Sie im Register "Konfiguration lokales Modem" im Feld "Ausgewähltes Modem" das Modem Multi Tech MultiModemZDX MT1932ZDX.
15. Wählen Sie die Schaltfläche "Modem programmieren". Daraufhin werden die Parameter in einen Speicherchip im entfernten Modem übertragen.
16. Möchten Sie prüfen, ob Ihr entferntes Modem richtig programmiert ist, wählen Sie die Schaltfläche "Modem testen".
17. Klicken Sie auf die Schaltfläche "OK". Daraufhin wird wieder das Dialogfeld "Kommunikation" angezeigt.



Bild 3-17 Register "Konfiguration entferntes Modem" im Dialogfeld "Modem einrichten"

18. Trennen Sie die Verbindung zwischen entferntem Modem und lokalem Rechner (PG bzw. PC).
19. Schließen Sie das entfernte Modem an Ihr Automatisierungssystem S7-200 an.
20. Schließen Sie das lokale Modem an Ihr Programmiergerät bzw. Ihren PC an.
21. Stellen Sie sicher, daß Ihr Setup dem im Register "Allgemeines" im Dialogfeld "Modem einrichten" gezeigten Setup entspricht (siehe auch Bild 3-14).
22. Haben Sie alle Einstellungen für die Konfiguration vorgenommen, wählen Sie die Schaltfläche "OK" und verlassen Sie das Dialogfeld "Kommunikation".
23. Zum Verbinden Ihres Modems wählen Sie den Menübefehl **Einrichten ► Modem verbinden...** Daraufhin wird das Dialogfeld "Verbinden" angezeigt (siehe Bild 3-18).
24. Haben Sie im Dialogfeld "Modem einrichten" im Register "Konfiguration lokales Modem" noch keine Telefonnummer im Feld "Telefonnummer für Verbindung" eingegeben oder wenn Sie die dort angegebene Telefonnummer ändern möchten, können Sie hier im Feld "Telefonnummer" eine Telefonnummer angeben.
25. Wählen Sie die Schaltfläche "Verbinden". Nun haben Sie Ihr Modem vollständig eingerichtet.



Bild 3-18 Dialogfeld "Verbinden"

3.4 Einrichten der Voreinstellungen in STEP 7-Micro/WIN

Bevor Sie ein neues Projekt anlegen, richten Sie sich die Voreinstellungen für Ihre Programmierumgebung ein. Zum Auswählen der Voreinstellungen gehen Sie folgendermaßen vor:

1. Wählen Sie den Menübefehl **Einrichten ► Voreinstellungen...** (siehe Bild 3-19).
2. Wählen Sie in dem folgenden Dialogfeld die gewünschten Voreinstellungen zum Programmieren aus.
3. Bestätigen Sie Ihre Angaben mit "OK" oder drücken Sie die Eingabetaste.

Hinweis

Wenn Sie eine andere Sprache ausgewählt haben, wird die neue Einstellung erst aktiviert, wenn Sie STEP 7-Micro/WIN verlassen und neu starten.



Bild 3-19 Auswählen der Voreinstellungen zum Programmieren

3.5 Erstellen und Speichern eines Projekts

Bevor Sie ein Programm erstellen können, müssen Sie ein Projekt einrichten bzw. öffnen. Wenn Sie ein neues Projekt anlegen, öffnet STEP 7-Micro/WIN die folgenden Editoren:

- Editor zum Bearbeiten von Kontaktplänen oder Anweisungslisten (je nachdem, welchen Editor Sie voreingestellt haben)
- Editor zum Bearbeiten von Datenbausteinen
- Statustabelle
- Symboltabelle

Anlegen eines neuen Projekts

Über das Menü Projekt können Sie ein neues Projekt anlegen (siehe Bild 3-20). Wählen Sie den Menübefehl **Projekt ► Neu....** Daraufhin öffnet sich das Dialogfeld "CPUs". Wenn Sie in dem aufklappbaren Listenfeld eine CPU auswählen, zeigt die Software nur die Optionen an, die für die ausgewählte CPU zur Verfügung stehen. Wählen Sie "Keine", enthält Ihr Programm keine CPU-spezifischen Einschränkungen. Wenn Sie das Programm in die CPU laden, prüft die CPU, ob Sie Funktionen einsetzen, die nicht verfügbar sind. Enthält Ihr Programm beispielsweise eine Operation, die von der jeweiligen CPU nicht unterstützt wird, dann wird das Programm zurückgewiesen.

Hinweis

STEP 7-Micro/WIN prüft nicht den Bereich der Parameter. Sie können z. B. VB9999 als Parameter einer KOP-Operation eingeben, obwohl dieser Parameter ungültig ist.

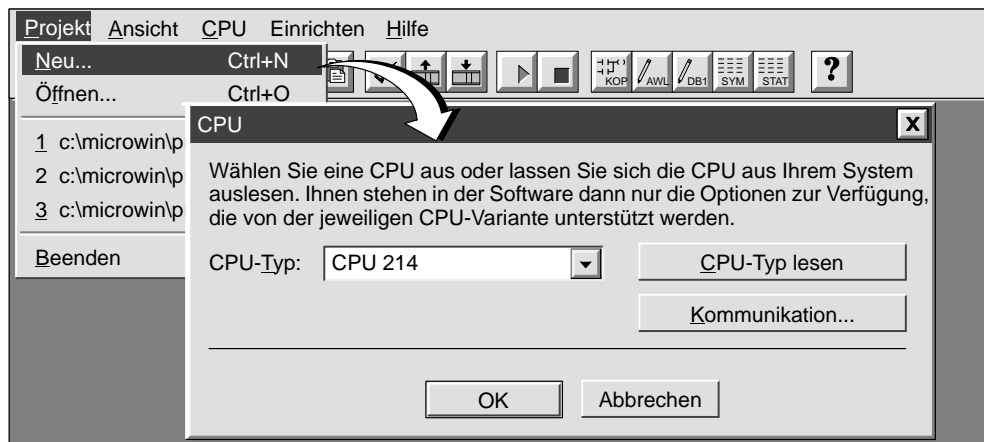


Bild 3-20 Anlegen eines neuen Projekts

Speichern eines Projekts

Zum Speichern aller Komponenten Ihres Projekts wählen Sie den Menübefehl **Projekt ► Speichern** oder klicken Sie auf die folgende Schaltfläche: .

Sie können das Projekt auch unter einem anderen Namen oder in einem anderen Verzeichnis ablegen. Wählen Sie hierzu den Menübefehl **Projekt ► Speichern unter....**

3.6 Erstellen eines Programms

In STEP 7-Micro/WIN können Sie Ihr Anwenderprogramm (OB1) mit dem Kontaktplan- oder dem Anweisungslisten-Editor erstellen.

Eingeben des Programms im Kontaktplan

Im KOP-Editor können Sie Ihr Programm mit Hilfe von graphischen Symbolen schreiben (siehe Bild 3-21). Die Symbolleiste enthält einige der am häufigsten verwendeten KOP-Elemente, die Sie in Ihr Programm eingeben können. Das erste aufklappbare Listenfeld (ganz links) enthält die Operationsfamilien. Sie greifen auf die Familien zu, indem Sie sie anklicken oder die Taste F2 drücken. Haben Sie eine Familie ausgewählt, enthält das zweite aufklappbare Listenfeld die Operationen der jeweiligen Familien. Wenn Sie sich eine Liste aller verfügbaren Operationen in alphabetischer Reihenfolge anzeigen lassen möchten, drücken Sie die Taste F9 oder wählen Sie "Alle Operationen". Mit dem Menübefehl **Ansicht ► Funktionsleiste der Operationen** zeigen Sie im KOP-Editor die Funktionsleiste der Operationen an

Zu jedem Netzwerk gibt es zwei Kommentare, die im folgenden beschrieben sind:

- Einzeilige Netzwerkkommentare sind im KOP-Editor immer sichtbar. Sie bearbeiten diese Kommentare, indem Sie den Kommentar an einer beliebigen Stelle anklicken.
- Netzwerkkommentare, die sich auf mehrere Zeilen ausdehnen, bearbeiten Sie durch Doppelklicken auf die Netzwerknummer. Mehrzeilige Netzwerkkommentare können nur in einem Dialogfeld angezeigt werden, sind jedoch im Ausdruck vollständig vorhanden.

Zum Eingeben Ihres Programms gehen Sie folgendermaßen vor:

1. Zum Eingeben eines Titels für das Programm wählen Sie den Menübefehl **Bearbeiten ► Programmtitel...** Geben Sie den neuen Programmtitel ein und bestätigen Sie mit "OK".
2. Wenn Sie KOP-Elemente eingeben möchten, klicken Sie auf die entsprechende Schaltfläche oder wählen ein Element aus der Operationsliste aus.
3. Geben Sie die Operanden bzw. Parameter in den Textfeldern ein und drücken Sie die Eingabetaste.

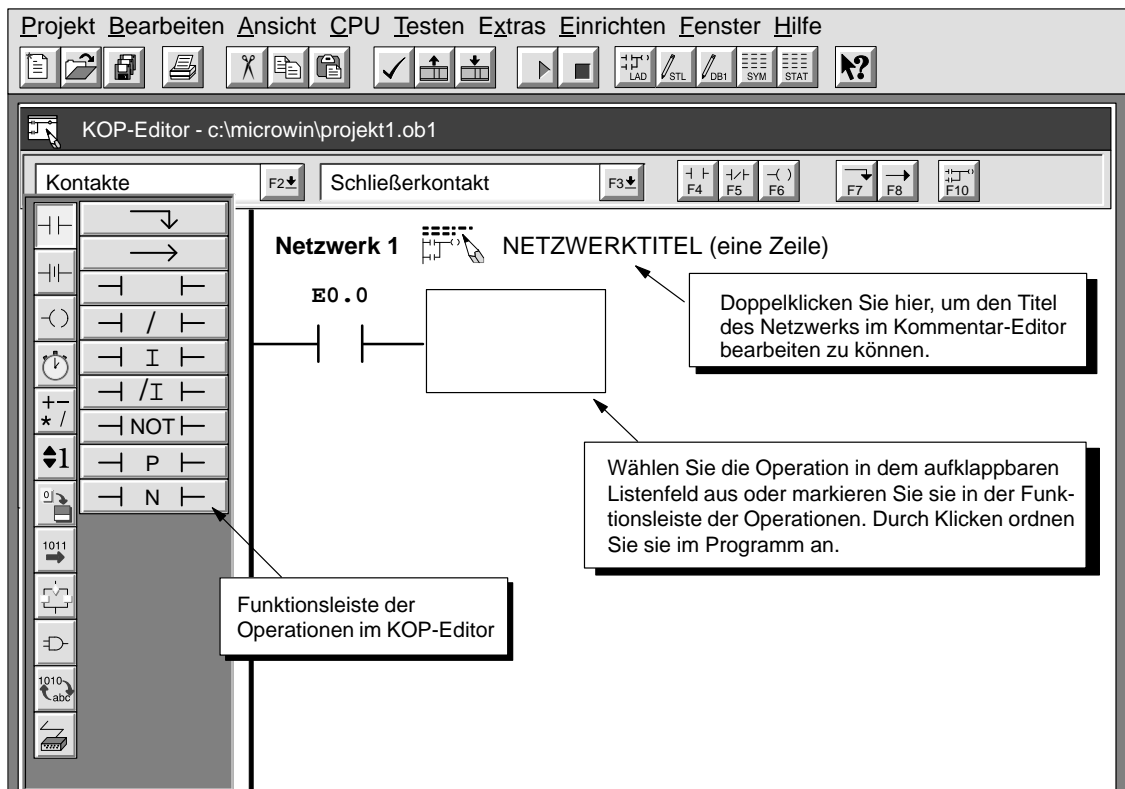


Bild 3-21 Fenster der KOP-Editors

Eingeben Ihres Programms in der Anweisungsliste

Bei dem AWL-Editor handelt es sich um einen Texteditor, der aufgrund des frei wählbaren Formats einen gewissen Grad an Flexibilität beim Eingeben der Programmanweisungen bietet. Bild 3-22 zeigt ein Beispiel für ein Programm in der Anweisungsliste.

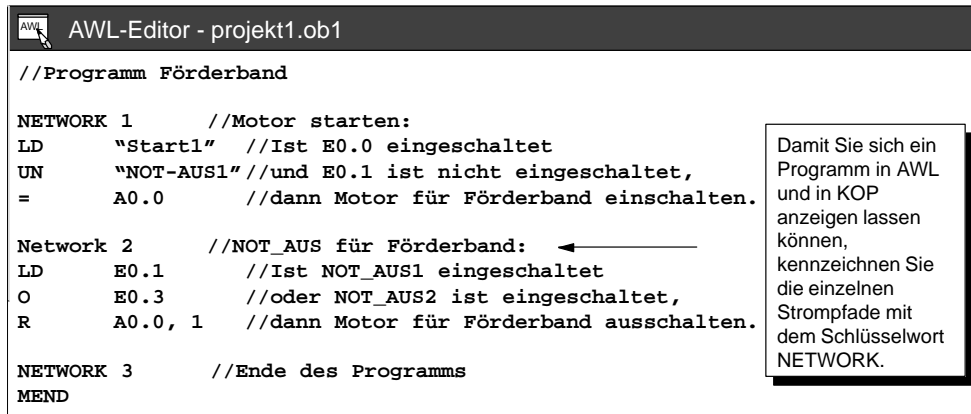



Bild 3-22 Fenster des AWL-Editors mit einem Beispielprogramm

Beachten Sie die folgenden Richtlinien beim Eingeben eines Programms in AWL:

- Damit Sie sich ein AWL-Programm in KOP anzeigen lassen können, müssen Sie das Programm in einzelne Netzwerke unterteilen, indem Sie das Schlüsselwort NETWORK eingeben. (Die Netzwerknummern werden beim Übersetzen bzw. Laden des Programms automatisch erzeugt.) Das Schlüsselwort NETWORK müssen Sie in angemessenen Abständen anordnen, damit das Programm auch in KOP angezeigt werden kann.
- Beginnen Sie jeden Kommentar mit einem doppelten Schrägstrich (//). Jede zusätzliche Kommentarzeile muß ebenfalls mit einem doppelten Schrägstrich beginnen.
- Beenden Sie jede Zeile mit einer Zeilenschaltung.
- Trennen Sie jede Operation von der Adresse bzw. dem Parameter durch Leerzeichen oder die TAB-Taste.
- Geben Sie zwischen dem Speicherbereich und der Adresse kein Leerzeichen ein (geben Sie z. B. E0.0, und nicht E 0.0 ein).
- Trennen Sie jeden Operanden innerhalb der Anweisung mit Komma, Leerzeichen oder der TAB-Taste.
- Geben Sie symbolische Namen in Hochkommata ein. Enthält Ihre Symboltabelle beispielsweise den symbolischen Namen Start1 für die Adresse E0.0, dann geben Sie die Anweisung folgendermaßen ein:

```
LD "Start1"
```

Übersetzen des Programms

Nachdem Sie ein oder mehrere Netzwerke eingegeben haben, können Sie die Syntax des Programms prüfen. Wählen Sie hierzu den Menübefehl **CPU ► Übersetzen** oder klicken Sie auf die Schaltfläche zum Übersetzen: .

Laden des Programms in die CPU

Wenn Sie Ihr Programm vollständig eingegeben haben, können Sie das Projekt in die CPU laden. Wählen Sie hierzu den Menübefehl **Projekt ► Laden aus PG...** oder klicken Sie auf die Schaltfläche zum Laden im Hauptfenster: .

Daraufhin öffnet sich ein Dialogfeld, in dem Sie die Komponenten des Projekts angeben können, die Sie in die CPU laden möchten (siehe Bild 3-23).

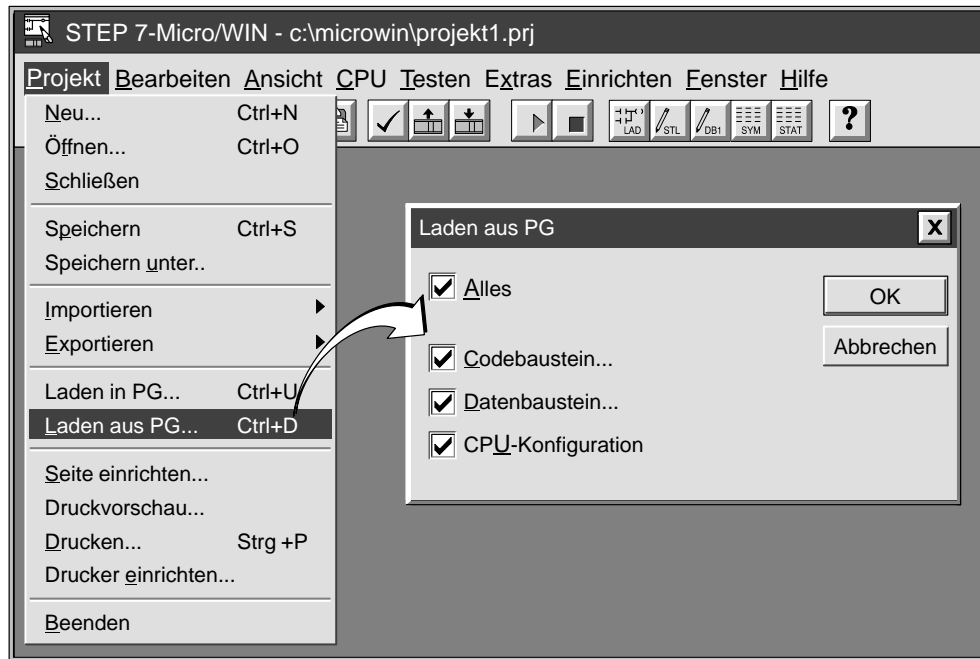


Bild 3-23 Laden der Komponenten des Projekts in die CPU

- Der Codebaustein (OB1) enthält Ihr Programm, das von der CPU bearbeitet werden soll.
- Der Datenbaustein (DB1) enthält die Werte, die von Ihrem Programm für die Initialisierung verwendet werden sollen.
- Die CPU-Konfiguration (CFG) enthält Setup-Informationen für das System. Diese umfassen Kommunikationsparameter, remanente Bereiche, Einstellungen für die Eingabefilter, Paßwörter und die Einstellungen der Ausgänge.

Bestätigen Sie Ihre Eingaben mit "OK" oder indem Sie die Eingabetaste drücken. Daraufhin werden die gewählten Komponenten in die CPU geladen.

Anzeigen eines Programms in KOP oder AWL

Sie können sich ein Programm als Kontaktplan oder als Anweisungsliste anzeigen lassen. Wählen Sie hierzu einen der Menübefehle **Ansicht ► AWL** oder **Ansicht ► KOP** (siehe Bild 3-24).

Wenn Sie sich ein Programm erst in AWL, dann in KOP und schließlich wieder in AWL anzeigen lassen, kann es sein, daß Ihr AWL-Programm folgende Unterschiede aufweist:

- Kleinbuchstaben in Anweisungen und Adressen werden zu Großbuchstaben.
- Leerzeichen zwischen Operationen und Adressen werden durch Tabulatoren ersetzt.

Wenn Sie im AWL-Editor den Menübefehl **CPU ► Übersetzen** wählen, können Sie sich die AWL-Operationen im Standardformat anzeigen lassen.

Hinweis

Bestimmte Folgen von AWL-Anweisungen können nicht in KOP angezeigt werden. In diesem Fall kennzeichnet die Meldung "Unzulässig" die Programmteile, die nicht in KOP dargestellt werden können.

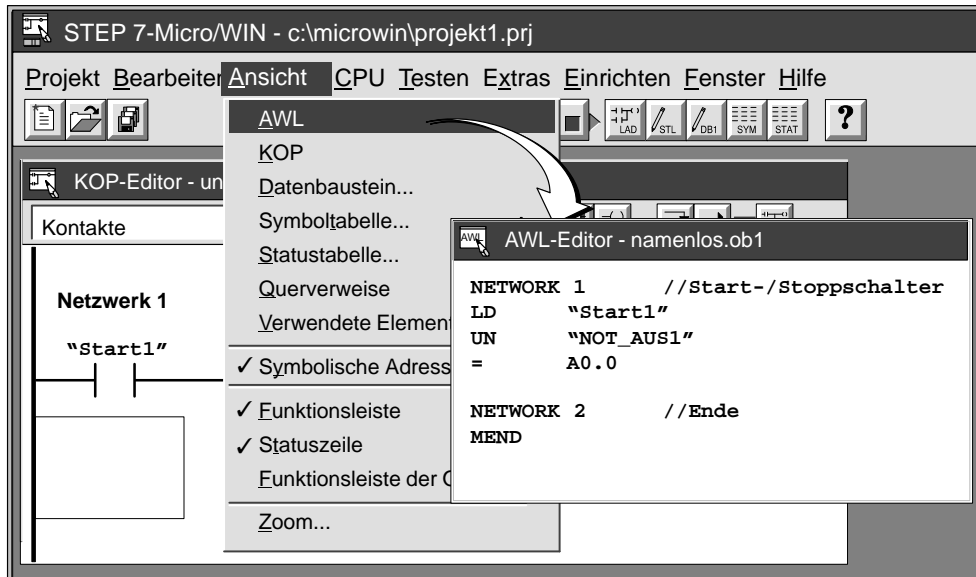


Bild 3-24 Anzeigen eines KOP-Programms in AWL

3.7 Erstellen eines Datenbausteins

Mit dem Datenbaustein-Editor können Sie die Variablen, die von Ihrem Programm verwendet werden sollen, vordefinieren oder initialisieren. Es ist nicht unbedingt erforderlich, einen Datenbaustein zu erstellen.

Der Datenbaustein-Editor wird standardmäßig als Fenstersymbol am unteren Rand des Hauptfensters angezeigt (sofern Sie dies mit dem Menübefehl **Einrichten ► Voreinstellungen...** eingestellt haben). Wenn Sie den Datenbaustein-Editor aufrufen möchten, doppelklicken Sie auf das Symbol bzw. Sie wählen den Befehl "Wiederherstellen/Vollbild" an der Schaltfläche (in Windows 95).

Eingeben von Werten für Datenbausteine

Der Datenbaustein-Editor ist ein Texteditor, der aufgrund des frei wählbaren Formats einen gewissen Grad an Flexibilität beim Eingeben von Werten für Datenbausteine bietet.

Beachten Sie die folgenden Richtlinien, wenn Sie Datenbausteine erstellen:

- Geben Sie in der ersten Spalte einer Zeile die Größe der Daten und die Anfangsadresse eines jeden Werts für den Variablenspeicher an.
- Die Anfangsadresse und die Datenwerte müssen mit einem Leerzeichen bzw. der TAB-Taste voneinander getrennt werden.

Bild 3-25 zeigt ein Beispiel für einen Datenbaustein mit Kommentaren, die jedes Element beschreiben.

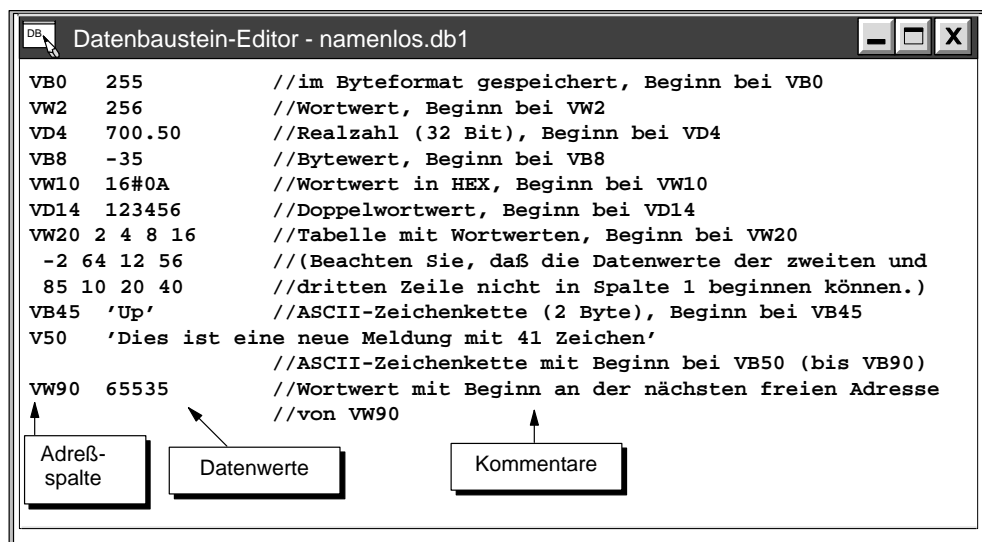


Bild 3-25 Beispiel für einen Datenbaustein



Warnung

STEP 7-Micro/WIN legt mit Hilfe der ersten Spalte in jeder Zeile des Datenbaustein-Editors die Anfangsadresse zum Speichern des Werts im Datenbaustein fest. Wenn Sie in Spalte 1 eine Zahl eingeben, wird die Zahl für die nachfolgenden Daten als die Anfangsadresse im Variablenspeicher ausgewertet. Soll die Zahl in Spalte 1 einen Datenwert und keine Adresse angeben, so kann es sein, daß dadurch unbeabsichtigt Daten, die Sie in den Datenbaustein eingegeben haben, mit den neuen Daten überschrieben werden.

Wenn Sie einen Datenbaustein, in dem Sie falsche Daten adressieren, in eine CPU laden, kann es zu unvorhersehbaren Auswirkungen im Prozeß kommen. Unvorhersehbarer Betrieb eines Automatisierungssystems kann zu Tod, schweren Körperverletzungen und/oder Sachschaden führen.

Geben Sie immer eine Größe und eine Adresse an, z. B. VB100, damit sichergestellt ist, daß die Daten in den richtigen Adressen im Variablenspeicher abgelegt werden. Prüfen Sie auch immer sorgfältig, daß Sie nicht aus Versehen einen Datenwert in Spalte 1 eingetragen haben.

Tabelle 3-4 zeigt Beispiele für die Schreibweise zum Eingeben von Werten für Datenbausteine.

Tabelle 3-4 Schreibweise zum Eingeben von Werten in einen Datenbaustein

Datentyp	Beispiel
Hexadezimal	16#AB
Ganze Zahl (dezimal)	10 <i>oder</i> 20
Ganze Zahl mit Vorzeichen (dezimal)	-10 <i>oder</i> +50
Realzahl (Gleitpunktzahl): verwenden Sie einen Punkt ("."), kein Komma (",")	10.57
Text (ASCII): Zeichenkette in Hochkommata (Hinweis: Bei ASCII-Konstanten wird das Dollarzeichen '\$' als Sonderzeichen verwendet, um ein Apostroph oder ein Dollarzeichen innerhalb einer Zeichenkette zu kennzeichnen.	'Siemens' 'So ist\$s' 'Nur \$\$25'

Tabelle 3-5 gibt die gültigen Kennzeichen zum Eingeben der Datengröße und der Anfangsadresse an.

Tabelle 3-5 Gültige Kennzeichen für die Datengröße

Datengröße	Beispiel	Beschreibung
Byte	VB10	Speichert die nachfolgenden Werte im Byteformat mit Beginn an der angegebenen Adresse.
Wort	VW22	Speichert die nachfolgenden Werte im Wortformat mit Beginn an der angegebenen Adresse.
Doppelwort	VD100	Speichert die nachfolgenden Werte im Doppelwortformat mit Beginn an der angegebenen Adresse.
Automatische Größe	V10	Speichert die Daten in der geringsten Größe (Byte, Wort bzw. Doppelwort), die zum Ablegen der Werte erforderlich ist. Die in dieser Zeile angegebenen Werte werden mit Beginn an der angegebenen Adresse im Variablenspeicher gespeichert.
Wie vorherige Größe	(Adreßspalte bleibt leer)	Speichert die Daten im Byte-, Wort- oder Doppelwortformat, je nachdem, welche Größe in der vorherigen Zeile angegeben wurde.





3.8 Arbeiten mit der Statustabelle

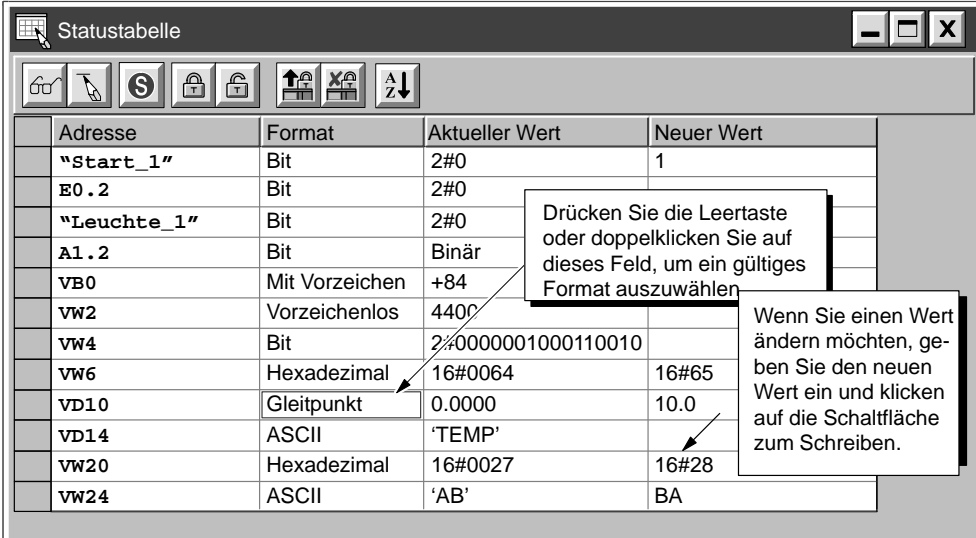
Mit der Statustabelle können Sie Variablen Ihres Programms lesen, schreiben oder forcen.

Die Statustabelle wird standardmäßig als Fenstersymbol am unteren Rand des Hauptfensters angezeigt (sofern Sie dies mit dem Menübefehl **Einrichten ► Voreinstellungen...** eingestellt haben). Wenn Sie die Statustabelle aufrufen möchten, doppelklicken Sie auf das Symbol bzw. Sie wählen den Befehl "Wiederherstellen/Vollbild" an der Schaltfläche (in Windows 95).

Bedienen und Beobachten von Variablen mit einer Statustabelle

Bild 3-26 zeigt ein Beispiel für eine Statustabelle. Zum Lesen oder Schreiben von Variablen in der Statustabelle gehen Sie folgendermaßen vor:

1. Geben Sie in dem ersten Feld in der Adreßspalte die Adresse oder den symbolischen Namen eines Elements aus Ihrem Programm ein, dessen Wert Sie lesen oder schreiben möchten. Drücken Sie dann die Eingabetaste. Wiederholen Sie diesen Schritt für alle Elemente, die Sie in die Tabelle aufnehmen möchten.
2. Handelt es sich bei dem Element um ein Bit (z. B. E, A oder M), wird in der zweiten Spalte Bitformat angezeigt. Handelt es sich bei dem Element um ein Byte, Wort oder Doppelwort, können Sie das Feld in der Formatspalte markieren und mit Doppelklick oder durch Drücken der Leertaste durch die gültigen Formate blättern.
3. Wenn Sie sich den aktuellen Wert des Elements in Ihrer Tabelle anzeigen lassen möchten, klicken Sie auf die Schaltfläche zum einfachen Lesen  oder auf die Schaltfläche zum ständigen Lesen  in der Statustabelle.
4. Möchten Sie die Aktualisierung beenden, klicken Sie auf die Schaltfläche zum Ständigen Lesen: .
5. Zum Ändern eines Werts geben Sie den neuen Wert in die Spalte "Neuer Wert" ein. Klicken Sie dann auf die Schaltfläche zum Schreiben , damit der Wert in die CPU geschrieben wird.





Adresse	Format	Aktueller Wert	Neuer Wert
"start_1"	Bit	2#0	1
E0.2	Bit	2#0	
"Leuchte_1"	Bit	2#0	
A1.2	Bit	Binär	
VB0	Mit Vorzeichen	+84	
VW2	Vorzeichenlos	4400	
VW4	Bit	2#0000001000110010	
VW6	Hexadezimal	16#0064	16#65
VD10	Gleitpunkt	0.0000	10.0
VD14	ASCII	'TEMP'	
VW20	Hexadezimal	16#0027	16#28
VW24	ASCII	'AB'	BA




Bild 3-26 Beispiel für eine Statustabelle

Forcen von Variablen in der Statustabelle

Wenn Sie eine Variable in der Statustabelle auf einen bestimmten Wert setzen möchten, gehen Sie folgendermaßen vor:

1. Geben Sie in dem ersten Feld in der Adreßspalte die Adresse oder den symbolischen Namen der Variable ein, dessen Wert Sie forcen möchten.
2. Handelt es sich bei dem Element um ein Bit (z. B. E, A oder M), kann das Format nicht geändert werden. Handelt es sich bei dem Element um ein Byte, Wort oder Doppelwort, können Sie das Feld in der Formatspalte markieren und mit Doppelklick oder durch Drücken der Leertaste durch die gültigen Formate blättern.
3. Wenn Sie die Variable mit dem aktuellen Wert forcen möchten, lesen Sie zunächst die aktuellen Werte aus dem Automatisierungssystem mit dem Menübefehl **Testen ▶ Einfaches Lesen** oder Sie klicken auf die Schaltfläche zum einfachen Lesen: .

Klicken Sie das Feld an bzw. navigieren Sie zu dem Feld, in dem sich der aktuelle Wert befindet, den Sie forcen möchten. Klicken Sie auf die Schaltfläche zum Forcen , während sich der Cursor auf dem aktuellen Wert befindet, mit dem Sie die Variable forcen möchten.

4. Zum Forcen einer Variablen mit einem neuen Wert, tragen Sie den gewünschten Wert in die Spalte "Neuer Wert" ein und klicken auf die Schaltfläche zum Forcen: .
5. Möchten Sie sich alle aktuell geforcten Variablen anzeigen lassen, klicken Sie auf die Schaltfläche zum Lesen der geforcten Werte: .
6. Möchten Sie alle geforcten Variablen der CPU entforcen, klicken Sie auf die Schaltfläche zum Entforcen: .

Bearbeiten von Adressen

Zum Bearbeiten eines Adreßfelds wählen Sie das gewünschte Feld mit den Cursortasten bzw. der Maus aus.

- Wenn Sie anfangen Daten einzugeben, werden vorhandene Daten gelöscht und die neuen Zeichen eingetragen.
- Wenn Sie mit der Maus doppelklicken oder die Taste F2 drücken, wird das Feld markiert. Sie können dann den Cursor mit den Cursortasten an die Stelle bewegen, die Sie bearbeiten möchten.

3.9 Arbeiten mit symbolischer Adressierung

Mit Hilfe der Symboltabelle können Sie Eingängen, Ausgängen und Adressen im internen Speicher symbolische Namen zuordnen (siehe Bild 3-27). Sie können die Symbole, die Sie diesen Adressen zugeordnet haben, im KOP-/AWL-Editor und in der Statustabelle in STEP 7-Micro/WIN verwenden. Der Datenbaustein-Editor unterstützt keine symbolischen Namen.

Richtlinien zum Eingeben von symbolischen Adressen

Die erste Spalte der Symboltabelle dient zum Markieren einer Reihe. Die anderen Spalten sind für den symbolischen Namen, die Adresse und den Kommentar vorgesehen. In jeder Reihe ordnen Sie einer Adresse eines digitalen Ein- oder Ausgangs, einer Adresse im Speicher, einem Sondermerker oder einem anderen Element einen symbolischen Namen zu. Der Kommentar zu einem symbolischen Namen ist optional. Beachten Sie die folgenden Richtlinien, wenn Sie eine Symboltabelle bearbeiten.

- Sie können symbolische Namen und absolute Adressen in beliebiger Reihenfolge eingeben.
- Sie können symbolische Namen mit maximal 23 Zeichen eingeben.
- Sie können maximal 1000 Symbole definieren.
- Die Symboltabelle beachtet Groß- und Kleinschreibung: "Pumpe1" beispielsweise ist ein anderer symbolischer Name als "pumpe1".
- Leerzeichen vor und nach einem symbolischen Namen werden vom Symbol-Editor automatisch gelöscht. Leerzeichen in einem symbolischen Namen durch einen Unterstrich ersetzt. Beispiel: "Starter Motor 2" wird zu "Starter_Motor_2".
- Doppelte symbolische Namen und/oder Adressen werden durch blaue Kursivschrift gekennzeichnet, werden nicht übersetzt und können im Programm nicht verwendet werden. Sich überschneidende Adressen werden nicht als doppelte Adressen gekennzeichnet; VBO und VW0 beispielsweise überschneiden sich im Speicher, werden aber nicht als doppelte Adressen gekennzeichnet.

Aufrufen des Symbol-Editors

Der Symbol-Editor wird standardmäßig als Fenstersymbol am unteren Rand des Hauptfensters dargestellt. Wenn Sie die Symboltabelle aufrufen möchten, doppelklicken Sie auf das Symbol bzw. Sie wählen den Befehl "Wiederherstellen/Vollbild" an der Schaltfläche (in Windows 95).

	Symbolischer Name	Adresse	Kommentar
	Start1	E0.0	Fertigungsstraße 1
	NOT_AUS1	E0.1	Fertigungsstraße 1
	Leuchte_1	A1.0	1, grüne Leuchte
	StarterMotor1	A1.1	Fertigungsstraße 1, Motor
	Mixer1_Zeit	T0	
	Mixer2_Zeit	T37	
	Zähler1	Z1	
	Relais_1	M0.0	
	Relais_1	M0.1	

Bild 3-27 Beispiel für eine Symboltabelle

Bearbeitungsfunktionen innerhalb der Symboltabelle

Die Symboltabelle verfügt über die folgenden Bearbeitungsfunktionen:

- **Bearbeiten ▶ Ausschneiden/Kopieren/Einfügen:** innerhalb eines Felds oder zwischen verschiedenen Feldern.
- **Bearbeiten ▶ Ausschneiden/Kopieren/Einfügen:** von mehreren zusammenhängenden Reihen.
- **Bearbeiten ▶ Reihe einfügen:** oberhalb der Reihe, in der sich der Cursor befindet. Sie können hierzu auch die Taste EINF verwenden.
- **Bearbeiten ▶ Reihe löschen:** eine oder mehrere benachbarte markierte Reihen. Sie können hierzu auch die Taste DEL verwenden.
- Zum Bearbeiten eines beliebigen Datenfelds wählen Sie das gewünschte Feld mit den Cursorstasten bzw. der Maus aus. Wenn Sie anfangen Daten einzugeben, werden vorhandene Daten gelöscht und die neuen Zeichen eingetragen. Wenn Sie mit der Maus doppelklicken oder die Taste **F2** drücken, wird das Feld markiert. Sie können dann den Cursor mit den Cursorstasten an die Stelle bewegen, die Sie bearbeiten möchten.

Sortieren von Tabelleneinträgen

Nachdem Sie die symbolischen Namen und die zugeordneten absoluten Adressen eingegeben haben, können Sie die Symboltabelle alphabetisch nach den symbolischen Namen oder numerisch nach den Adressen sortieren. Hierzu gehen Sie folgendermaßen vor:

- Wählen Sie den Menübefehl **Ansicht ▶ Sortierung – Name**, um die symbolischen Namen in alphabetischer Reihenfolge zu sortieren.
- Wenn Sie die symbolischen Adressen in der Symboltabelle nach den absoluten Adressen sortieren möchten, wählen Sie den Menübefehl **Ansicht ▶ Sortieren – Adresse**.

Eingeben eines Programmierbeispiels

4

Die Beispiele und Beschreibungen in diesem Handbuch beziehen sich auf die Version 2.1 der Programmiersoftware STEP 7-Micro/WIN. Vorherige Versionen der Software weisen zum Teil andere Funktionalität auf.

Dieses Kapitel erläutert Ihnen, wie Sie mit der Software STEP 7-Micro/WIN die folgenden Aufgaben ausführen:

- Eingeben eines Programmierbeispiels für einen Mischbehälter mit zwei Speisepumpen
- Anlegen einer Symboltabelle, einer Statustabelle und eines Datenbausteins
- Überwachen des Programms

STEP 7-Micro/WIN bietet Ihnen umfassende Online-Hilfe. Wählen Sie einen der Befehle im Menü **Hilfe** oder drücken Sie F1, um Hilfe zu aktuellen Fragestellungen zu erhalten.

Kapitelübersicht

Abschnitt	Beschreibung	Seite
4.1	Erstellen eines Programms für eine Beispielanwendung	4-2
4.2	Anlegen eines Projekts	4-6
4.3	Erstellen einer Symboltabelle	4-8
4.4	Eingeben des Programms im Kontaktplan	4-10
4.5	Erstellen einer Statustabelle	4-14
4.6	Laden und Überwachen des Programmierbeispiels	4-15

4.1 Erstellen eines Programms für eine Beispielanwendung

Voraussetzungen zum Ausführen des Programmierbeispiels

Wenn Sie das Programmierbeispiel in diesem Kapitel vollständig erstellen und in die CPU laden, können Sie das Programm mit Ihrer S7-200 CPU bearbeiten. Bild 4-1 zeigt die Komponenten, die zum Ausführen und Überwachen des Programmierbeispiels erforderlich sind:

- PC/PPI-Kabel bzw. MPI-Baugruppe und RS-485-Kabel zum Anschließen Ihres Computers an die S7-200 CPU
- S7-200 CPU
- Eingangssimulator
- Anschlußkabel und Spannungsversorgung
- STEP 7-Micro/WIN 32 Version 2.1 für Windows 95 und Windows NT (32 Bit) oder STEP 7-Micro/WIN 16 Version 2.1 für Windows 3.1x (16 Bit)

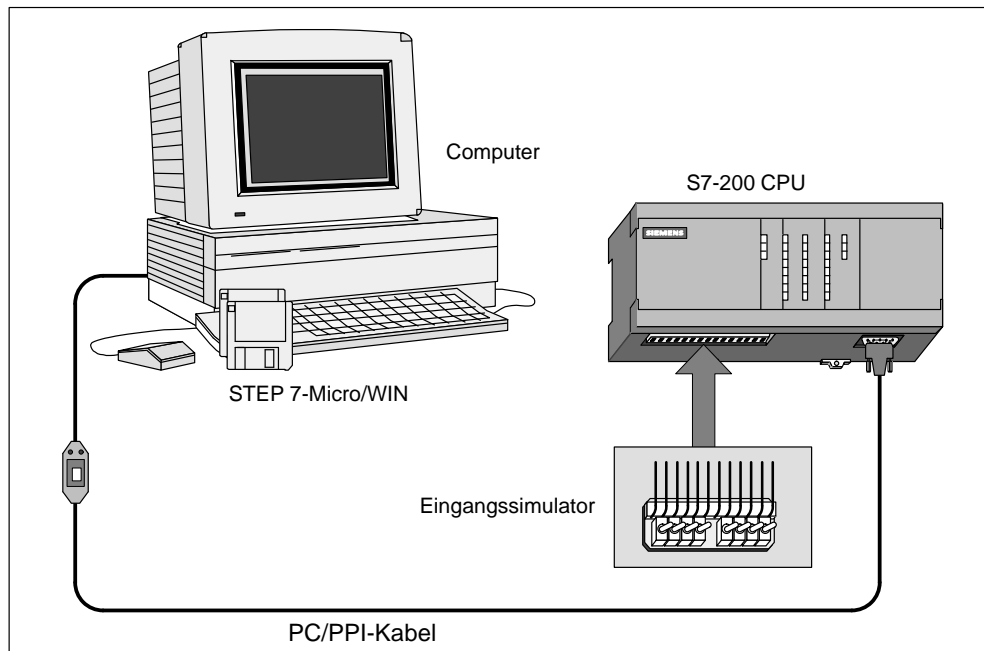


Bild 4-1 Voraussetzungen zum Ausführen des Programmierbeispiels

Beispielanwendung "Mischbehälter"

Bild 4-2 zeigt das Schema für einen Mischbehälter. Der Mischbehälter kann für verschiedene Anwendungen eingesetzt werden, z. B. zum Mischen von Farben. In dieser Anwendung reichen zwei Speiserohre, von denen jedes verschiedene Stoffe zuführt, von oben in den Behälter hinein. Durch ein einzelnes Rohr am unteren Teil des Behälters läuft die fertige Farbmischung ab. Mit dem Programmierbeispiel steuern Sie den Füllvorgang, überwachen den Füllstand und steuern den Heiz- und Mischzyklus. Dies geschieht in folgenden Schritten:

- Schritt 1: Behälter mit Farbe 1 füllen.
- Schritt 2: Behälter mit Farbe 2 füllen.
- Schritt 3: Zufluß beenden, wenn Schalter für "Behälter voll" geschlossen.
- Schritt 4: Pumpe eingeschaltet lassen, wenn Startschalter offen.
- Schritt 5: Heiz- und Mischzyklus beginnen.
- Schritt 6: Rührmotor und Dampfventil einschalten.
- Schritt 7: Farbmischung ablaufen lassen.
- Schritt 8: Jeden Zyklus zählen.

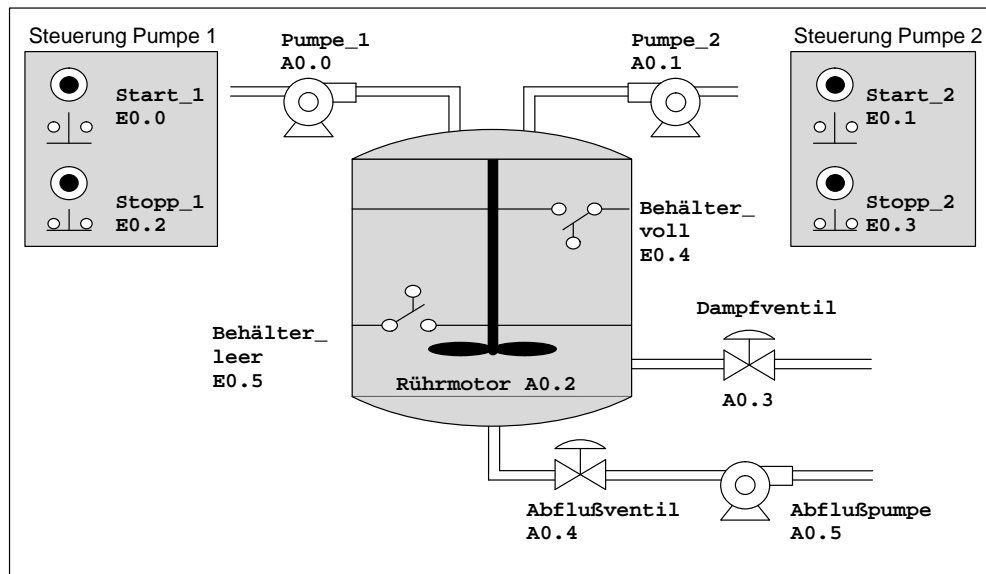


Bild 4-2 Programmierbeispiel: Mischbehälter

Programmierbeispiel in der Anweisungsliste (AWL) und im Kontaktplan (KOP)

Sie können das Programmierbeispiel sowohl in der Anweisungsliste (AWL) als auch im Kontaktplan (KOP) eingeben. Tabelle 4-1 zeigt das Programmierbeispiel in der Programmiersprache Anweisungsliste und Bild 4-3 zeigt das Beispiel in der Programmiersprache Kontaktplan. Die Abschnitte 4.2 bis 4.4 führen Sie schrittweise durch die erforderlichen Aufgaben zum Eingeben des Programms.

Tabelle 4-1 Programmierbeispiel in der Anweisungsliste

AWL	Beschreibung
NETWORK 1	//Behälter mit Farbe 1 füllen.
LD "Start_1"	
O "Pumpe_1"	
U "Stopp_1"	
UN "Behälter_voll"	
= "Pumpe_1"	
NETWORK 2	//Behälter mit Farbe 2 füllen.
LD "Start_2"	
O "Pumpe_2"	
U "Stopp_2"	
UN "Behälter_voll"	
= "Pumpe_2"	
NETWORK 3	//Merker setzen, wenn der maximale //Füllstand erreicht ist.
LD "Behälter_voll"	
S "Max_Füllstand", 1	
NETWORK 4	//Zeit starten, wenn der maximale //Füllstand erreicht ist.
LD "Max_Füllstand"	
TON "Mischzeit", +100	
NETWORK 5	//Rührmotor einschalten.
LDN "Mischzeit"	
U "Max_Füllstand"	
= "Rührmotor"	
= "Dampfventil"	
NETWORK 6	//Farbmischung ablaufen lassen.
LD "Mischzeit"	
UN "Behälter_leer"	
= "Abflußventil"	
= "Abflußpumpe"	
NETWORK 7	//Jeden Zyklus zählen.
LD "Behälter_leer"	
U "Mischzeit"	
LD "Rücksetzen"	
ZV "Zykluszähler", +12	
NETWORK 8	//Merker rücksetzen, wenn der minimale //Füllstand erreicht oder die Zeit //abgelaufen ist.
LD "Behälter_leer"	
U "Mischzeit"	
R "Max_Füllstand", 1	
NETWORK 9	//Hauptprogramm beenden.
MEND	

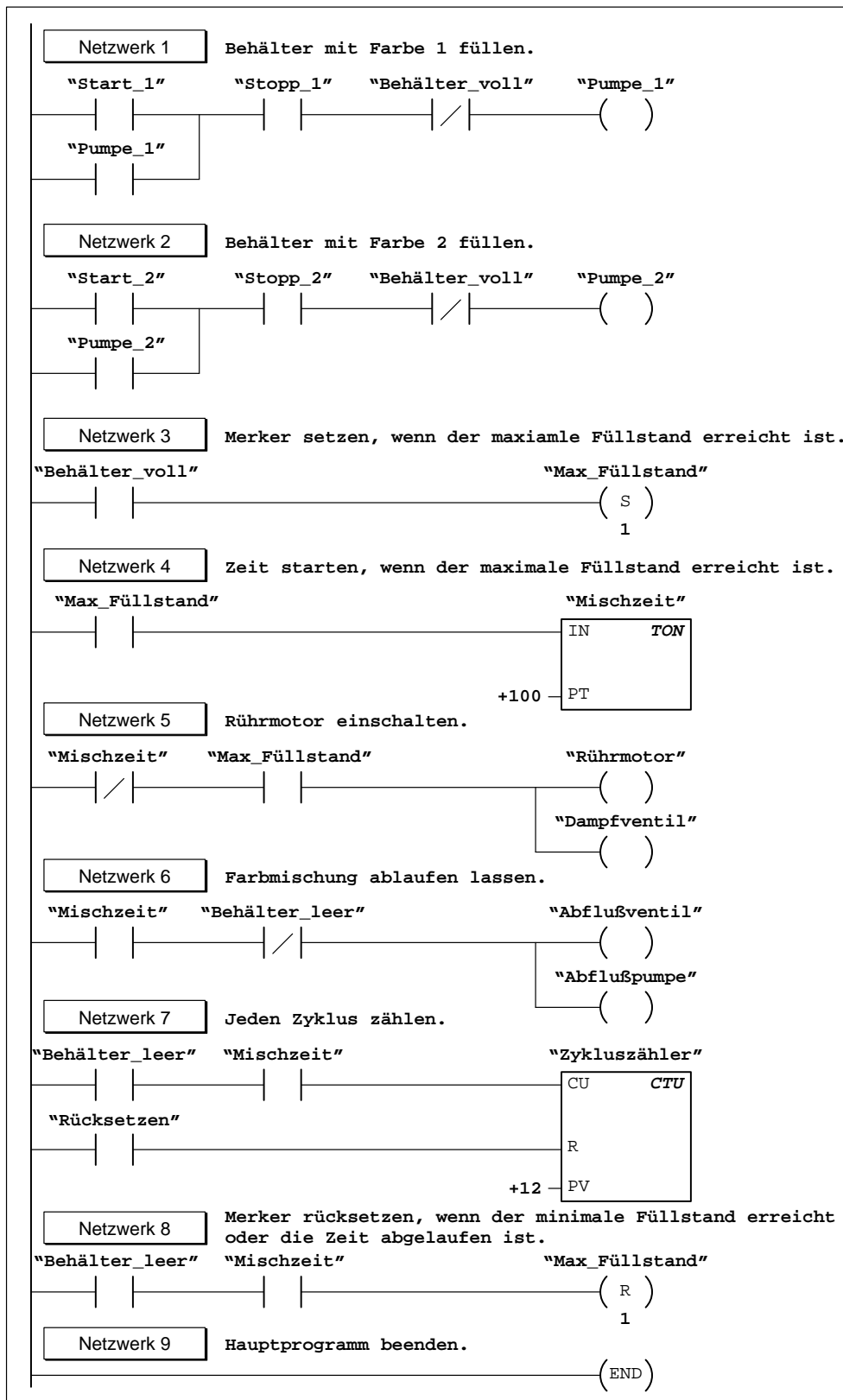



Bild 4-3 Programmierbeispiel im Kontaktplan

4.2 Anlegen eines Projekts

Anlegen eines neuen Projekts

Wenn Sie ein Projekt anlegen oder öffnen, ruft STEP 7-Micro/WIN folgende Editoren auf: je nach Voreinstellung den AWL- oder KOP-Editor (OB1), den Datenbaustein-Editor (DB1), die Statustabelle und die Symboltabelle.

Zum Anlegen eines neuen Projekts wählen Sie den Menübefehl **Projekt ► Neu...** (siehe Bild 4-4) oder Sie klicken auf die Schaltfläche zum Anlegen eines neuen Projekts in der Funktionsleiste .

Daraufhin öffnet sich das Dialogfeld "CPUs". Wählen Sie in dem aufklappbaren Listenfeld Ihre CPU aus.

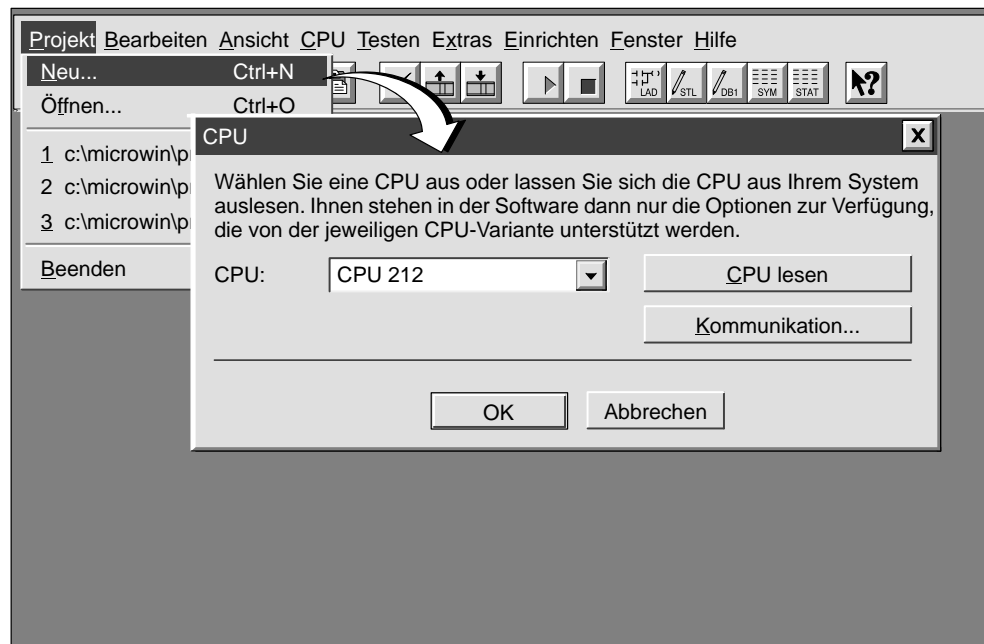


Bild 4-4 Anlegen eines neuen Projekts und Auswählen der CPU

Benennen des Programmierbeispiels

Sie können Ihr Projekt zu einem beliebigen Zeitpunkt benennen. Gehen Sie in diesem Beispiel folgendermaßen vor, um Ihr Projekt unter einem bestimmten Namen zu speichern (siehe Bild 4-5):

1. Wählen Sie den Menübefehl **Projekt ► Speichern unter...**
2. In dem Feld "Dateiname" geben Sie folgenden Namen ein: `projekt1.prj`
3. Klicken Sie auf "OK".

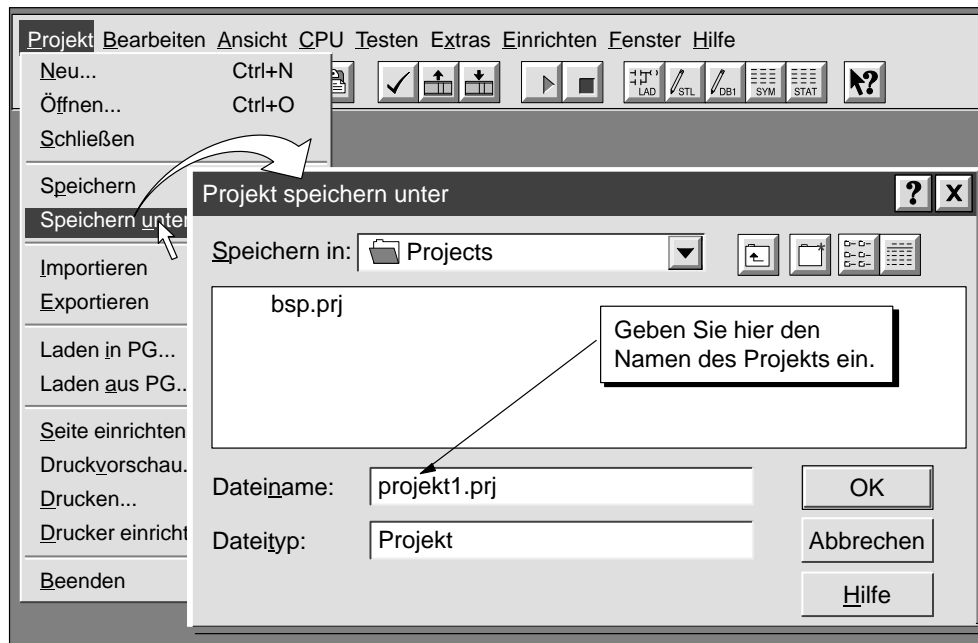


Bild 4-5 Benennen des Programmierbeispiels

4.3 Erstellen einer Symboltabelle

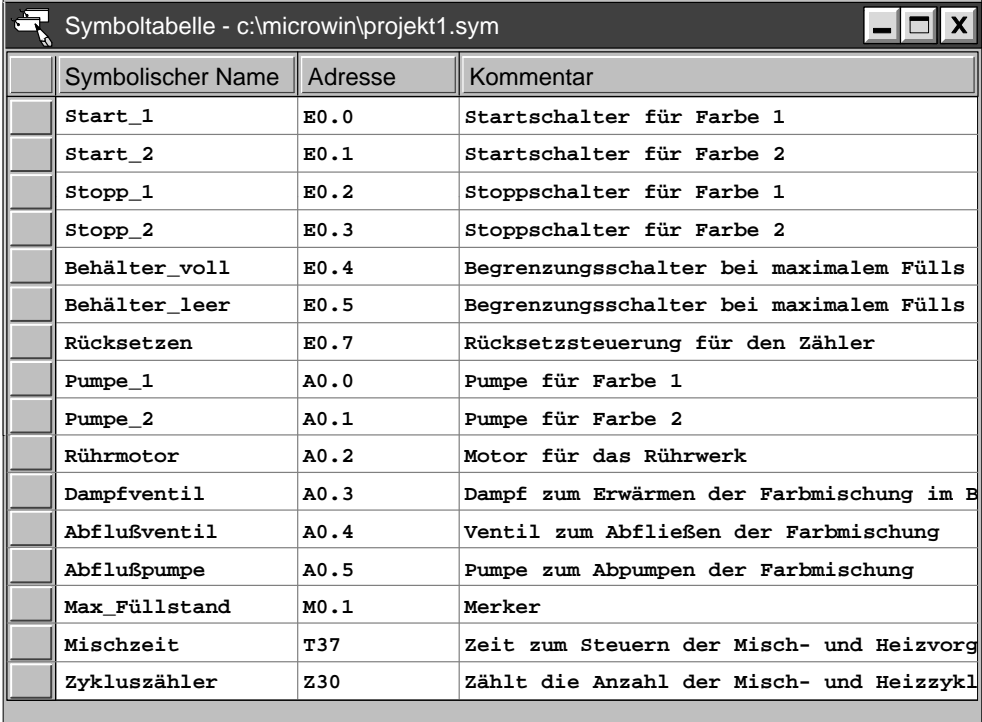
Aufrufen des Symboltabellen-Editors

Wenn Sie den absoluten Adressen des Programmierbeispiels symbolische Namen zuordnen möchten, öffnen Sie den Symboltabellen-Editor. Hierzu doppelklicken Sie auf das Symbol des Editors bzw. Sie wählen den Befehl "Wiederherstellen/Vollbild" an der Schaltfläche (in Windows 95). Sie können auch den Menübefehl **Ansicht ► Symboltabelle** wählen.

Eingeben von symbolischen Namen

Bild 4-6 zeigt die Liste der absoluten Adressen und die zugeordneten symbolischen Namen für das Programmierbeispiel. Zum Eingeben der symbolischen Namen gehen Sie folgendermaßen vor:

1. Markieren Sie das erste Feld in der Spalte "Symbolischer Name" und geben Sie **start_1 ein.**
2. Drücken Sie die Eingabetaste, um so das erste Feld in der Spalte "Adresse" zu markieren. Geben Sie die Adresse **E0.0** ein und drücken Sie die Eingabetaste. Daraufhin wird das erste Feld in der Spalte "Kommentar" markiert. (Kommentare sind optional, doch können sie sehr nützlich sein, um die einzelnen Elemente des Programms zu dokumentieren.)
3. Drücken Sie die Eingabetaste, um mit Ihren Eingaben in der nächsten Reihe zu beginnen. Wiederholen Sie die oben aufgeführten Schritte, bis Sie alle übrigen Adressen und symbolischen Namen eingegeben haben.
4. Speichern Sie die Symboltabelle mit dem Menübefehl **Projekt ► Speichern.**



Symbolischer Name	Adresse	Kommentar
Start_1	E0.0	Startschalter für Farbe 1
Start_2	E0.1	Startschalter für Farbe 2
Stopp_1	E0.2	Stoppschalter für Farbe 1
Stopp_2	E0.3	Stoppschalter für Farbe 2
Behälter_voll	E0.4	Begrenzungsschalter bei maximalem Fülls
Behälter_leer	E0.5	Begrenzungsschalter bei maximalem Fülls
Rücksetzen	E0.7	Rücksetzsteuerung für den Zähler
Pumpe_1	A0.0	Pumpe für Farbe 1
Pumpe_2	A0.1	Pumpe für Farbe 2
Rührmotor	A0.2	Motor für das Rührwerk
Dampfventil	A0.3	Dampf zum Erwärmen der Farbmischung im B
Abflußventil	A0.4	Ventil zum Abfließen der Farbmischung
Abflußpumpe	A0.5	Pumpe zum Abpumpen der Farbmischung
Max_Füllstand	M0.1	Merker
Mischzeit	T37	Zeit zum Steuern der Misch- und Heizvorg
Zykluszähler	Z30	Zählt die Anzahl der Misch- und Heizzykl

Bild 4-6 Symboltabelle für das Programmierbeispiel

Programmieren mit symbolischen Adressen

Bevor Sie Ihr Programm eingeben, stellen Sie die symbolische Adressierung ein. Hierzu wählen Sie den Menübefehl **Ansicht ► Symbolische Adressierung**. Befindet sich links neben dem Menübefehl ein Häkchen, ist die symbolische Adressierung aktiviert.

Hinweis

Bei symbolischen Namen müssen Sie auf korrekte Groß- und Kleinschreibung achten. Der Name, den Sie eingeben, muß in Groß- und Kleinbuchstaben genau dem Namen in der Symboltabelle entsprechen. Bei Abweichungen bleibt der Cursor auf dem Element, und die Meldung "Ungültiger Parameter" wird in der Statuszeile am unteren Rand der Bedienoberfläche angezeigt.

4.4 Eingeben des Programms im Kontaktplan

Aufrufen des KOP-Editors

Zum Aufrufen des KOP-Editors doppelklicken Sie auf dessen Schaltfläche am unteren Rand der Bedienoberfläche. Bild 4-7 zeigt einige der wesentlichen Werkzeuge, mit denen Sie im KOP-Editor arbeiten.

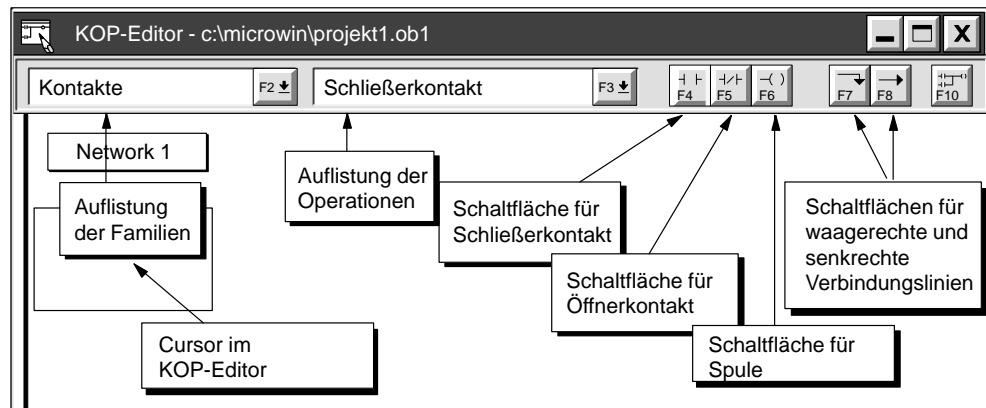


Bild 4-7 Grundlegende Werkzeuge im KOP-Editor

Funktionsleiste der Operationen im KOP-Editor

Mit dem Menübefehl **Ansicht ► Funktionsleiste der Operationen** zeigen Sie im KOP-Editor die Funktionsleiste der Operationen an (siehe Bild 4-8).

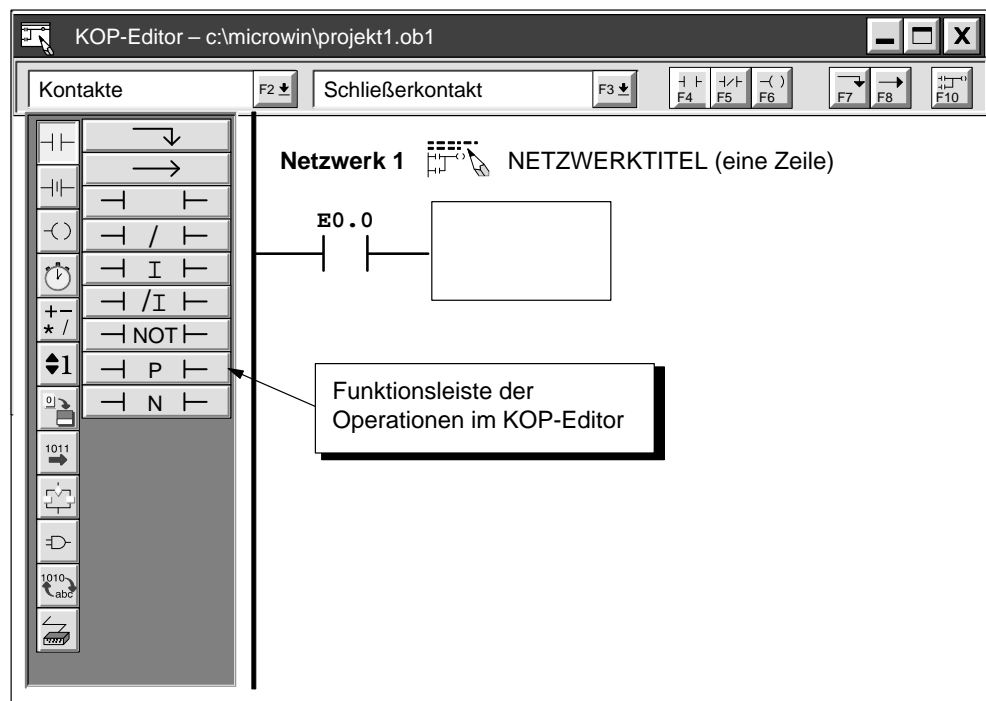


Bild 4-8 Grundlegende Werkzeuge im KOP-Editor

Eingeben des ersten Elements des Netzwerks

Gehen Sie folgendermaßen vor, um das erste Netzwerk für das Programmierbeispiel einzugeben:

1. Doppelklicken Sie auf oder neben das numerierte Schlüsselwort Netzwerk, um den Kommentar-Editor aufzurufen. Geben Sie den in Bild 4-9 gezeigten Kommentar ein und bestätigen Sie mit "OK".
2. Drücken Sie die Richtungstaste nach unten. Daraufhin wird der KOP-Cursor in der folgenden Zeile auf das erste Feld links positioniert.
3. Wählen Sie einen Schließerkontakt aus, indem Sie zunächst die Familie "Kontakte" und anschließend im Listenfeld der Operationen "Schließerkontakt" markieren.
4. Drücken Sie die Eingabetaste. Daraufhin wird ein Schließerkontakt mit der symbolischen Adresse "Start_1" eingefügt. (Die Software gibt jedesmal, wenn Sie einen Kontakt eingeben, als Voreinstellung die Adresse $\text{I}0.0$ an. $\text{I}0.0$ ist in diesem Beispiel dem symbolischen Namen "Start_1" zugeordnet.)
5. "Start_1" ist das erste Element in Netzwerk 1. Bestätigen Sie die Eingabe dieses ersten Elements und des symbolischen Namens, indem Sie die Eingabetaste drücken. Der KOP-Cursor springt daraufhin in das zweite Feld in dieser Reihe.

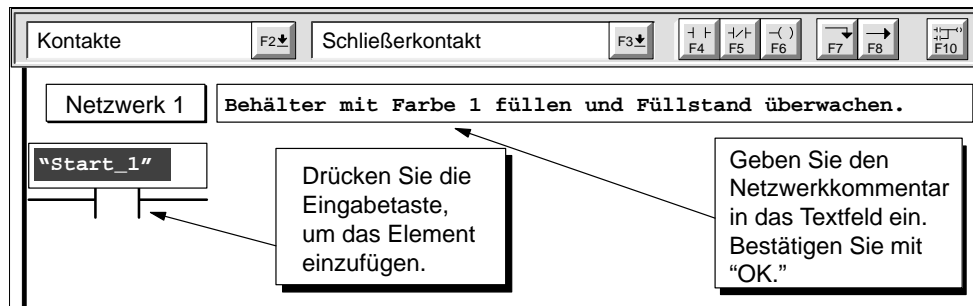


Bild 4-9 Eingeben des Netzwerkkommentars und des ersten Elements in KOP

Zum Eingeben der weiteren Kontakte des ersten Netzwerks gehen Sie folgendermaßen vor:

1. Drücken Sie die Eingabetaste, um das zweite Element einzugeben. Daraufhin wird ein Schließerkontakt mit dem voreingestellten symbolischen Namen "Start_1" angezeigt.
2. Geben Sie **Pumpe_1** ein und drücken Sie die Eingabetaste. Der Cursor springt daraufhin zum nächsten Feld.
3. Klicken Sie auf die Schaltfläche für "Öffnerkontakt" ("F5"). Daraufhin wird ein Öffnerkontakt auf dem Bildschirm angezeigt. Der symbolische Name "Start_1" oberhalb des Kontakts ist hervorgehoben.
4. Geben Sie **Behälter_voll** ein und drücken Sie die Eingabetaste.

Ihr KOP-Netzwerk sollte so aussehen, wie in Bild 4-10 gezeigt.

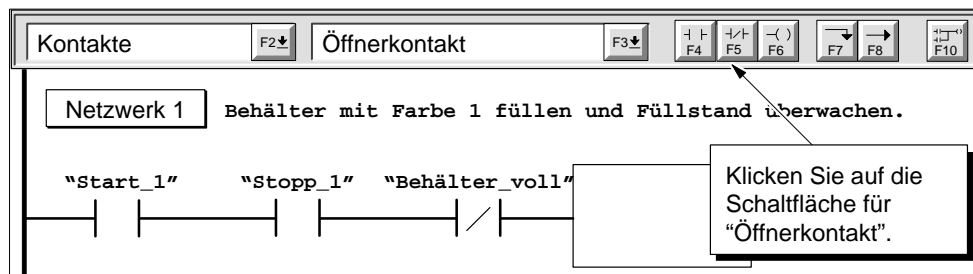


Bild 4-10 Eingeben der nächsten KOP-Elemente

Der KOP-Cursor befindet sich jetzt rechts von dem Öffnerkontakt "Behälter_voll". Zum Vervollständigen des Netzwerks gehen Sie folgendermaßen vor (siehe Bild 4-11).

1. Klicken Sie auf die Schaltfläche für "Spule" ("F6") und drücken Sie die Eingabetaste bzw. positionieren Sie den Mauszeiger innerhalb des KOP-Cursors und doppelklicken Sie. Daraufhin wird eine Spule mit dem Namen "Pumpe_1" angezeigt. (Jedesmal, wenn Sie eine Spule eingeben, wird als Voreinstellung die Adresse A0.0 angezeigt. A.0.0 ist in diesem Programmierbeispiel dem symbolischen Namen "Pumpe_1" zugeordnet.)
2. Bestätigen Sie die Spule und den symbolischen Namen, indem Sie die Eingabetaste drücken.
3. Positionieren Sie den Cursor wieder auf das erste Element in diesem Netzwerk. Verwenden Sie hierzu die Maus oder die Richtungstaste nach links.
4. Klicken Sie auf die Schaltfläche für "Senkrechte Verbindungslinie" ("F7") und drücken Sie die Eingabetaste. Daraufhin wird zwischen dem ersten und dem zweiten Kontakt eine senkrechte Verbindungslinie gezogen.
5. Klicken Sie auf die Schaltfläche für "Schließerkontakt" ("F4") in der Symbolleiste und drücken Sie anschließend die Eingabetaste. Daraufhin wird ein Kontakt mit dem Namen "Start_1" angezeigt.
6. Geben Sie **Pumpe_1** ein und drücken Sie die Eingabetaste.

Das erste Netzwerk ist damit vollständig eingegeben.

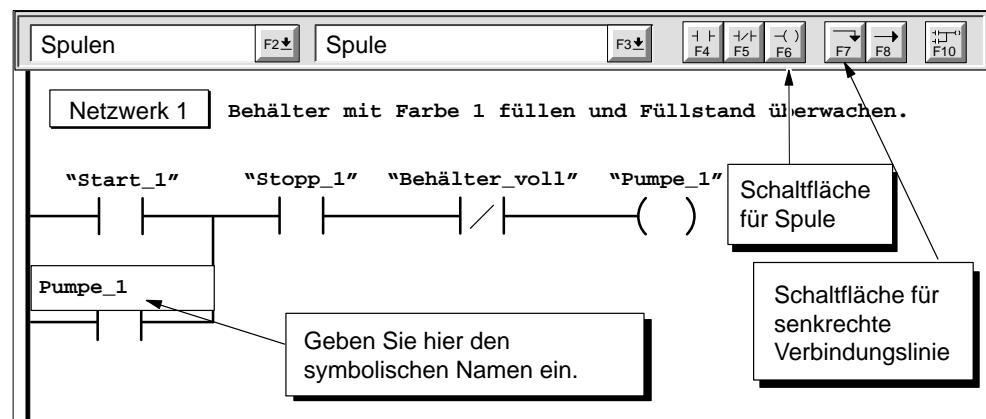


Bild 4-11 Vervollständigen des ersten Netzwerks

Eingeben des zweiten Netzwerks

Zum Eingeben des zweiten Netzwerks für das Programmierbeispiel gehen Sie folgendermaßen vor:

1. Positionieren Sie den Cursor auf dem zweiten Netzwerk. Verwenden Sie hierzu die Maus oder die Richtungstaste nach unten.
2. Geben Sie den in Bild 4-12 gezeigten Kommentar in das Kommentarfeld des Netzwerks ein. (Da der Kommentar für Netzwerk 2 ähnlich dem Kommentar von Netzwerk 1 ist, können Sie den Text in Netzwerk 1 auch markieren, kopieren und in das Kommentarfeld von Netzwerk 2 einfügen. Sie müssen dann nur die Nummer der Farbe von 1 in 2 ändern.)
3. Wiederholen Sie die Schritte, die Sie zum Eingeben von Netzwerk 1 ausgeführt haben. Verwenden Sie hierzu die symbolischen Namen, die in Bild 4-12 dargestellt sind.
4. Nachdem Sie Netzwerk 2 eingegeben haben, positionieren Sie den Cursor auf Netzwerk 3.

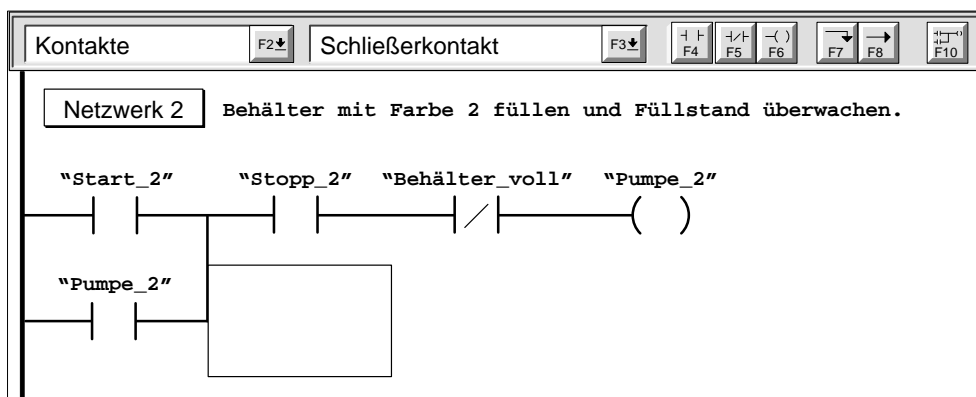



Bild 4-12 Eingeben des zweiten Netzwerks

Eingeben der übrigen Netzwerke


Zum Eingeben der übrigen Netzwerke gehen Sie genauso vor, wie für die vorherigen Netzwerke beschrieben. Die übrigen Netzwerke sind in Bild 4-3 dargestellt.

Übersetzen des Programms

Nach dem Eingeben des Programms prüfen Sie die Syntax, indem Sie den Menübefehl **CPU ▶ Übersetzen** wählen oder auf die Schaltfläche zum Übersetzen klicken .

Wenn Sie die Netzwerke korrekt eingegeben haben, wird die Meldung "Übersetzung erfolgreich" angezeigt. Diese Meldung zeigt außerdem die Anzahl der Netzwerke und den vom Programm belegten Speicherplatz an. Ist Ihnen beim Eingeben ein Fehler unterlaufen, zeigt Ihnen die Meldung an, in welchem Netzwerk der Fehler enthalten ist.

Speichern des Programmierbeispiels

Sie speichern Ihr Projekt mit dem Menübefehl **Projekt ▶ Speichern** oder indem Sie auf die Schaltfläche zum Speichern klicken: . Es werden alle Komponenten Ihres Beispielprojekts gespeichert.

4.5 Erstellen einer Statustabelle

Anlegen einer Statustabelle


Zum Überwachen des Status von bestimmten Elementen in Ihrem Programmierbeispiel müssen Sie eine Statustabelle erstellen, in der die Elemente enthalten sind, die Sie während der Programmbearbeitung überwachen möchten. Zum Öffnen des Tabellen-Editors doppelklicken Sie auf dessen Schaltfläche am unteren Rand der Bedienoberfläche. Gehen Sie dann folgendermaßen vor, um die Elemente für Ihr Programmierbeispiel einzugeben:

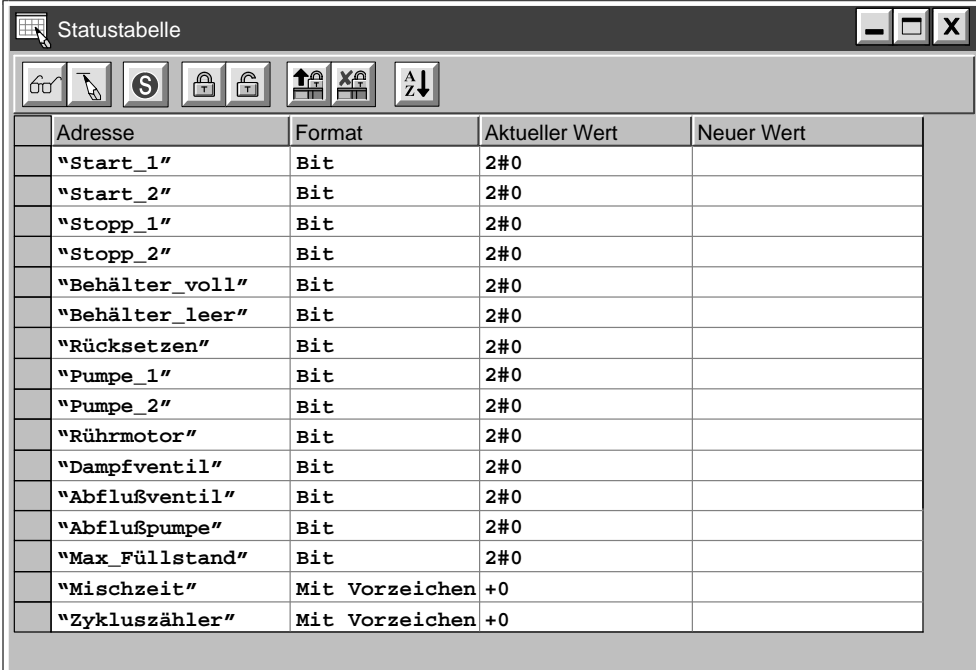
1. Markieren Sie das erste Feld in der Spalte "Adresse" und geben Sie `start_1` ein.
2. Bestätigen Sie Ihre Eingabe mit der Eingabetaste. Dieses Element kann nur im Binärformat (Bit) angezeigt werden ("1" oder "0"), deshalb können Sie das Format nicht ändern.
3. Markieren Sie die nächste Reihe. Wiederholen Sie die oben beschriebenen Schritte für die übrigen Elemente (siehe Bild 4-13).

Haben Sie ein Adreßfeld angewählt und die Reihe unterhalb des Felds ist leer, dann können Sie durch Drücken der Eingabetaste automatisch die Adresse in jeder weiteren Reihe um 1 erhöhen. Ausführliche Informationen zum Arbeiten mit der Statustabelle entnehmen Sie der Online-Hilfe.

Mit dem Menübefehl **Bearbeiten ► Reihe einfügen** (bzw. mit der Taste `EINFG`) fügen Sie eine neue Reihe ein, und zwar jeweils über der Reihe, in der sich der Cursor befindet.

4. Die Zeit T37 und der Zähler Z30 können jeweils in anderen Formaten angezeigt werden. Haben Sie das Feld in der Spalte "Format" markiert, können Sie mit der Leertaste nacheinander alle Formate aufrufen, die für das jeweilige Element gültig sind. Wählen Sie in diesem Beispiel für die Zeit und den Zähler das Format "Mit Vorzeichen" aus.

Sie speichern Ihre Statustabelle mit dem Menübefehl **Projekt ► Speichern** oder indem Sie auf die Schaltfläche zum Speichern klicken .



Adresse	Format	Aktueller Wert	Neuer Wert
"start_1"	Bit	2#0	
"start_2"	Bit	2#0	
"Stopp_1"	Bit	2#0	
"Stopp_2"	Bit	2#0	
"Behälter_voll"	Bit	2#0	
"Behälter_leer"	Bit	2#0	
"Rücksetzen"	Bit	2#0	
"Pumpe_1"	Bit	2#0	
"Pumpe_2"	Bit	2#0	
"Rührmotor"	Bit	2#0	
"Dampfventil"	Bit	2#0	
"Abflußventil"	Bit	2#0	
"Abflußpumpe"	Bit	2#0	
"Max_Füllstand"	Bit	2#0	
"Mischzeit"	Mit Vorzeichen	+0	
"Zykluszähler"	Mit Vorzeichen	+0	



Bild 4-13 Statustabelle für das Programmierbeispiel

4.6 Laden und Überwachen des Programmierbeispiels

Im nächsten Schritt müssen Sie Ihr Programm in die CPU laden und anschließend die CPU in den Betriebszustand RUN versetzen. Dann können Sie Ihr Programm testen und die Bearbeitung überwachen.

Laden des Projekts in die CPU

Die CPU muß sich im Betriebszustand STOP befinden, damit Sie das Programm in die CPU laden können. Gehen Sie folgendermaßen vor, um die CPU in den Betriebszustand STOP zu versetzen und das Programm zu laden:

1. Bringen Sie den Betriebsartenschalter der CPU (der sich unterhalb der Abdeckklappe der CPU befindet) in die Position TERM oder STOP.
2. Wählen Sie den Menübefehl **CPU ► STOP** oder klicken Sie auf die Schaltfläche für den Betriebszustand STOP  im Hauptfenster.
3. Bestätigen Sie den Vorgang mit "Ja".
4. Wählen Sie den Menübefehl **Projekt ► Laden aus PG...** oder klicken Sie auf die Schaltfläche zum Laden aus PG im Hauptfenster .
5. In dem Dialogfeld "Laden aus PG" können Sie angeben, welche Komponenten Sie in die CPU laden möchten. Drücken Sie anschließend die Eingabetaste oder klicken Sie auf die Schaltfläche "OK".

Eine Meldung zeigt Ihnen an, ob der Ladevorgang erfolgreich ausgeführt wurde.


Hinweis

STEP 7-Micro/WIN prüft nicht, ob die Operanden und Adressen von Ein- und Ausgängen in Ihrem Programm für die jeweilige CPU gültig sind. Wenn Sie versuchen, ein Programm in die CPU zu laden, dessen Adressen außerhalb des gültigen Bereichs für die CPU liegen oder dessen Operationen von der CPU nicht unterstützt werden, dann kann das Programm nicht in die CPU geladen werden. Es wird Ihnen eine Fehlermeldung angezeigt.

Achten Sie darauf, daß alle Operanden und Adressen von Ein- und Ausgängen sowie die Operationen in Ihrem Programm für die CPU, mit der Sie arbeiten, gültig sind.

Versetzen der CPU in den Betriebszustand RUN

Wurde das Programm erfolgreich in die CPU geladen, können Sie anschließend die CPU in den Betriebszustand RUN versetzen.

1. Wählen Sie den Menübefehl **CPU ► RUN** oder klicken Sie auf die Schaltfläche für den Betriebszustand RUN  im Hauptfenster.
2. Bestätigen Sie den Vorgang mit "Ja".

Überwachen des Status in KOP

Haben Sie den KOP-Status eingeschaltet, wird der aktuelle Zustand der Ereignisse in Ihrem Programm angezeigt. Öffnen Sie das Fenster für den KOP-Editor und wählen Sie den Menübefehl **Testen ▶ Status ein**.

Sofern Sie an die Eingangsklemmen Ihrer CPU einen Eingangssimulator angeschlossen haben, können Sie verschiedene Schalter einschalten und den Signalfluß sowie die Bearbeitung des Programms verfolgen. Schalten Sie beispielsweise die Eingänge **Æ0.0** und **Æ0.2** ein und der Schalter des Eingangs **Æ0.4** ("Behälter_voll") ist ausgeschaltet, dann ist der Signalfluß für das erste Netzwerk vollständig. Das Netzwerk wird dann wie in Bild 4-14 angezeigt.

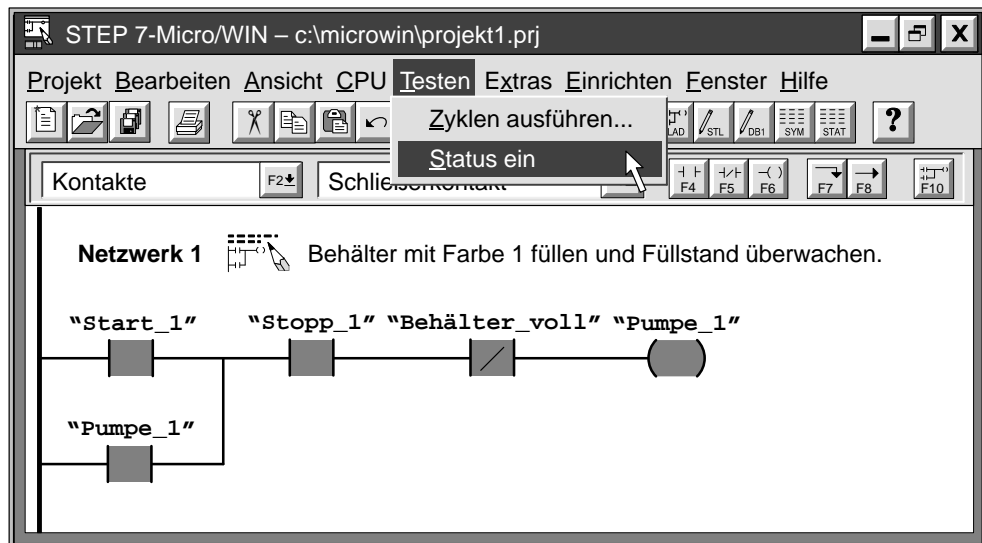


Bild 4-14 Überwachen des Status des ersten Netzwerks

Entspricht das Programm in STEP 7-Micro/WIN nicht dem Programm, das sich in der CPU befindet, werden Sie durch die in Bild 4-15 dargestellte Meldung darauf hingewiesen. Sie werden dann gefragt, ob Sie das Programm mit dem Programm in der CPU vergleichen möchten, ob Sie fortfahren oder abbrechen möchten.

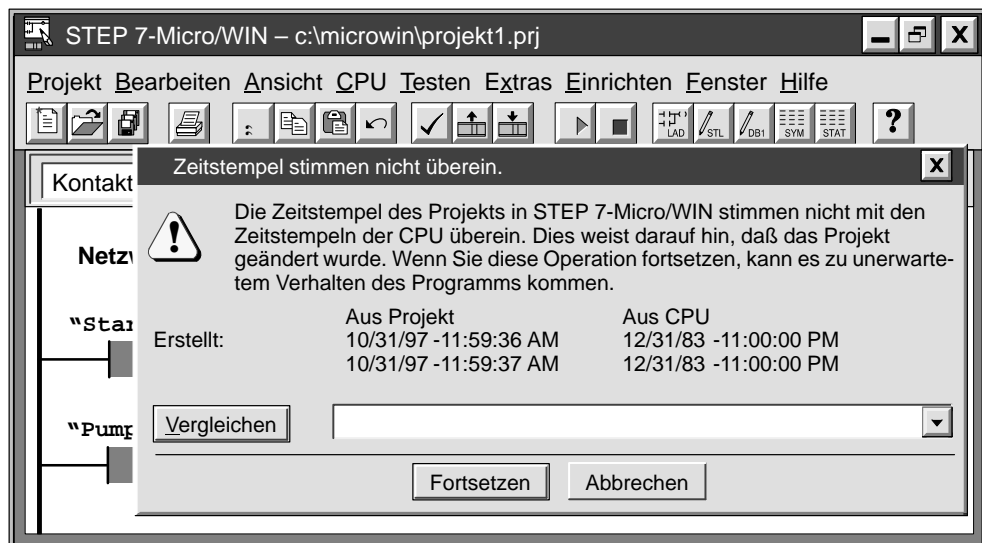





Bild 4-15 Warnung über unterschiedliche Zeitstempel

Anzeigen des aktuellen Status der Programmelemente

Mit der Statustabelle können Sie die aktuellen Werte von beliebigen Ein- und Ausgängen sowie von Adressen im Speicher überwachen und ändern. Öffnen Sie das Fenster für die Statustabelle und wählen Sie den Menübefehl **Testen ▶ Ständiges Lesen** (siehe Bild 4-16). Befindet sich die CPU im Betriebszustand RUN und werden Eingänge ein- und ausgeschaltet, dann zeigt die Statustabelle den aktuellen Zustand der einzelnen Elemente an.

- Wenn Sie sich den aktuellen Zustand der Elemente in Ihrem Programm anzeigen lassen möchten, klicken Sie auf die Schaltfläche zum einfachen Lesen  oder auf die Schaltfläche zum ständigen Lesen  in der Statustabelle.
- Möchten Sie die Aktualisierung beenden, klicken Sie auf die Schaltfläche zum ständigen Lesen  in der Statustabelle.

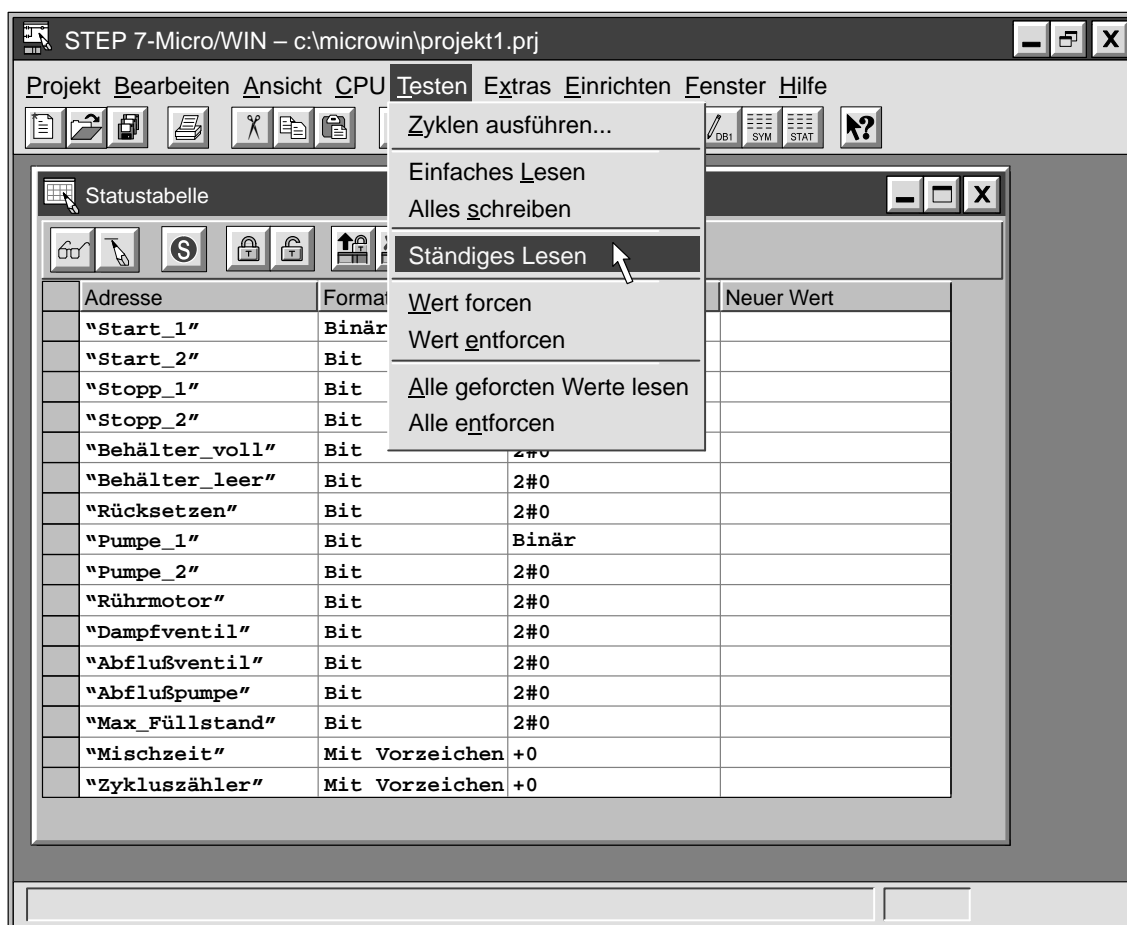


Bild 4-16 Überwachen des Status im Programmierbeispiel mit der Statustabelle

Erweiterte Funktionen in STEP 7-Micro/WIN

5

In diesem Kapitel erfahren Sie, wie Sie mit dem TD 200-Assistenten das Textdisplay TD 200 konfigurieren. Außerdem wird beschrieben, wie Sie mit dem Operations-Assistenten der S7-200 komplexe Operationen ganz einfach konfigurieren können und wie Sie mit allen weiteren neuen Funktionen in Version 2.1 der Software STEP 7-Micro/WIN arbeiten.

Kapitelübersicht

Abschnitt	Beschreibung	Seite
5.1	Konfigurieren des Textdisplay TD 200 mit dem TD 200-Assistenten	5-2
5.2	Arbeiten mit dem Operations-Assistenten der S7-200	5-12
5.3	Arbeiten mit dem Analogeingabefilter-Assistent	5-14
5.4	Erstellen einer Liste der Querverweise	5-17
5.5	Erstellen einer Liste der verwendeten Elemente	5-18
5.6	Arbeiten mit den Funktionen Suchen und Ersetzen	5-19
5.7	Dokumentieren Ihres Programms	5-21
5.8	Drucken Ihres Programms	5-23

5.1 Konfigurieren des Textdisplay TD 200 mit dem TD 200-Assistenten

Das TD 200 ist ein Textdisplay, auf dem Meldungen, die die S7-200 CPU aktiviert, angezeigt werden (siehe Bild 5-1). Das TD 200 müssen Sie weder konfigurieren noch programmieren. Die einzigen Betriebsparameter, die im TD 200 gespeichert werden, sind die Adresse des TD 200, die Adresse der CPU, die Baudrate und die Adresse des Parameterbausteins. Gespeichert wird die Konfiguration des TD 200 im Parameterbaustein, der sich im Variablen-speicher (Datenspeicher) der CPU befindet. Die Betriebsparameter des TD 200, z.B. die Sprache, die Aktualisierungsrate, Meldungen oder die Bits zur Meldungsfreigabe, werden in einem Programm in der CPU abgelegt.

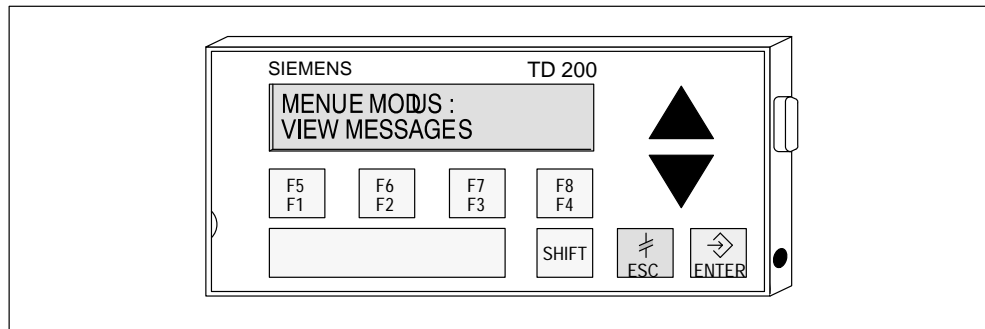


Bild 5-1 SIMATIC Textdisplay TD 200

Definition des Parameterbausteins für das TD 200

Der Parameterbaustein für das TD 200 besteht aus 10 bzw. 12 Bytes, in denen die Betriebsarten definiert sind und die auf die Adressen im Speicher der CPU zeigen, an denen die eigentlichen Meldungen abgelegt sind (siehe Bild 5-2). Wenn Sie das TD 200 einschalten, sucht es in der CPU an dem für das TD 200 konfigurierten Versatz nach der Kennung eines Parameterbausteins (ASCII-Zeichen "TD") oder nach einem Versatz auf die Adresse des Parameterbausteins. Anschließend liest es die Daten aus dem Baustein.

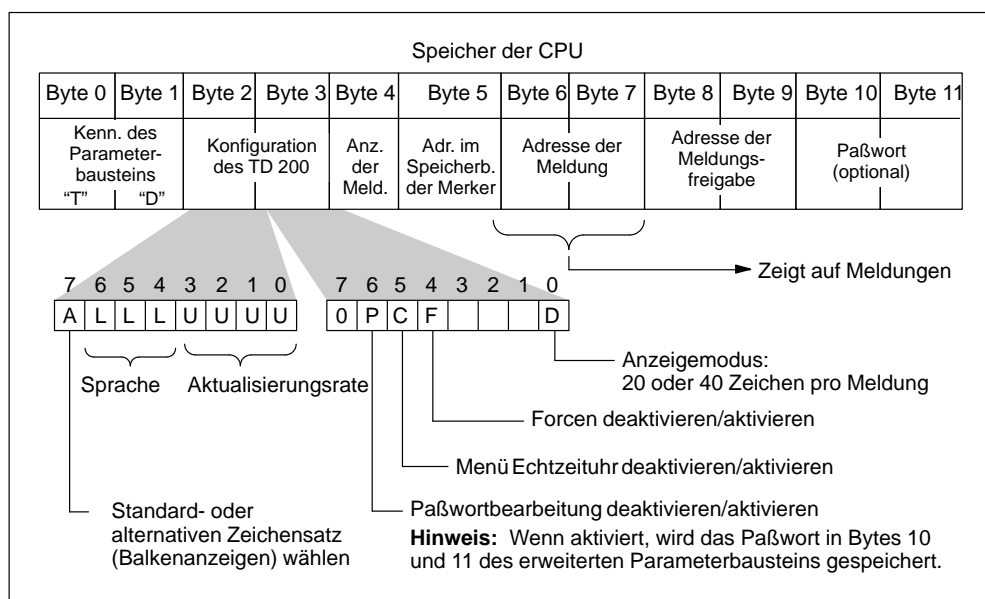


Bild 5-2 Parameterbaustein für das TD 200

Arbeiten mit dem Assistenten zum Konfigurieren des TD 200

STEP 7-Micro/WIN bietet Ihnen einen "Assistenten", mit dem Sie ganz bequem den Parameterbaustein und die Meldungen im Datenspeicher der S7-200 CPU konfigurieren können. Der TD 200-Assistent schreibt den Parameterbaustein und die Meldungstexte automatisch in den Datenbaustein-Editor, nachdem Sie alle Optionen ausgewählt und die Meldungen eingegeben haben. Den Datenbaustein können Sie dann in die CPU laden. Ausführliche Informationen zum TD 200 entnehmen Sie dem *SIMATIC Textdisplay TD 200 Benutzerhandbuch*.

Zum Erstellen des Parameterbausteins und der Meldungen für das TD 200 gehen Sie folgendermaßen vor:

1. Wählen Sie den Menübefehl **Extras ▶ TD 200-Assistent...** (siehe Bild 5-3).
2. Klicken Sie auf die Schaltfläche "Weiter >" oder wählen Sie einen bestehenden Parameterbaustein aus dem aufklappbaren Listefeld aus und folgen Sie den schrittweisen Anleitungen zum Erstellen bzw. Bearbeiten eines Parameterbausteins für das TD 200 im Variablenspeicher.

Sie können jederzeit auf die Schaltfläche "< Zurück" klicken, um zum vorherigen Dialogfeld zurückzukehren und die dort definierten Parameter einzusehen oder zu ändern.

3. Nachdem Sie alle Dialogfelder ausgefüllt haben, bestätigen Sie Ihre Eingaben mit "Beenden" und speichern so den Parameterbaustein. Sie können sich den eingerichteten Parameterbaustein im Datenbaustein-Editor anzeigen lassen.

Wenn Sie alle Bausteine in die S7-200 CPU laden, wird der Datenbaustein, der den Parameterbaustein für das TD 200 enthält, im Speicher der CPU abgelegt, wo er vom TD 200 gelesen werden kann.



Bild 5-3 Aufrufen des TD 200-Assistenten

Auswählen der Sprache und des Zeichensatzes für Balkenanzeigen

Im ersten Dialogfeld des TD 200-Assistenten können Sie die Sprache und den Zeichensatz auswählen. Wählen Sie in dem aufklappbaren Listenfeld die Sprache aus, in der die Menüs des TD 200 angezeigt werden sollen (siehe Bild 5-4). Wählen Sie über die runden Optionsfelder den Standard-Zeichensatz oder den alternativen Zeichensatz aus. Mit dem alternativen Zeichensatz können Sie auf dem TD 200 Balkendiagramme anzeigen. Der TD 200-Assistent setzt die entsprechenden Bits im Byte 2 des Parameterbausteins.

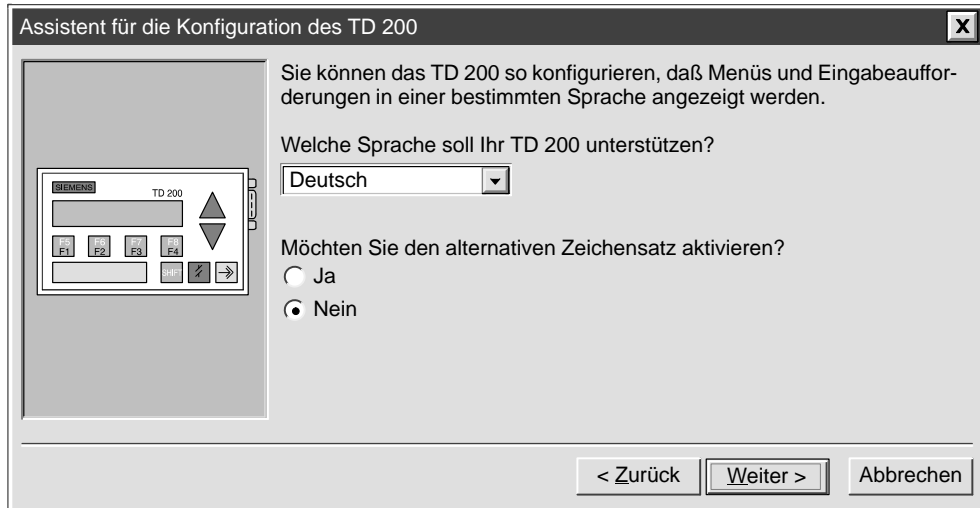


Bild 5-4 Sprache und Zeichensatz für das TD 200

Aktivieren der Echtzeituhr, der Funktion zum Forcen und des Paßwortschutzes

Über die runden Optionsfelder wählen Sie die jeweiligen Funktionen wie in Bild 5-5 gezeigt aus. Aktivieren Sie den Paßwortschutz, wird ein Feld aufgeblendet, in dem Sie ein Paßwort definieren müssen. Ausführliche Informationen zum TD 200 entnehmen Sie dem *SIMATIC Textdisplay TD 200 Benutzerhandbuch*. Der TD 200-Assistent setzt die entsprechenden Bits im Byte 3 des Parameterbausteins.

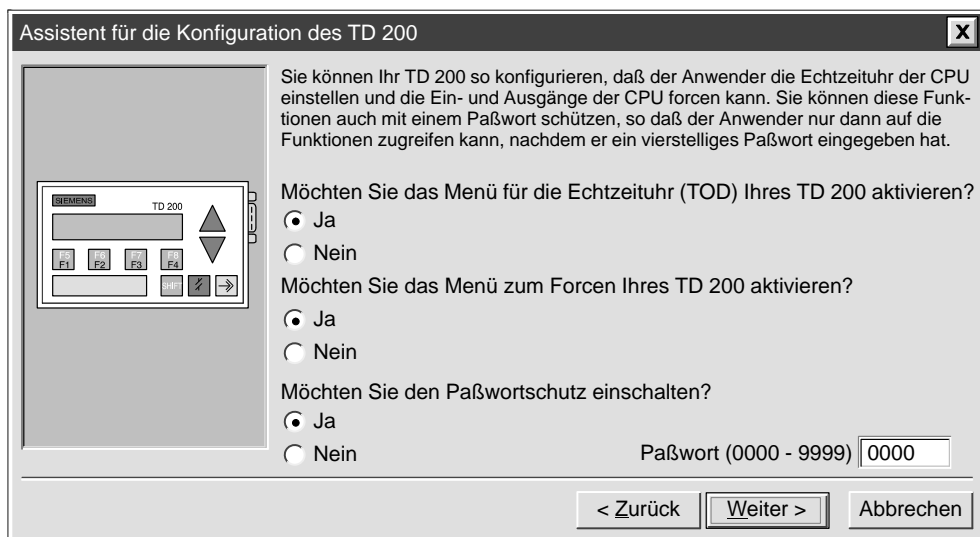


Bild 5-5 Aktivieren der Echtzeituhr, der Funktion zum Forcen und des Paßwortschutzes beim TD 200

Festlegen der Merker für die Funktionstasten und der Aktualisierungsrate für die Anzeige

Sie müssen die acht Bits eines Byte im Speicherbereich der Merker für die Funktionstasten des TD 200 reservieren. Gültige Werte liegen zwischen 0 und 15 in der CPU 212 und zwischen 0 und 31 in der CPU 214, CPU 215 und CPU 216. Der TD 200-Assistent schreibt die Werte in Byte 5 des Parameterbausteins. Wählen Sie in dem aufklappbaren Listenfeld die Aktualisierungsrate für die Anzeige aus (siehe Bild 5-6). Der TD 200-Assistent setzt die entsprechenden Bits im Byte 2 des Parameterbausteins.

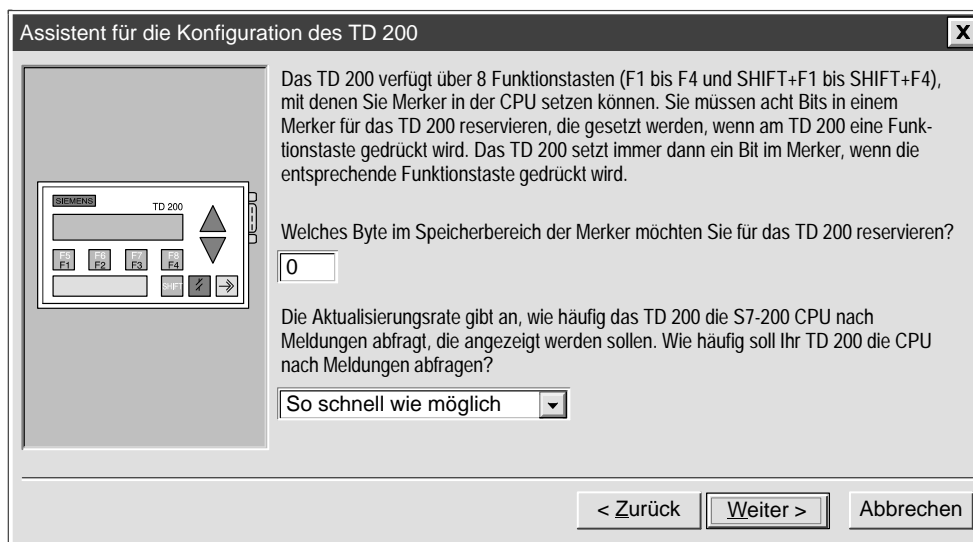


Bild 5-6 Merker für die Funktionstasten und die Aktualisierungsrate für die Anzeige im TD 200



Warnung

Das TD 200 setzt, immer wenn eine Funktionstaste gedrückt wird, ein Bit im Merker. Wenn Sie keine Funktionstasten verwenden möchten und deshalb auch keine Adresse im Speicherbereich der Merker für die Funktionstasten zuordnen, richtet das TD 200 standardmäßig das Byte M0 für die Funktionstasten ein. Belegt Ihr Programm Bits im Merker M0 und Sie drücken eine Funktionstaste, dann setzt das TD 200 das entsprechende Bit in M0 und überschreibt den von Ihrem Programm abgelegten Wert.

Diese unbeabsichtigten Änderungen im Merker können unerwartete Reaktionen in Ihrem Programm verursachen. Unvorhersehbarer Betrieb eines Automatisierungssystems kann zu Tod, schweren Körperverletzungen und/oder Sachschaden führen.

Reservieren Sie deshalb immer eine Adresse im Speicherbereich der Merker, auch wenn Ihr Programm keine Funktionstasten benötigt.

Auswählen von Größe und Anzahl der Meldungen

Wählen Sie über die runden Optionsfelder die Größe der Meldungen aus (Bit 0 in Byte 3 im Parameterbaustein). Geben Sie in das Textfeld eine Zahl zwischen 1 und 80 an, die angibt, wieviele Meldungen Sie definieren möchten. Der entsprechende Wert wird in Byte 4 des Parameterbausteins geschrieben (siehe Bild 5-7).

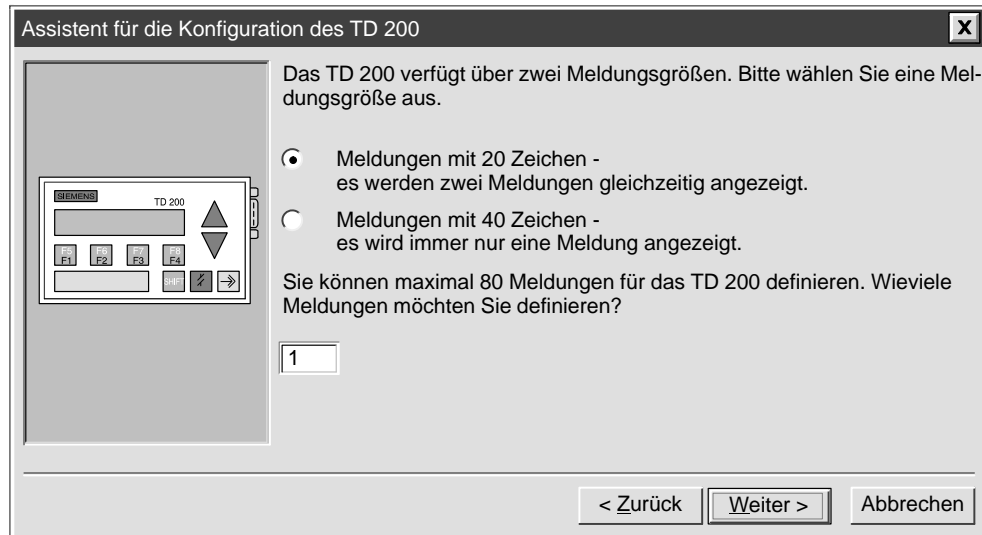


Bild 5-7 Größe und Anzahl der Meldungen für das TD 200

Angeben der Adressen für den Parameterbaustein, für die Freigabemerker und für die Meldungen

In dem in Bild 5-8 gezeigten Dialogfeld können Sie die Anfangsadressen für den Parameterbaustein, für die Freigabemerker und für die Meldungen angeben.

- Das TD 200 sucht immer nach einer Kennung des Parameterbausteins an dem konfigurierten Versatz in der CPU. Geben Sie die Adresse des Parameterbausteins in dem ersten Textfeld ein, wenn Sie möchten, daß der Parameterbaustein nicht an der Standardadresse, sondern an einer anderen Adresse abgelegt wird. Der entsprechende Wert wird in Byte 0 und 1 des Parameterbausteins geschrieben.
- Geben Sie dann eine Adresse im Variablenspeicher an, an der die Merker zur Meldungsfreigabe abgelegt werden sollen. Dieser Wert wird in Byte 8 und 9 des Parameterbausteins geschrieben.
- Geben Sie zum Schluß noch eine Adresse im Variablenspeicher an, an der die Meldungen in aufeinanderfolgenden Bytes abgelegt werden sollen. (Der Wert "32" ist nur die Voreinstellung.) Die von Ihnen angegebene Adresse wird in Bytes 6 und 7 des Parameterbausteins geschrieben. Die erforderliche Anzahl Bytes richtet sich nach der Anzahl der Meldungen, die Sie konfigurieren möchten, und wird im Dialogfeld angezeigt. Beachten Sie, daß eine Meldung mit 20 Zeichen auch 20 aufeinanderfolgende Bytes im Variablenspeicher belegt, während eine Meldung mit 40 Zeichen 40 aufeinanderfolgende Bytes benötigt.

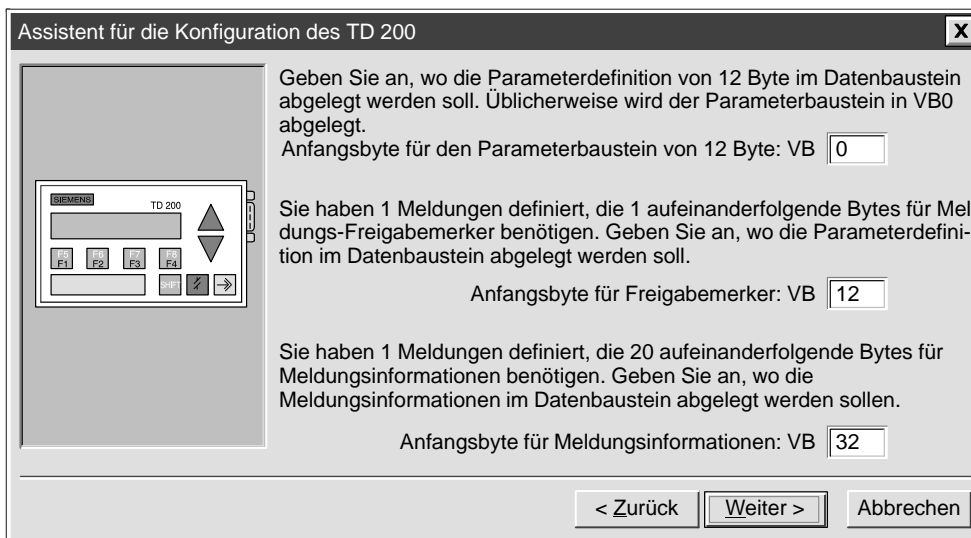


Bild 5-8 Adressen für den Parameterbaustein, die Freigabemerker und die Meldungen beim TD 200

Erstellen von Meldungen für das TD 200

In dem in Bild 5-9 gezeigten Dialogfeld können Sie den Text für die Meldungen mit 20 oder 40 Zeichen, die Sie in Bild 5-8 ausgewählt haben, eingeben. Die Meldungen werden im Variablenspeicher an der Anfangsadresse, die Sie in Bild 5-8 angegeben haben, abgelegt (siehe Bild 5-9).

Geben Sie Ihre Meldung zeichenweise (ein Zeichen pro Textfeld) ein. Haben Sie angegeben, daß Sie mehr als eine Meldung definieren möchten, wählen Sie die Schaltfläche "Nächste Meldung >", um den Text für weitere Meldungen einzugeben.

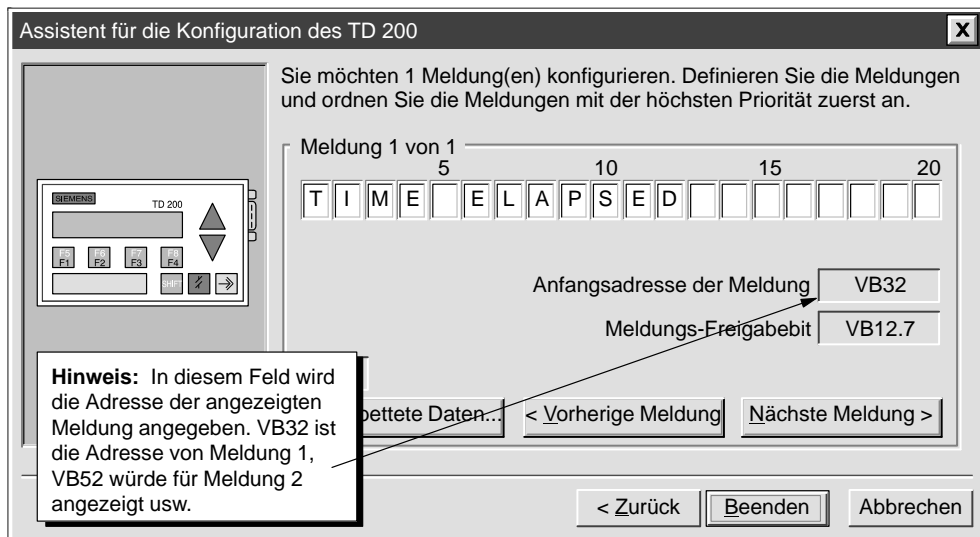


Bild 5-9 Dialogfeld zum Konfigurieren von Meldungen für das TD 200

Einbetten von Datenwerten in eine Meldung

Sie können in die Meldung, die auf dem TD 200 angezeigt wird, einen Datenwert einbetten. Sie können beispielsweise eine Meldung erstellen, die einen aus der CPU gelesenen Wert der abgelaufenen Zeit anzeigt. Damit der Datenwert angezeigt werden kann, müssen Sie in der Meldungen Platz reservieren.

Zum Einfügen eines Platzhalters für eine Datenvariable, positionieren Sie den Cursor an der Position der ersten Ziffer und wählen die Schaltfläche "Eingebettete Daten..." im unteren Teil des Dialogfelds. Daraufhin wird ein Dialogfeld angezeigt, in dem Sie das Format für den Datenwert und weitere Optionen auswählen können. Sie können z.B. definieren, ob die Meldung quittiert werden muß, ob der Datenwert bearbeitet werden kann und ob die Bearbeitung durch ein Paßwort geschützt werden soll.

Eingeben von internationalen Zeichen und Sonderzeichen

Wenn Sie im TD 200-Assistent in STEP 7-Micro/WIN gewisse Sonderzeichen oder internationale Zeichen eingeben, kann es sein, daß diese Zeichen in dem Anzeigefeld des TD 200 nicht korrekt wiedergegeben werden. Ist dies der Fall, geben Sie diese Zeichen mit Hilfe der in Tabelle 5-1 aufgeführten ALT-Tastenkombinationen im TD 200-Assistent ein.

Tabelle 5-1 ALT-Tastenkombinationen für internationale Zeichen und Sonderzeichen

Zeichen	ALT-Tastenkombination	Zeichen	ALT-Tastenkombination
ü	ALT+0129	ñ	ALT+0164
ä	ALT+0132	Ω	ALT+0234
æ	ALT+0145	Σ	ALT+0228
Æ	ALT+0146	Π	ALT+0227
å	ALT+0134	¥	ALT+0157
ö	ALT+0148	←	ALT+0195 (Pfeil nach links ←)
Å	ALT+0143	→	ALT+0180 (Pfeil nach rechts →)
°	ALT+0248		ALT+0200 (einfacher Balken)
α	ALT+0224		ALT+0201 (doppelter Balken)
β	ALT+0225		ALT+0202 (dreifacher Balken)
ε	ALT+0238		ALT+0203 (vierfacher Balken)
μ	ALT+0230		ALT+0204 (fünffacher Balken)
σ	ALT+0229	↑	ALT+0194 (Pfeil nach oben)
ç	ALT+0155		

Formatieren des eingebetteten Datenwerts

Bild 5-10 zeigt das Dialogfeld, in dem Sie die Parameter für den anzuzeigenden Wert definieren. Die Formate und Optionen, die Sie angeben, werden in ein Formatwort (zwei Bytes) geschrieben, das dem eingebetteten Wert vorangestellt wird. Wählen Sie die Größe, das Anzeigeformat, die Position des Dezimalkommata und andere Optionen für den eingebetteten Datenwert aus.

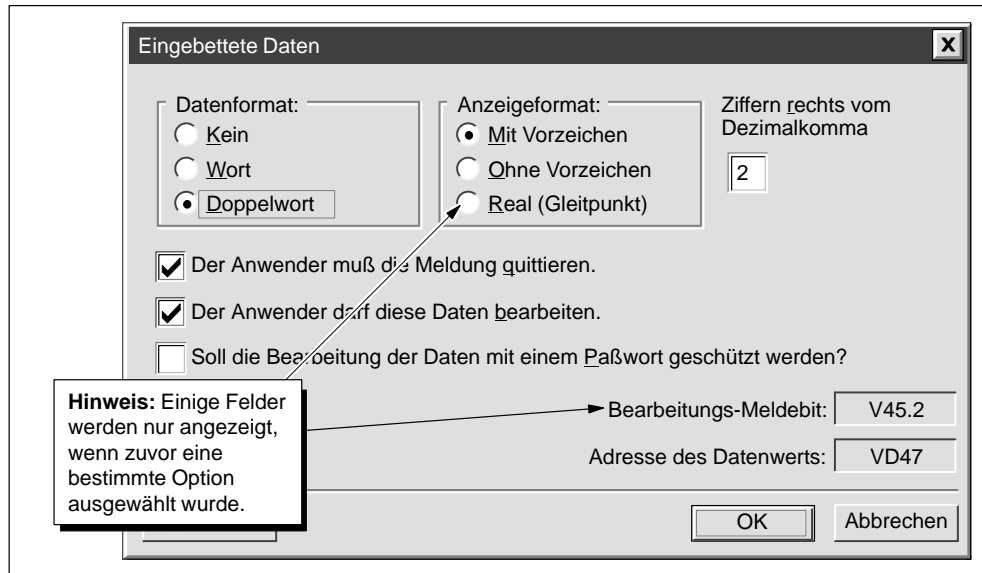


Bild 5-10 Dialogfeld "Eingebettete Daten" im TD 200-Assistenten

In Bild 5-11 sehen Sie das Dialogfeld des TD 200-Assistenten, nachdem Sie die Parameter für den eingebetteten Datenwert eingegeben haben. Die grau unterlegten Felder sind Platzhalter für den Datenwert. Haben Sie angegeben, daß der Anwender die Meldung quittieren muß, dann wird das Bit zur Quittierungsmeldung in dem Dialogfeld angezeigt.

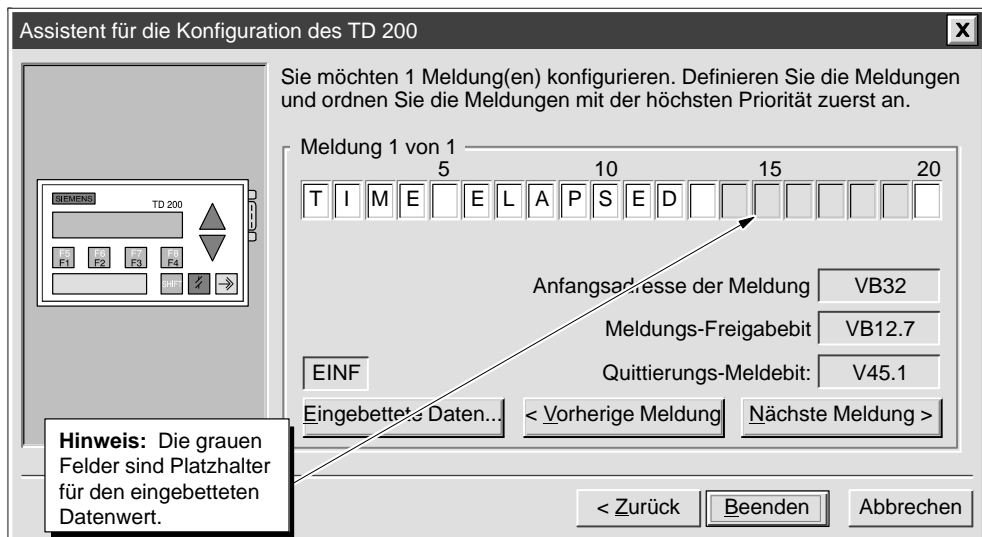
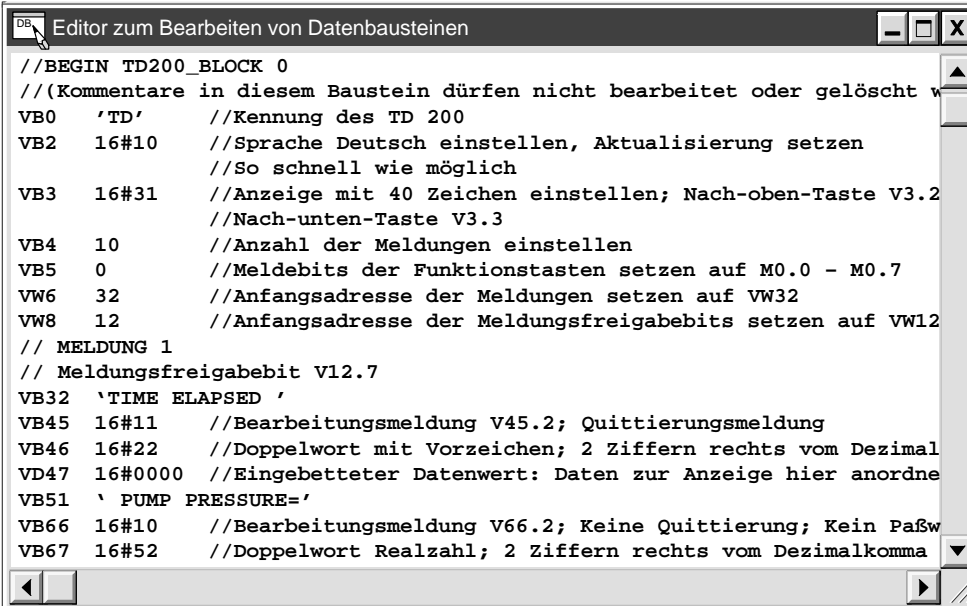


Bild 5-11 Platzhalter für einen eingebetteten Datenwert in einer Meldung für das TD 200

Vervollständigen und Anzeigen des Parameterbausteins für das TD 200

Wählen Sie die Schaltfläche "Nächste Meldung >" und geben Sie den Text für die folgende Meldung ein. Nachdem Sie alle Meldungen für das TD 200 eingegeben haben, klicken Sie auf die Schaltfläche "Beenden", um den konfigurierten Parameterbaustein und die Meldungen im Datenbaustein zu speichern.

Sie können sich den vom TD 200-Assistenten formatierten Parameterbaustein für das TD 200 ansehen, wenn Sie den Datenbaustein-Editor aufrufen. Bild 5-12 zeigt ein Beispiel für einen Parameterbaustein, der für eine Meldung mit 40 Zeichen konfiguriert wurde, in der Darstellung im Datenbaustein-Editor.



```

//BEGIN TD200_BLOCK 0
//(Kommentare in diesem Baustein dürfen nicht bearbeitet oder gelöscht werden)
VB0 'TD' //Kennung des TD 200
VB2 16#10 //Sprache Deutsch einstellen, Aktualisierung setzen
//So schnell wie möglich
VB3 16#31 //Anzeige mit 40 Zeichen einstellen; Nach-oben-Taste V3.2
//Nach-unten-Taste V3.3
VB4 10 //Anzahl der Meldungen einstellen
VB5 0 //Meldebits der Funktionstasten setzen auf M0.0 - M0.7
VW6 32 //Anfangsadresse der Meldungen setzen auf VW32
VW8 12 //Anfangsadresse der Meldungsfreigabebits setzen auf VW12
// MELDUNG 1
// Meldungsfreigabebit V12.7
VB32 'TIME ELAPSED '
VB45 16#11 //Bearbeitungsmeldung V45.2; Quittierungsmeldung
VB46 16#22 //Doppelwort mit Vorzeichen; 2 Ziffern rechts vom Dezimal
VD47 16#0000 //Eingebetteter Datenwert: Daten zur Anzeige hier anordnen
VB51 ' PUMP PRESSURE='
VB66 16#10 //Bearbeitungsmeldung V66.2; Keine Quittierung; Kein Paßw
VB67 16#52 //Doppelwort Realzahl; 2 Ziffern rechts vom Dezimalkomma

```

Bild 5-12 Datenbaustein mit Parameterbaustein für das TD 200

5.2 Arbeiten mit dem Operations-Assistenten der S7-200

In STEP 7-Micro/WIN gibt es den Operations-Assistenten der S7-200, mit dem Sie folgende komplexe Operationen schnell und einfach konfigurieren können.

- Konfigurieren der Funktionsweise eines PID-Reglers
- Konfigurieren der Operationen Aus Netz lesen und In Netz schreiben
- Konfigurieren eines Algorithmus zum Abfragen und zur Mittelwertbildung (Analog-Eingabefilter)
- Konfigurieren der Funktionsweise eines schnellen Zählers

Im Abschnitt 5.3 wird ein Beispiel für den Analogeingabefilter-Assistent dargestellt.

Aufrufen des Operations-Assistenten der S7-200

Zum Aufrufen des Operations-Assistenten der S7-200 gehen Sie folgendermaßen vor:

1. Wählen Sie den Menübefehl **Extras ▶ TD 200-Assistent...** (siehe Bild 5-13).
2. Markieren Sie den Assistenten, mit dem Sie arbeiten möchten.
3. Wählen Sie die Schaltfläche "Weiter >". Haben Sie Ihr Programm nach der letzten Bearbeitung nicht übersetzt, müssen Sie es jetzt übersetzen. Da die Übersetzung des Programms einige Zeit dauern kann (wenn es sich um ein langes Programm handelt), werden Sie gefragt, ob Sie fortfahren und übersetzen möchten. Es erscheint die Meldung "Übersetzung erforderlich. Um fortfahren zu können, muß Ihr Programm übersetzt werden. Soll es jetzt übersetzt werden?". Wählen Sie "OK", wenn Sie übersetzen möchten oder "Abbrechen", wenn Sie den Assistenten beenden und das Programm nicht übersetzen möchten.
4. Haben Sie jedoch einen Assistenten ausgewählt und Ihr Programm übersetzt, werden die Dialogfelder des jeweiligen Assistenten nacheinander aufgerufen.

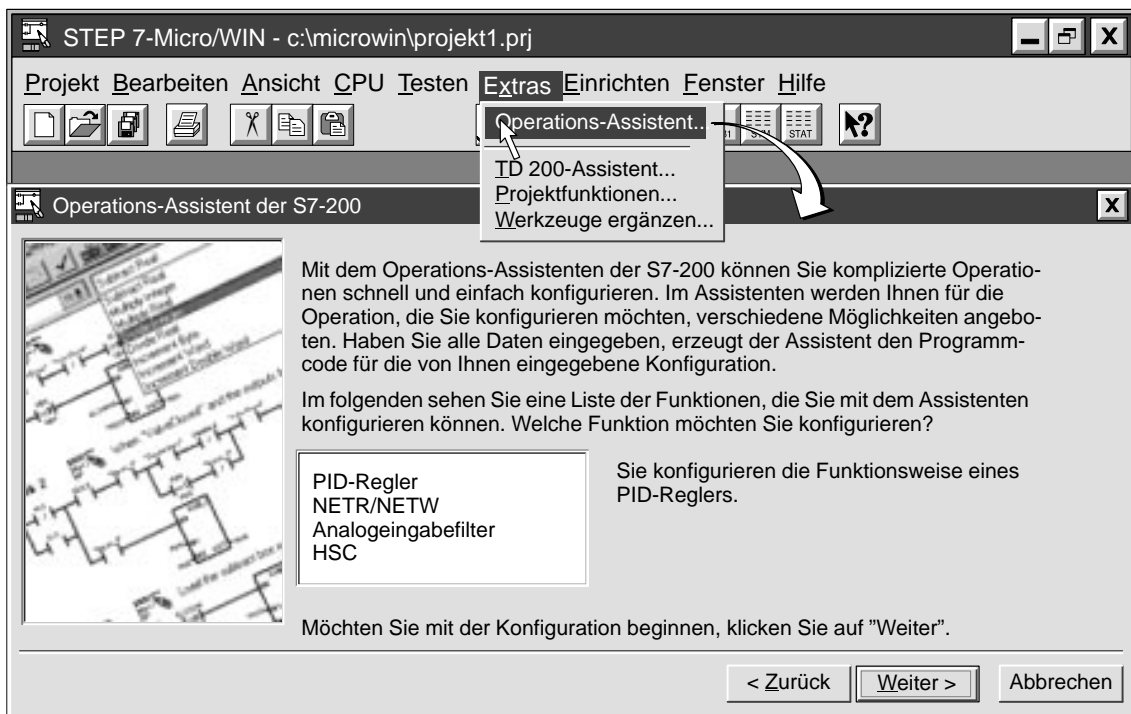


Bild 5-13 Arbeiten mit dem Operations-Assistenten der S7-200

Nachdem Sie alle Fragen im jeweiligen Assistenten beantwortet haben, erreichen Sie das letzte Dialogfeld des Operations-Assistenten der S7-200 (siehe Bild 5-14). In diesem Dialogfeld werden Ihnen die Programmsegmente aufgeführt, die für die von Ihnen angegebene Konfiguration erzeugt werden. Außerdem können Sie hier angeben, an welcher Stelle im Hauptprogramm der erzeugte Code angeordnet werden soll.

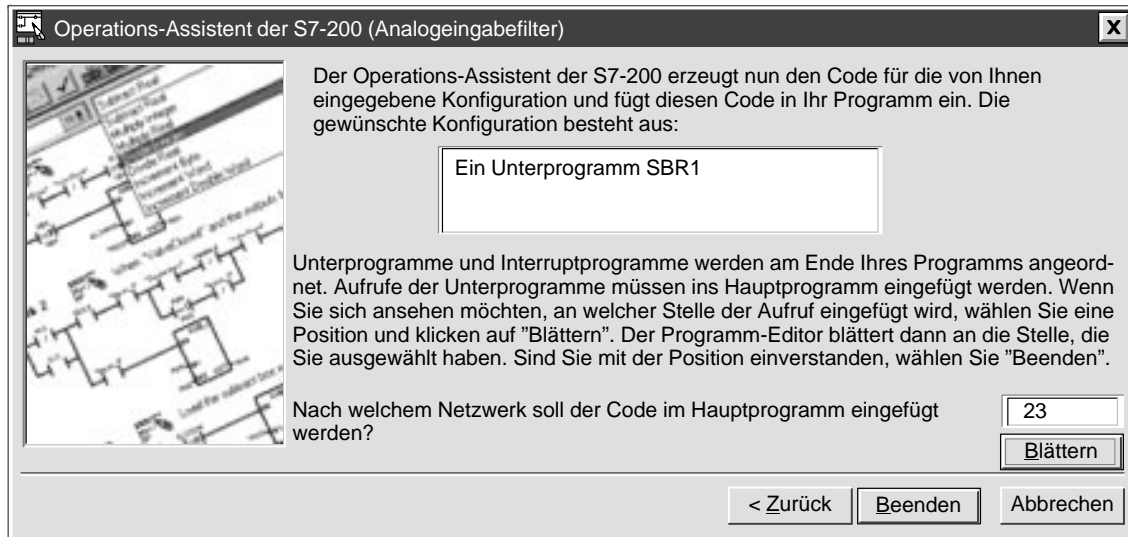


Bild 5-14 Vom Operations-Assistenten der S7-200 erzeugte Programmsegmente

5.3 Arbeiten mit dem Analogeingabefilter-Assistent

Mit dem Analogeingabefilter-Assistent können Sie Ihr Programm um ein Programm zur Mittelwertbildung erweitern. Das Analogmodul der S7-200 ist ein Hochgeschwindigkeitsmodul. Es kann schnelle Wechsel des Eingangesignals verarbeiten (einschließlich internem und externem Rauschen). Abweichungen zwischen den ausgelesenen Werten, die durch Störungen in einer Konstante oder in einem sich langsam ändernden Analogeingabesignal entstehen, können Sie verringern, indem Sie aus mehreren ausgelesenen Werten den Mittelwert bilden. Je weiter Sie die Anzahl der für die Mittelwertbildung herangezogenen Werte erhöhen, desto stärker können Sie eine entsprechend langsamere Reaktionszeit auf Änderungen im Eingangesignal feststellen. Ein Mittelwert, der aus einer großen Zahl von abgefragten Werten gebildet wird, kann die ausgelesenen Werte stabilisieren, während die Reaktionszeit auf Änderungen im Eingangesignal verlangsamt wird.

Basisfilter

Zum Einrichten des grundlegenden Filters müssen Sie drei Fragen beantworten:

1. Welchen Analogeingang möchten Sie filtern? (AEW0, AEW2, AEW4,...)
2. In welche Adresse soll der gefilterte Wert geschrieben werden? (VWx, AAWx, ...)
3. An welcher Adresse möchten Sie den Speicher für Zwischenberechnungen anordnen? Der Code zum Filtern benötigt 12 Bytes im Speicher für Berechnungen. (VBx, ...)

Zusätzliche Filtereinstellungen

Sie können zusätzliche Einstellungen für den konfigurierten Analogeingabefilter definieren.

- Anzahl der Abfragen
- Fehlerbedingungen

Angeben der Ein- und Ausgänge

Geben Sie an, welchen Analogeingang Sie filtern möchten, und in welchem Ausgang das Ergebnis abgelegt werden soll (siehe Bild 5-15). Sie können für den Ausgang sowohl eine absolute Adresse als auch einen symbolischen Namen eingeben.

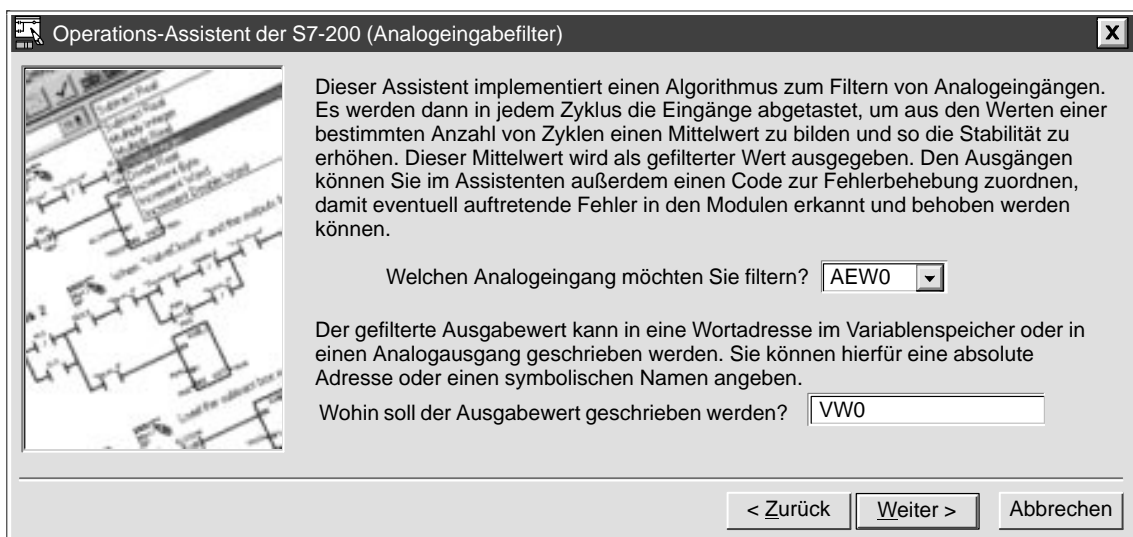


Bild 5-15 Angeben der Ein- und Ausgänge im Analogeingabefilter-Assistenten

Definieren der Adresse für den Speicherbereich von 12 Byte für Berechnungen

Geben Sie an, an welcher Adresse der Speicherbereich von 12 Byte für Berechnungen beginnen soll (siehe Bild 5-16). Außerdem müssen Sie die Nummer des Unterprogramms, in dem der erzeugte Code angeordnet werden soll, und die Anzahl der Abfragen angeben.

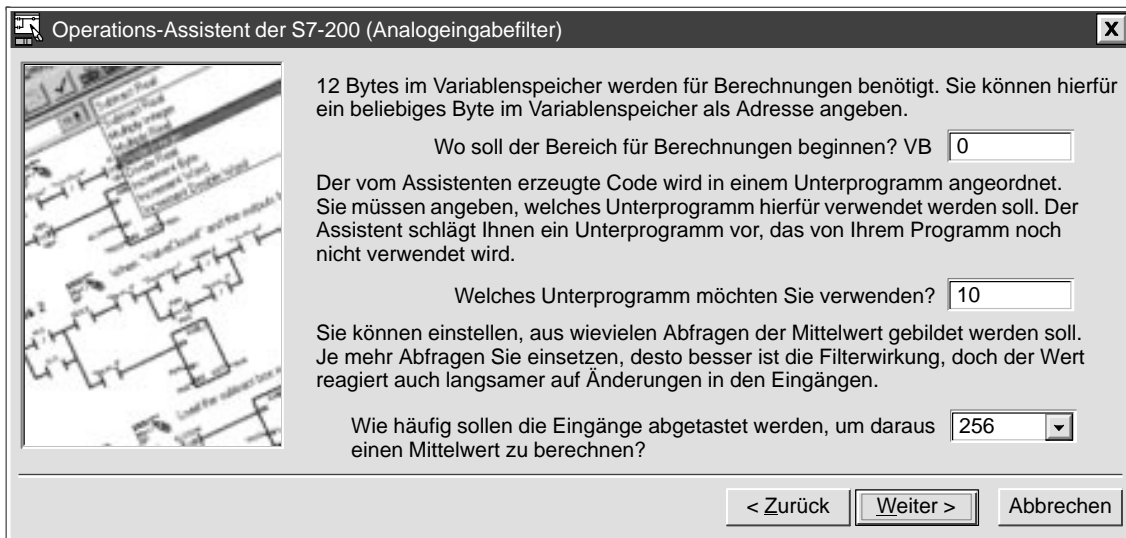


Bild 5-16 Definieren der Adresse für den Speicherbereich von 12 Byte für Berechnungen

Fehlerprüfung im Modul

Sie können Ihre Konfiguration um die Option erweitern, daß das Analogmodul auf Fehler geprüft werden soll. Sie müssen hierzu die Position des Analogmoduls relativ zur CPU angeben, damit der Code erzeugt werden kann, der die entsprechenden Sondermerker prüft. Sie müssen außerdem ein Bit angeben, das den Fehlerstatus des Moduls speichert. Tritt ein Modulfehler auf, wird dieses Bit gesetzt. Möchten Sie, daß im Fall eines Modulfehlers ein bestimmter Wert ausgegeben wird, müssen Sie den Wert, der ausgegeben werden soll, angeben (siehe Bild 5-17).

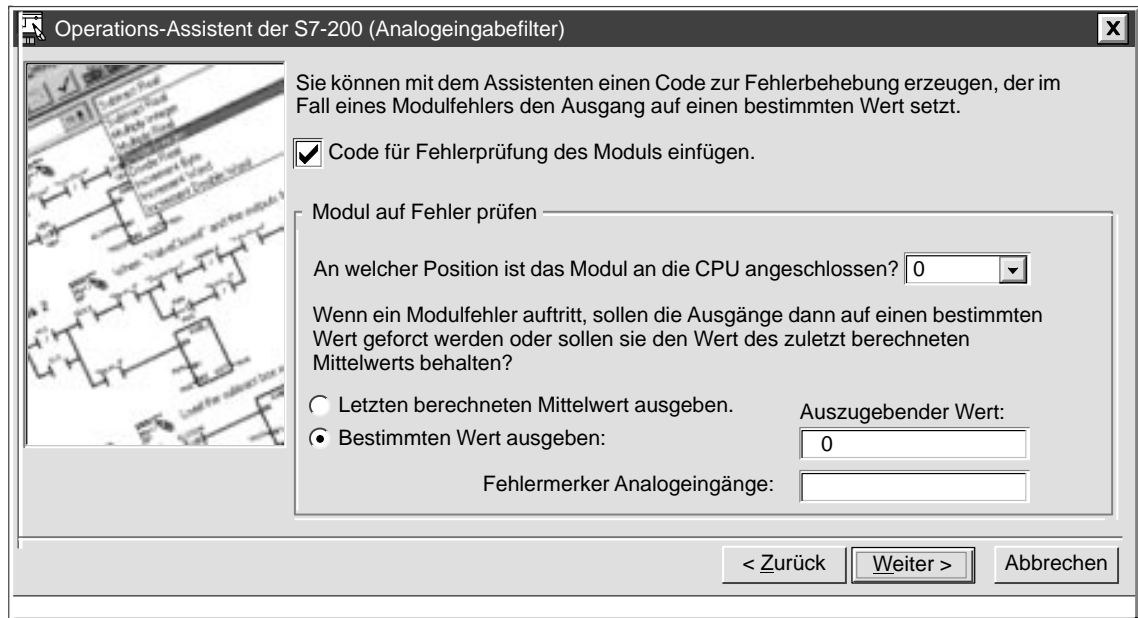


Bild 5-17 Analogeingabefilter-Assistent: Ausgeben eines bestimmten Werts bei Auftreten eines Modulfehlers

Sie können aber auch den zuletzt berechneten Mittelwert ausgeben, wenn ein Modulfehler auftritt (siehe Bild 5-18).

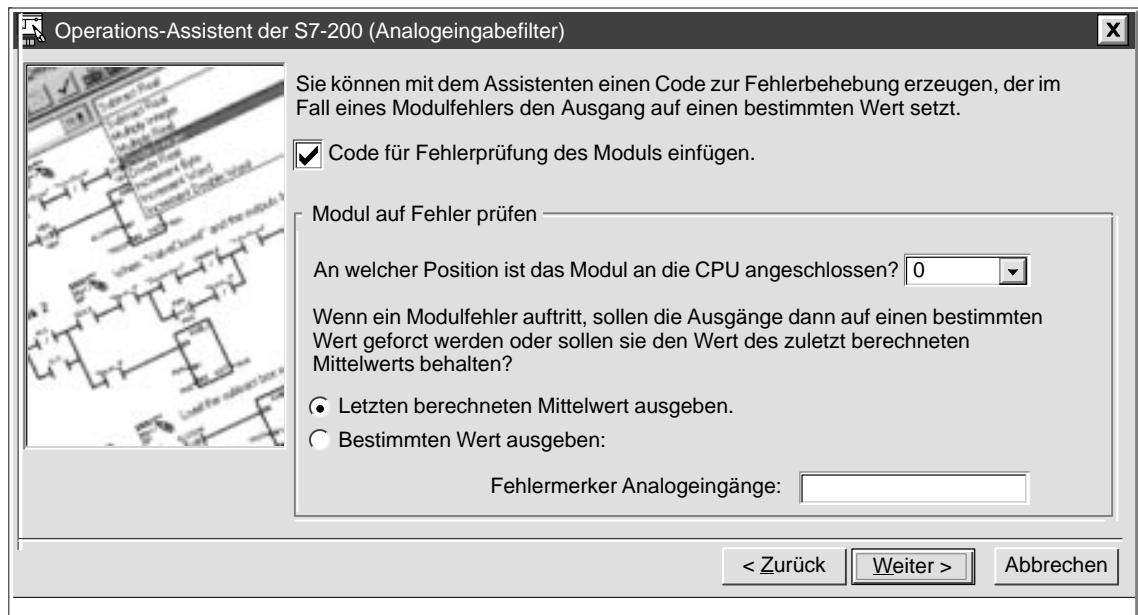


Bild 5-18 Analogeingabefilter-Assistent: Ausgeben des letzten berechneten Mittelwerts bei Auftreten eines Modulfehlers

5.4 Erstellen einer Liste der Querverweise

Beim Erstellen der Querverweise erzeugen Sie eine Liste der in Ihrem Programm verwendeten Adressen. In der Tabelle der Querverweise können Sie die Adressen überwachen, während Sie Ihr Programm schreiben. Möchten Sie die Querverweise erstellen, wird Ihr Programm zunächst übersetzt und anschließend die Tabelle der Querverweise erzeugt.

Die Tabelle der Querverweise zeigt den Namen des Elements, die Nummer des Netzwerks und die Operation (siehe Bild 5-19). Indirekte Adressen werden in der Tabelle der Querverweise durch die Symbole * oder & gekennzeichnet.

Zum Erstellen einer Tabelle der Querverweise gehen Sie folgendermaßen vor:

1. Wählen Sie den Menübefehl **Ansicht ► Querverweise**.
2. Ihr Programm wird übersetzt und die Tabelle der Querverweise wird erzeugt.
3. Beim Eingeben Ihres Programms können Sie die Tabelle der Querverweise geöffnet lassen. Ändern Sie Ihr Programm und klicken dann in die Tabelle der Querverweise, müssen Sie die Tabelle aktualisieren.
4. Wenn Sie sich ein Element in Ihrem Programm ansehen möchten, doppelklicken Sie auf das gewünschte Element in der Tabelle der Querverweise. Daraufhin wird das Element im Programm-Editor markiert.

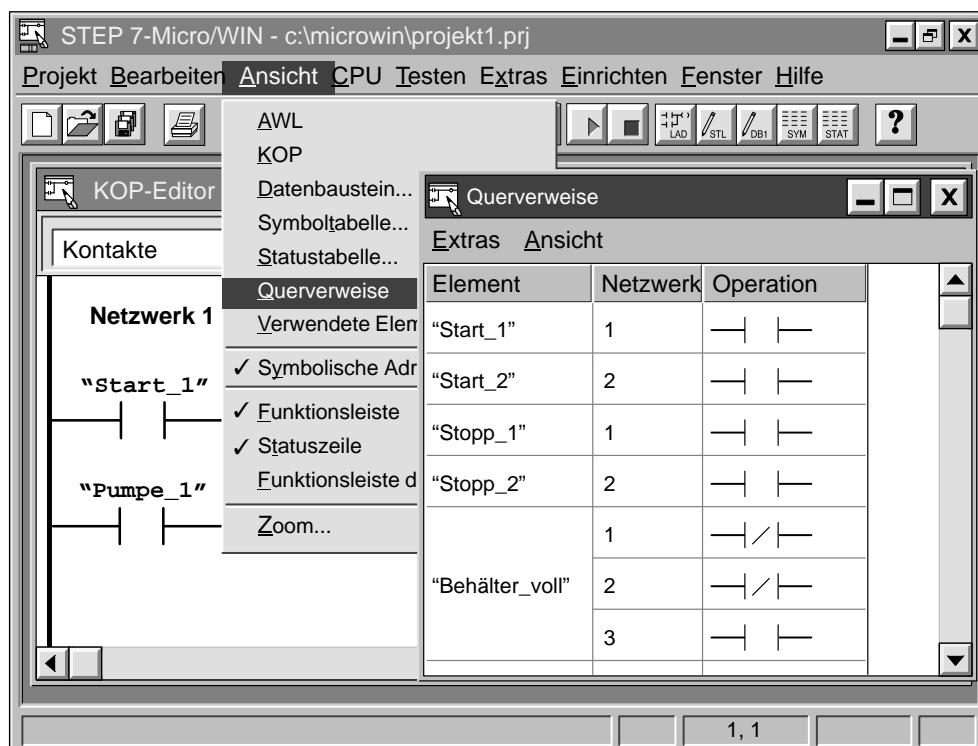


Bild 5-19 Anzeigen der Liste der Querverweise

5.5 Erstellen einer Liste der verwendeten Elemente

In der Tabelle der verwendeten Elemente können Sie sich die Adressen und Bereiche, die Sie in Ihrem Programm vergeben haben, anzeigen lassen. Die Tabelle der verwendeten Elemente zeigt diese Informationen in kompakterer Form an als die Tabelle der Querverweise. Der gezeigte Bereich beginnt mit der ersten verwendeten Adresse und endet mit der letzten verwendeten Adresse. Nicht verwendete Adressen werden als leere Reihen dargestellt (siehe Bild 5-20).

Es gibt zwei Möglichkeiten, um sich die verwendeten Elemente anzeigen zu lassen:

- Im Bitformat werden E, A, M und S angezeigt.
- Im Byteformat werden V, AEW, AAW, MB, SMB, T, Z und HSC angezeigt.

Beachten Sie folgendes:

- Lassen Sie sich die verwendeten Elemente im Byteformat anzeigen, wird eine Doppelwortadresse als vier aufeinanderfolgende D dargestellt. Werden nicht vier aufeinanderfolgende D angezeigt, kann es sein, daß Sie diese Adresse doppelt vergeben haben oder daß es sich um eine absichtliche Programmiertechnik handelt. (Ein Wort wird als zwei aufeinanderfolgende W angezeigt, ein Byte wird als ein B dargestellt und ein Bit als b.)
- Mit Gedankenstrichen gekennzeichnete Elemente (--) verweisen auf referenzierte Bereiche. Ein referenzierter Bereich besteht aus Adressen, die von einer Operation verwendet werden, ohne daß diese Adressen explizit angegeben sind. Die Operation Aus Netz lesen (NETR) beispielsweise verwendet eine Tabelle im Variablenspeicher, die 8 Byte belegt. Nur das erste Byte ist die einzige explizit angegebene Adresse dieses Bereichs.

Zum Erstellen einer Tabelle der verwendeten Elemente wählen Sie den Menübefehl **Ansicht** ► **Verwendete Elemente**. Ihr Programm wird übersetzt und die Tabelle der verwendeten Elemente erzeugt (siehe Bild 5-20). Sie können die Tabelle der verwendeten Elemente beim Eingeben Ihres Programms geöffnet lassen. Ändern Sie Ihr Programm und klicken dann in die Tabelle der verwendeten Elemente, müssen Sie die Tabelle aktualisieren.

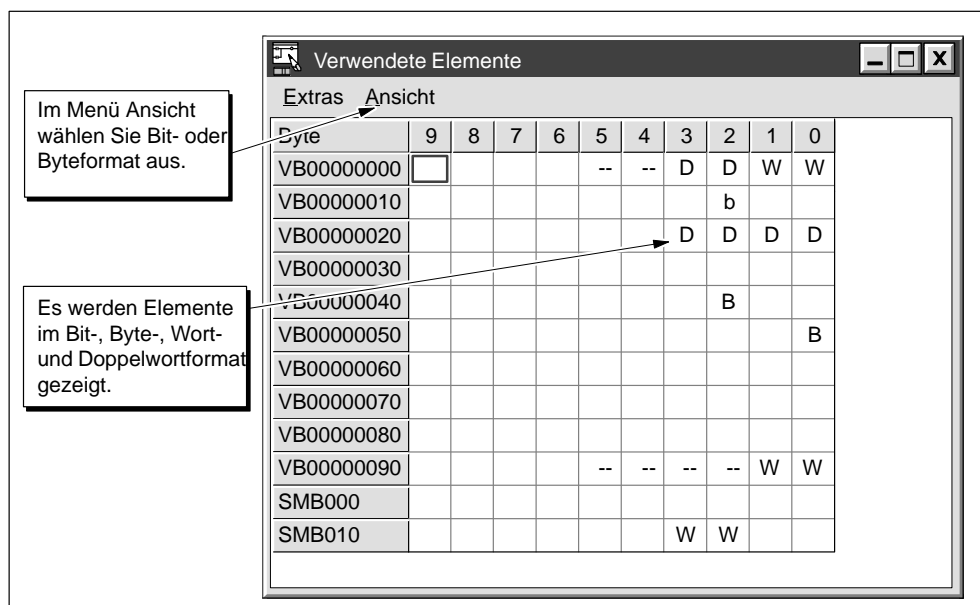


Bild 5-20 Anzeigen der Tabelle der verwendeten Elemente

5.6 Arbeiten mit den Funktionen Suchen und Ersetzen

Mit der Suchfunktion können Sie nach einem bestimmten Parameter suchen und mit der Funktion Ersetzen können Sie diesen Parameter durch einen anderen ersetzen (siehe Bild 5-21).

Suchen nach einem Parameter

Zum Suchen nach einem bestimmten Parameter gehen Sie folgendermaßen vor:

1. Wählen Sie den Menübefehl **Bearbeiten ► Suchen....** Bild 5-21 zeigt das Dialogfeld "Suchen".
2. Geben Sie an, wonach Sie suchen möchten.
3. Geben Sie an, in welche Richtung Sie suchen möchten.
4. Klicken Sie auf die Schaltfläche "Weitersuchen", um die Suche zu starten.

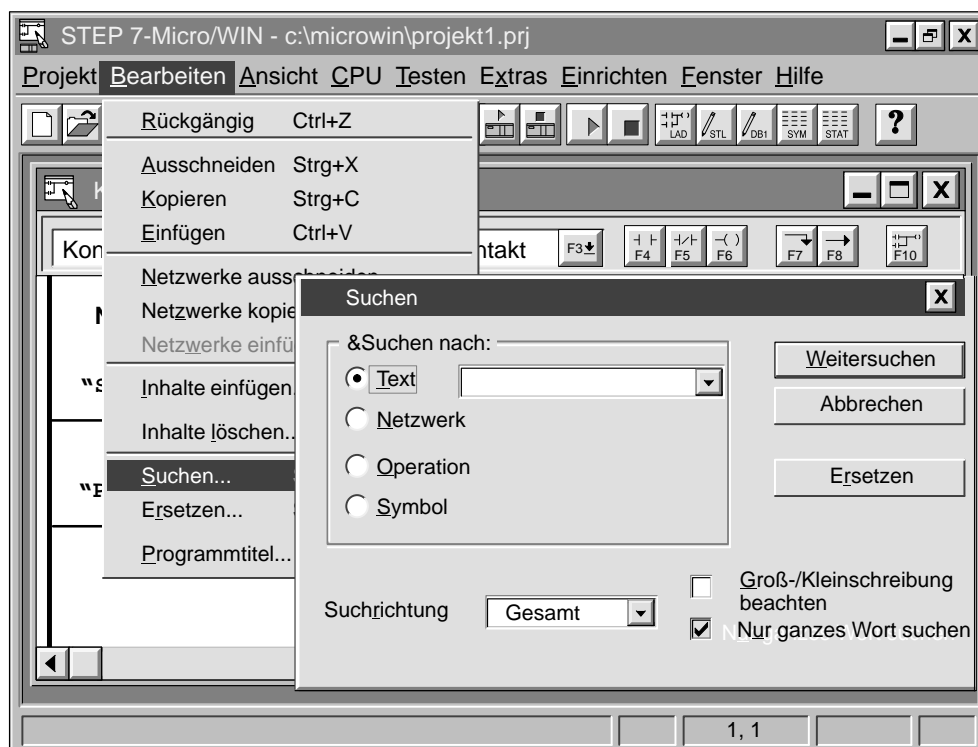


Bild 5-21 Dialogfeld "Suchen"

Ersetzen eines Parameters

Zum Ersetzen eines bestimmten Parameters gehen Sie folgendermaßen vor:

1. Wählen Sie den Menübefehl **Bearbeiten ► Ersetzen....** Bild 5-22 zeigt das Dialogfeld "Ersetzen".
2. Sie müssen den Parameter definieren, den Sie ersetzen möchten.
3. Mit der Schaltfläche "Ersetzen" ersetzen Sie einen gefundenen Parameter. Wenn Sie die Schaltfläche "Ersetzen" wählen, wird nach dem ersten Auftreten des Parameters gesucht. Wählen Sie die Schaltfläche "Ersetzen" erneut, um den Parameter zu ersetzen und nach dem nächsten zu suchen.
4. Wählen Sie die Schaltfläche "Alle ersetzen", werden angegebene Bereiche ignoriert und alle vorhandenen, angegebenen Parameters ersetzt.

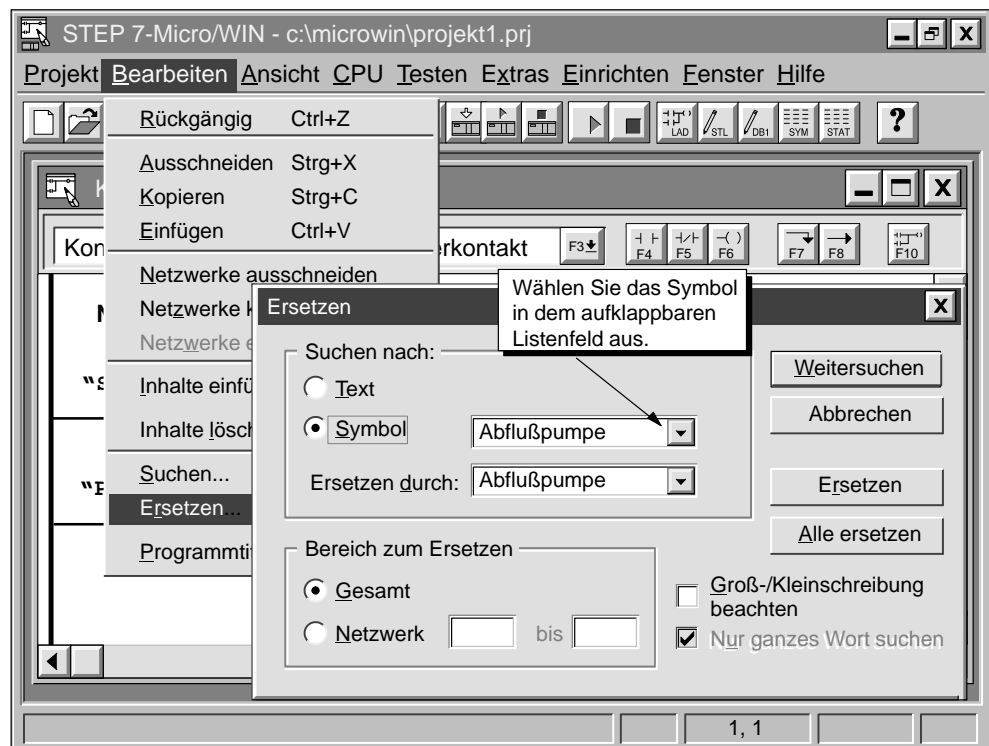


Bild 5-22 Dialogfeld "Ersetzen"

5.7 Dokumentieren Ihres Programms

Ihr KOP-Programm können Sie mit Programmtiteln, Netzwerktiteln und Netzwerkkommentaren dokumentieren. Ihr AWL-Programm können Sie mit beschreibenden Kommentaren dokumentieren.

Richtlinien zum Dokumentieren von KOP-Programmen

Der Titel des KOP-Programms dient dazu, eine kurze Beschreibung Ihres Projekts abzulegen. Zum Eingeben eines Titels für das Programm wählen Sie den Menübefehl **Bearbeiten ▶ Programmtitel...** Geben Sie den Programmtitel ein und bestätigen Sie mit "OK".

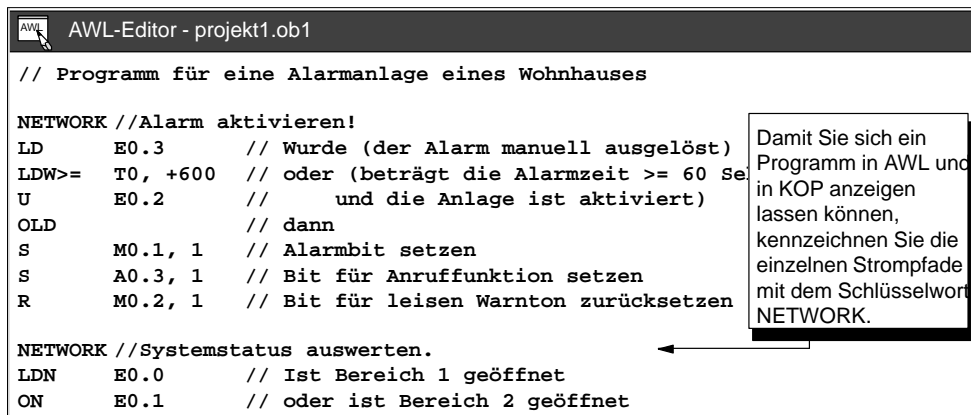
In den Netzwerktiteln in KOP können Sie kurz die Funktion des Netzwerks erläutern. Der einzeilige Netzwerktitel ist im KOP-Editor ständig sichtbar. Zum Bearbeiten des Netzwerktitels doppelklicken Sie auf das Feld des Netzwerktitels in Ihrem Programm. Geben Sie Ihre Beschreibung in das Feld "Titel" im Kommentar-Editor in KOP ein. Klicken Sie auf "OK".

In den Netzwerkkommentaren in KOP können Sie die Funktion des Netzwerks ausführlicher erläutern. Zum Eingeben von Netzwerkkommentaren doppelklicken Sie auf das Feld des Netzwerktitels in Ihrem Programm. Geben Sie Ihre Kommentare in das Feld "Kommentar" ein und bestätigen Sie mit "OK". Netzwerkkommentare sind im KOP-Editor nicht sichtbar, doch Sie können sie sich jederzeit anzeigen lassen, indem Sie auf das Feld des Netzwerktitels doppelklicken.

Zum Drucken Ihrer Netzwerkkommentare in KOP wählen Sie den Menübefehl **Projekt ▶ Drucken...** Wählen Sie die Schaltfläche "Seite einrichten", dann die Option "Netzwerkkommentare drucken" und bestätigen Sie mit "OK".

Richtlinien zum Dokumentieren von AWL-Programmen

Jeglicher Text in einer Zeile eines AWL-Programms, dem ein doppelter Schrägstrich (//) vorangestellt ist, wird als AWL-Kommentar betrachtet. Sie können auch am Anfang des Programms Kommentare einfügen, um die Funktion des Programms zu erläutern. Sie können auch ganze Zeilen nur mit Kommentaren einfügen oder Kommentare in Zeilen mit Anweisungen anhängen, um die Einzelheiten des Programms zu beschreiben (siehe Bild 5-23).



```

AWL-Editor - projekt1.ob1
// Programm für eine Alarmanlage eines Wohnhauses

NETWORK //Alarm aktivieren!
LD    E0.3      // Wurde (der Alarm manuell ausgelöst)
LDW>= T0, +600 // oder (beträgt die Alarmzeit >= 60 Se
U     E0.2      //      und die Anlage ist aktiviert)
OLD
S     M0.1, 1   // Alarmbit setzen
S     A0.3, 1   // Bit für Anruffunktion setzen
R     M0.2, 1   // Bit für leisen Warnton zurücksetzen

NETWORK //Systemstatus auswerten.
LDN   E0.0      // Ist Bereich 1 geöffnet
ON    E0.1      // oder ist Bereich 2 geöffnet
  
```

Damit Sie sich ein Programm in AWL und in KOP anzeigen lassen können, kennzeichnen Sie die einzelnen Strompfade mit dem Schlüsselwort NETWORK.

Bild 5-23 Dokumentieren Ihres AWL-Programms

Anzeigen von AWL-Programmen in KOP

Wenn Sie beabsichtigen, sich Ihr AWL-Programm in KOP anzeigen zu lassen, sollten Sie beim Erstellen des Programms in AWL folgende Richtlinien beachten (siehe Bild 5-23):

- Sie müssen die Strompfade im Code in AWL in einzelne Netzwerke unterteilen, indem Sie das Schlüsselwort "NETWORK" eingeben. Das Schlüsselwort "NETWORK" müssen Sie in angemessenen Abständen eingeben, damit das Programm in KOP angezeigt werden kann. Die Netzwerknummern werden beim Übersetzen bzw. Laden des Programms automatisch erzeugt.
- Die Kommentarzeilen in AWL, die sich vor dem ersten Schlüsselwort "NETWORK" befinden, stellen in KOP den Programmtitel dar.
- Ein AWL-Kommentar, der sich in der Zeile unterhalb des Schlüsselworts "NETWORK" befindet, wird in KOP zum Netzwerktitel.
- AWL-Kommentare, die sich zwischen der Zeile "NETWORK" und der ersten Anweisung des Netzwerks befinden, werden in KOP zu Netzwerkkommentaren. Hierzu ein Beispiel:

```
NETWORK // NETZWERKTITEL
//NETZWERKKOMMENTAR ZEILE 1
//NETZWERKKOMMENTAR ZEILE 2
LD E0.0
```

5.8 Drucken Ihres Programms

Mit der Druckfunktion können Sie Ihr komplettes Programm oder nur einzelne Teile des Programms drucken.

- Wählen Sie den Menübefehl **Projekt ► Drucken...**, um Ihr Programm zu drucken. Geben Sie an, was Sie drucken möchten und bestätigen Sie mit "OK" (siehe Bild 5-24).
- Klicken Sie auf die Schaltfläche "Seite einrichten", wenn Sie zusätzliche Druckoptionen einstellen möchten: Randeinstellungen, absolute/symbolische Adressen, Netzwerkkommentare sowie Kopf- und Fußzeilen.
- Wählen Sie die Schaltfläche "Drucker einrichten", um Ihren Drucker und das Papierformat anzugeben.

Zum Drucken Ihres Programms gehen Sie folgendermaßen vor:

1. Wählen Sie den Menübefehl **Projekt ► Drucken...**. Daraufhin wird das Dialogfeld "Drucken" angezeigt (siehe Bild 5-24).
2. Geben Sie in dem Feld "Druckbereich" an, was Sie drucken möchten.
3. Geben Sie in dem Feld "Netzwerkbereich in KOP" an, welche Netzwerke Sie drucken möchten.
4. Müssen Sie Ihren Drucker neu einrichten, wählen Sie eine der beiden Schaltflächen "Drucker einrichten" oder "Seite einrichten".
5. Bestätigen Sie mit "OK".

Hinweis

Wenn Sie die Querverweise und/oder die verwendeten Elemente drucken möchten, kann es sein, daß Sie Ihr Programm zunächst übersetzen müssen. Wie lange es dauert, bis Ihr Programm übersetzt ist, richtet sich nach der Größe Ihres Programms.

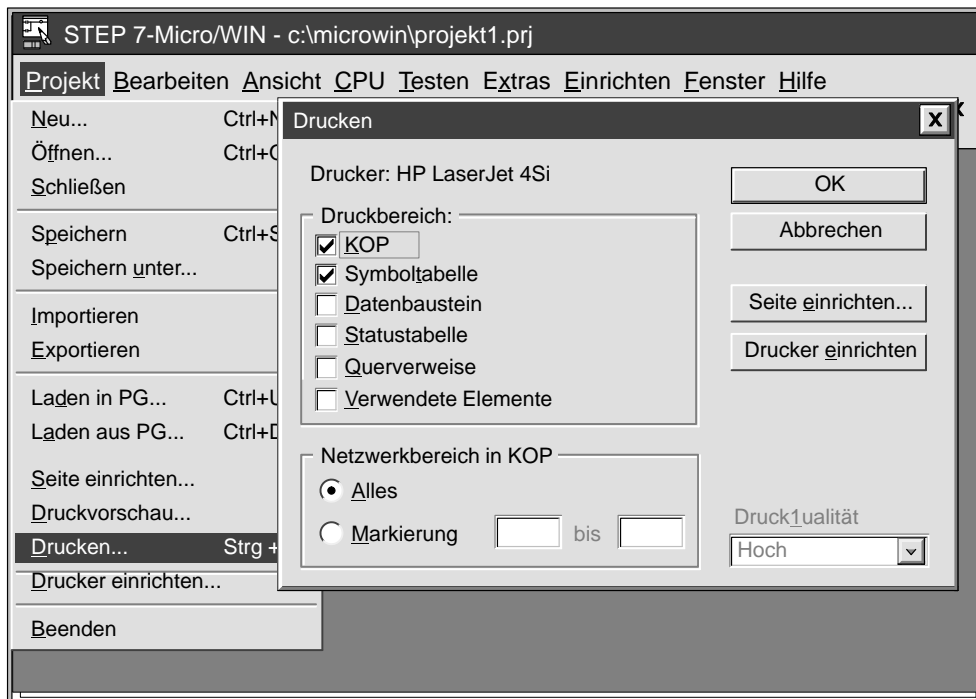


Bild 5-24 Dialogfeld "Drucken"

Grundlegendes zum Programmieren einer S7-200 CPU

6

Bevor Sie mit dem Programmieren von Anwendungen für Ihre S7-200 CPU beginnen, sollten Sie sich mit einigen wesentlichen Funktionalitäten der CPU vertraut machen.

Kapitelübersicht

Abschnitt	Beschreibung	Seite
6.1	Richtlinien für das Entwerfen einer Automatisierungslösung mit einer Micro-SPS	6-2
6.2	S7-200 Programme	6-4
6.3	S7-200 Programmiersprachen	6-5
6.4	Grundlegende Elemente zum Entwerfen eines Programms	6-8
6.5	Der Zyklus der CPU	6-10
6.6	Einstellen der Betriebsart für die CPU	6-13
6.7	Einrichten eines Paßworts für die CPU	6-14
6.8	Testen und Überwachen Ihres Programms	6-16
6.9	Fehlerbehebung bei der S7-200 CPU	6-19

6.1 Richtlinien für das Entwerfen einer Automatisierungslösung mit einer Micro-SPS

Es gibt viele Methoden, ein Automatisierungssystem zu entwerfen. Dieser Abschnitt erläutert Ihnen einige Grundregeln, die Sie in jedem Projekt anwenden können. Dabei sollten Sie sich selbstverständlich an die Verfahrensanweisungen in Ihrem Unternehmen halten und Ihre eigenen Erfahrungen berücksichtigen. Bild 6-1 zeigt einige der wesentlichen Schritte beim Entwerfen eines Automatisierungssystems.

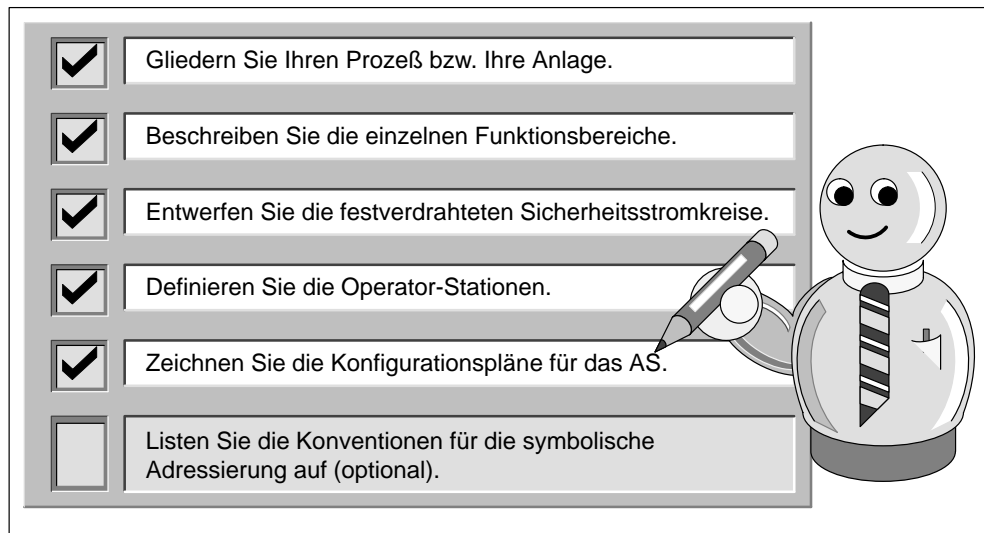


Bild 6-1 Wesentliche Schritte beim Entwerfen eines Automatisierungssystems

Gliedern des Prozesses bzw. der Anlage

Unterteilen Sie Ihren Prozeß bzw. Ihre Anlage in Abschnitte, die voneinander unabhängig sind. Diese Abschnitte legen die Grenzen zwischen mehreren Automatisierungssystemen fest und beeinflussen die Beschreibungen der Funktionsbereiche sowie die Zuordnung der Betriebsmittel.

Beschreiben der Funktionsbereiche

Beschreiben Sie die Funktionsweise jedes Prozeß- bzw. Anlagenabschnitts. Zeichnen Sie folgende Punkte auf:

- Eingänge/Ausgänge (E/A)
- Beschreibung der Funktionsweise
- Freigabebedingungen (d. h. Zustände, die erreicht sein müssen, bevor eine Bedienung möglich wird) für jeden Aktor (Magnetschalter, Motoren, Antriebe usw.)
- Beschreibung des Bedien- und Beobachtungssystems
- Schnittstellen zu anderen Prozeß- bzw. Anlagenabschnitten

Entwerfen der Sicherheitsstromkreise

Bestimmen Sie die Geräte, die aus Sicherheitsgründen festverdrahtete Schaltungen benötigen. Steuerungsgeräte können unsichere Betriebszustände einnehmen, woraus unerwartete Anlaufeigenschaften bzw. geänderte Funktionsabläufe der Anlage resultieren können. Besteht die Gefahr, daß bei unerwartetem bzw. fehlerhaftem Betrieb der Anlage schwere Körperverletzungen oder Sachschäden auftreten, sollten Sie mit elektromechanischen Programmeeingriffen, die unabhängig von der CPU arbeiten, unsichere Betriebszustände vermeiden.

Zum Entwerfen von Sicherheitsstromkreisen gehen Sie folgendermaßen vor:

- Definieren Sie falschen bzw. unerwarteten Betrieb von Aktoren, die Gefahrenpotentiale bergen.
- Definieren Sie die Bedingungen, unter denen der Betrieb ungefährlich ist, und legen Sie fest, wie diese Bedingungen unabhängig von der CPU erkannt werden.
- Definieren Sie, wie die CPU und die Erweiterungsmodule den Prozeß beeinflussen, wenn die Spannung eingeschaltet und wieder ausgeschaltet wird und wenn Fehler erkannt werden. Diese Informationen sollten nur zum Entwerfen des normalen und des erwarteten unnormalen Betriebs verwendet werden und dürfen nicht aus Sicherheitsgründen eingesetzt werden.
- Entwerfen Sie Korrekturen durch Handeingriff bzw. elektromechanische Programmeingriffe, mit denen gefährliche Abläufe unabhängig von der CPU gesperrt werden.
- Lassen Sie von den unabhängigen Stromkreisen Statusinformationen an die CPU übermitteln, so daß das Programm und jede Bedienerschnittstelle über die erforderlichen Informationen verfügt.
- Definieren Sie weitere Sicherheitsanforderungen, damit der Prozeß sicher ablaufen kann.

Definieren der Operator-Stationen

Erstellen Sie Pläne der Operator-Stationen anhand der Anforderungen in den Beschreibungen der Funktionsbereiche. Zeichnen Sie folgende Punkte auf:

- Position aller Operator-Stationen in bezug zum Prozeß bzw. zur Anlage
- Mechanische Anordnung der Geräte (Display, Schalter, Lampen usw.) der Operator-Station
- Verdrahtungspläne mit den zugehörigen Ein- und Ausgängen der CPU oder der Erweiterungsmodule

Zeichnen der Konfigurationspläne für das AS

Erstellen Sie die Konfigurationspläne für das Automatisierungssystem anhand der Anforderungen in den Beschreibungen der Funktionsbereiche. Zeichnen Sie folgende Punkte auf:

- Übersicht über die Position aller CPUs in bezug zum Prozeß bzw. zur Anlage
- Mechanische Anordnung der CPU und Erweiterungsmodule (einschließlich Schränke usw.)
- Verdrahtungspläne für alle CPUs und Erweiterungsmodule (einschließlich Gerätenummern, Kommunikationsadressen und Adressen der Ein- und Ausgänge)

Erstellen einer Liste mit symbolischen Adressen

Wenn Sie sich für die symbolische Adressierung entscheiden, müssen Sie den absoluten Adressen symbolische Namen zuordnen. Geben Sie nicht nur die physikalischen Ein- und Ausgänge, sondern auch alle anderen Elemente an, die Sie in Ihrem Programm verwenden.

6.2 S7-200 Programme

Verweise im Programm auf Ein- und Ausgänge

Die grundlegende Funktionsweise der S7-200 CPU ist sehr einfach:

- Die CPU liest den Signalzustand der Eingänge.
- Das Programm, das in der CPU gespeichert ist, wertet mit Hilfe der Eingänge die Logik aus. Während der Bearbeitung des Programms aktualisiert die CPU die Daten.
- Die CPU schreibt die Daten in die Ausgänge.

Bild 6-2 zeigt die Verbindung zwischen einem einfachen Schaltplan und der S7-200 CPU. In diesem Beispiel wird der Signalzustand des Schalters auf der Operator-Station, der den Abfluß öffnet, zu dem Zustand anderer Eingänge addiert. Die Berechnungen dieser Zustände bestimmen dann den Signalzustand des Ausgangs für den Magnetschalter, der den Abfluß schließt.

Die CPU bearbeitet das Programm zyklisch, sie liest und schreibt Daten.

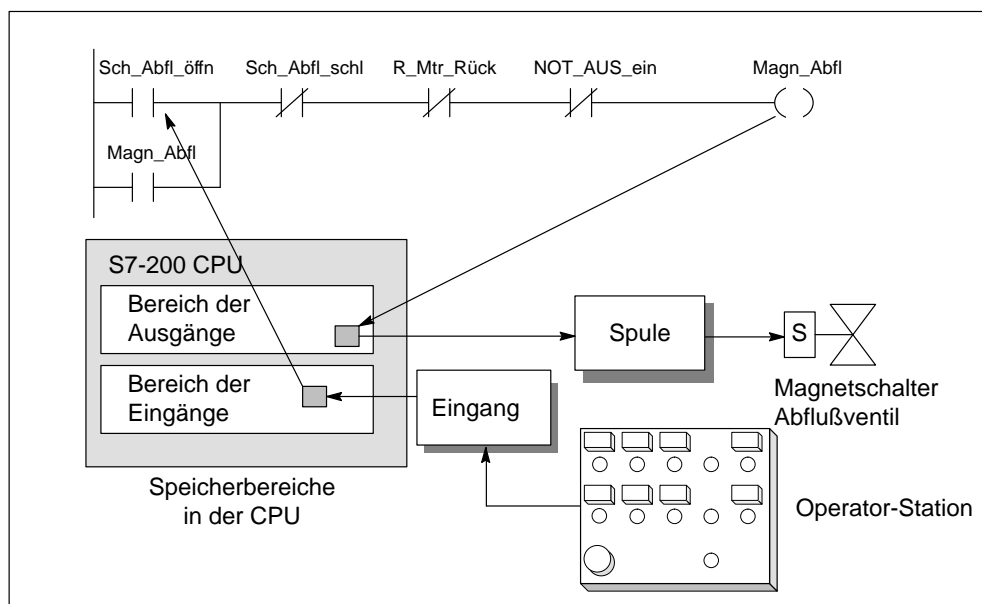


Bild 6-2 Verweise im Programm auf Ein- und Ausgänge

Zugreifen auf Daten in den Speicherbereichen

Die CPU speichert den Signalzustand von Ein- und Ausgängen in bestimmten Speicherbereichen. Bild 6-2 zeigt den vereinfachten Informationsfluß: Eingang → Speicherbereich → Programm → Speicherbereich → Ausgang. Jedem Speicherbereich ist eine Kennung (Mnemonic) zugeordnet (beispielsweise "E" für Eingang und "A" für Ausgang), über die auf die Daten in dem jeweiligen Speicherbereich zugegriffen wird.

STEP 7-Micro/WIN stellt für alle Speicherbereiche "absolute" Adressen zur Verfügung. Sie greifen auf bestimmte Adressen zu, indem Sie einen Operanden angeben (beispielsweise $\text{E}0.0$ für den ersten Eingang). In STEP 7-Micro/WIN können Sie den absoluten Adressen auch symbolische Namen zuordnen. Eine absolute Adresse eines Speicherbereichs besteht nicht nur aus der Bereichskennung (z.B. "V"), sondern auch aus der Größe (maximal vier Bytes bzw. 32 Bits) der Daten, auf die zugegriffen werden soll: B (Byte), W (Wort bzw. zwei Bytes) oder D (Doppelwort bzw. 4 Bytes). Die absolute Adresse umfaßt außerdem einen numerischen Wert. Dies ist entweder die Anzahl der Bytes vom Beginn des Speicherbereichs (Versatz) oder die Nummer des Elements (dieser Wert richtet sich nach der Bereichskennung (siehe Abschnitt 7.1).

6.3 S7-200 Programmiersprachen

Die S7-200 CPU (und STEP 7-Micro/WIN) unterstützt die folgenden Programmiersprachen:

- Eine Anweisungsliste (AWL) besteht aus mehreren Operationen, deren Mnemonik eine Funktion der CPU darstellt.
- Ein Kontaktplan (KOP) ist eine graphische Programmiersprache, die elektrischen Schaltplänen ähnelt.

STEP 7-Micro/WIN verfügt außerdem über zwei Darstellungsarten zum Anzeigen von Adressen und Operationen in einem Programm: International und SIMATIC. Beide Darstellungsarten, International und SIMATIC, beziehen sich auf denselben Operationssatz für die S7-200. Zwischen beiden Darstellungsarten gibt es eine direkte Übereinstimmung und die Funktionalität ist bei beiden Darstellungsarten identisch.

Grundlegende Elemente eines Kontaktplans

Wenn Sie ein Programm im Kontaktplan entwerfen, dann arbeiten Sie mit graphischen Komponenten, mit denen Sie die Netzwerke Ihrer Logik aufbauen. Die folgenden Elemente können Sie zum Erstellen Ihres Programms verwenden (siehe Bild 6-3):

- Kontakte: Jeder Kontakt stellt einen Schalter dar, durch den im geschlossenen Zustand Strom fließen kann.
- Spulen: Jede Spule stellt ein Relais dar, das durch Signalfuß stromführend wird.
- Boxen: Jede Box stellt eine Funktion dar, die ausgeführt wird, wenn Strom zur Box fließt.
- Netzwerke: Ein Netzwerk bildet einen vollständigen Stromkreis. Strom fließt von der linken Stromschiene über die geschlossenen Kontakte zu den Spulen oder Boxen, die dadurch aktiviert werden.

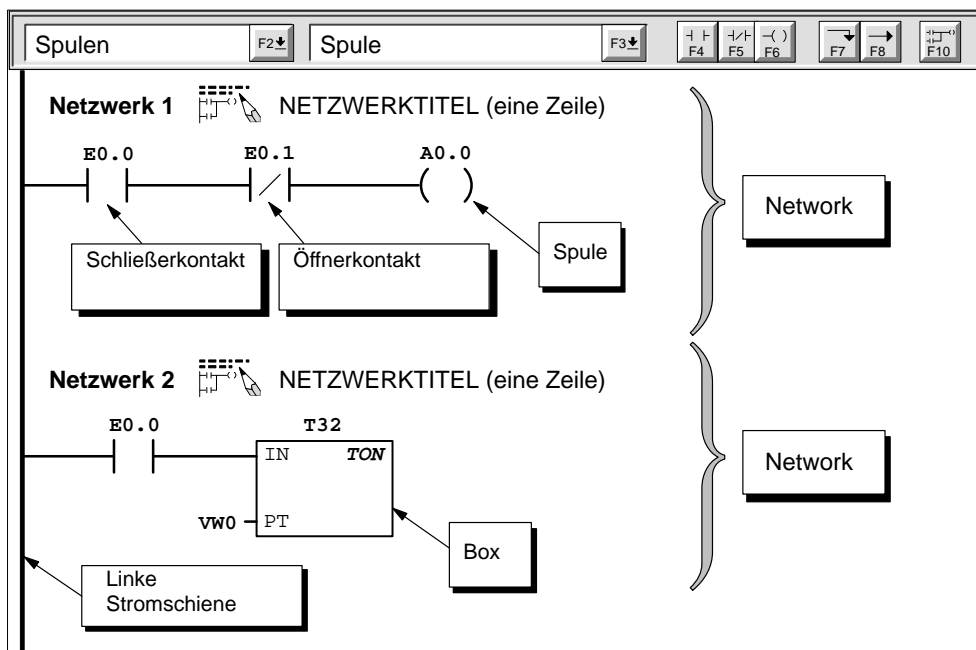


Bild 6-3 Grundlegende Elemente eines Kontaktplans

Anweisungen einer Anweisungsliste

Die Anweisungsliste ist eine Programmiersprache, in der jede Anweisung in Ihrem Programm eine Operation enthält, deren Mnemonik eine Funktion der CPU darstellt. Sie verbinden diese Operationen zu einem Programm, um so die Steuerung Ihrer Anwendung zu erstellen.

Bild 6-4 zeigt die grundlegenden Elemente eines Programms in der Anweisungsliste.

```

AWL-Editor - projekt1.ob1
//Programm Förderband

NETWORK //Motor starten:
LD "Start1" //Ist E0.0 eingeschaltet
UN "NOT-AUS1" //und E0.1 ist nicht eingeschaltet,
= A0.0 //dann Motor für Förderband einschalten.

NETWORK //NOT_AUS für Förderband:
LD E0.1 //Ist NOT_AUS1 eingeschaltet
O E0.3 //oder NOT_AUS2 ist eingeschaltet,
R A0.0, 1 //dann Motor für Förderband ausschalten.

NETWORK //Ende des Programms
MEND
    
```

Bild 6-4 Fenster des AWL-Editors mit einem Beispielprogramm

Die AWL-Operationen arbeiten mit einem logischen Stack in der CPU, um die Logik zu lösen. Dieser logische Stack ist neun Bit tief und ein Bit breit (siehe Bild 6-5). Die meisten AWL-Operationen arbeiten entweder mit dem ersten Bit oder mit dem ersten und dem zweiten Bit des Stack. Neue Werte können in den Stack geschoben bzw. ergänzt werden. Werden die beiden obersten Bits des Stack verknüpft, wird der Stack um ein Bit verringert.

Während die meisten AWL-Operationen den Wert im logischen Stack nur lesen, ändern einige AWL-Operationen die im Stack gespeicherten Werte. Bild 6-5 zeigt drei Beispiele dafür, wie einige Operationen mit dem Stack arbeiten.

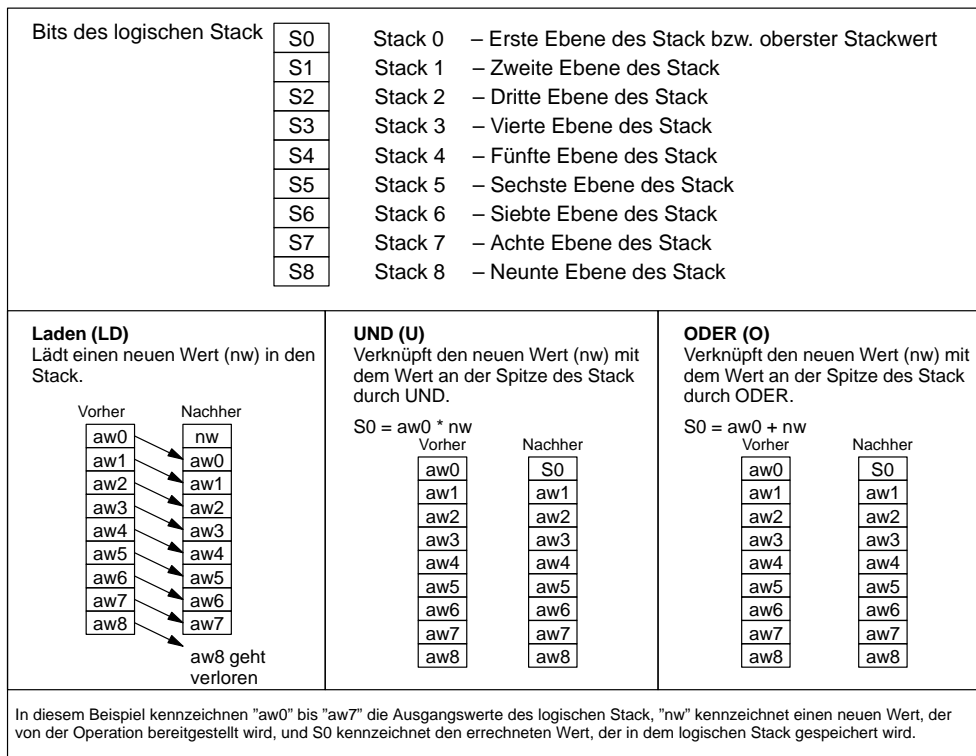


Bild 6-5 Logischer Stack der S7-200 CPU

6.4 Grundlegende Elemente zum Entwerfen eines Programms

Die S7-200 CPU bearbeitet Ihr Programm fortlaufend, um eine Aufgabe oder einen Prozeß zu steuern. Das Programm erstellen Sie mit STEP 7-Micro/WIN und laden es anschließend in die CPU. Aus dem Hauptprogramm können Sie verschiedene Unterprogramme und Interruptprogramme aufrufen.

Strukturieren des Programms

Programme für eine S7-200 CPU setzen sich aus drei Grundelementen zusammen: einem Hauptprogramm, Unterprogrammen (optional) und Interruptprogrammen (optional). In Bild 6-6 sehen Sie, daß ein S7-200 Programm in die folgenden Organisationseinheiten unterteilt ist:

- **Hauptprogramm:** In diesem Hauptteil des Programms ordnen Sie die Operationen an, die Ihre Anwendung steuern. Die Operationen des Hauptprogramms werden in jedem Zyklus der CPU der Reihe nach bearbeitet. Zum Beenden des Hauptprogramms verwenden Sie in KOP eine Spule zum absoluten Beenden des Programms und in AWL die Operation zum Beenden des Hauptprogramms (MEND) (siehe (1) in Bild 6-6).
- **Unterprogramme:** Diese optionalen Komponenten eines Programms werden nur dann bearbeitet, wenn sie aus dem Hauptprogramm aufgerufen werden. Ordnen Sie die Unterprogramme nach dem Ende des Hauptprogramms an (nach der Spule zum absoluten Beenden in KOP bzw. nach der Operation MEND in AWL). Beenden Sie jedes Unterprogramm mit der Operation RET (siehe (2) in Bild 6-6).
- **Interruptprogramme:** Diese optionalen Komponenten eines Programms werden nur bearbeitet, wenn ein Interruptereignis auftritt. Ordnen Sie die Interruptprogramme nach dem Ende des Hauptprogramms an (nach der Spule zum absoluten Beenden in KOP bzw. nach der Operation MEND in AWL). Beenden Sie jedes Interruptprogramm mit der Operation RETI, Interruptprogramm beenden (siehe (3) in Bild 6-6).

Unterprogramme und Interruptprogramme folgen auf die Spule zum absoluten Beenden in KOP bzw. auf die Operation MEND in AWL im Hauptprogramm. Es gibt keine weiteren Richtlinien, die Sie beim Anordnen der Unter- und Interruptprogramme in Ihrem Programm beachten müssen. Sie können Unterprogramme und Interruptprogramme im Anschluß an das Hauptprogramm in gemischter Reihenfolge anordnen. Wenn Sie Ihr Programm jedoch leicht lesbar und verständlich aufbauen möchten, sollten Sie alle Unterprogramme direkt an das Hauptprogramm anschließen und danach alle Interruptprogramme geschlossen nach den Unterprogrammen anordnen.

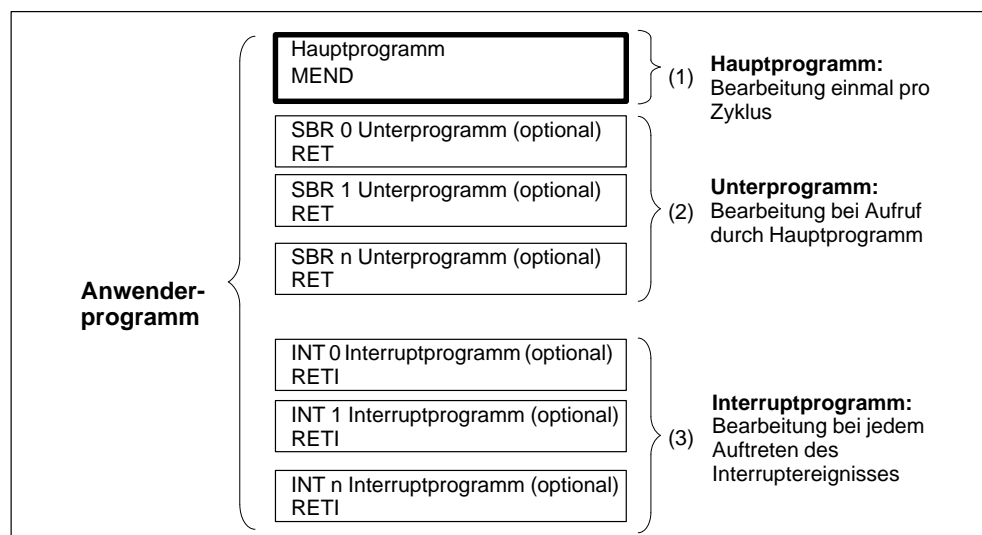


Bild 6-6 Aufbau eines Programms für die S7-200 CPU

Beispielprogramm mit Unterprogrammen und Interruptprogrammen

Bild 6-7 zeigt ein Beispielprogramm für einen zeitgesteuerten Interrupt, mit dem Sie den Wert eines Analogeingangs lesen können. In diesem Beispiel wird der Analogeingang alle 100 ms abgefragt.

Kontaktplan	AWL
Hauptprogramm	
<p>Netzwerk 1</p> <p>Netzwerk 2</p>	<pre> Network 1 LD SM0.1 //Wird das Bit des //ersten Zyklus //eingeschaltet, CALL 0 //Unterprogramm 0 //aufrufen. Network 2 MEND </pre>
Unterprogramme	
<p>Netzwerk 3</p> <p>Netzwerk 4</p> <p>Netzwerk 5</p>	<pre> Network 3 SBR 0 //Unterprogramm 0 //beginnen Network 4 LD SM0.0 //Merker ständig EIN MOVB 100, SMB34 //Intervall des //zeitgesteuerten //Interrupt auf 100 ms //setzen ENI //Alle Interruptereig- //nisse freigeben ATCH 0, 10 //Zeitgesteuerten //Interrupt 0 dem //Interruptprogramm 0 //zuordnen Network 5 RET //Unterprogramm beenden </pre>
Interruptprogramme	
<p>Netzwerk 6</p> <p>Netzwerk 7</p> <p>Netzwerk 8</p>	<pre> Network 6 INT 0 //Interruptprogramm 0 //beginnen Network 7 MOVW AEW4, VW100 //AEW4 abfragen Network RETI //Interruptprogramm //beenden </pre>

Bild 6-7 Beispielprogramm mit Unterprogrammen und Interruptprogrammen

6.5 Der Zyklus der CPU

Die S7-200 CPU bearbeitet eine Reihe von Aufgaben, einschließlich Ihr Programm, zyklisch. Diese zyklische Bearbeitung der Aufgaben wird Zyklus genannt. Während des Zyklus (siehe Bild 6-8) führt die CPU die meisten bzw. alle der folgenden Aufgaben aus:

- Lesen der Eingänge
- Bearbeiten des Programms
- Bearbeiten von Kommunikationsanforderungen
- Selbstdiagnose der CPU durchführen
- Schreiben in die Ausgänge

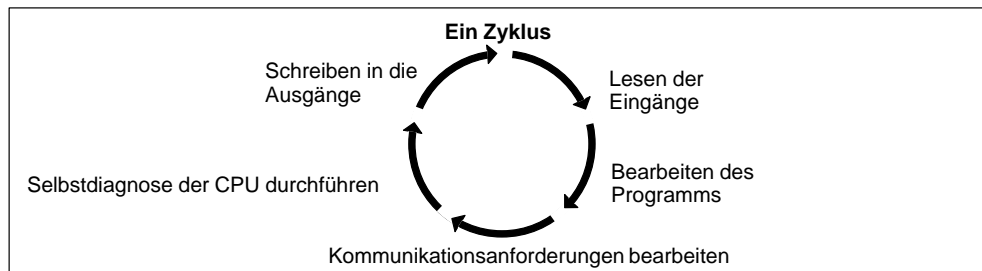


Bild 6-8 Zyklus der S7-200 CPU

Die Aufgaben, die während eines Zyklus ausgeführt werden, richten sich nach dem Betriebszustand der CPU. Eine S7-200 CPU verfügt über zwei Betriebszustände, den Betriebszustand STOP und den Betriebszustand RUN. In bezug auf den Zyklus liegt der wesentliche Unterschied zwischen den Betriebszuständen STOP und RUN darin, daß im Betriebszustand RUN Ihr Programm bearbeitet wird, während es im Betriebszustand STOP nicht bearbeitet wird.

Lesen der Digitaleingänge

Am Anfang eines Zyklus werden die aktuellen Werte der Digitaleingänge gelesen und anschließend in das Prozeßabbild der Eingänge geschrieben.

Für die CPU sind im Prozeßabbild der Eingänge Abschnitte mit jeweils acht Bit (ein Byte) vorgesehen. Stellt die CPU oder ein Erweiterungsmodul nicht zu jedem Bit des reservierten Byte einen physikalischen Eingang zur Verfügung, dann können Sie diese Bits nicht den nachfolgenden Modulen in der E/A-Kette zuordnen und sie auch nicht in Ihrem Programm verwenden. Die freien Eingänge im Prozeßabbild werden von der CPU am Anfang eines jeden Zyklus auf Null gesetzt. Kann Ihre CPU jedoch um mehrere Erweiterungsmodule ergänzt werden und Sie schöpfen diese Kapazität nicht vollständig aus (wenn Sie nicht die maximale Anzahl Erweiterungsmodule installiert haben), können Sie die freien Eingangsbits, die für die Erweiterungsmodule vorgesehen sind, als zusätzliche Merker verwenden.

Die CPU aktualisiert die Analogeingänge nicht automatisch als Teil des Zyklus und sie legt für die Analogeingänge auch kein Abbild im Speicher an. Auf die Analogeingänge müssen Sie direkt über ihr Programm zugreifen.

Bearbeiten des Programms

Während dieses Abschnitts im Zyklus bearbeitet die CPU das Programm von der ersten Operation bis zur Endeoperation. Sie können die Ein- und Ausgänge direkt ansteuern und so auf sie zugreifen, während das Programm oder ein Interruptprogramm bearbeitet wird.

Wenn Sie in Ihrem Programm Interrupts verwenden, dann werden die Interruptprogramme, die den Interruptereignissen zugeordnet sind, als Teil des Hauptprogramms gespeichert (siehe Abschnitt 6.4). Die Interruptprogramme werden jedoch nicht als Bestandteil des normalen Zyklus bearbeitet, sondern nur dann, wenn ein Interruptereignis auftritt (dies ist an jeder Stelle des Zyklus möglich).

Bearbeiten der Kommunikationsanforderungen

Während dieses Abschnitts im Zyklus bearbeitet die CPU alle Meldungen, die von den Kommunikationsschnittstellen empfangen wurden.

Durchführen der Selbstdiagnose in der CPU

Während dieses Abschnitts im Zyklus prüft die CPU die Firmware und den Programmspeicher (nur im Betriebszustand RUN). Außerdem wird der Zustand der Erweiterungsmodule geprüft.

Schreiben in die Digitalausgänge

Am Ende des Zyklus werden die Werte aus dem Prozeßabbild der Ausgänge in die Digitalausgänge geschrieben.

Für die CPU sind im Prozeßabbild der Eingänge Abschnitte mit jeweils acht Bit (ein Byte) vorgesehen. Stellt die CPU oder ein Erweiterungsmodul nicht zu jedem Bit des reservierten Byte einen physikalischen Ausgang zur Verfügung, dann können Sie diese Bits nicht den nachfolgenden Modulen in der E/A-Kette zuordnen. Sie können die freien Bits im Prozeßabbild der Ausgänge jedoch als zusätzliche interne Merker verwenden.

Die CPU aktualisiert die Analogausgänge nicht automatisch als Teil des Zyklus und sie legt für die Analogausgänge auch kein Abbild im Speicher an. Auf die Analogausgänge müssen Sie direkt über ihr Programm zugreifen.

Wird der Betriebszustand der CPU von RUN in STOP versetzt, dann werden die Digitalausgänge entweder auf die Werte gesetzt, die Sie in den Einstellungen der Ausgänge definiert haben, oder sie bleiben in ihrem derzeitigen Zustand (siehe Abschnitt 8.3). Analogausgänge behalten den zuletzt geschriebenen Wert.

Unterbrechen des Zyklus

Wenn Sie in Ihrem Programm Interrupts verwenden, dann werden die Interruptprogramme, die den Interruptereignissen zugeordnet sind, als Teil des Hauptprogramms gespeichert. Die Interruptprogramme werden jedoch nicht als Bestandteil des normalen Zyklus bearbeitet, sondern nur dann, wenn ein Interruptereignis auftritt (dies ist an jeder Stelle des Zyklus möglich). Interrupts werden von der CPU in der Reihenfolge ihres Auftretens und unter Beachtung der jeweiligen Prioritäten abgearbeitet.

Prozeßabbilder der Ein- und Ausgänge

Es ist normalerweise empfehlenswert, mit den Prozeßabbildern zu arbeiten und während der Bearbeitung des Programms nicht direkt auf die Ein- und Ausgänge zuzugreifen. Es gibt drei wesentliche Gründe für das Verwenden der Prozeßabbilder:

- Das System fragt zu Beginn eines Zyklus die Eingänge ab. Dadurch werden die Werte dieser Eingänge für die Dauer der Programmbearbeitung synchronisiert und "eingefroren". Die Ausgänge werden nach der Bearbeitung des Programms über das Prozeßabbild aktualisiert. Daraus ergibt sich eine stabilisierende Wirkung auf das System.
- Ihr Programm kann sehr viel schneller auf das Prozeßabbild als direkt auf die Ein- und Ausgänge zugreifen. Dies beschleunigt die Bearbeitung des Programms.
- Ein- und Ausgänge sind Biteinheiten, auf die im Bitformat zugegriffen werden muß. Auf die Prozeßabbilder können Sie jedoch im Bit-, Byte, Wort- und Doppelwortformat zugreifen. Deshalb bieten die Prozeßabbilder zusätzliche Flexibilität.

Ein weiterer Vorteil ist, daß die Prozeßabbilder groß genug sind, um die maximale Anzahl Ein- und Ausgänge zu bearbeiten. Da ein "echtes" System aus Ein- und Ausgängen besteht, gibt es im Prozeßabbild immer einige Adressen, die nicht verwendet werden. Diese freien Adressen können Sie als zusätzliche interne Merker verwenden (siehe Abschnitt 8.1).

Direktes Ansteuern der Ein- und Ausgänge

Mit den Operationen zum direkten Ansteuern der Ein- und Ausgänge können Sie direkt auf einen Ein- bzw. Ausgang zugreifen, auch wenn im allgemeinen die Prozeßabbilder als Quelle und Ziel von Zugriffen auf die Ein- und Ausgänge gelten. Wenn Sie auf einen Eingang direkt zugreifen, dann wird die entsprechende Adresse im Prozeßabbild der Eingänge nicht verändert. Wenn Sie auf einen Ausgang direkt zugreifen, dann wird gleichzeitig die entsprechende Adresse im Prozeßabbild der Ausgänge aktualisiert.

6.6 Einstellen der Betriebsart für die CPU

Die S7-200 CPU verfügt über zwei Betriebsarten:

- STOP: Die CPU bearbeitet das Programm nicht. Im Betriebszustand STOP können Sie ein Programm in die CPU laden und die CPU konfigurieren.
- RUN: Die CPU bearbeitet das Programm. Im Betriebszustand RUN können Sie kein Programm in die CPU laden und Sie können die CPU auch nicht konfigurieren.

Die Statusanzeigen auf der Vorderseite der CPU geben den aktuellen Betriebszustand an. Wenn Sie ein Programm in den Programmspeicher laden möchten, müssen Sie die CPU in die Betriebsart STOP versetzen.

Einstellen der Betriebsart mit dem Betriebsartenschalter

Mit dem Betriebsartenschalter (befindet sich unter der Abdeckklappe an der CPU) können Sie den Betriebszustand der CPU manuell einstellen:

- Wenn Sie den Betriebsartenschalter auf STOP stellen, wird die Bearbeitung des Programms angehalten.
- Wenn Sie den Betriebsartenschalter auf RUN stellen, wird die Bearbeitung des Programms aufgenommen.
- Wenn Sie den Betriebsartenschalter auf TERM stellen, wird der Betriebszustand der CPU nicht geändert, doch die Programmiersoftware (STEP 7-Micro/WIN) kann jetzt die Betriebsarten der CPU steuern.

Befindet sich der Betriebsartenschalter in einer der Positionen STOP oder TERM und wird die Spannungsversorgung unterbrochen, dann geht die CPU bei Spannungsrückkehr automatisch in den Betriebszustand STOP. Befindet sich der Betriebsartenschalter in der Stellung RUN, wenn die Spannungsversorgung unterbrochen wird, dann geht die CPU bei Spannungsrückkehr wieder in den Betriebszustand RUN.

Einstellen der Betriebsart mit STEP 7-Micro/WIN

Sie können den Betriebszustand der CPU auch mit STEP 7-Micro/WIN einstellen (siehe Bild 6-9). Damit die Software den Betriebszustand steuern kann, müssen Sie den Betriebsartenschalter auf der CPU in eine der Positionen TERM oder RUN bringen.

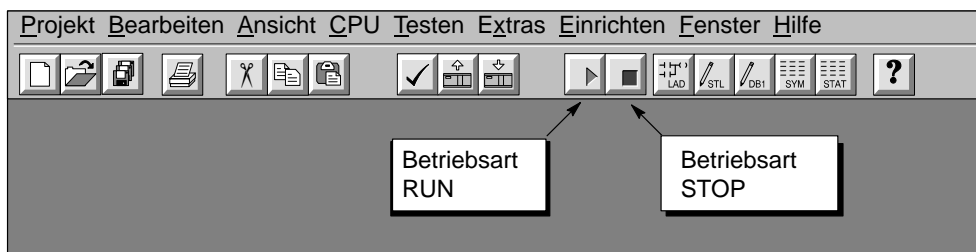


Bild 6-9 Einstellen der Betriebsart mit STEP 7-Micro/WIN

Einstellen der Betriebsart mit dem Programm

Sie können in Ihrem Programm die Operation STOP aufnehmen, die die CPU in den Betriebszustand STOP versetzt. Auf diese Weise können Sie in Abhängigkeit von Ihrer Logik die Programmbearbeitung unterbrechen. Ausführliche Informationen zur Operation STOP entnehmen Sie dem Kapitel 10.

6.7 Einrichten eines Paßworts für die CPU

Alle CPU-Varianten der S7-200 bieten Paßwortschutz und schränken dadurch den Zugriff auf bestimmte CPU-Funktionen ein. Durch das Einrichten eines Paßworts können nur berechnete Personen auf bestimmte Funktionen und den Speicher der CPU zugreifen. Ohne Paßwort ist der uneingeschränkte Zugriff auf die CPU möglich. Besteht ein Paßwortschutz, läßt die CPU keine der Funktionen zu, die in der Konfiguration des Paßworts eingeschränkt sind.

Schutzstufen der CPU

Die S7-200 CPUs bieten drei verschiedene Schutzstufen mit unterschiedlichen Zugriffseinschränkungen auf die Funktionen der CPU (siehe Tabelle 6-1). Jede Schutzstufe läßt auch ohne Eingabe eines Paßworts den uneingeschränkten Zugriff auf bestimmte Funktionen zu. Bei allen drei Schutzstufen haben Sie, wenn Sie das korrekte Paßwort eingeben, Zugriff auf alle Funktionen der CPU. Die Voreinstellung für die S7-200 CPUs ist die Schutzstufe 1 (keine Einschränkungen).

Wenn Sie ein Netz-Paßwort eingeben, dann wirkt sich dieses Paßwort nicht auf den Paßwortschutz der CPU aus. Ist ein Anwender berechnigt, auf geschützte Funktionen der CPU zuzugreifen, so sind deshalb nicht auch andere Anwender berechnigt, auf diese Funktionen zuzugreifen. Es hat immer nur ein Anwender uneingeschränkten Zugriff auf die CPU.

Hinweis

Nachdem Sie das Paßwort eingegeben haben, bleibt die Schutzstufe des Paßworts nach Trennen des Programmiergeräts von der CPU maximal eine Minute lang wirksam.

Tabelle 6-1 Schutzstufen bei der S7-200 CPU

Funktion	Schutzstufe 1	Schutzstufe 2	Schutzstufe 3
Lesen und Schreiben von Anwenderdaten	Nicht eingeschränkt	Nicht eingeschränkt	Nicht eingeschränkt
Starten, Stoppen und Neustarten der CPU			
Lesen und Einstellen der Echtzeituhr			
Lesen der geforderten Daten der CPU			
Laden des Anwenderprogramms, der Daten und der Konfiguration aus der CPU			
Laden in die CPU		Paßwort erforderlich	
Löschen des Anwenderprogramms, der Daten und der Konfiguration ¹			
Forcen von Daten und Ausführen einer bestimmten Anzahl Zyklen			
Kopieren in das Speichermodul			

¹ Der Löschschatz wird von dem Paßwort "clearplc" übersteuert.

Einrichten eines Paßworts für die CPU

Ein Paßwort für die CPU richten Sie mit STEP 7-Micro/WIN ein. Wählen Sie den Menübefehl **CPU ► Konfigurieren** und öffnen Sie das Register "Paßwort" (siehe Bild 6-10). Geben Sie die gewünschte Schutzstufe ein und bestätigen Sie das Paßwort für die CPU.

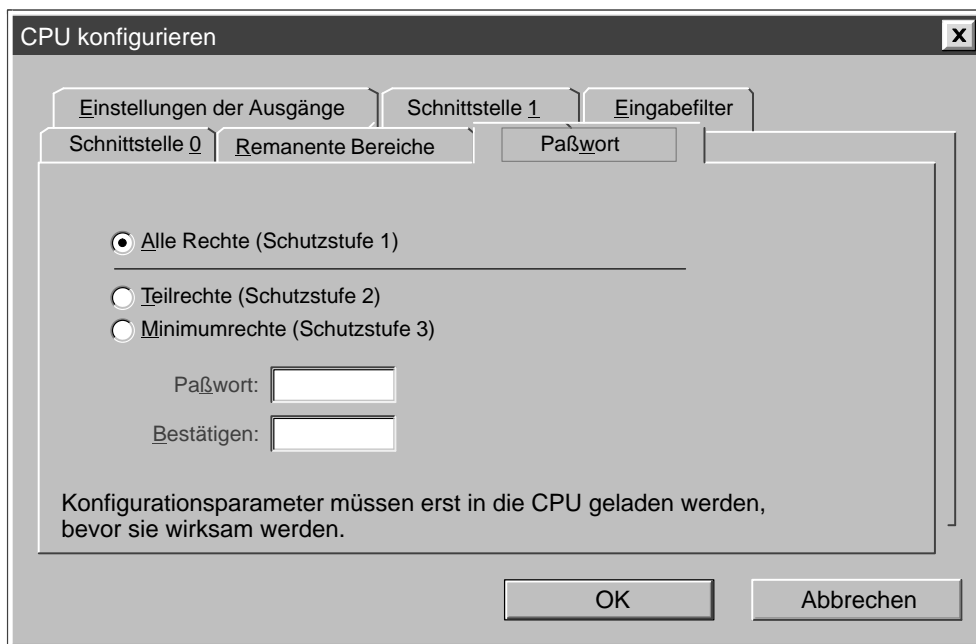


Bild 6-10 Einrichten eines Paßworts für die CPU

Maßnahmen bei vergessenem Paßwort

Wenn Sie Ihr Paßwort vergessen haben, müssen Sie den Speicher der CPU urlöschen und Ihr Programm erneut in die CPU laden. Beim Urlöschen des CPU-Speichers wird die CPU zunächst in den Betriebszustand STOP versetzt und anschließend auf die vom Werk voreingestellten Werte zurückgesetzt. Ausnahmen sind die Teilnehmeradresse und die Echtzeituhr der CPU, die nicht zurückgesetzt werden.

Zum Urlöschen Ihres Programms in der CPU wählen Sie den Menübefehl **CPU ► Urlöschen....** Daraufhin wird das Dialogfeld "Urlöschen" angezeigt. Wählen Sie die Option "Alles" und bestätigen Sie Ihre Eingaben mit "OK". Daraufhin wird ein Dialogfeld angezeigt, in dem die Paßwort-Berechtigung abgefragt wird. Wenn Sie das Paßwort "clearplc" (AS urlöschen) eingeben, können Sie den gesamten Speicher der CPU urlöschen.

Beim Urlöschen wird das Programm im Speichermodul nicht gelöscht. Da in dem Speichermodul neben dem Programm auch das zugeordnete Paßwort gespeichert ist, müssen Sie das Speichermodul erneut programmieren, um das vergessene Paßwort zu löschen.



Warnung

Beim Urlöschen der CPU werden die Ausgänge ausgeschaltet (Analogausgänge werden mit einem bestimmten Wert eingefroren).

Ist die S7-200 CPU beim Urlöschen an Geräte angeschlossen, dann kann es sein, daß Veränderungen in den Signalzuständen der Ausgänge an die Geräte übertragen werden. Haben Sie den vom Werk voreingestellten "sicheren Zustand" der Ausgänge geändert, dann können Zustandsänderungen in den Ausgängen unerwartete Reaktionen der Geräte auslösen. Dadurch können Körperverletzungen und/oder Sachschäden entstehen.

Treffen Sie deshalb alle notwendigen Sicherheitsvorkehrungen und vergewissern Sie sich, daß sich Ihr Prozeß in einem sicheren Zustand befindet, bevor Sie die CPU urlöschen.

6.8 Testen und Überwachen Ihres Programms

STEP 7-Micro/WIN stellt Ihnen verschiedene Werkzeuge zum Testen und Überwachen Ihres Programms zur Verfügung.

Überwachen Ihres Programms durch Ausführen einer bestimmten Anzahl Zyklen

Sie können angeben, daß die CPU Ihr Programm für eine bestimmte Anzahl von Zyklen bearbeiten soll (von 1 Zyklus bis zu 65.535 Zyklen). Wenn Sie auswählen, wieviele Zyklen die CPU ausführen soll, können Sie die Bearbeitung der Prozessvariablen beobachten. Mit dem Menübefehl **Testen ► Zyklen ausführen** geben Sie die Anzahl der Zyklen an, die die CPU ausführen soll. Bild 6-11 zeigt das Dialogfeld, in dem Sie die Anzahl der Zyklen angeben.

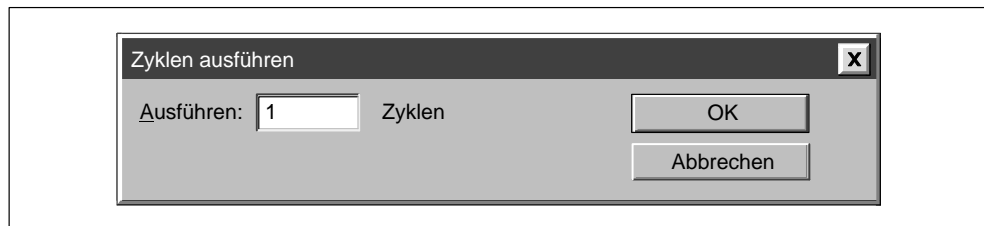


Bild 6-11 Ausführen des Programms für eine bestimmte Anzahl von Zyklen

Bedienen und Beobachten Ihres Programms mit einer Statustabelle

Sie können mit einer Statustabelle Variablen lesen, schreiben, forcen und beobachten, während das Programm bearbeitet wird (siehe Bild 6-12). Ausführliche Informationen zum Anlegen einer Statustabelle entnehmen Sie dem Abschnitt 3.8.

Adresse	Format	Aktueller Wert	Neuer Wert
"Start_1"	Bit	2#0	
"Start_2"	Bit	2#0	1
"Stopp_1"	Bit	2#0	
"Stopp_2"	Bit	2#0	
"Behälter_voll"	Bit	2#0	
"Behälter_leer"	Bit	2#0	
"Rücksetzen"	Bit	2#0	
"Pumpe_1"	Bit	2#0	
"Pumpe_2"	Bit	2#0	
"Rührmotor"	Bit	2#0	
"Dampfventil"	Bit	2#0	
"Abflußventil"	Bit	2#0	
"Abflußpumpe"	Bit	2#0	
"Max_Füllstand"	Bit	2#0	
"Mischzeit"	Mit Vorzeichen	+0	
"Zykluszähler"	Mit Vorzeichen	+0	

Bild 6-12 Bedienen und Beobachten von Variablen mit einer Statustabelle

Anzeigen des Status im KOP-Programm

Mit dem Programm-Editor in STEP 7-Micro/WIN können Sie online den Status eines Programms überwachen (siehe Bild 6-13). Dabei muß das Programm im Kontaktplan angezeigt werden. So können Sie den Zustand der Operationen im Programm überwachen, während die CPU das Programm bearbeitet.

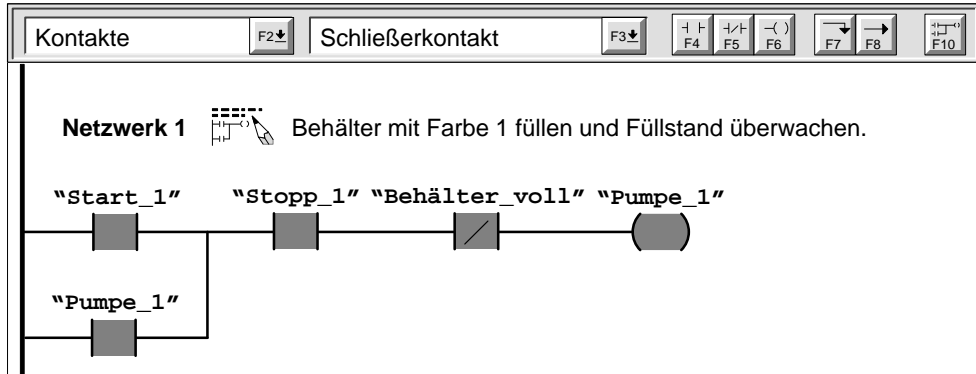


Bild 6-13 Anzeigen des Status in einem KOP-Programm

Forcen von Werten in der Statustabelle

Mit der S7-200 CPU können Sie einige oder alle Ein- und Ausgänge (E- und A-Bits) auf bestimmte Werte forcen. Zusätzlich können Sie insgesamt 16 interne Merker (V oder M) oder Analogein- bzw. Analogausgänge (AE oder AA) forcen. Werte im Variablenspeicher und Werte von Merkern können als Bytes, Wörter und Doppelwörter geforct werden. Analogwerte können nur als Wörter geforct werden, und zwar auf geraden Bytes (z.B. AEW6 oder AAW14). Alle geforcten Werte werden im nullspannungsfesten EEPROM der CPU abgelegt.

Während des Zyklus können die geforcten Datenwerte geändert werden (vom Programm, bei der Aktualisierung der Ein- und Ausgänge oder durch die Bearbeitung der Kommunikation). Deshalb überschreibt die CPU die geforcten Werte immer wieder zu verschiedenen Zeiten im Zyklus. Bild 6-14 zeigt, an welchen Stellen im Zyklus die CPU die geforcten Variablen aktualisiert.

Die Force-Funktion übersteuert Operationen zum direkten Lesen bzw. Schreiben von Ein- und Ausgängen. Die Force-Funktion übersteuert auch die Konfiguration eines Ausgangs, der beim Übergang in den Betriebszustand STOP einen bestimmten Signalzustand annehmen soll. Geht die CPU in STOP, dann behält der Ausgang den geforcten Wert bei und nimmt nicht den konfigurierten Wert an.

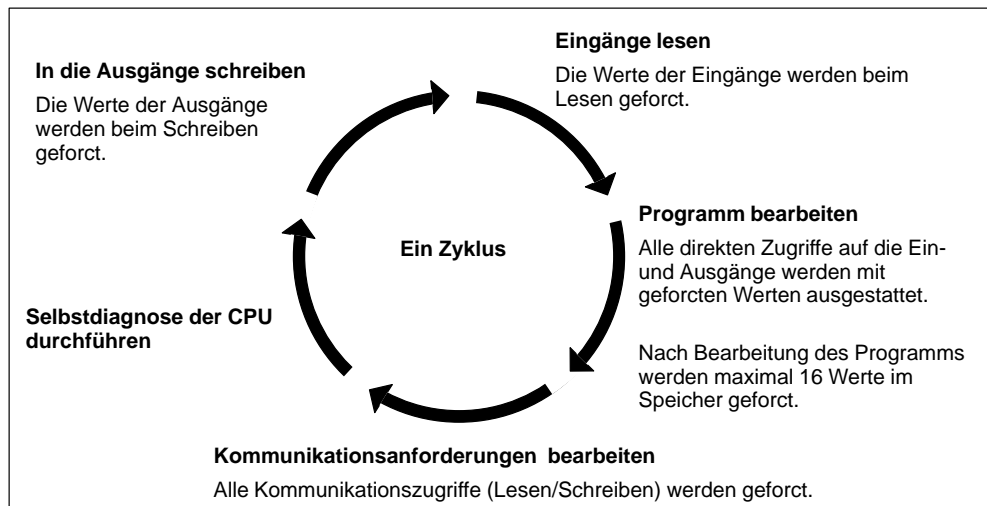


Bild 6-14 Zyklus einer S7-200 CPU

Bild 6-15 zeigt ein Beispiel für eine Statustabelle. Ausführliche Informationen zum Arbeiten mit der Statustabelle entnehmen Sie Abschnitt 3.8.

Adresse	Format	Aktueller Wert	Neuer Wert
"Start_1"	Bit	2#0	
"Start_2"	Bit	2#0	1
"Stopp_1"	Bit	2#0	
"Stopp_2"	Bit	2#0	
"Behälter_voll"	Bit	2#0	
"Behälter_leer"	Bit	2#0	
"Rücksetzen"	Bit	2#0	
"Pumpe_1"	Bit	2#0	
"Pumpe_2"	Bit	2#0	
"Rührmotor"	Bit	2#0	
"Dampfventil"	Bit	2#0	
"Abflußventil"	Bit	2#0	
"Abflußpumpe"	Bit	2#0	
"Max_Füllstand"	Bit	2#0	
"Mischzeit"	Mit Vorzeichen	+0	
"Zykluszähler"	Mit Vorzeichen	+0	

Bild 6-15 Forcen von Variablen in der Statustabelle

6.9 Fehlerbehebung bei der S7-200 CPU

Die S7-200 CPU unterteilt aufgetretene Fehler in schwere und leichte Fehler. Mit STEP 7-Micro/WIN können Sie sich die Fehlercodes anzeigen lassen, die von den aufgetretenen Fehlern erzeugt wurden. Bild 6-16 zeigt das Dialogfeld "CPU-Informationen", in dem der Fehlercode und die Beschreibung des Fehlers angezeigt werden. Eine vollständige Liste aller Fehlercodes entnehmen Sie Anhang C.

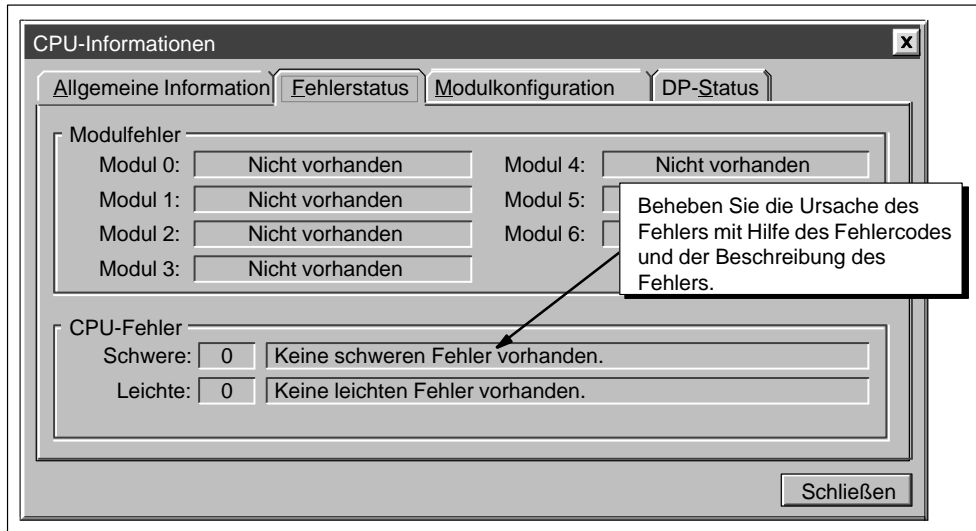


Bild 6-16 Register "Fehlerstatus" im Dialogfeld "CPU-Informationen"

Beheben von schweren Fehlern

Tritt ein schwerer Fehler auf, beendet Ihre CPU die Bearbeitung des Programms. Je nach dem Schweregrad des Fehlers kann die CPU einige oder auch gar keine Funktionen mehr ausführen. Ziel der Behebung von schweren Fehlern ist es, die CPU in einen sicheren Zustand zu bringen, so daß Informationen zu der Fehlerbedingung in der CPU abgefragt werden können. Erkennt die CPU einen schweren Fehler, geht sie in die Betriebsart STOP, schaltet die LED-Anzeige für Systemfehler (SF) und die LED-Anzeige für STOP ein und schaltet die Ausgänge aus. Die CPU bleibt in diesem Zustand, bis die Fehlerbedingung behoben ist.

Haben Sie die Fehlerbedingung des schweren Fehlers behoben, müssen Sie die CPU neu starten. Hierzu können Sie entweder die Spannungsversorgung aus- und wieder einschalten oder den Betriebsartenschalter der CPU von RUN bzw. TERM in STOP bringen. Durch das Neustarten der CPU wird die Fehlerbedingung gelöscht und beim Anlauf eine Diagnose ausgeführt, um zu prüfen, ob der schwere Fehler tatsächlich behoben ist. Wird dabei ein weiterer schwerer Fehler erkannt, leuchtet wieder die LED-Anzeige der CPU auf und deutet damit an, daß noch immer ein Fehler vorhanden ist. Andernfalls beginnt die CPU ihren normalen Betrieb.

Es gibt verschiedene mögliche Fehlerbedingungen, die die CPU kommunikationsunfähig machen. In diesen Fällen können Sie sich den Fehlercode der CPU nicht anzeigen lassen. Diese Art von Fehlern deuten meist auf Hardware-Fehler hin, die nur durch Reparatur der CPU behoben werden können. Solche Fehlerbedingungen können nicht durch Änderungen im Programm oder Löschen des CPU-Speichers beseitigt werden.

Beheben von leichten Fehlern

Leichte Fehler können den Betrieb der CPU teilweise einschränken. Die CPU ist jedoch weiterhin in der Lage, das Programm zu bearbeiten und die Ein- und Ausgänge zu aktualisieren. Sie können sich mit STEP 7-Micro/WIN die Fehlercodes anzeigen lassen, die von den leichten Fehlern erzeugt wurden (siehe Bild 6-16). Es gibt drei wesentliche Gruppen leichter Fehler:

- Fehler zur Laufzeit: Alle leichten Fehler, die während des Betriebszustands RUN erkannt werden, werden in Sondermerkern abgelegt. Ihr Programm kann diese Sondermerker überwachen und auswerten. Ausführliche Informationen zu Sondermerkern, die leichte Fehler zur Laufzeit anzeigen, entnehmen Sie Anhang D.

Beim Anlauf liest die CPU die Konfiguration der Ein- und Ausgänge und speichert diese Informationen im Systemdatenspeicher und in den Sondermerkern. Während des normalen Betriebs wird der Status der Ein- und Ausgänge regelmäßig aktualisiert und in den Sondermerkern abgelegt. Erkennt die CPU Unterschiede in der Konfiguration der Ein- und Ausgänge, dann setzt die CPU das Bit zum Anzeigen einer geänderten Konfiguration im Fehlerbyte des Moduls. Das Erweiterungsmodul wird dann solange nicht aktualisiert, bis dieses Bit zurückgesetzt wurde. Damit die CPU das Bit zurücksetzen kann, müssen die Ein- und Ausgänge des Moduls wieder mit der Konfiguration der Ein- und Ausgänge, die im Systemdatenspeicher abgelegt ist, übereinstimmen.

- Fehler beim Übersetzen des Programms: Wenn Sie ein Programm in die CPU laden, übersetzt die CPU das Programm. Erkennt die CPU, daß das Programm eine Übersetzungsregel verletzt, dann bricht die CPU den Ladevorgang ab und erzeugt einen Fehlercode. (War bereits ein Programm in die CPU geladen, ist dieses Programm im EEPROM noch vorhanden, es geht nicht verloren.) Nachdem Sie Ihr Programm korrigiert haben, können Sie es erneut laden.
- Programmierfehler zur Laufzeit: Sie (bzw. Ihr Programm) können Fehlerbedingungen hervorrufen, die auftreten, während das Programm bearbeitet wird. Ein indirekter Adreßpointer beispielsweise, der beim Übersetzen des Programms gültig war, kann während der Bearbeitung des Programms geändert worden sein, so daß er auf eine Adresse außerhalb des gültigen Bereichs zeigt. Dies wird als Programmierfehler zur Laufzeit bezeichnet. Anhand des Dialogfelds "CPU-Informationen" (siehe Bild 6-16) können Sie feststellen, welche Art von Fehler aufgetreten ist.

Die CPU geht nicht in den Betriebszustand STOP, wenn ein leichter Fehler erkannt wird. Es legt die Ereignisse in Sondermerkern (SM) ab und fährt mit der Programmbearbeitung fort. Sie können jedoch Ihr Programm so schreiben, daß bei Auftreten eines leichten Fehlers der Übergang in den Betriebszustand STOP erzwungen wird. Bild 6-17 zeigt ein Netzwerk eines Programms, das einen Sondermerker überwacht. Die Operation versetzt die CPU in den Betriebszustand STOP, wenn ein E/A-Fehler erkannt wird.

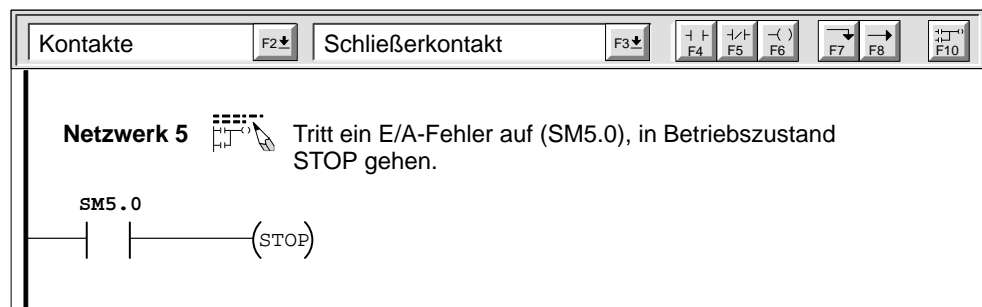


Bild 6-17 Erkennen von leichten Fehlerbedingungen in Ihrem Anwenderprogramm

Speicher der CPU: Datentypen und Adressierungsarten

7

Die S7-200 CPU verfügt über spezifische Speicherbereiche, damit Ihre Daten schneller und wesentlich effizienter bearbeitet werden können.

Kapitelübersicht

Abschnitt	Beschreibung	Seite
7.1	Direkte Adressierung der Speicherbereiche in der CPU	7-2
7.2	Indirekte Adressierung der Speicherbereiche in der CPU	7-9
7.3	Datenhaltung in der S7-200 CPU	7-11
7.4	Nullspannungsfestes Speichern von Daten mit Ihrem Programm	7-16
7.5	Speichern Ihres Programms im Speichermodul	7-17

7.1 Direkte Adressierung der Speicherbereiche in der CPU

Die S7-200 CPU speichert Informationen an verschiedenen Adressen im Speicher, die eindeutig angesprochen werden. Sie können die Adresse im Speicher, auf die Sie zugreifen möchten, explizit angeben. Dadurch hat Ihr Programm direkten Zugriff auf die Informationen.

Zugreifen auf Daten über Adressen

Wenn Sie auf ein Bit in einem Speicherbereich zugreifen möchten, müssen Sie die Adresse des Bit angeben. Diese Adresse besteht aus der Kennung des Speicherbereichs, der Adresse des Byte sowie der Nummer des Bit. In Bild 7-1 sehen Sie ein Beispiel für den Zugriff auf ein Bit (Adressierung im Byte.Bit-Format). In diesem Beispiel folgt auf den Speicherbereich und die Adresse des Byte (E=Eingang, 3=Byte 3) ein Punkt ("."), um die Adresse des Bit (Bit 4) abzutrennen.

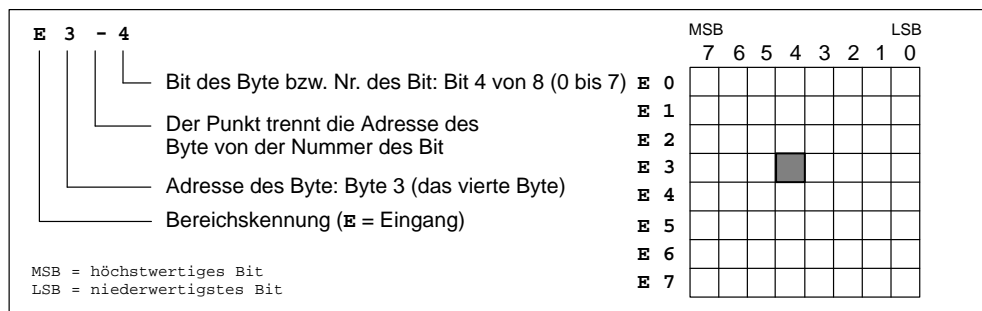


Bild 7-1 Zugriff auf ein Datenbit im Speicher der CPU (Byte.Bit-Adressierung)

Wenn Sie das Byteformat für die Adressierung verwenden, können Sie auf Daten in verschiedenen Speicherbereichen der CPU (V, E, A, M und SM) als Bytes, Wörter oder Doppelwörter zugreifen. Wenn Sie auf ein Byte, Wort oder Doppelwort an Daten im Speicher der CPU zugreifen möchten, dann müssen Sie diese Adresse ähnlich wie die Adresse eines Bit angeben. Sie geben die Bereichskennung, die Größe der Daten (Format des Zugriffs) und die Anfangsadresse des Byte-, Wort- oder Doppelwortwerts an (siehe Bild 7-2). Auf Daten in anderen Speicherbereichen der CPU (z. B. T, Z, HC und Akkumulatoren) greifen Sie zu, indem Sie für die Adresse eine Bereichskennung und die Nummer des Elements angeben.

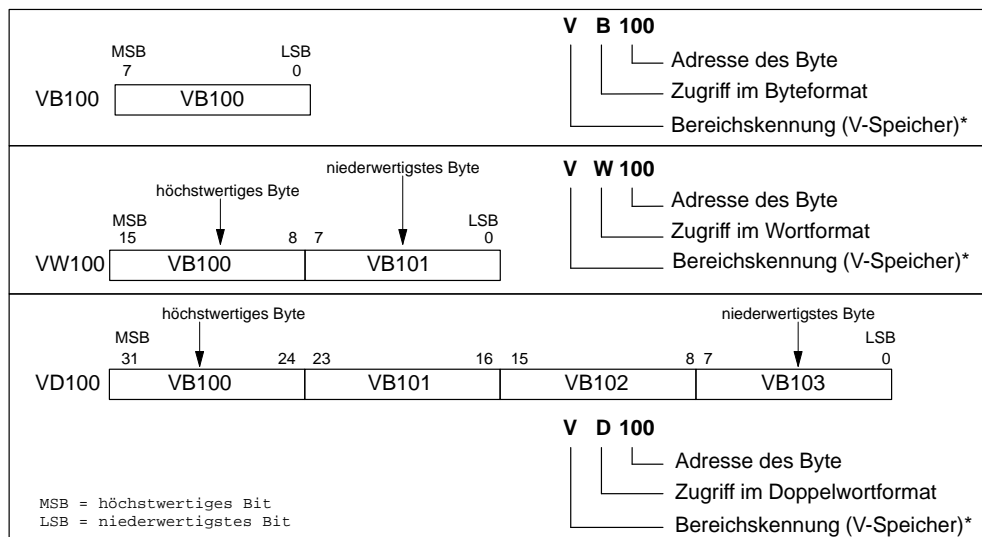


Bild 7-2 Zugriff auf dieselbe Adresse im Byte-, Wort- und Doppelwortformat

Darstellung von Zahlen

Tabelle 7-1 zeigt den Bereich der ganzzahligen Werte, die durch die unterschiedlichen Datengrößen dargestellt werden können.

Realzahlen (Gleitpunktzahlen) werden als einfachgenaue Zahlen (32 Bit) dargestellt, deren Format in den Richtlinien ANSI/IEEE 754-1985 beschrieben ist. Auf Realzahlenwerte wird im Doppelwortformat zugegriffen.

Tabelle 7-1 Kennzeichen der Datengröße (und zugeordnete Bereiche der ganzen Zahlen)

Datengröße	Bereich der Zahlen ohne Vorzeichen		Bereich der Zahlen mit Vorzeichen	
	Dezimal	Hexadezimal	Dezimal	Hexadezimal
B (Byte): 8 Bit	0 bis 255	0 bis FF	-128 bis 127	80 bis 7F
W (Wort): 16 Bit	0 bis 65.535	0 bis FFFF	-32.768 bis 32.767	8000 bis 7FFF
D (Doppelwort, Dwort): 32 Bit	0 bis 4.294.967.295	0 bis FFFF FFFF	-2.147.483.648 bis 2.147.483.647	8000 0000 bis 7FFF FFFF

Adressierung des Prozeßabbilds der Eingänge (E)

Wie in Abschnitt 6.5 beschrieben, fragt die CPU die physikalischen Eingänge zu Beginn eines jeden Zyklus ab und schreibt diese Werte in das Prozeßabbild der Eingänge. Auf das Prozeßabbild können Sie im Bit-, Byte-, Wort- und Doppelwortformat zugreifen.

Format: Bit *E[Adr. des Byte].[Adr. des Bit] E0.1*
 Byte, Wort, Doppelwort *E[Größe][Anfangsadr. des Byte] EB4*

Adressierung des Prozeßabbilds der Ausgänge (A)

Am Ende des Zyklus kopiert die CPU die Werte des Prozeßabbilds der Ausgänge in die physikalischen Ausgänge. Auf das Prozeßabbild können Sie im Bit-, Byte-, Wort- und Doppelwortformat zugreifen.

Format: Bit *A[Adr. des Byte].[Adr. des Bit] A1.1*
 Byte, Wort, Doppelwort *A[Größe][Anfangsadr. des Byte] AB5*

Adressierung des Variablenspeichers (V)

Im Variablenspeicher können Sie Zwischenergebnisse ablegen, die von den Operationen in Ihrem Programm errechnet werden. Sie können im Variablenspeicher auch andere Daten ablegen, die zu Ihrem Prozeß bzw. zu Ihrer Automatisierungslösung gehören. Auf den Variablenspeicher können Sie im Bit-, Byte-, Wort- und Doppelwortformat zugreifen.

Format: Bit *V[Adr. des Byte].[Adr. des Bit] V10.2*
 Byte, Wort, Doppelwort *V[Größe][Anfangsadr. des Byte] VW100*

Adressierung von Merkern (M)

Interne Merker (Speicherbereich der Merker, M) können Sie wie Steuerungsrelais verwenden, um Zwischenergebnisse von Operationen oder andere Steuerungsinformationen zu speichern. Auf Merker können Sie im Bit-, Byte-, Wort- und Doppelwortformat zugreifen.

Format: Bit *M[Adr. des Byte].[Adr. des Bit] M26.7*
 Byte, Wort, Doppelwort *M[Größe][Anfangsadr. des Byte] MD20*

Adressierung von Ablaufsteuerungsrelais (S)

Mit Ablaufsteuerungsrelais (S) gliedern Sie die Funktionsweise einer Anlage in einzelne Schritte bzw. in äquivalente Programmteile. Ablaufsteuerungsrelais unterteilen das Steuerprogramm in logische Segmente. Auf S-Bits können Sie im Bit-, Byte-, Wort- und Doppelwortformat zugreifen.

Format: Bit *S[Adr. des Byte].[Adr. des Bit] S3.1*
Byte, Wort, Doppelwort *S[Größe][Anfangsadr. des Byte] SB4*

Adressierung von Sondermerkern (SM)

Mit Sondermerkern können Sie Informationen zwischen der CPU und Ihrem Programm austauschen. Außerdem dienen Sondermerker dazu, besondere Funktionen der S7-200 CPU auszuwählen und zu steuern. Dazu gehören:

- Ein Bit, das nur im ersten Zyklus eingeschaltet ist
- Bits, die in bestimmten Takten ein- und ausgeschaltet werden
- Bits, die den Zustand von arithmetischen und anderen Operationen anzeigen

Ausführliche Informationen zu Sondermerkern entnehmen Sie Anhang D. Der Speicherbereich der Sondermerker basiert auf Bits, doch Sie können auf die Daten in diesen Bereichen im Bit-, Byte-, Wort- und Doppelwortformat zugreifen.

Format: Bit *SM[Adr. des Byte].[Adr. des Bit] SM0.1*
Byte, Wort, Doppelwort *SM[Größe][Anfangsadr. des Byte] SMB86*

Adressierung von Zeiten (T)

In der S7-200 CPU sind Zeiten Elemente, die Zeitinkremente zählen. Die Zeiten der S7-200 haben Auflösungen (Inkremente der Zeitbasis) von 1 ms, 10 ms und 100 ms. Jede Zeit verfügt über die folgenden zwei Variablen:

- Aktueller Wert: Diese ganze Zahl (16 Bit) mit Vorzeichen speichert den Zeitwert der Zeit.
- Zeitbit: Dieses Bit wird eingeschaltet (auf "1" gesetzt), wenn der aktuelle Wert der Zeit größer als oder gleich dem voreingestellten Wert ist (der voreingestellte Wert wird zusammen mit der Operation eingegeben).

Sie greifen auf die beiden Datenelemente über die Adresse der Zeit (T + Nummer der Zeit) zu. Ob auf das Zeitbit oder den aktuellen Wert der Zeit zugegriffen wird, richtet sich nach der jeweiligen Operation. Operationen mit Operanden im Bitformat greifen auf das Zeitbit zu, während Operationen mit Operanden im Wortformat auf den aktuellen Wert zugreifen. In Bild 7-3 sehen Sie, daß die Operation Schließerkontakt auf das Zeitbit zugreift, während die Operation Wort übertragen (MOVW) auf den aktuellen Wert der Zeit zugreift. Ausführliche Informationen zu den Operationen der S7-200 entnehmen Sie Kapitel 10.

Format: *T[Nummer der Zeit] T24*

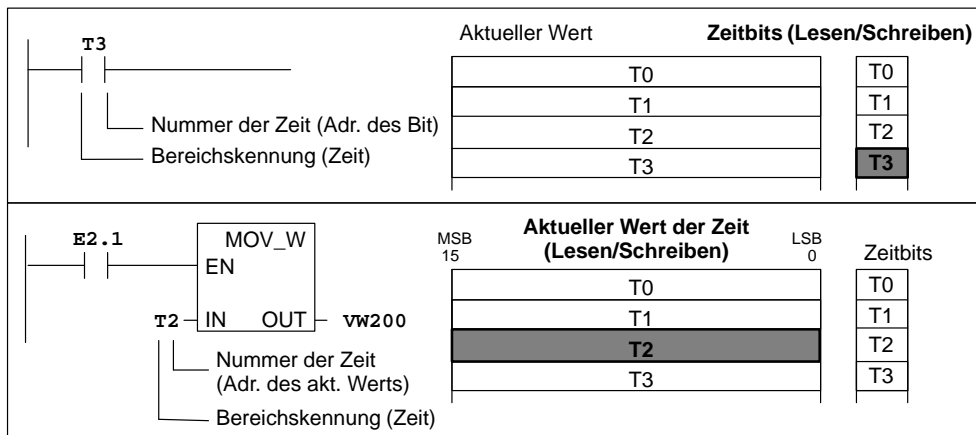


Bild 7-3 Zugriff auf Daten einer Zeit

Adressierung von Zählern (Z)

In der S7-200 CPU sind Zähler Elemente, die an den Zähleringängen die steigenden Flanken zählen. Die CPU verfügt über zwei Arten von Zählern: ein Zähler zählt nur vorwärts, während der andere Zähler sowohl vorwärts als auch rückwärts zählt. Jeder Zähler verfügt über die folgenden zwei Variablen:

- **Aktueller Wert:** Diese ganze Zahl (16 Bit) mit Vorzeichen speichert den Zählwert des Zählers.
- **Zählerbit:** Dieses Bit wird eingeschaltet (auf "1" gesetzt), wenn der aktuelle Wert des Zählers größer als oder gleich dem voreingestellten Wert ist (der voreingestellte Wert wird zusammen mit der Operation eingegeben).

Sie greifen auf die beiden Variablen über die Adresse des Zählers (Z + Nummer der Zeit) zu. Ob auf das Zählerbit oder den aktuellen Wert des Zählers zugegriffen wird, richtet sich nach der jeweiligen Operation. Operationen mit Operanden im Bitformat greifen auf das Zählerbit zu, während Operationen mit Operanden im Wortformat auf den aktuellen Wert zugreifen. In Bild 7-4 sehen Sie, daß die Operation Schließerkontakt auf das Zählerbit zugreift, während die Operation Wort übertragen (MOVW) auf den aktuellen Wert des Zählers zugreift. Ausführliche Informationen zu den Operationen der S7-200 entnehmen Sie Kapitel 10.

Format: **Z[Nummer des Zählers]** **Z20**

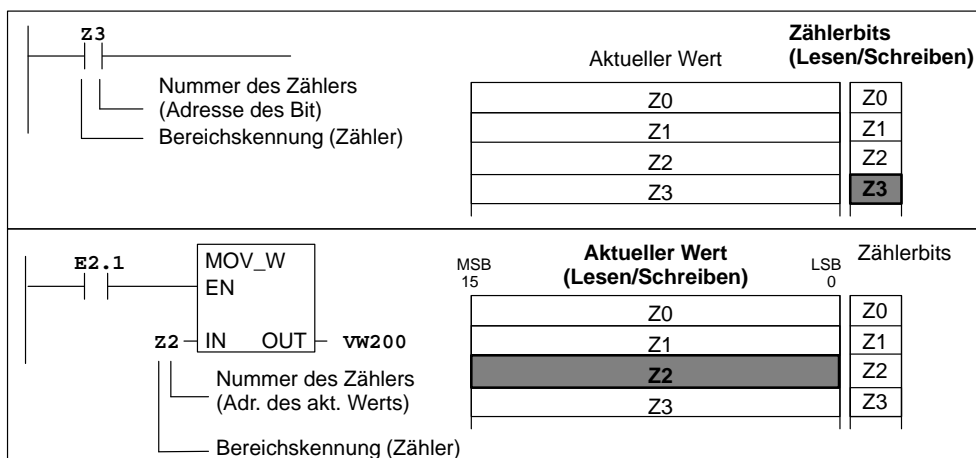


Bild 7-4 Zugriff auf Daten eines Zählers

Adressierung von Analogeingängen (AE)

Die S7-200 wandelt reale Analogwerte (z. B. Spannung, Temperatur) in digitale Wortwerte (16 Bit) um. Sie greifen auf diese Werte über die Bereichskennung (AE), die Größe der Daten (W) und die Anfangsadresse des Byte zu. Da es sich bei Analogeingängen um Wörter handelt, die immer auf geraden Bytes beginnen (also 0, 2, 4 usw.), sprechen Sie die Werte mit den Adressen gerader Bytes an (z. B. AEW0, AEW2, AEW4) (siehe Bild 7-5). Analogeingänge können nur gelesen werden.

Format: **AEW[Anfangsadresse des Byte] AEW4**

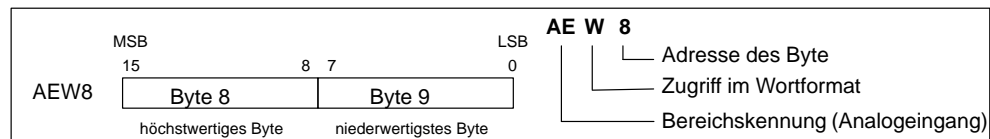


Bild 7-5 Zugriff auf einen Analogeingang

Adressierung von Analogausgängen (AA)

Die S7-200 wandelt digitale Wortwerte (16 Bit) in Strom bzw. Spannung um, und zwar proportional zum digitalen Wert. Sie greifen auf diese Werte über die Bereichskennung (AA), die Größe der Daten (W) und die Anfangsadresse des Byte zu. Da es sich bei Analogausgängen um Wörter handelt, die immer auf geraden Bytes beginnen (also 0, 2, 4 usw.), sprechen Sie die Werte mit den Adressen gerader Bytes an (z. B. AAW0, AAW2, AAW4) (siehe Bild 7-6). Ihr Programm kann die Werte von Analogausgängen nicht lesen.

Format: **AAW[Anfangsadresse des Byte] AAW4**

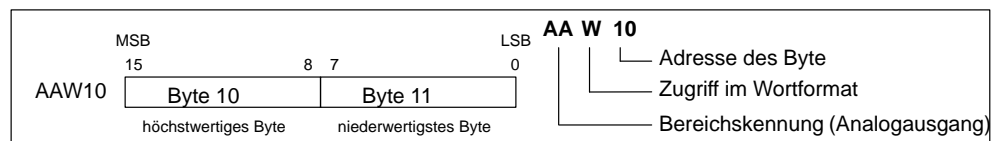


Bild 7-6 Zugriff auf einen Analogausgang

Adressierung von Akkumulatoren (AC)

Akkumulatoren sind Schreib-/Lese-Elemente, die wie Speicher verwendet werden. Sie können mit Akkumulatoren z. B. Parameter an Unterprogramme übergeben und auch wieder zurücknehmen oder Zwischenergebnisse von Berechnungen speichern. Die CPU verfügt über vier 32-Bit-Akkumulatoren (AC0, AC1, AC2 und AC3). Auf die Daten in den Akkumulatoren können Sie im Bit-, Byte-, Wort- und Doppelwortformat zugreifen. Wie Bild 7-7 zeigt, werden beim Zugreifen auf einen Akkumulator im Byte- oder Wortformat die niederwertigsten 8 oder 16 Bits des Werts verwendet. Wenn Sie auf einen Akkumulator im Doppelwortformat zugreifen, werden alle 32 Bits verwendet. Die Größe der Daten, auf die zugegriffen wird, richtet sich nach der Operation, mit der Sie auf den Akkumulator zugreifen.

Format: **AC[Nummer des Akkumulators] AC0**

Hinweis

Informationen zum Verwenden von Akkumulatoren mit Interruptprogrammen entnehmen Sie Abschnitt 10.14.

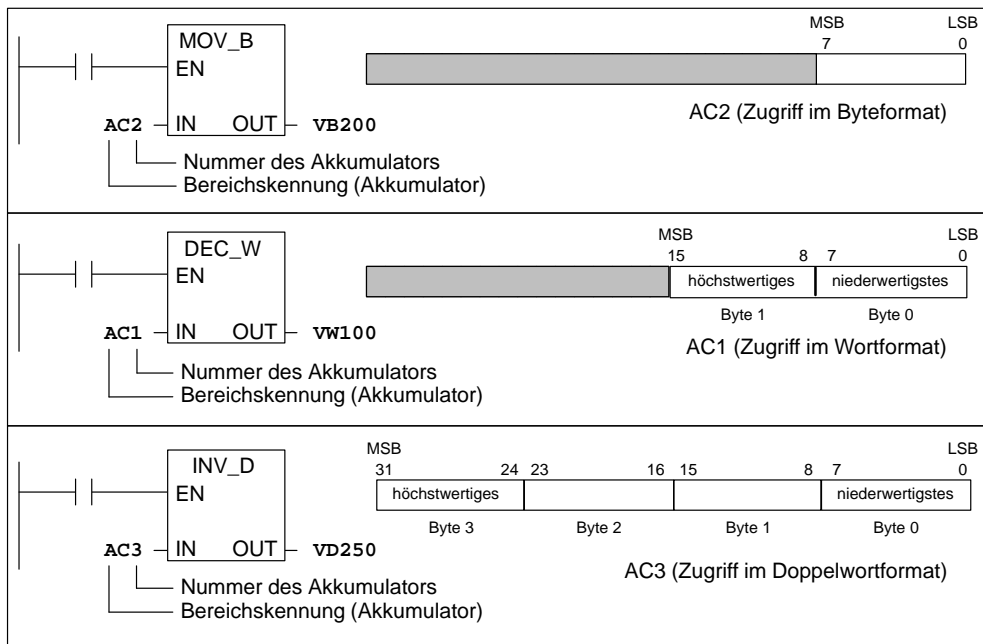


Bild 7-7 Adressierung von Akkumulatoren

Adressierung von schnellen Zählern (HC)

Schnelle Zähler zählen die Ereignisse schneller, als die CPU die Ereignisse abfragen kann. Schnelle Zähler verfügen über einen ganzzahligen 32-Bit-Zählwert (den aktuellen Wert). Wenn Sie auf den Zählwert eines schnellen Zählers zugreifen möchten, geben Sie die Adresse des schnellen Zählers mittels des Speicherbereichs (HC) und der Nummer des Zählers (z. B. HC0) an. Der aktuelle Wert eines schnellen Zählers ist schreibgeschützt und kann nur im Doppelwortformat (32 Bit) adressiert werden (siehe Bild 7-8).

Format: `HC[Nummer des Zählers]` HC1

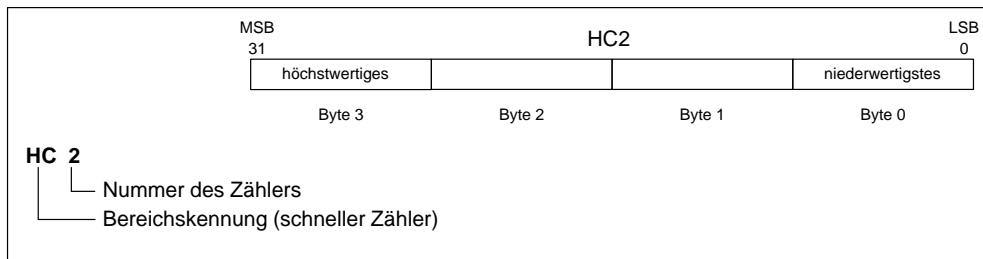


Bild 7-8 Zugriff auf den aktuellen Wert eines schnellen Zählers

Verwenden von Konstanten

In vielen Operationen für die S7-200 können Sie Konstanten verwenden. Konstanten können Bytes, Wörter und Doppelwörter sein. Die CPU speichert alle Konstanten als Binärwerte, die im Dezimal-, Hexadezimal- und ASCII-Format dargestellt werden können.

Dezimalformat: **[Dezimalwert]**
Hexadezimalformat: **16#[Hexadezimalwert]**
ASCII-Format: **'[ASCII-Text]'**

Bei der S7-200 CPU können Sie keine spezifischen Datentypen angeben (wenn Sie z. B. angeben möchten, daß die Konstante als ganze Zahl (16 Bit), als ganze Zahl mit Vorzeichen oder als ganze Zahl (32 Bit) gespeichert werden soll). Die S7-200 CPU prüft auch keine Datentypen. Die Operation Wert addieren kann einen in VW100 abgelegten Wert beispielsweise als ganze Zahl mit Vorzeichen verwenden, während die Operation EXKLUSIV ODER denselben Wert in VW100 als vorzeichenlosen Binärwert einsetzt.

Die folgenden Beispiele zeigen Konstanten im Dezimal-, Hexadezimal- und ASCII-Format:

- Dezimalkonstante: **20047**
- Hexadezimalkonstante: **16#4E4F**
- ASCII-Konstante: **•Text in Hochkommata.·**

7.2 Indirekte Adressierung der Speicherbereiche in der CPU

Die indirekte Adressierung verwendet Pointer, um auf Daten im Speicher zuzugreifen. In der S7-200 CPU können Sie mittels Pointern die folgenden Speicherbereiche indirekt adressieren: E, A, V, M, S, T (nur den aktuellen Wert) und Z (nur den aktuellen Wert). Einzelne Bit- oder Analogwerte können Sie nicht indirekt adressieren.

Erstellen eines Pointers

Wenn Sie eine Adresse indirekt ansprechen möchten, müssen Sie zunächst einen Pointer erstellen, der auf die Adresse zeigt. Pointer sind Doppelwortwerte, die eine Adresse im Speicher enthalten. Als Pointer können Sie nur Adressen des Variablenspeichers bzw. einen der Akkumulatoren (AC1, AC2 oder AC3) angeben. Zum Erstellen eines Pointers verwenden Sie die Operation Doppelwort übertragen (MOVD). Diese Operation überträgt die indirekt adressierte Adresse an die Adresse des Pointers. Dem Eingabeoperanden der Anweisung müssen Sie das Zeichen & voranstellen, um dadurch anzugeben, daß die Adresse und nicht der Inhalt der Adresse an die vom Ausgabeoperanden der Anweisung (Pointer) angegebene Adresse übertragen werden soll.

Beispiel:

MOVD	&VB100, VD204
MOVD	&MB4, AC2
MOVD	&Z4, VD6

Hinweis

Wenn Sie auf einen Wort- oder Doppelwortwert in einem der Speicherbereiche E, A, V, M oder S indirekt zugreifen möchten, müssen Sie die Adresse des Anfangsbyte des Werts als den Eingabeoperanden der Operation MOVD, die zum Erstellen des Pointers verwendet wird, angeben. Beispiel: VB100 ist die Adresse des Anfangsbyte von VW100 und MB4 ist die Adresse des Anfangsbyte von MD4. Verfügt der Wort- oder Doppelwortwert über einen symbolischen Namen, können Sie den symbolischen Namen nicht in der Operation MOVD, mit der Sie den Pointer erstellen, verwenden, weil Sie die Adresse des Anfangsbytes des Werts als Eingabeoperand der Operation angeben müssen. Zum Erstellen eines Pointers müssen Sie dann der Adresse des Anfangsbyte des Wort- bzw. Doppelwortwerts einen eigenen symbolischen Namen zuordnen.

Beispiel: `'Pumpe_Umdrehungen'` ist der symbolische Name von VW100
`'Pumpe_Umdrehungen_EB'` ist der symbolische Name von VB100
 (dies ist das Anfangsbyte des Wortwerts in VW100)

MOVD &"Pumpe_Umdrehungen", AC1
 unzulässig (&VW100 dürfen Sie nicht angeben)
MOVD &"Pumpe_Umdrehungen_EB", AC1
 korrekt (&VB100 ist in Ordnung)

Zugreifen auf Daten mit einem Pointer

Ein Stern (*) vor dem Operanden einer Operation zeigt an, daß es sich bei der Adresse um einen Pointer handelt. In dem Beispiel in Bild 7-9 gibt *AC1 an, daß AC1 ein Pointer ist, der die Adresse für den Wortwert enthält, der von der Operation Wort übertragen (MOVW) angegeben wird. In diesem Beispiel werden die Werte von V200 und V201 in den Akkumulator AC0 übertragen.

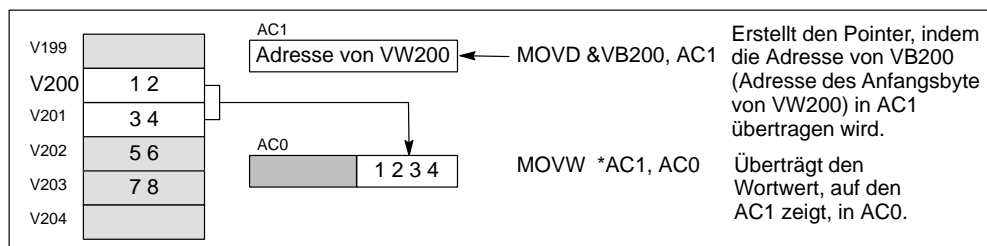


Bild 7-9 Verwenden eines Pointers für die indirekte Adressierung

Ändern von Pointern

Sie können den Wert eines Pointers ändern. Da es sich bei Pointern um 32-Bit-Werte handelt, müssen Sie Pointerwerte mit Doppelwort-Operationen ändern. Mit einfachen arithmetischen Operationen können Sie Pointerwerte ändern, z.B. durch Addieren oder Inkrementieren. Achten Sie darauf, daß Sie die Größe der Daten angeben, auf die Sie zugreifen möchten:

- Wenn Sie auf Bytes zugreifen, inkrementieren Sie den Wert des Pointers um 1.
- Wenn Sie auf Wörter oder aktuelle Werte von Zeiten oder Zählern zugreifen, addieren Sie den Wert 2 bzw. inkrementieren Sie den Pointerwert um 2.
- Wenn Sie auf Doppelwörter zugreifen, addieren Sie den Wert 4 bzw. inkrementieren Sie den Pointerwert um 4.

In Bild 7-10 sehen Sie ein Beispiel dafür, wie Sie einen Pointer auf eine indirekte Adresse erstellen, wie auf Daten indirekt zugegriffen wird und wie Sie den Pointer um 1 erhöhen können.

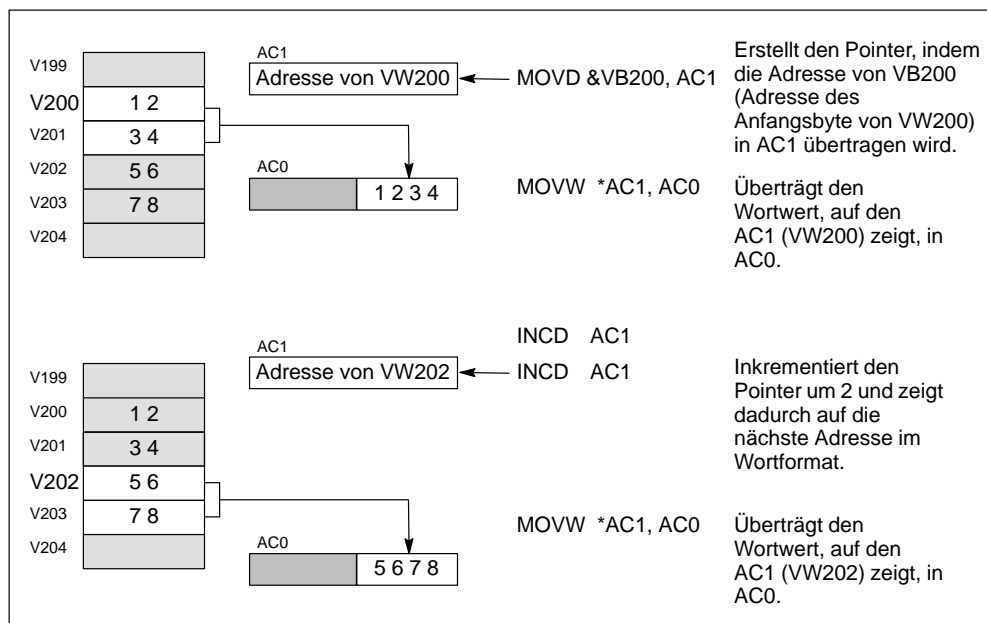


Bild 7-10 Ändern eines Pointers beim Zugreifen auf einen Wortwert

7.3 Datenhaltung in der S7-200 CPU

Die S7-200 CPU bietet Ihnen verschiedene Methoden, um sicherzustellen, daß Ihr Programm, die Programmdatei und die Konfigurationsdatei Ihrer CPU sicher abgelegt sind:

- Die CPU verfügt über einen EEPROM, in dem Sie Ihr gesamtes Programm, einige Datenbereiche und die Konfigurationsdatei für die CPU nullspannungsfest ablegen können (siehe Bild 7-11).
- Die CPU verfügt über einen Hochleistungskondensator, der die Datensicherheit des RAM-Speichers auch nach dem Abschalten der Spannungsversorgung der CPU gewährleistet. Je nach der Variante der CPU kann der Hochleistungskondensator den RAM-Speicher mehrere Tage lang puffern.
- Einige CPU-Varianten unterstützen ein optionales Batteriemodul, mit dem Sie die Zeit verlängern können, die der RAM-Speicher nach Abschalten der Spannungsversorgung der CPU die Daten puffert. Das Batteriemodul übernimmt nach dem Entladen des Hochleistungskondensators das Puffern der Daten.

Dieser Abschnitt erläutert Ihnen, wie die CPU Ihre Daten in verschiedenen Situationen im RAM-Speicher ablegt und von dort zurückholt.

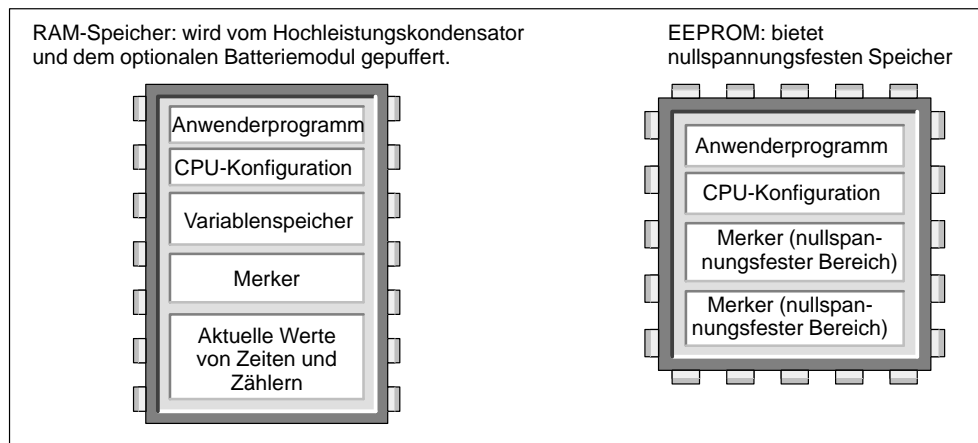


Bild 7-11 Speicherbereiche einer S7-200 CPU

Laden Ihres Programms in die und aus der CPU

Ihr Programm besteht aus drei Komponenten: dem Anwenderprogramm, dem Datenbaustein (optional) und der Konfiguration der CPU (optional). Beim Laden des Programms in die CPU werden diese Komponenten im RAM-Speicher der CPU abgelegt (siehe Bild 7-12). Außerdem kopiert die CPU das Anwenderprogramm, den Datenbaustein (DB1) und die CPU-Konfiguration automatisch in den EEPROM, damit die Komponenten dort nullspannungsfest gespeichert werden.

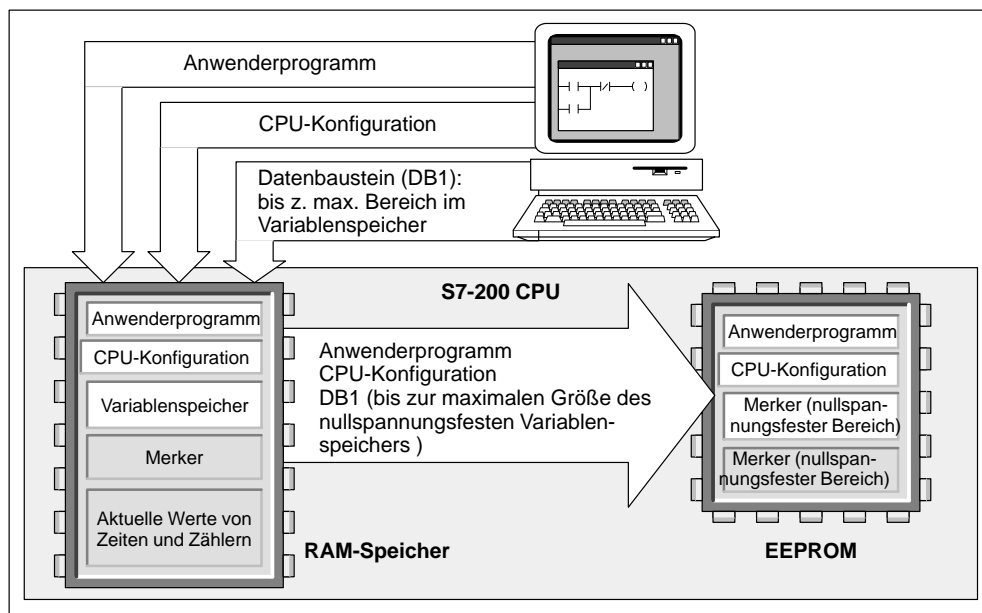


Bild 7-12 Laden der Programmkomponenten in die CPU

Wenn Sie ein Programm aus der CPU in Ihren PC bzw. Ihr PG laden (siehe Bild 7-13), dann werden das Anwenderprogramm und die CPU-Konfiguration aus dem RAM-Speicher in Ihrem Computer geladen. Wenn Sie den Datenbaustein aus der CPU laden, dann wird der nullspannungsfeste Bereich des Datenbausteins (im EEPROM gespeichert) mit dem eventuell vorhandenen restlichen Datenbaustein, der sich im RAM-Speicher befindetet, zusammengeführt. Ihrem Computer wird anschließend der vollständige Datenbaustein übermittelt. Die Größe des nullspannungsfesten Bereichs im Variablenspeicher richtet sich nach Ihrer CPU (siehe Abschnitt 10.1).

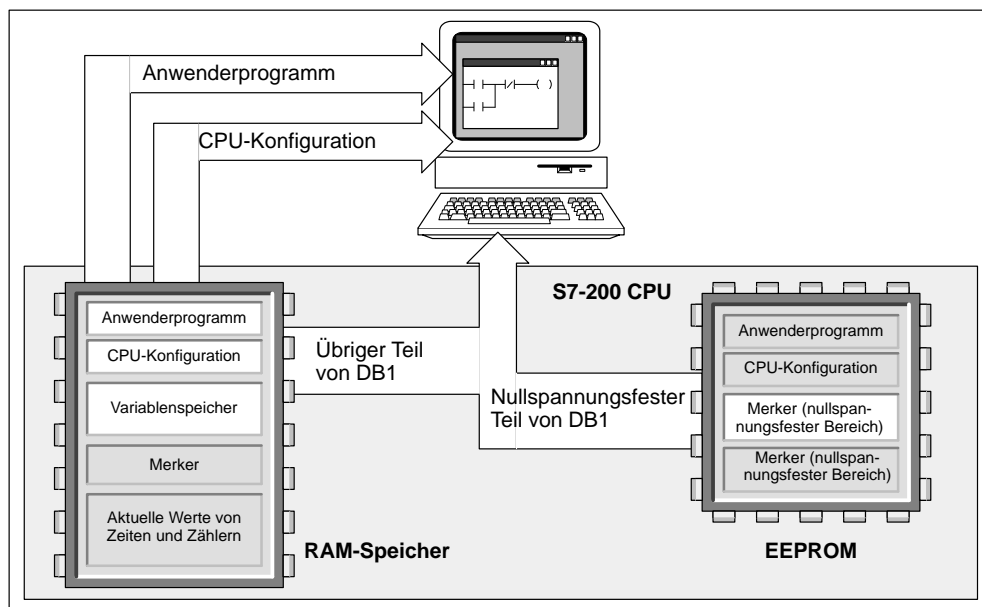


Bild 7-13 Laden der Programmkomponenten aus der CPU

Automatisches Speichern von Merkern (M) bei Spannungsverlust

Die ersten 14 Bytes im Speicherbereich der Merker (MB0 bis MB13) werden bei Spannungsverlust nullspannungsfest im EEPROM gespeichert, sofern Sie zuvor als remanent definiert wurden. Die CPU überträgt die remanenten Bereiche der Merker in den EEPROM (siehe Bild 7-14).

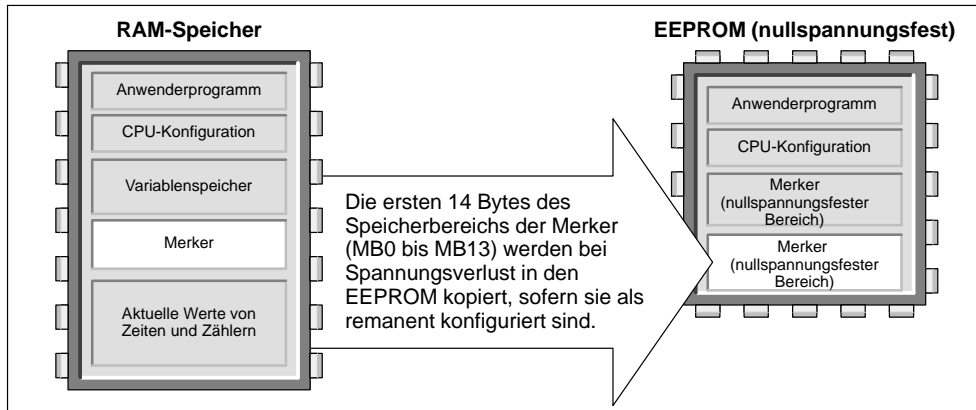


Bild 7-14 Speichern von Merkern im EEPROM bei Spannungsverlust

Zurückholen des Speichers beim Anlauf

Beim Einschalten der Spannungsversorgung holt die CPU das Anwenderprogramm und die CPU-Konfiguration aus dem EEPROM zurück (siehe Bild 7-15).

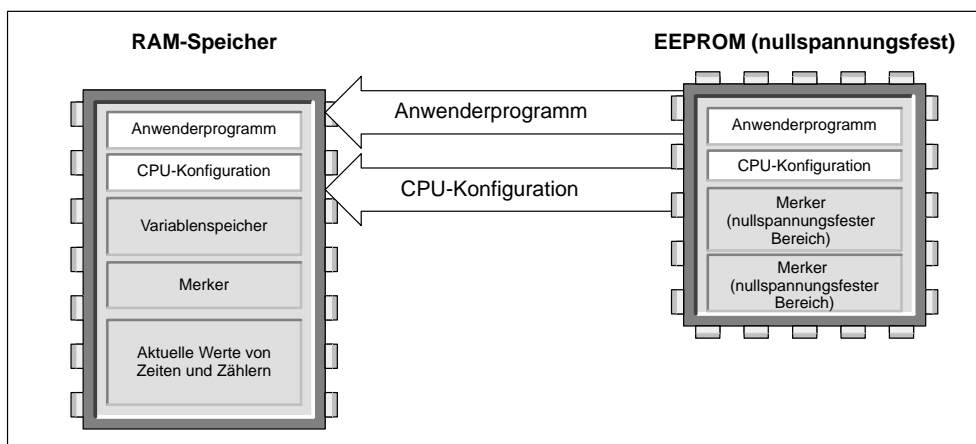


Bild 7-15 Zurückholen des Anwenderprogramms und der Systemkonfiguration beim Anlauf

Beim Anlauf prüft die CPU den RAM-Speicher daraufhin, ob der Hochleistungskondensator die Daten fehlerfrei gepuffert hat. Ist dies der Fall, werden die remanenten Bereiche des RAM-Speichers nicht verändert. Die nicht remanenten Bereiche des Variablenspeichers werden aus dem entsprechenden nullspannungsfesten Bereich des Variablenspeichers im EEPROM zurückgeholt (siehe Bild 7-16).

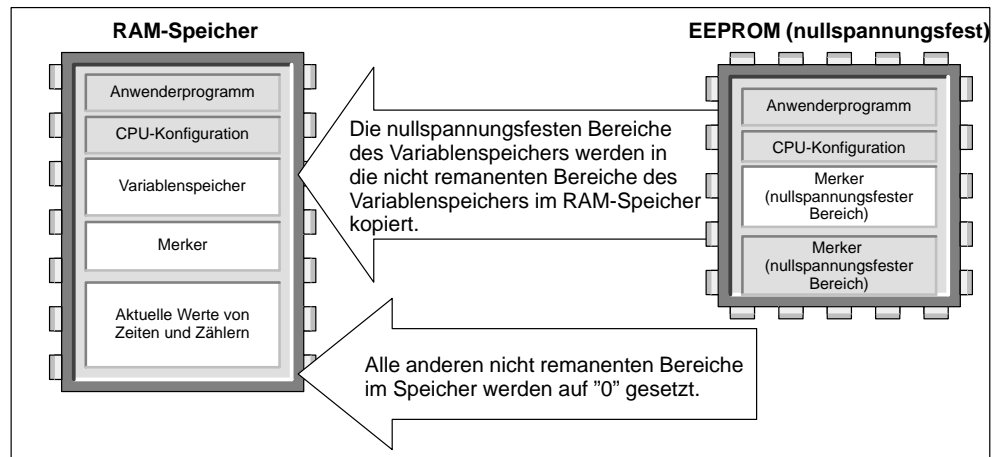


Bild 7-16 Zurückholen von Daten beim Anlauf (Daten wurden im RAM-Speicher gepuffert)

Konnte der Inhalt des RAM-Speichers nicht gepuffert werden (z. B. durch extrem langen Spannungsverlust), setzt die CPU den RAM-Speicher zurück (und zwar die remanenten und die nicht remanenten Bereiche) und setzt im ersten Zyklus nach dem Anlauf den Sondermerker Remanente Daten verloren (SM0.2). Die Daten im nullspannungsfesten EEPROM werden dann in den RAM-Speicher kopiert (siehe Bild 7-17).

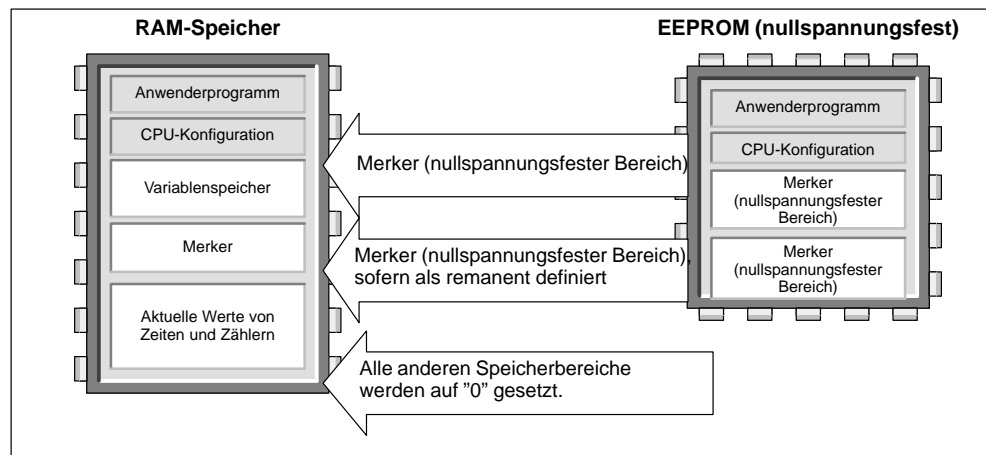


Bild 7-17 Zurückholen von Daten beim Anlauf (Daten wurden nicht im RAM-Speicher gepuffert)

Definieren von remanenten Bereichen im Speicher

Sie können maximal sechs Bereiche als remanent definieren und die Speicherbereiche auswählen, die Sie bei Spannungsverlust puffern möchten (siehe Bild 7-18). Für die folgenden Speicherbereiche können Sie bestimmte Adreßbereiche als remanent definieren: V, M, Z und T. Bei den Zeiten können nur remanente Zeiten (TONR) gepuffert werden.

Hinweis

Bei Zeiten und Zählern können nur die aktuellen Werte gepuffert werden. Zeit- und Zählerbits sind nicht remanent.

Möchten Sie bestimmte Speicherbereiche als remanent definieren, wählen Sie den Menübefehl **CPU ► Konfigurieren** und öffnen anschließend das Register "Remanente Bereiche". Bild 7-18 zeigt das Dialogfeld zum Definieren von remanenten Bereichen. Wenn Sie sich die voreingestellten remanenten Bereiche Ihrer CPU anzeigen lassen möchten, wählen Sie die Schaltfläche **Voreinstellungen**.

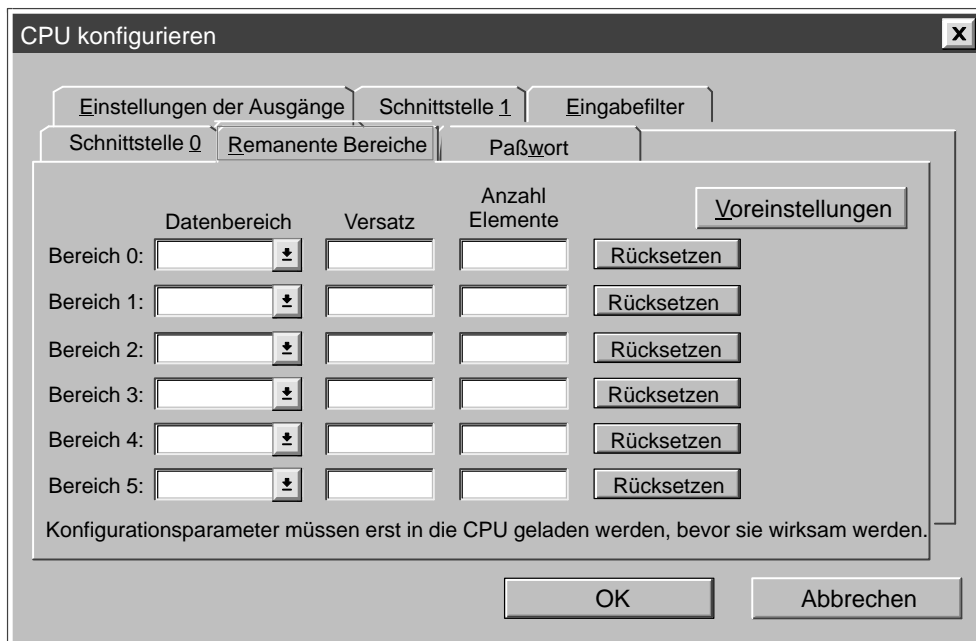


Bild 7-18 Konfigurieren der remanenten Bereiche im Speicher der CPU

7.4 Nullspannungsfestes Speichern von Daten mit Ihrem Programm

Sie können einen Wert (Byte, Wort oder Doppelwort), der sich im Variablenspeicher befindet, im EEPROM speichern. Auf diese Weise können Sie einen beliebigen Wert des Variablenspeichers im nullspannungsfesten Bereich des Variablenspeichers ablegen.

Eine Operation zum Speichern im EEPROM verlängert die Zykluszeit um ca. 15 ms bis 20 ms. Wenn Sie einen Wert in den nullspannungsfesten Bereich im EEPROM schreiben, wird der vorherige Wert an dieser Adresse überschrieben.

Hinweis

Eine Operation zum Speichern im EEPROM aktualisiert nicht die Daten im Speichermodul.

Kopieren von Variablenspeicher in den EEPROM

Sondermerker-Byte 31 (SMB31) und Sondermerker-Wort 32 (SMW32) weisen die CPU an, einen Wert des Variablenspeichers in den nullspannungsfesten Bereich des Variablenspeichers im EEPROM zu kopieren. Bild 7-19 zeigt das Format von SMB31 und SMW32. Gehen Sie folgendermaßen vor, wenn Sie die CPU so programmieren möchten, daß sie einen bestimmten Wert in den Variablenspeicher schreibt:

1. Laden Sie die Adresse des Werts im Variablenspeicher, den Sie nullspannungsfest speichern möchten, in SMW32.
2. Laden Sie die Größe der Daten in SM31.0 und SM31.1 (siehe Bild 7-19.)
3. Setzen Sie SM31.7 auf "1".

Am Ende eines jeden Zyklus prüft die CPU SM31.7. Ist SM31.7 = 1, dann wird der angegebene Wert im EEPROM gespeichert. Die Operation ist beendet, wenn die CPU SM31.7 auf "0" zurücksetzt. Ändern Sie den Wert im Variablenspeicher solange nicht, bis die Operation ausgeführt ist.

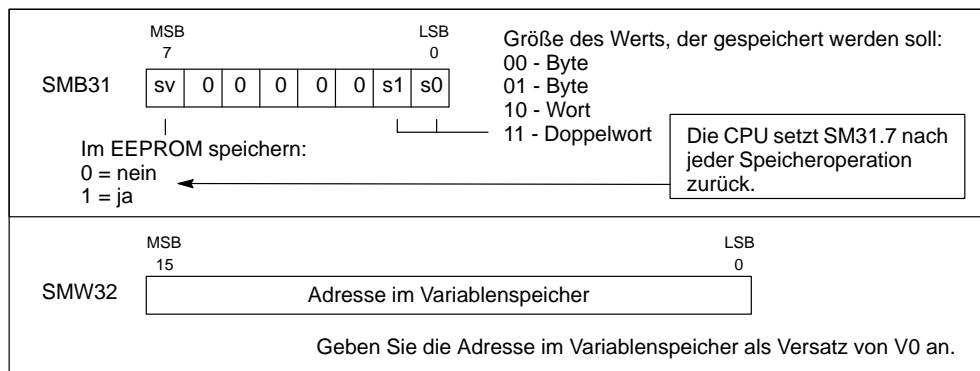


Bild 7-19 Format von SMB31 und SMW32

Einschränken des programmierten Speicherns in den EEPROM

Da die Anzahl der Speicheroperationen für den EEPROM begrenzt ist (mindestens 100.000, typischerweise 1.000.000), sollten Sie darauf achten, daß nur wichtige Werte im EEPROM gespeichert werden. Andernfalls kann durch einen Ausfall des EEPROM auch die CPU ausfallen. Typischerweise führen Sie Speicheroperationen für den EEPROM nur bei besonderen Ereignissen aus. Diese Ereignisse treten recht selten auf.

Beträgt die Zykluszeit der S7-200 beispielsweise 50 ms und ein Wert würde einmal pro Zyklus im EEPROM gespeichert werden, dann würde der EEPROM mindestens 5000 Sekunden halten, d.h. weniger als einundeinhalb Stunden. Würde der Wert allerdings nur einmal pro Stunde gespeichert werden, dann würde der EEPROM bereits mindestens 11 Jahre halten.

7.5 Speichern Ihres Programms im Speichermodul

Einige CPUs unterstützen ein optionales Speichermodul, das einen steckbaren EEPROM für Ihr Programm darstellt. Sie können das Speichermodul wie eine Diskette verwenden. Die CPU speichert folgende Komponenten im Speichermodul:

- Anwenderprogramm
- Daten des nullspannungsfesten Variablenspeichers im EEPROM
- CPU-Konfiguration

Informationen zu dem Speichermodul für Ihre CPU entnehmen Sie Anhang A.

Kopieren in das Speichermodul

Sie können Ihr Programm nur beim Anlauf der CPU mit gestecktem Speichermodul aus dem RAM-Speicher in das Modul kopieren.



Vorsicht

Elektrostatische Entladungen können das Speichermodul oder den für das Modul vorgesehenen Schacht in der CPU beschädigen.

Sie müssen auf einem leitfähigen, geerdeten Boden stehen und/oder ein geerdetes Armband tragen, wenn Sie mit dem Speichermodul arbeiten. Das Speichermodul ist in einem leitfähigen Behälter aufzubewahren.

Sie können das Speichermodul stecken oder ziehen, während die CPU eingeschaltet ist. Zum Stecken des Speichermoduls entfernen Sie den Schutzstreifen von dem Schacht für das Speichermodul. Der Schacht befindet sich unter der Abdeckklappe der CPU. Stecken Sie dann das Speichermodul in den Schacht. Das Speichermodul ist so geformt, daß es nur in eine Richtung in den Schacht gesteckt werden kann. Haben Sie das Speichermodul installiert, können Sie das Programm folgendermaßen in das Modul kopieren:

1. Sofern Sie das Programm noch nicht in die CPU geladen haben, laden Sie es jetzt.
2. Wählen Sie den Menübefehl **CPU ► Speichermodul**, um das Programm in das Speichermodul zu kopieren. In Bild 7-20 sehen Sie die Komponenten des CPU-Speichers, die im Speichermodul abgelegt werden.
3. Ziehen Sie das Speichermodul wieder aus dem Schacht (optional).

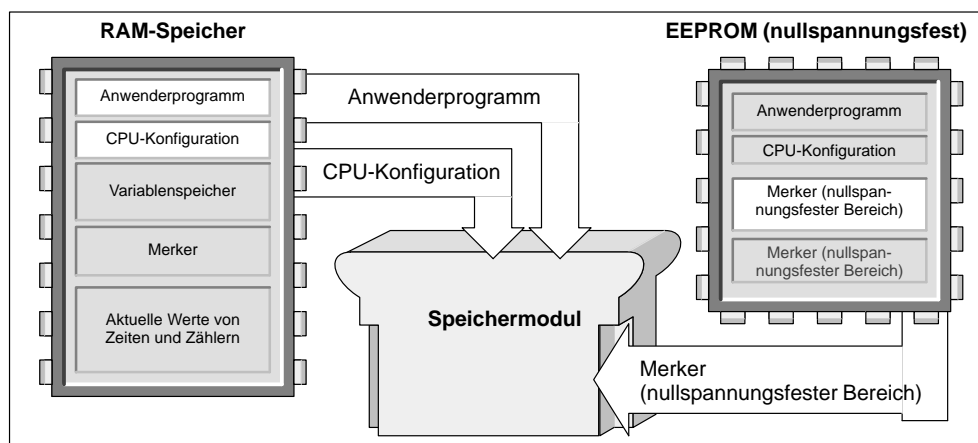


Bild 7-20 Kopieren des CPU-Speichers in das Speichermodul

Zurückholen des Programms und des Speichers aus dem Speichermodul

Wenn Sie ein Programm aus dem Speichermodul in die CPU laden möchten, müssen Sie die CPU ausschalten und mit gestecktem Speichermodul wieder einschalten. Die CPU führt nach dem Einschalten die folgenden Aufgaben aus, sofern ein Speichermodul gesteckt ist (siehe Bild 7-21):

- Der RAM-Speicher wird gelöscht.
- Der Inhalt des Speichermoduls wird in den RAM-Speicher kopiert.
- Das Anwenderprogramm, die CPU-Konfiguration und der Bereich des Variablenspeichers (bis zur maximalen Größe des remanenten Bereichs im Variablenspeicher) werden in den nullspannungsfesten EEPROM kopiert.

Hinweis

Lassen Sie die CPU mit leerem Speichermodul oder mit einem Speichermodul, das von einer anderen CPU-Variante programmiert wurde, anlaufen, tritt ein Fehler auf. Ziehen Sie dann das Speichermodul und lassen Sie die CPU erneut anlaufen. Anschließend können Sie das Speichermodul stecken und programmieren.

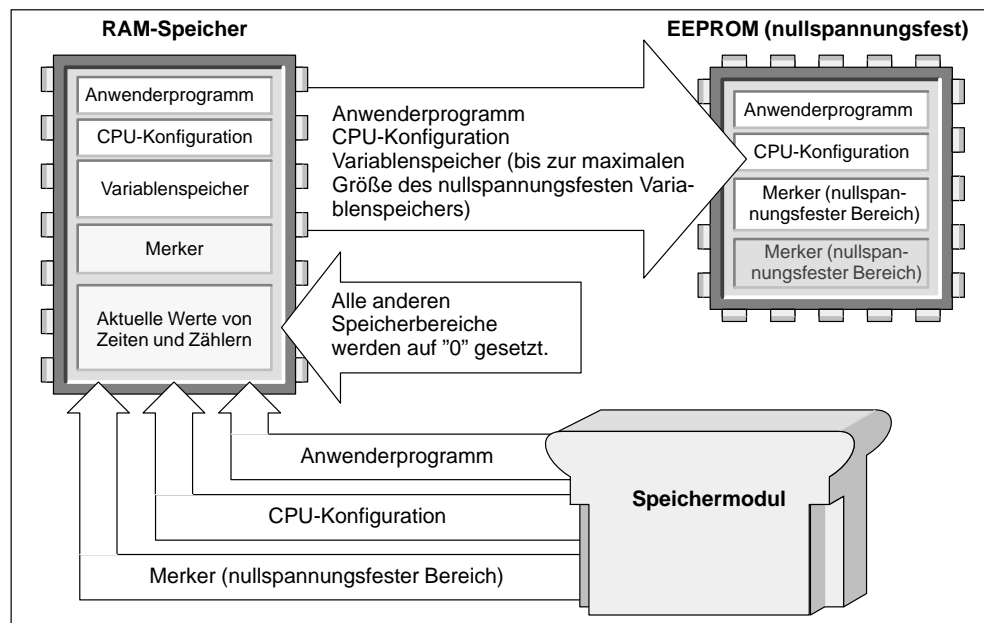


Bild 7-21 Wiederherstellen des Speichers beim Anlauf (mit gestecktem Speichermodul)

Steuerung über Ein- und Ausgänge

Über die Ein- und Ausgänge wird das System gesteuert. Die Eingänge überwachen die Signale der Feldgeräte (z. B. Sensoren und Schalter) und die Ausgänge steuern Pumpen, Motoren oder andere Geräte in Ihrem Prozeß. Es stehen Ihnen integrierte Ein- und Ausgänge (am Zentralgerät) und erweiterte Ein- und Ausgänge (an den Erweiterungsmodulen) zur Verfügung. Die S7-200 CPU bietet außerdem schnelle Ausgangsfunktionen.

Kapitelübersicht

Abschnitt	Beschreibung	Seite
8.1	Integrierte und erweiterte Ein- und Ausgänge	8-2
8.2	Konfigurieren von Eingabefiltern zur Rauschunterdrückung	8-5
8.3	Konfigurieren der Signalzustände der Ausgänge	8-6
8.4	Schnelle Ein- und Ausgänge	8-7
8.5	Analogpotentiometer	8-8

8.1 Integrierte und erweiterte Ein- und Ausgänge

Über die Ein- und Ausgänge wird das System gesteuert. Die Eingänge überwachen die Signale der Feldgeräte (z. B. Sensoren und Schalter) und die Ausgänge steuern Pumpen, Motoren oder andere Geräte in Ihrem Prozeß. Es stehen Ihnen integrierte Ein- und Ausgänge (am Zentralgerät) und erweiterte Ein- und Ausgänge (an den Erweiterungsmodulen) zur Verfügung:

- Die S7-200 CPU verfügt über eine bestimmte Anzahl integrierter Digitalein- und Digitalausgänge. Ausführliche Informationen zu der Anzahl der integrierten Ein- und Ausgänge Ihrer CPU entnehmen Sie den Datenblättern in Anhang A.
- Die S7-200 CPU unterstützt zusätzliche digitale und analoge Erweiterungsmodule. Ausführliche Informationen zu den verschiedenen Erweiterungsmodulen entnehmen Sie den Datenblättern in Anhang A.

Adressierung der integrierten und erweiterten Ein- und Ausgänge

Die integrierten Ein- und Ausgänge auf dem Zentralgerät (CPU) verfügen über feste Adressen. Sie können Ihre CPU um Ein- und Ausgänge erweitern, indem Sie an der rechten Seite der CPU Erweiterungsmodule anschließen. Die Adressen der Ein- und Ausgänge auf dem Erweiterungsmodul richten sich nach der Art der Ein- und Ausgänge und bei mehreren Modulen gleichen Typs auch nach der Anordnung des Moduls. Ein Ausgabemodul beispielsweise beeinflusst nicht die Adressen der Eingänge auf einem Eingabemodul und umgekehrt. Die Adressen der Ein- und Ausgänge von Analog- und Digitalmodulen sind ebenfalls voneinander unabhängig.

Für Digitalerweiterungsmodule sind Abschnitte von jeweils acht Bits (ein Byte) im Prozeßabbild vorgesehen. Ist auf einem Modul nicht für jedes Bit eines reservierten Byte ein physikalischer Ein- bzw. Ausgang vorhanden, dann gehen die freien Bits verloren und können keinem folgenden Erweiterungsmodul dieser CPU zugeordnet werden. Die freien Bits der reservierten Bytes von Ausgabemodulen können wie interne Merker verwendet werden. Bei Eingabemodulen werden die freien Bits in jedem Aktualisierungszyklus der Eingänge auf Null gesetzt und können deshalb nicht als interne Merker verwendet werden.

Die Ein- und Ausgänge von Analogerweiterungsmodulen werden immer in Zweierschritten zugeordnet. Ist auf einem Modul nicht für jeden dieser Ein- und Ausgänge ein physikalischer Ein- bzw. Ausgang vorhanden, gehen die Ein- und Ausgänge verloren und können keinem folgenden Erweiterungsmodul dieser CPU zugeordnet werden. Da Analogein- und Analogausgänge nicht über ein Abbild im Speicher verfügen, können die freien Analogein- und Analogausgänge nicht verwendet werden. Auf Analogein- und Analogausgänge wird immer sofort bei der Bearbeitung der Operation zugegriffen.

Beispiele für integrierte und erweiterte Ein- und Ausgänge

Die Bilder 8-1, 8-2 und 8-3 zeigen Beispiele, an denen Sie sehen, wie verschiedene Hardware-Konfigurationen sich auf die Adressen der Ein- und Ausgänge auswirken. Beachten Sie, daß einige der Konfigurationen Lücken in der Reihenfolge der Adressen aufweisen, die von Ihrem Programm nicht verwendet werden können, während andere freie Adressen von Ein- und Ausgängen wie interne Merker nutzbar sind.

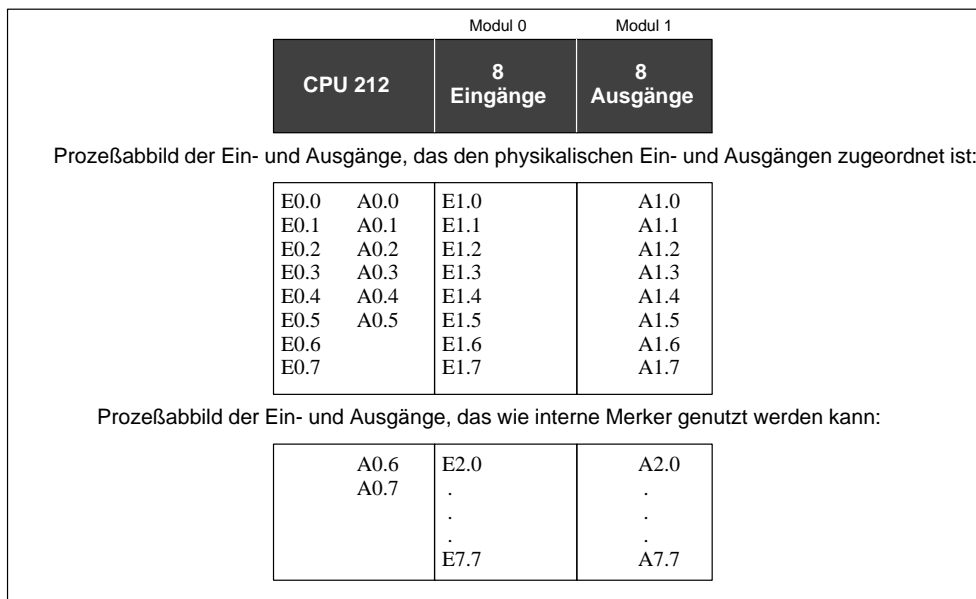


Bild 8-1 Beispiele für Adressen von Ein- und Ausgängen bei einer CPU 212

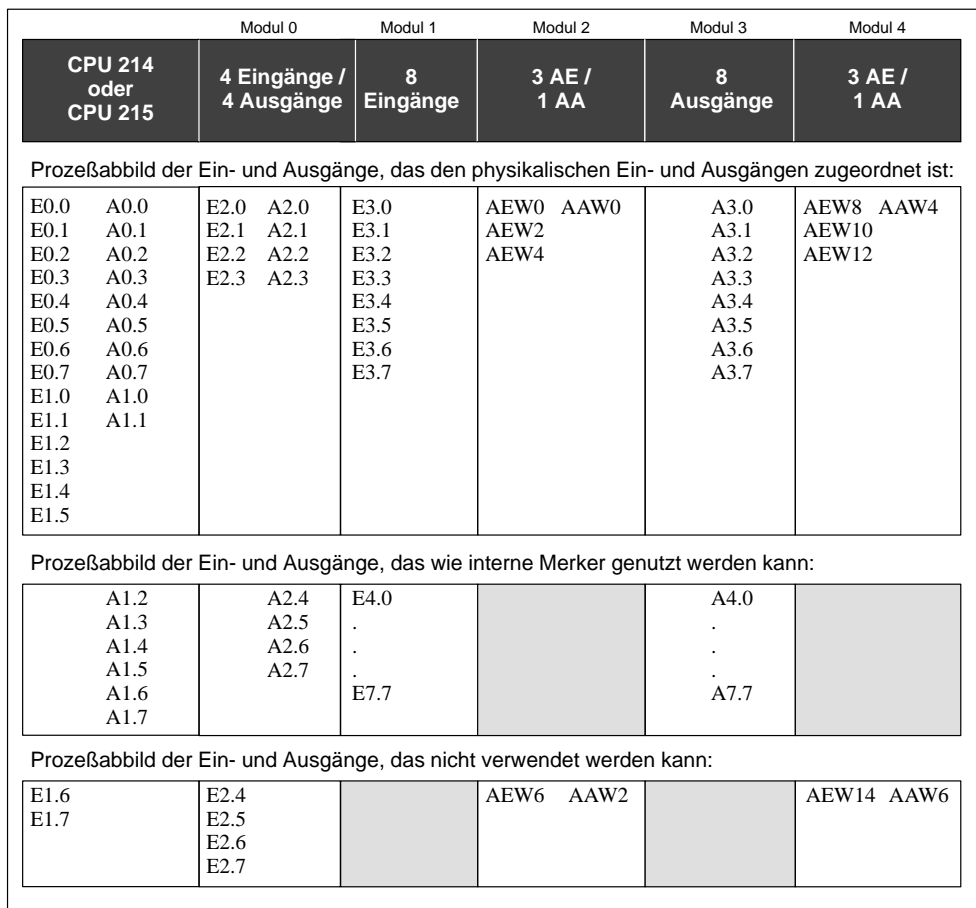


Bild 8-2 Beispiele für Adressen von Ein- und Ausgängen bei einer CPU 214 oder einer CPU 215

CPU 216		Modul 0		Modul 1		Modul 2	
		8 Eingänge/ 8 Ausgänge		16 Eingänge/ 16 Ausgänge		16 Eingänge/ 16 Ausgänge	
Prozeßabbild der Ein- und Ausgänge, das den physikalischen Ein- und Ausgängen zugeordnet ist:							
E0.0	A0.0	E3.0	A2.0	E4.0	A3.0	E6.0	A5.0
E0.1	A0.1	E3.1	A2.1	E4.1	A3.1	E6.1	A5.1
E0.2	A0.2	E3.2	A2.2	E4.2	A3.2	E6.2	A5.2
E0.3	A0.3	E3.3	A2.3	E4.3	A3.3	E6.3	A5.3
E0.4	A0.4	E3.4	A2.4	E4.4	A3.4	E6.4	A5.4
E0.5	A0.5	E3.5	A2.5	E4.5	A3.5	E6.5	A5.5
E0.6	A0.6	E3.6	A2.6	E4.6	A3.6	E6.6	A5.6
E0.7	A0.7	E3.7	A2.7	E4.7	A3.7	E6.7	A5.7
E1.0	A1.0			E5.0	A4.0	E7.0	A6.0
E1.1	A1.1			E5.1	A4.1	E7.1	A6.1
E1.2	A1.2			E5.2	A4.2	E7.2	A6.2
E1.3	A1.3			E5.3	A4.3	E7.3	A6.3
E1.4	A1.4			E5.4	A4.4	E7.4	A6.4
E1.5	A1.5			E5.5	A4.5	E7.5	A6.5
E1.6	A1.6			E5.6	A4.6	E7.6	A6.6
E1.7	A1.7			E5.7	A4.7	E7.7	A6.7
E2.0							
E2.1							
E2.2							
E2.3							
E2.4							
E2.5							
E2.6							
E2.7							

Bild 8-3 Beispiele für Adressen von Ein- und Ausgängen bei einer CPU 216

8.2 Konfigurieren von Eingabefiltern zur Rauschunterdrückung

Sie können für einige der S7-200 CPUs einen Eingabefilter auswählen, der für die physikalischen Eingänge eine Verzögerungszeit (einstellbar zwischen 0,2 ms bis 8,7 ms) definiert. (Ausführliche Informationen zu Ihrer CPU entnehmen Sie Anhang A.) Diese Verzögerungszeit wird zu der üblichen Antwortzeit von jeweils vier Eingängen addiert (siehe Bild 8-4). Die Verzögerung dient dazu, ein Rauschen, welches unbeabsichtigte Signalzustandsänderungen an den Eingängen verursachen kann, in der Eingangsverdrahtung zu filtern.

Der Eingabefilter ist Teil der CPU-Konfigurationsdaten, die in den Speicher der CPU geladen und dort abgelegt werden.

Wählen Sie den Menübefehl **CPU ► Konfigurieren...** und öffnen Sie das Register "Eingabefilter". Stellen Sie hier die Verzögerungszeiten für die Eingabefilter ein.

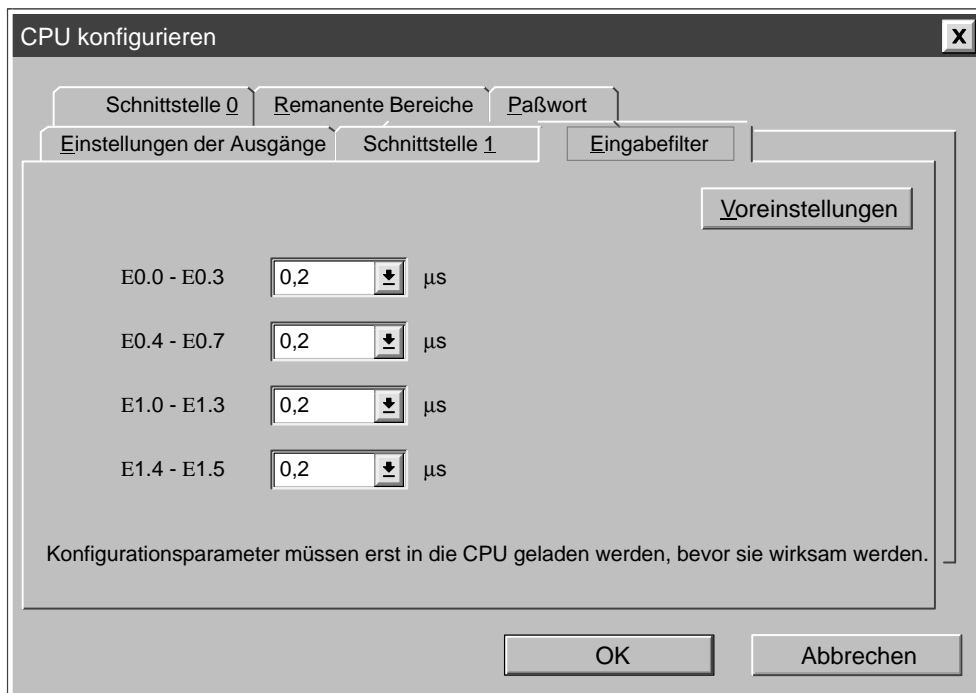


Bild 8-4 Konfigurieren der Eingabefilter zur Rauschunterdrückung

8.3 Konfigurieren der Signalzustände der Ausgänge

Mit der S7-200 CPU können Sie die Signalzustände der digitalen Ausgänge bei einem Übergang in den Betriebszustand STOP auf bestimmte Werte setzen, oder Sie können die Ausgänge in genau dem Zustand einfrieren, in dem sie sich beim Übergang in STOP befanden.

Die Einstellungen der Ausgänge sind Teil der CPU-Konfigurationsdaten für das System, die in den Speicher der CPU geladen und dort abgelegt werden.

Die Konfiguration der Ausgangswerte ist nur bei digitalen Ausgängen möglich. Analogausgänge werden bei einem Übergang in den Betriebszustand STOP eingefroren. Dies geschieht, weil Ihr Programm für die Aktualisierung der Analogausgänge verantwortlich ist. Die CPU aktualisiert die Analogein- und Analogausgänge als Systemfunktion. Für die Analogein- und Analogausgänge wird im Speicher der CPU kein Abbild abgelegt.

Wählen Sie den Menübefehl **CPU ► Konfigurieren...** und klicken Sie auf das Register "Einstellungen der Ausgänge" (siehe Bild 8-5). In diesem Dialogfeld haben Sie folgende zwei Möglichkeiten zum Konfigurieren der Ausgänge:

- Möchten Sie die Ausgänge in ihrem letzten Zustand einfrieren, aktualisieren Sie das Kontrollkästchen "Ausgänge einfrieren" und bestätigen Sie mit "OK".
- Wenn Sie die definierten Werte in die Ausgänge kopieren möchten, dann geben Sie jetzt die Einstellungen für die Ausgänge an. Klicken Sie für jeden Ausgang, den Sie bei einem Übergang in STOP auf "1" setzen möchten, das zugehörige Kontrollkästchen an. Abschließend bestätigen Sie Ihre Einstellungen mit "OK".

Standardmäßig kopiert die CPU diese Einstellungen in die Ausgänge. Voreingestellt ist für alle Ausgänge der Zustand "0".

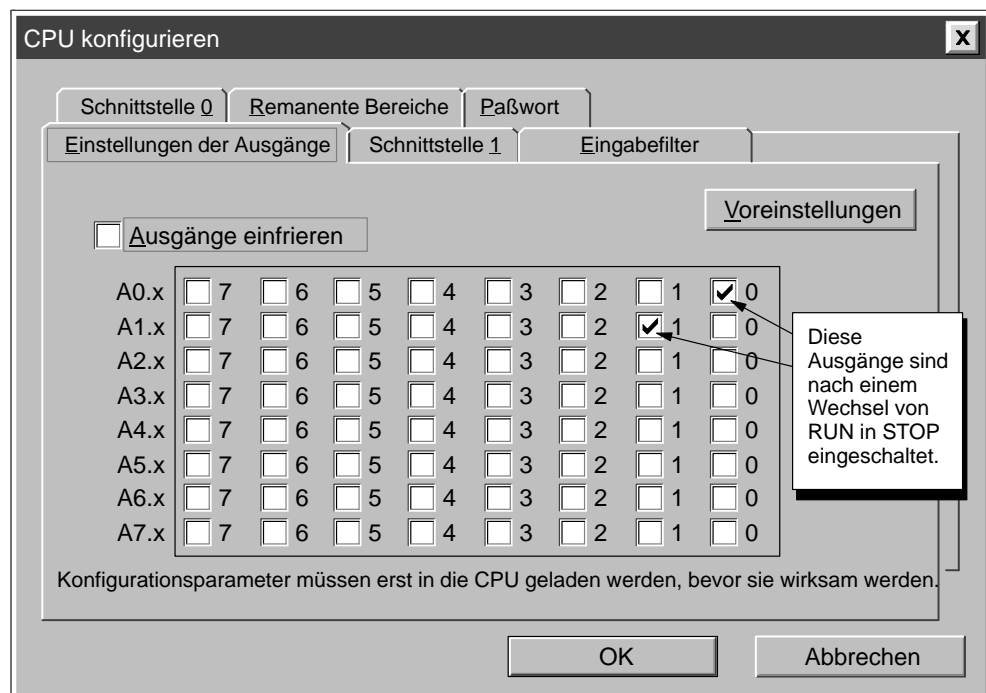


Bild 8-5 Konfigurieren der Signalzustände für die Ausgänge

8.4 Schnelle Ein- und Ausgänge

Ihre S7-200 CPU verfügt über schnelle Ein- und Ausgänge, mit denen Sie schnelle Ereignisse steuern können. Ausführliche Informationen zu den schnellen Ein- und Ausgängen Ihrer CPU entnehmen Sie den Datenblättern in Anhang A.

Schnelle Zähler

Schnelle Zähler zählen schnelle Ereignisse, die bei den Zyklusraten der S7-200 CPUs nicht gesteuert werden können. Ihre S7-200 CPU unterstützt einen schnellen Software- und zwei schnelle Hardware-Zähler (je nach Ihrer CPU):

- HSC0 ist ein Vorwärts-/Rückwärtszähler, der einen Takteingang unterstützt. Das Programm steuert die Zählrichtung (vorwärts oder rückwärts) über ein richtungsteuerndes Bit. Die maximale Zählgeschwindigkeit dieses Zählers beträgt 2 kHz.
- HSC1 und HSC2 sind universelle Hardware-Zähler, die für eine von zwölf verschiedenen Zählerarten konfiguriert werden können. Die verschiedenen Zählerarten sind in Tabelle 10-5 aufgeführt. Die maximale Zählgeschwindigkeit der Zähler HSC1 und HSC2 richtet sich nach Ihrer CPU (siehe Anhang A).

Jeder Zähler verfügt über besondere Eingänge, die Funktionen wie Taktgeber, Richtungssteuerung, Rücksetzen und Starten unterstützen. Bei A/B-Zählern können Sie einfache oder vierfache Zählgeschwindigkeiten auswählen. HSC1 und HSC2 sind vollkommen unabhängig voneinander und beeinflussen keine anderen schnellen Operationen. Beide Zähler laufen mit maximaler Geschwindigkeit, ohne sich gegenseitig zu beeinträchtigen.

Ausführliche Informationen zu den schnellen Zählern entnehmen Sie Abschnitt 10.5.

Schnelle Impulsausgänge

Die S7-200 CPU unterstützt schnelle Impulsausgänge. Bei diesen Zentralgeräten erzeugen die Ausgänge A0.0 und A0.1 entweder Impulsfolgen (PTO) oder sie steuern die Impulsdauermodulation (PWM).

- Die Funktion Impulsfolge stellt einen Rechteckausgang (50% relative Einschaltdauer) für eine bestimmte Anzahl von Impulsen und eine festgelegte Zykluszeit zur Verfügung. Die Anzahl der Impulse kann zwischen 1 und 4.294.967.295 liegen. Die Zykluszeit kann in Mikrosekunden von 250 μ s bis 65.535 μ s oder in Millisekunden von 2 ms bis 65.535 ms angegeben werden. Eine ungerade Mikro- oder Millisekundenzahl (z.B. 75 ms) ruft eine Verzerrung der relativen Einschaltdauer hervor.
- Die Funktion Impulsdauermodulation bietet Ihnen eine feste Zykluszeit mit variabler relativer Einschaltdauer. Zykluszeit und Impulsdauer können in Mikro- oder Millisekunden angegeben werden. Die Zykluszeit liegt zwischen 250 μ s und 65.535 μ s oder zwischen 2 ms und 65.535 ms. Die Impulsdauer liegt zwischen 0 μ s und 65.535 μ s oder zwischen 0 ms und 65.535 ms. Sind Impulsdauer und Zykluszeit gleich, dann beträgt die relative Einschaltdauer 100% und der Ausgang ist ständig eingeschaltet. Ist die Impulsdauer Null, dann beträgt auch die relative Einschaltdauer 0% und der Ausgang wird ausgeschaltet.

Ausführliche Informationen zu den Impulsausgängen entnehmen Sie Abschnitt 10.5.

8.5 Analogpotentiometer

Ihre S7-200 CPU verfügt über ein oder zwei Analogpotentiometer (unter der Abdeckklappe der CPU). Mit diesen Potentiometern können Sie Werte, die in Bytes von Sondermerkern (SMB28 und SMB29) gespeichert sind, erhöhen oder verringern. Diese schreibgeschützten Werte können dem Programm für eine Reihe von Funktionen dienen, z. B. beim Aktualisieren von aktuellen Werten von Zeiten und Zählern, beim Eingeben oder Ändern von voreingestellten Werten oder beim Einstellen von Grenzwerten.

SMB28 speichert den Digitalwert, der die Position des Analogpotentiometers 0 darstellt. SMB29 speichert den Digitalwert, der die Position des Analogpotentiometers 1 darstellt. Die Analogpotentiometer verfügen über einen Nennbereich von 0 bis 255 und über einen garantierten Bereich von 10 bis 200.

Mit einem kleinen Schraubendreher stellen Sie die Analogpotentiometer ein: drehen Sie rechts herum, wenn Sie den Wert des Analogpotentiometers erhöhen möchten, drehen Sie links herum, um den Wert zu verringern. Bild 8-6 zeigt ein Beispielprogramm für die Verwendung eines Analogpotentiometers.

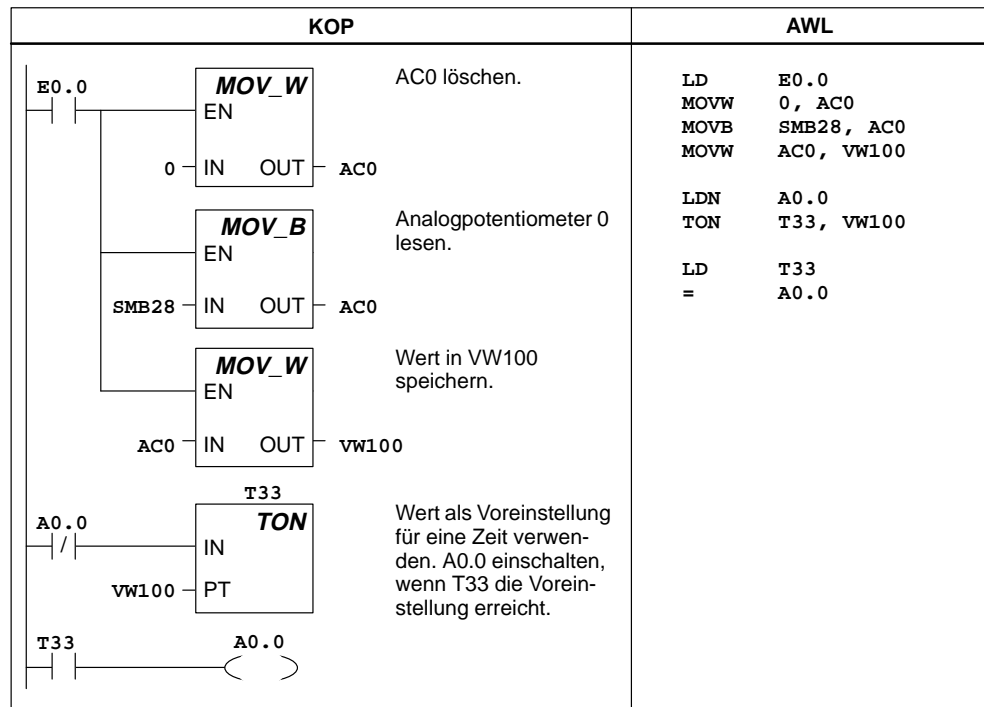


Bild 8-6 Beispiel für ein Analogpotentiometer

Kommunikation im Netz mit einer S7-200 CPU

9

Die S7-200 CPUs unterstützen verschiedene Kommunikationsarten, einschließlich der folgenden:

- PPI-Kommunikation (Punkt-zu-Punkt)
- Kommunikation im Netz mit mehreren Mastern
- Kommunikation im Netz der dezentralen Peripherie (DP)

Kapitelübersicht

Abschnitt	Beschreibung	Seite
9.1	Kommunikationsfähigkeiten der S7-200 CPU	9-2
9.2	Komponenten für die Kommunikation im Netz	9-6
9.3	Datenkommunikation über PC/PPI-Kabel	9-9
9.4	Datenkommunikation über MPI- oder CP-Baugruppe	9-13
9.5	DP-Standardkommunikation (dezentrale Peripherie)	9-15
9.6	Leistungsfähigkeit des Netzes	9-28

9.1 Kommunikationsfähigkeiten der S7-200 CPU

Protokolle für die Kommunikation im Netz

Die S7-200 CPUs unterstützen verschiedene Kommunikationsfähigkeiten. Je nachdem, mit welcher S7-200 CPU Sie arbeiten, kann Ihr Netz eines oder mehrere der folgenden Kommunikationsprotokolle unterstützen:

- Punkt-zu-Punkt-Schnittstelle (PPI)
- Mehrpunktfähige Schnittstelle (MPI)
- PROFIBUS-DP

Tabelle 9-1 zeigt ausführliche Angaben hierzu.

Tabelle 9-1 Kommunikationsfähigkeiten der S7-200 CPUs

CPU	Schnitt.	PPI-Slave	PPI-Master	PROFIBUS-DP-Slave	MPI-Slave	Frei programm. Kommunik.	Baudrate
212	0	Ja	Nein	Nein	Nein	Ja	9600 Baud, 19.200 Baud
214	0	Ja	Ja	Nein	Nein	Ja	9600 Baud, 19.200 Baud
215	0	Ja	Ja	Nein	Ja	Ja	9600 Baud, 19.200 Baud
	PPI, frei progr. DP	Nein	Nein	Ja	Ja	Nein	9600 Baud, 19.200 Baud, 93.750 Baud, 187.500 Baud, 500.000 Baud, 1 MBaud, 1,5 MBaud, 3 MBaud, 6 MBaud, 12 MBaud
216	0	Ja	Ja	Nein	Ja	Ja	9600 Baud, 19.200 Baud
	1	Ja	Ja	Nein	Ja	Ja	9600 Baud, 19.200 Baud

Diese Protokolle basieren auf der Kommunikationsarchitektur des Sieben-Lagen-Modells für die Kommunikation offener Systeme (OSI). Die PPI-, MPI- und PROFIBUS-DP-Protokolle sind in einem Token-Ring-Netz implementiert, das dem PROFIBUS-Standard (Process Field Bus) gemäß der EG-Richtlinie EN 50170 entspricht.

Bei diesen Protokollen handelt es sich um asynchrone Protokolle auf Zeichenbasis mit einem Startbit, acht Datenbits, gerader Parität und einem Stopbit. Datenübertragungsblöcke in der Kommunikation umfassen besondere Start- und Stoppzeichen, Teilnehmeradressen von Quelle und Ziel, die Länge des Datenübertragungsblocks und ein Prüfsummenzeichen für die Datenintegrität. Die drei Protokolle können in einem Netz gleichzeitig aktiv sein, ohne daß sie sich gegenseitig beeinträchtigen, solange die Baudrate für alle Protokolle die gleiche ist.

Das PROFIBUS-Netz setzt verdrehte Doppelleitungen gemäß RS-485 ein. So können maximal 32 Geräte in einem Netzsegment miteinander verbunden werden. Netzsegmente können maximal 1200 m umfassen. Die tatsächliche Länge richtet sich nach der Baudrate. Netzsegmente können über Busverstärker miteinander verbunden sein, damit mehr Geräte im Netz und längere Kabelverbindungen möglich sind. Ein Netz darf mit Busverstärkern je nach Baudrate maximal 9600 m umfassen (siehe Abschnitt 9.2).

Die Protokolle definieren zwei Arten von Geräten im Netz: Master und Slaves. Master können eine Anforderung an andere Geräte im Netz auslösen. Slaves können nur auf Anforderungen von Mastern reagieren. Slaves lösen nie selbst Anforderungen aus.

Die Protokolle unterstützen 127 Adressen (0 bis 126) in einem Netz. Ein Netz darf maximal 32 Master enthalten. Jedes Gerät im Netz muß eine eindeutige Adresse haben, damit es kommunizieren kann. SIMATIC Programmiergeräte und PCs, auf denen STEP 7-Micro/WIN installiert ist, haben die Adresse 0 voreingestellt. Operator Panels, wie das TD 200, OP3 und OP7, haben die voreingestellte Adresse 1. Die Automatisierungssysteme haben die voreingestellte Adresse 2. Die voreingestellte Adresse für die DP-Schnittstelle der CPU 215 ist 126.

PPI-Protokoll

Das PPI-Protokoll ist ein Master-/Slave-Protokoll. In diesem Protokoll senden die Master (andere CPUs, SIMATIC Programmiergeräte oder Textdisplays TD 200) Anforderungen an die Slaves, worauf die Slaves reagieren. Slaves initiieren keine Meldungen, sondern warten, bis ein Master eine Anforderung sendet oder die Slaves nach einer Reaktion abfragt. Alle S7-200 CPUs dienen im Netz als Slaves.

Einige S7-200 CPUs können, wenn Sie im Anwenderprogramm den PPI-Master-Modus aktiviert haben, als Master eingesetzt werden, solange sie sich im Betriebszustand RUN befinden. (Lesen Sie hierzu die Beschreibung von SMB30 in Anhang D.) Nach dem Aktivieren des PPI-Master-Modus können Sie mit den Operationen Aus Netz lesen (NETR) und In Netz schreiben (NETW) Meldungen aus anderen CPUs lesen bzw. in andere CPUs schreiben. Beschreibungen dieser Operationen entnehmen Sie Kapitel 10. Auch wenn die S7-200 CPU als PPI-Master eingesetzt wird, reagiert sie immer noch als Slave auf Anforderungen von anderen Mastern.

In der PPI-Kommunikation gibt es keine Einschränkungen, wieviele Master mit einer beliebigen Slave-CPU kommunizieren dürfen, doch es dürfen maximal 32 Master im Netz vorhanden sein.

MPI-Protokoll

Das MPI-Protokoll ist entweder ein Master-/Master- oder ein Master-/Slave-Protokoll. Die Funktionsweise des Protokolls richtet sich nach der Geräteart. Handelt es sich beim Zielgerät um eine S7-300 CPU, dann wird die Master-/Master-Verbindung hergestellt, weil alle S7-300 CPUs Master im Netz sind. Handelt es sich beim Zielgerät um eine S7-200 CPU, dann wird die Master-/Slave-Verbindung hergestellt, weil alle S7-200 CPUs Slaves im Netz sind.

Das MPI-Protokoll stellt immer eine Verbindung zwischen den beiden Geräten her, die miteinander kommunizieren. Eine Verbindung ist wie eine "private" Verknüpfung zwischen den beiden Geräten. Ein anderer Master kann eine zwischen zwei Geräten aufgebaute Verbindung nicht stören. Ein Master kann eine Verbindung nur für kurze Zeit aufbauen oder die Verbindung kann unbegrenzt bestehen bleiben.

Da es sich bei den Verbindungen um "private" Verknüpfungen zwischen den Geräten handelt und deshalb Ressourcen in der CPU erforderlich sind, kann eine CPU nur eine begrenzte Anzahl von Verbindungen unterstützen. Tabelle 9-2 führt die Anzahl und Art der MPI-Verbindungen auf, die von den S7-200 CPUs unterstützt werden. Jede CPU reserviert einige seiner Verbindungen für SIMATIC Programmiergeräte und Operator Panels. Die reservierte Verbindung für ein SIMATIC Programmiergerät bzw. für einen PC, auf dem STEP 7-Micro/WIN installiert ist, stellt sicher, daß Sie als Anwender immer in der Lage sind, mindestens ein SIMATIC Programmiergerät an die CPU anzuschließen und so auf die CPU zuzugreifen. Einige CPUs reservieren auch eine Verbindung für ein Operator Panel. Die reservierten Verbindungen können nicht von anderen Arten von Mastern (z.B. CPUs) in Anspruch genommen werden.

Tabelle 9-2 Anzahl und Art der logischen MPI-Verbindungen bei S7-200 CPUs

CPU	Schnittstelle	Gesamtzahl der Verbindungen	Anzahl und Art der reservierten logischen Verbindungen
215	0	Vier	Zwei: Eine für ein Programmiergerät Eine für ein Operator Panel
	PPI, frei prog., DP	Sechs	Zwei: Eine für ein Programmiergerät Eine für ein Operator Panel
216	0	Vier	Zwei: Eine für ein Programmiergerät Eine für ein Operator Panel
	1	Vier	Zwei: Eine für ein Programmiergerät Eine für ein Operator Panel

Die S7-300 und S7-400 CPUs können mit den S7-200 CPUs kommunizieren, indem sie eine Verbindung zu den nicht reservierten Verbindungen der S7-200 CPU aufbauen. Die S7-300 und S7-400 CPUs können mit den Operationen XGET und XPUT Daten aus der S7-200 lesen und Daten in die S7-200 schreiben (ausführliche Informationen hierzu entnehmen Sie den Programmierhandbüchern für die S7-300 bzw. S7-400).

Hinweis

Das MPI-Protokoll kann nicht für die Kommunikation mit S7-200 CPUs eingesetzt werden, in denen der PPI-Master-Modus aktiviert ist. Das MPI-Protokoll klassifiziert diese Geräte als Master und versucht mit ihnen über ein Master-/Master-Protokoll zu kommunizieren, das von den S7-200 CPUs nicht unterstützt wird.

PROFIBUS-DP-Protokoll

Das PROFIBUS-DP-Protokoll ist für schnelle Kommunikation mit Geräten der dezentralen Peripherie ausgelegt. Es sind viele PROFIBUS-Geräte von verschiedenen Herstellern erhältlich. Bei diesen Geräten kann es sich um einfache Ein- oder Ausgabemodule bis hin zu Motorsteuergeräten und speicherprogrammierbaren Steuerungen handeln.

PROFIBUS-DP-Netze haben üblicherweise einen Master und mehrere Slave-Ein-/Ausgabegeräte. Der Master ist so konfiguriert, daß ihm bekannt ist, welche Arten von Ein-/Ausgabeslaves an welchen Adressen an das Netz angeschlossen sind. Der Master initialisiert das Netz und prüft, daß die Slaves im Netz der Konfiguration entsprechen. Der Master schreibt Ausgabedaten in die Slaves und liest die Eingabedaten kontinuierlich. Wenn ein DP-Master einen Slave erfolgreich konfiguriert hat, ist dieser Slave dem Master zugeordnet. Befindet sich ein zweiter Master im Netz, hat dieser Master nur sehr eingeschränkten Zugriff auf die Slaves, die zu dem ersten Master gehören.

Die CPU 215 verfügt über eine Schnittstelle, die als PROFIBUS-DP-Schnittstelle dient (siehe Bild 9-1). Ausführliche Informationen zur DP-Funktionalität der CPU 215 entnehmen Sie dem Abschnitt 9.5.

Anwenderdefinierte Protokolle (frei programmierbare Kommunikation)

In der frei programmierbaren Kommunikation kann das Anwenderprogramm die Kommunikationsschnittstelle der S7-200 CPU steuern. Sie können in der frei programmierbaren Kommunikation anwenderdefinierte Kommunikationsprotokolle implementieren und so Schnittstellen zu vielen Arten von intelligenten Geräten herstellen.

Das Anwenderprogramm steuert die Funktionsweise der Kommunikationsschnittstelle über Empfangsinterrupts, Sendeinterrupts und die Operationen Meldung aus Zwischenspeicher übertragen (XMT) und Meldung in Zwischenspeicher empfangen (RCV). Das Kommunikationsprotokoll wird in der frei programmierbaren Kommunikation nur vom Anwenderprogramm gesteuert. Die frei programmierbare Kommunikation wird über SMB30 (Schnittstelle 0) und SMB130 (Schnittstelle 1) freigegeben und ist nur aktiv, wenn sich die CPU im Betriebszustand RUN befindet. Geht die CPU in den Betriebszustand STOP, wird die frei programmierbare Kommunikation beendet und die Kommunikationsschnittstelle kehrt zur üblichen Funktionsweise über das PPI-Protokoll zurück. Ausführliche Informationen zum Arbeiten mit der frei programmierbaren Kommunikation entnehmen Sie dem Abschnitt 10.14.

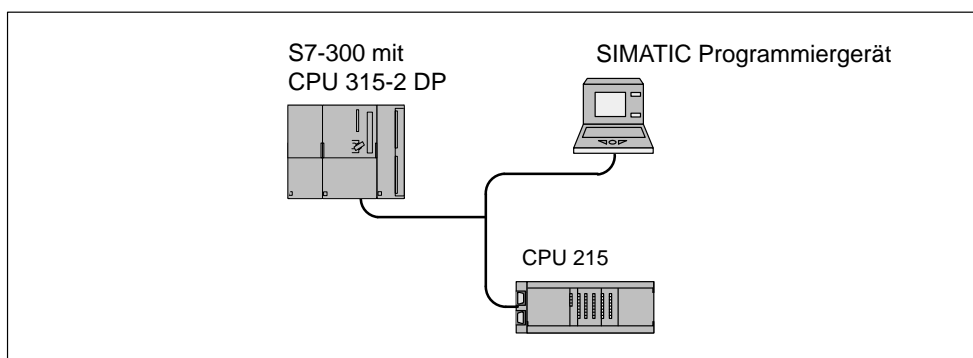


Bild 9-1 CPU 215: Über DP-Schnittstelle verbunden mit einer S7-300 CPU und einem Programmiergerät

9.2 Komponenten für die Kommunikation im Netz

Über die Kommunikationsschnittstelle der S7-200 können Sie sie an einen Netzbus anschließen. Im folgenden werden die Schnittstelle, die Steckverbinder für den Netzbus, die Netzkabel und die Busverstärker, mit denen Sie das Netz erweitern, beschrieben.

Kommunikationsschnittstelle

Die Kommunikationsschnittstellen der S7-200 CPUs arbeiten mit RS-485-Signalpegeln und verfügen über 9polige Sub-D-Stecker gemäß dem PROFIBUS-Standard nach der EG-Richtlinie EN 50170. Bild 9-2 zeigt den Stecker, der die physikalische Verbindung zur Kommunikationsschnittstelle herstellt. Tabelle 9-3 beschreibt die Signale.

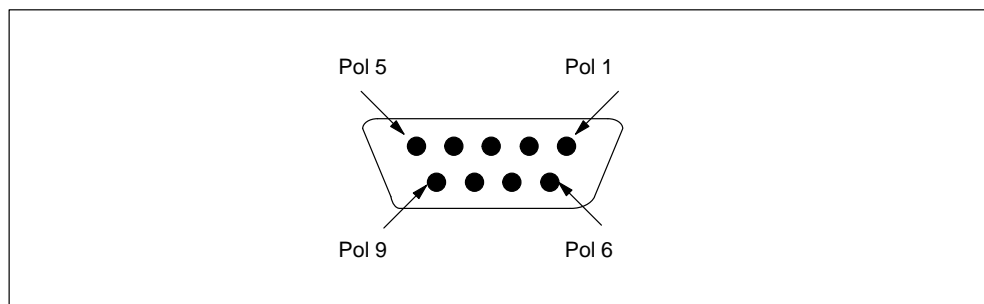


Bild 9-2 Anschlußbelegung der Kommunikationsschnittstelle der S7-200 CPU

Tabelle 9-3 Anschlußbelegung der Kommunikationsschnittstelle der S7-200

Pol	Bezeichnung PROFIBUS	Schnittstelle 0 und Schnittstelle 1	DP-Schnittstelle
1	Schirmung	Logischer Leiter	Logischer Leiter
2	24-V-Rückleiter	Logischer Leiter	Logischer Leiter
3	RS-485 Signal B	RS-485 Signal B	RS-485 Signal B
4	Anforderung zum Senden	Keine Verbindung	Anforderung zum Senden ¹
5	5-V-Rückleiter	Logischer Leiter	Galvanisch getrennter +5-V-Rückleiter ²
6	+5 V	+5 V, strombegrenzt durch 100-Ω-Widerstand in Reihe	+5 V, galvanisch getrennt, 90 mA
7	+24 V	+24 V	+24 V
8	RS-485 Signal A	RS-485 Signal A	RS-485 Signal A
9	-/-	Keine Verbindung	Keine Verbindung
Gehäuse des Steckverbinders	Schirmung	Logischer Leiter (CPU 212/214) Erdungsanschluß (CPU 215/216)	Erdungsanschluß

¹ $V_{OH} = 3,5 \text{ V}$, $1,6 \text{ mA}$, $V_{OL} = 0,6 \text{ V}$, $1,6 \text{ mA}$, $\text{Signal} = V_{OH}$, wenn die CPU sendet.

² Signale A, B und Anforderung zum Senden sind bei der DP-Schnittstelle galvanisch von der Logik der CPU getrennt und haben als Referenzspannung den galvanisch getrennten 5-V-Rückleiter.

Busanschlußstecker

Siemens bietet zwei Arten von Busanschlußsteckern, mit denen Sie mehrere Geräte schnell und einfach an ein Netz anschließen können. Beide Busanschlußstecker verfügen über zwei Sätze Anschlußschrauben, mit denen Sie die Eingangs- und Ausgangskabel für das Netz befestigen können. Beide Stecker verfügen außerdem über Schalter, mit denen Sie einen Abschlußwiderstand zuschalten können. Einer der beiden Stecker bietet nur eine Verbindung zur CPU. Der andere Stecker verfügt auch über eine Verbindung zur Programmierschnittstelle (siehe Bild 9-3). Die Bestellnummern entnehmen Sie Anhang G.

Der Busanschlußstecker mit Verbindung zur Programmierschnittstelle läßt es zu, daß ein SIMATIC Programmiergerät oder ein Operator Panel an das Netz angeschlossen wird, ohne daß dadurch bestehende Netzverbindungen gestört werden. Der Busanschlußstecker mit Verbindung zur Programmierschnittstelle leitet alle Signale aus der CPU an die Programmierschnittstelle weiter. Dieser Busanschlußstecker dient dazu, Geräte (z.B. das TD 200 oder ein OP3), die von der CPU gespeist werden, anzuschließen. Die Signale an den Spannungspolen im Stecker der Kommunikationsschnittstelle der CPU werden an die Programmierschnittstelle weitergeleitet.



Vorsicht

Wenn Sie Geräte miteinander verbinden, die nicht die gleiche Bezugsspannung haben, kann dies unerwünschte Ströme im Verbindungskabel hervorrufen.

Diese unerwünschten Ströme können Kommunikationsfehler verursachen oder Sachschaden in den Geräten hervorrufen.

Stellen Sie sicher, daß alle Geräte, die Sie über ein Kommunikationskabel miteinander verbinden, entweder den gleichen Bezugsleiter im Stromkreis haben oder galvanisch getrennt sind, damit keine unerwünschten Ströme auftreten. Ausführliche Informationen entnehmen Sie den Richtlinien für Erdung und Bezugsspannung galvanisch getrennter Stromkreise in Abschnitt 2.3.

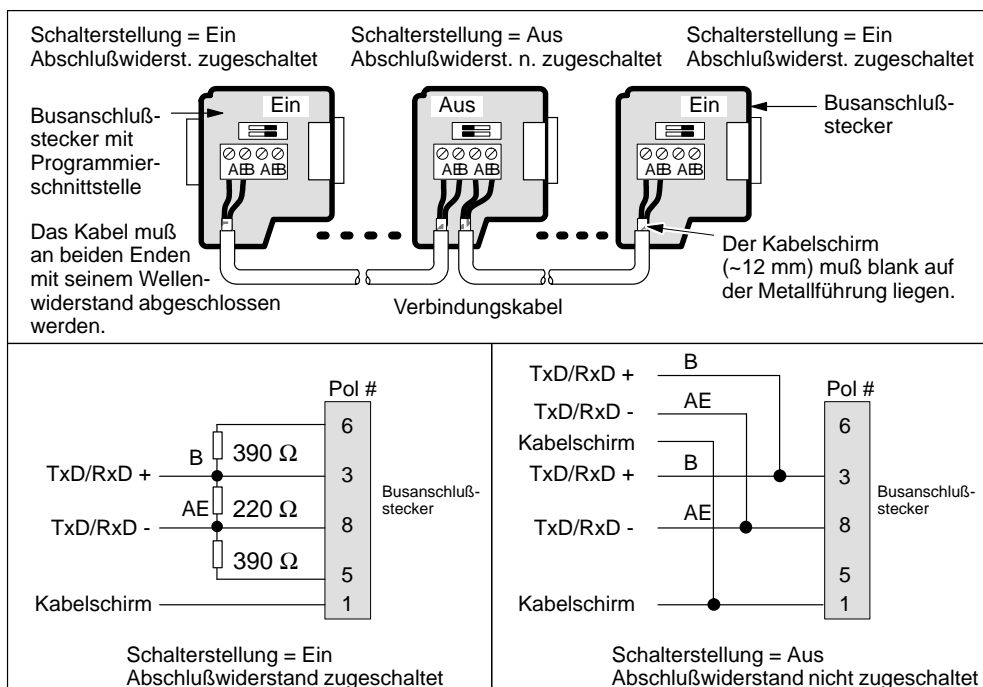


Bild 9-3 Verbindungskabel mit Abschlußwiderstand

Kabel für ein PROFIBUS-Netz

Tabelle 9-4 führt die allgemeinen technischen Daten für ein PROFIBUS-Netzkabel auf. Sie entnehmen Anhang G die Bestellnummer für Siemens PROFIBUS-Kabel, die diese Anforderungen erfüllen.

Tabelle 9-4 Allgemeine technische Daten eines PROFIBUS-Netzkabels

Leistungsmerkmale	Technische Daten
Art	Geschirmtes, verdrehtes Leiterpaar
Leiterquerschnitt	0,22 mm ² oder größer
Kabelkapazität	< 60 pF/m
Nennimpedanz	100 Ω bis 120 Ω

Die maximale Länge eines PROFIBUS-Netzsegments richtet sich nach der Baudrate und der Art des eingesetzten Kabels. Tabelle 9-5 führt die maximale Kabellänge pro Segment auf, die bei Einsatz eines Kabels zutrifft, das den in Tabelle 9-4 angegebenen technischen Daten entspricht.

Tabelle 9-5 Maximale Kabellänge eines Segments in einem PROFIBUS-Netz

Übertragungsgeschwindigkeit	Maximale Kabellänge für ein Segment
9600 Baud bis 93.750 Baud	1200 m
187.500 Baud	1000 m
500.000 Baud	400 m
1,5 MBaud	200 m
3 MBaud bis 12 MBaud	100 m

Busverstärker im Netz

Siemens bietet Busverstärker, mit denen Sie PROFIBUS-Netzsegmente verbinden können (siehe Bild 9-4). Durch den Einsatz von Busverstärkern kann die Gesamtausdehnung des Netzes erweitert und/oder zusätzliche Geräte im Netz aufgenommen werden. In PROFIBUS-Netzen sind maximal 32 Geräte und eine maximale Kabellänge von 1200 m pro Segment bei einer Baudrate von 9600 Baud zulässig. Pro Busverstärker können Sie bei 9600 Baud 32 weitere Geräte an das Netz anschließen und die Kabellänge um 1200 m verlängern. Sie dürfen maximal 9 Busverstärker in einem Netz einsetzen. Jeder Busverstärker schließt das Netzsegment mit einem Abschlußwiderstand ab. Die Bestellnummern entnehmen Sie Anhang G.

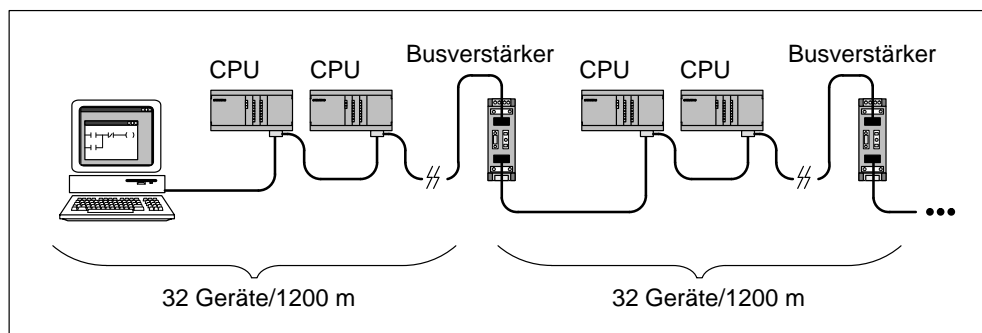


Bild 9-4 Netz mit Busverstärkern

9.3 Datenkommunikation über PC/PPI-Kabel

PC/PPI-Kabel

Die Kommunikationsschnittstellen eines Personal Computers sind im allgemeinen Schnittstellen, die mit dem RS-232-Standard kompatibel sind. Die Kommunikationsschnittstellen der S7-200 CPU arbeiten mit dem RS-485-Standard, damit mehrere Geräte an das gleiche Netz angeschlossen werden können. Mit dem PC/PPI-Kabel können Sie die RS-232-Schnittstelle eines Personal Computers mit der RS-485-Schnittstelle einer S7-200 CPU verbinden (siehe Bild 9-5). Sie können mit dem PC/PPI-Kabel auch die Kommunikationsschnittstelle einer S7-200 CPU an andere RS-232-kompatible Geräte anschließen.

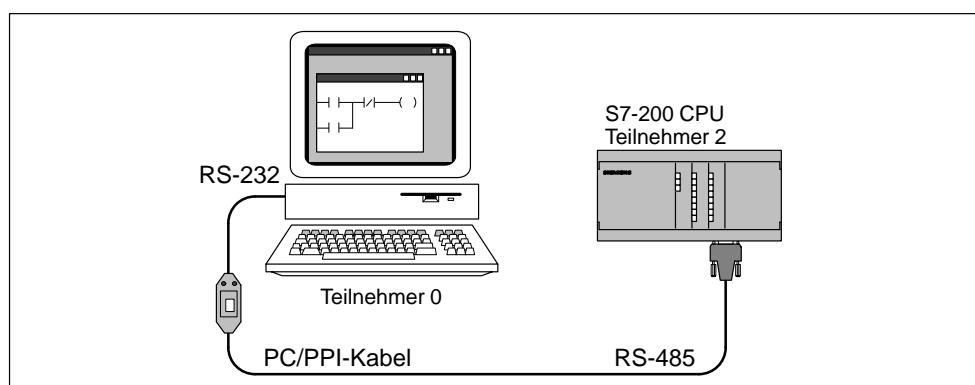


Bild 9-5 Kommunizieren mit einer S7-200 CPU über ein PC/PPI-Kabel

Einsetzen des PC/PPI-Kabels mit STEP 7-Micro/WIN

STEP 7-Micro/WIN kann über das PC/PPI-Kabel mit einer oder mehreren S7-200 CPUs kommunizieren (siehe Bild 9-6). Wenn Sie mit STEP 7-Micro/WIN arbeiten, müssen Sie darauf achten, daß die für das PC/PPI-Kabel ausgewählte Baudrate der für das Netz richtigen Baudrate entspricht. STEP 7-Micro/WIN unterstützt nur 9600 Baud und 19.200 Baud.

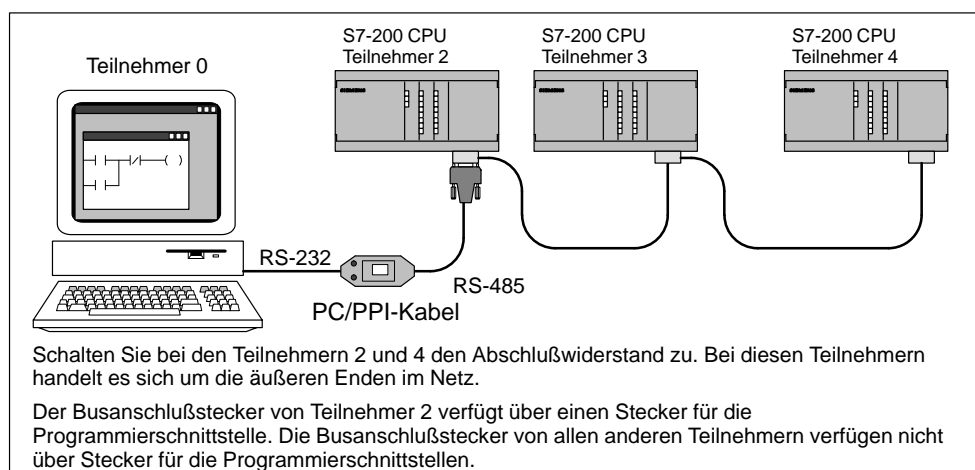


Bild 9-6 Kommunizieren mit einer CPU im Netz über PC/PPI-Kabel

Für die Kommunikation mit S7-200 CPUs ist in STEP 7-Micro/WIN standardmäßig das PPI-Protokoll für mehrere Master eingestellt. Bei diesem Protokoll dürfen neben STEP 7-Micro/WIN gleichzeitig andere Master (Textdisplays TD 200 und Operator Panels) im Netz vorhanden sein. Sie geben diesen Modus frei, indem Sie im Dialogfeld "PG/PC-Schnittstelle einstellen" im Register "Eigenschaften - PC/PPI-Kabel" das Kontrollkästchen "Multi-Master-Netzwerk" aktivieren (siehe Abschnitt 3.3).

STEP 7-Micro/WIN unterstützt auch ein PPI-Protokoll für nur einen Master. Wenn Sie mit dem Protokoll für einen Master arbeiten, geht STEP 7-Micro/WIN davon aus, daß es der einzige Master im Netz ist und ist deshalb nicht darauf vorbereitet, andere Master im Netz zuzulassen. Das Protokoll für einen Master sollten Sie einsetzen, wenn Sie über Modems senden oder in stark gestörten Netzen arbeiten. Sie wählen den Modus für einen Master aus, indem Sie im Dialogfeld "PG/PC-Schnittstelle einstellen" im Register "Eigenschaften - PC/PPI-Kabel" das Kontrollkästchen "Multi-Master-Netzwerk" deaktivieren (siehe Abschnitt 3.3).

Die technischen Daten des PC/PPI-Kabels entnehmen Sie dem Abschnitt A.40. Die Bestellnummer des Kabels entnehmen Sie Anhang G.

Einsetzen des PC/PPI-Kabels mit anderen Geräten und in der frei programmierbaren Kommunikation

Mit Hilfe des PC/PPI-Kabels und der frei programmierbaren Kommunikation können Sie die S7-200 CPUs an viele Geräte anschließen, die mit dem RS-232-Standard kompatibel sind.

Das PC/PPI-Kabel unterstützt Baudraten zwischen 600 Baud und 38.400 Baud. Mit Hilfe der DIP-Schalter am PC/PPI-Kabel stellen Sie das Kabel auf die gewünschte Baudrate ein. Tabelle 9-6 zeigt die Schalterstellungen für die entsprechenden Baudraten.

Tabelle 9-6 Schalterstellungen am PC/PPI-Kabel für die Baudraten

Baudrate	Schalter (1 = hoch/ein)
38400	0000
19.200	0010
9600	0100
4800	0110
2400	1000
1200	1010
600	1100

Die RS-232-Schnittstelle des PC/PPI-Kabels gehört zu den Datenübertragungseinrichtungen (DÜE). Die einzigen Signale an dieser Schnittstelle sind die zum Senden von Daten, zum Empfangen von Daten sowie Masse. Tabelle 9-7 zeigt die Pole und deren Funktion für die RS-232-Schnittstelle des PC/PPI-Kabels. Das PC/PPI-Kabel arbeitet nicht mit den RS-232-Steuersignalen wie Sendeaufforderung (RTS) und Sendebereitschaft (CTS).

Tabelle 9-7 PC/PPI-Kabel: Pole der RS-232-Schnittstelle

Pol	Funktion
2	Daten empfangen (von DÜE)
3	Daten senden (von DEE an DÜE)
5	Masse

Das PC/PPI-Kabel ist im Sendemodus, wenn Daten von der RS-232-Schnittstelle an die RS-485-Schnittstelle gesendet werden. Das PC/PPI-Kabel ist im Empfangsmodus, wenn es sich im Ruhezustand befindet oder wenn Daten von der RS-485-Schnittstelle an die RS-232-Schnittstelle gesendet werden. Das Kabel schaltet sofort vom Empfangs- in den Sendemodus um, wenn es Zeichen in der RS-232-Sendeleitung erkennt. Das Kabel kehrt zum Empfangsmodus zurück, wenn sich die RS-232-Sendeleitung für eine bestimmte Zeit, die als Umschaltezeit des Kabels definiert ist, im Ruhezustand befindet. Diese Zeit richtet sich nach der mit den DIP-Schaltern auf dem Kabel eingestellten Baudrate (siehe Tabelle 9-8).

Arbeiten Sie mit dem PC/PPI-Kabel in einem System, in dem auch die frei programmierbare Kommunikation eingesetzt wird, dann muß das Anwenderprogramm in der S7-200 die Umschaltezeit aus folgenden Gründen beachten:

- Die S7-200 CPU reagiert auf Meldungen, die von dem RS-232-Gerät gesendet werden.
Nachdem vom RS-232-Gerät eine Anforderung empfangen wurde, muß das Senden der Reaktionsmeldung von der S7-200 CPU für einen Zeitraum, der größer oder gleich der Umschaltezeit des Kabels ist, verzögert werden.
- Das RS-232-Gerät reagiert auf Meldungen, die von der S7-200 CPU gesendet werden.
Nachdem vom RS-232-Gerät eine Reaktion empfangen wurde, muß das Senden der nächsten Anforderung von der S7-200 CPU für einen Zeitraum, der größer oder gleich der Umschaltezeit des Kabels ist, verzögert werden.

In beiden Situationen erhält das PC/PPI-Kabel durch die Verzögerung genügend Zeit, um vom Sendemodus in den Empfangsmodus umzuschalten, damit die Daten von der RS-485-Schnittstelle an die RS-232-Schnittstelle gesendet werden können.

Tabelle 9-8 Umschaltezeit des PC/PPI-Kabels (vom Sende- zum Empfangsmodus)

Baudrate	Umschaltezeit (in ms)
38.400	1
19.200	1
9600	2
4800	4
2400	7
1200	14
600	28

Arbeiten mit Modem und PC/PPI-Kabel

Sie können mit dem PC/PPI-Kabel die RS-232-Kommunikationsschnittstelle eines Modems an eine S7-200 CPU anschließen. Modems arbeiten üblicherweise mit RS-232-Steuersignalen (wie RTS, CTS und DTR), damit der PC das Modem steuern kann. Das PC/PPI-Kabel arbeitet nicht mit diesen Signalen. Wenn Sie also mit einem Modem und PC/PPI-Kabel arbeiten, muß das Modem so konfiguriert sein, daß es ohne diese Signale arbeiten kann. Zumindest müssen Sie das Modem so konfigurieren, daß es die Signale RTS und DTR ignoriert. Die Befehle, mit denen Sie Ihr Modem konfigurieren können, entnehmen Sie dem Handbuch zu Ihrem Modem.

Wenn Sie ein Modem an ein PC/PPI-Kabel anschließen, müssen Sie das Modem über einen Null-Modem-Adapter an die RS-232-Schnittstelle des PC/PPI-Kabels anschließen. Modems sind Datenübertragungseinrichtungen (DÜE). Die RS-232-Schnittstelle des PC/PPI-Kabels gehört zu auch den Datenübertragungseinrichtungen (DÜE). Wenn Sie zwei Geräte mit gleicher Klassifizierung (beide DÜE) verbinden, müssen Sie die Pole zum Senden und Empfangen von Daten tauschen. Der Null-Modem-Adapter vertauscht die Leitungen für Senden und Empfangen. In Bild 9-7 sehen Sie einen typischen Aufbau und die Anschlußbelegung für den Null-Modem-Adapter.

Wenn Sie in STEP 7-Micro/WIN ein Modem einsetzen, muß es sich um ein vollduplexes Modem handeln, das 11-Bit-Zeichen unterstützt. Dem Abschnitt 3.3 entnehmen Sie ausführliche Informationen zum Arbeiten in STEP 7-Micro/WIN mit einem Modem. Wenn Sie mit Modem und Protokoll für frei programmierbare Kommunikation arbeiten, können Sie ein beliebiges Modem einsetzen, das die Zeichengröße des Protokolls unterstützt.

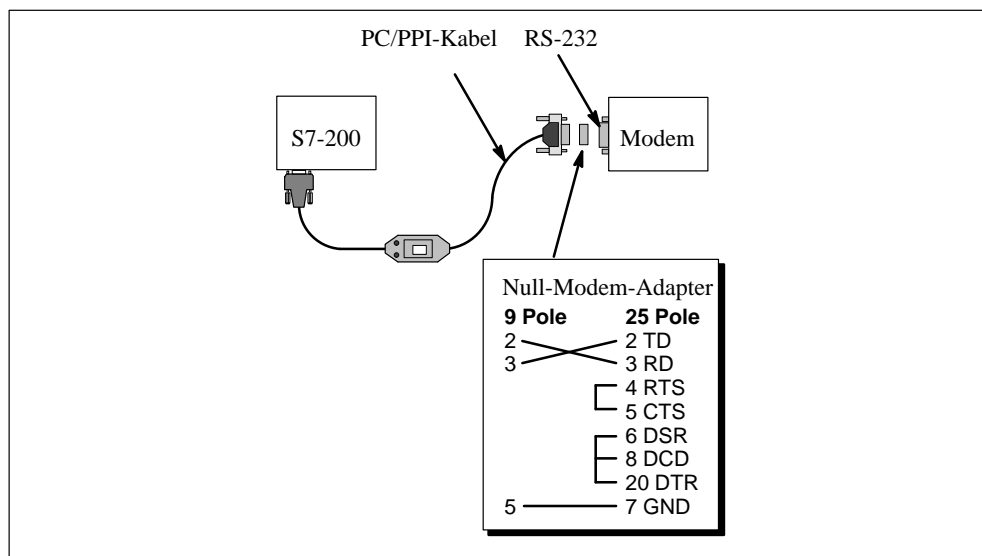


Bild 9-7 Modem mit Null-Modem-Adapter

9.4 Datenkommunikation über MPI- oder CP-Baugruppe

Siemens bietet verschiedene Netz-Schnittstellenkarten, die Sie in Ihren Computer oder in Ihr SIMATIC Programmiergerät integrieren können. Mit Hilfe dieser Baugruppen kann der PC oder das SIMATIC Programmiergerät als Master im Netz eingesetzt werden. Die Baugruppen enthalten spezifische Hardware, die Ihren PC bzw. das Programmiergerät darin unterstützen, ein Netz mit mehreren Mastern zu verwalten. Außerdem können so unterschiedliche Protokolle und mehrere Baudraten unterstützt werden (siehe Tabelle 9-9).

Tabelle 9-9 Baugruppen, die an ein Netz mit mehreren Mastern angeschlossen werden können

Name	Datentyp	Unterstützte Betriebssysteme	Kommentare
MPI	Short AT ISA oder in PG integriert	MS-DOS Windows 3.1x	Unterstützt PPI-Protokoll, 9600 Baud und 19.200 Baud
		Windows 95 Windows NT	Unterstützt PPI-, ¹ MPI- und PROFIBUS-DP-Protokolle, 9600 Baud bis 1,5 MBaud für PCs und PGs
CP 5411	Short AT ISA	Windows 95 Windows NT	Unterstützt PPI-, ¹ MPI- und PROFIBUS-DP-Protokolle, 9600 Baud bis 12 MBaud für PCs und PGs
CP 5511	PCMCIA, Typ II Plug & Play-Hardware	Windows 95 Windows NT	Unterstützt PPI-, ¹ MPI- und PROFIBUS-DP-Protokolle, 9600 Baud bis 12 MBaud für Notebook-PCs
CP 5611	Short PCI Plug & Play-Hardware	Windows 95 Windows NT	Unterstützt PPI-, ¹ MPI- und PROFIBUS-DP-Protokolle, 9600 Baud bis 12 MBaud für PCs

¹ Nur bei 9600 Baud oder 19.200 Baud

Die Baugruppe und das Protokoll richten Sie im Dialogfeld "PG/PC-Schnittstelle einstellen" in STEP 7-Micro/WIN oder in der Systemsteuerung unter Windows ein (siehe Abschnitt 3.3).

Wenn Sie in Windows 95 oder in Windows NT arbeiten, können Sie ein beliebiges Protokoll (PPI, MPI oder PROFIBUS) für die Baugruppen auswählen. Im allgemeinen gilt, daß Sie zum Kommunizieren mit S7-200 CPUs bei 9600 Baud oder 19.200 Baud das PPI-Protokoll einstellen sollten. Ausnahme hierzu ist die CPU 215. Wenn Sie mit dieser CPU über die DP-Schnittstelle kommunizieren, müssen Sie das MPI-Protokoll einstellen. Die DP-Schnittstelle der CPU 215 DP unterstützt Baudraten von 9600 Baud bis 12 MBaud. Diese Schnittstelle legt automatisch die Baudrate des Masters (CP- oder MPI-Baugruppe) fest und synchronisiert die eigene Baudrate mit der des Masters.

Beide Baugruppen verfügen über eine RS-485-Schnittstelle zum Anschließen an das PROFIBUS-Netz. Die Baugruppe CP 5511 PCMCIA besitzt einen Adapter, der die 9polige D-Schnittstelle zur Verfügung stellt. Sie schließen ein Ende des MPI-Kabels an die RS-485-Schnittstelle der Baugruppe und das andere Ende an eine Programmierschnittstelle in Ihrem Netz an (siehe Bild 9-8). Ausführliche Informationen zu Kommunikationsprozessoren entnehmen Sie dem Katalog ST 70 SIMATIC *Komponenten für die Automation*.

Konfigurieren eines Multi-Master-Netzes für PC mit MPI- oder CP-Baugruppe

Sie haben viele Konfigurationsmöglichkeiten, wenn Sie eine MPI-Baugruppe oder einen Kommunikationsprozessor einsetzen. Sie können eine Station, auf der die Programmiersoftware STEP 7-Micro/WIN installiert ist (PC mit MPI-Baugruppe oder CP-Karte bzw. SIMATIC Programmiergerät), an ein Netz mit mehreren Mastern anschließen. (Dies gilt auch für das PC/PPI-Kabel, wenn Sie den Multi-Master-Modus aktiviert haben.) Die Master umfassen Operator Panels und Textdisplays (TD 200). Bild 9-8 zeigt eine Konfiguration, in der zwei Textdisplays TD 200 im Netz vorhanden sind.

Bei dieser Konfiguration haben Sie die folgenden Kommunikationsmöglichkeiten:

- STEP 7-Micro/WIN (Teilnehmer 0) kann den Status von Teilnehmer 2 beobachten, während die Textdisplays TD 200 (Teilnehmer 5 und 1) mit den CPUs 214 (Teilnehmer 3 und 4) kommunizieren.
- Beide CPUs 214 können zum Senden von Meldungen mit Hilfe der Netzoperationen (NETR und NETW) freigegeben werden.
- Teilnehmer 3 kann Daten aus der CPU 212 (Teilnehmer 2) und aus Teilnehmer 4 (CPU 214) lesen und auch Daten in diese beiden CPUs schreiben.
- Teilnehmer 4 kann Daten aus der CPU 212 (Teilnehmer 2) und aus Teilnehmer 3 (CPU 214) lesen und auch Daten in diese beiden CPUs schreiben.

Sie können mehrere Master- und Slave-Geräte zu einem Netz verbinden. Die Leistungsfähigkeit des Netzes kann jedoch durch jeden weiteren Teilnehmer im Netz beeinträchtigt werden.

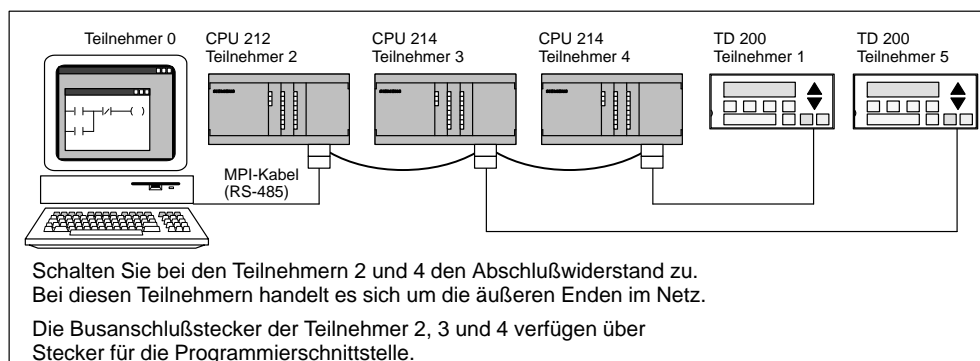


Bild 9-8 Kommunizieren mit S7-200 CPUs über MPI-Baugruppe oder CP-Karte

9.5 DP-Standardkommunikation (dezentrale Peripherie)

Der PROFIBUS-DP-Standard

PROFIBUS-DP (bzw. DP-Standard) ist ein Kommunikationsprotokoll für die dezentrale Peripherie gemäß der EG-Richtlinie EN 50170. Geräte, die dieser Norm entsprechen, sind kompatibel, auch wenn sie von verschiedenen Herstellern produziert wurden. "DP" bedeutet dezentrale Peripherie, d.h. entfernte Ein- und Ausgänge. "PROFIBUS" bedeutet "Process Field Bus."

In der CPU 215 ist das DP-Standardprotokoll implementiert, wie es für Slave-Geräte in den folgenden Normen zu Kommunikationsprotokollen definiert ist:

- EN 50 170 (PROFIBUS) - beschreibt den Buszugriff und das Übertragungsprotokoll und führt die Eigenschaften des Datenübertragungsmediums auf.
- EN 50 170 (DP-Standard) - beschreibt den schnellen, zyklischen Datenaustausch zwischen DP-Mastern und DP-Slaves. Diese Norm definiert außerdem die Vorgehensweisen zum Konfigurieren und Parametrieren, erläutert die Funktionsweise des zyklischen Datenaustauschs mit der dezentralen Peripherie und führt die unterstützten Diagnosemöglichkeiten auf.

Ein DP-Master wird so konfiguriert, daß ihm die Adressen, die Arten der Slaves und die für die Slaves erforderlichen Parametrierungen bekannt sind. Dem Master wird auch mitgeteilt, wo die aus den Slaves gelesenen Daten (Eingaben) und wo die in die Slaves zu schreibenden Daten (Ausgaben) abgelegt werden sollen. Der DP-Master baut das Netz auf und initialisiert die DP-Slaves. Der Master schreibt die Parametrierung und die Konfiguration der Ein- und Ausgänge in den Slave. Dann liest der Master die Diagnoseinformationen aus dem Slave, um sicherzustellen, daß der DP-Slave die Parametrierung und die Konfiguration der Ein- und Ausgänge angenommen hat. Anschließend beginnt der Master, mit dem Slave Daten auszutauschen. Bei jeder Transaktion mit dem Slave werden Ausgänge geschrieben und Eingänge gelesen. Der Datenaustausch läuft unendlich weiter. Ein Slave kann dem Master mitteilen, daß eine Ausnahmebedingung vorliegt, woraufhin der Master die Diagnoseinformationen aus dem Slave ausliest.

Hat ein DP-Master die Parametrierung und die Konfiguration der Ein- und Ausgänge in den DP-Slave geschrieben und der Slave hat beides angenommen, "gehört" dem Master dieser Slave. Ein Slave nimmt nur Anforderungen von dem Master an, zu dem er gehört. Andere Master im Netz können die Ein- und Ausgänge des Slave lesen, doch sie können nicht in den Slave schreiben.

Einsetzen der CPU 215 als DP-Slave

Die CPU 215 kann als DP-Slave an ein PROFIBUS-DP-Netz angeschlossen werden. Schnittstelle 1 der CPU 215 (mit "DP" auf dem Modul gekennzeichnet) ist die DP-Schnittstelle. Diese Schnittstelle arbeitet mit einer beliebigen Baudrate zwischen 9600 Baud und 12 MBaud. Als DP-Slave akzeptiert die CPU 215 verschiedene Konfigurationen der Ein- und Ausgänge vom Master, um unterschiedliche Daten vom und zum Master zu übertragen. Mit Hilfe dieser Funktion können Sie die übertragene Datenmenge an die Anforderungen Ihrer Anwendung anpassen. Im Gegensatz zu vielen DP-Geräte überträgt die CPU 215 nicht nur Ein- und Ausgabedaten. Die CPU 215 überträgt die Daten mit Hilfe eines Bereichs im Variablenspeicher vom und zum Master. Dadurch können beliebige Arten von Daten mit dem Master ausgetauscht werden. Eingänge, Zählerwerte, Zeitwerte oder andere berechnete Werte können zum Master übertragen werden, wenn sie zunächst im Variablenspeicher der CPU 215 gespeichert werden. Auch die vom Master empfangenen Daten werden im Variablenspeicher der CPU 215 abgelegt und können von dort in andere Speicherbereiche übertragen werden.

Die DP-Schnittstelle der CPU 215 kann an einen DP-Master im Netz angeschlossen sein und trotzdem als MPI-Slave mit anderen Mastern, z.B. einem SIMATIC Programmiergerät oder einer S7-300/S7-400 CPU, im gleichen Netz kommunizieren.

Bild 9-9 zeigt ein PROFIBUS-Netz mit einer CPU 215. In dieser Situation ist die CPU 315-2 der DP-Master und wurde von einem SIMATIC Programmiergerät mit der Programmiersoftware STEP 7 konfiguriert. Die CPU 215 ist ein DP-Slave, der der CPU 315-2 "gehört". Die Baugruppe ET 200 ist auch ein Slave der CPU 315-2. Die S7-400 CPU ist an das PROFIBUS-Netz angeschlossen und liest mit der Operation XGET aus dem Anwenderprogramm der S7-400 CPU Daten aus der CPU 215.

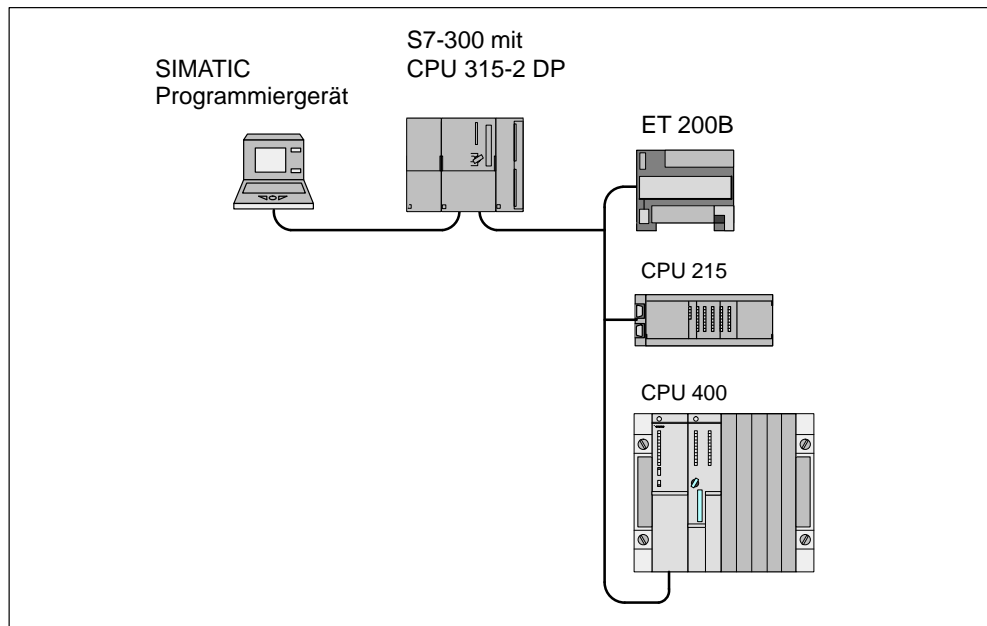


Bild 9-9 CPU 215 im PROFIBUS-Netz

Konfiguration

Die einzige Einstellung, die Sie für die CPU 215 vornehmen müssen, damit Sie sie als PRO-FIBUS-Slave einsetzen können, ist die Teilnehmeradresse der DP-Schnittstelle der CPU. Diese Adresse muß der Adresse in der Konfiguration des Masters entsprechen. Sie können die CPU-Konfiguration für die Adresse der DP-Schnittstelle in STEP 7-Micro/WIN ändern. Anschließend müssen Sie die neue Konfiguration in die CPU 215 laden.

Die Adresse der DP-Schnittstelle der CPU 215 können Sie auch einstellen, indem Sie ein DP-Konfigurationsgerät an die DP-Schnittstelle anschließen. Sie können die Adresse für die DP-Schnittstelle nur dann mit einem dieser Geräte einrichten, wenn die Adresse der DP-Schnittstelle in der CPU-Konfiguration in STEP 7-Micro/WIN auf die voreingestellte Adresse von 126 gesetzt ist. Die in STEP 7-Micro/WIN eingestellte Adresse für die DP-Schnittstelle übersteuert eine Adresse, die mit Hilfe eines DP-Konfigurationsgeräts eingerichtet wurde.

Hinweis

Zum Wiederherstellen der voreingestellten Adresse für die DP-Schnittstelle, nachdem sie einmal mit dem DP-Konfigurationsgerät geändert wurde, gehen Sie folgendermaßen vor:

1. Stellen Sie für die Adresse der DP-Schnittstelle in der CPU-Konfiguration in STEP 7-Micro/WIN einen noch nicht vergebenen Wert ein (nicht 126).
 2. Laden Sie die CPU-Konfiguration in die CPU 215.
 3. Stellen Sie dann in der CPU-Konfiguration in STEP 7-Micro/WIN für die Adresse der DP-Schnittstelle wieder die voreingestellte Adresse von 126 ein.
 4. Laden Sie die CPU-Konfiguration in die CPU 215.
-

Der Master tauscht mit jedem seiner Slaves Daten aus, indem die Informationen aus dem Ausgabebereich in den Ausgabepuffer des Slaves übertragen werden (auch "Empfangs-Briefkasten") genannt. Der Slave reagiert auf die Meldung vom Master, indem er einen Eingabepuffer (auch "Sende-Briefkasten" genannt) an den Master zurückschickt, der diesen im Eingabebereich speichert (siehe Bild 9-10).

Die CPU 215 kann vom DP-Master so konfiguriert werden, daß sie vom Master einen Ausgabepuffer mit Daten annimmt und einen Eingabepuffer mit Daten zurücksendet. Die Aus- und Eingabedatenpuffer befinden sich beide im Variablenspeicher der CPU 215. Beim Konfigurieren des DP-Masters definieren Sie im Rahmen der Parametrierung der CPU 215 eine Byteadresse im Variablenspeicher, an der der Ausgabedatenpuffer beginnen soll. Sie definieren auch die Konfiguration der Ein- und Ausgänge als Ausgabedatenmenge, die in die CPU 215 geschrieben werden soll und als Eingabedatenmenge, die von der CPU 215 empfangen werden soll. Die CPU 215 legt anhand der Konfiguration der Ein- und Ausgänge die Größe des Eingabe- und Ausgabepuffers fest. Der DP-Master schreibt die Parametrierung und die Konfiguration der Ein- und Ausgänge in die CPU 215.

Bild 9-10 zeigt ein Modell des Variablenspeichers in einer CPU 215 sowie die Adreßbereiche der Ein- und Ausgänge einer CPU als DP-Master. In diesem Beispiel hat der DP-Master eine E/A-Konfiguration von 16 Ausgangsbytes und 16 Eingangsbytes und einen Versatz im Variablenspeicher von 5000 definiert. Der Ausgabe- und der Eingabepuffer in der CPU 215 sind, wie in der E/A-Konfiguration festgelegt, beide 16 Bytes lang. Der Ausgabedatenpuffer beginnt bei V5000. Der Eingabedatenpuffer schließt sich direkt an den Ausgabedatenpuffer an und beginnt bei V5016. Die Ausgabedaten (vom Master) werden im Variablenspeicher an der Adresse V5000 abgelegt. Die Eingabedaten (zum Master) werden im Variablenspeicher an der Adresse V5016 abgelegt.

Hinweis

Wenn Sie mit einer Dateneinheit (konsistente Daten) von drei Bytes oder mit Dateneinheiten (konsistente Daten) von mehr als vier Bytes arbeiten, müssen Sie mit SFC 14 die Eingänge des DP-Slave lesen und mit SFC15 die Ausgänge des DP-Slave adressieren. Weitere Informationen hierzu entnehmen Sie dem Referenzhandbuch *System- und Standardfunktionen für S7-300/400*.

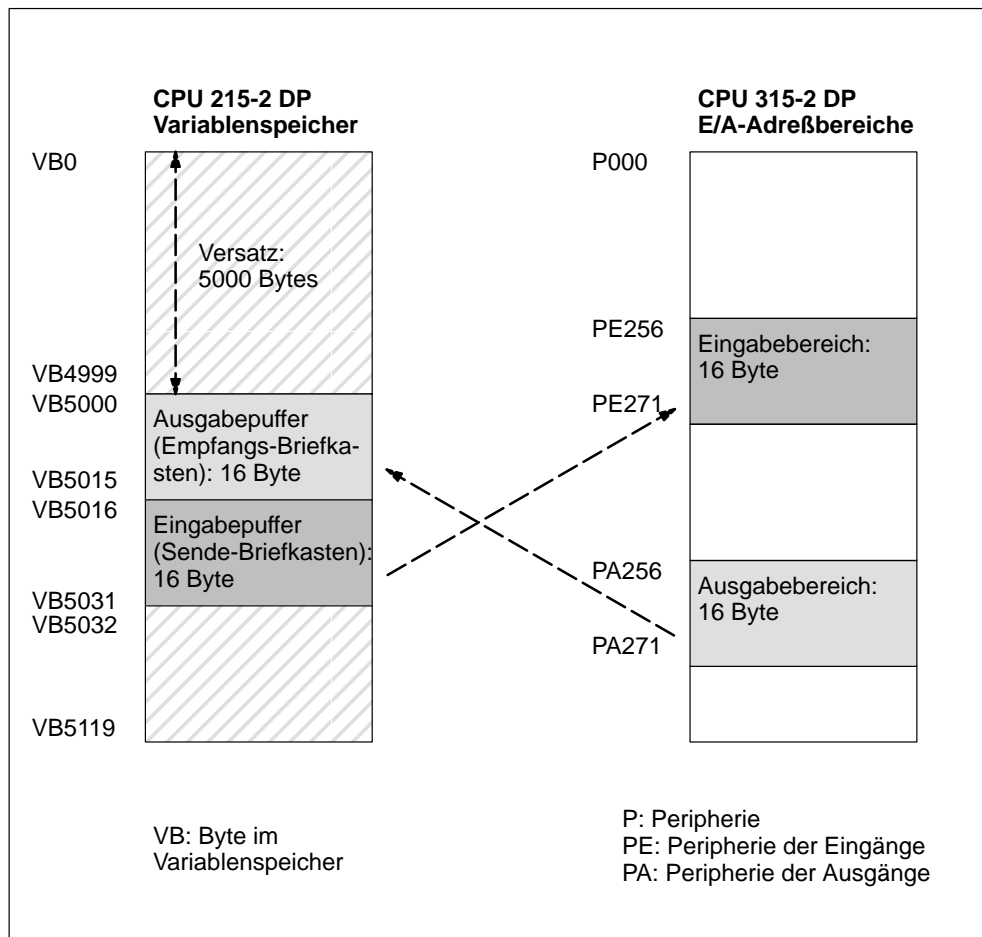


Bild 9-10 Beispiel: Variablenspeicher der CPU 215 und E/A-Adreßbereiche eines PROFIBUS-DP-Masters

Tabelle 9-10 führt die Konfigurationen auf, die von der CPU 215 unterstützt werden.

Tabelle 9-10 Konfigurationen der Ein- und Ausgänge, die von der CPU 215 unterstützt werden

Konfiguration	Größe des Eingabepuffers (Daten für den Master)	Größe des Ausgabepuffers (Daten vom Master)	Datenkonsistenz
1	1 Wort	1 Wort	Wortkonsistenz
2 (Voreinstellung)	2 Wörter	2 Wörter	
3	4 Wörter	4 Wörter	
4	8 Wörter	8 Wörter	
5	16 Wörter	16 Wörter	
6	32 Wörter	32 Wörter	
7	8 Wörter	2 Wörter	
8	16 Wörter	4 Wörter	
9	32 Wörter	8 Wörter	
10	2 Wörter	8 Wörter	
11	4 Wörter	16 Wörter	
12	8 Wörter	32 Wörter	
13	2 Bytes	2 Bytes	Bytekonsistenz
14	8 Bytes	8 Bytes	
15	32 Bytes	32 Bytes	
16	64 Bytes	64 Bytes	
17	4 Bytes	4 Bytes	Pufferkonsistenz
18	8 Bytes	8 Bytes	
19	12 Bytes	12 Bytes	
20	16 Bytes	16 Bytes	

Die Adressen der Eingabe- und Ausgabepuffer können so eingestellt werden, daß sich die Puffer an einer beliebigen Stelle im Variablenspeicher der CPU 215 befinden. Voreingestellt für die Eingabe- und Ausgabepuffer ist die Adresse VB0. Die Adressen für die Eingabe- und Ausgabepuffer sind Teil der Parametrierung, die der Master in die CPU 215 schreibt. Der Master muß so konfiguriert werden, daß er seine Slaves erkennt und die erforderlichen Parameter und die E/A-Konfiguration in jeden seiner Slaves schreibt.

Mit den folgenden Werkzeugen können Sie den DP-Master konfigurieren:

- Für SIMATIC S5-Master: Windows-Software COM ET 200 (COM PROFIBUS)
- Für SIMATIC S7-Master: Programmiersoftware STEP 7
- Für SIMATIC 505-Master: COM ET 200 (COM PROFIBUS) und TISOFT2

Ausführliche Informationen zur Verwendung dieser Konfigurations-/Programmierpakete entnehmen Sie den Handbüchern zu den jeweiligen Werkzeugen. Das Handbuch ET 200 Dezentrale Peripheriesysteme bietet ausführliche Informationen zu DP-Netzen und den Komponenten. Die Bestellnummer des Handbuchs entnehmen Sie Anhang G.

Datenkonsistenz

PROFIBUS unterstützt drei Arten der Datenkonsistenz:

- Durch die Bytekonsistenz wird sichergestellt, daß die Bytes als ganze Einheiten übertragen werden.
- Durch die Wortkonsistenz wird sichergestellt, daß die Übertragung von Worten nicht durch andere Vorgänge in der CPU unterbrochen werden kann. Das bedeutet, daß die beiden Bytes, aus denen sich ein Wort zusammensetzt, immer zusammen übertragen werden und nicht getrennt werden können.
- Durch die Pufferkonsistenz wird sichergestellt, daß der gesamte Datenpuffer als eine Einheit übertragen wird, und nicht durch andere Vorgänge in der CPU unterbrochen werden kann.

Durch Wort- und Pufferkonsistenz wird die CPU gezwungen, andere Vorgänge, z.B. Anwender-Interrupts, anzuhalten, während die Ein- und Ausgabedaten in der CPU geändert bzw. übertragen werden. Sie sollten mit Wortkonsistenz arbeiten, wenn es sich bei den zu übertragenden Datenwerten um ganze Zahlen handelt. Pufferkonsistenz sollten Sie einsetzen, wenn es sich bei den zu übertragenden Datenwerten um Doppelwörter oder Gleitpunktwerte handelt. Mit Pufferkonsistenz sollten Sie auch dann arbeiten, wenn sich eine Gruppe von Werten auf eine Berechnung bzw. ein Objekt bezieht.

Die Datenkonsistenz stellen Sie als Teil der E/A-Konfiguration des Masters ein. Die ausgewählte Datenkonsistenz wird während der Initialisierung des DP-Slave in den Slave geschrieben. Sowohl dem DP-Master als auch dem DP-Slave dient die eingestellte Datenkonsistenz dazu, sicherzustellen, daß die Datenwerte (Bytes, Wörter oder Puffer) ohne Unterbrechung im Master und im Slave übertragen werden.

Bild 9-11 zeigt die verschiedenen Arten der Konsistenz.

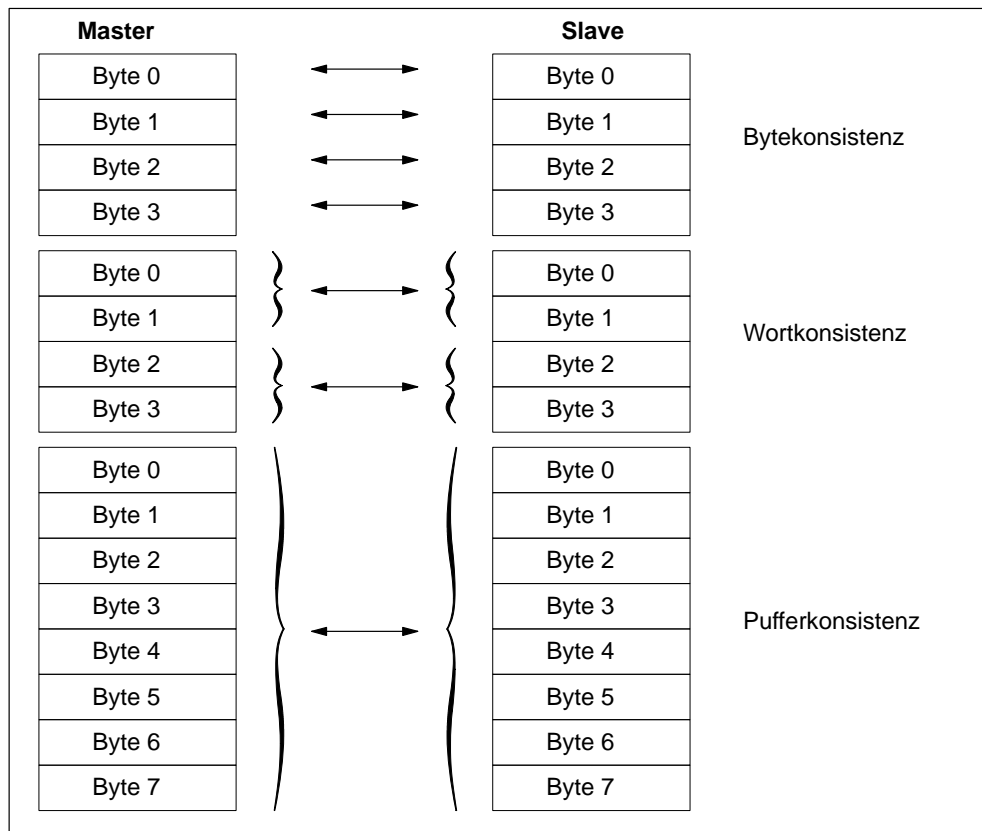


Bild 9-11 Byte-, Wort- und Pufferkonsistenz

Richtlinien für das Anwenderprogramm

Nachdem die CPU 215 erfolgreich von einem DP-Master konfiguriert wurde, gehen die CPU 215 und der DP-Master in den Modus zum Datenaustausch. Im Modus zum Datenaustausch schreibt der Master Ausgabedaten in die CPU 215 und die CPU 215 reagiert mit Eingabedaten. Die Ausgabedaten vom Master werden im Variablenspeicher (im Ausgabepuffer) mit Beginn an der Adresse abgelegt, die vom DP-Master während der Initialisierung definiert wurde. Die Eingabedaten für den Master werden aus der Adresse im Variablenspeicher geholt (dem Eingabepuffer), die direkt auf die Ausgabedaten folgt.

Die Anfangsadressen der Datenpuffer im Variablenspeicher und deren Länge müssen bekannt sein, wenn ein Anwenderprogramm für die CPU 215 erstellt werden soll. Das Anwenderprogramm der CPU 215 muß die vom Master gesendeten Ausgabedaten vom Ausgabepuffer in die jeweiligen Datenbereiche übertragen, in denen die Daten für den Zugriff bereitgestellt werden. Auch müssen die Eingabedaten aus ihren jeweiligen Datenbereichen in den Eingabepuffer übertragen werden, damit sie von dort an den Master gesendet werden können.

Die Ausgabedaten vom DP-Master werden im Variablenspeicher abgelegt, nachdem das Anwenderprogramm im Zyklus bearbeitet wurde. Gleichzeitig werden die Eingabedaten (für den Master) vom Variablenspeicher in einen internen Bereich kopiert, von dem die Daten dann an den Master übertragen werden können. Die Ausgabedaten vom Master werden erst in den Variablenspeicher geschrieben, wenn neue Daten vom Master vorhanden sind. Die Eingabedaten für den Master werden beim nächsten Datenaustausch mit dem Master an diesen übertragen.

SMB110 bis SMB115 bieten Statusinformationen zum DP-Slave der CPU 215. Diese Sondermerker verfügen über voreingestellte Werte, wenn die DP-Kommunikation mit dem Master nicht hergestellt ist. Nachdem ein Master die Parametrierung und die E/A-Konfiguration in die CPU 215 geschrieben hat, speichern diese Sondermerker die vom DP-Master eingestellte Konfiguration. Sie sollten SMB110 prüfen und sicherstellen, daß sich die CPU 215 im Modus zum Datenaustausch mit dem Master befindet, bevor Sie die Informationen aus SMB111 bis SMB115 verwenden (siehe Tabelle 9-11).

Hinweis

Sie können die Puffergrößen und die Pufferadressen für die CPU 215 nicht dadurch ändern, daß Sie die Sondermerker SMB112 bis SMB115 ändern. Nur der DP-Master kann die CPU 215 für den DP-Betrieb einrichten.

Tabelle 9-11 DP-Statusinformationen

Sonderm.	Beschreibung
SMB110	<div style="display: flex; align-items: center;"> <div style="margin-right: 10px;"> <small>MSB</small> 7 0 0 0 0 0 0 s s <small>LSB</small> 0 </div> <div> Schnittstelle 1: Statusbyte des DP-Standardprotokolls ss: Statusbyte DP-Standardprotokoll 00 = DP-Kommunikation nach Anlauf nicht initiiert 01 = Fehler in Konfiguration oder Parametrierung 10 = Modus zum Datenaustausch aktiviert 11 = Modus zum Datenaustausch nicht aktiviert SM111 bis SM115 werden jedesmal aktualisiert, wenn die CPU Konfig.- bzw. Parametr.daten aufnimmt. Diese Adr. werden aktualisiert, auch wenn ein Konfigurations- bzw. Parametrierungsfehler erkannt wird. Die Adr. werden bei jedem Einschalten der CPU gelöscht. </div> </div>
SMB111	Dieses Byte definiert die Adresse vom Master des Slave (0 bis 126).
SMB112 SMB113	Dieses Byte definiert die Adr. im Variablenspeicher des Ausgabepuffers (Versatz von VB0). SM112 ist das höchstwertige Byte (MSB) und SMB113 ist das niederwertigste Byte (LSB).
SMB114	Dieses Byte definiert die Anzahl der Ausgabebytes.
SMB115	Dieses Byte definiert die Anzahl der Eingabebytes.

LED-Statusanzeigen für die DP-Kommunikation

Die CPU 215 verfügt über eine LED-Anzeige, die den Betriebszustand der DP-Schnittstelle anzeigt:

- Nach dem Einschalten der CPU bleibt die DP-Anzeige solange ausgeschaltet, bis die DP-Kommunikation aufgerufen wird.
- Wird die DP-Kommunikation erfolgreich initiiert (die CPU 215 geht in den Modus zum Datenaustausch mit dem Master), dann leuchtet die DP-Anzeige grün auf und bleibt solange eingeschaltet, bis der Modus zum Datenaustausch wieder verlassen wird.
- Geht die Kommunikation verloren, so daß die CPU 215 den Modus zum Datenaustausch wieder verlassen muß, leuchtet die DP-Anzeige rot auf. Dieser Zustand bleibt solange erhalten, bis die CPU 215 ausgeschaltet wird oder der Modus zum Datenaustausch wiederaufgenommen wird.
- Liegt in der E/A-Konfiguration oder in der Parametrierung, die der DP-Master in die CPU 215 schreibt, ein Fehler vor, dann blinkt die DP-Anzeige rot.

Tabelle 9-12 führt die verschiedenen Zustände der LED-Anzeige für die DP-Kommunikation auf:

Tabelle 9-12 Statusanzeige der DP-LED

Zustand der LED	Beschreibung des Zustands
Aus	Seit letztem Anlauf noch keine DP-Standardkommunikation aufgerufen.
Rot blinkend	Fehler in Parametrierung oder Konfiguration, CPU nicht im Modus zum Datenaustausch.
Grün	Derzeit im Modus zum Datenaustausch.
Rot	Modus zum Datenaustausch verloren.

Gerätstammdaten: GSD-Datei

Die verschiedenen PROFIBUS-Geräte weisen unterschiedliche Eigenschaften zur Leistungsfähigkeit auf. Diese Eigenschaften unterscheiden sich bezüglich der Funktionalität (z.B. die Anzahl der Ein- und Ausgangssignale oder die Diagnosemeldungen) oder bezüglich der Busparameter wie Übertragungsgeschwindigkeit und Zeitüberwachung. Diese Parameter unterscheiden sich für jede Art von Gerät und sind auch herstellerspezifisch. Üblicherweise erfahren Sie Informationen hierzu in der Bedienungsanleitung des jeweiligen Geräts. Damit Sie eine einfache PROFIBUS-Konfiguration erstellen können, sind die Leistungsdaten eines bestimmten Geräts in einem elektronischen Datenblatt, genannt Gerätstammdaten (GSD-Datei), angegeben. Konfigurationswerkzeuge, denen die Gerätstammdaten zugrunde liegen, ermöglichen die einfache Integration von Geräten unterschiedlicher Hersteller in einem einzigen Netz.

Die GSD-Datei bietet eine umfassende Beschreibung der Eigenschaften eines Geräts in genau definiertem Format. Die GSD-Dateien werden vom Hersteller für jede Art von Gerät angelegt und dem PROFIBUS-Anwender zur Verfügung gestellt. Das Konfigurationssystem liest die Eigenschaften eines PROFIBUS-Geräts in der GSD-Datei und setzt diese Informationen zum Konfigurieren des Netzes ein.

Die aktuellen Versionen der Software COM ET 200 (jetzt COM PROFIBUS) bzw. der Software STEP 7 enthalten die Konfigurationsdateien für die CPU 215. Verfügt Ihre Version der Software nicht über eine GSD-Datei für die CPU 215, können Sie die GSD für die CPU 215 per Modemabfrage über den PROFIBUS Bulletin Board Service (BBS) laden. Wenn Sie auf den Bulletin Board Service zugreifen, müssen Sie die Eingabeaufforderungen des BBS beantworten. Sie erhalten dann Zugriff auf die Datenbank für die CPU 215 und können die gewünschte Datei kopieren. Es handelt sich hierbei um eine selbst-entpackende Datei, die die Dateien enthält, die für die PROFIBUS-Kommunikation erforderlich sind. Unter folgenden Telefonnummern erreichen Sie unseren BBS:

- In Nord- und Südamerika: (423) 461 - 2751
Zu kopierende Datei: S7215.EXE
- In Europa: (49) (911) 73 79 72
Zu kopierende Datei: W32150AX.200

Sie können sich auch im Internet unter folgender Adresse die aktuelle GSD-Datei laden:
www.profibus.com

Arbeiten Sie mit einem Master nicht von Siemens, entnehmen Sie der Dokumentation des Herstellers, wie Sie den Master mit Hilfe der GSD-Datei konfigurieren.

Beispiel für eine Gerätestammdaten-Datei für die CPU 215

Tabelle 9-13 zeigt ein Beispiel für eine aktuelle GSD-Datei für die CPU 215.

Tabelle 9-13 Beispiel für eine GSD-Datei für Master nicht von SIMATIC

```

;=====
; GSD-Data for the S7-215 DP slave with SPC3
; MLFB : 6ES7 215-2.D00-0XB0
; Date : 05-Oct-1996/release 14-March-97/09/29/97 (45,45)
; Version: 1.2 GSD
; Model-Name, Freeze_Mode_supp, Sync_mode_supp, 45,45k
; File : SIE_2150
;=====
#Profibus_DP
; Unit-Definition-List:
GSD_Revision=1
Vendor_Name="Siemens"
Model_Name="CPU 215-2 DP"
Revision="REV 1.00"
Ident_Number=0x2150
Protocol_Ident=0
Station_Type=0
Hardware_Release="A1.0"
Software_Release="Z1.0"
9.6_supp=1
19.2_supp=1
45.45_supp=1
93.75_supp=1
187.5_supp=1
500_supp=1
1.5M_supp=1
3M_supp=1
6M_supp=1
12M_supp=1
MaxTcdr_9.6=60
MaxTcdr_19.2=60
MaxTcdr_45.45=250
MaxTcdr_93.75=60
MaxTcdr_187.5=60
MaxTcdr_500=100
MaxTcdr_1.5M=150
MaxTcdr_3M=250
MaxTcdr_6M=450
MaxTcdr_12M=800
Redundancy = 0
Repeater_Ctrl_Sig = 2
24V_Pins = 2
Implementation_Type="SPC3"
Bitmap_Device="S7_2150"
;
; Slave-Specification:
OrderNumber="6ES7 215-2.D00-0XB0"
Periphery="SIMATIC S5"
;
Freeze_Mode_supp=1
Sync_Mode_supp=1
Set_Slave_Add_supp=1
Min_Slave_Intervall=1

```

Tabelle 9-13 Beispiel für eine GSD-Datei für Master nicht von SIMATIC, Fortsetzung

```

Max_Diag_Data_Len=6
Slave_Family=3@Tdf@SIMATIC
;
; UserPrmData-Definition
ExtUserPrmData=1 "I/O Offset in the V-memory"
Unsigned16 0 0-5119
EndExtUserPrmData
; UserPrmData: Length and Preset:
User_Prm_Data_Len=3
User_Prm_Data= 0,0,0
Ext_User_Prm_Data_Ref(1)=1
;
Modular_Station=1
Max_Module=1
Max_Input_Len=64
Max_Output_Len=64
Max_Data_Len=128
;
; Module-Definitions:
;
Module="2 Bytes Out/ 2 Bytes In      -- 0x31
EndModule
Module="8 Bytes Out/ 8 Bytes In      -- 0x37
EndModule
Module="32 Bytes Out/ 32 Bytes In    -- 0xC0,0x1F,0x1F
EndModule
Module="64 Bytes Out/ 64 Bytes In    -- 0xC0,0x3F,0x3F
EndModule

Module="1 Word Out/ 1 Word In        -- 0x70
EndModule
Module="2 Word Out/ 2 Word In        -- 0x71
EndModule
Module="4 Word Out/ 4 Word In        -- 0x73
EndModule
Module="8 Word Out/ 8 Word In        -- 0x77
EndModule
Module="16 Word Out/ 16 Word In      -- 0x7F
EndModule
Module="32 Word Out/ 32 Word In      -- 0xC0,0x5F,0x5F
EndModule

Module="2 Word Out/ 8 Word In        -- 0xC0,0x41,0x47
EndModule
Module="4 Word Out/ 16 Word In       -- 0xC0,0x43,0x4F
EndModule
Module="8 Word Out/ 32 Word In       -- 0xC0,0x47,0x5F
EndModule
Module="8 Word Out/ 2 Word In        -- 0xC0,0x47,0x41
EndModule
Module="16 Word Out/ 4 Word In       -- 0xC0,0x4F,0x43
EndModule
Module="32 Word Out/ 8 Word In       -- 0xC0,0x5F,0x47
EndModule
Module="4 Byte buffer I/O           -- 0xB3
EndModule
Module="8 Byte buffer I/O           -- 0xB7
EndModule
Module="12 Byte buffer I/O          -- 0xBB
EndModule
Module="16 Byte buffer I/O          -- 0xBF
EndModule

```

Programmierbeispiel für die DP-Kommunikation mit einer CPU 215 als Slave

Tabelle 9-14 zeigt Ihnen ein Programmierbeispiel in der Anweisungsliste für eine CPU 215, die mit den Informationen für die DP-Schnittstelle in den Sondermerkern arbeitet. Bild 9-12 zeigt dasselbe Programm im Kontaktplan. In diesem Programm werden die Adressen der DP-Puffer über SMW112 ermittelt und die Größen der Puffer aus SMB115 und SMB115 ausgelesen. Diese Informationen werden in dem Programm dazu verwendet, die Daten aus dem DP-Ausgabepuffer in das Prozeßabbild der Ausgänge der CPU 215 zu kopieren. Gleichermaßen werden die Daten aus dem Prozeßabbild der Eingänge der CPU 215 in den DP-Eingabepuffer kopiert.

Tabelle 9-14 Programmierbeispiel in der Anweisungsliste für die DP-Kommunikation mit einer CPU 215 als Slave

```

C-Listing
//Die Daten für die DP-Konfiguration in den Sondermerkern zeigen, wie der
//Master den DP-Slave konfiguriert hat. Das Programm verwendet folgende
Daten:
//  SMB110    DP-Status
//  SMB111    Adresse des Master
//  SMB112    Versatz im Variablenspeicher für die Ausgabedaten (Wortwert)
//  SMB114    Anzahl der Ausgabebytes
//  SMB115    Anzahl der Eingabebytes
//  VD1000    Pointer auf die Ausgabedaten
//  VD1004    Pointer auf die Eingabedaten

NETWORK
LD      SM0.0                //In jedem Zyklus
MOVD    &VB0, VD1000        //Pointer auf Ausgabedaten erstellen,
MOVW    SMW112, VW1002      //Ausgaberversatz addieren,
MOVD    &VB0, VD1004        //Pointer auf Eingabedaten erstellen,
MOVW    SMW112, VW1006      //Ausgaberversatz addieren,
MOVW    +0, AC0             //Akkumulator löschen,
MOVB    SMB114, AC0         //Anzahl an Ausgabebytes laden.
+I      AC0, VW1006         //Versatz-Pointer

NETWORK
LDB>=   SMB114, 9           //Ist die Anzahl der Ausgabebytes > 8,
MOVB    8, VB1008          //Zählwert Ausgabedaten = 8
NOT     //Andernfalls
MOVB    SMB114, VB1008     //Zählwert Ausgabedaten = Anzahl der
//Ausgabebytes.

NETWORK
LDB>=   SMB115, 9           //Ist die Anzahl der Eingabebytes > 8,
MOVB    8, VB1009          //Zählwert Eingabedaten = 8
NOT     //Andernfalls
MOVB    SMB115, VB1009     //Zählwert Eingabedaten = Anzahl der
//Eingabebytes.

NETWORK
LD      SM0.0                //In jedem Zyklus
BMB     *VD1000, AB0, VB1008 //DP-Ausgabedaten in Ausgänge kopieren,
BMB     EB0, *VD1004, VB1009 //Eingänge in DP-Eingabedaten kopieren.

NETWORK
MEND

```

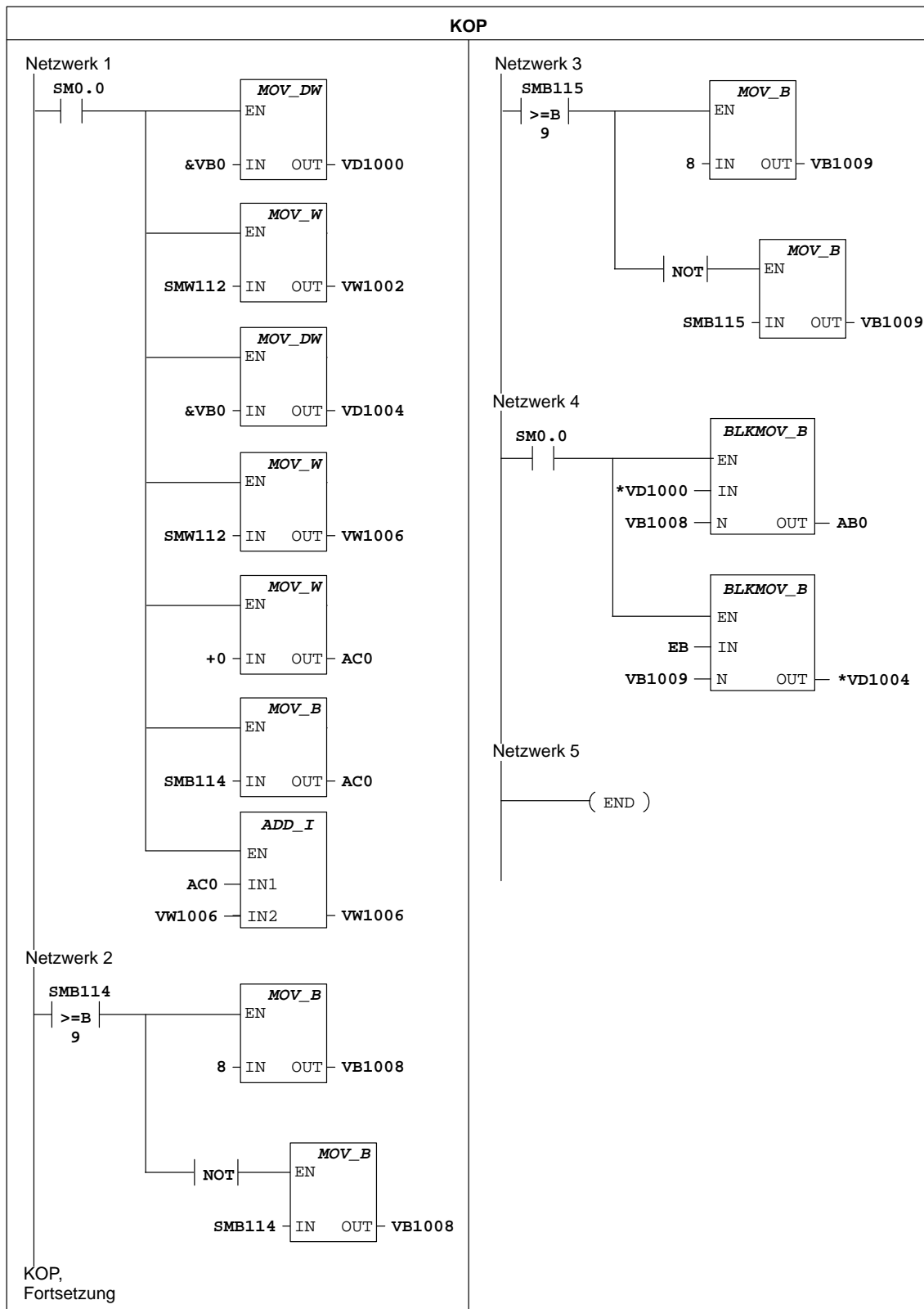


Bild 9-12 Programmierbeispiel im Kontaktplan für die DP-Kommunikation mit einer CPU 215 als Slave

9.6 Leistungsfähigkeit des Netzes

Einschränkungen

Die Leistungsfähigkeit eines Netzes hängt von vielen verschiedenen Variablen ab. Im allgemeinen gibt es zwei Faktoren, die die Leistungsfähigkeit eines Netzes stark beeinflussen: die Baudrate und die Anzahl der an das Netz angeschlossenen Teilnehmer.

Beispiel für ein Netz mit Token-Passing

In einem Netz mit Token-Passing verfügt der Teilnehmer mit dem Token als einziger Teilnehmer über die Sendeberechtigung. Deshalb ist in einem Netz mit Token-Passing, wie auch in einem PPI-Netz, die Token-Umlaufzeit ein wichtiger Faktor. Dies ist die erforderliche Zeit, damit der Token nacheinander an jeden der Master im logischen Ring weitergegeben wird. In Bild 9-13 sehen Sie die Funktionsweise eines Netzes mit mehreren Mastern.

Das Netz in Bild 9-13 besteht aus vier S7-200 CPUs, wobei jede über ein eigenes TD 200 verfügt. Zwei CPUs 214 sammeln Daten von allen anderen CPUs.

Hinweis

Das hier dargestellte Beispiel beruht auf einem Netz wie das in Bild 9-13 gezeigte. Die Konfiguration umfaßt Textdisplays TD 200. Die CPUs 214 arbeiten mit den Operationen NETR und NETW. Die Formeln zum Errechnen der Token-Verweilzeit und der Token-Umlaufzeit, die in Bild 9-14 dargestellt sind, basieren ebenfalls auf einer solchen Konfiguration.

Die Software COM PROFIBUS ermittelt anhand von Analysen die Leistungsfähigkeit des Netzes.

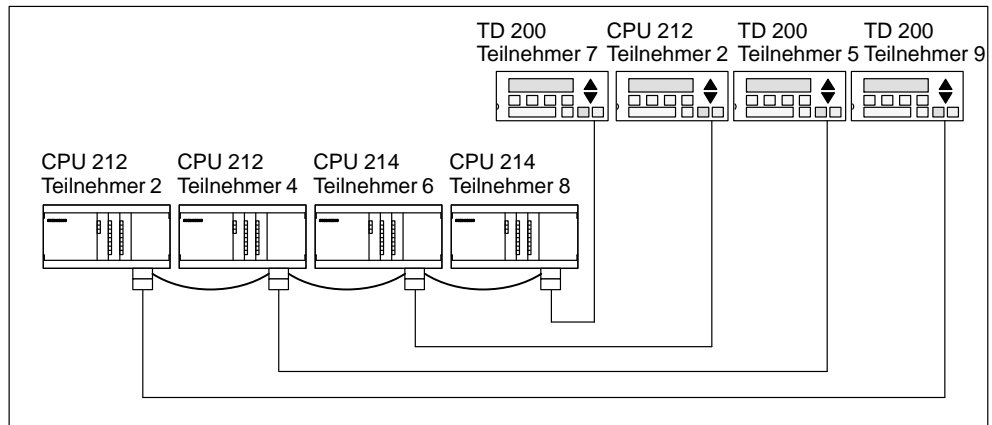


Bild 9-13 Beispiel für ein Netz mit Token-Passing

In dieser Konfiguration kommuniziert das TD 200 (Teilnehmer 3) mit der CPU 212 (Teilnehmer 2), das TD 200 (Teilnehmer 5) kommuniziert mit der CPU 212 (Teilnehmer 4) usw. Die CPU 214 (Teilnehmer 6) sendet Meldungen an die Teilnehmer 2, 4 und 8, die CPU 214 (Teilnehmer 8) sendet Meldungen an die Teilnehmer 2, 4 und 6. Dieses Netz besteht aus sechs Master-Geräten (den vier TD 200 und den beiden CPUs 214) und aus zwei Slave-Geräten (den beiden CPUs 212).

Senden von Meldungen

Damit ein Master eine Meldung senden kann, muß er im Besitz des Token sein. Beispiel: Wenn der Token bei Teilnehmer 3 verweilt, löst der Teilnehmer eine Meldungsanforderung in Teilnehmer 2 aus. Anschließend übergibt Teilnehmer 3 den Token an Teilnehmer 5. Teilnehmer 5 löst eine Meldungsanforderung in Teilnehmer 4 aus und übergibt den Token an Teilnehmer 6. Teilnehmer 6 löst eine Meldungsanforderung in Teilnehmer 2, 4 oder 8 aus und übergibt den Token an Teilnehmer 7. Diese Abläufe, das Auslösen von Meldungen und das Weitergeben des Tokens, setzen sich im gesamten logischen Ring fort, und zwar von Teilnehmer 3 zu Teilnehmer 5, Teilnehmer 6, Teilnehmer 7, Teilnehmer 8, Teilnehmer 9 und schließlich zurück zu Teilnehmer 3. Der Token muß also den logischen Ring vollständig umlaufen, damit ein Master eine Informationsanforderung senden kann. In einem logischen Ring mit sechs Teilnehmern, die während ihrer jeweiligen Token-Verweilzeit eine Anforderung senden, um einen Doppelwortwert (vier Bytes an Daten) zu lesen oder zu schreiben, beträgt die Token-Umlaufzeit ungefähr 900 ms bei einer Geschwindigkeit von 9600 Baud. Bei einer größeren Anzahl Bytes an Daten, auf die pro Meldung zugegriffen werden muß, oder bei einer größeren Anzahl Teilnehmern erhöht sich die Token-Umlaufzeit.

Token-Umlaufzeit

Die Token-Umlaufzeit richtet sich danach, wie lange der Token bei den einzelnen Teilnehmern verweilt. Sie errechnen die Token-Umlaufzeit für ein S7-200 Netz mit mehreren Master, indem Sie die Token-Verweilzeiten der einzelnen Master addieren. Ist der PPI-Master-Modus aktiviert (im PPI-Protokoll Ihres Netzes), können Sie mit der CPU 214, der CPU 215 oder der CPU 216 anderen CPUs mit Hilfe der Operationen Aus Netz lesen (NETR) und In Netz schreiben (NETW) Meldungen senden. Beschreibungen dieser Operationen entnehmen Sie Kapitel 10. Senden Sie mit Hilfe dieser Operationen Meldungen, können Sie mit der in Bild 9-14 gezeigten Formel die ungefähre Token-Verweilzeit errechnen, sofern die folgenden Voraussetzungen erfüllt sind:

- Jeder Teilnehmer sendet eine Anforderung, wenn er über den Token verfügt.
- Die Anforderung umfaßt eine Schreib- oder Leseoperation auf zusammenhängende Daten.
- Es entstehen keine Zugriffskonflikte auf den einzigen Kommunikationspuffer in der CPU.
- Alle Zykluszeiten der CPUs liegen unter 10 ms.

$$\text{Token-Verweilzeit } (T_{\text{vwz}}) = (\text{Zeitbedarf } 128 + n \text{ Daten Zeichen}) * 11 \text{ Bits/Zeichen} * 1/\text{Baudrate}$$

$$\text{Token-Umlaufzeit } (T_{\text{ulz}}) = T_{\text{vwz}} \text{ von Master } 1 + T_{\text{ulz}} \text{ von Master } 2 + \dots + T_{\text{ulz}} \text{ von Master } m$$

n ist die Anzahl der Zeichen an Daten (Bytes)

m ist die Anzahl der Master

Die Token-Umlaufzeit für das oben aufgeführte Beispiel, wobei jeder der sechs Master über die gleiche Token-Verweilzeit verfügt, errechnet sich folgendermaßen :

$$\begin{aligned} T \text{ (Token-Verweilzeit)} &= (128 + 4 \text{ Zeichen}) * 11 \text{ Bits/Zeichen} * 1/9600 \text{ Bitzeit/s} \\ &= 151,25 \text{ ms/Master} \end{aligned}$$

$$\begin{aligned} T \text{ (Token-Umlaufzeit)} &= 151,25 \text{ ms/Master} * 6 \text{ Master} \\ &= 907,5 \text{ ms} \end{aligned}$$

(Eine "Bitzeit" entspricht der Länge einer Signalperiode.)

Bild 9-14 Formeln zum Errechnen der Token-Verweilzeit und der Token-Umlaufzeit bei Einsatz von NETR und NETW

Diagramm zur Token-Umlaufzeit

Die Tabellen 9-15 und 9-16 zeigen Diagramme, in denen die Token-Umlaufzeit in Abhängigkeit von der Anzahl der Teilnehmer und dem Datenvolumen jeweils bei 19.200 Baud und bei 9600 Baud dargestellt wird. Die angegebenen Zeiten beziehen sich auf den Fall, daß Sie die Operationen Aus Netz lesen (NETR) und In Netz schreiben (NETW) in der CPU 214, der CPU 215 oder der CPU 216 einsetzen.

Tabelle 9-15 Token-Umlaufzeit in Abhängigkeit von der Anzahl der Teilnehmer und dem Datenvolumen bei 19.200 Baud

Anzahl der gesendeten Bytes pro Teilnehmer bei 19.200 Baud	Anzahl der Teilnehmer und Zeit in Sekunden								
	2 Teiln.	3 Teiln.	4 Teiln.	5 Teiln.	6 Teiln.	7 Teiln.	8 Teiln.	9 Teiln.	10 Teiln.
1	0,15	0,22	0,30	0,37	0,44	0,52	0,59	0,67	0,74
2	0,15	0,22	0,30	0,37	0,45	0,52	0,60	0,67	0,74
3	0,15	0,23	0,30	0,38	0,45	0,53	0,60	0,68	0,75
4	0,15	0,23	0,30	0,38	0,45	0,53	0,61	0,68	0,76
5	0,15	0,23	0,30	0,38	0,46	0,53	0,61	0,69	0,76
6	0,15	0,23	0,31	0,38	0,46	0,54	0,61	0,69	0,77
7	0,15	0,23	0,31	0,39	0,46	0,54	0,62	0,70	0,77
8	0,16	0,23	0,31	0,39	0,47	0,55	0,62	0,70	0,78
9	0,16	0,24	0,31	0,39	0,47	0,55	0,63	0,71	0,78
10	0,16	0,24	0,32	0,40	0,47	0,55	0,63	0,71	0,79
11	0,16	0,24	0,32	0,40	0,48	0,56	0,64	0,72	0,80
12	0,16	0,24	0,32	0,40	0,48	0,56	0,64	0,72	0,80
13	0,16	0,24	0,32	0,40	0,48	0,57	0,65	0,73	0,81
14	0,16	0,24	0,33	0,41	0,49	0,57	0,65	0,73	0,81
15	0,16	0,25	0,33	0,41	0,49	0,57	0,66	0,74	0,82
16	0,17	0,25	0,33	0,41	0,50	0,58	0,66	0,74	0,83

Tabelle 9-16 Token-Umlaufzeit in Abhängigkeit von der Anzahl der Teilnehmer und dem Datenvolumen bei 9600 Baud

Anzahl der gesendeten Bytes pro Teilnehmer bei 9600 Baud	Anzahl der Teilnehmer und Zeit in Sekunden								
	2 Teiln.	3 Teiln.	4 Teiln.	5 Teiln.	6 Teiln.	7 Teiln.	8 Teiln.	9 Teiln.	10 Teiln.
1	0,30	0,44	0,59	0,74	0,89	1,03	1,18	1,33	1,48
2	0,30	0,45	0,60	0,74	0,89	1,04	1,19	1,34	1,49
3	0,30	0,45	0,60	0,75	0,90	1,05	1,20	1,35	1,50
4	0,30	0,45	0,61	0,76	0,91	1,06	1,21	1,36	1,51
5	0,30	0,46	0,61	0,76	0,91	1,07	1,22	1,37	1,52
6	0,31	0,46	0,61	0,77	0,92	1,07	1,23	1,38	1,54
7	0,31	0,46	0,62	0,77	0,93	1,08	1,24	1,39	1,55
8	0,31	0,47	0,62	0,78	0,94	1,09	1,25	1,40	1,56
9	0,31	0,47	0,63	0,78	0,94	1,10	1,26	1,41	1,57
10	0,32	0,47	0,63	0,79	0,95	1,11	1,27	1,42	1,58
11	0,32	0,48	0,64	0,80	0,96	1,11	1,27	1,43	1,59
12	0,32	0,48	0,64	0,80	0,96	1,12	1,28	1,44	1,60
13	0,32	0,48	0,65	0,81	0,97	1,13	1,29	1,45	1,62
14	0,33	0,49	0,65	0,81	0,98	1,14	1,30	1,46	1,63
15	0,33	0,49	0,66	0,82	0,98	1,15	1,31	1,47	1,64
16	0,33	0,50	0,66	0,83	0,99	1,16	1,32	1,49	1,65

Optimieren der Leistungsfähigkeit im Netz

Die beiden Faktoren, die sich am stärksten auf die Leistungsfähigkeit eines Netzes auswirken, sind die Baudrate und die Anzahl der Master. Wenn Sie das Netz mit der höchsten von allen Geräten unterstützten Baudrate betreiben, hat dies die größten Auswirkungen auf die Leistungsfähigkeit des Netzes. Wenn Sie die Anzahl der Master im Netz so gering wie möglich halten, steigert dies ebenfalls die Leistungsfähigkeit des Netzes. Jeder Master im Netz erhöht die Anforderungen an das Netz. Weniger Master verkürzen die Bearbeitungszeiten.

Folgende Faktoren wirken sich ebenfalls auf die Leistungsfähigkeit eines Netzes aus:

- Auswahl der Adressen für Master und Slaves
- GAP-Aktualisierungsfaktor
- Höchste Teilnehmeradresse

Die Adressen der Master sollten so eingestellt sein, daß sich alle Master an aufeinanderfolgenden Adressen befinden, ohne daß zwischen den Adressen Lücken entstehen. Ist zwischen den Adressen der Master eine Lücke vorhanden, prüfen die Master diese Adresse regelmäßig daraufhin, ob ein neuer Master online gehen will. Diese Prüfung erfordert Zeit und erhöht dadurch die Gesamtbearbeitungszeit im Netz. Befinden sich zwischen den Adressen der Master keine Lücken, wird auch keine Prüfung durchgeführt und die Bearbeitungszeit verringert sich.

Die Adressen der Slaves können auf beliebige Werte eingestellt sein, ohne daß sie die Leistungsfähigkeit des Netzes beeinträchtigen, solange sich die Adressen der Slaves nicht zwischen denen der Master befinden. Befinden sich Adressen von Slaves zwischen denen der Master, erhöht dies die Bearbeitungszeit im Netz genauso, als wenn sich Lücken zwischen den Adressen der Master befinden.

Die S7-200 CPUs können so konfiguriert werden, daß Lücken (GAPs) zwischen den Adressen nur in bestimmten Abständen geprüft werden. Dies erreichen Sie, indem Sie in der CPU-Konfiguration in STEP 7-Micro/WIN einen GAP-Aktualisierungsfaktor für eine CPU-Schnittstelle definieren. Der GAP-Aktualisierungsfaktor teilt der CPU mit, wie häufig die Lücken zwischen den Adressen auf neue Master geprüft werden sollen. Ein GAP-Aktualisierungsfaktor von 1 läßt die CPU die Lücke zwischen den Adressen jedesmal dann prüfen, wenn die CPU im Besitz des Tokens ist. Ein GAP-Aktualisierungsfaktor von 2 läßt die CPU die Lücke zwischen den Adressen jedes zweite Mal, wenn die CPU im Besitz des Tokens ist, auf neue Master prüfen. Je höher Sie den GAP-Aktualisierungsfaktor einstellen, desto weniger Bearbeitungszeit benötigt das Netz, wenn Lücken zwischen den Adressen der Master vorhanden sind. Sind zwischen den Adressen der Master keine Lücken vorhanden, wirkt sich der GAP-Aktualisierungsfaktor nicht auf die Leistungsfähigkeit des Netzes aus. Haben Sie einen hohen GAP-Aktualisierungsfaktor eingestellt, müssen Sie mit langen Verzögerungszeiten rechnen, wenn ein neuer Master online gehen soll, da die Adressen nur selten auf neue Master geprüft werden. Der GAP-Aktualisierungsfaktor wird nur eingesetzt, wenn eine CPU als PPI-Master betrieben wird.

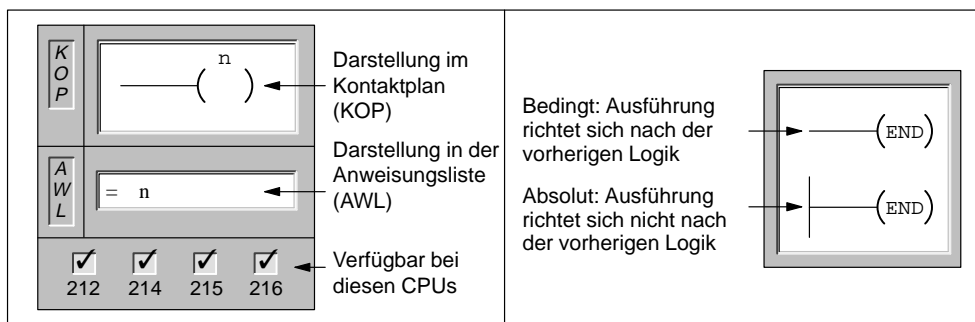
Die höchste Teilnehmeradresse (höchste Stationsadresse = HSA) definiert die höchste Adresse, an der ein Master nach einem anderen Master suchen soll. Wenn Sie eine höchste Teilnehmeradresse einstellen, begrenzen Sie dadurch den Adreßbereich, der von dem letzten Master (höchste Adresse) im Netz geprüft werden muß. Durch das Einschränken des Adreßbereichs wird die Zeit verringert, die benötigt wird, um nach neuen Mastern abzufragen und diese online zu schalten. Die höchste Teilnehmeradresse hat keinen Einfluß auf die Adressen der Slaves. Die Master können weiterhin mit Slaves kommunizieren, deren Adressen über die höchste Teilnehmeradresse hinausgehen. Die höchste Teilnehmeradresse wird nur eingesetzt, wenn eine CPU als PPI-Master betrieben wird. Sie stellen die höchste Teilnehmeradresse in der CPU-Konfiguration in STEP 7-Micro/WIN für eine CPU-Schnittstelle ein.

Im allgemeinen sollten Sie die höchste Teilnehmeradresse für alle Master auf den gleichen Wert setzen. Die höchste Teilnehmeradresse sollte der Adresse des höchsten Masters entsprechen oder größer sein. Bei den S7-200 CPUs ist als höchste Teilnehmeradresse die Adresse 126 voreingestellt.

Operationssatz

10

In diesem Kapitel wird anhand der folgenden Konventionen die Darstellungsart einer Operation im Kontaktplan (KOP) und in der Anweisungsliste (AWL) aufgeführt. Außerdem sind die CPUs gekennzeichnet, bei denen die jeweilige Operation verfügbar ist:



Kapitelübersicht

Abschnitt	Beschreibung	Seite
10.1	Gültige Bereiche für die S7-200 CPUs	10-2
10.2	Operationen mit Kontakten	10-4
10.3	Operationen mit Vergleichskontakten	10-7
10.4	Operationen mit Ausgängen	10-10
10.5	Operationen mit Zeiten, Zählern, schnellen Zählern, schnellen Ausgängen und Impulsausgaben	10-13
10.6	Arithmetische Operationen und Operationen für den PID-Regler	10-50
10.7	Operationen zum Inkrementieren und Dekrementieren	10-66
10.8	Übertragungs- und Tabellenoperationen	10-68
10.9	Schiebe- und Rotieroperationen	10-78
10.10	Operationen für die Programmsteuerung	10-84
10.11	Stackoperationen	10-99
10.12	Verknüpfungsoperationen	10-102
10.13	Umwandlungsoperationen	10-108
10.14	Interrupt- und Kommunikationsoperationen	10-114

10.1 Gültige Bereiche für die S7-200 CPUs

Tabelle 10-1 Speicherbereiche und Funktionen der S7-200 CPUs

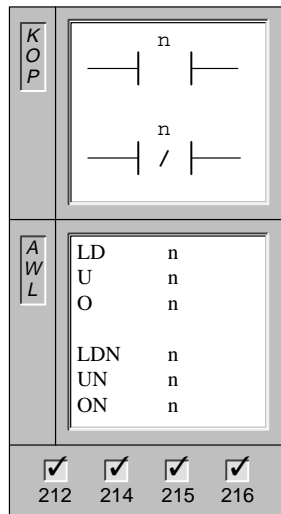
Beschreibung	CPU 212	CPU 214	CPU 215	CPU 216
Größe Anwenderprogramm	512 Wörter	2 K Wörter	4 K Wörter	4 K Wörter
Größe Anwenderdaten	512 Wörter	2 K Wörter	2,5 K Wörter	2,5 K Wörter
Prozeßabbild der Eingänge	E0.0 bis E7.7	E0.0 bis E7.7	E0.0 bis E7.7	E0.0 bis E7.7
Prozeßabbild der Ausgänge	A0.0 bis A7.7	A0.0 bis A7.7	A0.0 bis A7.7	A0.0 bis A7.7
Analogeingänge (nur Lesen)	AEW0 bis AEW30	AEW0 bis AEW30	AEW0 bis AEW30	AEW0 bis AEW30
Analogausgänge (nur Schreiben)	AAW0 bis AAW30	AAW0 bis AAW30	AAW0 bis AAW30	AAW0 bis AAW30
Variablenspeicher (V) Nullspannungsfester Bereich (max.)	V0.0 bis V1023.7 V0.0 bis V199.7	V0.0 bis V4095.7 V0.0 bis V1023.7	V0.0 bis V5119.7 V0.0 bis V5119.7	V0.0 bis V5119.7 V0.0 bis V5119.7
Merker (M) Nullspannungsfester Bereich (max.)	M0.0 bis M15.7 MB0 bis MB13	M0.0 bis M31.7 MB0 bis MB13	M0.0 bis M31.7 MB0 bis MB13	M0.0 bis M31.7 MB0 bis MB13
Sondermerker (SM) Schreibgeschützt	SM0.0 bis SM45.7 SM0.0 bis SM29.7	SM0.0 bis SM85.7 SM0.0 bis SM29.7	SM0.0 bis SM194.7 SM0.0 bis SM29.7	SM0.0 bis SM194.7 SM0.0 bis SM29.7
Zeiten Speichernde Einschaltverzögerung 1 ms	64 (T0 bis T63) T0	128 (T0 bis T127) T0, T64	256 (T0 bis T255) T0, T64	256 (T0 bis T255) T0, T64
Speichernde Einschaltverzögerung 10 ms	T1 bis T4	T1 bis T4, T65 bis T68	T1 bis T4, T65 bis T68	T1 bis T4, T65 bis T68
Speichernde Einschaltverzögerung 100 ms	T5 bis T31	T5 bis T31, T69 bis T95	T5 bis T31, T69 bis T95	T5 bis T31, T69 bis T95
Einschaltverzögerung 1 ms	T32	T32, T96	T32, T96	T32, T96
Einschaltverzögerung 10 ms	T33 bis T36	T33 bis T36, T97 bis T100	T33 bis T36, T97 bis T100	T33 bis T36, T97 bis T100
Einschaltverzögerung 100 ms	T37 bis T63	T37 bis T63, T101 bis T127	T37 bis T63, T101 bis T255	T37 bis T63, T101 bis T255
Zähler	Z0 bis Z63	Z0 bis Z127	Z0 bis Z255	Z0 bis Z255
Schnelle Zähler	HC0	HC0 bis HC2	HC0 bis HC2	HC0 bis HC2
Ablaufsteuerungsrelais	S0.0 bis S7.7	S0.0 bis S15.7	S0.0 bis S31.7	S0.0 bis S31.7
Akkumulatoren	AC0 bis AC3	AC0 bis AC3	AC0 bis AC3	AC0 bis AC3
Sprünge/Sprungmarken	0 bis 63	0 bis 255	0 bis 255	0 bis 255
Aufrufe/Unterprogramme	0 bis 15	0 bis 63	0 bis 63	0 bis 63
Interruptprogramme	0 bis 31	0 bis 127	0 bis 127	0 bis 127
Interruptereignisse	0, 1, 8 bis 10, 12	0 bis 20	0 bis 23	0 bis 26
PID-Regler	Nicht unterstützt	Nicht unterstützt	0 bis 7	0 bis 7
Schnittstellen	0	0	0	0 und 1

Tabelle 10-2 Operandenbereich der S7-200 CPUs

Zugriffsformat	CPU 212	CPU 214	CPU 215	CPU 216
Bit (Byte.Bit)	V 0.0 bis 1023.7	V 0.0 bis 4095.7	V 0.0 bis 5119.7	V 0.0 bis 5119.7
	E 0.0 bis 7.7	E 0.0 bis 7.7	E 0.0 bis 7.7	E 0.0 bis 7.7
	Q 0.0 bis 7.7	Q 0.0 bis 7.7	Q 0.0 bis 7.7	Q 0.0 bis 7.7
	M 0.0 bis 15.7	M 0.0 bis 31.7	M 0.0 bis 31.7	M 0.0 bis 31.7
	SM 0.0 bis 45.7	SM 0.0 bis 85.7	SM 0.0 bis 194.7	SM 0.0 bis 194.7
	T 0 bis 63	T 0 bis 127	T 0 bis 255	T 0 bis 255
	Z 0 bis 63	Z 0 bis 127	Z 0 bis 255	Z 0 bis 255
	S 0.0 bis 7.7	S 0.0 bis 15.7	S 0.0 bis 31.7	S 0.0 bis 31.7
Byte	VB 0 bis 1023	VB 0 bis 4095	VB 0 bis 5119	VB 0 bis 5119
	EB 0 bis 7	EB 0 bis 7	EB 0 bis 7	EB 0 bis 7
	AB 0 bis 7	AB 0 bis 7	AB 0 bis 7	AB 0 bis 7
	MB 0 bis 15	MB 0 bis 31	MB 0 bis 31	MB 0 bis 31
	SMB 0 bis 45	SMB 0 bis 85	SMB 0 bis 194	SMB 0 bis 194
	AC 0 bis 3	AC 0 bis 3	AC 0 bis 3	AC 0 bis 3
	SB 0 bis 7	SB 0 bis 15	SB 0 bis 31	SB 0 bis 31
	Konstante	Konstante	Konstante	Konstante
Wort	VW 0 bis 1022	VW 0 bis 4094	VW 0 bis 5118	VW 0 bis 5118
	T 0 bis 63	T 0 bis 127	T 0 bis 255	T 0 bis 255
	Z 0 bis 63	Z 0 bis 127	Z 0 bis 255	Z 0 bis 255
	EW 0 bis 6	EW 0 bis 6	EW 0 bis 6	EW 0 bis 6
	AW 0 bis 6	AW 0 bis 6	AW 0 bis 6	AW 0 bis 6
	MW 0 bis 14	MW 0 bis 30	MW 0 bis 30	MW 0 bis 30
	SMW 0 bis 44	SMW 0 bis 84	SMW 0 bis 193	SMW 0 bis 193
	AC 0 bis 3	AC 0 bis 3	AC 0 bis 3	AC 0 bis 3
	AEW 0 bis 30	AEW 0 bis 30	AEW 0 bis 30	AEW 0 bis 30
	AAW 0 bis 30	AAW 0 bis 30	AAW 0 bis 30	AAW 0 bis 30
	SW 0 bis 6	SW 0 bis 14	SW 0 bis 30	SW 0 bis 30
	Konstante	Konstante	Konstante	Konstante
	Doppelwort	VD 0 bis 1020	VD 0 bis 4092	VD 0 bis 5116
ED 0 bis 4		ED 0 bis 4	ED 0 bis 4	ED 0 bis 4
AD 0 bis 4		AD 0 bis 4	AD 0 bis 4	AD 0 bis 4
MD 0 bis 12		MD 0 bis 28	MD 0 bis 28	MD 0 bis 28
SMD 0 bis 42		SMD 0 bis 82	SMD 0 bis 191	SMD 0 bis 191
AC 0 bis 3		AC 0 bis 3	AC 0 bis 3	AC 0 bis 3
HC 0		HC 0 bis 2	HC 0 bis 2	HC 0 bis 2
SD 0 bis 4		SD 0 bis 12	SD 0 bis 28	SD 0 bis 28
Konstante		Konstante	Konstante	Konstante

10.2 Operationen mit Kontakten

Standardkontakte



Der **Schließer** ist geschlossen (ein), wenn der Bitwert an Adresse n gleich 1 ist.

In AWL wird der Schließer von den Operationen **Bitwert laden**, **Bitwert durch UND verknüpfen** und **Bitwert durch ODER verknüpfen** dargestellt. Diese Operationen laden den Bitwert der Adresse n als obersten Stackwert bzw. verknüpfen den Bitwert mit dem obersten Stackwert durch UND bzw. ODER.

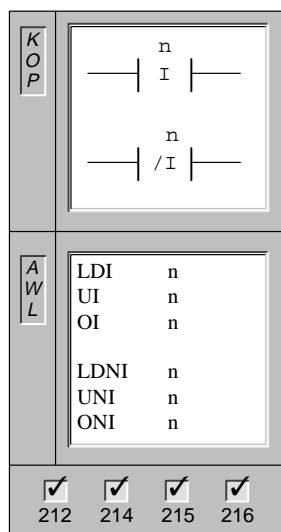
Der **Öffner** ist geschlossen (ein), wenn der Bitwert an Adresse n gleich 0 ist.

In AWL wird der Öffner von den Operationen **Negierten Bitwert laden**, **Negierten Bitwert durch UND verknüpfen** und **Negierten Bitwert durch ODER verknüpfen** dargestellt. Diese Operationen laden den negierten Bitwert der Adresse n als obersten Stackwert bzw. verknüpfen den negierten Bitwert mit dem obersten Stackwert durch UND bzw. ODER.

Operanden: n: E, A, M, SM, T, Z, V, S

Beim Aktualisieren am Anfang des CPU-Zyklus erhalten diese Operationen den angegebenen Wert aus dem Prozeßabbild.

Direkte Kontakte



Der **direkte Schließer** ist geschlossen (ein), wenn der Bitwert des angegebenen Eingangs n gleich 1 ist.

In AWL wird der direkte Schließer von den Operationen **Bitwert direkt laden**, **Bitwert direkt durch UND verknüpfen** und **Bitwert direkt durch ODER verknüpfen** dargestellt. Diese Operationen laden den Bitwert des angegebenen Eingangs n direkt als obersten Stackwert bzw. verknüpfen den Bitwert direkt mit dem obersten Stackwert durch UND bzw. ODER.

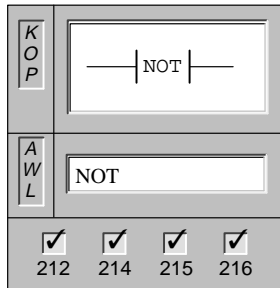
Der **direkte Öffner** ist geschlossen (ein), wenn der Bitwert des angegebenen Eingangs n gleich 0 ist.

In AWL wird der direkte Öffner von den Operationen **Negierten Bitwert direkt laden**, **Negierten Bitwert direkt durch UND verknüpfen** und **Negierten Bitwert direkt durch ODER verknüpfen** dargestellt. Diese Operationen laden den negierten Bitwert der Adresse n direkt als obersten Stackwert bzw. verknüpfen den negierten Bitwert direkt mit dem obersten Stackwert durch UND bzw. ODER.

Operanden: n: I

Die direkte Operation liest den angesprochenen Wert aus dem physikalischen Eingang, wenn die Operation ausgeführt wird. Das Prozeßabbild wird nicht aktualisiert.

NOT

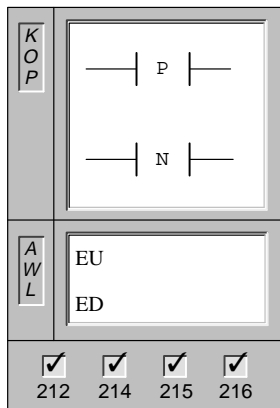


Der Kontakt **NOT** ändert den Zustand des Signalfusses. Erreicht der Signalfuß den Kontakt NOT, wird er am Kontakt gestoppt. Erreicht der Signalfuß den Kontakt NOT nicht, wird am Kontakt Signalfuß erzeugt.

In AWL ändert die Operation Obersten Stackwert invertieren den obersten Stackwert von "0" in "1" oder von "1" in "0".

Operanden: keine

Steigende Flanke und fallende Flanke



Der Kontakt **Steigende Flanke** läßt den Signalfuß bei jeder steigenden Flanke einen Zyklus lang zu.

In AWL wird der Kontakt Steigende Flanke durch die Operation **Erkennung steigende Flanke** dargestellt. Wird im obersten Stackwert eine steigende Flanke (Wechsel von "0" auf "1") erkannt, wird der oberste Stackwert auf "1" gesetzt. Wird keine steigende Flanke erkannt, wird der Wert auf "0" gesetzt.

Der Kontakt **Erkennung fallende Flanke** läßt den Signalfuß bei jeder fallenden Flanke einen Zyklus lang zu.

In AWL wird der Kontakt Fallende Flanke durch die Operation **Erkennung fallende Flanke** dargestellt. Wird im obersten Stackwert eine fallende Flanke (Wechsel von "1" auf "0") erkannt, wird der oberste Stackwert auf "1" gesetzt. Wird keine fallende Flanke erkannt, wird der Wert auf "0" gesetzt.

Operanden: keine

Beispiele für Kontakte

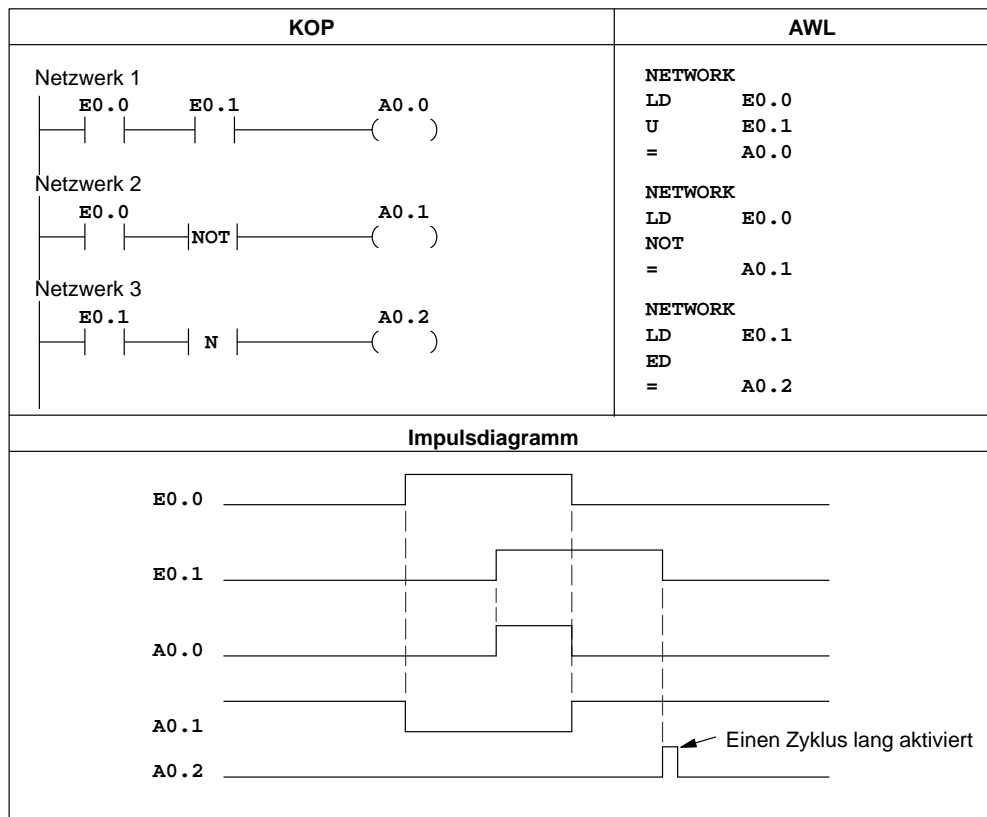
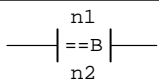
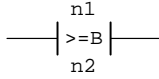
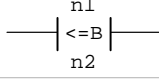


Bild 10-1 Beispiel für Verknüpfungsoperationen mit Kontakten in KOP und AWL

10.3 Operationen mit Vergleichskontakten

Bytevergleich

K O P	$n1$  $n2$
	$n1$  $n2$
	$n1$  $n2$
A W L	LDB= n1, n2
	UB= n1, n2
	OB= n1, n2
	LDB>= n1, n2
	UB>= n1, n2
	OB>= n1, n2
	LDB<= n1, n2
	UB<= n1, n2
	OB<= n1, n2
<input checked="" type="checkbox"/> 212 <input checked="" type="checkbox"/> 214 <input checked="" type="checkbox"/> 215 <input checked="" type="checkbox"/> 216	

Die Operation **Bytevergleich** vergleicht die beiden Werte n1 und n2 miteinander. Sie können folgende Vergleiche anstellen: n1 = n2, n1 >= n2 und n1 <= n2.

Operanden: n1, n2: VB, EB, AB, MB, SMB, AC, *VD, *AC, SB

In KOP ist der Kontakt aktiviert, wenn der Vergleich wahr ist.

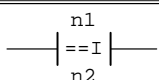
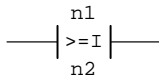
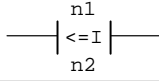
In AWL laden die Operationen den Wert "1" als obersten Stackwert bzw. verknüpfen den Wert "1" mit dem obersten Stackwert durch UND bzw. ODER, wenn der Vergleich wahr ist.

Bytevergleiche sind vorzeichenlos.

Hinweis: Sie können die Vergleiche <>, < und > ausführen, indem Sie die Operation NOT zusammen mit einer der Operationen >=, = oder <= einsetzen. Die beiden folgenden Operationen entsprechen dem Vergleich <> (ungleich) zwischen VB100 und dem Wert 50:

```
LDB= VB100, 50
NOT
```

Wortvergleich

K O P	$n1$  $n2$
	$n1$  $n2$
	$n1$  $n2$
A W L	LDW= n1, n2
	UW= n1, n2
	OW= n1, n2
	LDW>= n1, n2
	UW>= n1, n2
	OW>= n1, n2
	LDW<= n1, n2
	UW<= n1, n2
	OW<= n1, n2
<input checked="" type="checkbox"/> 212 <input checked="" type="checkbox"/> 214 <input checked="" type="checkbox"/> 215 <input checked="" type="checkbox"/> 216	

Die Operation **Wortvergleich** vergleicht die beiden Werte n1 und n2 miteinander. Sie können folgende Vergleiche anstellen: n1 = n2, n1 >= n2 und n1 <= n2.

Operanden: n1, n2: VW, T, Z, EW, AW, MW, SMW, AC, AEW, Konstante, *VD, *AC, SW

In KOP ist der Kontakt aktiviert, wenn der Vergleich wahr ist.

In AWL laden die Operationen den Wert "1" als obersten Stackwert bzw. verknüpfen den Wert "1" mit dem obersten Stackwert durch UND bzw. ODER, wenn der Vergleich wahr ist.

Wortvergleiche haben ein Vorzeichen (16#7FFF > 16#8000).

Hinweis: Sie können die Vergleiche <>, < und > ausführen, indem Sie die Operation NOT zusammen mit einer der Operationen >=, = oder <= einsetzen. Die beiden folgenden Operationen entsprechen dem Vergleich <> (ungleich) zwischen VW100 und dem Wert 50:

```
LDW= VW100, 50
NOT
```

Doppelwortvergleich

K O P	$n1$ — ==D — $n2$
	$n1$ — >=D — $n2$
	$n1$ — <=D — $n2$
A W L	LDD= $n1, n2$
	UD= $n1, n2$
	OD= $n1, n2$
	LDD>= $n1, n2$
	UD>= $n1, n2$
	OD>= $n1, n2$
	LDD<= $n1, n2$
	UD<= $n1, n2$
	OD<= $n1, n2$
<input checked="" type="checkbox"/> 212	
<input checked="" type="checkbox"/> 214	
<input checked="" type="checkbox"/> 215	
<input checked="" type="checkbox"/> 216	

Die Operation **Doppelwortvergleich** vergleicht die beiden Werte $n1$ und $n2$ miteinander. Sie können folgende Vergleiche anstellen: $n1 = n2$, $n1 \geq n2$ und $n1 \leq n2$.

Operanden: $n1, n2$: VD, ED, AD, MD, SMD, AC, HC, Konstante, *VD, *AC, SD

In KOP ist der Kontakt aktiviert, wenn der Vergleich wahr ist.

In AWL laden die Operationen den Wert "1" als obersten Stackwert bzw. verknüpfen den Wert "1" mit dem obersten Stackwert durch UND bzw. ODER, wenn der Vergleich wahr ist.

Doppelwortvergleiche haben ein Vorzeichen (16#7FFFFFFF > 16#80000000).

Hinweis: Sie können die Vergleiche <>, < und > ausführen, indem Sie die Operation NOT zusammen mit einer der Operationen >=, = oder <= einsetzen. Die beiden folgenden Operationen entsprechen dem Vergleich <> (ungleich) zwischen VD100 und dem Wert 50:

LDD= VD100, 50
NOT

Realzahlenvergleich

K O P	$n1$ — ==R — $n2$
	$n1$ — >=R — $n2$
	$n1$ — <=R — $n2$
A W L	LDR= $n1, n2$
	UR= $n1, n2$
	OR= $n1, n2$
	LDR>= $n1, n2$
	UR>= $n1, n2$
	OR>= $n1, n2$
	LDR<= $n1, n2$
	UR<= $n1, n2$
	OR<= $n1, n2$
<input type="checkbox"/> 212	
<input checked="" type="checkbox"/> 214	
<input checked="" type="checkbox"/> 215	
<input checked="" type="checkbox"/> 216	

Die Operation **Realzahlenvergleich** vergleicht die beiden Werte $n1$ und $n2$ miteinander. Sie können folgende Vergleiche anstellen: $n1 = n2$, $n1 \geq n2$ und $n1 \leq n2$.

Operanden: $n1, n2$: VD, ED, AD, MD, SMD, AC, *VD, *AC, SD

In KOP ist der Kontakt aktiviert, wenn der Vergleich wahr ist.

In AWL laden die Operationen den Wert "1" als obersten Stackwert bzw. verknüpfen den Wert "1" mit dem obersten Stackwert durch UND bzw. ODER, wenn der Vergleich wahr ist.

Realzahlenvergleiche haben ein Vorzeichen.

Hinweis: Sie können die Vergleiche <>, < und > ausführen, indem Sie die Operation NOT zusammen mit einer der Operationen >=, = oder <= einsetzen. Die beiden folgenden Operationen entsprechen dem Vergleich <> (ungleich) zwischen VD100 und dem Wert 50:

LDR= VD100, 50
NOT

Beispiele für Vergleichskontakte

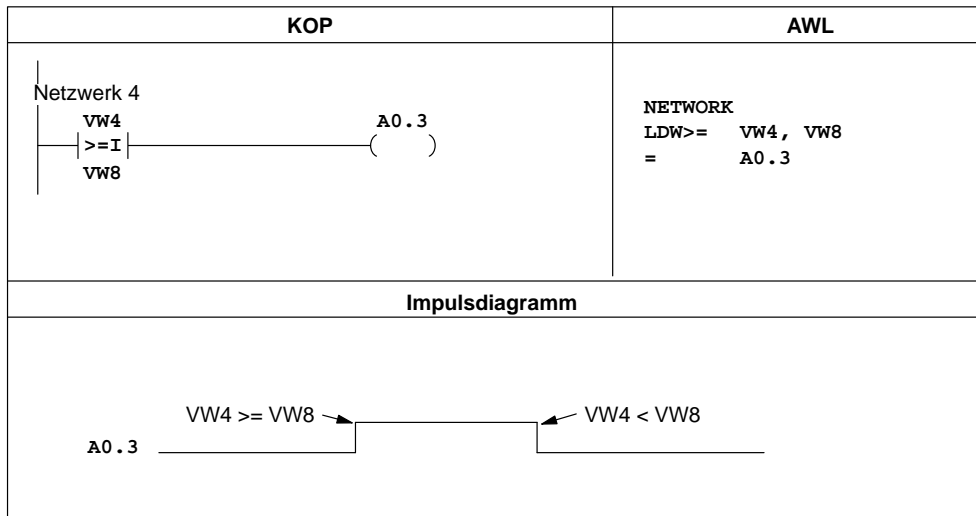


Bild 10-2 Beispiele für Vergleichskontakte in KOP und AWL

10.4 Operationen mit Ausgängen

Zuweisen

K O P	$\text{---} \left(\begin{matrix} n \\ \end{matrix} \right)$
A W L	$= n$
<input checked="" type="checkbox"/> 212 <input checked="" type="checkbox"/> 214 <input checked="" type="checkbox"/> 215 <input checked="" type="checkbox"/> 216	

Wird die Operation **Zuweisen** ausgeführt, dann wird der angegebene Parameter (n) eingeschaltet.

In AWL kopiert die Operation Zuweisen den obersten Stackwert in den angegebenen Parameter (n).

Operanden: n: E, A, M, SM, T, Z, V, S

Bitwert direkt zuweisen

K O P	$\text{---} \left(\begin{matrix} n \\ I \end{matrix} \right)$
A W L	$=I n$
<input checked="" type="checkbox"/> 212 <input checked="" type="checkbox"/> 214 <input checked="" type="checkbox"/> 215 <input checked="" type="checkbox"/> 216	

Wird die Operation **Bitwert direkt zuweisen** ausgeführt, dann wird der angegebene physikalische Ausgang (n) direkt eingeschaltet.

In AWL kopiert die Operation Bitwert direkt zuweisen den obersten Stackwert direkt in den angegebenen physikalischen Ausgang (n).

Operanden: n: A

Das "I" zeigt direkten Zugriff an. Der neue Wert wird bei Ausführung der Operation sowohl in das Prozeßabbild als auch direkt in den physikalischen Ausgang geschrieben. Hierin unterscheidet sich eine direkte Operation von anderen Operationen, bei denen der Wert für den angesprochenen Ein- bzw. Ausgang nur in das Prozeßabbild geschrieben wird.

Setzen, Rücksetzen

K O P	$\text{---} \left(\begin{matrix} S_BIT \\ S \\ N \end{matrix} \right)$ $\text{---} \left(\begin{matrix} S_BIT \\ R \\ N \end{matrix} \right)$
A W L	$S \quad S_BIT, N$ $R \quad S_BIT, N$
<input checked="" type="checkbox"/> 212 <input checked="" type="checkbox"/> 214 <input checked="" type="checkbox"/> 215 <input checked="" type="checkbox"/> 216	

Werden die Operationen **Setzen** und **Rücksetzen** ausgeführt, dann wird die angegebene Anzahl (N) an Ein- bzw. Ausgängen mit Beginn bei S_BIT gesetzt (eingeschaltet) bzw. rückgesetzt (ausgeschaltet).

Operanden: S_BIT: E, A, M, SM, T, Z, V, S
 N: EB, AB, MB, SMB, VB, AC, Konstante, *VD, *AC, SB

Der Bereich der Ein- und Ausgänge, der gesetzt oder rückgesetzt werden kann, liegt zwischen 1 und 255 Ein- bzw. Ausgängen. Wird für die Operation Rücksetzen für den Parameter S_BIT ein Zeit- oder Zählerbit angegeben, dann werden sowohl das Zeit-/Zählerbit als auch der aktuelle Wert der Zeit bzw. des Zählers rückgesetzt.

Direktes Setzen und Rücksetzen

K O P	S_BIT — (S_I) N
	S_BIT — (R_I) N
A W L	SI S_BIT, N
	RI S_BIT, N
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
212	214
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
215	216

Werden die Operationen **Bitwert direkt setzen** und **Bitwert direkt rücksetzen** ausgeführt, dann wird die angegebene Anzahl (N) an Ein- bzw. Ausgängen mit Beginn bei S_BIT gesetzt (eingeschaltet) bzw. rückgesetzt (ausgeschaltet).

Operanden: S_BIT: A
N: EB, AB, MB, SMB, VB, AC, Konstante, *VD, *AC, SB

Der Bereich der Ein- und Ausgänge, der gesetzt oder rückgesetzt werden kann, liegt zwischen 1 und 64 Ein- bzw. Ausgängen.

Das "I" zeigt direkten Zugriff an. Der neue Wert wird bei Ausführung der Operation sowohl in das Prozeßabbild als auch direkt in den physikalischen Ausgang geschrieben. Hierin unterscheidet sich eine direkte Operation von anderen Operationen, bei denen der Wert für den angesprochenen Ein- bzw. Ausgang nur in das Prozeßabbild geschrieben wird.

Nulloperation

K O P	N — (NOP)
	NOP N
A W L	NOP N
	NOP N
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
212	214
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
215	216

Die **Nulloperation** hat keinen Einfluß auf die Bearbeitung des Anwenderprogramms. Der Operand N ist eine Zahl zwischen 0 und 255.

Operanden: N: 0 bis 255

Die Nulloperation müssen Sie in einem Hauptprogramm, einem Unterprogramm oder einem Interruptprogramm anordnen.

Beispiel für Operationen mit Ausgängen

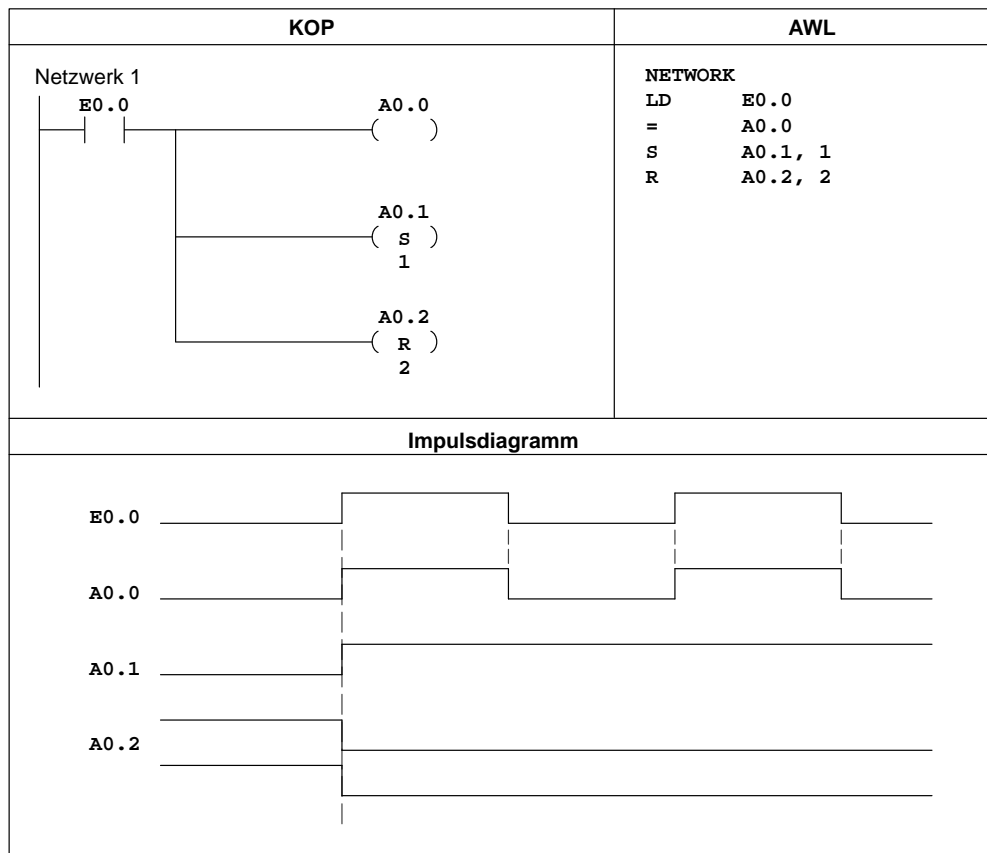
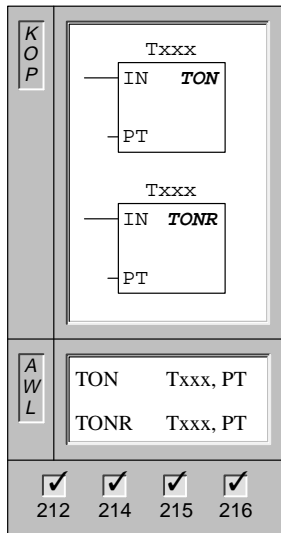


Bild 10-3 Beispiel für Operationen mit Ausgängen in KOP und AWL

10.5 Operationen mit Zeiten, Zählern, schnellen Zählern, schnellen Ausgängen und Impulsausgaben

Zeit als Einschaltverzögerung und als speichernde Einschaltverzögerung starten



Die Operationen **Zeit als Einschaltverzögerung starten** und **Zeit als speichernde Einschaltverzögerung starten** zählen bis zum maximalen Zeitwert, wenn Sie aktiviert werden. Ist der aktuelle Wert (Txxx) \geq dem voreingestellten Wert (PT), dann wird das Zeitbit eingeschaltet.

Wird die Operation **Zeit als Einschaltverzögerung starten** deaktiviert, dann wird die Zeit zurückgesetzt. Wird die Operation **Zeit als speichernde Einschaltverzögerung starten** deaktiviert, dann wird die Zeit angehalten. Beide Zeiten stoppen, wenn sie den Höchstwert erreicht haben.

Operanden:	Txxx:	<u>TON</u>	<u>TONR</u>
	1 ms	T32, T96	T0, T64
	10 ms	T33 bis T36 T97 bis T100	T1 bis T4 T65 bis T68
	100 ms	T37 bis T63 T101 bis T255	T5 bis T31 T69 bis T95
	PT:	VW, T, Z, EW, AW, MW, SMW, AC, AEW, Konstante, *VD, *AC, SW	

Die Zeiten TON und TONR stehen mit drei verschiedenen Auflösungen zur Verfügung. Die Auflösung richtet sich nach der Nummer der Zeit (siehe Tabelle 10-3). Jede Erhöhung um "1" beim aktuellen Wert stellt ein Mehrfaches der Zeitbasis dar. Bei einem Zähler mit einer Auflösung von 10 ms beispielsweise entspricht ein Zählwert von 50 dem aktuellen Wert 500 ms.

Tabelle 10-3 Nummern der Zeit und Auflösungen

Zeit	Auflösung	Höchstwert	CPU 212	CPU 214	CPU 215/216
TON	1 ms	32,767 Sekunden (s)	T32	T32, T96	T32, T96
	10 ms	327,67 s	T33 bis T36	T33 bis T36, T97 bis T100	T33 bis T36, T97 bis T100
	100 ms	3276,7 s	T37 bis T63	T37 bis T63, T101 bis T127	T37 bis T63, T101 bis T255
TONR	1 ms	32,767 s	T0	T0, T64	T0, T64
	10 ms	327,67 s	T1 bis T4	T1 bis T4, T65 bis T68	T1 bis T4, T65 bis T68
	100 ms	3276,7 s	T5 bis T31	T5 bis T31, T69 bis T95	T5 bis T31, T69 bis T95

Beschreibung der Zeitoperationen für die S7-200

Mit Zeiten können Sie zeitgesteuerte Funktionen ausführen. Die S7-200 verfügt über zwei verschiedene Zeitoperationen: Zeit als Einschaltverzögerung starten (TON) und Zeit als speichernde Einschaltverzögerung starten (TONR). Die beiden Zeiten (TON und TONR) unterscheiden sich in ihrer Reaktion auf den Zustand des Freigabeeingangs. Beide Zeiten TON und TONR zählen den Zeitwert, wenn der Freigabeeingang aktiviert ist. Bei ausgeschaltetem Freigabeeingang laufen die Zeiten nicht, doch während die Zeit TON automatisch zurückgesetzt wird, behält die Zeit TONR ihren letzten Zeitwert bei und wird nicht zurückgesetzt. Deshalb setzen Sie die Zeit TON am besten ein, wenn Sie ein einzelnes Zeitintervall benötigen. Die Zeit TONR verwenden Sie am besten, wenn Sie mehrere Zeitintervalle akkumulieren möchten.

Die Zeiten der S7-200 verfügen über folgende Eigenschaften:

- Die Zeiten werden über einen einzigen Freigabeeingang gesteuert und besitzen einen aktuellen Wert, der den Zeitwert angibt, der seit Freigabe der Zeit abgelaufen ist. Die Zeiten verfügen außerdem über einen voreingestellten Wert (PT), der jedesmal, wenn der aktuelle Wert aktualisiert oder die Zeitoperation ausgeführt wird, mit dem aktuellen Wert verglichen wird.
- Entsprechend dem Ergebnis des Vergleichs zwischen voreingestelltem und aktuellem Wert wird ein Zeitbit gesetzt bzw. zurückgesetzt.
- Ist der aktuelle Wert größer als oder gleich dem voreingestellten Wert, dann wird das Zeitbit (T-Bit) eingeschaltet.

Hinweis

Einige der aktuellen Zeitwerte können als remanent definiert werden. Die Zeitbits sind nicht remanent und werden nur als Ergebnis des Vergleichs zwischen aktuellem und voreingestelltem Wert gesetzt.

Wenn Sie eine Zeit zurücksetzen, dann wird der aktuelle Wert der Zeit auf Null gesetzt und das Zeitbit ausgeschaltet. Zum Rücksetzen können Sie für jede beliebige Zeit die Operation Rücksetzen verwenden. Die Zeit TONR kann allerdings nur mit der Operation Rücksetzen zurückgesetzt werden. Wenn Sie den Wert Null in den aktuellen Wert der Zeit schreiben, wird das Zeitbit nicht zurückgesetzt. Auch wird der aktuelle Wert der Zeit nicht zurückgesetzt, wenn Sie den Wert Null in das Zeitbit schreiben.

Mit mehreren Zeiten mit einer Auflösung von 1 ms können Sie ein Interruptereignis erzeugen. Ausführliche Informationen zu zeitgesteuerten Interrupts entnehmen Sie Abschnitt 10.14.

Aktualisieren von Zeiten mit einer Auflösung von 1 ms

Die S7-200 CPU verfügt über Zeiten, die einmal pro Millisekunde (Zeiten mit einer Auflösung von 1 ms) von dem Systemprogramm aktualisiert werden, das für die Aktualisierung der Zeitbasis zuständig ist. Diese Zeiten dienen zur präzisen Steuerung einer Operation.

Der aktuelle Wert einer aktiven Zeit mit einer Auflösung von 1 ms wird vom Systemprogramm automatisch aktualisiert. Nach der Freigabe der Zeit ist die Bearbeitung der Operationen TON/TONR für eine Zeit mit einer Auflösung von 1 ms nur erforderlich, um den EIN/AUS-Zustand der Zeit zu steuern.

Da der aktuelle Wert und das Zeitbit bei einer Zeit mit einer Auflösung von 1 ms vom Systemprogramm aktualisiert werden (unabhängig vom Zyklus des Automatisierungssystems und vom Anwenderprogramm), können der aktuelle Wert und das T-Bit dieser Zeit an beliebiger Stelle im Zyklus aktualisiert werden. Sie können in einem Zyklus auch mehrfach aktualisiert werden, sofern die Zykluszeit eine Millisekunde überschreitet. Deshalb ist es nicht gewährleistet, daß diese Werte während der Bearbeitung des Hauptprogramms konstant bleiben.

Wenn Sie eine freigegebene Zeit mit einer Auflösung von 1 ms zurücksetzen, dann wird die Zeit ausgeschaltet, der aktuelle Wert der Zeit auf Null gesetzt und das Zeitbit gelöscht.

Hinweis

Das Systemprogramm, das für die Zeitbasis des Systems von 1 ms zuständig ist, ist nicht vom Freigeben und Sperren der Zeiten abhängig. Eine Zeit mit einer Auflösung von 1 ms wird zu einem beliebigen Zeitpunkt innerhalb des aktuellen Intervalls von 1 ms freigegeben. Deshalb kann das Intervall für eine Zeit mit einer Auflösung von 1 ms maximal 1 ms lang sein. Sie müssen die Voreinstellung auf einen Wert setzen, der um 1 größer ist als das kürzeste gewünschte Intervall. Um beispielsweise ein Intervall von mindestens 56 ms zur Verfügung zu haben und dafür eine Zeit mit einer Auflösung von 1 ms zu verwenden, müssen Sie den voreingestellten Zeitwert auf 57 setzen.

Aktualisieren von Zeiten mit einer Auflösung von 10 ms

Die S7-200 CPU stellt Zeiten zur Verfügung, die die Anzahl der Intervalle von 10 ms zählen, die nach Freigabe der aktiven Zeit mit einer Auflösung von 10 ms vergangen sind. Diese Zeiten werden zu Beginn eines jeden Zyklus aktualisiert, indem die Anzahl der abgelaufenen Intervalle von 10 ms (seit Beginn des vorherigen Zyklus) zum aktuellen Wert der Zeit addiert werden.

Der aktuelle Wert einer aktiven Zeit mit einer Auflösung von 10 ms wird automatisch zu Beginn eines jeden Zyklus aktualisiert. Nach der Freigabe der Zeit ist die Bearbeitung der Operationen TON/TONR für eine Zeit mit einer Auflösung von 10 ms nur erforderlich, um den EIN/AUS-Zustand der Zeit zu steuern. Im Gegensatz zu den Zeiten mit einer Auflösung von 1 ms wird der aktuelle Wert einer Zeit mit einer Auflösung von 10 ms nur einmal pro Zyklus aktualisiert und ändert sich während der Bearbeitung des Anwenderprogramms nicht.

Wenn Sie eine freigegebene Zeit mit einer Auflösung von 10 ms zurücksetzen, dann wird die Zeit ausgeschaltet, der aktuelle Wert der Zeit auf Null gesetzt und das Zeitbit gelöscht.

Hinweis

Da das Akkumulieren der Intervalle von 10 ms unabhängig vom Freigeben und Sperren der Zeiten geschieht, werden Zeiten mit einer Auflösung von 10 ms innerhalb eines bestimmten Intervalls von 10 ms freigegeben. Deshalb kann das Intervall für eine Zeit mit einer Auflösung von 10 ms maximal 10 ms lang sein. Sie müssen die Voreinstellung auf einen Wert setzen, der um 1 größer ist als das kürzeste gewünschte Intervall. Um beispielsweise ein Intervall von mindestens 140 ms zur Verfügung zu haben und dafür eine Zeit mit einer Auflösung von 10 ms zu verwenden, müssen Sie den voreingestellten Zeitwert auf 15 setzen.

Aktualisieren von Zeiten mit einer Auflösung von 100 ms

Die meisten Zeiten, die von der S7-200 zur Verfügung gestellt werden, verwenden eine Auflösung von 100 ms. Diese Zeiten zählen die Anzahl der Intervalle mit 100 ms, die seit der letzten Aktualisierung dieser Zeit mit einer Auflösung von 100 ms vergangen sind. Zeiten mit einer Auflösung von 100 ms werden aktualisiert, indem der Akkumulationswert mit einer Auflösung von 100 ms (seit Beginn des vorherigen Zyklus) jedesmal dann zu dem aktuellen Wert der Zeit addiert wird, wenn die Zeitoperation ausgeführt wird.

Das Aktualisieren einer Zeit mit einer Auflösung von 100 ms geschieht nicht automatisch, da der aktuelle Wert der Zeit nur dann mit dem Akkumulationswert aktualisiert wird, wenn die Zeitoperation ausgeführt wird. Deshalb wird der aktuelle Wert einer Zeit nicht aktualisiert, wenn zwar eine Zeit freigegeben, aber die Zeitoperation nicht in jedem Zyklus ausgeführt wird. Dadurch verliert die Zeit an Wert. Wenn die Zeitoperation für eine Zeit mit einer Auflösung von 100 ms in einem Zyklus mehrfach ausgeführt wird, dann wird der Akkumulationswert mehrfach zu dem aktuellen Wert der Zeit addiert. Dadurch gewinnt die Zeit an Wert. Sie sollten also Zeiten mit einer Auflösung von 100 ms nur dann verwenden, wenn die Zeitoperation genau einmal pro Zyklus ausgeführt wird. Wird eine Zeit mit einer Auflösung von 100 ms zurückgesetzt, dann wird der aktuelle Wert der Zeit auf Null gesetzt und das Zeitbit gelöscht.

Hinweis

Da das Addieren der Intervalle von 100 ms unabhängig vom Freigeben und Sperren der Zeiten geschieht, werden Zeiten mit einer Auflösung von 100 ms zu einem beliebigen Zeitpunkt innerhalb eines bestimmten Intervalls von 100 ms freigegeben. Deshalb kann das Intervall für eine Zeit mit einer Auflösung von 100 ms maximal 100 ms lang sein. Sie müssen die Voreinstellung auf einen Wert setzen, der um 1 größer ist als das kürzeste gewünschte Intervall. Um beispielsweise ein Intervall von mindestens 2100 ms zur Verfügung zu haben und dafür eine Zeit mit einer Auflösung von 100 ms zu verwenden, müssen Sie den voreingestellten Zeitwert auf 22 setzen.

Aktualisieren des aktuellen Werts einer Zeit

Welche Auswirkungen das unterschiedliche Aktualisieren der aktuellen Werte von Zeiten hat, richtet sich danach, wie Sie die Zeiten einsetzen. Betrachten Sie beispielsweise die Zeitoperation in Bild 10-4.

- Wird eine Zeit mit einer Auflösung von 1 ms verwendet, dann wird A0.0 immer dann einen Zyklus lang eingeschaltet, wenn der aktuelle Wert der Zeit aktualisiert wird, und zwar nach der Ausführung des Öffnerkontakts T32 und vor der Ausführung des Schließerkontakts T32.
- Wird eine Zeit mit einer Auflösung von 10 ms verwendet, dann wird A0.0 niemals eingeschaltet, weil das Zeitbit T33 vom Beginn des Zyklus bis zu dem Zeitpunkt eingeschaltet ist, zu dem die Zeitbox ausgeführt wird. Nach der Ausführung der Zeitbox werden der aktuelle Wert der Zeit und das Zeitbit auf Null gesetzt. Wird der Schließerkontakt T33 ausgeführt, dann ist T33 nicht aktiviert und A0.0 wird ausgeschaltet.
- Wird eine Zeit mit einer Auflösung von 100 ms verwendet, dann ist A0.0 immer dann einen Zyklus lang eingeschaltet, wenn der aktuelle Wert der Zeit den voreingestellten Wert erreicht.

Wenn Sie den Öffnerkontakt A0.0 anstelle des Zeitbits als Freigabeeingang der Zeitbox verwenden, ist sichergestellt, daß der Ausgang A0.0 einen Zyklus lang eingeschaltet wird, sobald der Zeitwert gleich der Voreinstellung ist (siehe Bild 10-4). Die Bilder 10-5 und 10-6 zeigen Beispiele für die Zeitoperationen im Kontaktplan und in der Anweisungsliste.

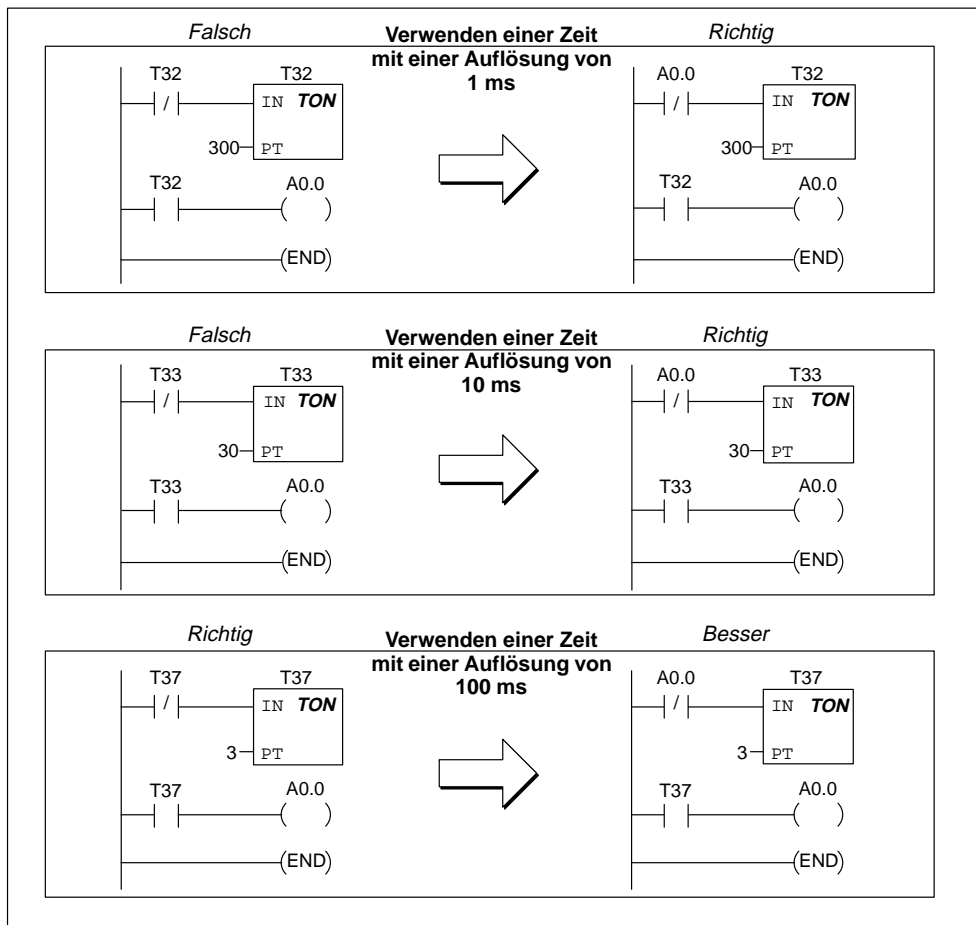


Bild 10-4 Beispiel für automatisches Auslösen einer Zeit

Beispiel für die Operation Zeit als Einschaltverzögerung starten

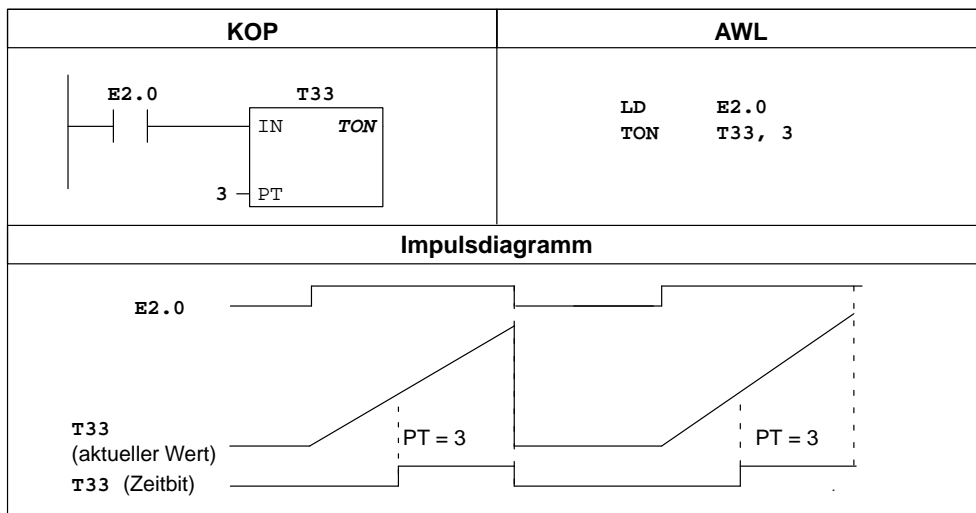


Bild 10-5 Beispiel für die Operation Zeit als Einschaltverzögerung starten in KOP und AWL

Beispiel für die Operation Zeit als speichernde Einschaltverzögerung starten

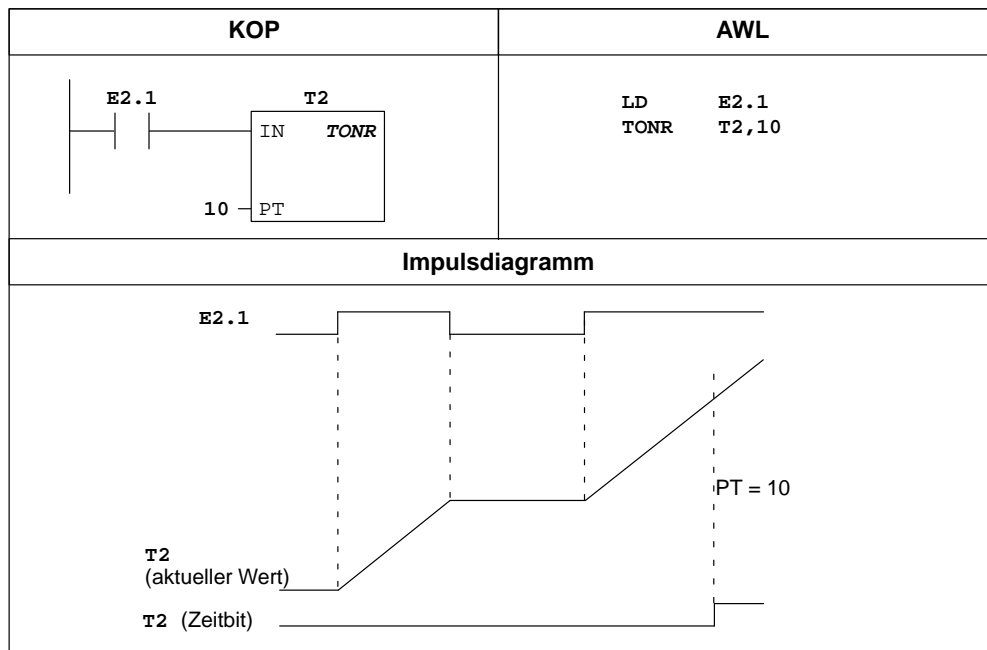
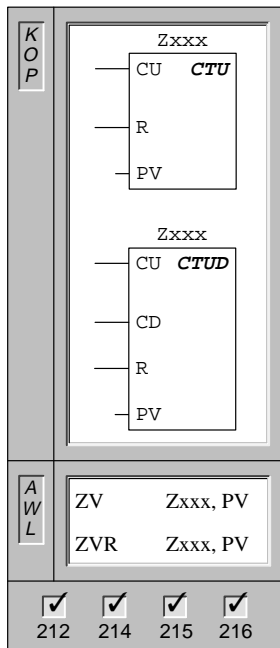


Bild 10-6 Beispiel für die Operation Zeit als speichernde Einschaltverzögerung starten in KOP und AWL

Vorwärtszählen und Vorwärts-/Rückwärtszählen



Die Operation **Vorwärtszählen** zählt bei steigender Flanke am Vorwärtszähleingang (CU) bis zum Höchstwert vorwärts. Ist der aktuelle Wert größer als oder gleich dem voreingestellten Wert (PV), dann wird das Zählerbit (Zxxx) aktiviert. Der Zähler wird zurückgesetzt, wenn der Rücksetzeingang aktiviert wird.

In AWL ist der Rücksetzeingang der oberste Stackwert, und der Vorwärtszähleingang ist der Wert in der zweiten Stackebene.

Die Operation **Vorwärts-/Rückwärtszählen** zählt bei steigender Flanke am Vorwärtszähleingang (CU) vorwärts. Bei steigender Flanke am Rückwärtszähleingang (CD) zählt die Operation rückwärts. Ist der aktuelle Wert (Zxxx) größer als oder gleich dem voreingestellten Wert (PV), dann wird das Zählerbit (Zxxx) aktiviert. Der Zähler wird zurückgesetzt, wenn der Rücksetzeingang (R) aktiviert wird.

In AWL ist der Rücksetzeingang der oberste Stackwert. Der Vorwärtszähleingang ist der Wert in der zweiten Stackebene und der Rückwärtszähleingang ist der Wert in der dritten Stackebene.

Operanden: Zxxx: 0 bis 255
 PV: VW, T, Z, EW, AW, MW, SMW, AC, AEW, Konstante, *VD, *AC, SW

Beschreibung der Zähloperationen für die S7-200

Der Vorwärtszähler (CTU/ZV) zählt bei steigender Flanke am Vorwärtszähleingang vom aktuellen Wert des Zählers an vorwärts. Der Zähler wird zurückgesetzt, wenn der Rücksetzeingang aktiviert wird oder die Operation Rücksetzen ausgeführt wird. Der Zähler stoppt, wenn der Höchstwert (32.767) erreicht ist.

Der Vorwärts-/Rückwärtszähler (CTUD/ZVR) zählt bei steigender Flanke am Vorwärtszähleingang vorwärts und bei steigender Flanke am Rückwärtszähleingang rückwärts. Der Zähler wird zurückgesetzt, wenn der Rücksetzeingang aktiviert wird oder die Operation Rücksetzen ausgeführt wird. Wird der Maximalwert (32.767) erreicht, bewirkt die nächste steigende Flanke am Vorwärtszähleingang, daß der Zähler umschlägt und erneut beim Minimalwert (-32.767) zu zählen beginnt. Wird beim Zählen der Minimalwert (-32.767) erreicht, schlägt der Zähler bei der nächsten steigenden Flanke am Rückwärtszähleingang um und zählt beim Maximalwert (32.767) weiter.

Wenn Sie einen Zähler mit der Operation Rücksetzen zurücksetzen, werden sowohl das Zählerbit als auch der aktuelle Wert des Zählers zurückgesetzt.

Die Vorwärts- und Vorwärts-/Rückwärtszähler verfügen über einen aktuellen Wert, in dem der aktuelle Zählwert gespeichert wird. Die Zähler verfügen auch über einen voreingestellten Wert (PV), der bei Ausführung der Operation mit dem aktuellen Wert verglichen wird. Ist der aktuelle Wert größer als oder gleich dem voreingestellten Wert, dann wird das Zählerbit (Z-Bit) aktiviert. In allen anderen Fällen wird das Zählerbit ausgeschaltet.

Mit der Nummer des Zählers sprechen Sie den aktuellen Wert und auch das Zählerbit des Zählers an.

Hinweis

Da jeder Zähler einen eigenen aktuellen Wert besitzt, dürfen Sie nicht mehreren Zählern die gleiche Nummer zuordnen (Vorwärts- und Vorwärts/Rückwärts-Zähler greifen auf den gleichen aktuellen Wert zu).

Beispiel für Zähloperationen

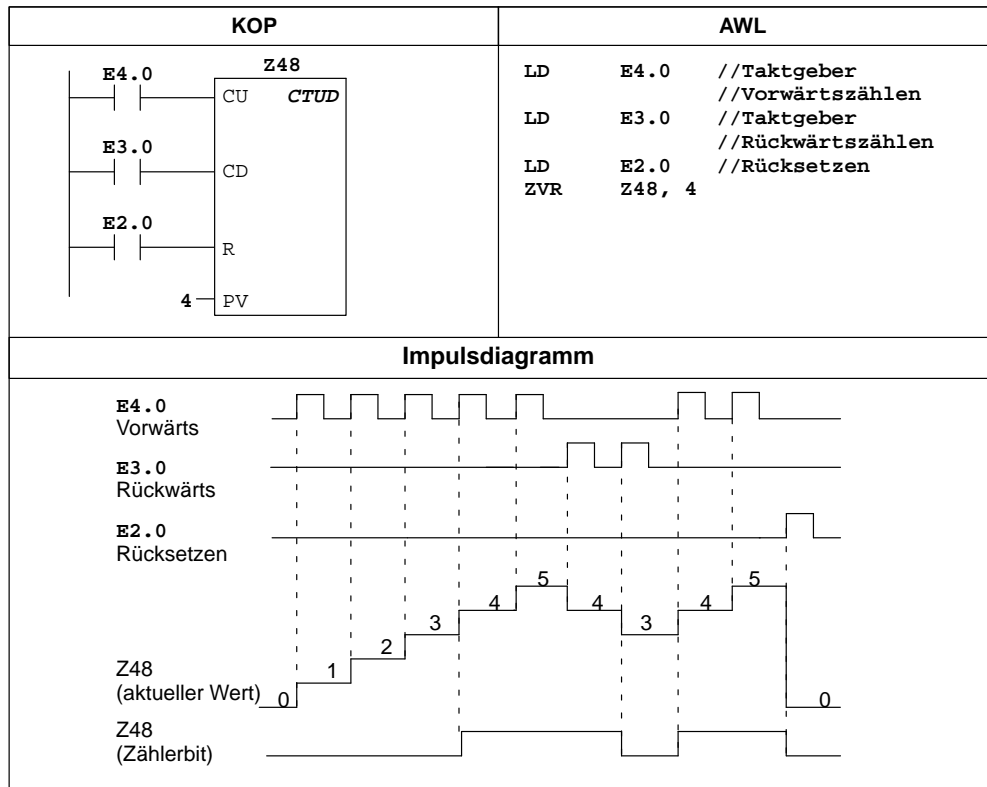
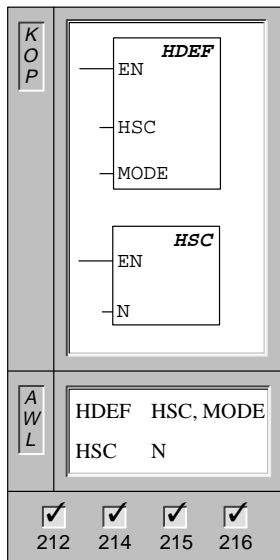


Bild 10-7 Beispiel für Zähloperationen in KOP und AWL

Modus für schnellen Zähler definieren, Schnellen Zähler aktivieren



Die Operation **Modus für schnellen Zähler definieren** weist dem angegebenen Zähler (HSC) einen Modus (MODE) zu (siehe Tabelle 10-5).

Die Operation **Schnellen Zähler aktivieren** konfiguriert und steuert den Betriebszustand der schnellen Zähler über den Signalzustand der Bits des Sondermerkers für den HSC. Der Parameter N gibt die Nummer des schnellen Zählers an.

Sie können für jeden Zähler nur eine Box HDEF verwenden.

Operanden: HSC: 0 bis 2
 MODE: 0 (HSC0)
 0 bis 11 (HSC1 oder 2)
 N: 0 bis 2

Beschreibung der Operationen mit schnellen Zählern

Schnelle Zähler zählen schnelle Ereignisse, die bei den Zyklusraten der CPUs nicht gesteuert werden können.

- HSC0 ist ein Vorwärts-/Rückwärtszähler, der einen Takteingang unterstützt. Das Programm steuert die Zählrichtung (vorwärts oder rückwärts) über ein richtungsteuerndes Bit. Die maximale Zählgeschwindigkeit dieses Zählers beträgt 2 kHz.
- HSC1 und HSC2 sind universelle Hardware-Zähler, die für eine von zwölf verschiedenen Zählerarten konfiguriert werden können. Die verschiedenen Zählerarten sind in Tabelle 10-5 aufgeführt. Die maximale Zählgeschwindigkeit der Zähler HSC1 und HSC2 richtet sich nach Ihrer CPU (siehe Anhang A).

Jeder Zähler verfügt über besondere Eingänge, die Funktionen wie Taktgeber, Richtungssteuerung, Rücksetzen und Starten unterstützen. Bei Zwei-Phasen-Zählern können beide Taktgeber mit maximaler Geschwindigkeit laufen. Bei A/B-Zählern können Sie einfache oder vierfache Zählgeschwindigkeiten auswählen. HSC1 und HSC2 sind vollkommen unabhängig voneinander und beeinflussen keine anderen schnellen Operationen. Beide Zähler laufen mit maximaler Geschwindigkeit, ohne sich gegenseitig zu beeinträchtigen.

Bild 10-16 zeigt ein Beispiel für die Initialisierung von HSC1.

Verwenden von schnellen Zählern

Schnelle Zähler werden typischerweise als Antrieb für Zählwerke eingesetzt, bei denen eine Welle, die mit einer konstanten Drehzahl läuft, mit einem Winkelschrittgeber versehen ist. 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 gewünschten Ausgänge werden für die Zeitspanne aktiviert, während der aktuelle Wert des Zählers kleiner als der voreingestellte Wert ist. Der Zähler wird so eingerichtet, daß ein Interrupt auftritt, wenn der aktuelle Wert des Zählers gleich dem voreingestellten Wert ist oder wenn der Zähler zurückgesetzt wird.

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 gesamten Zyklus des Automatisierungssystems implementiert werden. Da Sie Interrupts bestimmten Interruptprogrammen zuordnen können, kann jede neue Voreinstellung in einem getrennten Interruptprogramm geladen werden, damit so der Zustand einfach gesteuert werden kann und das Programm geradlinig und leicht zu verfolgen ist. Sie können natürlich auch alle Interruptereignisse in einem einzigen Interruptprogramm bearbeiten. Ausführliche Informationen hierzu entnehmen Sie dem Abschnitt zu Interrupt-Operationen.

Impulsdiagramme für schnelle Zähler

Die folgenden Impulsdiagramme (Bilder 10-8, 10-9, 10-10 und 10-11) zeigen, wie jeder Zähler entsprechend seiner Klasse arbeitet. Der Betrieb der Rücksetz- und Starteingänge ist in einem getrennten Impulsdiagramm dargestellt und gilt für alle Zähler, die diese Eingänge verwenden. In den Diagrammen für die Rücksetz- und Starteingänge ist die Aktivität beider Eingänge als "hoch" programmiert.

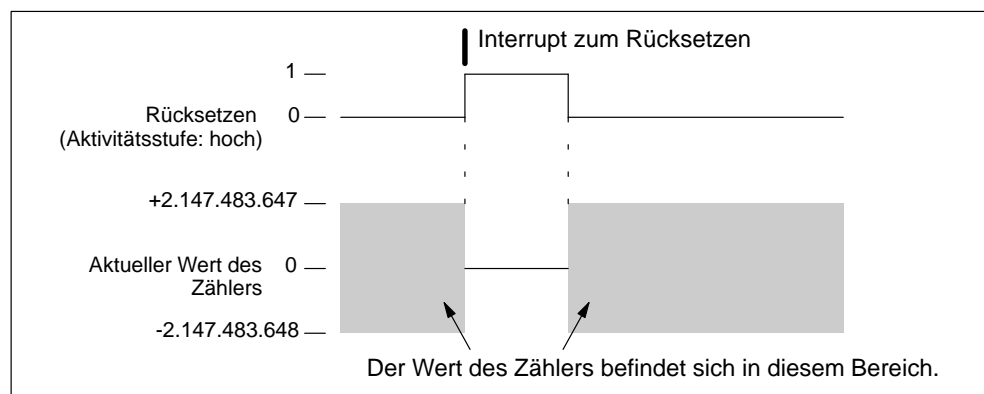


Bild 10-8 Beispiel für den Betrieb eines Zählers mit Rücksetz- und ohne Starteingang

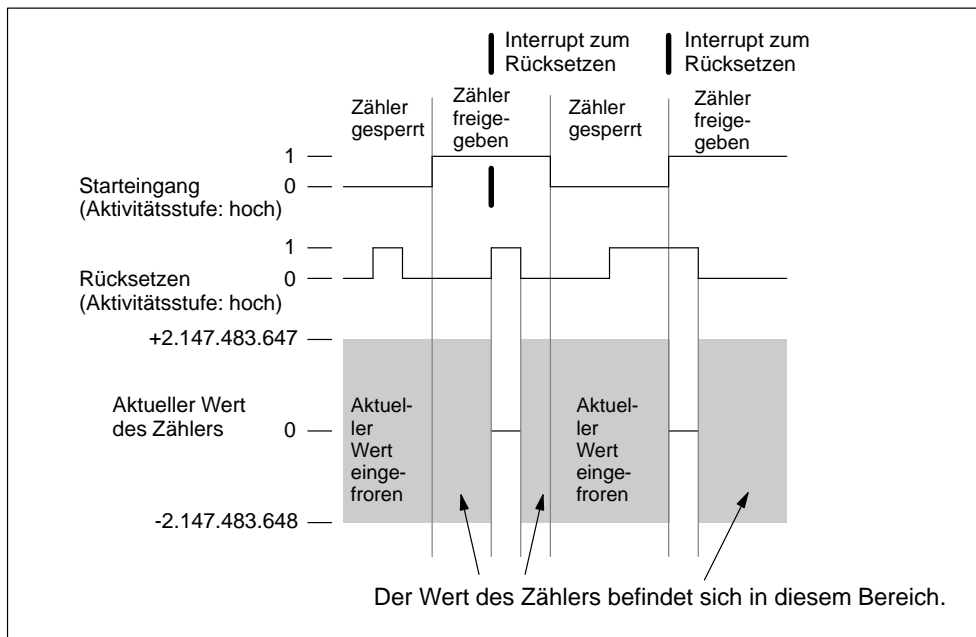


Bild 10-9 Beispiel für den Betrieb eines Zählers mit Rücksetz- und Starteingang

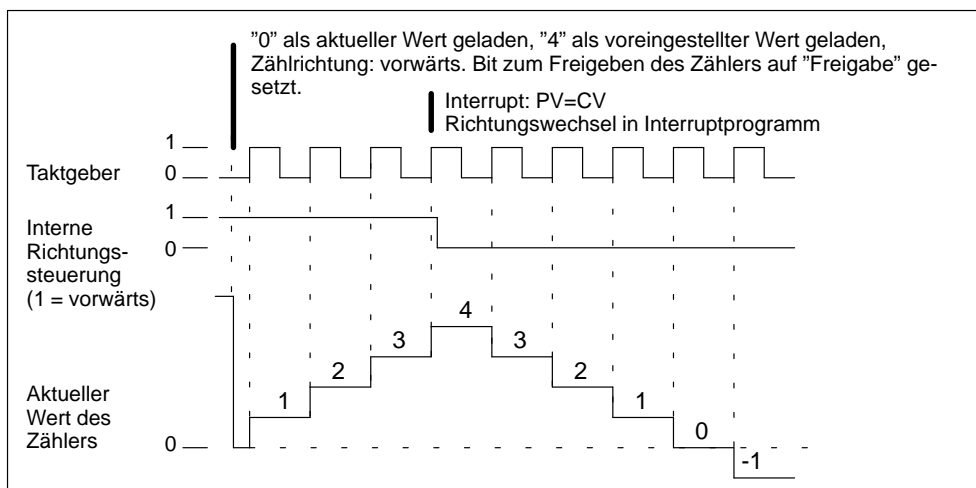


Bild 10-10 Beispiel für den Betrieb von HSC0 in Zählerart 0 und HSC1 bzw. HSC2 in einer der Zählerarten 0, 1 oder 2

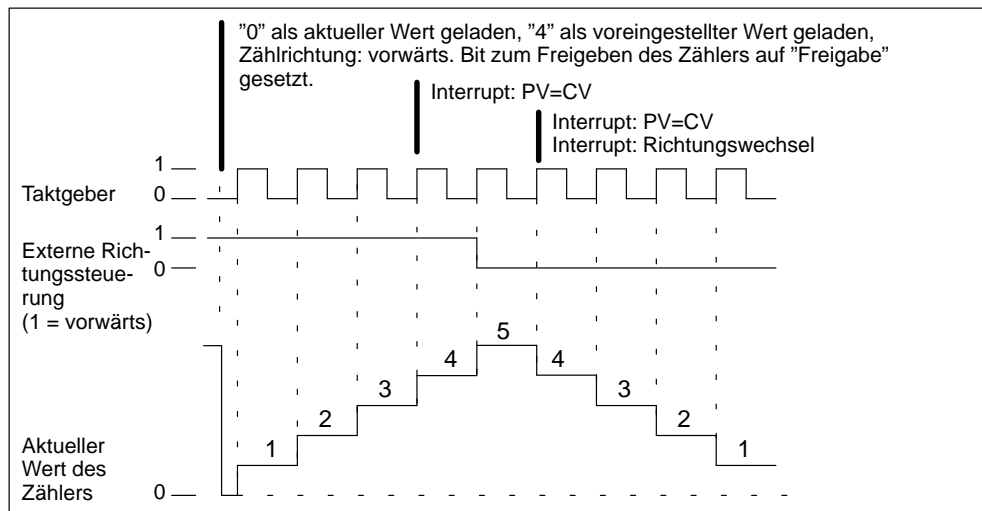


Bild 10-11 Beispiel für den Betrieb von HSC1 bzw. HSC2 in einer der Zählerarten 3, 4 oder 5

Wenn Sie für HSC1 oder HSC2 eine der Zählerarten 6, 7 oder 8 verwenden und dabei innerhalb von 0,3 Mikrosekunden sowohl am Vorwärts- als auch am Rückwärtszähleingang eine steigende Flanke auftritt, kann es sein, daß der schnelle Zähler diese beiden Ereignisse als simultan interpretiert. In diesem Fall wird der aktuelle Wert nicht geändert, und es wird auch kein Wechsel in der Zählrichtung angezeigt. Vergehen zwischen dem Auftreten einer steigenden Flanke am Vorwärts- und dem Auftreten einer steigenden Flanke am Rückwärtszähleingang mehr als 0,3 Mikrosekunden, dann empfängt der schnelle Zähler beide Ereignisse getrennt. In keinem der beiden Fälle tritt ein Fehler auf und der Zähler behält den korrekten Zählwert (siehe Bilder 10-12, 10-13 und 10-14).

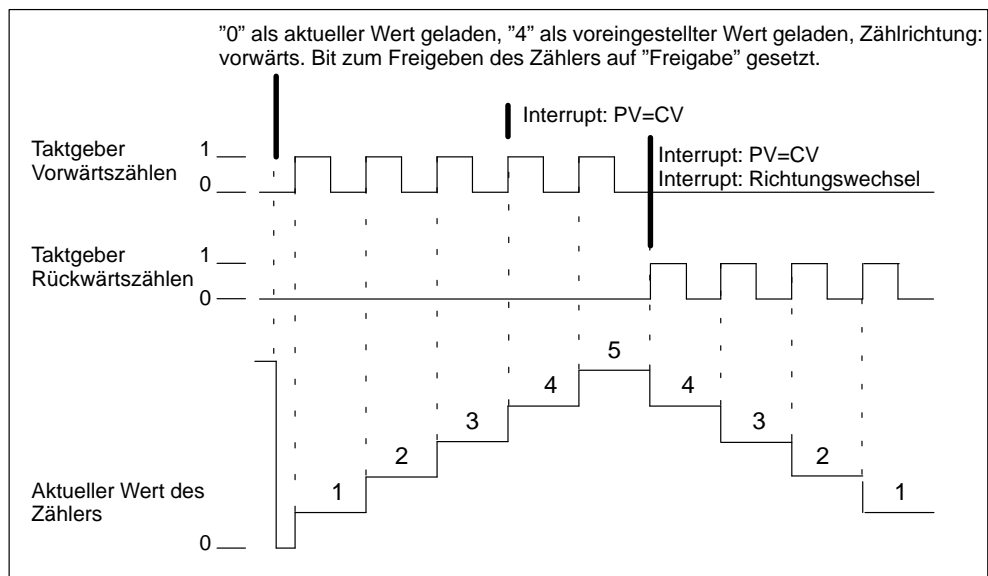


Bild 10-12 Beispiel für den Betrieb von HSC1 bzw. HSC2 in einer der Zählerarten 6, 7 oder 8

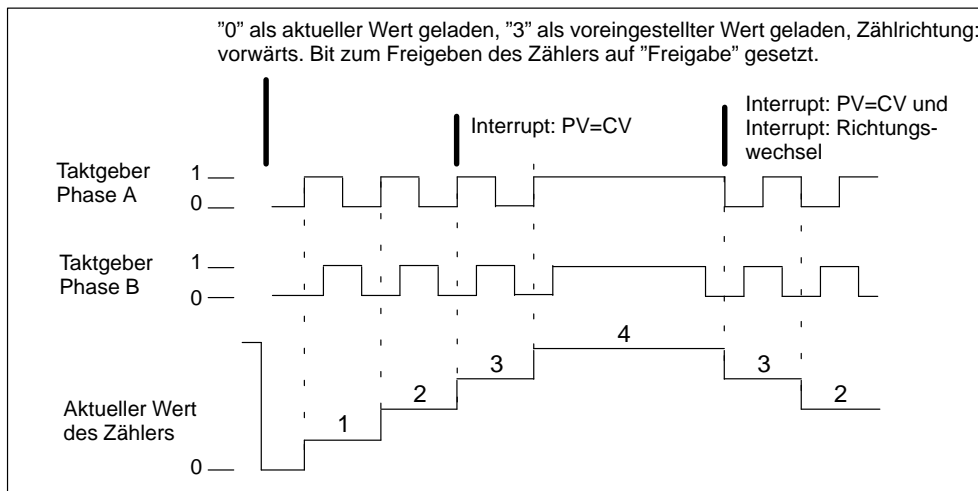


Bild 10-13 Beispiel für den Betrieb von HSC1 bzw. HSC2 in einer der Zählerarten 9, 10 oder 11 (A/B-Zähler, einfache Geschwindigkeit)

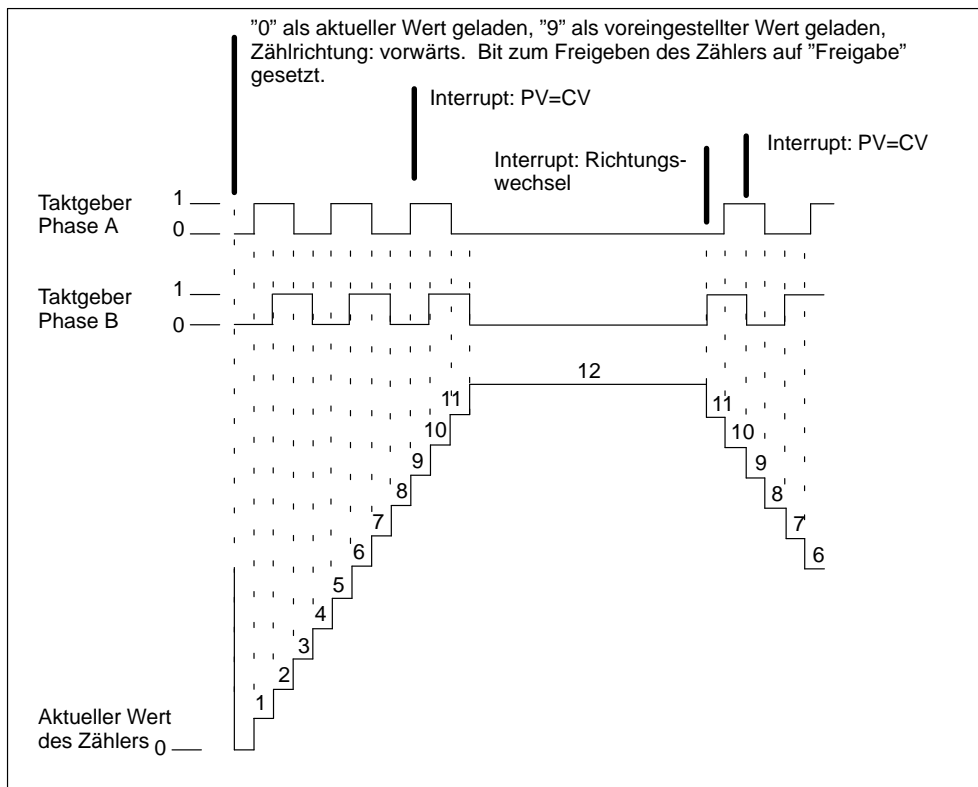


Bild 10-14 Beispiel für den Betrieb von HSC1 bzw. HSC2 in einer der Zählerarten 9, 10 oder 11 (A/B-Zähler, vierfache Geschwindigkeit)

Verdrahten der Eingänge der schnellen Zähler

Tabelle 10-4 zeigt die Eingänge von schnellen Zählern, die für Funktionen wie Taktgeber, Richtungssteuerung, Rücksetzen und Starten verwendet werden. Die Funktionen der Eingänge sind in Tabelle 10-5 beschrieben.

Tabelle 10-4 Eingänge der schnellen Zähler

Schneller Zähler	Belegte Eingänge
HSC0	E0.0
HSC1	E0.6, E0.7, E1.0, E1.1
HSC2	E1.2, E1.3, E1.4, E1.5

Adressierung von schnellen Zählern

Wenn Sie auf den Zählwert eines schnellen Zählers zugreifen möchten, geben Sie die Adresse des schnellen Zählers mittels des Speicherbereichs (HC) und der Nummer des Zählers (z. B. HC0) an. Der aktuelle Wert eines schnellen Zählers ist schreibgeschützt und kann nur im Doppelwortformat (32 Bit) adressiert werden (siehe Bild 10-15).

Format: $HC[Nummer\ des\ Zählers]$ HC1

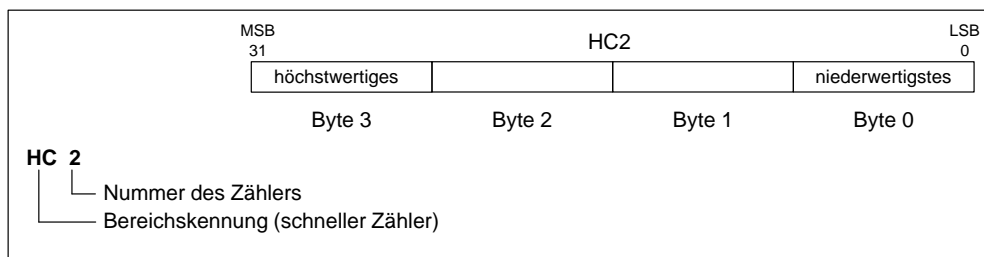


Bild 10-15 Zugriff auf den aktuellen Wert eines schnellen Zählers

Tabelle 10-5 Zählerarten der schnellen Zähler

HSC0					
Zählerart	Beschreibung	E0.0			
0	Ein-Phasen-Vorwärts-/Rückwärtszähler mit interner Richtungssteuerung SM37.3 = 0, Rückwärtszählen SM37.3 = 1, Vorwärtszählen	Taktgeber			
HSC1					
Zählerart	Beschreibung	E0.6	E0.7	E1.0	I1.1
0	Ein-Phasen-Vorwärts-/Rückwärtszähler mit interner Richtungssteuerung SM47.3 = 0, Rückwärtszählen SM47.3 = 1, Vorwärtszählen	Taktgeber		Rücksetzen	Starten
1					
2					
3	Ein-Phasen-Vorwärts-/Rückwärtszähler mit externer Richtungssteuerung E0.7 = 0, Rückwärtszählen E0.7 = 1, Vorwärtszählen	Taktgeber	Richtung	Rücksetzen	Starten
4					
5					
6	Zwei-Phasen-Zähler mit Takteingängen zum Vorwärts-/Rückwärtszählen	Taktgeber (vorwärts)	Taktgeber (rückwärts)	Rücksetzen	Starten
7					
8					
9	A/B-Zähler Phase A liegt 90 Grad vor B bei Rechtsdrehung, Phase B liegt 90 Grad vor A bei Linksdrehung	Taktgeber (Phase A)	Taktgeber (Phase B)	Rücksetzen	Starten
10					
11					
HSC2					
Zählerart	Beschreibung	I1.2	E1.3	I1.4	I1.5
0	Ein-Phasen-Vorwärts-/Rückwärtszähler mit interner Richtungssteuerung SM 57.3 = 0, Rückwärtszählen SM 57.3 = 1, Vorwärtszählen	Taktgeber		Rücksetzen	Starten
1					
2					
3	Ein-Phasen-Vorwärts-/Rückwärtszähler mit externer Richtungssteuerung E1.3 = 0, Rückwärtszählen E1.3 = 1, Vorwärtszählen	Taktgeber	Richtung	Rücksetzen	Starten
4					
5					
6	Zwei-Phasen-Zähler mit Takteingängen zum Vorwärts-/Rückwärtszählen	Taktgeber (vorwärts)	Taktgeber (rückwärts)	Rücksetzen	Starten
7					
8					
9	A/B-Zähler Phase A liegt 90 Grad vor B bei Rechtsdrehung, Phase B liegt 90 Grad vor A bei Linksdrehung	Taktgeber (Phase A)	Taktgeber (Phase B)	Rücksetzen	Starten
10					
11					

Beschreibung der unterschiedlichen schnellen Zähler (HSC0, HSC1, HSC2)

Alle Zähler (HSC0, HSC1 und HSC2) arbeiten in der gleichen Zählerart auf die gleiche Weise. Für HSC1 und HSC2 gibt es jeweils vier grundlegende Zählerarten (siehe Tabelle 10-5). Sie können jeden Zähler folgendermaßen verwenden: ohne Rücksetz- und Starteingang, mit Rücksetz- aber ohne Starteingang oder mit Rücksetz- und Starteingang.

Wenn Sie den Rücksetzeingang aktivieren, setzt dieser den aktuellen Wert zurück. Der aktuelle Wert bleibt solange zurückgesetzt, bis Sie den Rücksetzeingang deaktivieren. Wenn Sie den Starteingang aktivieren, beginnt der Zähler zu zählen. Wird der Starteingang deaktiviert, wird der aktuelle Wert des Zählers konstant gehalten und Taktereignisse werden ignoriert. Wird der Rücksetzeingang aktiviert, während der Starteingang nicht aktiv ist, dann wird das Rücksetzen ignoriert und der aktuelle Wert nicht verändert. Der Starteingang bleibt deaktiviert. Wird der Starteingang eingeschaltet, während der Rücksetzeingang aktiv ist, dann wird der aktuelle Wert gelöscht.

Sie müssen die Zählerart auswählen, bevor Sie einen schnellen Zähler einsetzen können. Hierzu verwenden Sie die Operation HDEF (Modus für schnellen Zähler definieren). HDEF ordnet einem schnellen Zähler (HSC0, HSC1 oder HSC2) eine Zählerart zu. Für jeden schnellen Zähler können Sie nur eine Operation HDEF ausführen. Einen schnellen Zähler definieren Sie, indem Sie mit dem Merker des ersten Zyklus SM0.1 (dieses Bit wird für den ersten Zyklus eingeschaltet und danach ausgeschaltet) ein Unterprogramm aufrufen, das die Operation HDEF enthält.

Auswählen der Aktivitätsstufe und der einfachen bzw. vierfachen Geschwindigkeit

HSC1 und HSC2 verfügen über drei Steuerbits, mit denen Sie den aktiven Zustand des Rücksetz- und Starteingangs konfigurieren und die einfache bzw. vierfache Geschwindigkeit (nur bei A/B-Zählern) auswählen können. Diese Bits befinden sich im Steuerbyte des entsprechenden Zählers und werden nur verwendet, wenn die Operation HDEF ausgeführt wird. Die Bits werden in Tabelle 10-6 beschrieben.

Bevor die Operation HDEF ausgeführt werden kann, müssen Sie die Steuerbits für HSC1 und HSC2 auf den gewünschten Zustand setzen. Andernfalls übernimmt der Zähler die voreingestellte Konfiguration für die gewählte Zählerart. Die voreingestellte Aktivität am Rücksetz- und am Starteingang ist für HSC1 und HSC2 "hoch". Bei A/B-Zählern ist die vierfache Zählgeschwindigkeit (viermal die Frequenz des Taktgebers) voreingestellt. Wird die Operation HDEF ausgeführt, können Sie die Zählereinstellung nicht mehr ändern, es sei denn, Sie versetzen die CPU in den Betriebszustand STOP.

Tabelle 10-6 Steuerbits für Aktivitätsstufe am Rücksetz- und Starteingang; Auswahlbits für einfache bzw. vierfache Geschwindigkeit von HSC1 bzw. HSC2

HSC1	HSC2	Beschreibung (nur wenn HDEF ausgeführt wird)
SM47.0	SM57.0	Steuerbit für Aktivitätsstufe am Rücksetzeingang: 0 - hoch, 1 - niedrig.
SM47.1	SM57.1	Steuerbit für Aktivitätsstufe am Starteingang: 0 - hoch, 1 - niedrig.
SM47.2	SM57.2	Zählgeschwindigkeit A/B-Zähler: 0 - vierfache Zählgeschwindigkeit, 1 - einfache Zählgeschwindigkeit.

Steuerbyte

Wenn Sie Zähler und Zählerart definiert haben, können Sie die dynamischen Parameter des Zählers programmieren. Jeder schnelle Zähler verfügt über ein Steuerbyte, das den Zähler aktiviert oder sperrt und auch festlegt, in welche Richtung gezählt werden soll (nur Zählerarten 0, 1, und 2). Das Steuerbyte gibt auch die Anfangszählrichtung für alle anderen Zählerarten sowie aktuelle und voreingestellte Werte, die geladen werden sollen, an. Das Steuerbyte und die zugeordneten aktuellen und voreingestellten Werte werden bei Ausführung der Operation HSC überprüft. Tabelle 10-7 beschreibt die der Steuerbits.

Tabelle 10-7 Steuerbits für HSC0, HSC1 und HSC2

HSC0	HSC1	HSC2	Beschreibung
SM37.0	SM47.0	SM57.0	Nach Ausführung von HDEF nicht verwendet (von HSC0 nie verwendet).
SM37.1	SM47.1	SM57.1	Nach Ausführung von HDEF nicht verwendet (von HSC0 nie verwendet).
SM37.2	SM47.2	SM57.2	Nach Ausführung von HDEF nicht verwendet (von HSC0 nie verwendet).
SM37.3	SM47.3	SM57.3	Steuerbit für Zählrichtung: 0 - rückwärts, 1 - vorwärts.
SM37.4	SM47.4	SM57.4	Zählrichtung in HSC schreiben: 0 - nicht aktualisieren, 1 - Richtung aktualisieren.
SM37.5	SM47.5	SM57.5	Neuen voreingestellten Wert in HSC schreiben: 0 - nicht aktualisieren, 1 - voreingestellten Wert aktualisieren.
SM37.6	SM47.6	SM57.6	Neuen aktuellen Wert in HSC schreiben: 0 = nicht aktualisieren; 1 = aktuellen Wert aktualisieren
SM37.7	SM47.7	SM57.7	Freigabe HSC: 0 - HSC sperren, 1 - HSC aktivieren.

Einstellen von aktuellen und voreingestellten Werten

Jeder schnelle Zähler verfügt über einen aktuellen Wert und einen voreingestellten Wert mit jeweils 32 Bits. Beide sind ganzzahlige Werte mit Vorzeichen. Um einen neuen aktuellen Wert oder einen voreingestellten Wert in den schnellen Zähler zu laden, müssen Sie das Steuerbyte und die Bytes des Sondermerkers einrichten, die die aktuellen Werte und/oder die voreingestellten Werte enthalten. Führen Sie dann die Operation HSC aus, um die neuen Werte in den schnellen Zähler zu übertragen. Tabelle 10-8 beschreibt die Bytes der Sondermerker, die die neuen aktuellen und voreingestellten Werte enthalten.

Zusätzlich zu den Steuerbytes und den Bytes, die die neuen aktuellen und voreingestellten Werte enthalten, kann der aktuelle Wert eines schnellen Zählers auch gelesen werden, indem der Speicherbereich HC (aktueller Wert des schnellen Zählers) und die Nummer des Zählers (0, 1 oder 2) angegeben werden. Auf diese Weise können Sie den aktuellen Wert direkt lesen. Zum Schreiben müssen Sie jedoch die oben beschriebene Operation HSC verwenden.

Tabelle 10-8 Aktuelle und voreingestellte Werte von HSC0, HSC1 und HSC2

Aktueller Wert von HSC0, HSC1 und HSC2			
HSC0	HSC1	HSC2	Beschreibung
SM38	SM48	SM58	Höchstwertiges Byte des neuen aktuellen Werts (32 Bit).
SM39	SM49	SM59	Zweithöchstwertiges Byte des neuen aktuellen Werts (32 Bit).
SM40	SM50	SM60	Zweitniederwertigstes Byte des neuen aktuellen Werts (32 Bit).
SM41	SM51	SM61	Zweitniederwertigstes Byte des neuen aktuellen Werts (32 Bit).
Voreingestellter Wert von HSC0, HSC1 und HSC2			
HSC0	HSC1	HSC2	Beschreibung
SM42	SM52	SM62	Höchstwertiges Byte des neuen voreingestellten Werts (32 Bit).
SM43	SM53	SM63	Zweithöchstwertiges Byte des neuen voreingestellten Werts (32 Bit).
SM44	SM54	SM64	Zweitniederwertigstes Byte des neuen voreingestellten Werts (32 Bit).
SM45	SM55	SM65	Zweitniederwertigstes Byte des neuen voreingestellten Werts (32 Bit).

Statusbyte

Jeder schnelle Zähler besitzt ein Statusbyte, das Statusmerker zur Verfügung stellt. Diese Statusbits geben die aktuelle Zählrichtung an. Sie geben außerdem an, ob der aktuelle Wert gleich dem voreingestellten Wert oder größer als dieser ist. Tabelle 10-9 beschreibt die Statusbits für die schnellen Zähler.

Tabelle 10-9 Statusbits für HSC0, HSC1 und HSC2

HSC0	HSC1	HSC2	Beschreibung
SM36.0	SM46.0	SM56.0	Nicht verwendet.
SM36.1	SM46.1	SM56.1	Nicht verwendet.
SM36.2	SM46.2	SM56.2	Nicht verwendet.
SM36.3	SM46.3	SM56.3	Nicht verwendet.
SM36.4	SM46.4	SM56.4	Nicht verwendet.
SM36.5	SM46.5	SM56.5	Statusbit für aktuelle Zählrichtung: 0 - rückwärts, 1 - vorwärts.
SM36.6	SM46.6	SM56.6	Statusbit für aktueller Wert gleich voreingestellter Wert: 0 - ungleich, 1 - gleich.
SM36.7	SM46.7	SM56.7	Statusbit für aktueller Wert ist größer als voreingestellter Wert: 0 - kleiner als oder gleich, 1 - größer als.

Hinweis

Die Statusbits für HSC0, HSC1 und HSC2 sind nur während der Bearbeitung des Interruptprogramms für den schnellen Zähler gültig. Wenn Sie die Zustände der schnellen Zähler überwachen, können Sie Interrupts für Ereignisse freigeben, die sich auf die bearbeitete Operation auswirken.

HSC-Interrupts

HSC0 unterstützt eine Interruptbedingung: Es tritt ein Interrupt auf, wenn der aktuelle Wert gleich dem voreingestellten Wert ist. HSC1 und HSC2 verfügen über drei Interruptbedingungen: Es tritt ein Interrupt auf, wenn der aktuelle Wert gleich dem voreingestellten Wert ist, wenn ein Rücksetzeingang extern aktiviert wird oder wenn die Zählrichtung wechselt. Jede dieser Interruptbedingungen kann einzeln freigegeben oder gesperrt werden. Ausführliche Informationen zur Verwendung von Interrupts entnehmen Sie dem Abschnitt zu den Interrupt-Operationen.

Die folgenden Beschreibungen zu Initialisierung und Bearbeitungsreihenfolge sollen Ihnen die Funktionsweise der schnellen Zähler näher erläutern. In den Beschreibungen dient HSC1 als Beispiel. Bei den Erklärungen zur Initialisierung wird davon ausgegangen, daß die S7-200 gerade in den Betriebszustand RUN versetzt wurde und der Merker des ersten Zyklus deshalb "wahr" ist. Ist dies nicht der Fall, bedenken Sie bitte, daß die Operation HDEF nur einmal für jeden schnellen Zähler ausgeführt werden kann, nachdem das System in den Betriebszustand RUN versetzt wurde. Wenn Sie die Operation HDEF ein zweites Mal für einen schnellen Zähler ausführen, tritt ein Fehler zur Laufzeit auf, und die Zählereinstellungen bleiben so, wie sie mit der ersten Operation HDEF für diesen Zähler eingerichtet wurden.

Initialisieren der Zählerarten 0, 1 und 2

Gehen Sie folgendermaßen vor, um HSC1 als Ein-Phasen-Vorwärts-/Rückwärtszähler mit interner Richtungssteuerung (Zählerarten 0, 1 oder 2) zu initialisieren:

1. Rufen Sie mit dem Merker des ersten Zyklus ein Unterprogramm auf, in dem die Initialisierung durchgeführt wird. Wenn Sie ein Unterprogramm aufrufen, rufen nachfolgende Zyklen das Unterprogramm nicht mehr auf, wodurch sich die Zykluszeit verkürzt und das Programm übersichtlicher strukturiert ist.
2. Im Unterprogramm für die Initialisierung laden Sie SM47 mit den gewünschten Einstellungen. Beispiel:
 SM47 = 16#F8 bewirkt folgende Ergebnisse:
 Aktiviert den Zähler.
 Schreibt einen neuen aktuellen Wert.
 Schreibt einen neuen voreingestellten Wert.
 Stellt Vorwärtszählen ein.
 Setzt die Aktivitätsstufe der Start- und Rücksetzeingänge auf "hoch".
3. Führen Sie die Operation HDEF aus. Der Eingang HSC ist dabei auf "1" gesetzt und der Eingang MODE entweder auf "0" für kein externes Rücksetzen bzw. Starten, auf "1" für externes Rücksetzen und kein Starten bzw. auf "2" für externes Rücksetzen und Starten.
4. Laden Sie den gewünschten aktuellen Wert in SM48 (Doppelwortwert). (Wenn Sie den Wert "0" laden, setzen Sie den Merker zurück).
5. Laden Sie den gewünschten voreingestellten Wert in SM52 (Doppelwortwert).
6. Wenn Sie das Ereignis "Aktueller Wert = Voreingestellter Wert" erkennen möchten, programmieren Sie einen Interrupt. Ordnen Sie hierzu das Interruptereignis CV = PV (Ereignis 13) einem Interruptprogramm zu. Ausführliche Informationen zur Interruptbearbeitung entnehmen Sie dem Abschnitt zu den Interrupt-Operationen in diesem Kapitel.
7. Wenn Sie externes Rücksetzen erkennen möchten, programmieren Sie einen Interrupt. Ordnen Sie hierzu das Interruptereignis Externes Rücksetzen (Ereignis 15) einem Interruptprogramm zu.
8. Führen Sie die Operation Alle Interruptereignisse freigeben (ENI) aus, um die Interrupts für HSC1 freizugeben.
9. Führen Sie dann die Operation HSC aus, damit die S7-200 den Zähler HSC1 programmiert.
10. Beenden Sie das Unterprogramm.

Initialisieren der Zählerarten 3, 4 und 5

Gehen Sie folgendermaßen vor, um HSC1 als Ein-Phasen-Vorwärts-/Rückwärtszähler mit externer Richtungssteuerung (Zählerarten 3, 4 oder 5) zu initialisieren:

1. Rufen Sie mit dem Merker des ersten Zyklus ein Unterprogramm auf, in dem die Initialisierung durchgeführt wird. Wenn Sie ein Unterprogramm aufrufen, rufen nachfolgende Zyklen das Unterprogramm nicht mehr auf, wodurch sich die Zykluszeit verkürzt und das Programm übersichtlicher strukturiert ist.
2. Im Unterprogramm für die Initialisierung laden Sie SM47 mit den gewünschten Einstellungen. Beispiel:
SM47 = 16#F8 bewirkt folgende Ergebnisse:
Aktiviert den Zähler.
Schreibt einen neuen aktuellen Wert.
Schreibt einen neuen voreingestellten Wert.
Stellt Vorwärtszählen ein.
Setzt die Aktivitätsstufe der Start- und Rücksetzeingänge auf "hoch".
3. Führen Sie die Operation HDEF aus. Der Eingang HSC ist dabei auf "1" gesetzt und der Eingang MODE entweder auf "3" für kein externes Rücksetzen bzw. Starten, auf "4" für externes Rücksetzen und kein Starten bzw. auf "5" für externes Rücksetzen und Starten.
4. Laden Sie den gewünschten aktuellen Wert in SM48 (Doppelwortwert). (Wenn Sie den Wert "0" laden, setzen Sie den Merker zurück).
5. Laden Sie den gewünschten voreingestellten Wert in SM52 (Doppelwortwert).
6. Wenn Sie das Ereignis "Aktueller Wert = Voreingestellter Wert" erkennen möchten, programmieren Sie einen Interrupt. Ordnen Sie hierzu das Interruptereignis CV = PV (Ereignis 13) einem Interruptprogramm zu. Ausführliche Informationen zur Interruptbearbeitung entnehmen Sie dem Abschnitt zu den Interrupt-Operationen in diesem Kapitel.
7. Wenn Sie einen Wechsel der Zählrichtung erkennen möchten, programmieren Sie einen Interrupt. Ordnen Sie hierzu das Interruptereignis Richtungswechsel (Ereignis 14) einem Interruptprogramm zu.
8. Wenn Sie externes Rücksetzen erkennen möchten, programmieren Sie einen Interrupt. Ordnen Sie hierzu das Interruptereignis Externes Rücksetzen (Ereignis 15) einem Interruptprogramm zu.
9. Führen Sie die Operation Alle Interruptereignisse freigeben (ENI) aus, um die Interrupts für HSC1 freizugeben.
10. Führen Sie dann die Operation HSC aus, damit die S7-200 den Zähler HSC1 programmiert.
11. Beenden Sie das Unterprogramm.

Initialisieren der Zählerarten 6, 7 und 8

Gehen Sie folgendermaßen vor, um HSC1 als Zwei-Phasen-Vorwärts-/Rückwärtszähler mit Vorwärts-/Rückwärts-Taktgebern (Zählerarten 6, 7 oder 8) zu initialisieren:

1. Rufen Sie mit dem Merker des ersten Zyklus ein Unterprogramm auf, in dem die Initialisierung durchgeführt wird. Wenn Sie ein Unterprogramm aufrufen, rufen nachfolgende Zyklen das Unterprogramm nicht mehr auf, wodurch sich die Zykluszeit verkürzt und das Programm übersichtlicher strukturiert ist.
2. Im Unterprogramm für die Initialisierung laden Sie SM47 mit den gewünschten Einstellungen. Beispiel:
 SM47 = 16#F8 bewirkt folgende Ergebnisse:
 Aktiviert den Zähler.
 Schreibt einen neuen aktuellen Wert.
 Schreibt einen neuen voreingestellten Wert.
 Stellt Vorwärtszählen ein.
 Setzt die Aktivitätsstufe der Start- und Rücksetzeingänge auf "hoch".
3. Führen Sie die Operation HDEF aus. Der Eingang HSC ist dabei auf "1" gesetzt und der Eingang MODE entweder auf "6" für kein externes Rücksetzen bzw. Starten, auf "7" für externes Rücksetzen und kein Starten bzw. auf "8" für externes Rücksetzen und Starten.
4. Laden Sie den gewünschten aktuellen Wert in SM48 (Doppelwortwert). (Wenn Sie den Wert "0" laden, setzen Sie den Merker zurück).
5. Laden Sie den gewünschten voreingestellten Wert in SM52 (Doppelwortwert).
6. Wenn Sie das Ereignis "Aktueller Wert = Voreingestellter Wert" erkennen möchten, programmieren Sie einen Interrupt. Ordnen Sie hierzu das Interruptereignis CV = PV (Ereignis 13) einem Interruptprogramm zu. Ausführliche Informationen zur Interruptbearbeitung entnehmen Sie dem Abschnitt zu den Interrupt-Operationen in diesem Kapitel.
7. Wenn Sie einen Wechsel der Zählrichtung erkennen möchten, programmieren Sie einen Interrupt. Ordnen Sie hierzu das Interruptereignis Richtungswechsel (Ereignis 14) einem Interruptprogramm zu.
8. Wenn Sie externes Rücksetzen erkennen möchten, programmieren Sie einen Interrupt. Ordnen Sie hierzu das Interruptereignis Externes Rücksetzen (Ereignis 15) einem Interruptprogramm zu.
9. Führen Sie die Operation Alle Interruptereignisse freigeben (ENI) aus, um die Interrupts für HSC1 freizugeben.
10. Führen Sie dann die Operation HSC aus, damit die S7-200 den Zähler HSC1 programmiert.
11. Beenden Sie das Unterprogramm.

Initialisieren der Zählerarten 9, 10 und 11

Gehen Sie folgendermaßen vor, um HSC1 als A/B-Zähler (Zählerarten 9, 10 oder 11) zu initialisieren:

1. Rufen Sie mit dem Merker des ersten Zyklus ein Unterprogramm auf, in dem die Initialisierung durchgeführt wird. Wenn Sie ein Unterprogramm aufrufen, rufen nachfolgende Zyklen das Unterprogramm nicht mehr auf, wodurch sich die Zykluszeit verkürzt und das Programm übersichtlicher strukturiert ist.
2. Im Unterprogramm für die Initialisierung laden Sie SM47 mit den gewünschten Einstellungen.

Beispiel (einfache Zählgeschwindigkeit):

SM47 = 16#FC bewirkt folgende Ergebnisse:
Aktiviert den Zähler.
Schreibt einen neuen aktuellen Wert.
Schreibt einen neuen voreingestellten Wert.
Stellt Vorwärtszählen ein.
Setzt die Aktivitätsstufe der Start- und Rücksetzeingänge auf "hoch".

Beispiel (vierfache Zählgeschwindigkeit):

SM47 = 16#F8 bewirkt folgende Ergebnisse:
Aktiviert den Zähler.
Schreibt einen neuen aktuellen Wert.
Schreibt einen neuen voreingestellten Wert.
Stellt Vorwärtszählen ein.
Setzt die Aktivitätsstufe der Start- und Rücksetzeingänge auf "hoch".

3. Führen Sie die Operation HDEF aus. Der Eingang HSC ist dabei auf "1" gesetzt und der Eingang MODE entweder auf "9" für kein externes Rücksetzen bzw. Starten, auf "10" für externes Rücksetzen und kein Starten bzw. auf "11" für externes Rücksetzen und Starten.
4. Laden Sie den gewünschten aktuellen Wert in SM48 (Doppelwortwert). (Wenn Sie den Wert "0" laden, setzen Sie den Merker zurück).
5. Laden Sie den gewünschten voreingestellten Wert in SM52 (Doppelwortwert).
6. Wenn Sie das Ereignis "Aktueller Wert = Voreingestellter Wert" erkennen möchten, programmieren Sie einen Interrupt. Ordnen Sie hierzu das Interruptereignis CV = PV (Ereignis 13) einem Interruptprogramm zu. Ausführliche Informationen zur Interruptbearbeitung entnehmen Sie dem Abschnitt zu den Interrupt-Operationen in diesem Kapitel.
7. Wenn Sie einen Wechsel der Zählrichtung erkennen möchten, programmieren Sie einen Interrupt. Ordnen Sie hierzu das Interruptereignis Richtungswechsel (Ereignis 14) einem Interruptprogramm zu.
8. Wenn Sie externes Rücksetzen erkennen möchten, programmieren Sie einen Interrupt. Ordnen Sie hierzu das Interruptereignis Externes Rücksetzen (Ereignis 15) einem Interruptprogramm zu.
9. Führen Sie die Operation Alle Interruptereignisse freigeben (ENI) aus, um die Interrupts für HSC1 freizugeben.
10. Führen Sie dann die Operation HSC aus, damit die S7-200 den Zähler HSC1 programmiert.
11. Beenden Sie das Unterprogramm.

Richtungswechsel in der Zählerart 0, 1 oder 2

Gehen Sie folgendermaßen vor, um für HSC1 als Ein-Phasen-Zähler mit interner Richtungssteuerung (Zählerarten 0, 1 oder 2) einen Richtungswechsel zu konfigurieren:

1. Laden Sie SM47, um die gewünschte Richtung einzustellen.
 - SM47 = 16#90 Aktiviert den Zähler.
Setzt die Zählrichtung des HSC auf Rückwärtszählen.
 - SM47 = 16#98 Aktiviert den Zähler.
Setzt die Zählrichtung des HSC auf Vorwärtszählen.
2. Führen Sie dann die Operation HSC aus, damit die S7-200 den Zähler HSC1 programmiert.

Laden eines neuen aktuellen Werts (beliebige Zählerart)

Gehen Sie folgendermaßen vor, um den aktuellen Wert von HSC1 (beliebige Zählerart) zu ändern:

Wenn Sie den aktuellen Wert ändern, wird der Zähler automatisch gesperrt. Solange er gesperrt ist, wird weder gezählt, noch können Interrupts auftreten.

1. Laden Sie SM47, um den gewünschten aktuellen Wert einzugeben.
 - SM47 = 16#C0 Aktiviert den Zähler.
Schreibt einen neuen aktuellen Wert.
2. Laden Sie den gewünschten aktuellen Wert in SM48 (Doppelwortwert). (Wenn Sie den Wert "0" laden, setzen Sie den Merker zurück).
3. Führen Sie dann die Operation HSC aus, damit die S7-200 den Zähler HSC1 programmiert.

Laden eines neuen voreingestellten Werts (beliebige Zählerart)

Gehen Sie folgendermaßen vor, um den voreingestellten Wert von HSC1 (beliebige Zählerart) zu ändern:

1. Laden Sie SM47, um den gewünschten voreingestellten Wert einzugeben.
 - SM47 = 16#A0 Aktiviert den Zähler.
Schreibt einen neuen voreingestellten Wert.
2. Laden Sie den gewünschten voreingestellten Wert in SM52 (Doppelwortwert).
3. Führen Sie dann die Operation HSC aus, damit die S7-200 den Zähler HSC1 programmiert.

Sperren eines schnellen Zählers (beliebige Zählerart)

Gehen Sie folgendermaßen vor, um den schnellen Zähler HSC1 (beliebige Zählerart) zu sperren:

1. Laden Sie SM47, um den schnellen Zähler zu sperren.
 - SM47 = 16#00 Sperrt den Zähler.
2. Führen Sie die Operation HSC aus, damit die S7-200 den Zähler HSC1 sperrt.

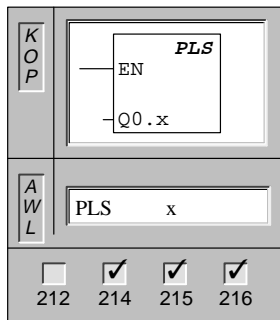
Die oben beschriebenen Vorgehensweisen zeigen Ihnen, wie Sie die Richtung, den aktuellen Wert oder den voreingestellten Wert einzeln ändern. Sie können aber auch mehrere oder alle Einstellungen in der oben aufgeführten Reihenfolge ändern, indem Sie den Wert von SM47 entsprechend einstellen und anschließend die Operation HSC ausführen.

Beispiel für schnelle Zähler

KOP	AWL
<p>Netzwerk 1</p> <p>SM0.1 ———— 0 ———— (CALL) 0 Im ersten Zyklus Unterprogramm 0 aufrufen.</p> <p>Netzwerk 2</p> <p>————— (END) ———— Hauptprogramm beenden.</p> <p>Netzwerk 3</p> <p>0 ———— SBR Unterprogramm 0 beginnen.</p> <p>Netzwerk 4</p> <p>SM0.0 ———— EN ———— MOV_B 16#F8 ———— IN OUT ———— SMB47 Setzt die Aktivitätsstufe von Start- und Rücksetzeingängen auf "hoch". Stellt vierfache Geschwindigkeit ein.</p> <p>1 ———— EN ———— HDEF 11 ———— HSC MODE HSC1, konfiguriert für vierfache Geschwindigkeit mit Rücksetz- und Starteingängen.</p> <p>0 ———— EN ———— MOV_DW IN OUT ———— SMD48 Aktuellen Wert von HSC1 zurücksetzen.</p> <p>50 ———— EN ———— MOV_DW IN OUT ———— SMD52 Voreingestellten Wert von HSC1 auf 50 setzen.</p> <p>0 ———— EN ———— ATCH 13 ———— INT EVENT Aktueller Wert von HSC1 = voreingestellter Wert (Ereignis 13), ist Interruptprogramm 0 zugeordnet.</p> <p>————— (ENI) ———— Alle Interruptereignisse freigeben.</p> <p>1 ———— EN ———— HSC N HSC1 programmieren.</p> <p>Netzwerk 5</p> <p>————— (RET) ———— Unterprogramm beenden.</p> <p>Netzwerk 6</p> <p>0 ———— INT Interrupt 0 beginnen.</p> <p>Netzwerk 7</p> <p>SM0.0 ———— EN ———— MOV_DW 0 ———— IN OUT ———— SMD48 Aktuellen Wert von HSC1 zurücksetzen.</p> <p>16#C0 ———— EN ———— MOV_B IN OUT ———— SMB47 Neuen aktuellen Wert schreiben und Zähler freigeben.</p> <p>1 ———— EN ———— HSC N HSC1 programmieren.</p> <p>Netzwerk 8</p> <p>————— (RETI) ———— Interruptprogramm beenden.</p>	<p>Network 1</p> <p>LD SM0.1 CALL 0</p> <p>Network 2</p> <p>MEND</p> <p>Network 3</p> <p>SBR 0</p> <p>Network 4</p> <p>LD SM0.0 MOVB 16#F8, SMB47 HDEF 1, 11 MOVD 0, SMD48 MOVD 50, SMD52 ATCH 0, 13 ENI HSC 1</p> <p>Network 5</p> <p>RET</p> <p>Network 6</p> <p>INT 0</p> <p>Network 7</p> <p>LD SM0.0 MOVD 0, SMD48 MOVB 16#C0, SMB47 HSC 1</p> <p>Network 8</p> <p>RETI</p>

Bild 10-16 Beispiel für die Initialisierung von HSC1 in KOP und AWL

Impulsausgabe



Die Operation **Impulsausgabe** prüft die Sondermerker für diesen Impulsausgang (0.x). Die Impulsoperation, die in den Sondermerkern definiert ist, wird dann aufgerufen.

Operanden: x: 0 bis 1

Beschreibung der Operationen mit schnellen Ausgängen für die S7-200

Einige CPUs können über die Ausgänge A0.0 und A0.1 entweder schnelle Impulsfolgen (PTO) erzeugen oder die Impulsdauermodulation (PWM) steuern. Die Funktion Impulsfolge stellt einen Rechteckausgang (50% relative Einschaltdauer) für eine bestimmte Anzahl von Impulsen und eine festgelegte Zykluszeit zur Verfügung. Die Anzahl der Impulse kann zwischen 1 und 4.294.967.295 liegen. Die Zykluszeit kann in Mikrosekunden (von 250 bis 65.535) oder in Millisekunden (von 2 bis 65.535) angegeben werden. Eine ungerade Mikro- oder Millisekundenzahl ruft eine Verzerrung der relativen Einschaltdauer hervor.

Die Funktion PWM bietet Ihnen eine feste Zykluszeit mit variabler relativer Einschaltdauer. Zykluszeit und Impulsdauer können in Mikro- oder Millisekunden angegeben werden. Die Zykluszeit liegt zwischen 250 und 65.535 Mikrosekunden oder zwischen 2 und 65.535 Millisekunden. Die Impulsdauer liegt zwischen 0 und 65.535 Mikrosekunden oder zwischen 0 und 65.535 Millisekunden. Sind Impulsdauer und Zykluszeit gleich, dann beträgt die relative Einschaltdauer 100% und der Ausgang ist ständig eingeschaltet. Ist die Impulsdauer Null, dann beträgt auch die relative Einschaltdauer 0% und der Ausgang wird ausgeschaltet.

Wird eine Zykluszeit von weniger als zwei Zeiteinheiten angegeben, dann stellt die Zykluszeit zwei Zeiteinheiten als Voreinstellung ein.

Hinweis

Bei den Funktionen PTO und PWM sind die Schaltzeiten der Ausgänge für die Schaltvorgänge ein/aus und aus/ein verschieden. Dieser Unterschied in den Schaltzeiten zeigt sich als Verzerrung der relativen Einschaltdauer. Ausführliche Angaben zu den Schaltzeiten entnehmen Sie Anhang A. Die Ausgänge PTO/PWM benötigen eine Mindestlast von 10% der Nennlast, um saubere Übergänge (ein/aus und aus/ein) erzeugen zu können.

Ändern der Impulsdauer

Die Funktion PWM wird kontinuierlich ausgeführt. Soll die Impulsdauer geändert werden, muß die Funktion PWM kurzzeitig zum Aktualisieren gesperrt werden. Dies geschieht asynchron zum PWM-Zyklus und könnte ein unerwünschtes Impulszittern in dem gesteuerten Gerät verursachen. Sind synchrone Aktualisierungen der Impulsdauer erforderlich, wird der Impulsausgang an einen der Interrupteingänge (E0.0 bis E0.4) zurückgeleitet. Sie können den PWM-Zyklus synchronisieren, indem Sie zu dem Zeitpunkt, zu dem die Impulsdauer geändert werden soll, am Interrupteingang den Interrupt Steigende Flanke freigeben. Bild 10-19 zeigt Ihnen hierzu ein Beispiel.

Die Impulsdauer wird im Interruptprogramm geändert und das Interruptereignis vom Programm getrennt bzw. gesperrt. Dadurch treten außer beim Ändern der Impulsdauer keine Interrupts auf.

Aufrufen der Funktionen PTO/PWM

Jeder PTO/PWM-Generator verfügt über ein Steuerbyte (8 Bit), jeweils einen Wert für die Zykluszeit und die Impulsdauer (beides vorzeichenlose 16-Bit-Werte) und einen Impulszählwert (vorzeichenloser 32-Bit-Wert). Diese Werte sind in bestimmten Sondermerkern abgelegt. Nachdem Sie die Sondermerker für eine spezifische Funktion konfiguriert haben, können Sie die Funktion mit der Operation Impulsausgabe (PLS) aufrufen. Wird die Operation ausgeführt, liest die S7-200 die Sondermerker und programmiert den PTO/PWM-Generator entsprechend der Konfiguration.

PTO-Pipeline

Zusätzlich zu den Steuerungsinformationen verfügt die Operation PTO über zwei Statusbits, die anzeigen, ob die angegebene Anzahl Impulse erzeugt wurde oder ob ein Pipeline-Überlauf aufgetreten ist.

Bei der Funktion PTO können Sie die Definitionen von zwei Impulsausgängen entweder miteinander verknüpfen oder nach der Pipeline-Methode miteinander verketteten. Dadurch kann die Kontinuität von aufeinanderfolgenden Impulsfolgen an den Ausgängen unterstützt werden. Laden Sie die Pipeline, indem Sie zunächst die erste PTO-Definition einrichten und anschließend die Operation PLS ausführen. Unmittelbar nachdem Sie die Operation PLS ausgeführt haben, können Sie die zweite Definition einrichten und die Operation PLS erneut ausführen.

Sollten Sie eine dritte Definition einrichten, bevor die erste Operation PTO beendet wurde (d.h. bevor die vollständige Anzahl der Ausgangsimpulse der ersten Definition erzeugt wurde), dann wird das Bit Überlauf PTO-Pipeline (SM66.6 bzw. SM76.6) auf "1" gesetzt. Wenn Sie die CPU in den Betriebszustand RUN versetzen, wird dieses Bit auf "0" gesetzt. Nach einem Überlauf muß das Bit vom Programm wieder auf "0" gesetzt werden, damit spätere Überläufe erkannt werden können.

Tabelle 10-10 zeigt die Sondermerker für die Impulsausgänge 0 und 1.

Hinweis

Die Voreinstellung für die Steuerbits, die Zykluszeit, die Impulsdauer und den Impulszählwert ist Null.

Tabelle 10-10 Sondermerker für die Pipeline-Verkettung von zwei Impulsausgängen mit den Funktionen PTO und PWM

A0.0	A0.1	Statusbit für Impulsausgänge	
SM66.6	SM76.6	Überlauf PTO-Pipeline	0 - kein Überlauf, 1 - Überlauf
SM66.7	SM76.7	PTO-Leerlauf	0 - Bearbeitung, 1 - Leerlauf
A0.0	A0.1	Steuerbits für PTO/PWM-Ausgänge	
SM67.0	SM77.0	PTO/PWM: Zykluszeit aktualisieren	0 - keine Aktualisierung, 1 - Zykluszeit aktualisieren
SM67.1	SM77.1	PWM: Zeitwert der Impulsdauer aktualis.	0 - keine Aktualisierung, 1 - Impulsdauer aktualisieren
SM67.2	SM77.2	PTO: Impulswert aktualisieren	0 - keine Aktualisierung, 1 - Impulswert aktualisieren
SM67.3	SM77.3	PTO/PWM: Zeitbasis wählen	0 = 1 µs/Takt; 1 = 1ms/Takt
SM67.4	SM77.4	Nicht verwendet.	
SM67.5	SM77.5	Nicht verwendet.	
SM67.6	SM77.6	PTO/PWM: Funktion wählen	0 - PTO; 1 - PWM
SM67.7	SM77.7	PTO/PWM: Freigabe	0 - PTO/PWM sperren, 1 - PTO/PWM freigeben
A0.0	A0.1	Werte für die Zykluszeit bei PTO/PWM-Ausgängen (Bereich: 2 bis 65.535)	
SM68	SM78	Höchstwertiges Byte der PTO/PWM-Zykluszeit	
SM69	SM79	Niederwertigstes Byte der PTO/PWM-Zykluszeit	
A0.0	A0.1	Werte für die Impulsdauer bei PWM-Ausgängen (Bereich: 0 bis 65.535)	
SM70	SM80	Höchstwertiges Byte der PTO/PWM-Impulsdauer	
SM71	SM81	Niederwertigstes Byte der PTO/PWM-Impulsdauer	
A0.0	A0.1	Impulswerte für Impulsausgänge (Bereich: 1 bis 4.294.967.295)	
SM72	SM82	Höchstwertiges Byte des PTO-Impulswerts	
SM73	SM83	Zweithöchstwertiges Byte des PTO-Impulswerts	
SM74	SM84	Zweitniederwertigstes Byte des PTO-Impulswerts	
SM75	SM85	Niederwertigstes Byte des PTO-Impulswerts	

Mit Hilfe der Tabelle 10-11 können Sie schnell den Wert festlegen, den Sie in dem PTO/PWM-Steuerungsregister ablegen müssen, um die gewünschte Operation aufzurufen. Verwenden Sie SMB67 für PTO/PWM0 und SMB77 für PTO/PWM1. Wenn Sie einen neuen Impulszahlwert (SMD72 bzw. SMD82), eine Impulsdauer (SMW70 bzw. SMW80) oder eine Zykluszeit (SMW68 bzw. SMW78) laden möchten, müssen Sie diese Werte genauso wie das Steuerungsregister vor Ausführung der Operation PLS laden.

Tabelle 10-11 Referenztable für PTO/PWM-Hexadezimalwerte

Steuerungsregister (Hexadezimalwert)	Ergebnis der Operation PLS					
	Freigabe	Funktion	Zeitbasis	Impulszahlwert	Impulsdauer	Zykluszeit
16#81	Ja	PTO	1 µs/Takt			Laden
16#84	Ja	PTO	1 µs/Takt	Laden		
16#85	Ja	PTO	1 µs/Takt	Laden		Laden
16#89	Ja	PTO	1 ms/Takt			Laden
16#8C	Ja	PTO	1 ms/Takt	Laden		
16#8D	Ja	PTO	1 ms/Takt	Laden		Laden
16#C1	Ja	PWM	1 µs/Takt			Laden
16#C2	Ja	PWM	1 µs/Takt		Laden	
16#C3	Ja	PWM	1 µs/Takt		Laden	Laden
16#C9	Ja	PWM	1 ms/Takt			Laden
16#CA	Ja	PWM	1 ms/Takt		Laden	
16#CB	Ja	PWM	1 ms/Takt		Laden	Laden

Initialisierung und Abläufe der Funktionen PTO/PWM

In den folgenden Abschnitten werden Vorgehensweisen zum Initialisieren und Einrichten der Funktionen PTO und PWM beschrieben. Sie verschaffen Ihnen einen guten Überblick über deren Funktionsweise. Bei den Beschreibungen wird Ausgang A0.0 verwendet. Es wird vorausgesetzt, daß die S7-200 zuvor in den Betriebszustand RUN versetzt wurde, so daß deshalb der Merker des ersten Zyklus "wahr" ist. Ist dies nicht der Fall bzw. muß die Funktion PTO/PWM neu initialisiert werden, können Sie das Initialisierungsprogramm mit einer anderen Bedingung und nicht mit dem Merker des ersten Zyklus aufrufen.

Initialisierung der Funktion PWM

Gehen Sie zum Initialisieren der Funktion PWM an Ausgang A0.0 folgendermaßen vor:

1. Setzen Sie mit dem Merker des ersten Zyklus den Ausgang auf "1" und rufen Sie das Unterprogramm auf, in dem die Initialisierung durchgeführt wird. Wenn Sie ein Unterprogramm aufrufen, rufen nachfolgende Zyklen das Unterprogramm nicht mehr auf. Dadurch verkürzt sich die Zykluszeit und das Programm ist übersichtlicher strukturiert.
2. Laden Sie im Unterprogramm für die Initialisierung den Wert 16#C3 in SM67. Hiermit geben Sie für die Funktion PWM an, daß in Mikrosekunden inkrementiert werden soll (Sie können auch den Wert 16#CB laden, wenn Sie bei der Funktion PWM in Millisekunden inkrementieren möchten). Diese Hexadezimalwerte setzen das Steuerbyte, mit dem die Operation PTO/PWM freigegeben und die Funktion PWM ausgewählt wird. Außerdem gibt das Steuerbyte an, daß in Mikro- bzw. in Millisekunden inkrementiert wird und daß die Werte für Impulsdauer und Zykluszeit aktualisiert werden sollen.
3. Laden Sie die gewünschte Zykluszeit in SM68 (Wortwert).
4. Laden Sie die gewünschte Impulsdauer in SM70 (Wortwert).
5. Führen Sie die Operation PLS aus, damit die S7-200 den PTO/PWM-Generator programmiert.
6. Laden Sie den Wert 16#C2 in SM67, damit in Mikrosekunden inkrementiert wird (oder 16#CA für Millisekunden). Dadurch wird die Angabe zum Aktualisieren der Zykluszeit im Steuerbyte zurückgesetzt und die Impulsdauer kann geändert werden. Es wird ein neuer Wert für die Impulsdauer geladen. Anschließend wird die Operation PLS ausgeführt, ohne daß das Steuerbyte ändert wird.
7. Beenden Sie das Unterprogramm.

Optionale Vorgehensweise zum synchronen Aktualisieren. Sind synchrone Aktualisierungen erforderlich, gehen Sie folgendermaßen vor:

1. Führen Sie die Operation Alle Interruptereignisse freigeben (ENI) aus.
2. Ordnen Sie mit Hilfe der Bedingung, mit der Sie die Impulsdauer aktualisieren, dem Interrupt Steigende Flanke ein Interruptprogramm zu (ATCH). Die Bedingung, die Sie zum Zuordnen des Ereignisses verwenden, darf nur für die Dauer eines Zyklus aktiv sein.
3. Erstellen Sie ein Interruptprogramm, das die Impulsdauer aktualisiert und anschließend den Interrupt der steigenden Flanke sperrt.

Hinweis

Bei der optionalen Vorgehensweise zum synchronen Aktualisieren ist es erforderlich, den PWM-Ausgang zu einem der Interrupteingänge zurückzuleiten.

Ändern der Impulsdauer für PWM-Ausgänge

Zum Ändern der Impulsdauer von PWM-Ausgängen in einem Unterprogramm gehen Sie folgendermaßen vor:

1. Rufen Sie ein Unterprogramm auf, um die gewünschte Impulsdauer in SM70 (Wortwert) zu laden.
2. Führen Sie die Operation PLS aus, damit die S7-200 den PTO/PWM-Generator programmiert.
3. Beenden Sie das Unterprogramm.

Initialisierung der Funktion PTO

Gehen Sie zum Initialisieren der Funktion PTO folgendermaßen vor:

1. Setzen Sie mit dem Merker des ersten Zyklus den Ausgang auf "0" und rufen Sie das Unterprogramm auf, in dem die Initialisierung durchgeführt wird. Wenn Sie ein Unterprogramm aufrufen, rufen nachfolgende Zyklen das Unterprogramm nicht mehr auf. Dadurch verkürzt sich die Zykluszeit und das Programm ist übersichtlicher strukturiert.
2. Laden Sie im Unterprogramm für die Initialisierung den Wert 16#85 in SM67. Hiermit geben Sie für die Funktion PTO an, daß in Mikrosekunden inkrementiert werden soll (Sie können auch den Wert 16#8D laden, wenn Sie bei der Funktion PTO in Millisekunden inkrementieren möchten). Diese Hexadezimalwerte setzen das Steuerbyte, mit dem die Operation PTO/PWM freigegeben und die Funktion PTO ausgewählt wird. Außerdem gibt das Steuerbyte an, daß in Mikro- bzw. in Millisekunden inkrementiert wird und daß die Werte für Impulsdauer und Zykluszeit aktualisiert werden sollen.
3. Laden Sie die gewünschte Zykluszeit in SM68 (Wortwert).
4. Laden Sie die gewünschte Impulsdauer in SM72 (Wortwert).
5. Dieser Schritt ist optional: Möchten Sie nach Ausführung der Operation Impulsfolge eine zugeordnete Operation ausführen, können Sie einen Interrupt programmieren, indem Sie das Ereignis Impulsfolge beendet (Interruptklasse 19) einem Interruptprogramm zuordnen und die Operation Alle Interruptereignisse freigegeben ausführen. Ausführliche Informationen zur Bearbeitung von Interrupts entnehmen Sie Abschnitt 10.14.
6. Führen Sie PLS aus, damit die S7-200 den PTO/PWM-Generator programmiert.
7. Beenden Sie das Unterprogramm.

Ändern der Zykluszeit der Funktion PTO

Zum Ändern der Zykluszeit in einem Interrupt- oder Unterprogramm gehen Sie folgendermaßen vor:

1. Laden Sie den Wert 16#81 in SM67. Hiermit geben Sie für die Funktion PTO an, daß in Mikrosekunden inkrementiert werden soll (Sie können auch den Wert 16#89 laden, wenn Sie in Millisekunden inkrementieren möchten). Diese Hexadezimalwerte setzen das Steuerbyte, mit dem die Operation PTO/PWM freigegeben und die Funktion PTO ausgewählt wird. Außerdem gibt das Steuerbyte an, daß in Mikro- bzw. in Millisekunden inkrementiert wird und daß der Wert für die Zykluszeit aktualisiert werden soll.
2. Laden Sie die gewünschte Zykluszeit in SM68 (Wortwert).
3. Führen Sie die Operation PLS aus, damit die S7-200 den PTO/PWM-Generator programmiert.
4. Beenden Sie das Interrupt- bzw. das Unterprogramm. (Unterprogramme können nicht aus Interruptprogrammen aufgerufen werden.)

Ändern des Impulswerts der Funktion PTO

Zum Ändern des Impulswerts in einem Interrupt- oder Unterprogramm gehen Sie folgendermaßen vor:

1. Laden Sie den Wert 16#84 in SM67. Hiermit geben Sie für die Funktion PTO an, daß in Mikrosekunden inkrementiert werden soll (Sie können auch den Wert 16#8C laden, wenn Sie in Millisekunden inkrementieren möchten). Diese Hexadezimalwerte setzen das Steuerbyte, mit dem die Operation PTO/PWM freigegeben und die Funktion PTO ausgewählt wird. Außerdem gibt das Steuerbyte an, daß in Mikro- bzw. in Millisekunden inkrementiert wird und daß der Wert für den Zählwert aktualisiert werden soll.
2. Laden Sie die gewünschte Impulsdauer in SM72 (Wortwert).
3. Führen Sie die Operation PLS aus, damit die S7-200 den PTO/PWM-Generator programmiert.
4. Beenden Sie das Interrupt- bzw. das Unterprogramm. (Unterprogramme können nicht aus Interruptprogrammen aufgerufen werden.)

Ändern der Zykluszeit und des Impulszählwerts der Funktion PTO

Zum Ändern der Zykluszeit und des Zählwerts in einem Interrupt- oder Unterprogramm gehen Sie folgendermaßen vor:

1. Laden Sie den Wert 16#85 in SM67. Hiermit geben Sie für die Funktion PTO an, daß in Mikrosekunden inkrementiert werden soll (Sie können auch den Wert 16#8D laden, wenn Sie in Millisekunden inkrementieren möchten). Diese Hexadezimalwerte setzen das Steuerbyte, mit dem die Operation PTO/PWM freigegeben und die Funktion PTO ausgewählt wird. Außerdem gibt das Steuerbyte an, daß in Mikro- bzw. in Millisekunden inkrementiert wird und daß die Werte für Zykluszeit und Zählwert aktualisiert werden sollen.
2. Laden Sie die gewünschte Zykluszeit in SM68 (Wortwert).
3. Laden Sie die gewünschte Impulsdauer in SM72 (Wortwert).
4. Führen Sie die Operation PLS aus, damit die S7-200 den PTO/PWM-Generator programmiert.
5. Beenden Sie das Interrupt- bzw. das Unterprogramm. (Unterprogramme können nicht aus Interruptprogrammen aufgerufen werden.)

Aktive Funktionen PTO/PWM

Ist eine der Funktionen PTO oder PWM an A0.0 oder A0.1 aktiv, dann ist der jeweilige Ausgang gesperrt. Weder die Werte, die für den Ausgang im Prozeßabbild abgelegt sind, noch geforcte Werte werden in den Ausgang übertragen, solange eine der Funktionen PTO oder PWM aktiv ist. Die Operation PTO gilt als aktiv, solange sie aktiviert und noch nicht beendet ist. Direkte Ausgangsoperationen, die in diese Ausgänge schreiben, während die Funktion PTO bzw. PWM aktiv ist, verursachen keine Verzerrung der Wellenform bei den Funktionen PTO bzw. PWM.

Hinweis

Wird die Funktion PTO vor der vollständigen Bearbeitung deaktiviert, wird die aktuelle Impulsausgabe beendet und der Ausgang A0.0 bzw. A0.1 kehrt wieder zur üblichen Steuerung des Prozeßabbilds zurück. Durch das Reaktivieren der Funktion PTO beginnt die Impulsfolge erneut und verwendet dabei die zuletzt geladenen Angaben für die Impulsausgabe.

Auswirkungen auf die Ausgänge

Die Funktion PTO/PWM und das Prozeßabbild verwenden die Ausgänge A0.0 und A0.1 gemeinsam. Die Anfangs- und Endzustände der Wellenformen der Funktionen PTO und PWM werden von dem Wert des entsprechenden Bits im Prozeßabbild beeinflusst. Wird eine Impulsfolge an A0.0 oder an A0.1 ausgegeben, dann legt das Prozeßabbild den Anfangs- und Endzustand des Ausganges fest und bewirkt dadurch, daß die Impulsfolge hoch oder niedrig beginnt.

Bei den Übergängen der PTO-Pipeline und denen der PWM-Impulsdauer werden die Funktionen PTO und PWM kurzzeitig gesperrt. Dadurch kann es an den Übergängen zu einer kurzen Unterbrechung in den Wellenformen der Ausgänge kommen. Um nachteilige Auswirkungen dieser Unterbrechungen zu minimieren, setzen Sie das Bit im Prozeßabbild für die Funktion PTO auf "0" und für die Funktion PWM auf "1". Bild 10-17 zeigt die daraus entstehenden Wellenformen der Funktionen PTO und PWM. Beachten Sie, daß beim Übergang der Funktion PTO der letzte halbe Zyklus auf eine Impulsdauer von ca. 120 µs verkürzt wird. Wird bei der Funktion PWM die optionale Vorgehensweise zum synchronen Aktualisieren eingesetzt, dann wird der erste hohe Zeitimpuls nach dem Übergang um ca. 120 µs verlängert.

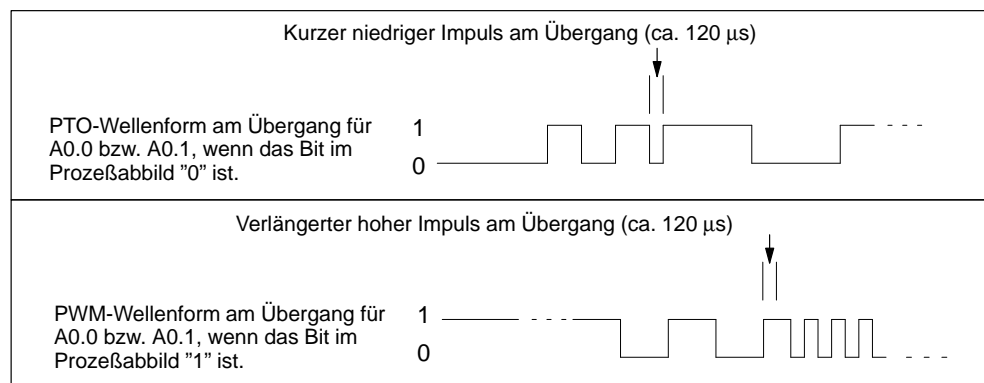


Bild 10-17 Beispiel für die Form von Impulsfolgen an A0.0 bzw. A0.1

Beispiel für eine Impulsfolge

KOP		AWL
Netzwerk 1		Network 1 LD SM0.1 R A0.0, 1 CALL 0
Netzwerk 2	(END)	Network 2 MEND
Netzwerk 3	0	Network 3 SBR 0
Netzwerk 4		Network 4 LD SM0.0 MOVB 16#8D, SMB67 MOVW 500, SMW68 MOVD 4, SMD72 ATCH 3, 19 ENI PLS 0
Netzwerk 5	(RET)	Network 5 RET

Bild 10-18 Beispiel für eine Impulsfolge

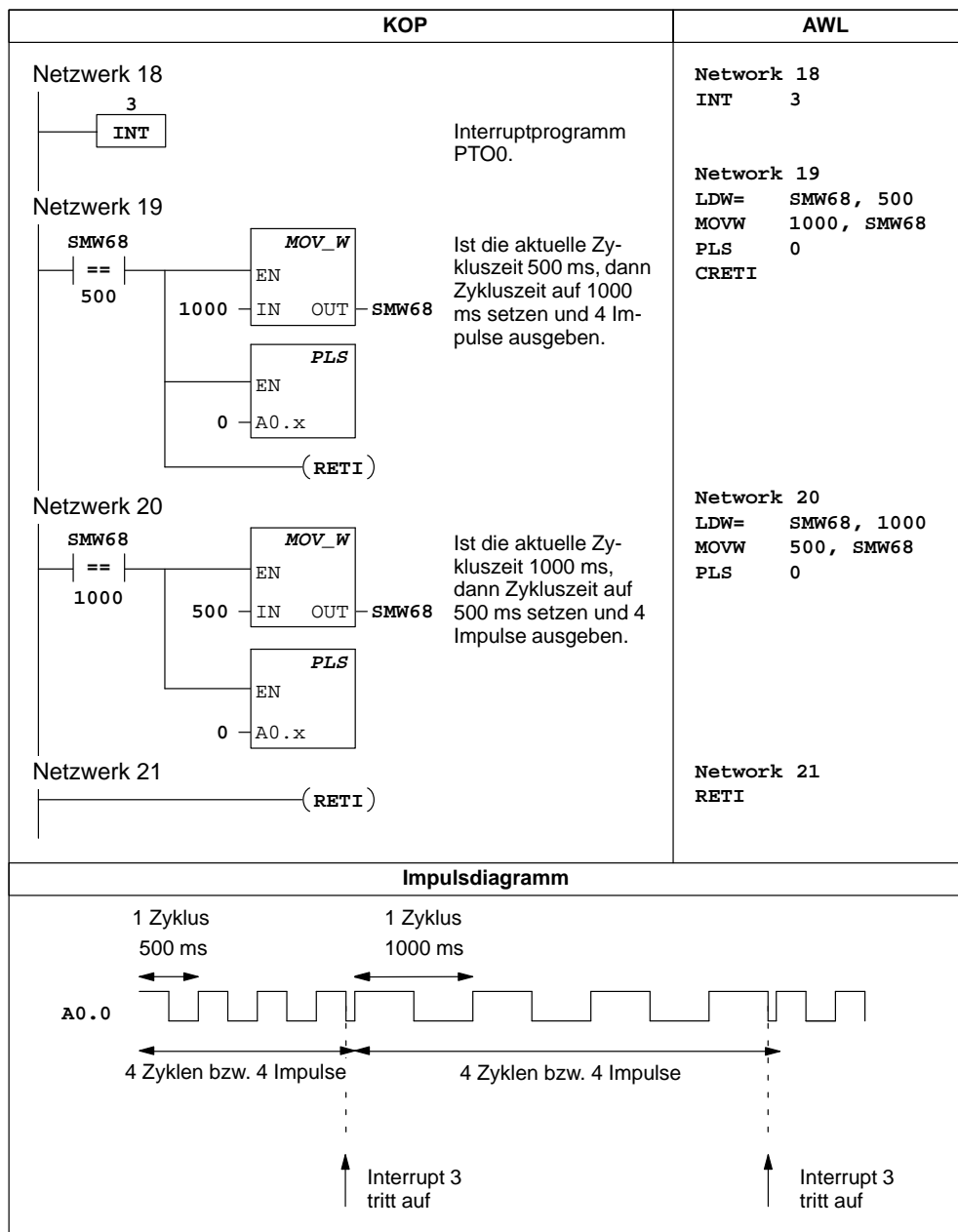


Bild 10-18 Beispiel für eine Impulsfolge, Fortsetzung

Beispiel für Impulsdauermodulation

Bild 10-19 zeigt ein Beispiel für Impulsdauermodulation. Soll die Impulsdauer geändert werden, muß die Funktion PWM kurzzeitig zum Aktualisieren gesperrt werden. Dies geschieht asynchron zum PWM-Zyklus und könnte ein unerwünschtes Impulzzittern in dem gesteuerten Gerät verursachen. Muß die Impulsdauer synchron aktualisiert werden, wird der Impulsausgang zum Interrupteingang (E0.0) zurückgeleitet. Soll die Impulsdauer geändert werden, wird der Interrupteingang freigegeben, so daß bei der nächsten steigenden Flanke an E0.0 die Impulsdauer synchron zum PWM-Zyklus geändert wird.

Die Impulsdauer wird im Interruptprogramm geändert und das Interruptereignis vom Programm getrennt bzw. gesperrt. Dadurch treten außer beim Ändern der Impulsdauer keine Interrupts auf.

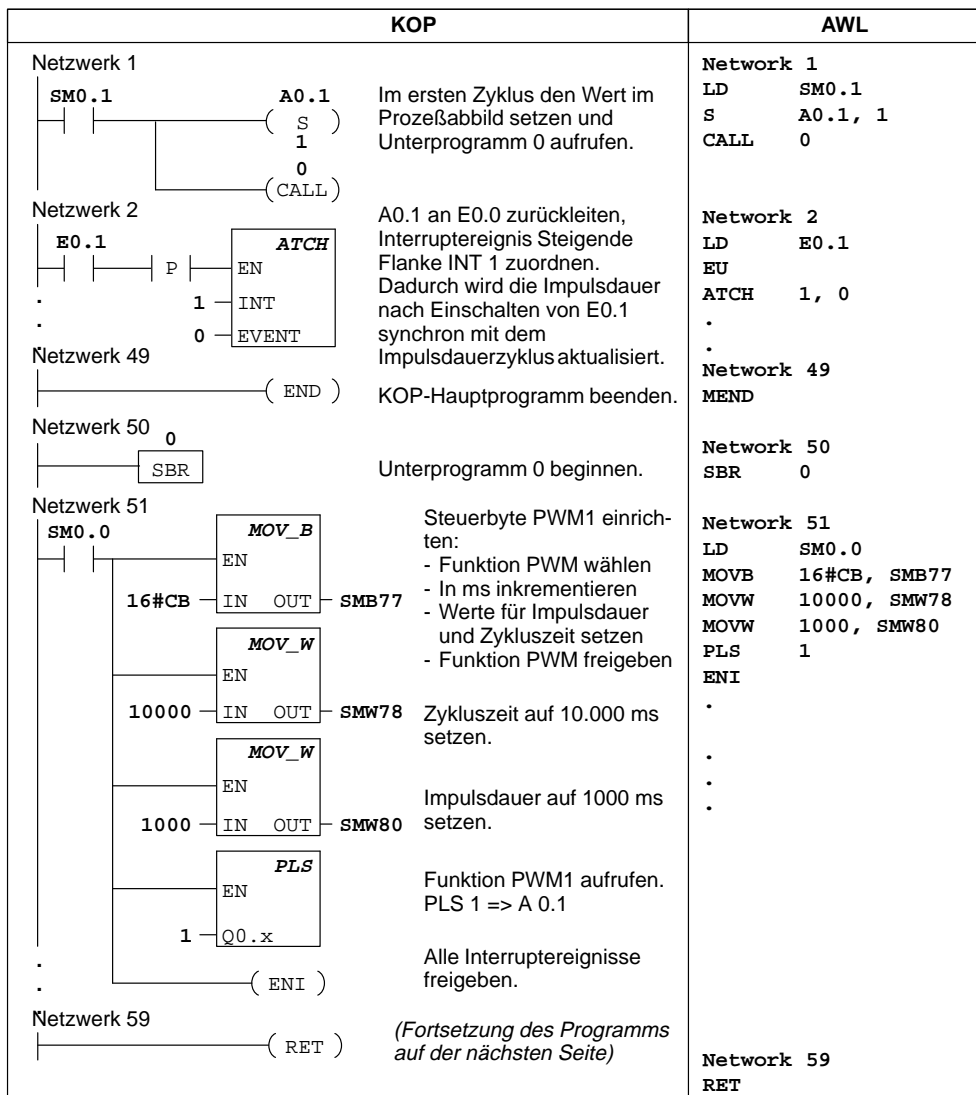


Bild 10-19 Beispiel für einen schnellen Ausgang mit Impulsdauermodulation

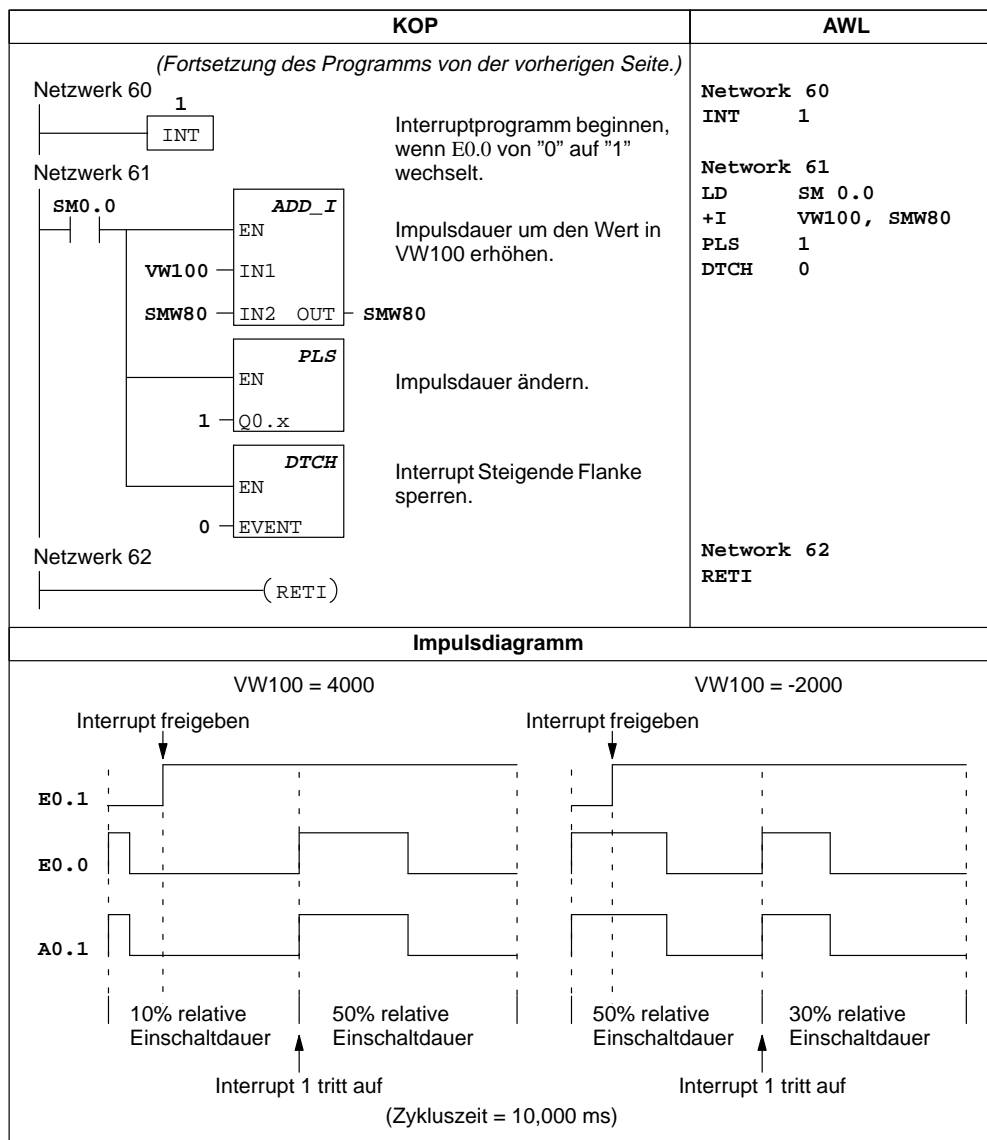
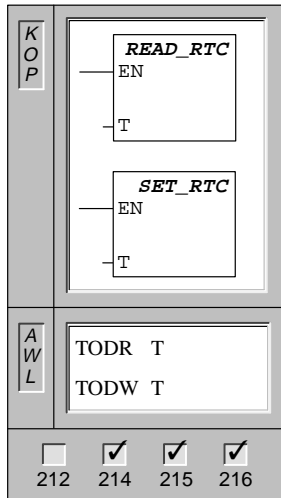


Bild 10-19 Beispiel für einen schnellen Ausgang mit Impulsdauermodulation, Fortsetzung

Echtzeituhr lesen und Echtzeituhr schreiben



Die Operation **Echtzeituhr lesen** liest die aktuelle Uhrzeit und das aktuelle Datum aus der Echtzeituhr der CPU und lädt beides in einen 8-Byte-Puffer (mit Beginn an Adresse T).

Die Operation **Echtzeituhr schreiben** schreibt die aktuelle Uhrzeit und das aktuelle Datum, die beide in einen 8-Byte-Puffer (mit Beginn an Adresse T) geladen sind, in die Echtzeituhr.

In AWL werden die Operationen Read_RTC und Set_RTC durch die Mnemonik TODR (Echtzeituhr lesen) und TODW (Echtzeituhr schreiben) dargestellt.

Operanden: T: VB, EB, AB, MB, SMB, *VD, *AC, SB

Die Echtzeituhr startet nach längerem Stromausfall oder nach Speicherverlust mit folgendem Datum und folgender Zeit:

Datum: 01-Jan-90
 Zeit: 00:00:00
 Wochentag: Sonntag

Die Echtzeituhr der S7-200 verwendet die beiden niederwertigsten Ziffern für die Jahresangabe. Deshalb wird das Jahr 2000 als das Jahr 00 dargestellt (auf 99 folgt 00).

Sie müssen alle Datum- und Zeitwerte im BCD-Format codieren (z.B. 16#97 für das Jahr 1997). Verwenden Sie hierzu die folgenden Datenformate:

Jahr/Monat	jjmm	jj -	0 bis 99	mm -	1 bis 12
Tag/Stunde	tthh	tt -	1 bis 31	hh -	0 bis 23
Minute/Sekunde	mmss	mm -	0 bis 59	ss -	0 bis 59
Wochentag	000t	t -	0 bis 7	1 =	Sonntag
				0 =	Wochentag wird ausgeschaltet (bleibt 0)

Hinweis

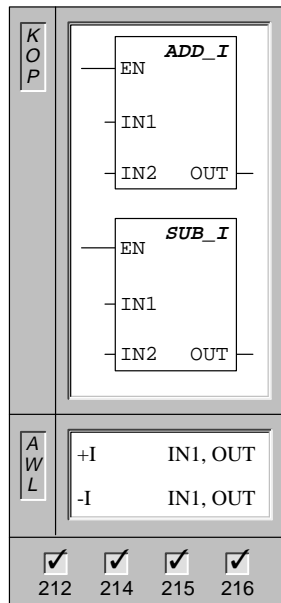
Die S7-200 CPU prüft nicht, ob der Wochentag mit dem Datum übereinstimmt. Dadurch kann es zu unzulässigen Daten wie 30. Februar kommen. Sie sollten daher immer sicherstellen, daß Sie das Datum korrekt eingegeben haben.

Verwenden Sie die Operationen TODR und TODW nie sowohl im Hauptprogramm als auch in einem Interruptprogramm. Soll eine der Operationen TODR/TODW in einem Interruptprogramm ausgeführt werden, während eine andere Operation TODR bzw. TODW bearbeitet wird, kann die Operation im Interruptprogramm nicht ausgeführt werden. SM4.5 wird gesetzt und zeigt an, daß zwei Operationen gleichzeitig versucht haben, auf die Uhr zuzugreifen.

Das Automatisierungssystem S7-200 benötigt die Jahresinformationen nicht und wird deshalb durch den Wechsel ins nächste Jahrtausend (2000) nicht beeinflusst. Anwenderprogramme jedoch, deren arithmetische Operationen oder Vergleichsoperationen den Jahreswert einsetzen, müssen die zweistellige Darstellung und den Jahrtausendwechsel beachten.

10.6 Arithmetische Operationen und Operationen für den PID-Regler

Ganze Zahlen (16 Bit) addieren und subtrahieren



Die Operationen **Ganze Zahlen (16 Bit) addieren** und **Ganze Zahlen (16 Bit) subtrahieren** addieren bzw. subtrahieren zwei ganze Zahlen (16 Bit) und liefern ein Ergebnis (16 Bit) in OUT.

Operanden: IN1, IN2: VW, T, Z, EW, AW, MW, SMW, AC, AEW, Konstante, *VD, *AC, SW

OUT: VW, T, Z, EW, AW, MW, SMW, AC, *VD, *AC, SW

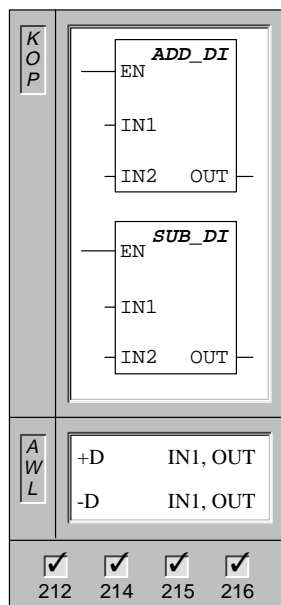
In KOP: $IN1 + IN2 = OUT$
 $IN1 - IN2 = OUT$

In AWL: $IN1 + OUT = OUT$
 $OUT - IN1 = OUT$

Hinweis: Wenn Sie in KOP programmieren, können Sie angeben, daß IN1 gleich OUT ist. Auf diese Weise sparen Sie Speicherplatz.

Diese Operationen beeinflussen die folgenden Sondermerker: SM1.0 (Null); SM1.1 (Überlauf); SM1.2 (negativ)

Ganze Zahlen (32 Bit) addieren und subtrahieren



Die Operationen **Ganze Zahlen (32 Bit) addieren** und **Ganze Zahlen (32 Bit) subtrahieren** addieren bzw. subtrahieren zwei ganze Zahlen (32 Bit) und liefern ein Ergebnis (32 Bit) in OUT.

Operanden: IN1, IN2: VD, ED, AD, MD, SMD, AC, HC, Konstante, *VD, *AC, SD

OUT: VD, ED, AD, MD, SMD, AC, *VD, *AC, SD

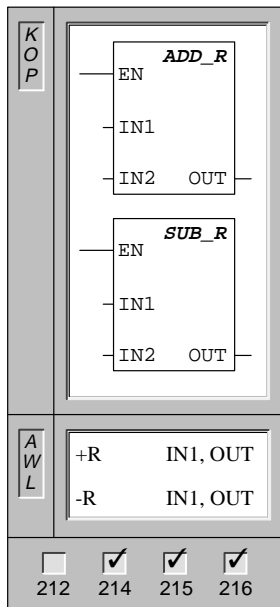
In KOP: $IN1 + IN2 = OUT$
 $IN1 - IN2 = OUT$

In AWL: $IN1 + OUT = OUT$
 $OUT - IN1 = OUT$

Hinweis: Wenn Sie in KOP programmieren, können Sie angeben, daß IN1 gleich OUT ist. Auf diese Weise sparen Sie Speicherplatz.

Diese Operationen beeinflussen die folgenden Sondermerker: SM1.0 (Null); SM1.1 (Überlauf); SM1.2 (negativ)

Realzahlen addieren und subtrahieren



Die Operationen **Realzahlen addieren** und **Realzahlen subtrahieren** addieren bzw. subtrahieren zwei ganze Zahlen (32 Bit) und liefern eine Realzahl als Ergebnis (OUT).

Operanden: IN1, IN2: VD, ED, AD, MD, SMD, AC, Konstante, *VD, *AC, SD

OUT: VD, ED, AD, MD, SMD, AC, *VD, *AC SD

In KOP: $IN1 + IN2 = OUT$
 $IN1 - IN2 = OUT$

In AWL: $IN1 + OUT = OUT$
 $OUT - IN1 = OUT$

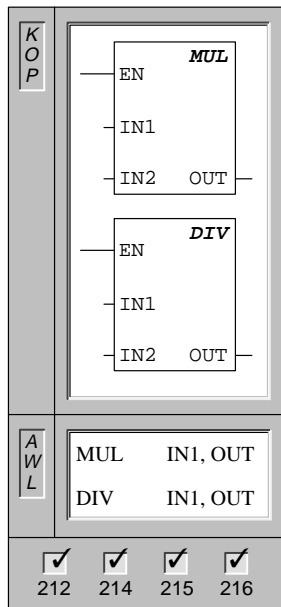
Hinweis: Wenn Sie in KOP programmieren, können Sie angeben, daß IN1 gleich OUT ist. Auf diese Weise sparen Sie Speicherplatz.

Diese Operationen beeinflussen die folgenden Sondermerker: SM1.0 (Null); SM1.1 (Überlauf/unzulässiger Wert); SM1.2 (negativ)

Hinweis

Realzahlen bzw. Gleitpunktzahlen werden in dem Format dargestellt, das in der Richtlinie ANSI/IEEE 754-1985 (einfachgenau) beschrieben ist. Ausführliche Informationen zu diesen Zahlen entnehmen Sie den Richtlinien.

Ganze Zahlen multiplizieren und dividieren



Die Operation **Ganze Zahlen (16 Bit) multiplizieren** multipliziert zwei ganze Zahlen (16 Bit) und liefert ein Ergebnis (32 Bit) in OUT.

In AWL wird von dem Wert (32 Bit) in OUT das niederwertigste Wort (16 Bit) als einer der Faktoren verwendet.

Die Operation **Ganze Zahlen (16 Bit) dividieren** dividiert zwei ganze Zahlen (16 Bit) und liefert ein Ergebnis (32 Bit) in OUT. Das Ergebnis (32 Bit) in OUT besteht aus einem Quotienten (16 niederwertigste Bits) und einem Divisionsrest (16 höchstwertige Bits).

In AWL wird von dem Wert (32 Bit) in OUT das niederwertigste Wort (16 Bit) als Dividend verwendet.

Operanden: IN1, IN2: VW, T, Z, EW, AW, MW, SMW, AC, AEW, Konstante, *VD, *AC, SW

OUT: VD, ED, AD, MD, SMD, AC, *VD, *AC, SD

In KOP: $IN1 * IN2 = OUT$
 $IN1 / IN2 = OUT$

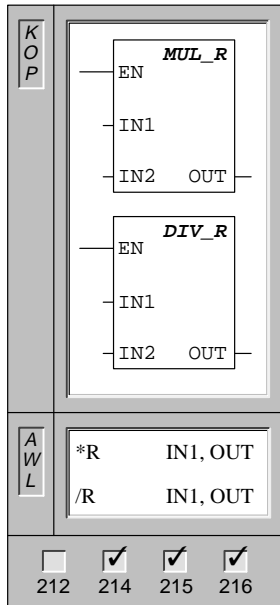
In AWL: $IN1 * OUT = OUT$
 $OUT / IN1 = OUT$

Hinweis: Wenn Sie in KOP programmieren, können Sie angeben, daß IN1 gleich OUT ist. Auf diese Weise sparen Sie Speicherplatz.

Diese Operationen beeinflussen die folgenden Sondermerker:

SM1.0 (Null); SM1.1 (Überlauf); SM1.2 (negativ); SM1.3 (Division durch Null)

Realzahlen multiplizieren und dividieren



Die Operation **Realzahlen multiplizieren** multipliziert zwei Realzahlen (32 Bit) und liefert ein Ergebnis (32 Bit) in OUT.

Die Operation **Realzahlen dividieren** dividiert zwei Realzahlen (32 Bit) und liefert ein Ergebnis (32 Bit) in OUT.

Operanden: IN1, IN2: VD, ED, AD, MD, SMD, AC, Konstante, *VD, *AC, SD

OUT: VD, ED, AD, MD, SMD, AC, *VD, *AC, SD

In KOP: $IN1 * IN2 = OUT$
 $IN1 / IN2 = OUT$

In AWL: $IN1 * OUT = OUT$
 $OUT / IN1 = OUT$

Hinweis: Wenn Sie in KOP programmieren, können Sie angeben, daß IN1 gleich OUT ist. Auf diese Weise sparen Sie Speicherplatz.

Diese Operationen beeinflussen die folgenden Sondermerker:

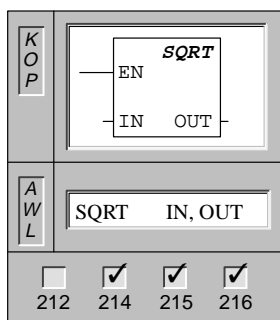
SM1.0 (Null); SM1.1 (Überlauf); SM1.2 (negativ); SM1.3 (Division durch Null)

Werden SM1.1 oder SM1.3 gesetzt, dann werden die anderen Statusbits für arithmetische Operationen und die ursprünglichen Eingangsoperanden nicht geändert.

Hinweis

Realzahlen bzw. Gleitpunktzahlen werden in dem Format dargestellt, das in der Richtlinie ANSI/IEEE 754-1985 (einfachgenau) beschrieben ist. Ausführliche Informationen zu diesen Zahlen entnehmen Sie den Richtlinien.

Quadratwurzel einer Realzahl ziehen



Die Operation **Quadratwurzel einer Realzahl ziehen** zieht die Quadratwurzel der Realzahl (32 Bit) in IN. Das Ergebnis (OUT) ist ebenfalls eine Realzahl (32 Bit).

$\sqrt{IN} = OUT$

Operanden: IN: VD, ED, AD, MD, SMD, AC, Konstante, *VD, *AC, SD

OUT: VD, ED, AD, MD, SMD, AC, *VD, *AC, SD

Diese Operationen beeinflussen die folgenden Sondermerker:

SM1.0 (Null); SM1.1 (Überlauf); SM1.2 (negativ)

Beispiele für arithmetische Operationen

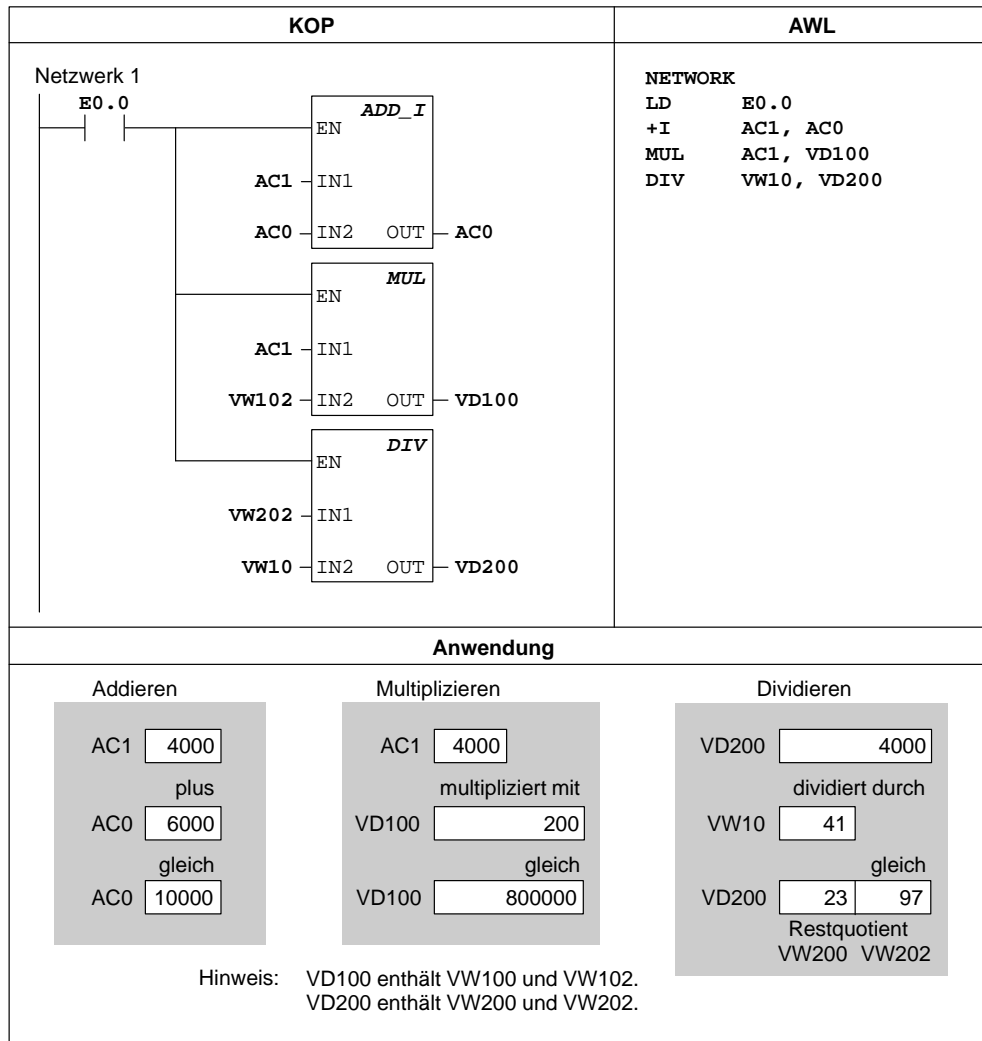
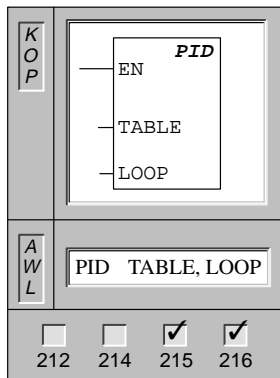


Bild 10-20 Beispiele für arithmetische Operationen in KOP und AWL

PID-Regler



Die Operation **PID-Regler** berechnet die PID-Regelkreise für den angegebenen Regelkreis LOOP mit Hilfe der Informationen zu Eingabewerten und Konfiguration im Parameter TABLE.

Operanden: TABLE: VB
 LOOP: 0 bis 7

Diese Operationen beeinflussen die folgenden Sondermerker:
 SM1.1 (Überlauf)

Die Operation PID-Regler (Proportional-, Integral-, Differentialregler) dient zum Berechnen des PID-Reglers. Der oberste Stackwert muß aktiviert sein (Signalfluß vorhanden), damit die PID-Berechnungen durchgeführt werden können. Die Operation verfügt über zwei Parameter: TABLE enthält die Anfangsadresse der Tabelle für den Regelkreis und LOOP enthält die Nummer des Regelkreises, wobei es sich um eine Konstante von 0 bis 7 handeln kann. In einem Programm dürfen Sie maximal acht Operationen PID verwenden. Geben zwei PID-Operationen die gleiche Nummer für den Regelkreis an (auch wenn sie verschiedene Adressen für die Tabelle verwenden), wirken sich die Berechnungen für den PID-Regler aufeinander aus und das Ergebnis ist nicht mehr vorhersehbar.

Die Tabelle für den Regelkreis speichert neun Parameter, mit denen die Funktion des Regelkreises überwacht und gesteuert werden kann. Diese Parameter umfassen den aktuellen und den vorherigen Wert der Prozeßvariablen (des Istwerts), den Sollwert, die Stellgröße, die Verstärkung, die Abtastzeit, die Integralzeit (Rücksetzen), die Differentialzeit und die Integralsumme (Bias).

Damit die PID-Berechnung mit der gewünschten Abtastzeit durchgeführt werden kann, muß die Operation PID entweder in einem Interruptprogramm für einen zeitgesteuerten Interrupt oder in einem Hauptprogramm bei zeitgesteuerter Geschwindigkeit ausgeführt werden. Die Abtastgeschwindigkeit muß als Eingang der Operation PID über die Tabelle für den Regelkreis bereitgestellt werden.

PID-Algorithmus

In stetig wirkenden Regeleinrichtungen regelt ein PID-Regler die Stellgröße, um die Regeldifferenz (e) auf Null zu bringen. Die Regeldifferenz ist der Unterschied zwischen Sollwert und Prozeßvariable (Istwert). Das Prinzip des PID-Reglers basiert auf der folgenden Gleichung, die die Stellgröße M(t) als Ergebnis eines Proportionalanteils, eines Integralanteils und eines Differentialanteils darstellt:

$$M(t) = K_C * e + K_C \int_0^t e dt + M_{initial} + K_C * de/dt$$

Stellgröße = Proportionalanteil + Integralanteil + Differentialanteil

Erklärung:

- M(t) Stellgröße in Abhängigkeit von der Zeit
- K_C Verstärkung
- e Regeldifferenz (Differenz zwischen Sollwert und Istwert)
- M_{initial} Anfangswert der Stellgröße

Damit die Regeleinrichtung in einen Digitalrechner implementiert werden kann, muß die kontinuierlich arbeitende Funktion in regelmäßiges Abtasten der Regeldifferenz mit anschließender Berechnung der Stellgröße umgesetzt werden. Die folgende Gleichung gilt als Basis für die Umsetzung einer Regeleinrichtung mit einem Digitalrechner:

M_n	=	$K_C * e_n$	+	$K_I * \sum_1^n$	+	$M_{initial}$	+	$K_D * (e_n - e_{n-1})$
Stellgröße	=	Proportionalanteil	+	Integralanteil	+	Differentialanteil		

Erklärung:

- M_n errechnete Stellgröße bei Abtastzeit n
- K_C Verstärkung
- e_n Wert der Regeldifferenz bei Abtastzeit n
- e_{n-1} vorheriger Wert der Regeldifferenz (bei Abtastzeit n - 1)
- K_I proportionale Konstante des Integralanteils
- $M_{initial}$ Anfangswert der Stellgröße
- K_D proportionale Konstante des Differentialanteils

In dieser Gleichung wird deutlich, daß der Integralanteil das Ergebnis aller Regeldifferenzen vom ersten Abtasten bis zum aktuellen Abtasten darstellt. Der Differentialanteil ist das Ergebnis des aktuellen Ab tastens und des vorherigen Ab tastens, während der Proportionalanteil nur das Ergebnis des aktuellen Ab tastens ist. Es ist weder sinnvoll noch nützlich, in einem Digitalrechner alle Regeldifferenzen zu speichern.

Ein Digitalrechner berechnet die Stellgröße jedesmal, wenn die Regeldifferenz abgetastet wird. Diese Berechnungen beginnen beim ersten Abtasten. Deshalb müssen nur der vorherige Wert der Regeldifferenz und der vorherige Wert des Integralanteils gespeichert werden. Da die Funktionen bei Regeleinrichtungen am Digitalrechner ständig wiederholt werden, läßt sich die bei jedem Abtasten auszuführende Gleichung vereinfachen. Im folgenden wird die vereinfachte Gleichung dargestellt:

M_n	=	$K_C * e_n$	+	$K_I * e_n + MX$	+	$K_D * (e_n - e_{n-1})$
Stellgröße	=	Proportionalanteil	+	Integralanteil	+	Differentialanteil

Erklärung:

- M_n errechnete Stellgröße bei Abtastzeit n
- K_C Verstärkung
- e_n Wert der Regeldifferenz bei Abtastzeit n
- e_{n-1} vorheriger Wert der Regeldifferenz (bei Abtastzeit n - 1)
- K_I proportionale Konstante des Integralanteils
- MX vorheriger Wert des Integralanteils (bei Abtastzeit n - 1)
- K_D proportionale Konstante des Differentialanteils

Die CPU verwendet eine abgewandelte Form der oben dargestellten vereinfachten Gleichung zum Berechnen der Stellgröße in einem Regelkreis. Im folgenden wird die abgewandelte Gleichung dargestellt:

M_n	=	MP_n	+	MI_n	+	MD_n
Stellgröße	=	Proportionalanteil	+	Integralanteil	+	Differentialanteil

Erklärung:

- M_n errechnete Stellgröße bei Abtastzeit n
- MP_n Wert des Proportionalanteils der Stellgröße bei Abtastzeit n
- MI_n Wert des Integralanteils der Stellgröße bei Abtastzeit n
- MD_n Wert des Differentialanteils der Stellgröße bei Abtastzeit n

Der Proportionalanteil

Der Proportionalanteil MP ist das Produkt der Verstärkung (K_C), die die Genauigkeit bei der Berechnung der Stellgröße und bei der Regeldifferenz (e) angibt. Die Regeldifferenz ist die Differenz zwischen dem Sollwert (SW) und dem Istwert (IW), der Prozeßvariablen, bei einer angegebenen Abtastzeit. Die von der CPU verwendete Gleichung für den Proportionalanteil lautet wie folgt:

$$MP_n = K_C * (SW_n - IW_n)$$

Erklärung:

MP_n	Wert des Proportionalanteils der Stellgröße bei Abtastzeit n
K_C	Verstärkung
SW_n	Sollwert bei Abtastzeit n
IW_n	Istwert (Wert der Prozeßvariablen) bei Abtastzeit n

Der Integralanteil

Der Integralanteil MI ist proportional zu der Summe der Regeldifferenz über der Zeit. Die von der CPU verwendete Gleichung für den Integralanteil lautet wie folgt:

$$MI_n = K_C * T_S / T_I * (SW_n - IW_n) + MX$$

Erklärung:

MI_n	Wert des Integralanteils der Stellgröße bei Abtastzeit n
K_C	Verstärkung
T_S	Abtastzeit im Regelkreis
T_I	Differentialzeit des Regelkreises (wird auch Vorhaltezeit genannt)
SW_n	Sollwert bei Abtastzeit n
IW_n	Istwert (Wert der Prozeßvariablen) bei Abtastzeit n
MX	Wert des Integralanteils bei Abtastzeit n - 1 (auch Integralsumme oder Bias genannt)

Die Integralsumme oder Bias (MX) ist die laufende Summe aller vorherigen Werte des Integralanteils. Nach jeder Berechnung von MI_n wird die Integralsumme mit dem Wert von MI_n aktualisiert. Hierbei kann es sich um eine Anpassung oder eine Begrenzung handeln (ausführliche Informationen hierzu entnehmen Sie dem Abschnitt "Variablen und Bereiche"). Der Anfangswert der Integralsumme wird typischerweise kurz vor der ersten Berechnung der Stellgröße für den Regelkreis auf den Wert der Stellgröße ($M_{initial}$) gesetzt. Der Integralanteil enthält verschiedene Konstanten: die Verstärkung (K_C), die Abtastzeit (T_S) und die Integralzeit (T_I). Die Abtastzeit ist die Zykluszeit, bei der der PID-Regler die Stellgröße neu berechnet. Die Integralzeit ist die Zeit, mit der der Einfluß des Integralanteils bei der Berechnung der Stellgröße gesteuert wird.

Der Differentialanteil

Der Differentialanteil MD ist proportional zu der Änderung der Regeldifferenz. Die Gleichung für den Differentialanteil lautet wie folgt:

$$MD_n = K_C * T_D / T_S * ((SW_n - IW_n) - (SW_{n-1} - IW_{n-1}))$$

Damit bei Änderungen des Sollwerts Schrittländerungen oder Sprünge in der Stellgröße aufgrund des Differentialverhaltens vermieden werden, wird für diese Gleichung angenommen, daß der Sollwert eine Konstante ist ($SW_n = SW_{n-1}$). Deshalb wird die Änderung des Istwerts (der Prozeßvariablen) und nicht die Änderung der Regeldifferenz berechnet. Dies zeigt folgende Gleichung:

$$MD_n = K_C * T_D / T_S * (SW_n - IW_n - SW_n + IW_{n-1})$$

oder auch:

$$MD_n = K_C * T_D / T_S * (IW_{n-1} - IW_n)$$

Erklärung:

MD_n	Wert des Differentialanteils der Stellgröße bei Abtastzeit n
K_C	Verstärkung
T_S	Abtastzeit im Regelkreis
T_D	Differentialzeit des Regelkreises (wird auch Vorhaltezeit genannt)
SW_n	Sollwert bei Abtastzeit n
SW_{n-1}	Sollwert bei Abtastzeit n - 1
IW_n	Istwert (Wert der Prozeßvariablen) bei Abtastzeit n
IW_{n-1}	Istwert (Wert der Prozeßvariablen) bei Abtastzeit n - 1

Für die Berechnung des nächsten Differentialanteils muß der Istwert und nicht die Regeldifferenz gespeichert werden. Beim ersten Abtasten wird der Wert von IW_{n-1} mit dem Wert von IW_n initialisiert.

Auswählen eines Reglers

In vielen Regeleinrichtungen sind häufig nur ein oder zwei verschiedene Regler erforderlich. Es kann beispielsweise nur ein Proportionalregler oder es können ein Proportional- und ein Integralregler eingesetzt werden. Sie können die erforderlichen Regler auswählen, indem Sie den konstanten Parameter auf einen bestimmten Wert setzen.

Benötigen Sie kein Integralverhalten (keinen I-Anteil in der PID-Berechnung), dann müssen Sie einen unendlichen Wert für die Integralzeit angeben. Auch ohne Integralanteil darf der Wert für den Integralanteil wegen des Anfangswerts der Integralsumme MX nicht Null sein.

Benötigen Sie kein Differentialverhalten (keinen D-Anteil in der PID-Berechnung), dann müssen Sie für die Differentialzeit den Wert 0,0 angeben.

Benötigen Sie kein Proportionalverhalten (keinen P-Anteil in der PID-Berechnung), sondern nur I- oder ID-Regler, dann müssen Sie für die Verstärkung den Wert 0,0 angeben. Die Verstärkung im Regelkreis ist ein Faktor in den Gleichungen zum Berechnen des Integral- und des Differentialanteils. Wenn Sie also für die Verstärkung den Wert 0,0 angeben, dann resultiert daraus, daß für die Verstärkung der Wert 1,0 in der Berechnung des Integral- und des Differentialanteils eingesetzt wird.

Umwandeln und Normalisieren der Eingabewerte

Ein Regelkreis verfügt über zwei Eingangsvariablen, den Sollwert und den Istwert (Prozeßvariable). Der Sollwert ist üblicherweise ein fester Wert wie z. B. beim Einstellen einer Geschwindigkeit für einen Fahrgeschwindigkeitsregler (Tempomat) im Pkw. Die Prozeßvariable ist ein Wert, der auch auf die Stellgröße des Regelkreises bezieht und deshalb die Auswirkungen mißt, die die Stellgröße auf das geregelte System hat. In dem Beispiel des Tempomats ist die Prozeßvariable der Eingabewert eines Drehzahlmessers, der die Drehgeschwindigkeit der Räder mißt.

Beide Werte, der Sollwert und der Istwert, sind Analogwerte, deren Größe, Bereich und Einheiten unterschiedlich sein können. Bevor diese Werte von der Operation PID verwendet werden können, müssen die Werte in normalisierte Gleitpunktdarstellungen umgewandelt werden.

Hierzu muß zunächst der Analogwert, der als ganze Zahl (16 Bit) vorliegt, in einen Gleitpunktwert bzw. in eine Realzahl umgewandelt werden. Die folgenden Anweisungen zeigen, wie eine ganze Zahl in eine Realzahl umgewandelt werden kann:

```
XORD  AC0, AC0           // Akkumulator zurücksetzen
MOVW  AEW0, AC0         // Analogwert im Akkumulator speichern.
LDW>= AC0, 0           // Ist der Analogwert positiv,
JMP   0                 // in Realzahl umwandeln.
NOT   0                 // Andernfalls,
ORD   16#FFFF0000, AC0 // Wert in AC0 mit Vorzeichen versehen.
LBL   0
DTR   AC0, AC0         // Ganze Zahl (32 Bit) in Realzahl umwandeln.
```

Als nächstes muß die Realzahl, die den Analogwert darstellt, in einen normalisierten Wert zwischen 0,0 und 1,0 umgewandelt werden. Mit Hilfe der folgenden Gleichung normalisieren Sie den Sollwert oder den Wert der Prozeßvariablen:

$$R_{\text{Norm}} = (R_{\text{ur}} / \text{Spanne}) + \text{Versatz}$$

Erklärung:

R_{Norm}	normalisierte Realzahl des Analogwerts
R_{ur}	nicht normalisierte Realzahl des Analogwerts
Versatz	0,0 bei einpoligen Werten 0,5 bei zweipoligen Werten
Spanne	maximal möglicher Wert abzüglich des minimal möglichen Werts 32000 bei einpoligen Werten (typischerweise) 64000 bei zweipoligen Werten (typischerweise)

Die folgenden Anweisungen zeigen, wie ein zweipoliger Wert in AC0 (dessen Spanne 64000 ist) im Anschluß an die oben dargestellten Anweisungen normalisiert werden kann:

```
/R    64000,0, AC0      // Wert im Akkumulator normalisieren
+R    0,5, AC0         // Versatz für den Wert auf einen Bereich von 0,0 bis
1,0
MOVR  AC0, VD100      // Normalisierten Wert in TABLE speichern
```

Umwandeln der Stellgröße des Regelkreises in einen skalierten ganzzahligen Wert

Die Stellgröße ist wie die Drosselklappe in dem Beispiel des Tempomats im Pkw regelbar. Die Stellgröße ist eine normalisierte Realzahl zwischen 0,0 und 1,0. Bevor mit der Stellgröße ein Analogausgang geregelt werden kann, muß die Stellgröße in einen skalierten ganzzahligen Wert (16 Bit) umgewandelt werden. Dies geschieht genau umgekehrt wie beim Umwandeln des Sollwerts oder des Istwerts in einen normalisierten Wert. Zunächst müssen Sie die Stellgröße in eine skalierte Realzahl umwandeln. Hierzu verwenden Sie folgende Gleichung:

$$R_{\text{Skal}} = (M_n - \text{Versatz}) * \text{Spanne}$$

Erklärung:

R_{Skal}	skalierte Realzahl der Stellgröße
M_n	normalisierte Realzahl der Stellgröße
Versatz	0,0 bei einpoligen Werten 0,5 bei zweipoligen Werten
Spanne	maximal möglicher Wert abzüglich des minimal möglichen Werts 32000 bei einpoligen Werten (typischerweise) 64000 bei zweipoligen Werten (typischerweise)

Die folgenden Anweisungen zeigen, wie Sie die Stellgröße skalieren:

```

MOVR  VD108, AC0      // Stellgröße in Akkumulator übertragen
-R    0,5, AC0        // Anweisung nur bei zweipoligen Werten angeben
*R    64000,0, AC0    // Wert im Akkumulator skalieren
    
```

Anschließend muß die skalierte Realzahl, die die Stellgröße darstellt, in eine ganze Zahl (16 Bit) umgewandelt werden. Die folgenden Anweisungen zeigen, wie Sie diese Umwandlung durchführen:

```

TRUNC AC0, AC0        // Realzahl in ganze Zahl (32 Bit) wandeln
MOVW  AC0, AAW0      // Ganze Zahl (16 Bit) in Analogausgang schreiben
    
```

Vorwärts- und Rückwärtsverhalten in Regelkreisen

Der Regelkreis zeigt Vorwärtsverhalten, wenn die Verstärkung positiv ist. Eine negative Verstärkung bewirkt ein Rückwärtsverhalten. (Bei einem I- oder ID-Regler mit einer Verstärkung von 0.0 bewirken Sie Vorwärtsverhalten, wenn Sie positive Werte für die Integral- und die Differentialzeit angeben. Bei Angabe von negativen Werten für diese Zeiten wird Rückwärtsverhalten ausgelöst.)

Variablen und Bereiche

Die Prozeßvariable (der Istwert) und der Sollwert sind Eingabewerte bei der PID-Berechnung. Deshalb werden die Felder in der Tabelle für den Regelkreis gelesen aber von der Operation PID nicht geändert.

Die Stellgröße wird vom PID-Regler berechnet, so daß das Feld für die Stellgröße in der Tabelle für den Regelkreis nach jeder PID-Berechnung aktualisiert wird. Die Stellgröße wird zwischen 0,0 und 1,0 festgesetzt. Das Feld für die Stellgröße kann als Eingabewert für eine anfängliche Stellgröße verwendet werden, wenn von der manuellen Regelung zur automatischen Regelung mittels PID gewechselt werden soll (ausführliche Informationen hierzu entnehmen Sie dem folgenden Abschnitt zu den Betriebsarten).

Wird ein Integralregler verwendet, dann wird der Wert der Integralsumme bei der PID-Berechnung aktualisiert und die aktualisierte Integralsumme als Eingabewert bei der nächsten PID-Berechnung verwendet. Wenn die errechnete Stellgröße außerhalb des Bereichs liegt (d.h. die Stellgröße wäre kleiner als 0,0 oder größer als 1,0), dann wird die Integralsumme nach der folgenden Gleichung angepaßt:

$$\mathbf{MX} = \mathbf{1,0} - (\mathbf{MP}_n + \mathbf{MD}_n) \quad \text{wenn die errechnete Stellgröße } M_n > 1,0$$

oder

$$\mathbf{MX} = - (\mathbf{MP}_n + \mathbf{MD}_n) \quad \text{wenn die errechnete Stellgröße } M_n < 0,0$$

Erklärung:

MX	Wert der angepaßten Integralsumme
MP _n	Wert des Proportionalanteils der Stellgröße bei Abtastzeit n
MD _n	Wert des Differentialanteils der Stellgröße bei Abtastzeit n
M _n	Wert der Stellgröße bei einer Abtastzeit n

Passen Sie die Integralsumme wie beschrieben an, verbessert sich die Ansprechempfindlichkeit des Systems, wenn sich die errechnete Stellgröße wieder im zulässigen Bereich befindet. Die errechnete Integralsumme wird auch auf den Bereich von 0,0 bis 1,0 festgesetzt und in das Feld für die Integralsumme in der Tabelle für den Regelkreis geschrieben. Dies geschieht nach Ausführung der PID-Berechnung. Der Wert, der in der Tabelle für den Regelkreis abgelegt ist, wird für die nächste PID-Berechnung verwendet.

Sie können den Wert der Integralsumme in der Tabelle für den Regelkreis vor Ausführung der Operation PID ändern, um so auf bestimmte Situationen in verschiedenen Anwendungen über die Integralsumme Einfluß zu nehmen. Gehen Sie aber vorsichtig vor, wenn Sie die Integralsumme manuell anpassen. Es muß sich bei jedem Wert, der für die Integralsumme in die Tabelle für den Regelkreis geschrieben wird, um eine Realzahl zwischen 0,0 und 1,0 handeln.

Für die Prozeßvariablen wird ein Vergleichswert in der Tabelle gespeichert, der für den Differentialanteil eines PID-Reglers eingesetzt werden kann. Diesen Wert dürfen Sie nicht ändern.

Betriebsarten

Es gibt keine integrierte Betriebsartensteuerung für die PID-Regelkreise der S7-200. Die PID-Berechnung wird durch Signalfluß an der Box PID aktiviert. Deshalb werden die PID-Berechnungen im Automatikbetrieb zyklisch ausgeführt. Im manuellen Betrieb werden keine PID-Berechnungen ausgeführt.

Die Operation PID hat ein Verlaufsbit für den Signalzustand, ähnlich wie bei Zähloperationen. Die Operation erkennt anhand dieses Bits eine steigende Flanke. Bei einer steigenden Flanke führt die Operation eine Reihe von Schritten aus, damit der Übergang vom manuellen Betrieb zum Automatikbetrieb reibungslos abläuft. Damit der Übergang in den Automatikbetrieb nicht sprunghaft verläuft, muß der Wert der Stellgröße für manuellen Betrieb als Eingabewert für die Operation PID bereitgestellt werden (als Wert für M_n in der Tabelle eingetragen), bevor in den Automatikbetrieb gewechselt wird. Die Operation PID bearbeitet die Werte in der Tabelle für den Regelkreis folgendermaßen, damit ein glatter Übergang vom manuellen zum automatischen Betrieb bei steigender Flanke gewährleistet ist:

- Sollwert (SW_n) wird gleich dem Istwert (IW_n) gesetzt.
- Alter Istwert (IW_{n-1}) wird gleich dem Istwert (IW_n) gesetzt.
- Integralsumme (MX) wird gleich der Stellgröße (M_n) gesetzt.

Standardmäßig ist das PID-Verlaufsbit "gesetzt". Dieser Zustand wird beim Anlauf der CPU und bei jedem Wechsel des Betriebszustand der CPU von STOP in RUN hergestellt. Besteht zum ersten Mal nach dem Wechsel in den Betriebszustand RUN an der Box PID Signalfluß, dann wird keine Flanke im Signalfluß erkannt, und es werden auch keine Schritte eingeleitet, damit ein Übergang in den Automatikbetrieb nicht sprunghaft verläuft.

Alarmer und Sonderoperationen

Die Operation PID ist eine einfache und dennoch leistungsstarke Operation zur PID-Berechnung. Sind andere Funktionen erforderlich, wie beispielsweise Alarmfunktionen oder besondere Berechnungen von Variablen im Regelkreis, dann müssen Sie diese Funktionen mittels der von Ihrer CPU unterstützten Operationen implementieren.

Fehlerbedingungen

Beim Übersetzen meldet die CPU einen Übersetzungsfehler, wenn die Parameter der Operation, die Anfangsadresse der Tabelle für den Regelkreis oder die Nummer für den PID-Regelkreis außerhalb des zulässigen Bereichs liegen. Die Übersetzung ist dann nicht erfolgreich.

Einige der Eingabewerte in der Tabelle für den Regelkreis werden von der Operation PID nicht auf ihren Bereich überprüft. Sie müssen deshalb darauf achten, daß die Prozeßvariablen/Istwerte und die Sollwerte (sowie die Integralsumme und die vorherigen Prozeßvariablen, sofern diese als Eingabewerte eingesetzt werden) Realzahlen in dem Bereich von 0,0 bis 1,0 sind.

Wird bei Ausführung der arithmetischen Funktionen für die PID-Berechnung ein Fehler erkannt, dann wird der Sondermerker SM1.1 gesetzt (Überlauf bzw. ungültiger Wert) und die Ausführung der Operation PID wird beendet. (Die Aktualisierung der Werte für die Stellgröße in der Tabelle für den Regelkreis kann unvollständig sein. Verwenden Sie diese Werte deshalb nicht, sondern korrigieren Sie den Eingabewert, der den Fehler verursacht hat, bevor Sie den PID-Regler erneut ausführen.)

Tabelle für den Regelkreis

Die Tabelle für den Regelkreis umfaßt 36 Bytes und hat folgendes Format (siehe Tabelle 10-12):

Tabelle 10-12 Format der Tabelle für den Regelkreis

Versatz	Feld	Format	Datentyp	Beschreibung
0	Prozeßvariable/ Istwert (IW_n)	Doppelwort - Realzahl	In	Enthält den Istwert bzw. die Prozeßvariable, die zwischen 0,0 und 1,0 skaliert sein muß.
4	Sollwert (SW_n)	Doppelwort - Realzahl	In	Enthält den Sollwert, der zwischen 0,0 und 1,0 skaliert sein muß.
8	Stellgröße (M_n)	Doppelwort - Realzahl	In/Out	Enthält die errechnete Stellgröße, die zwischen 0,0 und 1,0 skaliert ist.
12	Verstärkung (K_C)	Doppelwort - Realzahl	In	Enthält die Verstärkung, bei der es sich um eine proportionale Konstante handelt. Sie kann positiv oder negativ sein.
16	Abtastzeit (T_S)	Doppelwort - Realzahl	In	Enthält die Abtastzeit in Sekunden. Der Wert muß positiv sein.
20	Integralzeit (T_I)	Doppelwort - Realzahl	In	Enthält die Integralzeit in Minuten. Der Wert muß positiv sein.
24	Differentialzeit (T_D)	Doppelwort - Realzahl	In	Enthält die Differentialzeit in Minuten. Der Wert muß positiv sein.
28	Integralsumme/ Bias(MX)	Doppelwort - Realzahl	In/Out	Enthält die Integralsumme bzw. Bias zwischen 0,0 und 1,0.
32	Vorheriger Istwert/ Prozeßvariable (IW_{n-1})	Doppelwort - Realzahl	In/Out	Enthält den vorherigen Wert der Prozeßvariablen bzw. den vorherigen Istwert von der letzten Ausführung der Operation PID.

Programmierbeispiel für die Operation PID-Regler

In diesem Beispiel soll in einem Wasserbehälter ein konstanter Wasserdruck erhalten bleiben. Aus dem Behälter wird ständig Wasser entnommen, jedoch mit unterschiedlicher Geschwindigkeit. Eine Pumpe mit variabler Antriebsdrehzahl pumpt Wasser in den Behälter, und zwar mit einer Drehzahl, die den erforderlichen Wasserdruck im Behälter erhält und dafür sorgt, daß der Behälter nicht leer wird.

Der Sollwert für dieses System ist die Einstellung des Wasserstands auf ein Niveau von 75%. Die Prozeßvariable (der Istwert) wird durch einen Schwimmer geliefert, der anzeigt, wieviel Wasser im Behälter ist. Dieser Istwert kann zwischen 0% und 100% liegen. Die Stellgröße ist ein Wert zwischen 0% und 100% für die Antriebsdrehzahl der Pumpe.

Der Sollwert wird vorab festgelegt und direkt in die Tabelle für den Regelkreis eingetragen. Die Prozeßvariable (der Istwert) wird als einpoliger Analogwert vom Schwimmer geliefert. Die Stellgröße wird als einpoliger Analogwert geschrieben und dient zum Regeln der Pumpendrehzahl. Die Spanne für den Analogeingang und den Analogausgang ist bei beiden 32000.

In diesem Beispiel werden nur Proportional- und Integralregler eingesetzt. Die Verstärkung im Regelkreis und die Zeitkonstanten wurden in Berechnungen ermittelt und können angepaßt werden, um eine optimale Regelung zu erzielen. Die errechneten Werte für die Zeitkonstanten sind folgende:

K_C ist 0,25

T_S ist 0,1 s

T_I sind 30 min

Die Pumpendrehzahl wird manuell geregelt, bis der Wasserstand im Behälter 75% beträgt. Dann wird das Ventil geöffnet, damit das Wasser aus dem Behälter abfließt. Gleichzeitig wird die Pumpe von manuellem Betrieb auf Automatikbetrieb umgeschaltet. Ein Digitaleingang wird für den Übergang von manuellem zu automatischem Betrieb verwendet. Dieser Eingang ist folgendermaßen definiert:

E0.0 - manueller/automatischer Betrieb; 0 - manuell, 1 - automatisch

Im manuellen Betrieb schreibt der Operator die Pumpendrehzahl als Realzahl zwischen 0,0 und 1,0 in VD108.

Bild 10-21 zeigt das Programm für diese Anwendung.

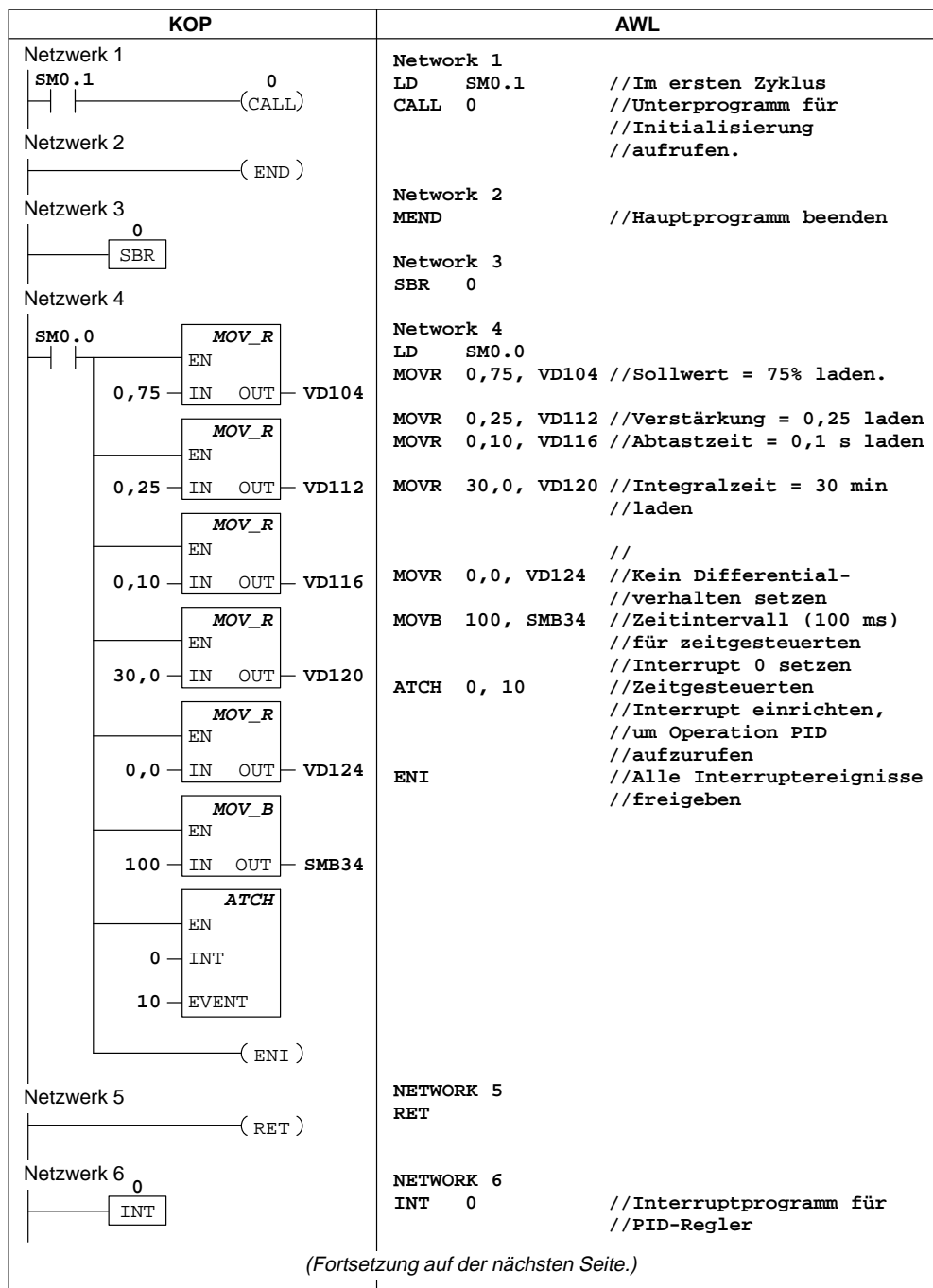


Bild 10-21 Beispiel für einen PID-Regler

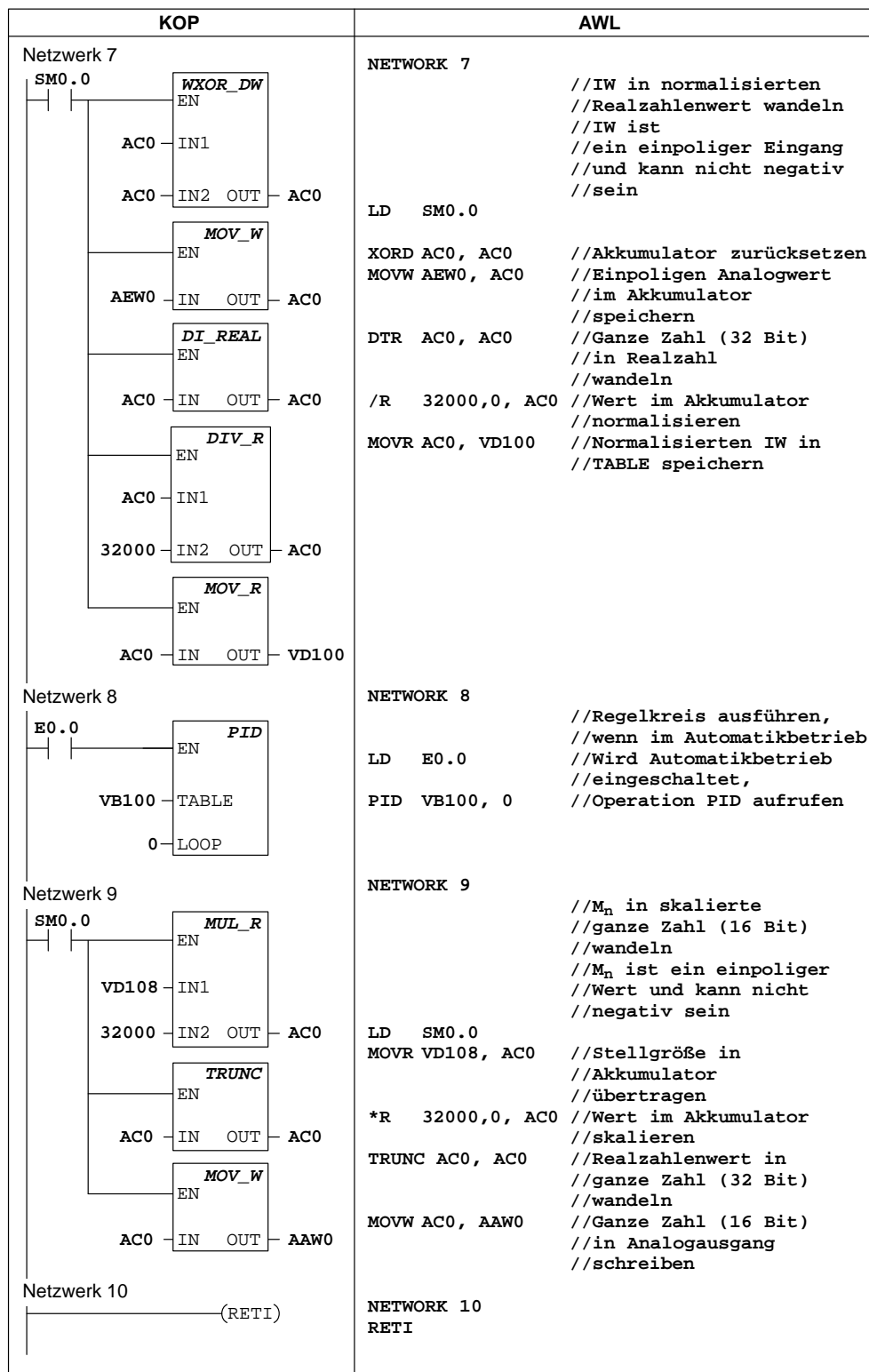
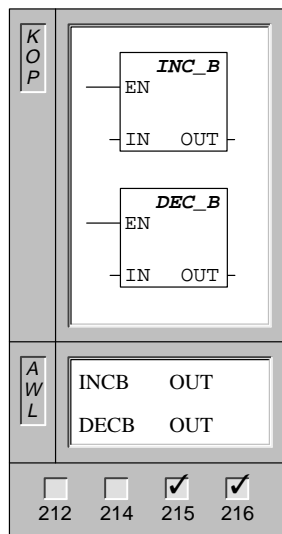


Bild 10-21 Beispiel für einen PID-Regler, Fortsetzung

10.7 Operationen zum Inkrementieren und Dekrementieren

Byte um 1 erhöhen und Byte um 1 vermindern



Die Operationen **Byte um 1 erhöhen** und **Byte um 1 vermindern** addieren bzw. subtrahieren den Wert "1" zu bzw. von dem Wert des Eingangsbyte.

Operanden: IN: VB, EB, AB, MB, SMB, SB, AC, Konstante, *VD, *AC, SB
 OUT: VB, EB, AB, MB, SMB, SB, AC, *VD, *AC, SB

In KOP: IN + 1 = OUT
 IN - 1 = OUT

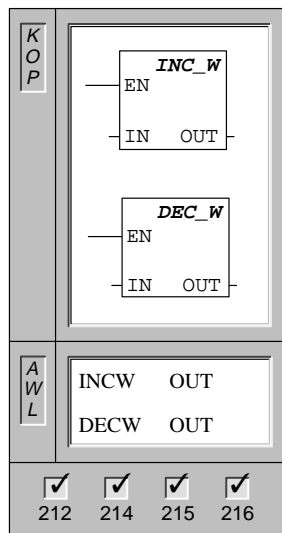
In AWL: OUT + 1 = OUT
 OUT - 1 = OUT

Die Operationen Byte um 1 erhöhen und Byte um 1 vermindern sind vorzeichenlos.

Hinweis: Wenn Sie in KOP programmieren, können Sie angeben, daß IN1 gleich OUT ist. Auf diese Weise sparen Sie Speicherplatz.

Diese Operationen beeinflussen die folgenden Sondermerker: SM1.0 (Null); SM1.1 (Überlauf)

Wort um 1 erhöhen und Wort um 1 vermindern



Die Operationen **Wort um 1 erhöhen** und **Wort um 1 vermindern** addieren bzw. subtrahieren den Wert "1" zu bzw. von dem Wert des Eingangswort.

Operanden: IN: VW, T, Z, EW, AW, MW, SMW, AC, AEW, Konstante, *VD, *AC, SW
 OUT: VW, T, Z, EW, AW, MW, SMW, AC, *VD, *AC, SW

In KOP: IN + 1 = OUT
 IN - 1 = OUT

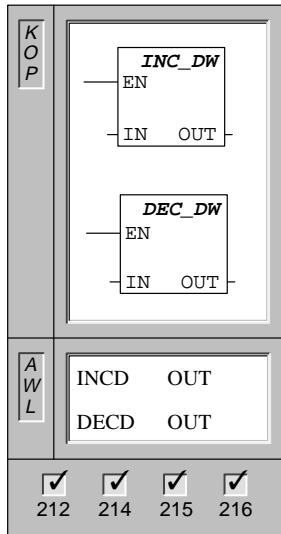
In AWL: OUT + 1 = OUT
 OUT - 1 = OUT

Die Operationen Wort um 1 erhöhen und Wort um 1 vermindern haben ein Vorzeichen (116#7FFF > 16#8000).

Hinweis: Wenn Sie in KOP programmieren, können Sie angeben, daß IN1 gleich OUT ist. Auf diese Weise sparen Sie Speicherplatz.

Diese Operationen beeinflussen die folgenden Sondermerker: SM1.0 (Null); SM1.1 (Überlauf); SM1.2 (negativ)

Doppelwort um 1 erhöhen und Doppelwort um 1 vermindern



Die Operationen **Doppelwort um 1 erhöhen** und **Doppelwort um 1 vermindern** addieren bzw. subtrahieren den Wert "1" zu bzw. von dem Wert des Eingangsdoppelworts.

Operanden: IN: VD, ED, AD, MD, SMD, AC, HC, Konstante, *VD, *AC, SD
 OUT: VD, ED, AD, MD, SMD, AC, *VD, *AC, SD

In KOP: IN + 1 = OUT
 IN - 1 = OUT

In AWL: OUT + 1 = OUT
 OUT - 1 = OUT

Die Operationen Wort um 1 erhöhen und Wort um vermindern haben ein Vorzeichen (16#7FFFFFFF > 16#80000000).

Hinweis: Wenn Sie in KOP programmieren, können Sie angeben, daß IN1 gleich OUT ist. Auf diese Weise sparen Sie Speicherplatz.

Diese Operationen beeinflussen die folgenden Sondermerker: SM1.0 (Null); SM1.1 (Überlauf); SM1.2 (negativ)

Beispiel für Inkrementieren und Dekrementieren

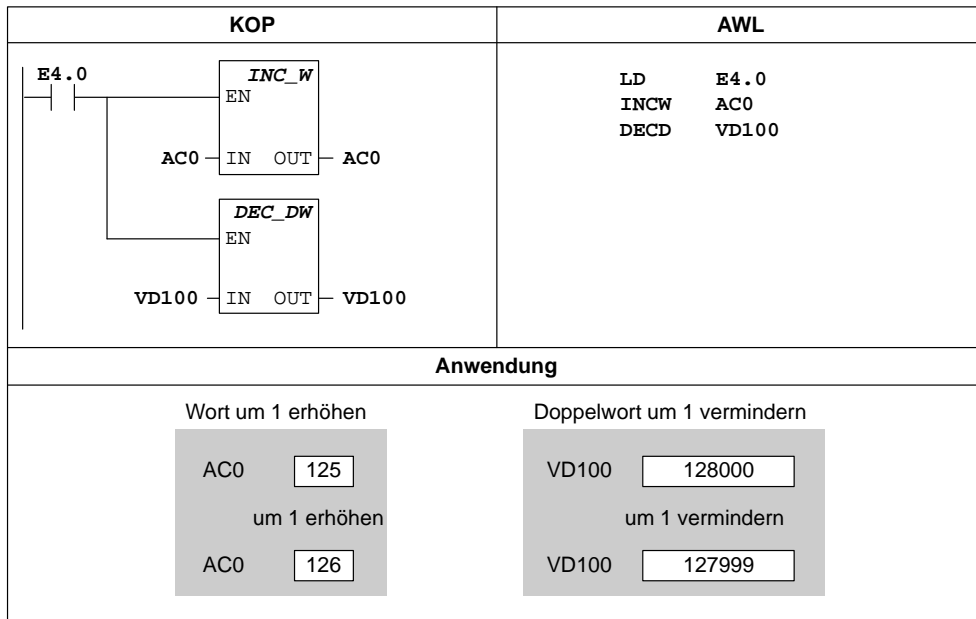
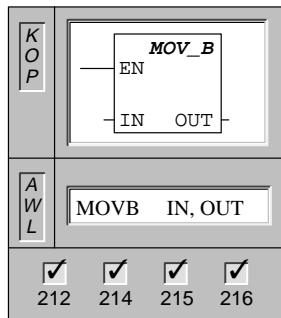


Bild 10-22 Beispiel für Inkrementieren und Dekrementieren in KOP und AWL

10.8 Übertragungs- und Tabellenoperationen

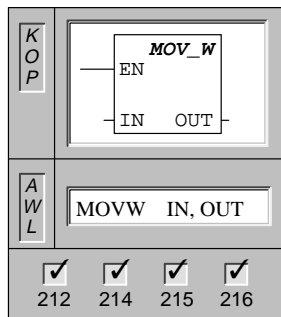
Byte übertragen



Die Operation **Byte übertragen** überträgt das Eingangsbyte (IN) zum Ausgangsbyte (OUT). Das Eingangsbyte wird dadurch nicht verändert.

Operanden: IN: VB, EB, AB, MB, SMB, AC, *VD, *AC, SB
 OUT: VB, EB, AB, MB, SMB, AC, *VD, *AC, *AC, SB

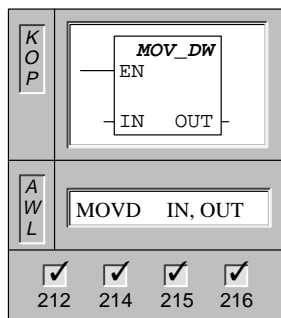
Wort übertragen



Die Operation **Wort übertragen** überträgt das Eingangswort (IN) zum Ausgangswort (OUT). Das Eingangswort wird dadurch nicht verändert.

Operanden: IN: VW, T, Z, EW, AW, MW, SMW, AC, AEW, Konstante, *VD, *AC, SW
 OUT: VW, T, Z, EW, AW, MW, SMW, AC, AAW, *VD, *AC, SW

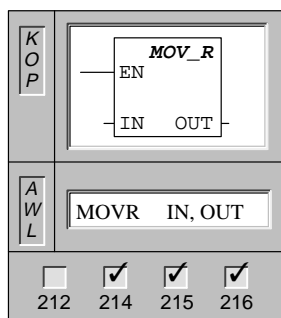
Doppelwort übertragen



Die Operation **Doppelwort übertragen** überträgt das Eingangsdoppelwort (IN) zum Ausgangsdoppelwort (OUT). Das Eingangsdoppelwort wird dadurch nicht verändert.

Operanden: IN: VD, ED, AD, MD, SMD, AC, HC, Konstante, *VD, *AC, &VB, &EB, &MB, &T, &Z, &SB, SD
 OUT: VD, ED, AD, MD, SMD, AC, *VD, *AC, SD

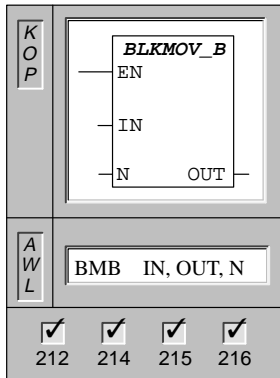
Realzahl übertragen



Die Operation **Realzahl übertragen** überträgt die Eingangsrealzahl (Doppelwort, 32 Bit) (IN) zum Ausgangsdoppelwort (OUT). Das Eingangsdoppelwort wird dadurch nicht verändert.

Operanden: IN: VD, ED, AD, MD, SMD, AC, Konstante, *VD, *AC, SD
 OUT: VD, ED, AD, MD, SMD, AC, *VD, *AC, SD

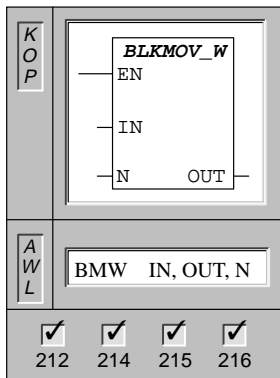
Anzahl an Bytes übertragen



Die Operation **Anzahl an Bytes übertragen** überträgt eine angegebene Anzahl an Bytes (N) von dem Eingangsfeld, das bei IN beginnt, zu dem Ausgangsfeld, das bei OUT beginnt. N kann zwischen 1 und 255 liegen.

Operanden: IN, OUT: VB, EB, AB, MB, SMB, *VD, *AC, SB
 N: VB, EB, AB, MB, SMB, AC, Konstante, *VD, *AC, SB

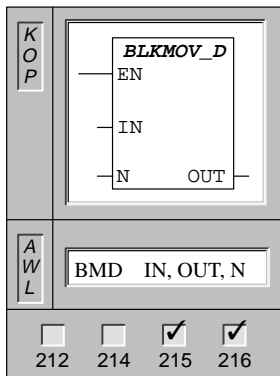
Anzahl an Wörtern übertragen



Die Operation **Anzahl an Wörtern übertragen** überträgt eine angegebene Anzahl an Wörtern (N) von dem Eingangsfeld, das bei IN beginnt, zu dem Ausgangsfeld, das bei OUT beginnt. N kann zwischen 1 und 255 liegen.

Operanden: IN: VW, T, Z, EW, AW, MW, SMW, AEW, *VD, *AC, SW
 OUT: VW, T, Z, EW, AW, MW, SMW, AAW, *VD, *AC, SW
 N: VB, EB, AB, MB, SMB, AC, Konstante, *VD, *AC, SB

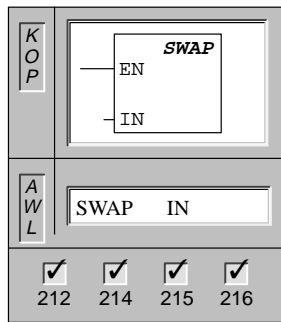
Anzahl an Doppelwörtern übertragen



Die Operation **Anzahl an Doppelwörtern übertragen** überträgt eine angegebene Anzahl an Doppelwörtern (N) von dem Eingangsfeld, das bei IN beginnt, zu dem Ausgangsfeld, das bei OUT beginnt. N kann zwischen 1 und 255 liegen.

Operanden: IN, OUT: VD, ED, AD, MD, SMD, *VD, *AC, SD
 N: VB, EB, AB, MB, SMB, AC, Konstante, *VD, *AC, SB

Bytes im Wort tauschen



Die Operation **Bytes im Wort tauschen** vertauscht das höchstwertige Byte mit dem niederwertigsten Byte des Worts (IN).

Operanden: IN: VW, T, Z, EW, AW, MW, SMW, AC, *VD, *AC, SW

Beispiele für Übertragungsoperationen und die Operation SWAP

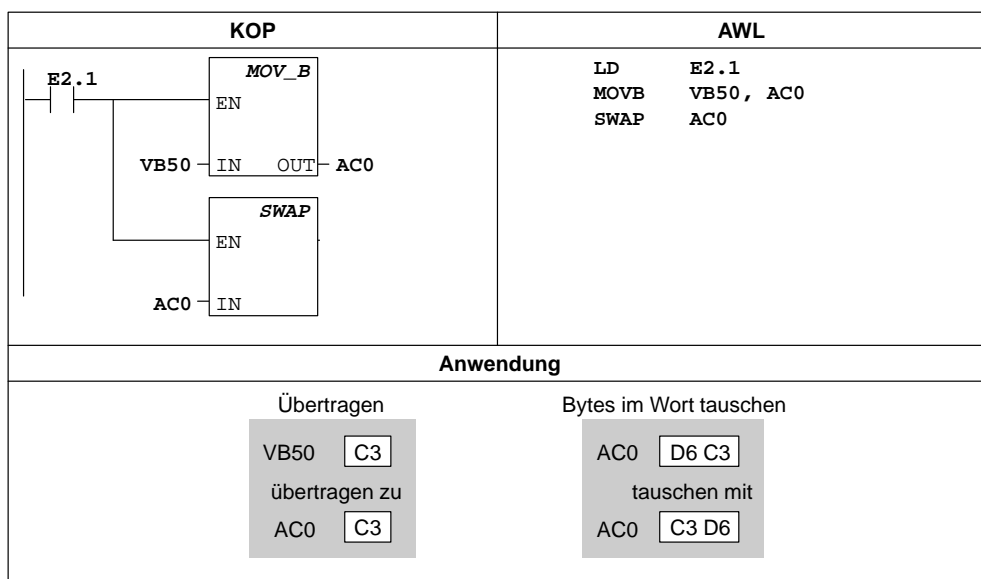


Bild 10-23 Beispiel für die Übertragungsoperationen in KOP und AWL

Beispiel für die Operation Anzahl an Bytes übertragen

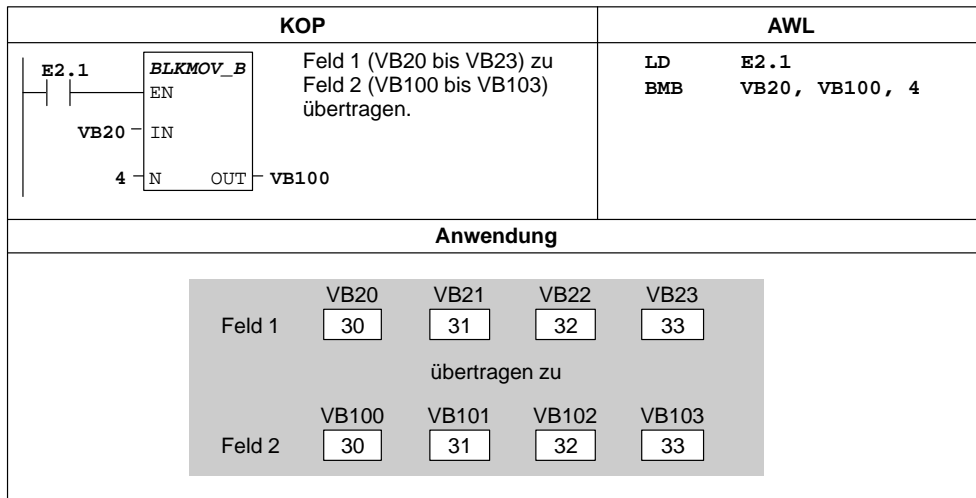
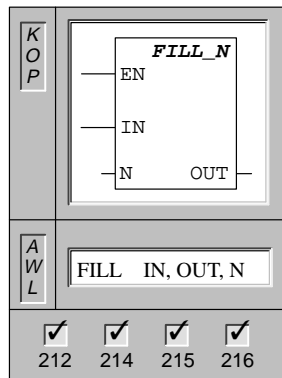


Bild 10-24 Beispiel für die Operation Anzahl an Bytes übertragen in KOP und AWL

Speicher mit Bitmuster belegen



Die Operation **Speicher mit Bitmuster belegen** belegt einen Speicherbereich, der bei dem Ausgangswort OUT beginnt, mit dem Wortmuster des Eingangs IN für die angegebene Anzahl an Wörtern N. N kann zwischen 1 und 255 liegen.

Operanden: IN: VW, T, Z, EW, AW, MW, SMW, AEW, Konstante, *VD, *AC, SW
 OUT: VW, T, Z, EW, AW, MW, SMW, AAW, *VD, *AC, SW
 N: VB, EB, AB, MB, SMB, AC, Konstante, *VD, *AC, SB

Beispiel für die Operation Speicher mit Bitmuster belegen

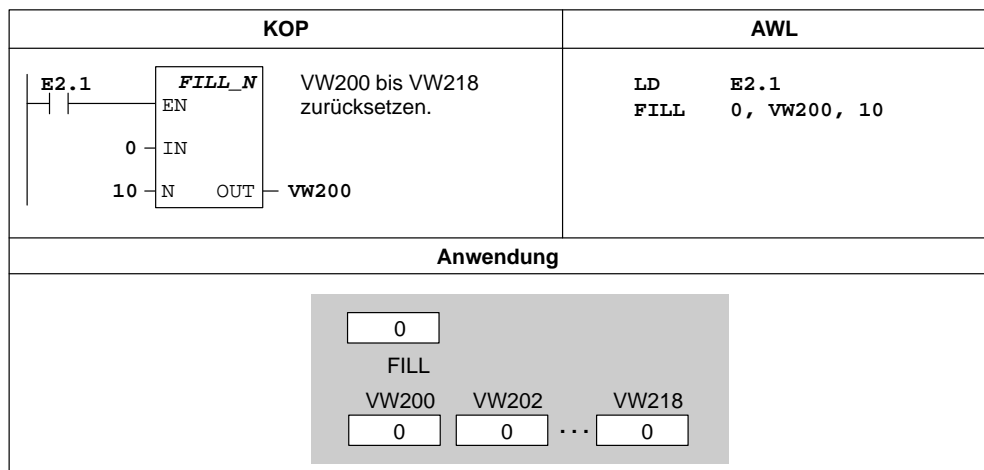
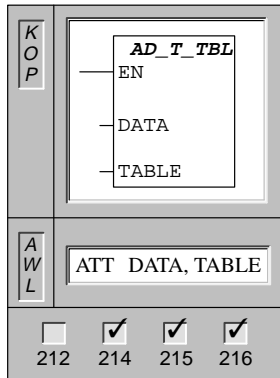


Bild 10-25 Beispiel für die Operation Speicher mit Bitmuster belegen in KOP und AWL

Wert in Tabelle eintragen



Die Operation **Wert in Tabelle eintragen** trägt Wortwerte (DATA) in die Tabelle (TABLE) ein.

Operanden: DATA: VW, T, Z, EW, AW, MW, SMW, AC, AEW, Konstante, *VD, *AC, SW
 TABLE: VW, T, Z, EW, AW, MW, SMW, *VD, *AC, SW

Der erste Wert in der Tabelle gibt die maximale Länge der Tabelle (TL) an. Der zweite Wert (EC) gibt die Anzahl der Tabelleneinträge an (siehe Bild 10-26.) Neue Daten werden in der Tabelle nach dem letzten Eintrag ergänzt. Jedesmal wenn neue Daten eingetragen werden, wird die Anzahl der Einträge um "1" erhöht. Eine Tabelle kann maximal 100 Einträge umfassen. Nicht darin eingeschlossen sind die Parameter, die die maximale Länge der Tabelle und die tatsächliche Anzahl der Einträge angeben.

Diese Operation beeinflusst die folgenden Sondermerker:

SM1.4 wird auf "1" gesetzt, wenn Sie versuchen, zu viele Werte in die Tabelle einzutragen.

Beispiel für die Operation Wert in Tabelle eintragen

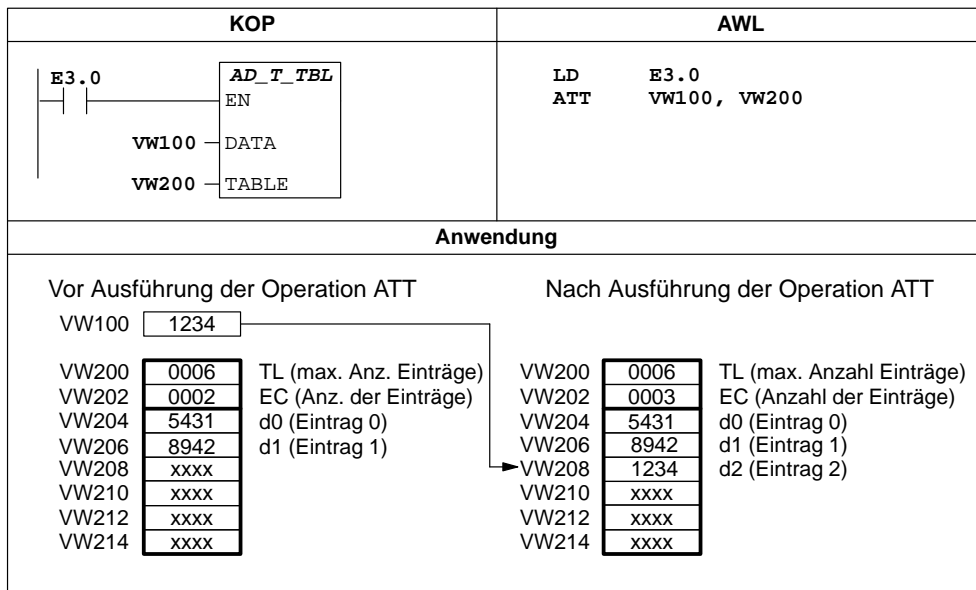
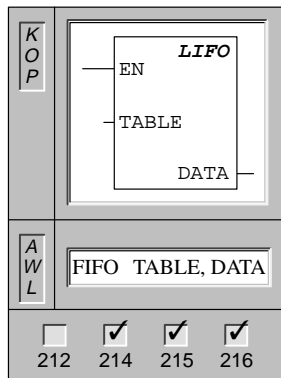


Bild 10-26 Beispiel für die Operation Wert in Tabelle eintragen

Letzten Wert aus Tabelle löschen



Die Operation **Letzten Wert aus Tabelle löschen** löscht den letzten Eintrag der Tabelle (TABLE) und gibt den Wert an die Adresse (DATA) aus. Die Anzahl der Einträge (EC) verringert sich jedesmal, wenn diese Operation ausgeführt wird, um "1".

Operanden: TABLE: VW, T, Z, EW, AW, MW, SMW, *VD, *AC, SW
 DATA: VW, T, Z, EW, AW, MW, SMW, AC, AAW, *VD, *AC, SW

Diese Operation beeinflusst die folgenden Sondermerker:

SM1.5 wird auf "1" gesetzt, wenn Sie versuchen, einen Eintrag aus einer leeren Tabelle zu löschen.

Beispiel für die Operation Letzten Wert aus Tabelle löschen

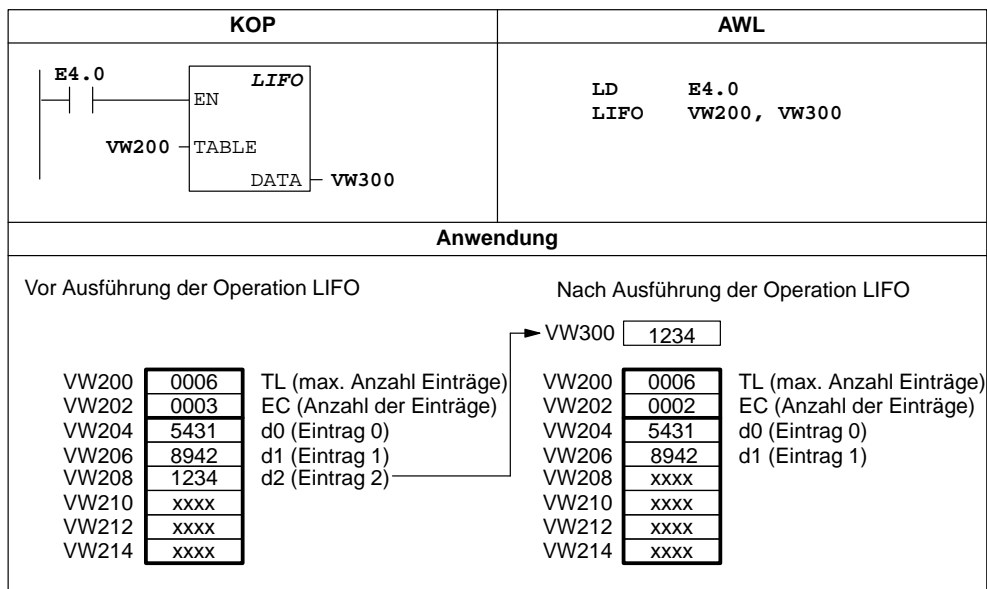
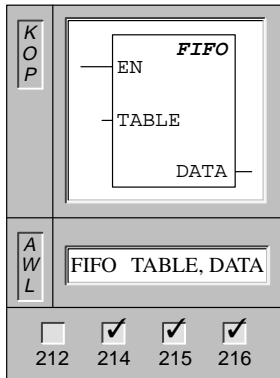


Bild 10-27 Beispiel für die Operation FIFO in KOP und AWL

Ersten Wert aus Tabelle löschen



Die Operation **Ersten Wert aus Tabelle löschen** löscht den ersten Eintrag der Tabelle (TABLE) und gibt den Wert an die Adresse (DATA) aus. Alle übrigen Einträge werden um eine Stelle nach oben verschoben. Die Anzahl der Einträge (EC) verringert sich jedesmal, wenn diese Operation ausgeführt wird, um "1".

Operanden: TABLE: VW, T, Z, EW, AW, MW, SMW, *VD, *AC, SW
 DATA: VW, T, Z, EW, AW, MW, SMW, AC, AAW, *VD, *AC, SW

Diese Operation beeinflusst die folgenden Sondermerker:

SM1.5 wird auf "1" gesetzt, wenn Sie versuchen, einen Eintrag aus einer leeren Tabelle zu löschen.

Beispiel für die Operation Ersten Wert aus Tabelle löschen

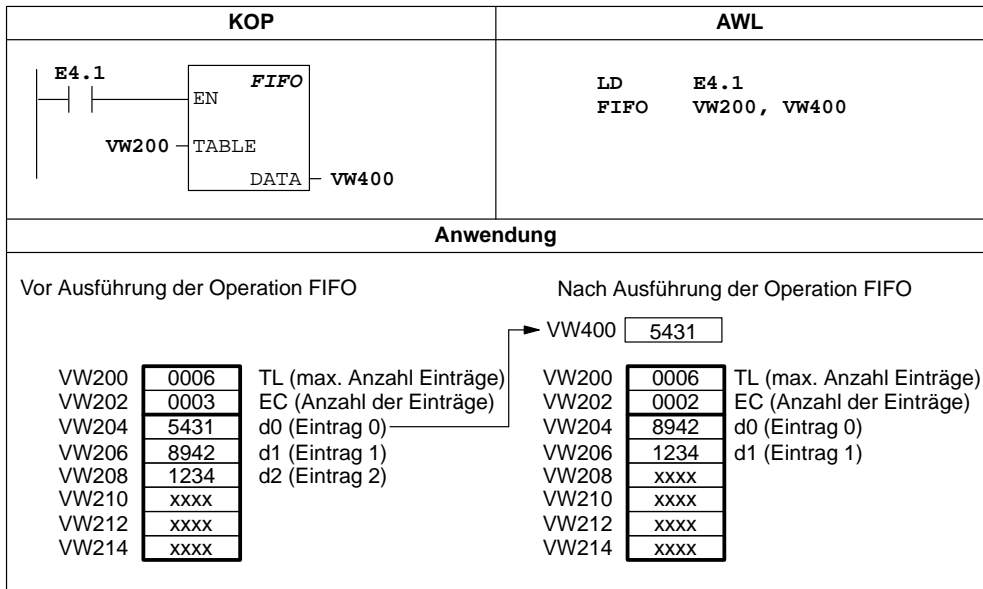
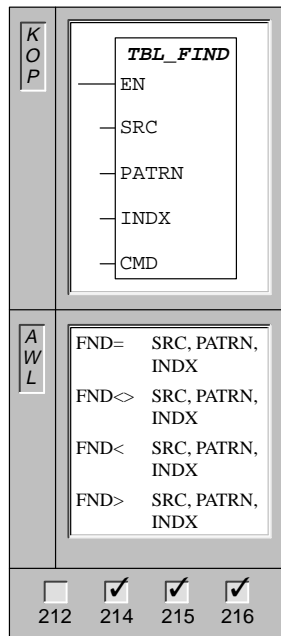


Bild 10-28 Beispiel für die Operation FIFO in KOP und AWL

Wert in Tabelle suchen



Die Operation **Wert in Tabelle suchen** durchsucht die Tabelle (SRC) beginnend bei dem Tabelleneintrag, der von INDX angegeben wird, nach dem Datenwert (PATRN), der den angegebenen Kriterien =, <>, < oder > entspricht.

In KOP gibt der Parameter CMD das Kriterium über einen numerischen Wert von 1 bis 4 an. Dieser Wert entspricht dem jeweiligen Kriterium =, <>, < oder >.

Operanden: SRC: VW, T, Z, EW, AW, MW, SMW, *VD, *AC, SW
 PATRN: VW, T, Z, EW, AW, MW, SMW, AC, AEW, Konstante, *VD, *AC, SW
 INDX: VW, T, Z, EW, AW, MW, SMW, AC, *VD, *AC, SW
 CMD: 1 (=) 2 (<>) 3 (<) 4 (>)

Wird ein entsprechender Tabelleneintrag gefunden, zeigt INDX diesen Eintrag an. Um den nächsten Tabelleneintrag zu suchen, muß INDX zunächst um "1" erhöht werden. Gibt es keinen passenden Eintrag in der Tabelle, entspricht der Wert von INDX der Anzahl der Einträge in der Tabelle.

Die Einträge in der Tabelle (der Bereich, der durchsucht werden soll) sind von 0 bis zum maximalen Wert 99 durchnummeriert. Eine Tabelle kann maximal 100 Einträge umfassen. Nicht darin eingeschlossen sind die Parameter, die die maximale Anzahl der Einträge und die tatsächliche Anzahl der Einträge in der Tabelle angeben.

Hinweis

Wenn Sie Suchoperationen in Tabellen verwenden, die mit den Operationen ATT, LIFO und FIFO erstellt wurden, dann entsprechen sich die Anzahl der Einträge und die Dateneinträge direkt. Im Gegensatz zu den Operationen ATT, LIFO und FIFO, bei denen in einem Wort die maximale Anzahl der Einträge angegeben wird, benötigen die Suchoperationen dieses Wort nicht. Deshalb ist der Operand SRC einer Suchoperation um eine Wortadresse (zwei Bytes) höher als der Operand TABLE einer entsprechenden Operation ATT, LIFO oder FIFO (siehe Bild 10-29).

Tabellenformat für ATT, LIFO und FIFO			Tabellenformat für TBL_FIND		
VW200	0006	TL (max. Anz. Einträge)	VW202	0006	EC (Anz. der Einträge)
VW202	0006	EC (Anz. der Einträge)	VW204	xxxx	d0 (Eintrag 0)
VW204	xxxx	d0 (Eintrag 0)	VW206	xxxx	d1 (Eintrag 1)
VW206	xxxx	d1 (Eintrag 1)	VW208	xxxx	d2 (Eintrag 2)
VW208	xxxx	d2 (Eintrag 2)	VW210	xxxx	d3 (Eintrag 3)
VW210	xxxx	d3 (Eintrag 3)	VW212	xxxx	d4 (Eintrag 4)
VW212	xxxx	d4 (Eintrag 4)	VW214	xxxx	d5 (Eintrag 5)
VW214	xxxx	d5 (Eintrag 5)			

Bild 10-29 Unterschiedliche Tabellenformate bei den Suchoperationen und den Operationen ATT, LIFO und FIFO

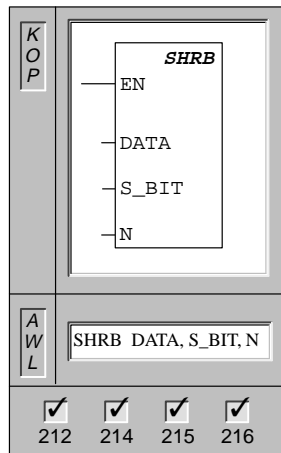
Beispiel für die Operation Wert in Tabelle suchen

KOP		AWL																					
<div style="border: 1px solid black; padding: 5px; display: inline-block;"> E2.1 VW202 — SRC 16#3130 — PATRN AC1 — INDX 1 — CMD </div>	Ist E3.1 eingeschaltet, dann wird die Tabelle VW202 nach einem Wert, der der Angabe 3130 in Hexadezimalziffern entspricht, durchsucht.	LD E2.1 FND= VW202, 16#3130, AC1																					
Anwendung																							
<p>Sie durchsuchen diese Tabelle. Wenn die Tabelle mit einer der Operationen ATT, LIFO oder FIFO erstellt wurde, enthält VW200 die maximal zulässige Anzahl an Einträgen und wird von den Suchoperationen nicht benötigt.</p> <table style="margin-left: 20px;"> <tr><td>VW202</td><td style="border: 1px solid black; text-align: center;">0006</td><td>EC (Anzahl der Einträge)</td></tr> <tr><td>VW204</td><td style="border: 1px solid black; text-align: center;">3133</td><td>d0 (Eintrag 0)</td></tr> <tr><td>VW206</td><td style="border: 1px solid black; text-align: center;">4142</td><td>d1 (Eintrag 1)</td></tr> <tr><td>VW208</td><td style="border: 1px solid black; text-align: center;">3130</td><td>d2 (Eintrag 2)</td></tr> <tr><td>VW210</td><td style="border: 1px solid black; text-align: center;">3030</td><td>d3 (Eintrag 3)</td></tr> <tr><td>VW212</td><td style="border: 1px solid black; text-align: center;">3130</td><td>d4 (Eintrag 4)</td></tr> <tr><td>VW214</td><td style="border: 1px solid black; text-align: center;">4541</td><td>d5 (Eintrag 5)</td></tr> </table>			VW202	0006	EC (Anzahl der Einträge)	VW204	3133	d0 (Eintrag 0)	VW206	4142	d1 (Eintrag 1)	VW208	3130	d2 (Eintrag 2)	VW210	3030	d3 (Eintrag 3)	VW212	3130	d4 (Eintrag 4)	VW214	4541	d5 (Eintrag 5)
VW202	0006	EC (Anzahl der Einträge)																					
VW204	3133	d0 (Eintrag 0)																					
VW206	4142	d1 (Eintrag 1)																					
VW208	3130	d2 (Eintrag 2)																					
VW210	3030	d3 (Eintrag 3)																					
VW212	3130	d4 (Eintrag 4)																					
VW214	4541	d5 (Eintrag 5)																					
AC1	<input style="width: 50px;" type="text" value="0"/>	AC1 muß auf "0" gesetzt sein, damit ab dem obersten Tabelleneintrag gesucht wird.																					
Tabelle durchsuchen																							
AC1	<input style="width: 50px;" type="text" value="2"/>	AC1 enthält die Nummer des ersten Eintrags, der den Suchkriterien entspricht.																					
AC1	<input style="width: 50px;" type="text" value="3"/>	Erhöhen Sie INDX um "1", bevor Sie die übrigen Einträge der Tabelle durchsuchen.																					
Tabelle durchsuchen																							
AC1	<input style="width: 50px;" type="text" value="4"/>	AC1 enthält die Nummer des zweiten Eintrags, der den Suchkriterien entspricht.																					
AC1	<input style="width: 50px;" type="text" value="5"/>	Erhöhen Sie INDX um "1", bevor Sie die übrigen Einträge der Tabelle durchsuchen.																					
Tabelle durchsuchen																							
AC1	<input style="width: 50px;" type="text" value="6"/>	AC1 enthält einen Wert, der der Anzahl der Einträge in der Tabelle entspricht. Die gesamte Tabelle wurde durchsucht, ohne einen weiteren passenden Eintrag zu finden.																					
AC1	<input style="width: 50px;" type="text" value="0"/>	Bevor Sie die Tabelle erneut durchsuchen können, müssen Sie den Wert von INDX auf "0" zurücksetzen.																					

Bild 10-30 Beispiel für Suchoperationen in KOP und AWL

10.9 Schiebe- und Rotieroperationen

Wert in Schieberegister schieben



Die Operation **Wert in Schieberegister schieben** schiebt den Wert von DATA in das Schieberegister. S_BIT gibt das niederwertigste Bit des Schieberegisters an. N zeigt die Länge des Schieberegisters und die Richtung, in die geschoben wird, an (positive Schiebefunktion = N, negative Schiebefunktion = -N).

Operanden: DATA, S_BIT: E, A, M, SM, T, Z, V, S
 N: VB, EB, AB, MB, SMB, SB, AC, Konstante, *VD, *AC, SB

Beschreibung der Operation Wert in Schieberegister schieben

Die Operation Wert in Schieberegister schieben dient dazu, einen Produktfluß oder Daten auf einfache Weise in Reihenfolge zu bringen und zu steuern. Mit der Operation Wert in Schieberegister schieben können Sie einmal pro Zyklus das gesamte Register um ein Bit verschieben. Das Schieberegister wird definiert durch das niederwertigste Bit (S_BIT) und die Anzahl der Bits, die durch die Länge (N) angegeben wird. Bild 10-32 zeigt ein Beispiel für die Operation Wert in Schieberegister schieben.

Die Adresse des höchstwertigen Bit im Schieberegister (MSB.b) kann durch folgende Gleichung errechnet werden:

$$\text{MSB.b} = [(\text{Byte von S_BIT}) + ((N) - 1 + (\text{Bit von S_BIT})) / 8] . [\text{Divisionsrest der Division durch } 8]$$

Sie müssen ein Bit subtrahieren, da S_BIT zu den Bits des Schieberegisters gehört.

Beispiel: Sind S_BIT = V33.4 und N = 14, dann ist MSB.b = V35.1 oder:

$$\begin{aligned} \text{MSB.b} &= \text{V}33 + ((14) - 1 + 4) / 8 \\ &= \text{V}33 + 17 / 8 \\ &= \text{V}33 + 2 \text{ mit Divisionsrest } 1 \\ &= \text{V}35.1 \end{aligned}$$

Bei einer negativen Schiebefunktion, die durch einen negativen Wert der Länge (N) angezeigt wird, werden die Eingangsdaten (DATA) in das höchstwertige Bit des Schieberegisters geschoben. Das niederwertigste Bit (S_BIT) wird aus dem Schieberegister hinausgeschoben.

Bei einer positiven Schiebefunktion, die durch einen positiven Wert der Länge (N) angezeigt wird, werden die Eingangsdaten (DATA) in das niederwertigste Bit des Schieberegisters - angezeigt durch S_BIT - geschoben. Das höchstwertige Bit wird aus dem Schieberegister hinausgeschoben.

Die hinausgeschobenen Daten werden im Überlaufmerker (SM1.1) abgelegt. Das Schieberegister hat eine maximale Länge von 64 Bits (positiv oder negativ). Bild 10-31 zeigt Schiebefunktionen mit positiven und negativen Werten für die Länge N.

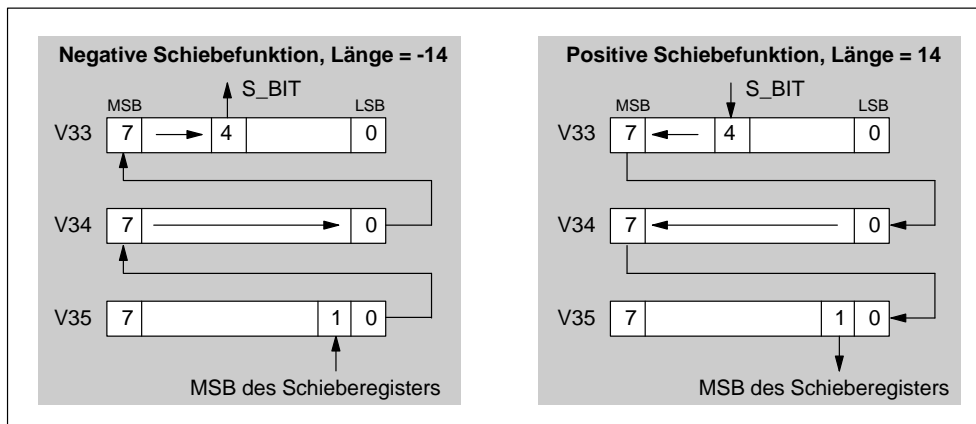


Bild 10-31 Positive und negative Schiebefunktionen im Schieberegister

Beispiel für die Operation Wert in Schieberegister schieben

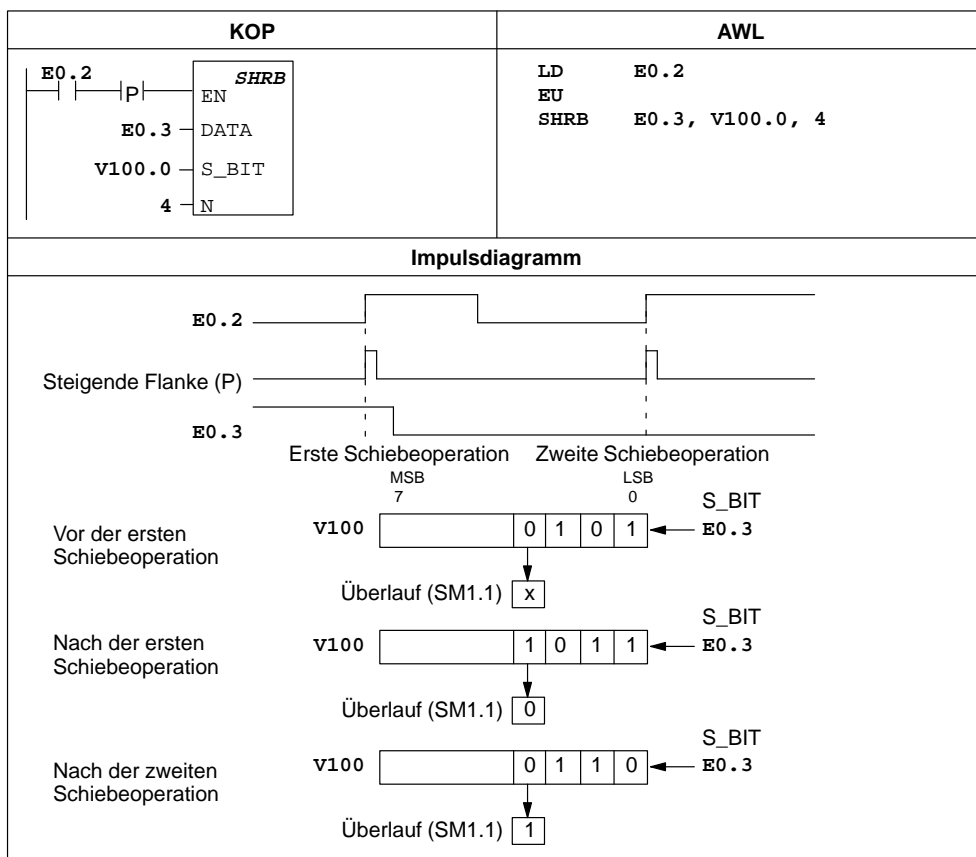
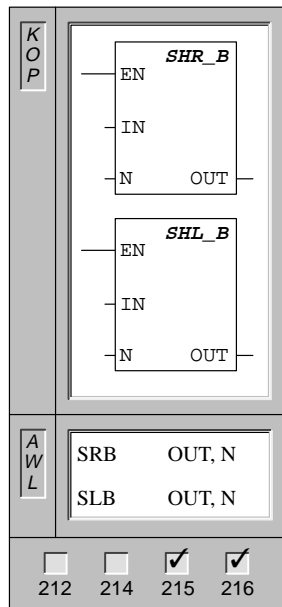


Bild 10-32 Beispiel für die Operation Wert in Schieberegister schieben in KOP und AWL

Byte rechts schieben und Byte links schieben



Die Operationen **Byte rechts schieben** und **Byte links schieben** schieben den Bytewert (IN) um den Schiebewert (N) nach rechts bzw. links und laden das Ergebnis in das Ausgangsbyte (OUT).

Operanden: IN: VB, EB, AB, MB, SMB, SB, AC, *VD, *AC
 N: VB, EB, AB, MB, SMB, SB, AC, Konstante, *VD, *AC
 OUT: VB, EB, AB, MB, SMB, SB, AC, *VD, *AC

Die Schiebeoperationen belegen die Plätze der hinausgeschobenen Bits mit Nullen.

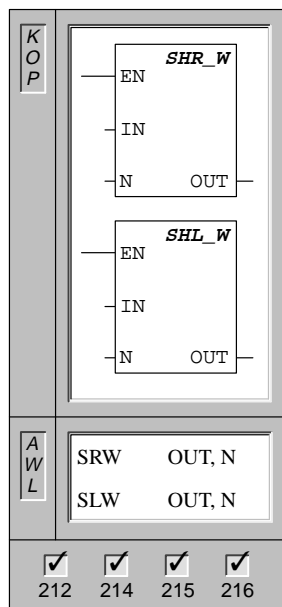
Ist der Schiebewert (N) größer als oder gleich 8, dann wird der Wert maximal 8mal geschoben. Ist der Schiebewert größer als 0, dann nimmt der Überlaufmerker den Wert des zuletzt hinausgeschobenen Bit an.

Die Operationen Byte rechts schieben und Byte links schieben sind vorzeichenlos.

Hinweis: Wenn Sie in KOP programmieren, können Sie angeben, daß IN1 gleich OUT ist. Auf diese Weise sparen Sie Speicherplatz.

Diese Operationen beeinflussen die folgenden Sondermerker: SM1.0 (Null); SM1.1 (Überlauf)

Wort rechts schieben und Wort links schieben



Die Operationen **Wort rechts schieben** und **Wort links schieben** schieben den Wortwert (IN) um den Schiebewert (N) nach rechts bzw. links und laden das Ergebnis in das Ausgangswort (OUT).

Operanden: IN: VW, T, Z, EW, MW, SMW, AC, AW, AEW, Konstante, *VD, *AC, SW
 N: VB, EB, AB, MB, SMB, AC, *VD, *AC, SB
 OUT: VW, T, Z, EW, AW, MW, SMW, AC, *VD, *AC, SW

Die Schiebeoperationen belegen die Plätze der hinausgeschobenen Bits mit Nullen.

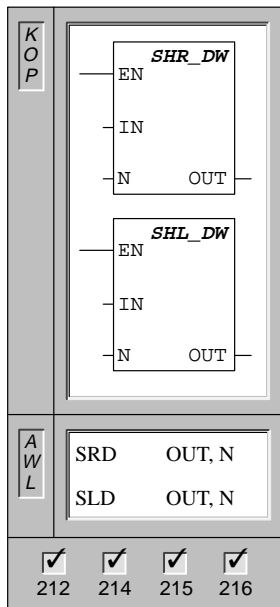
Ist der Schiebewert (N) größer als oder gleich 16, dann wird der Wert maximal 16mal geschoben. Ist die Schiebezahl größer als Null, dann nimmt der Überlaufmerker den Wert des hinausgeschobenen Bit an.

Die Operationen Wort rechts schieben und Wort links schieben sind vorzeichenlos.

Hinweis: Wenn Sie in KOP programmieren, können Sie angeben, daß IN1 gleich OUT ist. Auf diese Weise sparen Sie Speicherplatz.

Diese Operationen beeinflussen die folgenden Sondermerker: SM1.0 (Null); SM1.1 (Überlauf)

Doppelwort rechts schieben und Doppelwort links schieben



Die Operationen **Doppelwort rechts schieben** und **Doppelwort links schieben** schieben den Doppelwortwert (IN) um den Schiebewert (N) nach rechts bzw. links und laden das Ergebnis in das Ausgangsdoppelwort (OUT).

Operanden: IN: VD, ED, AD, MD, SMD, AC, HC, Konstante, *VD, *AC, SD
 N: VB, EB, AB, MB, SMB, AC, *VD, *AC, SB
 OUT: VD, ED, AD, MD, SMD, AC, *VD, *AC, SD

Die Schiebeoperationen belegen die Plätze der hinausgeschobenen Bits mit Nullen.

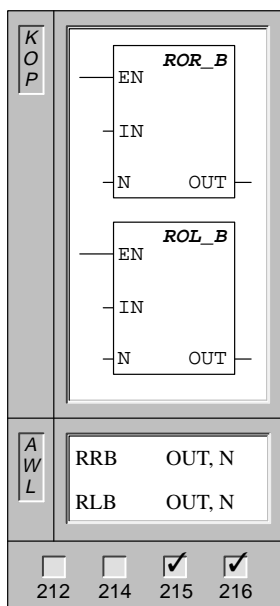
Ist der Schiebewert (N) größer als oder gleich 32, dann wird der Wert maximal 32mal geschoben. Ist der Schiebewert größer als 0, dann nimmt der Überlaufmerker den Wert des zuletzt hinausgeschobenen Bit an.

Die Operationen Doppelwort rechts schieben und Doppelwort links schieben sind vorzeichenlos.

Hinweis: Wenn Sie in KOP programmieren, können Sie angeben, daß IN1 gleich OUT ist. Auf diese Weise sparen Sie Speicherplatz.

Diese Operationen beeinflussen die folgenden Sondermerker: SM1.0 (Null); SM1.1 (Überlauf)

Byte rechts rotieren und Byte links rotieren



Die Operationen **Byte rechts rotieren** und **Byte links rotieren** rotieren den Bytewert (IN) um den Schiebewert (N) nach rechts bzw. links und laden das Ergebnis in das Ausgangsbyte (OUT).

Operanden: IN: VB, EB, AB, MB, SMB, SB, AC, *VD, *AC, SB
 N: VB, EB, AB, MB, SMB, SB, AC, Konstante, *VD, *AC, SB
 OUT: VB, EB, AB, MB, SMB, SB, AC, *VD, *AC, SB

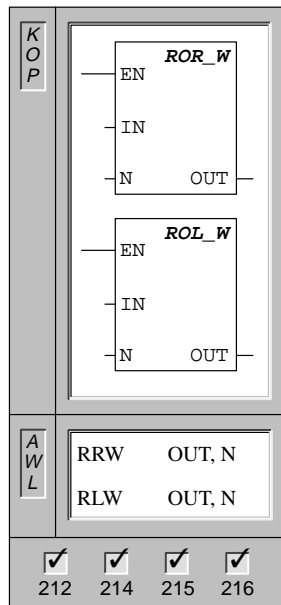
Ist der Schiebewert (N) größer als oder gleich 8, dann wird vor dem Rotieren eine Modulo-8-Operation ausgeführt. Daraus ergibt sich ein Schiebewert von 0 bis 7. Ist die Schiebzahl gleich Null, dann wird nicht rotiert. Wird rotiert, dann wird der Wert des hinausrotierten Bit in den Überlaufmerker kopiert.

Die Operationen Byte rechts rotieren und Byte links rotieren sind vorzeichenlos.

Hinweis: Wenn Sie in KOP programmieren, können Sie angeben, daß IN1 gleich OUT ist. Auf diese Weise sparen Sie Speicherplatz.

Diese Operationen beeinflussen die folgenden Sondermerker: SM1.0 (Null); SM1.1 (Überlauf)

Wort rechts rotieren und Wort links rotieren



Die Operationen **Wort rechts rotieren** und **Wort links rotieren** rotieren den Wortwert (IN) um den Schiebewert (N) nach rechts bzw. links und laden das Ergebnis in das Ausgangswort (OUT).

Operanden: IN: VW, T, Z, EW, MW, SMW, AC, AW, AEW, Konstante, *VD, *AC, SW
 N: VB, EB, AB, MB, SMB, AC, *VD, *AC, SB
 OUT: VW, T, Z, EW, AW, MW, SMW, AC, *VD, *AC, SW

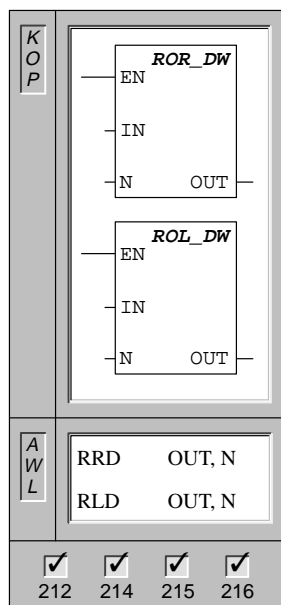
Ist der Schiebewert (N) größer als oder gleich 16, dann wird vor dem Rotieren eine Modulo-16-Operation ausgeführt. Daraus ergibt sich ein Schiebewert von 0 bis 15. Ist die Schiebezahl gleich Null, dann wird nicht rotiert. Wird rotiert, dann wird der Wert des hinausrotierten Bit in den Überlaufmerker kopiert.

Die Operationen Wort rechts rotieren und Wort links rotieren sind vorzeichenlos.

Hinweis: Wenn Sie in KOP programmieren, können Sie angeben, daß IN1 gleich OUT ist. Auf diese Weise sparen Sie Speicherplatz.

Diese Operationen beeinflussen die folgenden Sondermerker: SM1.0 (Null); SM1.1 (Überlauf)

Doppelwort rechts rotieren und Doppelwort links rotieren



Die Operationen **Doppelwort rechts rotieren** und **Doppelwort links rotieren** rotieren den Doppelwortwert (IN) um den Schiebewert (N) nach rechts bzw. links und laden das Ergebnis in das Ausgangsdoppelwort (OUT).

Operanden: IN: VD, ED, AD, MD, SMD, AC, HC, Konstante, *VD, *AC, SD
 N: VB, EB, AB, MB, SMB, AC, *VD, *AC, SB
 OUT: VD, ED, AD, MD, SMD, AC, *VD, *AC, SD

Ist der Schiebewert (N) größer als oder gleich 32, dann wird vor dem Rotieren eine Modulo-32-Operation ausgeführt. Daraus ergibt sich ein Schiebewert von 0 bis 31. Ist die Schiebezahl gleich Null, dann wird nicht rotiert. Wird rotiert, dann wird der Wert des hinausrotierten Bit in den Überlaufmerker kopiert.

Die Operationen Doppelwort rechts rotieren und Doppelwort links rotieren sind vorzeichenlos.

Hinweis: Wenn Sie in KOP programmieren, können Sie angeben, daß IN1 gleich OUT ist. Auf diese Weise sparen Sie Speicherplatz.

Diese Operationen beeinflussen die folgenden Sondermerker: SM1.0 (Null); SM1.1 (Überlauf)

Beispiele für Schiebe- und Rotieroperationen

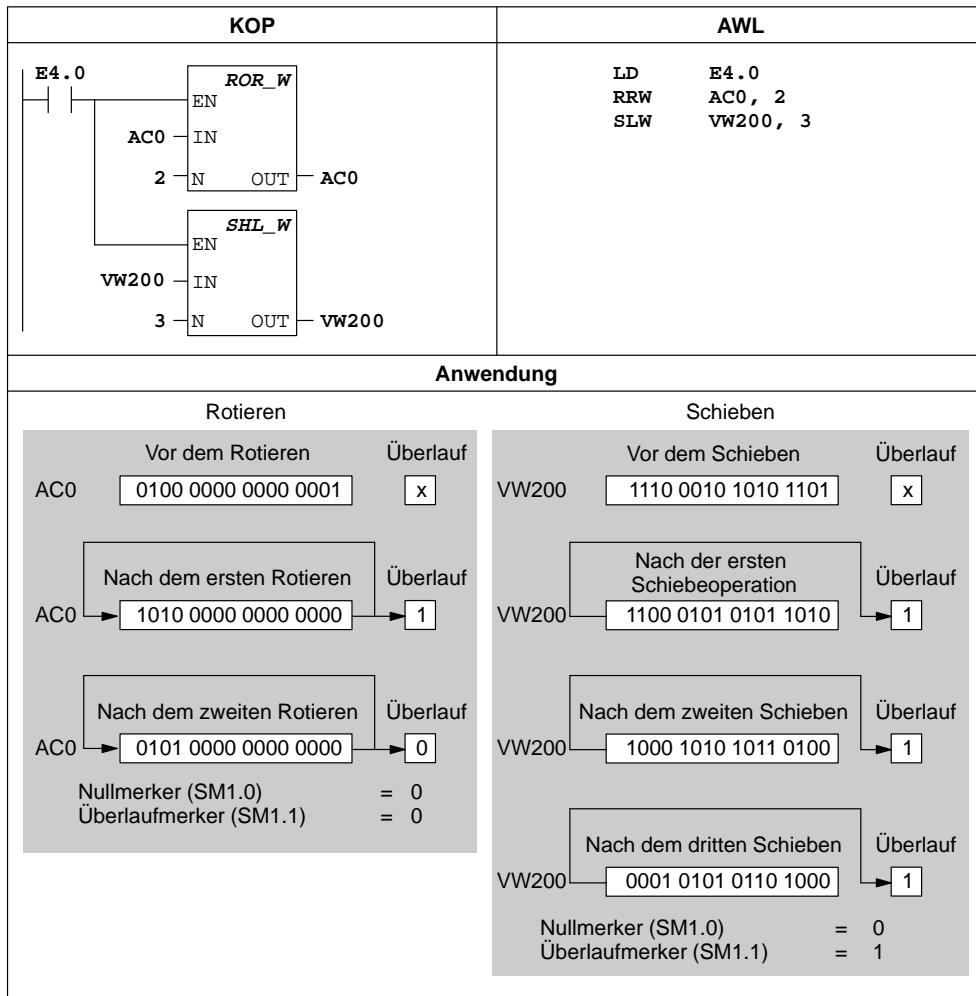


Bild 10-33 Beispiele für Schiebe- und Rotieroperationen in KOP und AWL

10.10 Operationen für die Programmsteuerung

Bearbeitung beenden

K O P	(END)		
	(END)		
A W L	END MEND		
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
212	214	215	216

Die Operation **Bearbeitung bedingt beenden** beendet das Hauptprogramm abhängig von dem Zustand der vorherigen Verknüpfung.

Mit der Spule **Bearbeitung absolut beenden** muß das Hauptprogramm beendet werden.

In AWL wird die Operation Bearbeitung absolut beenden durch die Mnemonik **MEND** dargestellt.

Operanden: keine

Alle Anwenderprogramme müssen das Hauptprogramm mit der Operation Bearbeitung absolut beenden abschließen. Die Operation Bearbeitung bedingt beenden wird verwendet, wenn die Bearbeitung vor der Operation Bearbeitung absolut beenden beendet werden soll.

Hinweis

Sie können die Operationen Bearbeitung bedingt beenden und Bearbeitung absolut beenden im Hauptprogramm verwenden, Sie dürfen sie jedoch nicht in Unterprogrammen und Interruptprogrammen einsetzen.

In STOP gehen

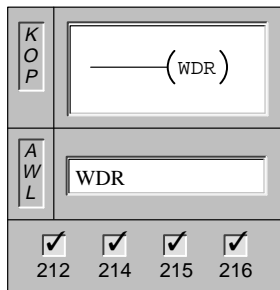
K O P	(STOP)		
A W L	STOP		
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
212	214	215	216

Die Operation **In STOP gehen** beendet die Bearbeitung des Anwenderprogramms sofort, indem sie die CPU aus dem Betriebszustand RUN in den Betriebszustand STOP versetzt.

Operanden: keine

Wird die Operation STOP in einem Interruptprogramm ausgeführt, dann wird dieses sofort beendet und alle anstehenden Interrupts werden ignoriert. Der Rest des Programms wird bearbeitet und die CPU geht am Ende des Zyklus in den Betriebszustand STOP.

Überwachungszeit rücksetzen



Mit der Operation **Überwachungszeit rücksetzen** kann die Überwachungszeit der CPU nachgetriggert werden. Dadurch verlängert sich die maximal zulässige Zykluszeit, ohne daß ein Überwachungszeitfehler gemeldet wird.

Operanden: keine

Richtlinien zum Rücksetzen der Überwachungszeit mit der Operation WDR

Verwenden Sie die Operation Überwachungszeit rücksetzen mit Vorsicht. Wenn Sie mit Programmschleifen die Ausführung eines Zyklus verhindern oder exzessiv verzögern, können die folgenden Prozesse nicht vor Zyklusende ausgeführt werden:

- Kommunikation (ausgenommen frei programmierbare Kommunikation)
- Aktualisieren der Ein- und Ausgänge (ausgenommen beim direkten Ansteuern der Ein- und Ausgänge)
- Aktualisieren der geforderten Werte
- Aktualisieren der Sondermerker (SM0, SM5 bis SM29 werden nicht aktualisiert)
- Diagnosen zur Laufzeit
- Zeiten mit einer Auflösung von 10 ms und 100 ms akkumulieren in Zyklen, die länger sind als 25 Sekunden, den Zeitwert nicht korrekt
- Operation STOP in einem Interruptprogramm

Hinweis

Wenn Sie davon ausgehen, daß die Zykluszeit wahrscheinlich 300 ms überschreiten oder die Interruptaktivität stark ansteigen wird, so daß der Hauptzyklus länger als 300 ms unterbrochen wird, sollten Sie die Überwachungszeit mit der Operation WDR nachtriggern.

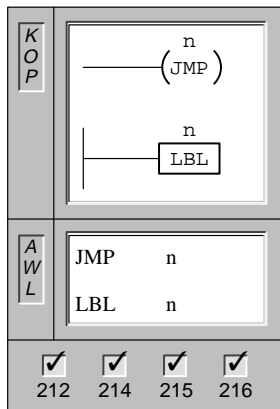
Die CPU geht innerhalb von 1,4 Sekunden in den Betriebszustand STOP, wenn Sie den Schalter der CPU in die Stellung STOP bringen.

Beispiel für die Operationen STOP, END und WDR

KOP	AWL
<p>Netzwerk 1</p> <p>Netzwerk 15</p> <p>Netzwerk 78</p>	<p>Network</p> <p>LD SM5.0</p> <p>STOP</p> <p>.</p> <p>.</p> <p>Network</p> <p>LD M5.6</p> <p>WDR</p> <p>..</p> <p>.</p> <p>Network</p> <p>MEND</p>
<p>Wird ein E/A-Fehler erkannt, dann Übergang in STOP erzwingen.</p>	
<p>Ist M5.6 eingeschaltet, dann die Überwachungszeit nachtriggern (WDR), um die Zykluszeit zu verlängern.</p>	
<p>Hauptprogramm beenden.</p>	

Bild 10-34 Beispiel für die Operationen STOP, END und WDR in KOP und AWL

Zu Sprungmarke springen und Sprungmarke definieren



Die Operation **Zu Sprungmarke springen** verzweigt innerhalb des Programms zu der angegebenen Sprungmarke (n). Wird ein Sprung ausgeführt, ist der oberste Stackwert immer "1".

Die Operation **Sprungmarke definieren** gibt das Ziel (n) an, zu dem gesprungen werden soll.

Operanden: n: 0 bis 255

Die Sprungoperation und die dazugehörige Sprungmarke müssen sich beide entweder im Hauptprogramm, in einem Unterprogramm oder in einem Interruptprogramm befinden. Sie können nicht vom Hauptprogramm aus zu einer Sprungmarke springen, die sich in einem Unterprogramm oder in einem Interruptprogramm befindet. Sie können auch nicht von einem Unterprogramm oder Interruptprogramm zu einer Sprungmarke springen, die außerhalb des jeweiligen Unterprogramms bzw. Interruptprogramms liegt.

Bild 10-35 zeigt ein Beispiel für die Operationen Zu Sprungmarke springen und Sprungmarke definieren.

Beispiel für die Operation Zu Sprungmarke springen

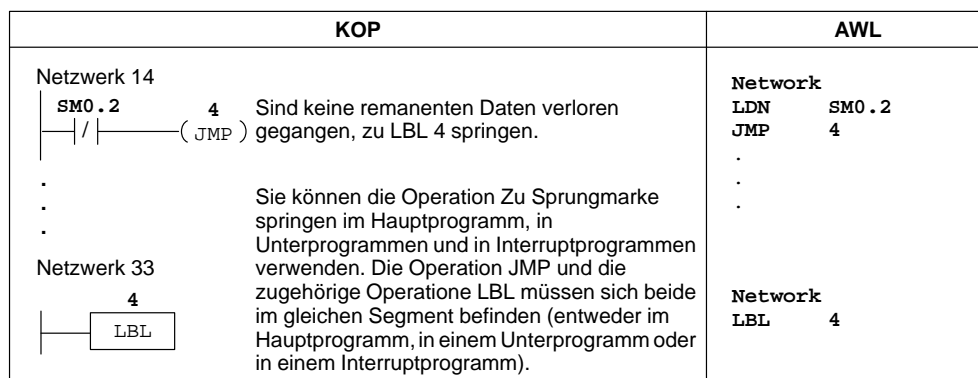
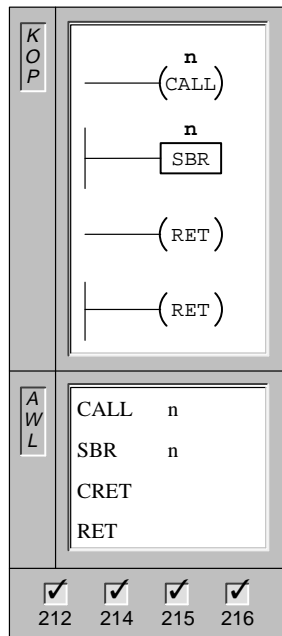


Bild 10-35 Beispiel für die Operationen Zu Sprungmarke springen und Sprungmarke definieren in KOP und AWL

Unterprogramm aufrufen, beginnen und beenden



Die Operation **Unterprogramm aufrufen** ruft ein Unterprogramm (n) auf.

Die Operation **Unterprogramm beginnen** kennzeichnet den Beginn eines Unterprogramms (n).

Die Operation **Unterprogramm bedingt beenden** beendet ein Unterprogramm in Abhängigkeit vom dem Zustand der vorherigen Verknüpfung.

Mit der Operation **Unterprogramm absolut beenden** muß jedes Unterprogramm beendet werden.

Operanden: n: 0 bis 63

Ist die Bearbeitung eines Unterprogramms beendet, dann wird das Hauptprogramm an der Operation weiterbearbeitet, die auf die Operation CALL folgt.

Sie können Unterprogramme in maximal acht Ebenen verschachteln (innerhalb eines Unterprogramms ein weiteres Unterprogramm aufrufen). Rekursion (ein Unterprogramm ruft sich selbst auf) ist zulässig, doch sollten Sie Rekursion in Unterprogrammen nur mit Vorsicht einsetzen.

Beim Aufrufen eines Unterprogramms wird der gesamte Stack gespeichert, der oberste Stackwert wird auf "1" gesetzt, alle weiteren Stackwerte werden auf "0" gesetzt und das aufgerufene Unterprogramm wird bearbeitet. Ist die Bearbeitung des Unterprogramms beendet, wird der Stack mit den Werten, die zum Zeitpunkt des Aufrufs gespeichert waren, wiederhergestellt. Anschließend wird das aufrufende Programm weiterbearbeitet.

Wird ein Unterprogramm aufgerufen, ist der oberste Stackwert immer "1". Deshalb können Sie in dem Netzwerk, das auf die Operation SBR folgt, Ausgänge und Boxen direkt an die linke Stromschiene anschließen. In AWL kann die Ladeoperation, die auf die Operation SBR folgt, entfallen.

Akkumulatoren werden sowohl vom Hauptprogramm als auch von den Unterprogrammen verwendet. Der Aufruf eines Unterprogramms bewirkt nicht, daß die Akkumulatoren gespeichert und später wiederhergestellt werden.

Bild 10-36 zeigt ein Beispiel für Operationen mit Unterprogrammen.

Einschränkungen

Einschränkungen für die Verwendung von Unterprogrammen:

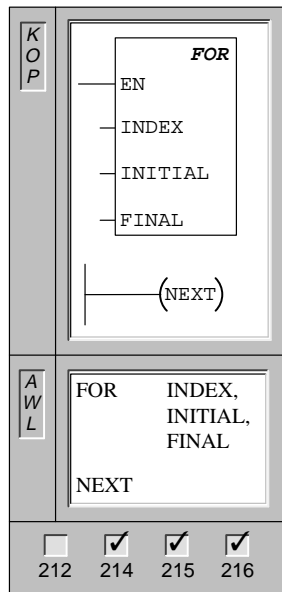
- Ordnen Sie alle Unterprogramme nach dem Ende des Hauptprogramms an.
- Die Operationen LSCR, SCRE, SCRT und END dürfen Sie in Unterprogrammen nicht verwenden.
- Sie müssen jedes Unterprogramm mit der Operation Unterprogramm absolut beenden (RET) beenden.

Beispiel für die Operation Unterprogramm aufrufen

KOP		AWL
Netzwerk 1		Network LD SM0.1 CALL 10 . .
Netzwerk 39		Network MEND . .
Netzwerk 50		Network SBR 10 . .
Netzwerk 65		Network LD M14.3 CRET . .
Netzwerk 68		Network RET

Bild 10-36 Beispiel für Operationen mit Unterprogrammen in KOP und AWL

Programmschleife mit FOR und Ende Programmschleife mit NEXT



Die Operation **Programmschleife mit FOR** führt den Code zwischen FOR und NEXT aus. Sie müssen den aktuellen Zählwert der Programmschleifen (INDEX), den Anfangswert (INITIAL) und den Endwert (FINAL) angeben.

Die Operation **Ende Programmschleife mit NEXT** kennzeichnet das Ende einer Programmschleife, die mit FOR beginnt, und setzt den obersten Stackwert auf "1".

Operanden: INDEX: VW, T, Z, EW, AW, MW, SMW, AC, *VD, *AC, SW

INITIAL: VW, T, Z, EW, AW, MW, SMW, AC, AEW, Konstante, *VD, *AC, SW

FINAL: VW, T, Z, EW, AW, MW, SMW, AC, AEW, Konstante, *VD, *AC, SW

Beträgt beispielsweise der Wert von INITIAL 1 und der Wert von FINAL 10, dann werden die Operationen, die sich zwischen FOR und NEXT befinden, zehnmal ausgeführt, wobei sich der Zählwert INDEX jeweils um 1 erhöht: 1, 2, 3, ... 10.

Ist der Anfangswert größer als der Endwert, wird die Schleife nicht ausgeführt. Nach jeder Ausführung der Operationen zwischen FOR und NEXT wird der Wert von INDEX um "1" erhöht und das Ergebnis mit dem Endwert verglichen. Ist INDEX größer als der Endwert, wird die Schleife beendet.

Mit den Operationen FOR und NEXT können Sie Programmschleifen steuern, die für einen bestimmten Zählwert wiederholt werden. Jede Operation FOR benötigt eine Operation NEXT. Sie können Programmschleifen mit FOR/NEXT bis zu einer Tiefe von acht Ebenen verschachteln (eine Programmschleife mit FOR/NEXT innerhalb einer anderen Programmschleife mit FOR/NEXT).

Bild 10-37 zeigt ein Beispiel für die Operationen Programmschleife mit FOR und Ende Programmschleife mit NEXT.

Beispiel für die Operationen FOR und NEXT

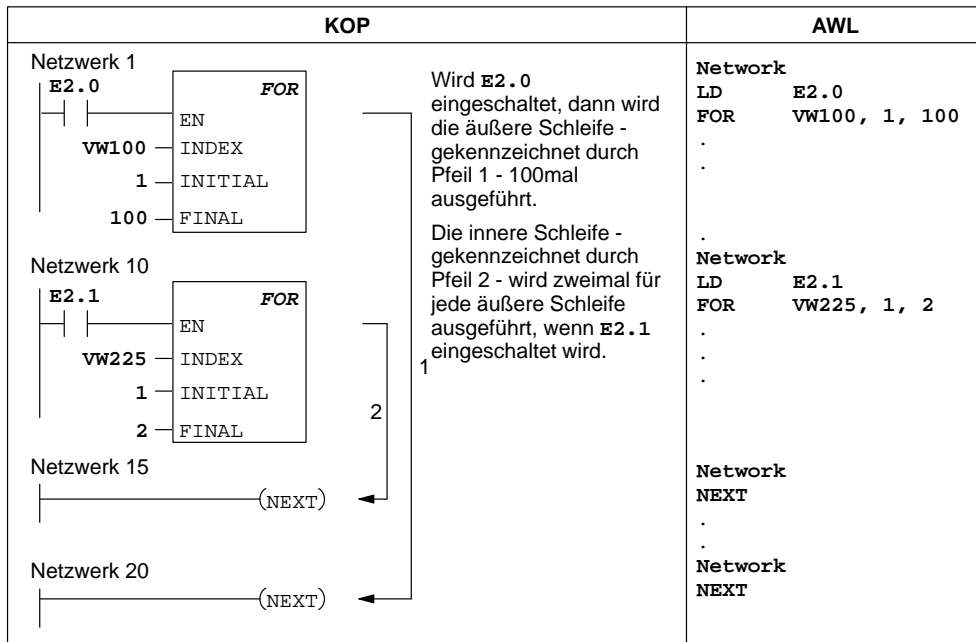
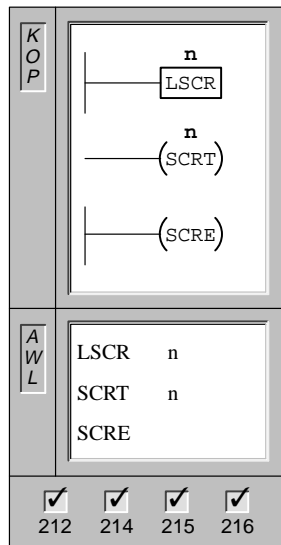


Bild 10-37 Beispiel für die Operationen FOR und NEXT in KOP und AWL

Operationen für das Ablaufsteuerungsrelais



Die Operation **Ablaufsteuerungsrelais laden** kennzeichnet den Beginn eines SCR-Segments. Ist $n = 1$, wird der Signalfluß zum SCR-Segment freigegeben. Das SCR-Segment muß mit einer Operation SCRE beendet werden.

Die Operation **Flanke Ablaufsteuerungsrelais** kennzeichnet das SCR-Bit, das freigegeben werden soll (das nächste S-Bit, das gesetzt werden soll). Fließt Strom zur Spule, dann wird das angegebene S-Bit eingeschaltet und das S-Bit der Operation LSCR (das dieses SCR-Segment freigegeben hat) wird ausgeschaltet.

Die Operation **Ende Ablaufsteuerungsrelais** kennzeichnet das Ende eines SCR-Segments.

Operanden: n: S

Beschreibung der Operationen für das Ablaufsteuerungsrelais

Im Kontaktplan und in der Anweisungsliste unterteilen Ablaufsteuerungsrelais (SCRs) Operationen oder Schritte einer Anlage in äquivalente Programmsegmente. Durch Ablaufsteuerungsrelais wird das Steuerprogramm in logischen Segmenten strukturiert.

Die Operation LSCR lädt den Wert des S-Bit, das von der Operation angegeben wird, in den Stack des Ablaufsteuerungsrelais und in den logischen Stack. Das SCR-Segment wird durch das Ergebnis des SCR-Stack aktiviert bzw. deaktiviert. Der oberste Stackwert wird in das angegebene S-Bit geladen, so daß die Boxen und Spulen ohne zwischengeschalteten Kontakt direkt an die linke Stromschiene angeschlossen werden können. Bild 10-38 zeigt den S-Stack und den logischen Stack und die Auswirkungen der Operation LSCR.

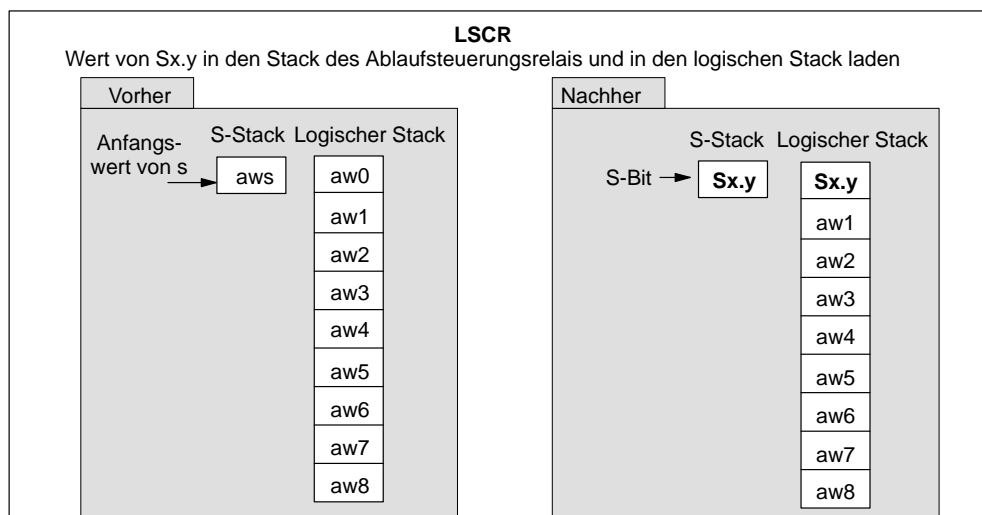


Bild 10-38 Auswirkungen der Operation LSCR auf den logischen Stack

Beachten Sie bei den Operationen für das Ablaufsteuerungsrelais folgende Hinweise:

- Alle Operationen zwischen der Operation LSCR und der Operation SCRE bilden das SCR-Segment und sind hinsichtlich der Ausführung von dem Wert des S-Stack abhängig. Die Logik zwischen der Operation SCRE und der nächsten Operation LSCR richtet sich nicht nach dem Wert des S-Stack.
- Die Operation SCRT setzt ein S-Bit, das das nächste Ablaufsteuerungsrelais freigibt. Dieses S-Bit setzt außerdem das S-Bit zurück, das geladen wurde, um diesen Abschnitt des SCR-Segments freizugeben.

Einschränkungen

Einschränkungen für die Verwendung von Ablaufsteuerungsrelais:

- Sie können Ablaufsteuerungsrelais im Hauptprogramm verwenden, Sie dürfen sie jedoch nicht in Unterprogrammen und Interruptprogrammen einsetzen.
- Die Operationen JMP und LBL dürfen Sie nicht in SCR-Segmenten anordnen. Dies bedeutet, daß Sprünge in ein SCR-Segment, innerhalb eines SCR-Segments und aus einem SCR-Segment heraus unzulässig sind. Sie können mit den Sprungoperationen SCR-Segmente überspringen.
- Die Operationen FOR, NEXT und END dürfen Sie nicht in SCR-Segmenten verwenden.

Beispiel für Ablaufsteuerungsrelais

Bild 10-39 zeigt ein Beispiel für die Funktionsweise von Ablaufsteuerungsrelais.

- In diesem Beispiel wird mit dem Sondermerker SM0.1 (Merker des ersten Zyklus) S0.1 gesetzt. S0.1 ist im ersten Zyklus im aktiven Schritt 1.
- Nach einer Verzögerung von 2 Sekunden ruft T37 eine Weiterschaltung zu Schritt 2 hervor. Diese Weiterschaltung deaktiviert das SCR-Segment für Schritt 1 (S0.1) und aktiviert das SCR-Segment für Schritt 2 (S0.2).

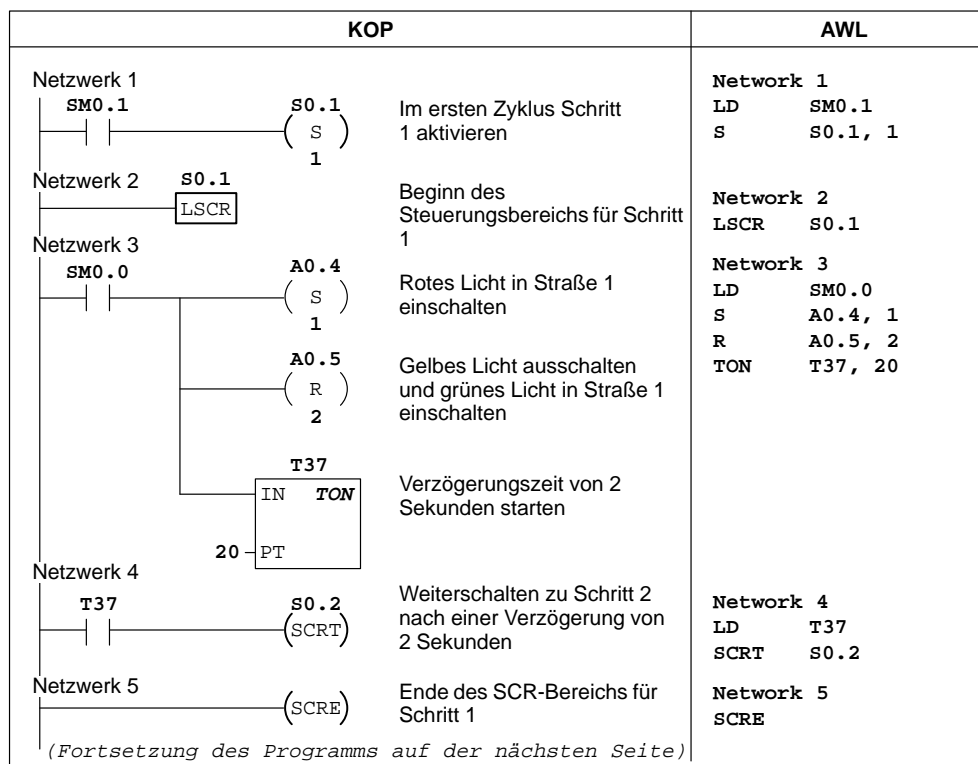


Bild 10-39 Beispiel für Ablaufsteuerungsrelais (SCR)

KOP	AWL
(Fortsetzung des Programms von der vorherigen Seite)	
Netzwerk 6 : I0.2 	Beginn des Steuerungsbereichs für Schritt 2 Network 6 LSCR S0.2
Netzwerk 7 	Grünes Licht in Straße 3 einschalten Network 7 LD SM0.0 S A0.2, 1 TON T38, 250
Netzwerk 8 	Weitschalten zu Schritt 3 nach einer Verzögerung von 25 Sekunden Network 8 LD T38 SCRT S0.3
Netzwerk 9 	Ende des SCR-Bereichs für Schritt 2 Network 9 SCRE . . .

Bild 10-39 Beispiel für Ablaufsteuerungsrelais (SCR), Fortsetzung

Teilung von Ablaufketten

In vielen Anwendungen ist es erforderlich, eine einzige Ablaufkette in zwei oder mehrere getrennte Ablaufketten zu unterteilen. Wird eine Ablaufkette in mehrere Ablaufketten unterteilt, müssen alle neu beginnenden Ablaufketten gleichzeitig aktiviert werden. Dieser Vorgang wird in Bild 10-40 gezeigt.

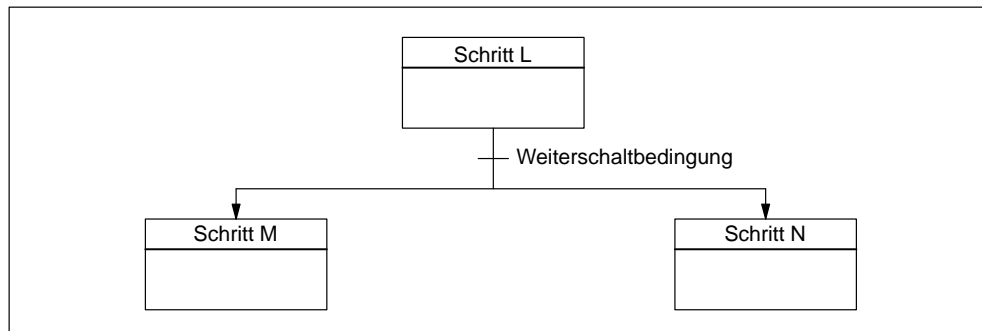


Bild 10-40 Teilung einer Ablaufkette

Die Teilung von Ablaufketten kann in ein SCR-Programm implementiert werden, indem mehrere Operationen SCRT durch die gleiche Weiterschaltbedingung aktiviert werden (siehe Bild 10-41).

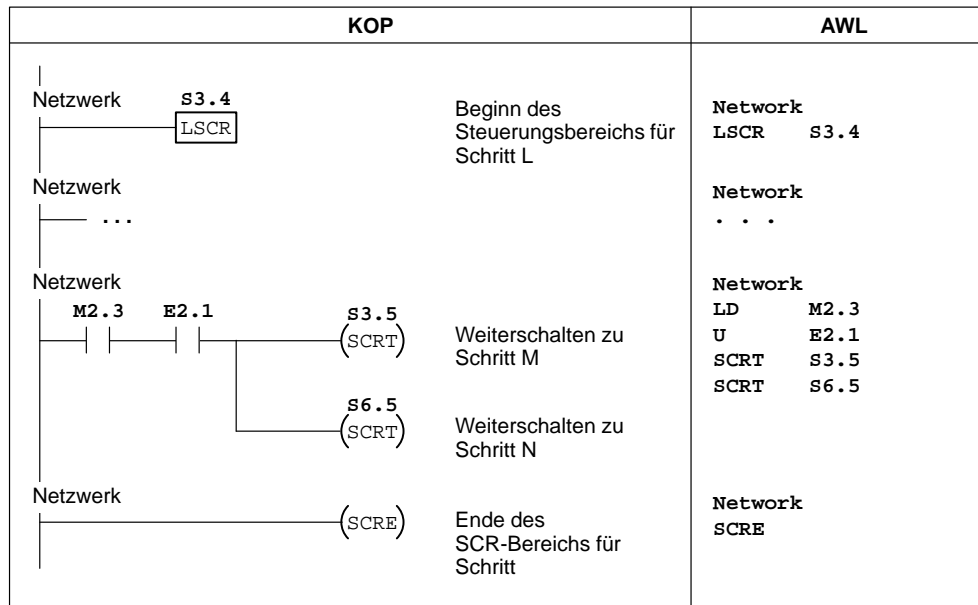


Bild 10-41 Beispiel für die Teilung einer Ablaufkette

Zusammenführung von Ablaufketten

Eine ähnliche Situation entsteht, wenn zwei oder mehrere Ablaufketten zu einer Ablaufkette verbunden werden sollen. Wenn mehrere Ablaufketten in einer Ablaufkette enden, nennt man dies Zusammenführung. Bei der Zusammenführung von Ablaufketten müssen alle Ablaufketten beendet sein, bevor der nächste Schritt ausgeführt werden kann. Bild 10-42 zeigt die Zusammenführung von zwei Ablaufketten.

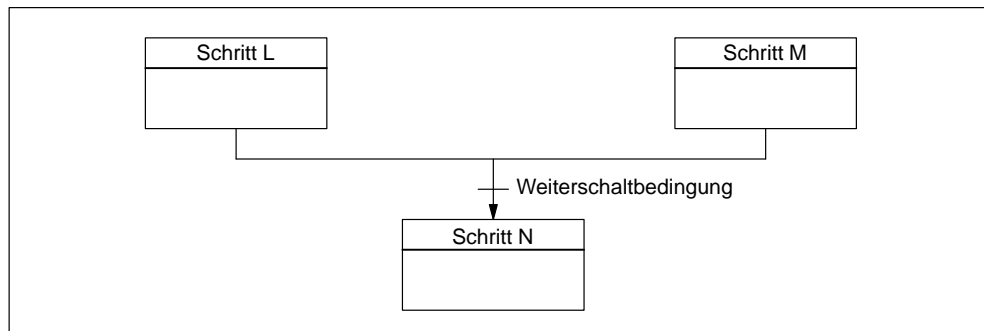


Bild 10-42 Zusammenführung von Ablaufketten

Die Zusammenführung von Ablaufketten kann in einem SCR-Programm implementiert werden, indem von Schritt L zu Schritt L' und von Schritt M zu Schritt M' weiterschaltet wird. Sind beide SCR-Bits, die L' und M' darstellen, wahr, dann kann der Schritt N aktiviert werden (siehe folgendes Beispiel).

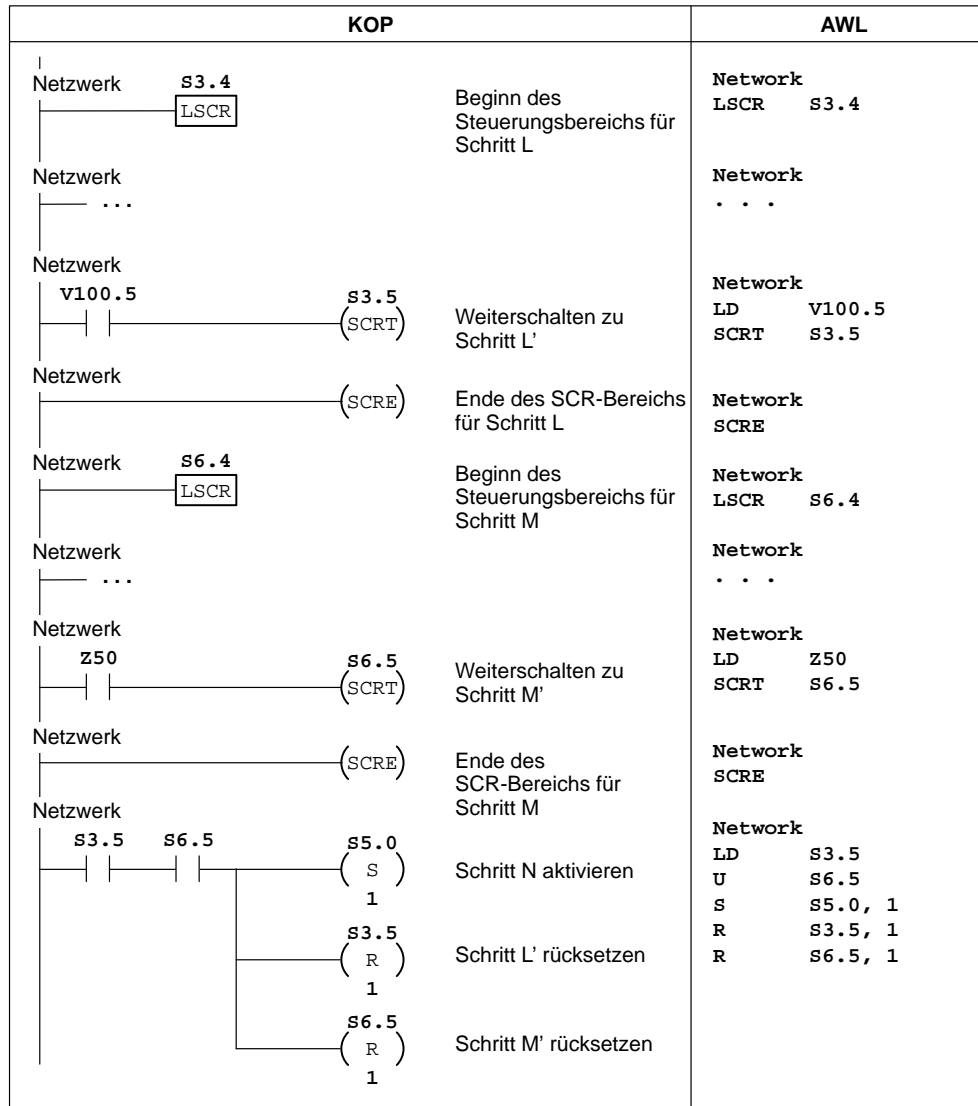


Bild 10-43 Beispiel für die Zusammenführung von Ablaufketten

In anderen Situationen kann eine Ablaufkette zu einer von mehreren möglichen Ablaufketten umgeleitet werden. Dies richtet sich danach, welche Weiterschaltbedingung als erste wahr wird. Dieser Fall wird in Bild 10-44 dargestellt.

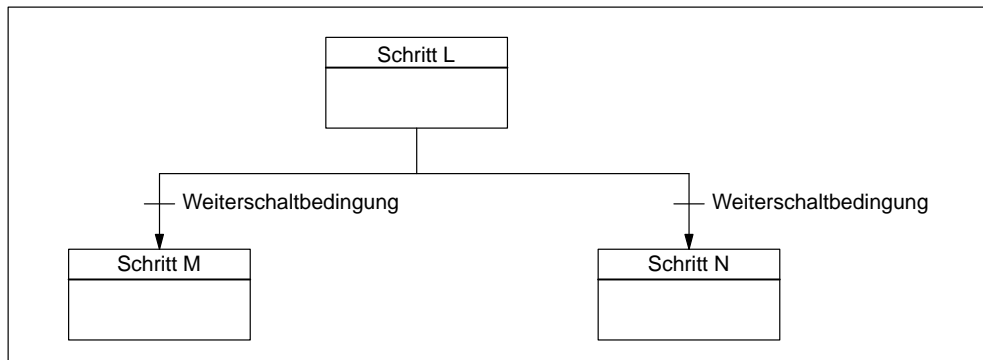


Bild 10-44 Umlenkung der Ablaufkette je nach Weiterschaltbedingung

Hierzu sehen Sie ein SCR-Programm in Bild 10-45.

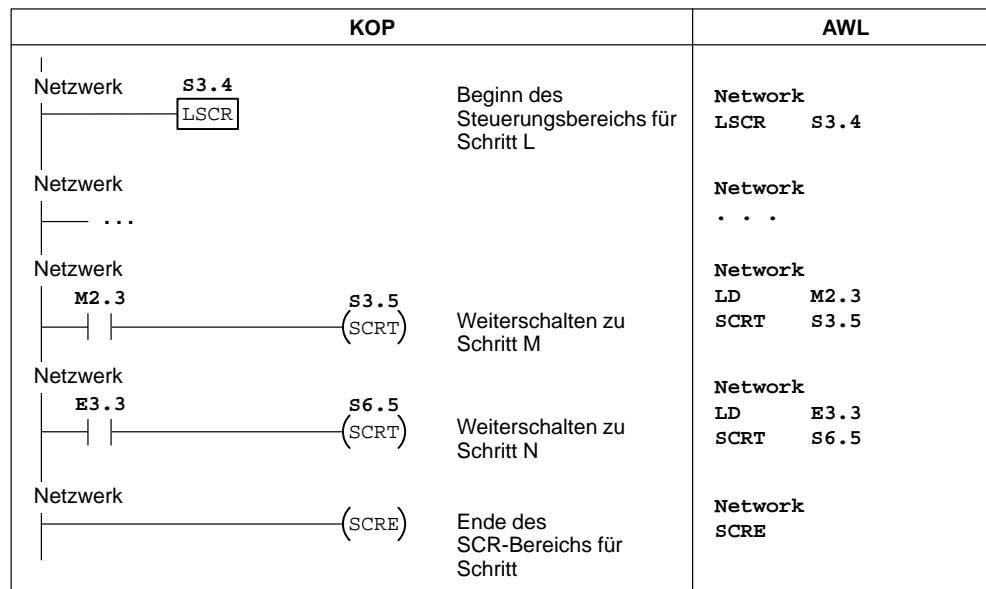
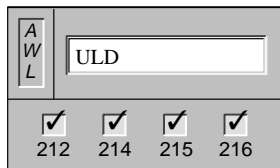


Bild 10-45 Beispiel für Weiterschaltbedingungen

10.11 Stackoperationen

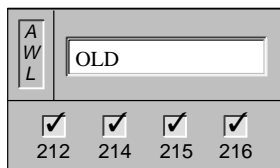
Erste und zweite Stackebene durch UND verknüpfen



Die Operation **Erste und zweite Stackebene durch UND verknüpfen** verknüpft die Werte der ersten (obersten) und der zweiten Ebene des Stack durch UND. Das Ergebnis wird in die Spitze des Stack geladen. Nach Ausführung der Operation ULD enthält der Stack ein Bit weniger.

Operanden: keine

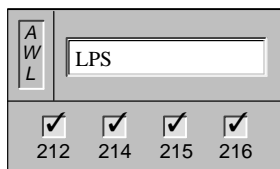
Erste und zweite Stackebene durch ODER verknüpfen



Die Operation **Erste und zweite Stackebene durch ODER verknüpfen** verknüpft die Werte der ersten (obersten) und der zweiten Ebene des Stack durch ODER. Das Ergebnis wird in die Spitze des Stack geladen. Nach Ausführung der Operation OLD enthält der Stack ein Bit weniger.

Operanden: keine

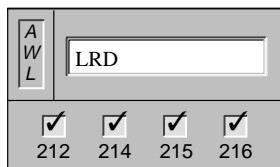
Obersten Stackwert duplizieren



Die Operation **Obersten Stackwert duplizieren** dupliziert den obersten Stackwert und schiebt ihn in den Stack. Der unterste Stackwert wird aus dem Stack geschoben und geht verloren.

Operanden: keine

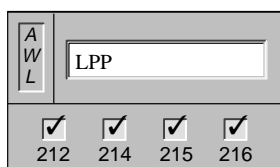
Zweiten Stackwert kopieren



Die Operation **Zweiten Stackwert kopieren** kopiert den zweiten Stackwert in die Spitze des Stack. Es wird kein Wert in den Stack geladen und auch kein Wert aus dem Stack geschoben. Der vorherige oberste Stackwert wird mit dem neuen Wert überschrieben.

Operanden: keine

Obersten Stackwert aus Stack schieben



Die Operation **Obersten Stackwert aus Stack schieben** schiebt den obersten Wert aus dem Stack. Der zweite Stackwert wird in die Spitze des Stack geschoben.

Operanden: keine

Stackoperationen

Bild 10-46 zeigt die Funktionsweise der Operationen Erste und zweite Stackebene durch UND verknüpfen und Erste und zweite Stackebene durch ODER verknüpfen.

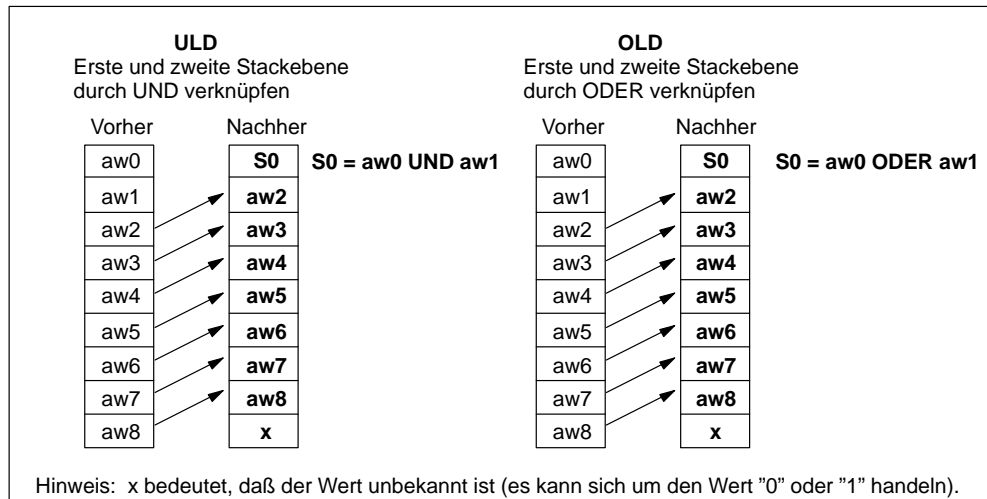


Bild 10-46 Stackoperationen ULD und OLD

Bild 10-47 zeigt die Funktionsweise der Operationen Obersten Stackwert duplizieren, Zweiten Stackwert kopieren und Obersten Stackwert aus Stack schieben.

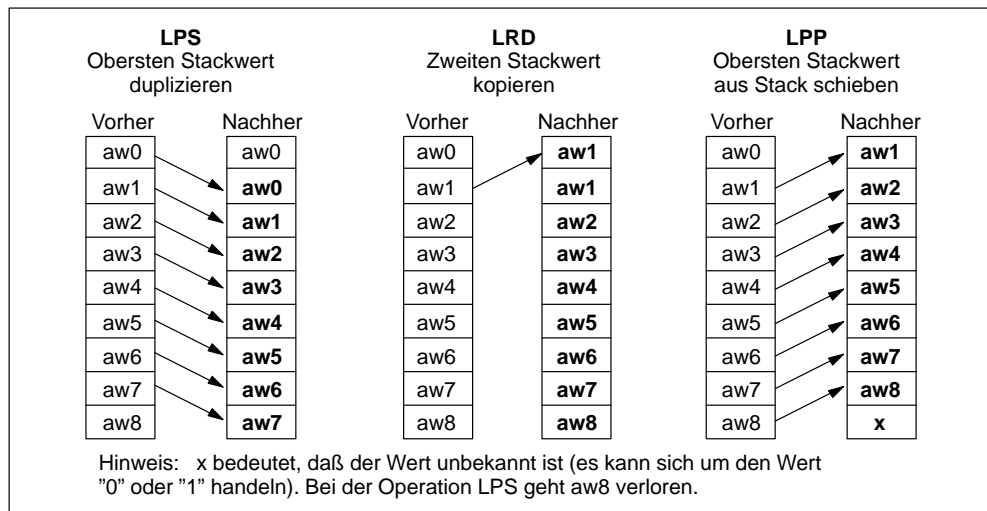


Bild 10-47 Stackoperationen LPS, LRD und LPP

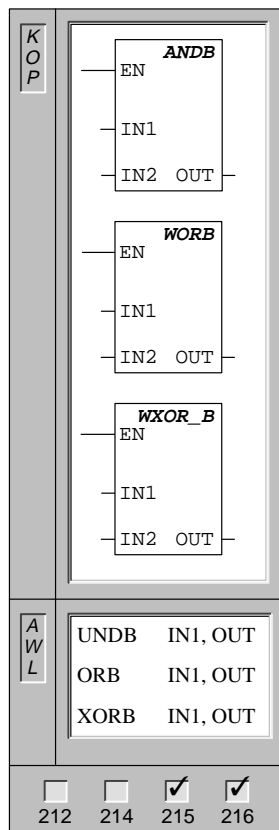
Beispiel für Stackoperationen

KOP	AWL
<p>Netzwerk 1</p> <p>Netzwerk 2</p>	<pre> NETWORK LD E0.0 LD E0.1 LD E2.0 U E2.1 OLD ULD = A5.0 NETWORK LD E0.0 LPS LD E0.5 O E0.6 ULD = A7.0 LRD LD E2.1 O E1.3 ULD = A6.0 LPP U E1.0 = A3.0 </pre>

Bild 10-48 Beispiel für Stackoperationen in KOP und AWL

10.12 Verknüpfungsoperationen

Bytes durch UND, ODER und EXKLUSIV ODER verknüpfen



Die Operation **Bytes durch UND verknüpfen** verknüpft die entsprechenden Bits der beiden Bytes IN1 und OUT durch UND und lädt das Ergebnis in das Byte OUT.

Die Operation **Bytes durch ODER verknüpfen** verknüpft die entsprechenden Bits der beiden Bytes IN1 und OUT durch ODER und lädt das Ergebnis in das Byte OUT.

Die Operation **Bytes durch EXKLUSIV ODER verknüpfen** verknüpft die entsprechenden Bits der beiden Bytes IN1 und OUT durch EXKLUSIV ODER und lädt das Ergebnis in das Byte OUT.

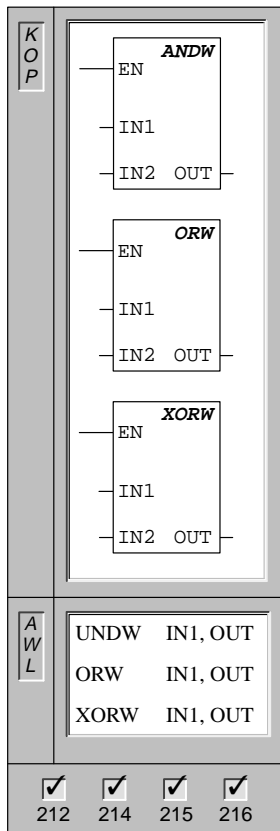
Operanden: IN1, IN2: VB, EB, AB, MB, SMB, AC, *VD, *AC, SB

OUT: VB, EB, AB, MB, SMB, SB, AC, *VD, *AC, SB

Hinweis: Wenn Sie in KOP programmieren, können Sie angeben, daß IN1 gleich OUT ist. Auf diese Weise sparen Sie Speicherplatz.

Diese Operationen beeinflussen die folgenden Sondermerker: SM1.0 (Null)

Wörter durch UND, ODER und EXKLUSIV ODER verknüpfen



Die Operation **Wörter durch UND verknüpfen** verknüpft die entsprechenden Bits zweier Wörter (IN1, OUT) durch UND und lädt das Ergebnis in das Wort OUT.

Die Operation **Wörter durch ODER verknüpfen** verknüpft die entsprechenden Bits zweier Wörter (IN1, OUT) durch ODER und lädt das Ergebnis in das Wort OUT.

Die Operation **Wörter durch EXKLUSIV ODER verknüpfen** verknüpft die entsprechenden Bits zweier Wörter (IN1, OUT) durch EXKLUSIV ODER und lädt das Ergebnis in das Wort OUT.

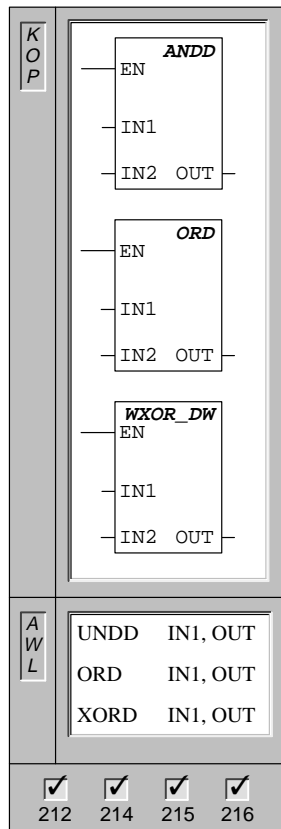
Operanden: IN1, IN2: VW, T, Z, EW, AW, MW, SMW, AC, AEW, Konstante, *VD, *AC, SW

 OUT: VW, T, Z, EW, AW, MW, SMW, AC, *VD, *AC, SW

Hinweis: Wenn Sie in KOP programmieren, können Sie angeben, daß IN1 gleich OUT ist. Auf diese Weise sparen Sie Speicherplatz.

Diese Operationen beeinflussen die folgenden Sondermerker: SM1.0 (Null)

Doppelwörter durch UND, ODER und EXKLUSIV ODER verknüpfen



Die Operation **Doppelwörter durch UND verknüpfen** verknüpft die entsprechenden Bits zweier Doppelwörter (IN1, OUT) durch UND und lädt das Ergebnis in das Wort OUT.

Die Operation **Doppelwörter durch ODER verknüpfen** verknüpft die entsprechenden Bits zweier Doppelwörter (IN1, OUT) durch ODER und lädt das Ergebnis in das Wort OUT.

Die Operation **Doppelwörter durch EXKLUSIV ODER verknüpfen** verknüpft die entsprechenden Bits zweier Doppelwörter (IN1, OUT) durch EXKLUSIV ODER und lädt das Ergebnis in das Wort OUT.

Operanden: IN1, IN2: VD, ED, AD, MD, SMD, AC, HC,
Konstante, *VD, *AC, SD

OUT: VD, ED, AD, MD, SMD, AC, *VD,
*AC, SD

Hinweis: Wenn Sie in KOP programmieren, können Sie angeben, daß IN1 gleich OUT ist. Auf diese Weise sparen Sie Speicherplatz.

Diese Operationen beeinflussen die folgenden Sondermerker:
SM1.0 (Null)

Beispiel für Verknüpfungsoperationen

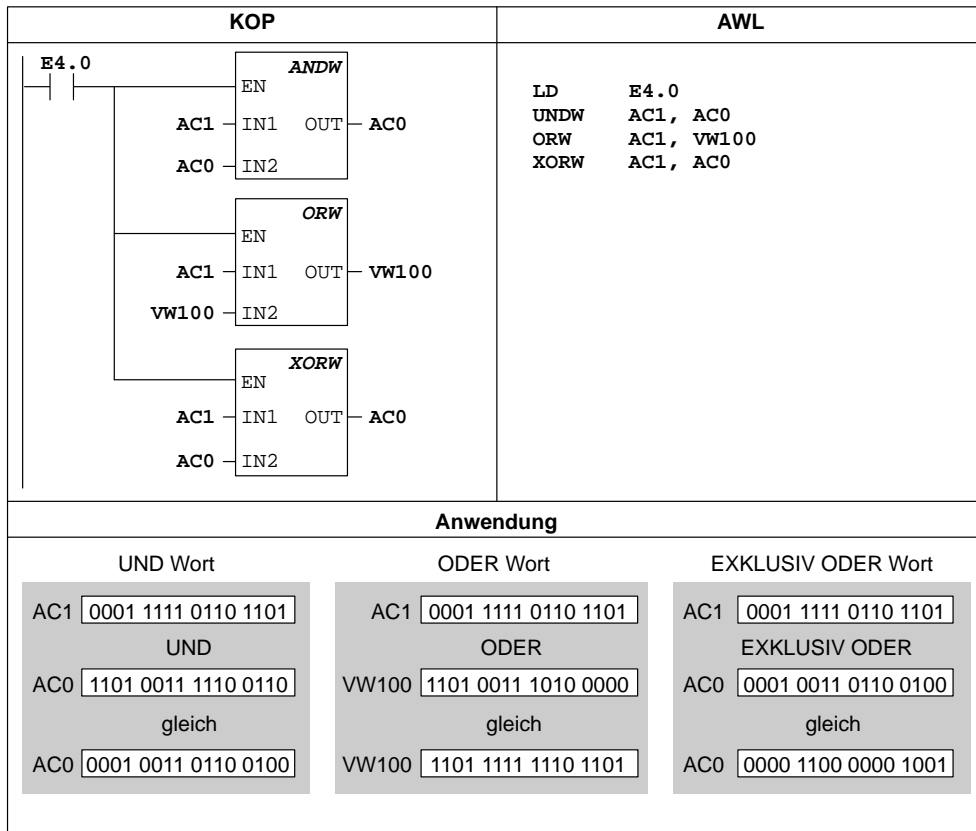
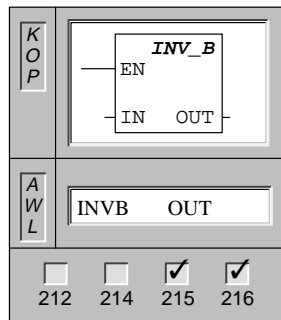


Bild 10-49 Beispiel für Verknüpfungsoperationen

Einerkomplement von Byte erzeugen



Die Operation **Einerkomplement von Byte erzeugen** bildet das Einerkomplement vom Wert des Eingangsbytes und lädt das Ergebnis in das Byte OUT.

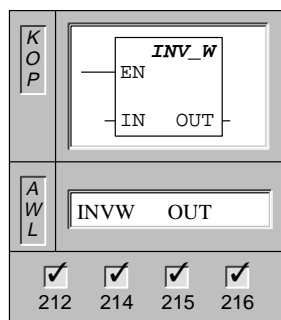
Operanden: IN: VB, EB, AB, MB, SMB, SB, AC, *VD, *AC, SB
 OUT: VB, EB, AB, MB, SMB, SB, AC, *VD, *AC, SB

Hinweis: Wenn Sie in KOP programmieren, können Sie angeben, daß IN1 gleich OUT ist. Auf diese Weise sparen Sie Speicherplatz.

Diese Operation beeinflusst die folgenden Sondermerker:

SM1.0 (Null)

Einerkomplement von ganzer Zahl (16 Bit) erzeugen



Die Operation **Einerkomplement von ganzer Zahl (16 Bit) erzeugen** bildet das Einerkomplement vom Wert des Eingangsworts und lädt das Ergebnis in das Wort OUT.

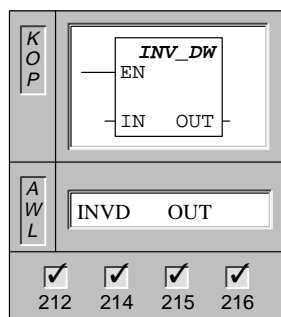
Operanden: IN: VW, T, Z, EW, AW, MW, SMW, AC, AEW, Konstante, *VD, *AC, SW
 OUT: VW, T, Z, EW, AW, MW, SMW, AC, *VD, *AC, SW

Hinweis: Wenn Sie in KOP programmieren, können Sie angeben, daß IN1 gleich OUT ist. Auf diese Weise sparen Sie Speicherplatz.

Diese Operation beeinflusst die folgenden Sondermerker:

SM1.0 (Null)

Einerkomplement von ganzer Zahl (32 Bit) erzeugen



Die Operation **Einerkomplement von ganzer Zahl (32 Bit) erzeugen** bildet das Einerkomplement vom Wert des Eingangsdoppelworts und lädt das Ergebnis in das Doppelwort OUT.

Operanden: IN: VD, ED, AD, MD, SMD, AC, HC, Konstante, *VD, *AC, SD
 OUT: VD, ED, AD, MD, SMD, AC, *VD, *AC, SD

Hinweis: Wenn Sie in KOP programmieren, können Sie angeben, daß IN1 gleich OUT ist. Auf diese Weise sparen Sie Speicherplatz.

Diese Operation beeinflusst die folgenden Sondermerker:

SM1.0 (Null)

Beispiel für die Operation Einerkomplement

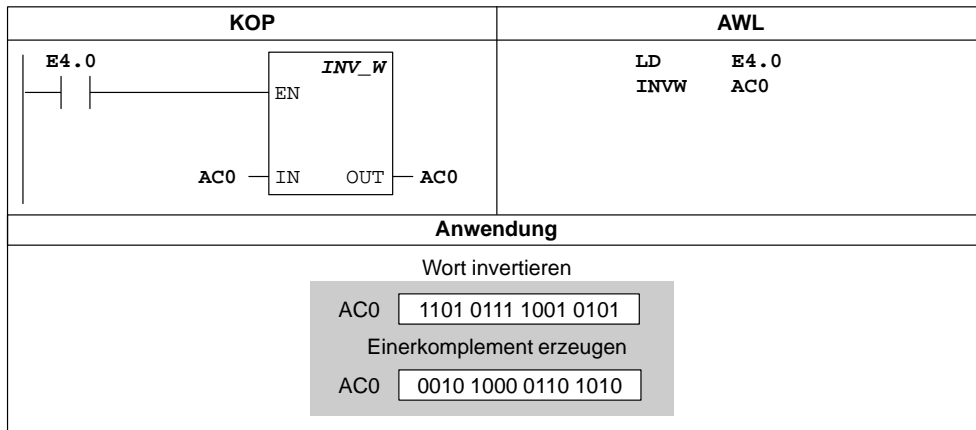
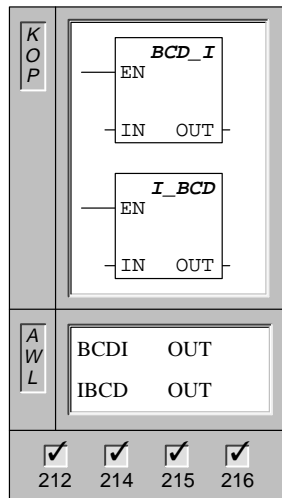


Bild 10-50 Beispiel für die Operation Einerkomplement erzeugen in KOP und AWL

10.13 Umwandlungsoperationen

BCD in ganze Zahl wandeln und Ganze Zahl in BCD wandeln



Die Operation **BCD in ganze Zahl wandeln** wandelt einen binär-codierten Dezimalwert (IN) in einen ganzzahligen Wert um und lädt das Ergebnis in OUT.

Die Operation **Ganze Zahl in BCD wandeln** wandelt einen ganzzahligen Wert (IN) in einen binär-codierten Dezimalwert um und lädt das Ergebnis in OUT.

Operanden: IN: VW, T, Z, EW, AW, MW, SMW, AC, AEW, Konstante, *VD, *AC, SW

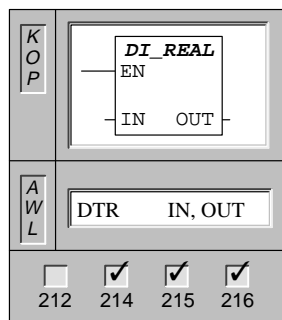
OUT: VW, T, Z, EW, AW, MW, SMW, AC, *VD, *AC, SW

Hinweis: Wenn Sie in KOP programmieren, können Sie angeben, daß IN gleich OUT ist. Auf diese Weise sparen Sie Speicherplatz.

Diese Operationen beeinflussen die folgenden Sondermerker:

SM1.6 (ungültiger BCD-Wert)

Ganze Zahl (32 Bit) in Realzahl wandeln

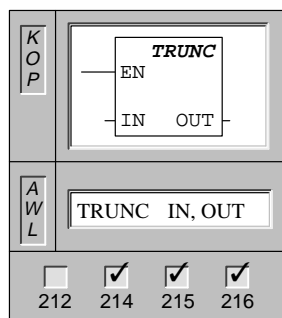


Die Operation **Ganze Zahl (32 Bit) in Realzahl wandeln** wandelt eine ganze Zahl (32 Bit) mit Vorzeichen (IN) in eine Realzahl (32 Bit) um (OUT).

Operanden: IN: VD, ED, AD, MD, SMD, AC, HC, Konstante, *VD, *AC, SD

OUT: VD, ED, AD, MD, SMD, AC, *VD, *AC, SD

Realzahl in ganze Zahl (32 Bit) wandeln



Die Operation **Realzahl in ganze Zahl (32 Bit) wandeln** wandelt eine Realzahl (32 Bit) (IN) in eine ganze Zahl (32 Bit) mit Vorzeichen (OUT) um. Nur der ganzzahlige Teil der Realzahl wird umgewandelt (auf Null runden).

Operanden: IN: VD, ED, AD, MD, SMD, AC, Konstante, *VD, *AC, SD

OUT: VD, ED, AD, MD, SMD, AC, *VD, *AC, SD

Diese Operation beeinflusst die folgenden Sondermerker:

SM1.1 (Überlauf)

Beispiel für die Umwandlung einer Realzahl

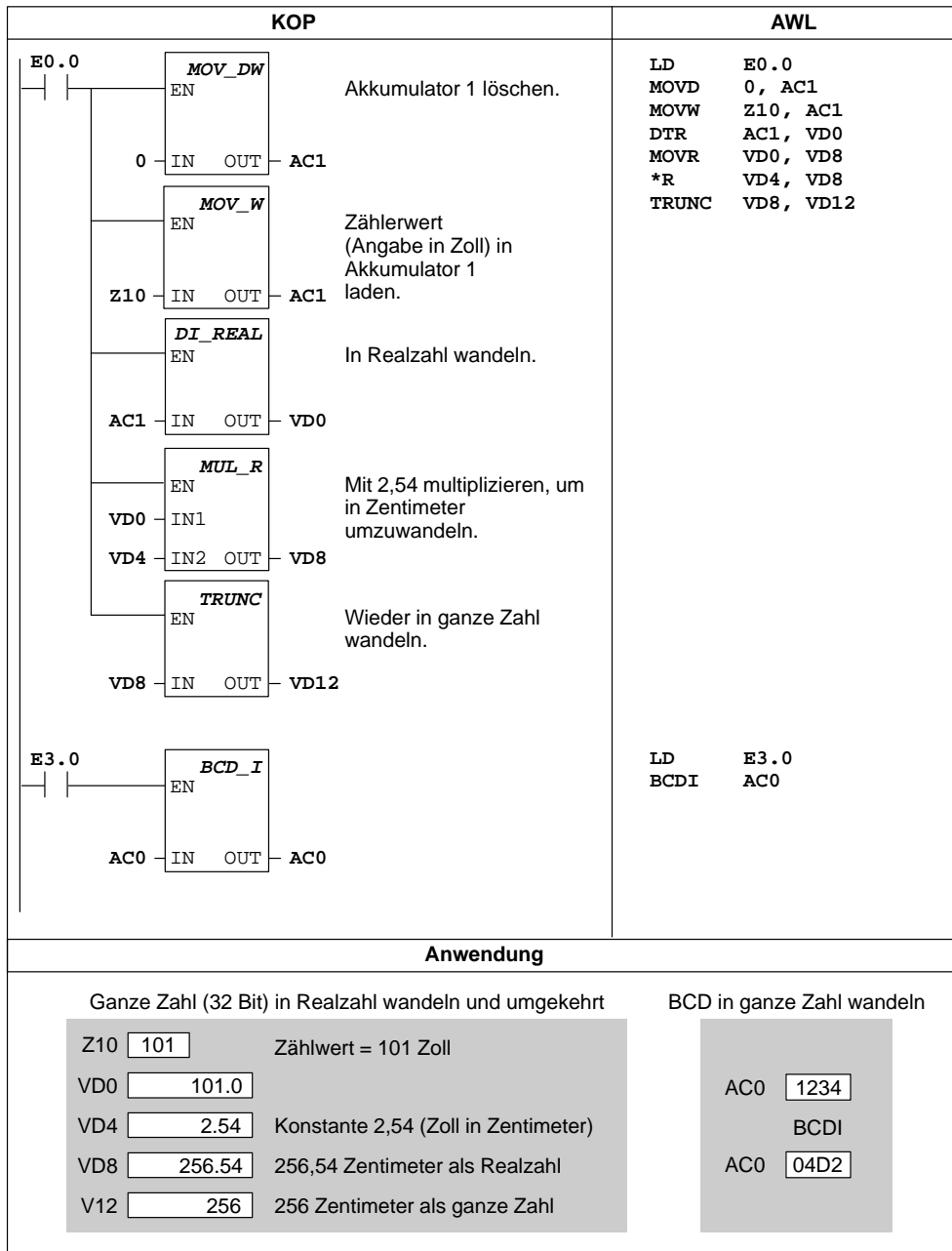
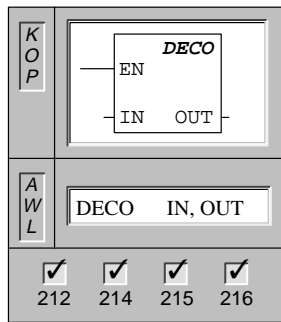


Bild 10-51 Beispiel für die Umwandlung einer Realzahl

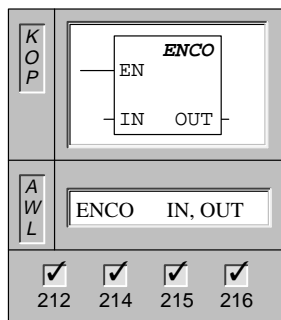
Bit in Hexadezimalzahl wandeln



Die Operation **Bit in Hexadezimalzahl wandeln** setzt das Bit im Ausgangswort (OUT), das der Bitnummer entspricht, die durch das niederwertigste Halb-Byte (4 Bit) des Eingangsbytes (IN) dargestellt wird. Die übrigen Bits des Ausgangsworts werden auf "0" gesetzt.

Operanden: IN: VB, EB, AB, MB, SMB, AC, *VD, *AC, SB
 OUT: VW, T, Z, EW, AW, MW, SMW, AC, AAW, *VD, *AC, SW

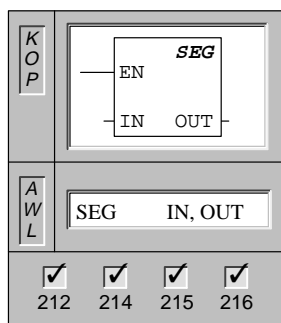
Hexadezimalzahl in Bit wandeln



Die Operation **Hexadezimalzahl in Bit wandeln** schreibt die Bitnummer des niederwertigsten Bits im Eingangswort (IN) in das niederwertigste Halb-Byte (4 Bit) des Ausgangsbytes (OUT).

Operanden: IN: VW, T, Z, EW, AW, MW, SMW, AC, AEW, Konstante, *VD, *AC, SW
 OUT: VB, EB, AB, MB, SMB, AC, *VD, *AC, SB

Bitmuster für Sieben-Segment-Anzeige erzeugen



Die Operation **Bitmuster für Sieben-Segment-Anzeige erzeugen** erzeugt ein Bitmuster (OUT), das die Segmente einer sieben teiligen Anzeige beleuchtet. Die beleuchteten Segmente stellen das Zeichen in der niederwertigsten Ziffer des Eingangsbytes (IN) dar.

Operanden: IN: VB, EB, AB, MB, SMB, AC, *VD, *AC, SB
 OUT: VB, EB, AB, MB, SMB, AC, *VD, *AC, *AC, SB

Bild 10-52 zeigt die Codierung die von dieser Operation für die Beleuchtung einer Sieben-Segment-Anzeige verwendet wird.

(IN) LSD	Segment-Anzeige	(OUT) - g f e d c b a		(IN) LSD	Segment-Anzeige	(OUT) - g f e d c b a
0		0 0 1 1 1 1 1 1		8		0 1 1 1 1 1 1 1
1		0 0 0 0 0 1 1 0		9		0 1 1 0 0 1 1 1
2		0 1 0 1 1 0 1 1		AE		0 1 1 1 0 1 1 1
3		0 1 0 0 1 1 1 1		B		0 1 1 1 1 1 0 0
4		0 1 1 0 0 1 1 0		C		0 0 1 1 1 0 0 1
5		0 1 1 0 1 1 0 1		D		0 1 0 1 1 1 1 0
6		0 1 1 1 1 1 0 1		E		0 1 1 1 1 0 0 1
7		0 0 0 0 0 1 1 1		F		0 1 1 1 0 0 0 1

Bild 10-52 Codierung einer Sieben-Segment-Anzeige

Beispiele für Umwandlungen von Hexadezimalzahlen

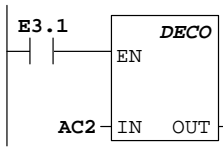
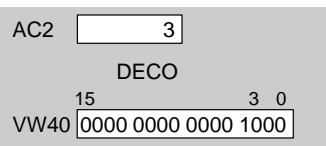
KOP	AWL
 <p>Das Bit setzen, das dem Fehlercode in Akkumulator 2 entspricht.</p>	<pre>LD E3.1 DECO AC2, VW40</pre>
Anwendung	
<p>Akkumulator 2 enthält den Fehlercode 3. Die Operation DECO setzt das Bit in VW40, das diesem Fehlercode entspricht.</p>	

Bild 10-53 Beispiel für das Setzen eines Fehlerbits mit der Operation DECO

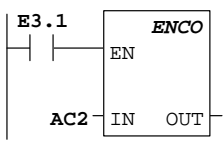
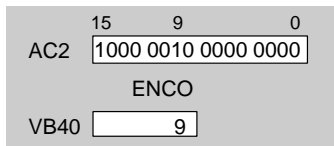
KOP	AWL
 <p>Fehlerbit in Akkumulator 2 in Fehlercode in VB40 umwandeln.</p>	<pre>LD E3.1 ENCO AC2, VB40</pre>
Anwendung	
<p>Akkumulator 2 enthält das Fehlerbit. Die Operation ENCO wandelt die niederwertigsten Bits in eine Fehlercode um, der in VB40 gespeichert wird.</p>	

Bild 10-54 Beispiel für das Umwandeln eines Fehlerbits in einen Fehlercode mit der Operation ENCO

Beispiel für Bitmuster für Sieben-Segment-Anzeige erzeugen

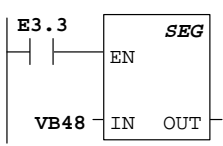
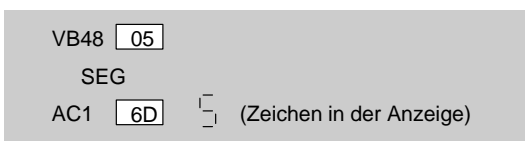
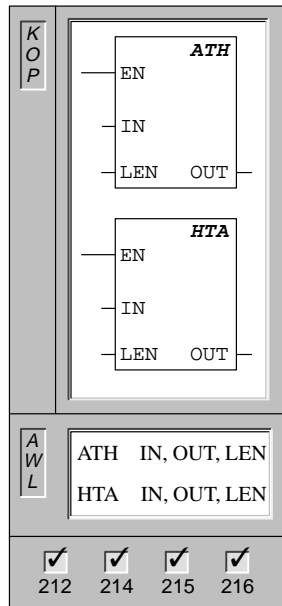
KOP	AWL
	<pre>LD E3.3 SEG VB48, AC1</pre>
Anwendung	
	

Bild 10-55 Beispiel für die Operation Bitmuster für Sieben-Segment-Anzeige erzeugen

ASCII-Zeichenkette in Hexadezimalzahl wandeln und Hexadezimalzahl in ASCII-Zeichenkette wandeln



Die Operation **ASCII-Zeichenkette in Hexadezimalzahl wandeln** wandelt die ASCII-Zeichenkette mit der Länge (LEN), die bei dem Zeichen (IN) beginnt, in Hexadezimalziffern, die an der Adresse (OUT) beginnen, um. Die ASCII-Zeichenkette darf maximal 255 Zeichen lang sein.

Die Operation **Hexadezimalzahl in ASCII-Zeichenkette wandeln** wandelt die Hexadezimalziffern, die an Eingangsbyte IN beginnen, in eine ASCII-Zeichenkette, die an der Adresse OUT beginnt, um. Wieviele Hexadezimalziffern umgewandelt werden sollen, wird durch die Länge (LEN) angegeben. Maximal können 255 Hexadezimalziffern umgewandelt werden.

Operanden: IN, OUT: VB, EB, AB, MB, SMB, *VD, *AC, SB
 LEN: VB, EB, AB, MB, SMB, AC, *VD, *AC, SB

Zulässige ASCII-Zeichen sind die Hexadezimalwerte 30 bis 39 und 41 bis 46.

Diese Operationen beeinflussen die folgenden Sondermerker: SM1.7 (unzulässiges ASCII-Zeichen)

Beispiel für die Operation ASCII-Zeichenkette in Hexadezimalzahl wandeln

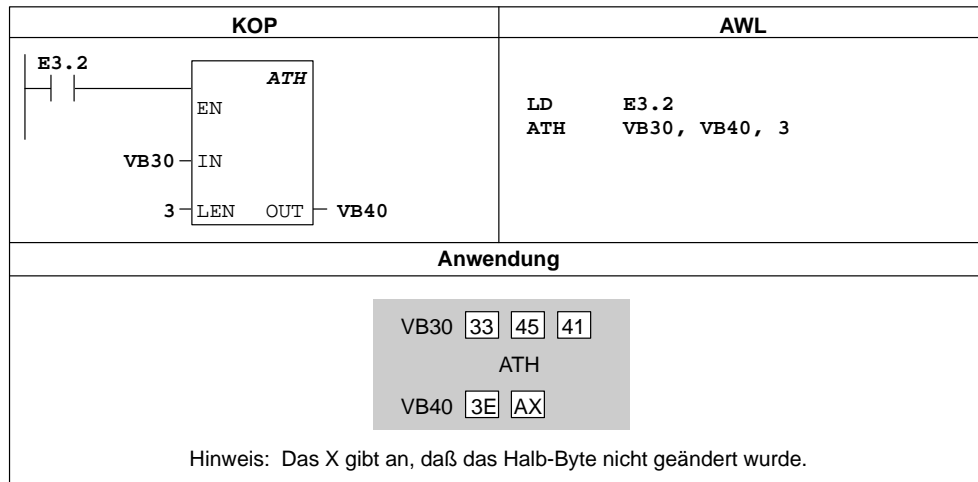
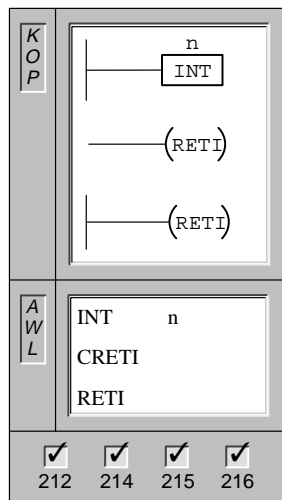


Bild 10-56 Beispiel für die Umwandlung einer ASCII-Zeichenkette in Hexadezimalziffern

10.14 Interrupt- und Kommunikationsoperationen

Interruptprogramm beginnen und Interruptprogramm beenden



Die Operation **Interruptprogramm beginnen** markiert den Anfang eines Interruptprogramms (n).

Die Operation **Interruptprogramm bedingt beenden** beendet einen Interrupt in Abhängigkeit von dem Zustand der vorherigen Verknüpfung.

Mit der Spule **Interruptprogramm absolut beenden** muß jedes Interruptprogramm beendet werden.

Operanden: n: 0 bis 127

Interruptprogramme

Sie können jedes Interruptprogramm mit einer Interruptmarke versehen, die den Beginn des Programms kennzeichnet. Das Interruptprogramm besteht aus den Operationen, die Sie zwischen der Interruptmarke und der Operation zum absoluten Beenden des Programms anordnen. Das Interruptprogramm wird in Reaktion auf ein zugeordnetes internes oder externes Ereignis ausgeführt. Sie können das Programm beenden (und damit die Steuerung wieder an das Hauptprogramm übergeben), indem Sie die Operation Interruptprogramm absolut beenden (RETI) oder die Operation Interruptprogramm bedingt beenden ausführen. Mit der Operation Interruptprogramm absolut beenden muß jedes Interruptprogramm abgeschlossen werden.

Richtlinien zur Verwendung von Interruptprogrammen

Mit der Interruptbearbeitung können Sie schnell auf besondere interne oder externe Ereignisse reagieren. Sie sollten Ihr Interruptprogramm so aufbauen, daß es eine bestimmte Aufgabe ausführt und anschließend die Steuerung wieder an das Hauptprogramm übergibt. Programmieren Sie möglichst kurze Interruptprogramme mit präzisen Angaben, so daß die Programme schnell bearbeitet werden können und andere Prozesse nicht lange unterbrochen werden. Mißachten Sie diese Richtlinie, kann es zu unvorhersehbaren Zuständen kommen, die den Betrieb der vom Hauptprogramm gesteuerten Geräte stören können. Für Interruptprogramme gilt die Devise "je kürzer, desto besser".

Einschränkungen

Beachten Sie beim Arbeiten mit Interruptprogrammen folgende Hinweise:

- Schließen Sie alle Interruptprogramme an das Ende des KOP-Hauptprogramms an.
- Die Operationen DISI, ENI, CALL, HDEF, FOR/NEXT, LSCR, SCRE, SCRT und END können Sie in Interruptprogrammen nicht verwenden.
- Beenden Sie jedes Interruptprogramm absolut (Operation RETI).

Systemunterstützung für Interrupts

Kontakte, Spulen und Akkumulatoren können von Interrupts beeinflusst werden. Deshalb speichert das System den Stack, die Akkumulatoren und die Sondermerker (SM), die den Status von Akkumulatoren und Anweisungen anzeigen, und lädt sie später neu. Dadurch wird verhindert, daß das Hauptprogramm durch eine Verzweigung zu oder von einem Interruptprogramm gestört wird.

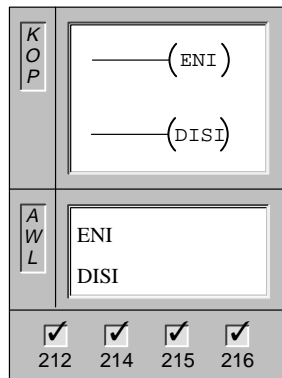
Gemeinsames Nutzen von Daten im Hauptprogramm und in Interruptprogrammen

Sie können Daten in einem Hauptprogramm und in einem oder in mehreren Interruptprogrammen gemeinsam nutzen. So kann beispielsweise ein Teil Ihres Hauptprogramms Daten zur Verfügung stellen, die von einem Interruptprogramm verarbeitet werden und umgekehrt. Wenn Ihr Hauptprogramm und ein Interruptprogramm Daten gemeinsam nutzen, müssen Sie sich der Tatsache bewusst sein, daß Interruptereignisse asynchron zum Hauptprogramm ablaufen. Deshalb können sie zu jedem beliebigen Zeitpunkt während der Bearbeitung Ihres Hauptprogramms auftreten. Fehler in der Konsistenz der gemeinsam genutzten Daten können durch die Ausführung von Interruptprogrammen entstehen, wenn die Bearbeitung der Operationen in Ihrem Hauptprogramm von Interruptereignissen unterbrochen wird.

Es gibt eine Reihe von Programmier Techniken, mit denen Sie sicherstellen, daß Fehler beim gemeinsamen Nutzen von Daten im Haupt- und im Interruptprogramm vermieden werden. Diese Techniken schränken den Zugriff auf gemeinsam genutzte Daten ein bzw. sie erlauben keine Unterbrechungen der Anweisungsfolgen, die auf gemeinsam genutzte Daten zugreifen.

- Bei einem AWL-Programm, das eine einzige Variable gemeinsam nutzt: Handelt es sich bei den gemeinsam genutzten Daten um eine einzige Byte-, Wort- oder Doppelwortvariable und ist Ihr Programm in AWL geschrieben, dann dürfen Zwischenergebnisse von Operationen mit gemeinsam genutzten Daten nur an Adressen im Speicher bzw. in Akkumulatoren abgelegt werden, die nicht gemeinsam genutzt werden.
- Bei einem KOP-Programm, das eine einzige Variable gemeinsam nutzt: Handelt es sich bei den gemeinsam genutzten Daten um eine einfache Byte-, Wort- oder Doppelwortvariable und ist Ihr Programm in KOP geschrieben, dann müssen Sie mit Hilfe der Übertragungsoperationen (MOV_B, MOV_W, MOV_DW, MOV_R) auf die gemeinsam genutzten Adressen im Speicher zugreifen. Viele KOP-Operationen entsprechen Anweisungsfolgen in AWL, die unterbrochen werden können. Jede dieser Übertragungsoperationen entspricht jedoch einer einzigen AWL-Anweisung, deren Bearbeitung nicht von Interruptereignissen beeinflusst werden kann.
- Bei AWL- oder KOP-Programmen, die mehrere Variablen gemeinsam nutzen: Handelt es sich bei den gemeinsam genutzten Daten um mehrere zusammengehörige Bytes, Wörter und Doppelwörter, dann kann die Bearbeitung des Interruptprogramms von den Operationen Alle Interruptereignisse sperren (DISI) und Alle Interruptereignisse freigeben (ENI) gesteuert werden. An der Stelle in Ihrem Hauptprogramm, an der Sie Operationen anordnen, die auf gemeinsamen Speicher zugreifen, müssen Sie die Interruptereignisse sperren. Nachdem alle Operationen, die mit dem gemeinsamen Speicher arbeiten, ausgeführt wurden, müssen Sie die Interruptereignisse wieder freigeben. Während der Zeit, in der die Interruptereignisse gesperrt sind, können keine Interruptprogramme ausgeführt werden und auf gemeinsamen Speicher zugreifen. Diese Programmier Technik kann allerdings verzögerte Reaktionen auf Interruptereignisse verursachen.

Alle Interruptereignisse freigeben und Alle Interruptereignisse sperren



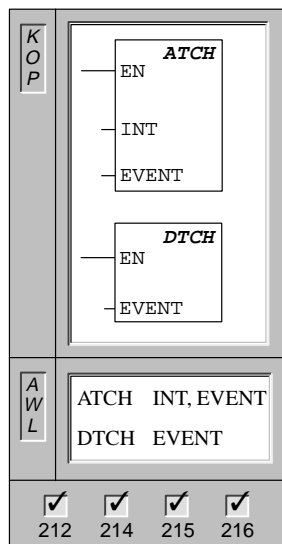
Die Operation **Alle Interruptereignisse freigeben** gibt die Bearbeitung aller zugeordneten Interruptereignisse frei.

Die Operation **Alle Interruptereignisse sperren** sperrt die Bearbeitung aller Interruptereignisse.

Operanden: keine

Wenn Sie in den Betriebszustand RUN wechseln, sperren Sie die Interrupts. Befindet sich die CPU im Betriebszustand RUN, können Sie mit der Operation ENI alle Interruptereignisse freigeben. Durch die Operation Alle Interruptereignisse sperren werden die Interrupts in eine Warteschlange aufgenommen. Sie können mit dieser Operation kein Interruptprogramm aufrufen.

Interrupt zuordnen und Interrupt trennen



Die Operation **Interrupt zuordnen** ordnet einem Interruptereignis (EVENT) die Nummer eines Interruptprogramms (INT) zu und gibt dann das Ereignis frei.

Die Operation **Interrupt trennen** trennt ein Interruptereignis (EVENT) von allen Interruptprogrammen und sperrt dann das Ereignis.

Operanden: INT : 0 bis 127
EVENT: 0 bis 20

Beschreibung der Operationen Interrupt zuordnen und Interrupt trennen

Bevor Sie ein Interruptprogramm aufrufen können, müssen Sie zwischen dem Interruptereignis und dem Teil des Programms, den Sie bei Auftreten des Interruptereignisses bearbeiten möchten, eine Verbindung herstellen. Mit der Operation Interrupt zuordnen (ATCH) ordnen Sie dem Interruptereignis (durch die Nummer des Ereignisses gekennzeichnet) einen Teil des Programms zu (durch die Nummer eines Interruptprogramms gekennzeichnet). Sie können einem einzigen Interruptprogramm mehrere Interruptereignisse zuordnen. Ein Interruptereignis kann jedoch nicht mehreren Interruptprogrammen gleichzeitig zugeordnet sein. Tritt ein Ereignis bei freigegebenen Interrupts auf, wird nur das Interruptprogramm, das diesem Ereignis zuletzt zugeordnet wurde, bearbeitet.

Wenn Sie einem Interruptprogramm ein Interruptereignis zuordnen, wird das Ereignis automatisch freigegeben. Wenn Sie die Operation Alle Interruptereignisse sperren ausführen, werden alle auftretenden Interrupts in eine Warteschlange gestellt, bis Sie die Interruptsperre mit der Operation Alle Interruptereignisse freigeben wieder aufheben.

Einzelne Interruptereignisse können Sie sperren, indem Sie die Zuordnung von Ereignis und Programm mit der Operation DTCH (Interrupt trennen) aufheben. Diese Operation setzt den Interrupt in einen inaktiven Zustand, in dem er übergangen (nicht bearbeitet) wird.

Tabelle 10-13 führt die verschiedenen Interruptereignisse auf.

Tabelle 10-13 Beschreibung der Interruptereignisse

Nummer des Ereignisses	Beschreibung des Interrupts	212	214	215	216
0	Steigende Flanke, E 0.0*	Ja	Ja	Ja	Ja
1	Fallende Flanke, E 0.0*	Ja	Ja	Ja	Ja
2	Steigende Flanke, E0.1		Ja	Ja	Ja
3	Fallende Flanke, E0.1		Ja	Ja	Ja
4	Steigende Flanke, E0.2		Ja	Ja	Ja
5	Fallende Flanke, E0.2		Ja	Ja	Ja
6	Steigende Flanke, E0.3		Ja	Ja	Ja
7	Fallende Flanke, E0.3		Ja	Ja	Ja
8	Schnittstelle 0: Zeichen empfangen	Ja	Ja	Ja	Ja
9	Schnittstelle 0: Übertragung beendet	Ja	Ja	Ja	Ja
10	Zeitgesteuerter Interrupt 0, SMB34	Ja	Ja	Ja	Ja
11	Zeitgesteuerter Interrupt 1, SMB35		Ja	Ja	Ja
12	HSC0: CV = PV (aktueller Wert = voreingestellter Wert)*	Ja	Ja	Ja	Ja
13	HSC1: CV = PV (aktueller Wert = voreingestellter Wert)*		Ja	Ja	Ja
14	HSC1: Richtungswechsel		Ja	Ja	Ja
15	HSC1: externes Rücksetzen		Ja	Ja	Ja
16	HSC2: CV = PV (aktueller Wert = voreingestellter Wert)		Ja	Ja	Ja
17	HSC2: Richtungswechsel		Ja	Ja	Ja
18	HSC2: externes Rücksetzen		Ja	Ja	Ja
19	PLS0: Zählwert erreicht		Ja	Ja	Ja
20	PLS1: Zählwert erreicht		Ja	Ja	Ja
21	Zeit T32: CT = PT			Ja	Ja
22	Zeit T96: CT = PT			Ja	Ja
23	Schnittstelle 0: Meldungsempfang beendet			Ja	Ja
24	Schnittstelle 1: Meldungsempfang beendet				Ja
25	Schnittstelle 1: Zeichen empfangen				Ja
26	Schnittstelle 1: Übertragung beendet				Ja
* Ist Ereignis 12 (HSC0, PV = CV) einem Interrupt zugeordnet, dann können die Ereignisse 0 und 1 keinem Interrupt zugeordnet sein. Ist eines der Ereignisse 0 oder 1 einem Interrupt zugeordnet, dann kann Ereignis 12 keinem Interrupt zugeordnet sein.					

Interrupts der Kommunikationsschnittstellen

Die serielle Programmierschnittstelle des Automatisierungssystem kann mit Hilfe eines KOP- oder AWL-Programms gesteuert werden. Die Kommunikation über diese Schnittstelle wird frei programmierbare Kommunikation genannt. Bei der frei programmierbaren Kommunikation definiert Ihr Programm die Baudrate, die Bits pro Zeichen, die Parität und das Protokoll. Sende- und Empfangsinterrupts erleichtern die programmgesteuerte Kommunikation. Ausführliche Informationen hierzu entnehmen Sie dem Abschnitt zu Sende- und Empfangsoperationen.

E/A-Interrupts

Zu den E/A-Interrupts gehören Interrupts bei steigender oder fallender Flanke, Interrupts des schnellen Zählers und Impulsfolge-Interrupts. Die CPU kann bei steigender und/oder bei fallender Flanke an einem Eingang einen Interrupt erzeugen. Tabelle 10-14 führt die Eingänge auf, die bei den verschiedenen CPUs für Interrupts zur Verfügung stehen. Die Ereignisse Steigende Flanke und Fallende Flanke können für jeden dieser Eingänge erfaßt werden. Mit diesen Ereignissen können auch Fehlerbedingungen angezeigt werden, die bei Auftreten des Ereignisses sofort beachtet werden müssen.

Tabelle 10-14 Unterstützte E/A-Interrupts

E/A-Interrupts	CPU 212	CPU 214	CPU 215	CPU 216
Ein- und Ausgänge	E0.0	E0.0 bis E0.3	E0.0 bis E0.3	E0.0 bis E0.3

Mit den Interrupts der schnellen Zähler können Sie auf folgende Ereignisse reagieren: aktueller Wert gleich voreingestellter Wert, Zählrichtung wechselt und verläuft umgekehrt zur Drehrichtung der Welle, Zähler wird extern zurückgesetzt. Mit jedem dieser Ereignisse von schnellen Zählern können Sie auf schnelle Ereignisse reagieren, die bei den Zyklusraten des Automatisierungssystems nicht gesteuert werden können.

Die Impulsfolge-Interrupts melden sofort, wenn die angegebene Anzahl Impulse ausgegeben wurde. Impulsfolgen werden häufig zum Steuern von Schrittmotoren eingesetzt.

Sie können die oben beschriebenen Interrupts freigeben, indem Sie dem entsprechenden E/A-Ereignis ein Interruptprogramm zuordnen.

Zeitgesteuerte Interrupts

Zeitgesteuerte Interrupts umfassen Interrupts und die Interrupts der Zeiten T32/T96. Die CPU kann einen oder mehrere zeitgesteuerte Interrupts unterstützen (siehe Tabelle 10-15). Sie können Aktionen, die zyklisch ausgeführt werden sollen, mit einem zeitgesteuerten Interrupt definieren. Die Zykluszeit wird in Schritten von 1 ms angegeben, der Bereich liegt zwischen 5 ms und 255 ms. Die Zykluszeit für den zeitgesteuerten Interrupt 0 schreiben Sie in SMB34, die Zykluszeit für den zeitgesteuerten Interrupt 1 schreiben Sie in SMB35.

Tabelle 10-15 Unterstützte zeitgesteuerte Interrupts

Zeitgesteuerte Interrupts	CPU 212	CPU 214	CPU 215	CPU 216
Anzahl der unterstützten zeitgesteuerten Interrupts	1	2	2	2

Das zeitgesteuerte Interruptereignis ruft jedesmal, wenn die Zeit abläuft, das entsprechende Interruptprogramm auf. Mit zeitgesteuerten Interruptereignissen steuern Sie im allgemeinen das regelmäßige Abfragen der Analogeingänge.

Ein zeitgesteuerter Interrupt wird freigegeben und die Zeit beginnt zu laufen, wenn Sie einem zeitgesteuerten Interruptereignis ein Interruptprogramm zuordnen. Dabei erfaßt das System die Zykluszeit, damit spätere Änderungen sich nicht auf die Zykluszeit auswirken. Möchten Sie die Zykluszeit ändern, müssen Sie einen neuen Wert für die Zykluszeit angeben und anschließend das Interruptprogramm erneut dem zeitgesteuerten Interrupt zuordnen. Beim erneuten Zuordnen löscht die Funktion den angesammelten Zeitwert der vorherigen Zuordnung, und die Zeit beginnt mit dem neuen Wert zu laufen.

Nach der Freigabe läuft der zeitgesteuerte Interrupt kontinuierlich und bearbeitet das zugeordnete Interruptprogramm jedesmal, wenn das angegebene Zeitintervall abläuft. Wenn Sie die Betriebsart RUN verlassen oder die Zuordnung von Interrupt und Interruptprogramm trennen (DTCH), wird der zeitgesteuerte Interrupt gesperrt. Wenn Sie die Operation Alle Interruptereignisse sperren ausführen, treten die zeitgesteuerten Interrupts zwar weiterhin auf, werden jedoch in eine Warteschlange aufgenommen (entweder bis die Interrupts wieder freigegeben werden oder die Warteschlange voll ist). Bild 10-58 zeigt ein Beispiel für einen zeitgesteuerten Interrupt.

Die zeitgesteuerten Interrupts T32/T96 dienen zum zeitgesteuerten Reagieren nach Ablauf eines angegebenen Intervalls. Diese Interrupts werden nur von den Einschaltverzögerungen (TON) mit einer Auflösung von 1 ms, T32 und T96, unterstützt. Ansonsten verfügen die Zeiten T32 und T96 über die übliche Funktionalität. Ist der Interrupt freigegeben, wird das zugeordnete Interruptprogramm ausgeführt, wenn bei der Aktualisierung der Zeiten mit einer Auflösung von 1 ms im Zyklus der CPU der aktuelle Wert der aktiven Zeit gleich dem voreingestellten Wert der Zeit ist (siehe Abschnitt 10.5). Sie geben diese Interrupts frei, indem Sie dem Interruptereignis T32/T96 ein Interruptprogramm zuordnen.

Prioritäten der Interruptereignisse und Warteschlangen

Die Prioritäten der Interrupts werden nach folgendem festen Prioritätenschema vergeben:

- Kommunikation (höchste Priorität)
- E/A-Interrupts
- Zeitgesteuerte Interrupts (niedrigste Priorität)

Interrupts werden von der CPU in der Reihenfolge ihres Auftretens und unter Beachtung der jeweiligen Prioritäten abgearbeitet. Es ist immer nur ein Programm zur Interruptbearbeitung aktiv. Wird gerade ein Interruptprogramm bearbeitet, dann wird dieses Programm beendet. Es kann nicht von einem später auftretenden Interruptprogramm unterbrochen werden, auch wenn die Priorität des Programms höher ist. Tritt ein Interrupt auf, während ein anderer Interrupt bearbeitet wird, wird der spätere Interrupt in eine Warteschlange zur späteren Bearbeitung aufgenommen.

Tabelle 10-16 zeigt die drei Warteschlangen für Interrupts und die maximale Anzahl Interrupts, die pro Warteschlange aufgenommen werden können.

Tabelle 10-16 Warteschlangen für Interrupts und maximale Anzahl an Einträgen pro Warteschlange

Warteschlange	CPU 212	CPU 214	CPU 215	CPU 216
Kommunikationsinterrupts	4	4	4	8
E/A-Interrupts	4	16	16	16
Zeitgesteuerte Interrupts	2	4	8	8

Es können mehr Interrupts auftreten, als die Warteschlange aufnehmen kann. Deshalb verfügt das System über Überlaufmerker für Warteschlangen, die die Art der Interruptereignisse angeben, die nicht in die Warteschlange aufgenommen werden konnten. Tabelle 10-17 erläutert die Sondermerker, die bei einem Warteschlangen-Überlauf gesetzt werden. Diese Bits 4.0, 4.1 und 4.2 dürfen Sie nur in einem Interruptprogramm verwenden, weil sie zurückgesetzt werden, wenn die Warteschlange abgearbeitet ist und die Bearbeitung des Hauptprogramms wieder aufgenommen wird.

Tabelle 10-17 Definitionen der Sondermerker für Warteschlangen-Überlauf

Beschreibung (0 = kein Überlauf, 1 = Überlauf)	Sondermerker
Überlauf der Warteschlange für Kommunikationsinterrupts	SM4.0
Überlauf der Warteschlange für E/A-Interrupts	SM4.1
Überlauf der Warteschlange für zeitgesteuerte Interrupts	SM4.2

Tabelle 10-18 zeigt das Interruptereignis, die Priorität und die zugeordnete Nummer des Ereignisses.

Tabelle 10-18 Beschreibung der Interruptereignisse

Nummer des Ereignisses	Beschreibung des Interrupts	Prioritätsklasse	Prio.
8	Schnittstelle 0: Zeichen empfangen	Kommunikations-Interrupts: höchste Prioritätsklasse	0
9	Schnittstelle 0: Übertragung beendet		0*
23	Schnittstelle 0: Meldungsempfang beendet		0*
24	Schnittstelle 1: Meldungsempfang beendet		1
25	Schnittstelle 1: Zeichen empfangen		1*
26	Schnittstelle 1: Übertragung beendet		1*
0	Steigende Flanke, E 0.0**	E/A-Interrupts: mittlere Prioritätsklasse	0
2	Steigende Flanke, E0.1		1
4	Steigende Flanke, E0.2		2
6	Steigende Flanke, E0.3		3
1	Fallende Flanke, E 0.0**		4
3	Fallende Flanke, E0.1		5
5	Fallende Flanke, E0.2		6
7	Fallende Flanke, E0.3		7
12	HSC0: CV = PV (aktueller Wert = voreingestellter Wert)**		0
13	HSC1: CV = PV (aktueller Wert = voreingestellter Wert)*		8
14	HSC1: Richtungswechsel		9
15	HSC1: externes Rücksetzen		10
16	HSC2: CV = PV (aktueller Wert = voreingestellter Wert)		11
17	HSC2: Richtungswechsel		12
18	HSC2: externes Rücksetzen		13
19	PLS0: Zählwert erreicht	14	
20	PLS1: Zählwert erreicht	15	
10	Zeitgesteuerter Interrupt 0	Zeitgesteuerte Interrupts: niedrigste Prioritätsklasse	0
11	Zeitgesteuerter Interrupt 1		1
21	Zeit T32: CT = PT		2
22	Zeit T96: CT = PT		3

* Da die Kommunikation halbduplex ist, haben Send- und Empfangsinterrupt die gleiche Priorität.

** Ist Ereignis 12 (HSC0, PV = CV) einem Interrupt zugeordnet, dann können die Ereignisse 0 und 1 keinem Interrupt zugeordnet sein. Ist eines der Ereignisse 0 oder 1 einem Interrupt zugeordnet, dann kann Ereignis 12 keinem Interrupt zugeordnet sein.

Beispiel für Interrupts

Bild 10-57 zeigt ein Beispiel für Interrupt-Operationen.

KOP		AWL
<p>Netzwerk 1</p>	<p>Im ersten Zyklus: Interruptprogramm 4 als Interrupt Steigende Flanke an E0.0 definieren.</p>	<p>Network 1</p> <pre>LD SM0.1 ATCH 4, 0 ENI</pre>
<p>Netzwerk 2</p>	<p>Wird ein E/A-Fehler erkannt, Interrupt Steigende Flanke an E0.0 sperren. (Dieser Strompfad ist optional.)</p>	<p>Network 2</p> <pre>LD SM5.0 DTCH 0</pre>
<p>Netzwerk 3</p>	<p>Alle Interruptereignisse sperren, wenn M5.0 eingeschaltet wird.</p>	<p>Network 3</p> <pre>LD M5.0 DISI</pre>
<p>Netzwerk 50</p>	<p>KOP-Hauptprogramm beenden.</p>	<p>Network 50</p> <pre>MEND</pre>
<p>Netzwerk 60</p>	<p>Interruptprogramm für Interrupt Steigende Flanke an E/A.</p>	<p>Network 60</p> <pre>INT 4</pre>
<p>Netzwerk 65</p>	<p>Aufgrund eines E/A-Fehlers Interruptprogramm bedingt beenden.</p>	<p>Network 65</p> <pre>LD SM5.0 CRETI</pre>
<p>Netzwerk 66</p>	<p>Interruptprogramm für Interrupt Steigende Flanke an E0.0 beenden.</p>	<p>Network 66</p> <pre>RETI</pre>

Bild 10-57 Beispiel für Interrupt-Operationen in KOP und AWL

Bild 10-58 zeigt, wie Sie mit einem zeitgesteuerten Interrupt den Wert eines Analogeingangs auslesen können.



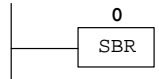
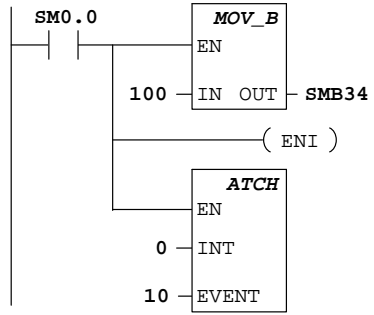

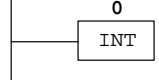
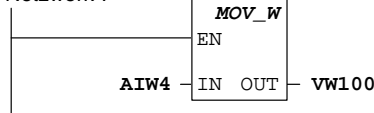
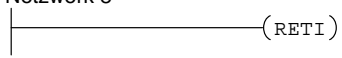
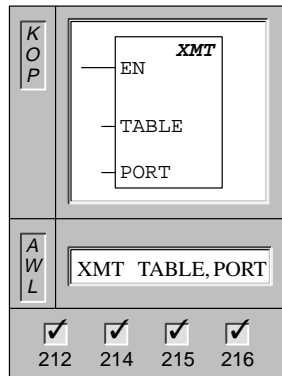
KOP		AWL
Hauptprogramm		
<p>Netzwerk 1</p>  <p>Netzwerk 2</p> 	<p>Network 1 LD SM0.1 CALL 0</p> <p>Network 2 MEND</p>	
Unterprogramme		
<p>Netzwerk 3</p>  <p>Netzwerk 4</p>  <p>Netzwerk 5</p> 	<p>Unterprogramm 0 beginnen.</p> <p>Merker ständig EIN: Zeitgesteuertes Intervall von 100 ms setzen.</p> <p>Alle Interruptereignisse freigeben</p> <p>Zeitgesteuerten Interrupt 0 Interruptprogramm 0 zuordnen.</p> <p>Unterprogramm beenden.</p>	<p>Network 3 SBR 0</p> <p>Network 4 LD SM0.0 MOVB 100, SMB34</p> <p>ENI</p> <p>ATCH 0, 10</p> <p>Network 5 RET</p>
Interruptprogramme		
<p>Netzwerk 6</p>  <p>Netzwerk 7</p>  <p>Netzwerk 8</p> 	<p>Interruptprogramm 0 beginnen.</p> <p>AEW4 abfragen.</p> <p>Interruptprogramm beenden.</p>	<p>Network 6 INT 0</p> <p>Network 7 MOVW AEW4, VW100</p> <p>Network 8 RETI</p>

Bild 10-58 Beispiel für das Lesen eines Analogeingangs mit einem zeitgesteuerten Interrupt

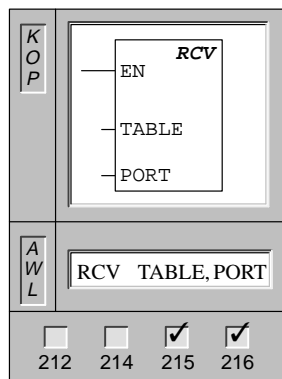
Meldung aus Zwischenspeicher übertragen und Meldung in Zwischenspeicher empfangen



Die Operation **Meldung aus Zwischenspeicher übertragen** veranlaßt das Übertragen der Daten aus dem Zwischenspeicher (TABLE). Der erste Eintrag im Puffer gibt an, wieviele Bytes übertragen werden sollen. PORT gibt die Kommunikationsschnittstelle an, die für das Übertragen verwendet werden soll.

Operanden: TABLE: VB, EB, AB, MB, SMB, *VD, *AC, SB
 PORT: 0 bis 1

Die Operation XMT wird bei der frei programmierbaren Kommunikation zum Übertragen von Daten über die Kommunikationsschnittstelle(n) verwendet.



Die Operation **Meldung in Zwischenspeicher empfangen** veranlaßt Änderungen im Setup, die den Meldungsempfang initiieren oder beenden. Damit die Operation Meldung in Zwischenspeicher empfangen funktionieren kann, müssen Sie eine Start- und eine Endebedingung angeben. Meldungen, die über die angegebene Schnittstelle (PORT) empfangen werden, werden im Zwischenspeicher (TABLE) abgelegt. Der erste Eintrag im Puffer gibt die Anzahl der empfangenen Bytes an.

Operanden: TABLE: VB, EB, AB, MB, SMB, *VD, *AC, SB
 PORT: 0 bis 1

Die Operation RCV wird bei der frei programmierbaren Kommunikation zum Empfangen von Daten über die Kommunikationsschnittstelle(n) verwendet.

Beschreibung der frei programmierbaren Kommunikation

In der frei programmierbaren Kommunikation kann die serielle Kommunikationsschnittstelle der CPU mit Hilfe des Anwenderprogramms gesteuert werden. Haben Sie die frei programmierbare Kommunikation gewählt, steuert das KOP-Programm den Betrieb der Kommunikationsschnittstelle über Empfangsinterrupts, Übertragungsinterrupts, die Übertragungsoperation (XMT) und die Empfangsoperation (RCV). Während der frei programmierbaren Kommunikation wird das Kommunikationsprotokoll komplett vom KOP-Programm gesteuert. SMB30 (für die Schnittstelle 0) und SMB130 (für die Schnittstelle 1, sofern Ihre CPU über zwei Schnittstellen verfügt) dienen zum Auswählen von Baudrate und Parität.

Beindet sich die CPU im Betriebszustand STOP, dann wird die frei programmierbare Kommunikation gesperrt und die normale Kommunikation wiederhergestellt (z.B. Zugriff über das Programmiergerät).

Im einfachsten Fall können Sie eine Meldung an einen Drucker oder an ein Anzeigergerät schicken und dafür nur die Operation XMT einsetzen. Sie können aber auch Daten an einen Barcode-Leser, eine Waage oder ein Schweißgerät übertragen. In jedem Fall müssen Sie Ihr Programm so schreiben, daß es das Protokoll des Geräts unterstützt, mit dem die CPU in der frei programmierbaren Kommunikation kommunizieren möchten.

Die frei programmierbare Kommunikation kann nur im Betriebszustand RUN der CPU aktiviert werden. Sie geben die frei programmierbare Kommunikation frei, indem Sie in dem Feld zum Auswählen des Protokolls in SMB30 (Schnittstelle 0) oder in SMB130 (Schnittstelle 1) den Wert "01" einstellen. Während der frei programmierbaren Kommunikation können Sie nicht mit dem Programmiergerät kommunizieren.

Hinweis

Die Aktivierung der frei programmierbaren Kommunikation kann mit dem Sondermerker SM0.7 gesteuert werden. Dieser Sondermerker stellt die aktuelle Position des Betriebsartenschalters dar. Ist SM0.7 = 0, dann ist der Schalter in der Stellung TERM. Ist SM0.7 = 1, dann ist der Schalter in der Stellung RUN. Aktivieren Sie die frei programmierbare Kommunikation nur, wenn der Schalter auf RUN steht, dann können Sie den Betrieb der CPU mit dem Programmiergerät bedienen und beobachten, indem Sie den Schalter auf eine andere Position stellen.

Initialisierung der frei programmierbaren Kommunikation

SMB30 und SMB130 konfigurieren die Kommunikationsschnittstellen 0 und 1 für die frei programmierbare Kommunikation. In diesen Sondermerkern stellen Sie die Baudrate, die Parität und die Anzahl der Datenbits ein. Die Steuerbytes sind in Tabelle 10-19 beschrieben.

Tabelle 10-19 Sondermerker SMB30 und SMB130

Schnittstelle 0	Schnittstelle 1	Beschreibung								
Format von SMB30	Format von SMB130	<div style="display: flex; justify-content: space-between; align-items: center;"> MSB 7 <div style="border: 1px solid black; padding: 2px;"> <table style="border-collapse: collapse; text-align: center;"> <tr> <td style="border: 1px solid black; width: 15px; height: 15px;">p</td> <td style="border: 1px solid black; width: 15px; height: 15px;">p</td> <td style="border: 1px solid black; width: 15px; height: 15px;">d</td> <td style="border: 1px solid black; width: 15px; height: 15px;">b</td> <td style="border: 1px solid black; width: 15px; height: 15px;">b</td> <td style="border: 1px solid black; width: 15px; height: 15px;">b</td> <td style="border: 1px solid black; width: 15px; height: 15px;">m</td> <td style="border: 1px solid black; width: 15px; height: 15px;">m</td> </tr> </table> </div> LSB 0 </div> Steuerbyte für die frei programmierbare Kommunikation	p	p	d	b	b	b	m	m
p	p	d	b	b	b	m	m			
SM30.6 und SM30.7	SM130.6 und SM130.7	pp Auswahl der Parität 00 = keine Parität 01 = gerade Parität 10 = keine Parität 11 = ungerade Parität								
SM30.5	SM130.5	d Datenbits pro Zeichen 0 = 8 Bits pro Zeichen 1 = 7 Bits pro Zeichen								
SM30.2 bis SM30.4	SM130.2 bis SM130.4	bbb Baudrate für die frei programmierbare Kommunikation 000 = 38.400 Baud (bei der CPU 212: 19.200 Baud) 001 = 19.200 Baud 010 = 9.600 Baud 011 = 4.800 Baud 100 = 2.400 Baud 101 = 1.200 Baud 110 = 600 Baud 111 = 300 Baud								
SM30.0 und SM30.1	SM130.0 und SM130.1	mm Auswahl des Protokolls 00 = Protokoll der Punkt-zu-Punkt-Schnittstelle (PPI/Slave-Modus) 01 = Protokoll der frei programmierbaren Kommunikation 10 = PPI/Master-Modus 11 = Reserviert (Voreinstellung PPI/Slave-Modus)								
Hinweis: Für den Betrieb der Schnittstelle 0 wird ein Stoppbit für alle Konfigurationen erzeugt. Ausnahme: 7 Bit pro Zeichen, keine Parität. In diesem Fall werden zwei Stoppbits erzeugt. Für den Betrieb der Schnittstelle 1 wird ein Stoppbit für alle Konfigurationen erzeugt.										

Senden von Daten mit der Operation XMT

Die Operation XMT erleichtert Ihnen das Senden von Daten. Mit der Operation XMT können Sie einen Puffer mit maximal 255 Zeichen senden. Wenn das letzte Zeichen des Puffers gesendet wurde, wird ein Interrupt erzeugt (Interruptereignis 9 für Schnittstelle 0 und Interruptereignis 26 für Schnittstelle 1), sofern dem Ereignis Übertragung beendet ein Interruptprogramm zugeordnet ist. Sie können auch ohne Interrupt Daten senden (z. B. wenn Sie eine Meldung an einen Drucker schicken möchten), indem Sie SM4.5 oder SM4.6 auf das Ende der Übertragung überwachen.

Empfangen von Daten mit der Operation RCV

Die Operation RCV erleichtert Ihnen das Empfangen von Daten. Mit der Operation RCV können Sie einen Puffer mit maximal 255 Zeichen empfangen. Wenn das letzte Zeichen im Puffer empfangen wurde, wird ein Interrupt erzeugt (Interruptereignis 23 für Schnittstelle 0 und Interruptereignis 24 für Schnittstelle 1), sofern dem Ereignis Empfangen beendet ein Interruptprogramm zugeordnet ist. Sie können auch ohne Interrupt Daten empfangen, indem Sie SM86 überwachen.

SMB86 (oder SMB186) ist nicht auf Null, wenn die Operation RCV inaktiv ist. SMB86 (oder SMB186) ist auf Null, wenn Daten empfangen werden.

Mit der Operation RCV können Sie Bedingungen für den Beginn und das Ende einer Meldung auswählen. Die Bedingungen für den Beginn und das Ende einer Meldung sind in Tabelle 10-20 beschrieben (SM86 bis SM94 für Schnittstelle 0 und SM186 bis SB194 für Schnittstelle 1).

Hinweis

Eine freigegebene Funktion zum Empfangen von Meldungen wird sofort automatisch beendet, wenn ein Überlauf- oder Paritätsfehler auftritt. Sie müssen ein Kriterium für den Beginn (x oder z) und ein Kriterium für das Ende (y, t oder maximale Zeichenzahl erreicht) definieren, damit die Funktion zum Empfangen von Meldungen fehlerfrei arbeiten kann.

Tabelle 10-20 Sondermerker SMB86 bis SMB94 und SMB186 bis SMB194

Schnittstelle 0	Schnittstelle 1	Beschreibung								
SMB86	SMB186	<div style="display: flex; justify-content: space-between; align-items: center;"> <div style="text-align: center;"> <small>MSB</small> 7 <table border="1" style="border-collapse: collapse; text-align: center;"> <tr> <td style="padding: 2px;">n</td> <td style="padding: 2px;">r</td> <td style="padding: 2px;">e</td> <td style="padding: 2px;">0</td> <td style="padding: 2px;">0</td> <td style="padding: 2px;">t</td> <td style="padding: 2px;">c</td> <td style="padding: 2px;">p</td> </tr> </table> </div> <div style="text-align: center;"> <small>LSB</small> 0 </div> </div> <p style="text-align: right;">Statusbyte zum Empfangen von Meldungen</p> <p>n: 1 = Meldungsempfang vom Anwender gesperrt r: 1 = Meldungsempfang beendet: Fehler in Eingangsparametern oder fehlendes Kriterium zum Starten bzw. Beenden e: 1 = Endezeichen empfangen t: 1 = Meldungsempfang beendet: Zeit abgelaufen c: 1 = Meldungsempfang beendet: maximale Zeichenzahl erreicht p: 1 = Meldungsempfang beendet: Paritätsfehler</p>	n	r	e	0	0	t	c	p
n	r	e	0	0	t	c	p			
SMB87	SMB187	<div style="display: flex; justify-content: space-between; align-items: center;"> <div style="text-align: center;"> <small>MSB</small> 7 <table border="1" style="border-collapse: collapse; text-align: center;"> <tr> <td style="padding: 2px;">n</td> <td style="padding: 2px;">x</td> <td style="padding: 2px;">y</td> <td style="padding: 2px;">z</td> <td style="padding: 2px;">m</td> <td style="padding: 2px;">t</td> <td style="padding: 2px;">0</td> <td style="padding: 2px;">0</td> </tr> </table> </div> <div style="text-align: center;"> <small>LSB</small> 0 </div> </div> <p style="text-align: right;">Steuerbyte zum Empfangen von Meldungen</p> <p>n: 0 = Funktion zum Empfangen von Meldungen ist gesperrt 1 = Funktion zum Empfangen von Meldungen ist freigegeben Das Bit zum Sperren/Freigeben des Meldungsempfangs wird jedesmal, wenn die Operation RCV bearbeitet wird, geprüft. x: 0 = SMB88 oder SMB188 ignorieren. 1 = Mit dem Wert von SMB88 oder von SMB188 den Beginn der Meldung erkennen. y; 0 = SMB89 oder SMB189 ignorieren. 1 = Mit dem Wert von SMB89 oder von SMB189 den Beginn der Meldung erkennen. z: 0 = SMW90 oder SMB190 ignorieren. 1 = Mit dem Wert von SMW90 den Beginn der Meldung erkennen. m: 0 = Zeit mißt die Zeit zwischen den Zeichen. 1 = Zeit mißt die Zeit der Meldungen t: 0 = SMW92 oder SMW192 ignorieren. 1 = Empfang beenden, wenn die Zeitdauer von SMW92 oder SMW192 überschritten ist.</p> <p>Diese Bits definieren die Kriterien zum Kennzeichnen einer Meldung (einschließlich der Kriterien für den Beginn und das Ende einer Meldung). Um den Beginn einer Meldung zu erkennen, werden die freigegebenen Kriterien für den Beginn einer Meldung logisch durch UND verknüpft und müssen in Reihe auftreten (Leerlauflinie gefolgt von einem Startzeichen). Damit das Ende einer Meldung erkannt wird, werden die freigegebenen Kriterien für das Ende einer Meldung durch ODER verknüpft.</p> <p>Gleichungen für die Kriterien für den Beginn und das Ende einer Meldung: Beginn der Meldung = z * x Ende der Meldung = y + t + maximale Zeichenzahl erreicht</p> <p>Hinweis: Eine freigegebene Funktion zum Empfangen von Meldungen wird sofort automatisch beendet, wenn ein Überlauf- oder Paritätsfehler auftritt. Sie müssen ein Kriterium für den Beginn (x oder z) und ein Kriterium für das Ende (y, t oder maximale Zeichenzahl erreicht) definieren, damit die Funktion zum Empfangen von Meldungen fehlerfrei arbeiten kann.</p>	n	x	y	z	m	t	0	0
n	x	y	z	m	t	0	0			
SMB88	SMB188	Zeichen für den Beginn einer Meldung								
SMB89	SMB189	Zeichen für das Ende einer Meldung								

Tabelle 10-20 Sondermerker SMB86 bis SMB94 und SMB186 bis SMB194, Fortsetzung

Schnittstelle 0	Schnittstelle 1	Beschreibung
SMB90 SMB91	SMB190 SMB191	Dauer der Leerlaufzeit in Millisekunden. Das erste Zeichen, das nach Ablauf der Zeit für die Leerlaufzeit empfangen wird, kennzeichnet den Beginn einer neuen Meldung. SM90 (oder SM190) ist das höchstwertige Byte und SM91 (oder SM191) ist das niederwertigste Byte.
SMB92 SMB93	SMB192 SMB193	Wert für die Zeitüberwachung beim Messen der Zeit zwischen den Zeichen und der Zeit der Meldungen in Millisekunden. Ist die Zeit überschritten, wird das Empfangen von Meldungen beendet. SM92 (oder SM192) ist das höchstwertige Byte und SM93 (oder SM193) ist das niederwertigste Byte.
SMB94	SMB194	Maximale Zeichenzahl, die empfangen werden kann (1 bis 255 Bytes). Hinweis: Dieser Bereich muß auf die maximal zu erwartende Puffergröße eingestellt werden, auch wenn der Empfangsabbruch durch die Zeichenzählfunktion nicht eingesetzt wird.

Empfangen von Daten mit Hilfe von Zeichen-Interrupts

Damit Sie bei den unterstützten Protokollen eine größere Flexibilität haben, können Sie auch interruptgesteuert Daten empfangen. Hierbei erzeugt jedes empfangene Zeichen einen Interrupt. Das empfangene Zeichen wird in SMB2 abgelegt und der Status der Parität (sofern aktiviert) wird in SM3.0 abgelegt. Dies geschieht unmittelbar vor der Ausführung des Interruptprogramms, das dem Ereignis Zeichen empfangen zugeordnet ist.

- SMB2 dient als Puffer für empfangene Zeichen bei der frei programmierbaren Kommunikation. Die Zeichen, die während der frei programmierbaren Kommunikation empfangen werden, werden in diesem Speicher abgelegt, damit das Anwenderprogramm schnell auf die Werte zugreifen kann.
- SMB3 wird bei der frei programmierbaren Kommunikation eingeschaltet und enthält ein Bit, das gesetzt wird, wenn bei einem der empfangenen Zeichen ein Paritätsfehler erkannt wird. Alle anderen Bits dieses Byte sind reserviert. Mit diesem Bit können Sie die Meldung verwerfen oder eine negative Quittierung erzeugen.

Hinweis

SMB2 und SMB3 werden von Schnittstelle 0 und Schnittstelle 1 gemeinsam genutzt. Löst der Empfang eines Zeichens in Schnittstelle 0 die Ausführung des Interruptprogramms aus, das dem Ereignis (Interruptereignis 8) zugeordnet ist, dann enthält SMB2 das in Schnittstelle 0 empfangene Zeichen und SMB3 enthält den Paritätsstatus des Zeichens. Löst der Empfang eines Zeichens in Schnittstelle 1 die Ausführung des Interruptprogramms aus, das dem Ereignis (Interruptereignis 25) zugeordnet ist, dann enthält SMB2 das in Schnittstelle 1 empfangene Zeichen und SMB3 enthält den Paritätsstatus des Zeichens.

Beispiel für die Operationen Meldung in Zwischenspeicher empfangen und Meldung aus Zwischenspeicher übertragen

Dieses Programmierbeispiel zeigt die Verwendung der Operationen Meldung in Zwischenspeicher empfangen und Meldung aus Zwischenspeicher übertragen. Das Programm empfängt solange eine Zeichenkette, bis das Zeichen für Zeilenschaltung empfangen wird. Daraufhin wird die Meldung an den Sender zurückgeschickt.

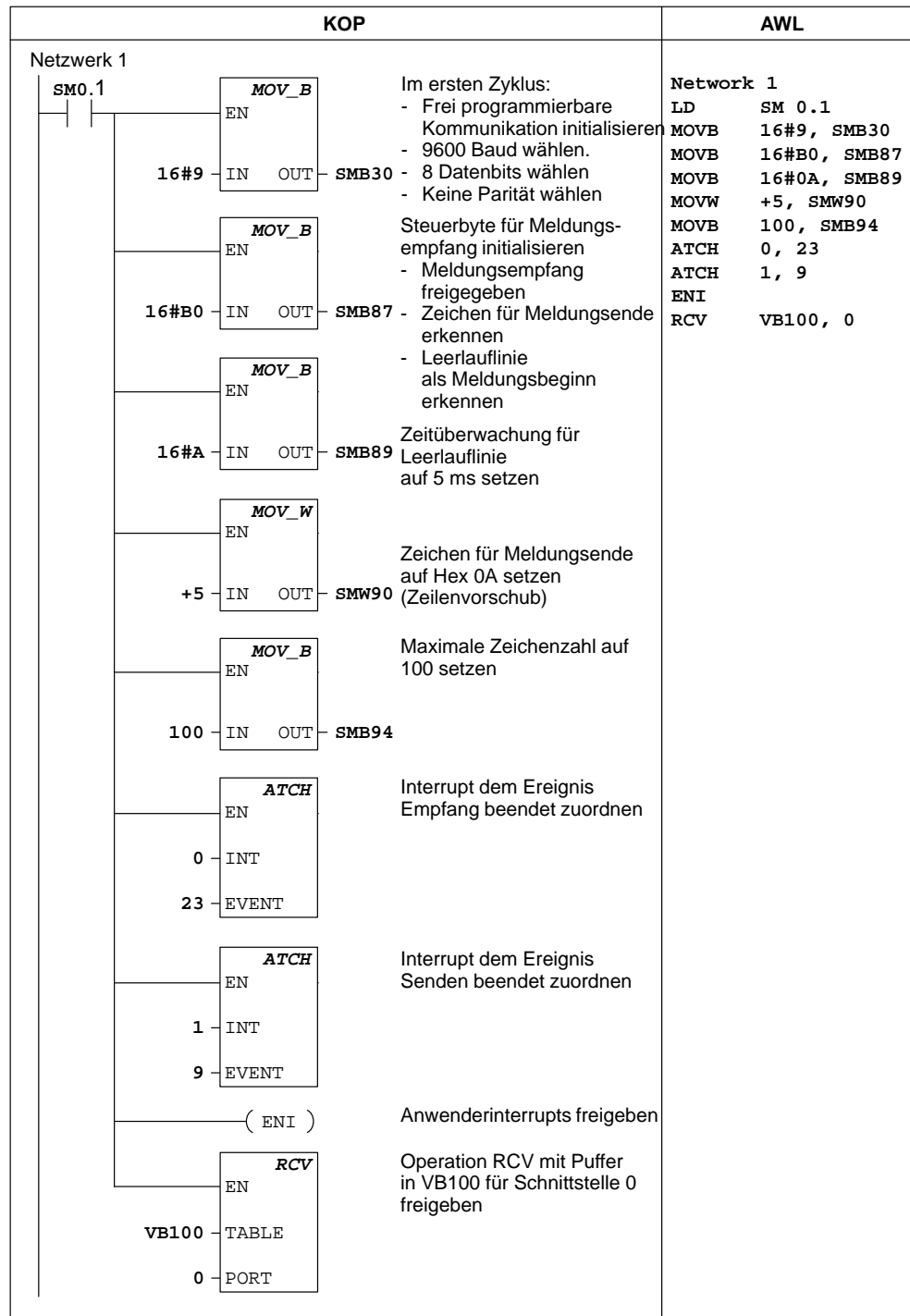


Bild 10-59 Beispiel für die Operation Meldung aus Zwischenspeicher übertragen

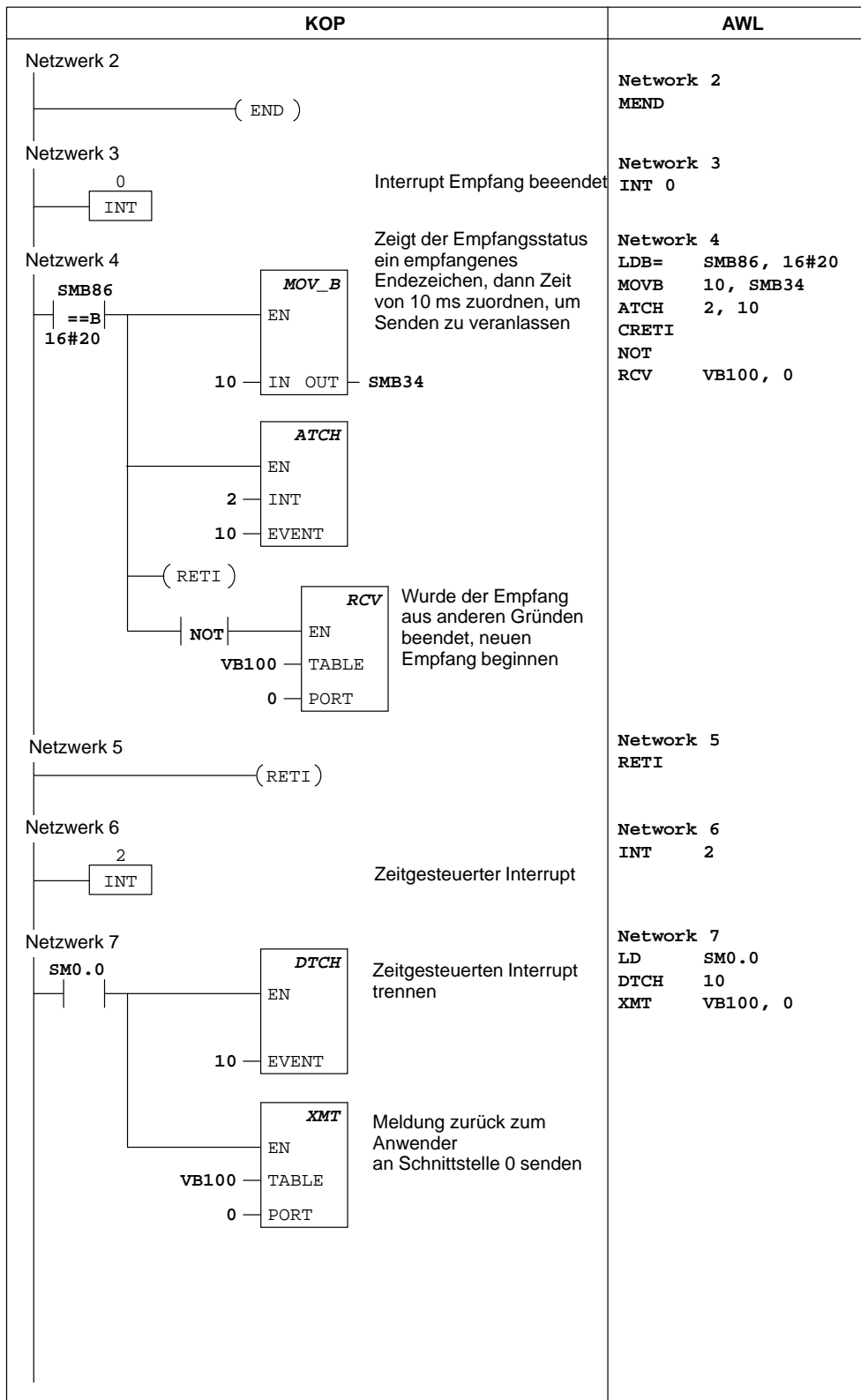


Bild 10-60 Beispiel für die Operation Meldung aus Zwischenspeicher übertragen, Fortsetzung

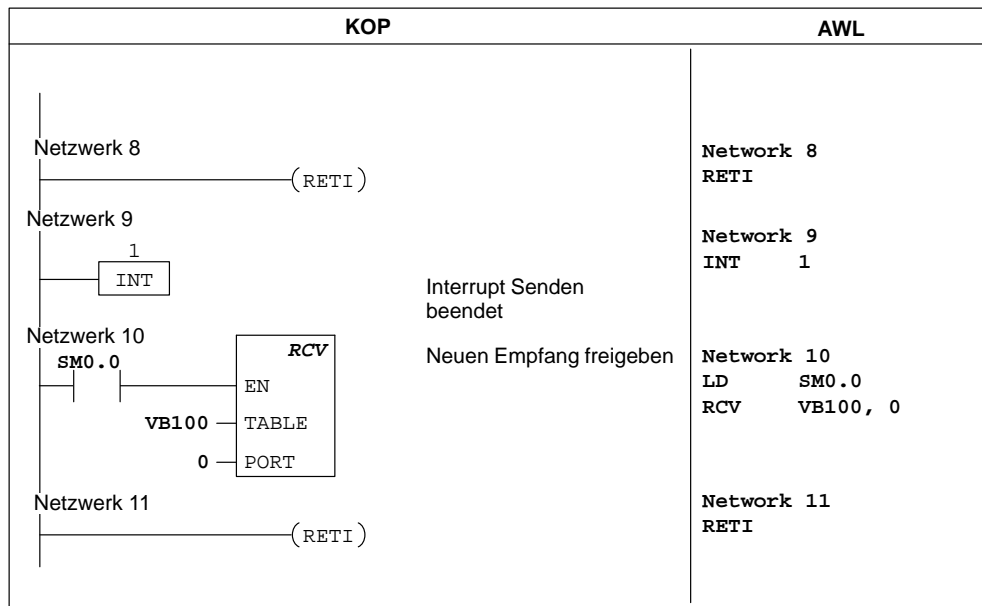
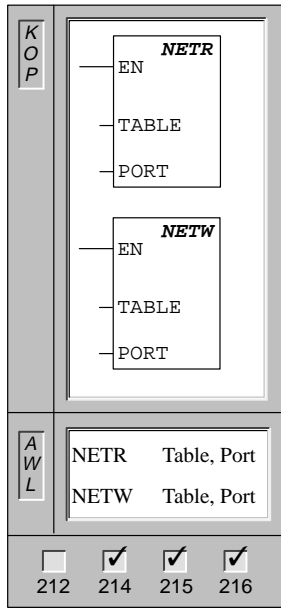


Bild 10-60 Beispiel für die Operation Meldung aus Zwischenspeicher übertragen, Fortsetzung

Aus Netz lesen und In Netz schreiben



Die Operation **Aus Netz lesen** löst eine Kommunikationsoperation aus, die entsprechend der Definition in der Tabelle (TABLE) über die angegebene Schnittstelle (PORT) aus einem fernen Gerät Daten liest.

Die Operation **In Netz schreiben** löst eine Kommunikationsoperation aus, die entsprechend der Definition in der Tabelle (TABLE) über die angegebene Schnittstelle (PORT) in ein fernes Gerät Daten schreibt.

Operanden: TABLE: VB, MB, *VD, *AC
 PORT: 0 bis 1

Mit der Operation NETR können Sie maximal 16 Bytes an Informationen aus einer fernen Station lesen. Mit der Operation NETW können Sie maximal 16 Bytes an Informationen in eine ferne Station schreiben. Maximal acht Operationen NETR und NETW dürfen gleichzeitig in der S7-200 aktiviert sein, z.B. vier Operationen NETR und vier Operationen NETW oder zwei Operationen NETR und sechs Operationen NETW.

Bild 10-60 definiert die Tabelle, auf die sich der Parameter TABLE für die Operationen NETR und NETW bezieht.

Byte-Versatz	7	0						
0	D	AE	E	0	Fehlercode			
1	Adresse der fernen Station							
2	Pointer auf den Daten-							
3	bereich in der							
4	fernen Station							
5	(E, A, M, S oder V)							
6	Länge der Daten							
7	Datenbyte 0							
8	Datenbyte 1							
	:							
	:							
22	Datenbyte 15							

D Beendet (Op. ausgeführt): 0 = nicht ausgeführt 1 = ausgeführt
 A Aktiv (Op. befindet sich in Warteschlange): 0 = nicht aktiv 1 = aktiv
 E Fehler (Op. enthält eine Fehlerbedingung): 0 = kein Fehler 1 = Fehler

Adresse der fernen Station: Adresse des AS, auf dessen Daten zugegriffen werden soll.

Pointer auf den Datenbereich in der entfernten Station: indirekter Pointer auf die Daten, auf die zugegriffen werden soll.

Länge der Daten: Anzahl der Datenbytes, auf die in der fernen Station zugegriffen werden soll (zwischen 1 und 16 Bytes).

Datenbereich zum Empfangen oder Übertragen: zwischen 1 und 16 Bytes, die wie folgt für Daten reserviert sind:

Bei der Operation NETR handelt es sich um den Datenbereich, in dem die Werte, die aus der fernen Station gelesen werden sollen, nach der Ausführung der Operation NETR gespeichert werden.

Bei der Operation NETW handelt es sich um den Datenbereich, in dem die Werte, die in die ferne Station geschrieben werden sollen, vor der Ausführung der Operation NETW gespeichert werden.

Fehlercode	Definition
0	Kein Fehler.
1	Ablauf der Zeit; ferne Station reagiert nicht.
2	Empfangsfehler; Paritätsfehler, Formfehler bzw. Prüfsummenfehler in Antwort.
3	Offline-Fehler; durch doppelte Stationsadresse oder fehlerhafte Hardware.
4	Fehler; Überlauf Warteschlange; mehr als acht Boxen NETR/NETW aktiviert.
5	Protokollverletzung; Versuch, NETR/NETW auszuführen, ohne PPI+ in SMB30 freizugeben.
6	Ungültiger Parameter; NETR/NETW-Tabelle enthält ungültigen Wert.
7	Keine Ressource; ferne Station belegt (Ladevorgang in Bearbeitung).
8	Fehler Lage 7; Verletzung Anwendungsprotokoll.
9	Meldungsfehler; falsche Datenadresse bzw. inkorrekte Datenlänge
A-F	Nicht verwendet (reserviert für zukünftige Verwendung).

Bild 10-60 Definition von TABLE für die Operationen NETR und NETW

Beispiel für die Operationen Aus Netz lesen und In Netz schreiben

Bild 10-61 zeigt ein Beispiel für die Verwendung der Operationen NETR und NETW. In diesem Beispiel handelt es sich um eine Produktionsstraße, in der Becher mit Butter gefüllt und zu einer von vier Verpackungsmaschinen weitergeleitet werden. Die Verpackungsmaschine packt jeweils acht Becher Butter in einen Karton. Eine Weiche steuert, zu welcher Verpackungsmaschine die einzelnen Becher Butter weitergeleitet werden. Vier CPUs 212 steuern die vier Verpackungsmaschinen. Eine CPU 214, die mit einem Textdisplay TD 200 ausgestattet ist, steuert die Weiche. Bild 10-61 zeigt den Aufbau des Netzes.

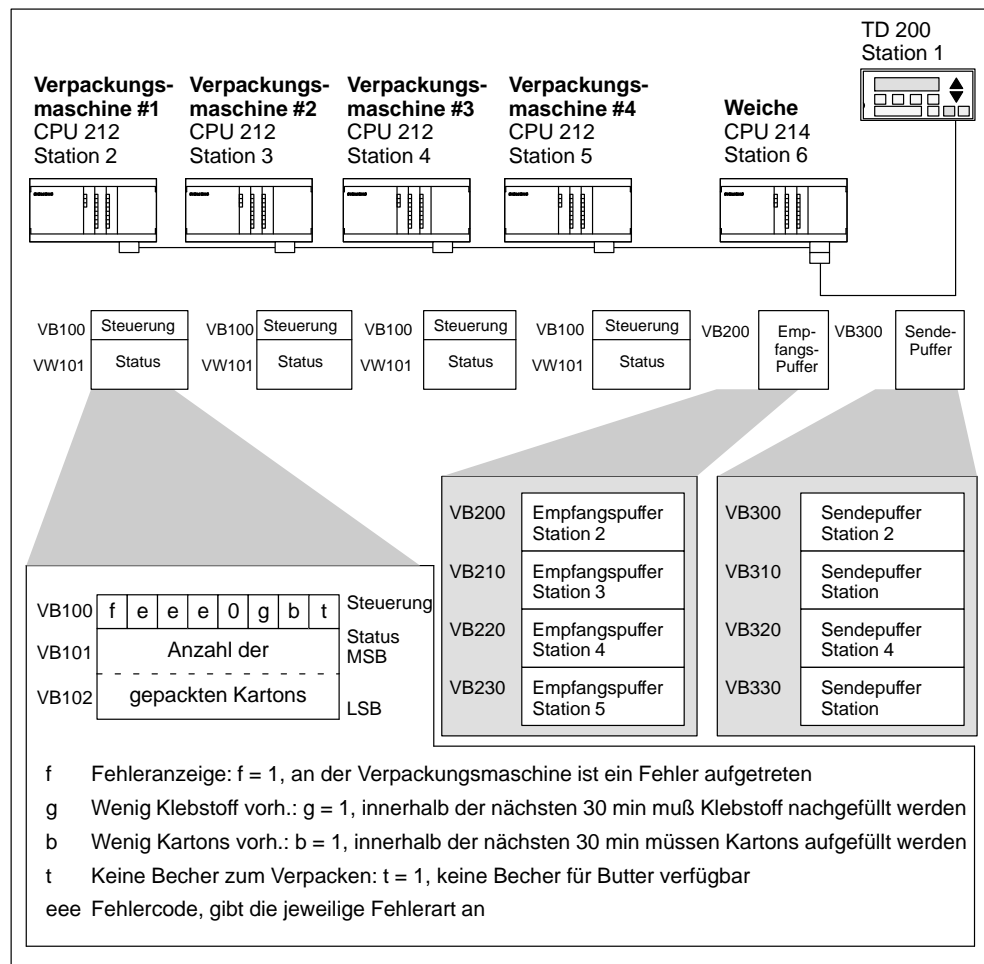


Bild 10-61 Beispiel für die Operationen NETR und NETW

Bild 10-62 zeigt eine ausführliche Darstellung des Empfangs- und des Sendepuffers, die zum Zugreifen auf die Daten der Station 2 dienen (diese Puffer befinden sich in VB200 und VB300).

Die CPU 214 liest mit der Operation NETR regelmäßig die Steuerungs- und Statusinformationen aus den einzelnen Verpackungsmaschinen aus. Jedesmal, wenn eine Verpackungsmaschine 100 Kartons gepackt hat, sendet die Weiche mit der Operation NETW eine Meldung, um das Statuswort zurückzusetzen.

Bild 10-63 zeigt das Programm, mit dem das Steuerbyte gelesen, die gepackten Kartons gezählt und der Zählwert der Kartons für jede Verpackungsmaschine einzeln zurückgesetzt wird (in diesem Fall Verpackungsmaschine Nr. 1).

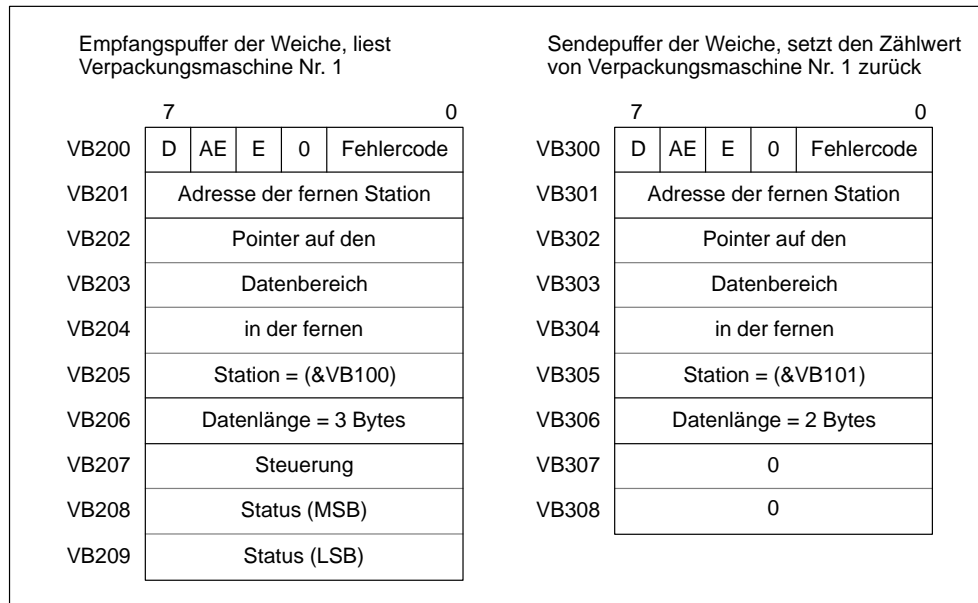


Bild 10-62 Definition von TABLE für die Operationen NETR und NETW im Beispielprogramm

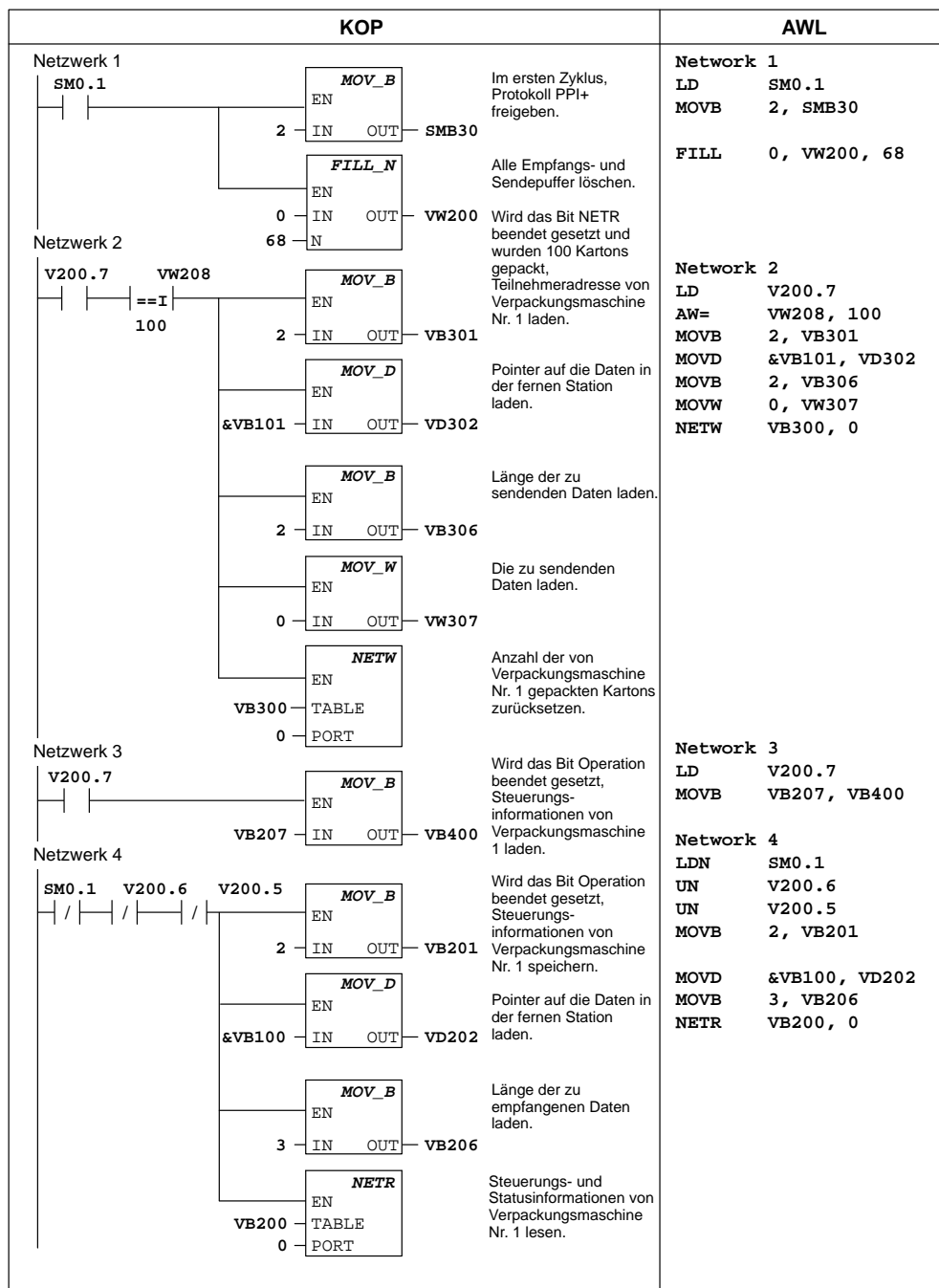


Bild 10-63 Beispiel für die Operationen NETR und NETW

Technische Daten

A

Kapitelübersicht

Abschnitt	Beschreibung	Seite
A.1	Allgemeine technische Daten	A-3
A.2	CPU 212, DC-Versorgung, DC-Eingänge, DC-Ausgänge	A-6
A.3	CPU 212, AC-Versorgung, DC-Eingänge, Relaisausgänge	A-8
A.4	CPU 212, 24-V-AC-Versorgung, DC-Eingänge, Relaisausgänge	A-10
A.5	CPU 212, AC-Versorgung, AC-Eingänge, AC-Ausgänge	A-12
A.6	CPU 212 AC-Versorgung, DC-Eingänge (stromliefernd), Relaisausgänge	A-14
A.7	CPU 212, AC-Versorgung, 24-V-AC-Eingänge, AC-Ausgänge	A-16
A.8	CPU 212, AC-Versorgung, AC-Eingänge, Relaisausgänge	A-18
A.9	CPU 214, DC-Versorgung, DC-Eingänge, DC-Ausgänge	A-20
A.10	CPU 214, AC-Versorgung, DC-Eingänge, Relaisausgänge	A-22
A.11	CPU 214, AC-Versorgung, AC-Eingänge, AC-Ausgänge	A-24
A.12	CPU AC-Versorgung, DC-Eingänge (stromliefernd), Relaisausgänge	A-26
A.13	CPU 214, AC-Versorgung, 24-V-AC-Eingänge, AC-Ausgänge	A-28
A.14	CPU 214, AC-Versorgung, AC-Eingänge, Relaisausgänge	A-30
A.15	CPU 215, DC-Versorgung, DC-Eingänge, DC-Ausgänge	A-32
A.16	CPU 215, AC-Versorgung, DC-Eingänge, Relaisausgänge	A-34
A.17	CPU 216, DC-Versorgung, DC-Eingänge, DC-Ausgänge	A-36
A.18	CPU 216, AC-Versorgung, DC-Eingänge, Relaisausgänge	A-38
A.19	Erweiterungsmodul EM 221 Digitaleingabe 8 x 24 V DC	A-40
A.20	Erweiterungsmodul EM 221, Digitaleingabe 8 x 120 V AC	A-41
A.21	Erweiterungsmodul EM 221 Digitaleingabe (stromliefernd) 8 x 24 V DC	A-42
A.22	Erweiterungsmodul EM 221, Digitaleingabe 8 x 24 V AC	A-43
A.23	Erweiterungsmodul EM 222, Digitalausgabe 8 x 24 V DC	A-44
A.24	Erweiterungsmodul EM 222, Digitalausgabe 8 x Relais	A-45
A.25	Erweiterungsmodul EM 222, Digitalausgabe 8 x 120/230 V AC	A-46
A.26	Erweiterungsmodul EM 223 Digitalein-/Digitalausgabe 4 x 24-V-DC-Eingang/4 x 24-V-DC-Ausgang	A-48
A.27	Erweiterungsmodul EM 223, Digitalein-/Digitalausgabe 8 x 24-V-DC-Eingang/8 x 24-V-DC-Ausgang	A-50

Abschnitt	Beschreibung	Seite
A.28	Erweiterungsmodul EM 223, Digitalein-/Digitalausgabe 16 x 24-V-DC-Eingang/16 x 24-V-DC-Ausgang	A-52
A.29	Erweiterungsmodul EM 223, Digitalein-/Digitalausgabe 4 x 24-V-DC-Eingang/4 x Relaisausgang	A-54
A.30	Erweiterungsmodul EM 223, Digitalein-/Digitalausgabe 4 x 120-V-AC-Eingang/4 x 120/230-V-AC-Ausgang	A-55
A.31	Erweiterungsmodul EM 223, Digitalein-/Digitalausgabe 8 x 24-V-DC-Eingang/8 x Relaisausgang	A-56
A.32	Erweiterungsmodul EM 223, Digitalein-/Digitalausgabe 16 x 24-V-DC-Eingang/16 x Relaisausgang	A-58
A.33	Erweiterungsmodul EM 231, Analogeingabe AE 3 x 12 Bit	A-60
A.34	Erweiterungsmodul EM 232, Analogausgabe AA 2 x 12 Bit	A-66
A.35	Erweiterungsmodul EM 235, Analogein-/Analogausgabe AE 3 / AA 1 x 12 Bit	A-69
A.36	Speichermodul 8 K x 8	A-78
A.37	Speichermodul 16 K x 8	A-79
A.38	Batterieminuten	A-80
A.39	Steckleitung für Erweiterungsmodule	A-81
A.40	PC/PPI-Kabel	A-82
A.41	DC-Eingangssimulator für die CPU 212	A-84
A.42	DC-Eingangssimulator für die CPU 214	A-85
A.43	DC-Eingangssimulator für die CPU 215/216	A-86

A.1 Allgemeine technische Daten

Nationale und internationale Richtlinien

Die im folgenden aufgeführten Richtlinien wurden zum Festlegen der jeweiligen Leistungsmerkmale und technischen Daten sowie zum Prüfen der Produktreihe S7-200 herangezogen. Tabelle A-1 definiert die Übereinstimmung mit diesen Richtlinien.

- Underwriters Laboratories, Inc.: UL 508 registriert (Industrial Control Equipment)
- Canadian Standards Association: CSA C22.2 Nummer 142, geprüft (Process Control Equipment)
- Factory Mutual Research: FM Klasse I, Kategorie 2, Gefahrenbereichsgruppen A, B, C und D, T4A
- VDE 0160: Elektronische Geräte zur Verwendung in Starkstromanlagen
- EG-Richtlinie zu Niederspannungen
EN 61131-2: Automatisierungssysteme - Anforderungen an die Geräte
- EG-Richtlinie zur elektromagnetischen Verträglichkeit 89/336/EWG
Richtlinien zur elektromagnetischen Störabstrahlung:
EN 50081-1: Wohn-, Gewerbe-, Leichtindustrie-Umgebungen
EN 50081-2: Industrieumgebung
Richtlinien zur elektromagnetischen Störfestigkeit:
EN 50082-2: Industrieumgebung

Technische Daten

Die S7-200 CPUs und alle S7-200 Erweiterungsmodule entsprechen den in Tabelle A-1 aufgeführten technischen Daten.

Tabelle A-1 Technische Daten für die Produktreihe S7-200

Umgebungsbedingungen - Transport und Lagerung	
IEC 68-2-2, Test Bb, trockene Wärme und IEC 68-2-1, Test Ab, Kälte	-40° C bis +70° C
IEC 68-2-30, Test Db, feuchte Wärme	25° C bis 55° C, 95% Luftfeuchtigkeit
IEC 68-2-31, Umkippen	100 mm, viermal, ohne Verpackung
IEC 68-2-32, freier Fall	1 m, fünfmal, in Versandverpackung
Umgebungsbedingungen - Betrieb	
Funktionsbereich	0° C bis 55° C, max. 95% Luftfeuchtigkeit ohne Kondensation
IEC 68-2-14, Test Nb	5° C bis 55° C, 3° C/Minute
IEC 68-2-27 mech. Stoßbeanspruchung	15 G, Impuls 11 ms, 6 Stöße auf jeder der 3 Achsen
IEC 68-2-6 Sinusschwingung	0,35 mm Spitze-Spitze 10 bis 57 Hz; 2 G Montage in Schalttafel, 1G Montage auf Hutschiene, 57 bis 150 Hz; 10 Ablenkungen pro Achse, 1 Oktave/Minute
EN 60529, IP20 mechanischer Schutz	Schutz gegen direkte Berührung von Hochspannung wie mit genormten Sonden ermittelt. Externer Schutz erforderlich gegen Staub, Schmutz, Wasser und Fremdkörper mit einem Durchmesser von maximal 12,5 mm.
Elektromagnetische Verträglichkeit - Störfestigkeit¹ nach EN50082-2¹	
EN 61000-4-2 (IEC 801-2) Elektrostatische Entladung	Entladung durch die Luft an allen Oberflächen und Kommunikationsschnittstellen: 8 kV
EN 50140 (IEC 801-3) Abgestrahltes elektromagnetisches Feld EN50204	26 MHz bis 1 GHz 10 V/m, 80% Modulation mit 1 kHz Signal 900 MHz ± 5 MHz, 10 V/m, 50% relative Einschaltdauer, 200 Hz Wiederholfrequenz
EN 61000-4-4 (IEC 801-4) Schnelle transiente Störgröße	2 kV, 5 kHz bei Kopplungsnetz zu AC- und DC-Systemspannung 2 kV, 5 kHz bei Kopplungsklemme zu Digitalein-/Digitalausgängen und Kommunikation
EN 61000-4-5 (IEC 801-5) Stoßwellenfestigkeit	2 kV asymmetrisch, 1 kV symmetrisch 5 positive/5 negative Impulse 0°, +90°, -90° Phasenwinkel (24-V-DC-Stromkreise erfordern externen Schutz vor Stromstößen)
VDE 0160 Nichtperiodische Überspannung	bei 85-V-AC-Leitung, 90° Phasenwinkel, 390 V Spitze anlegen, 1,3 ms Impuls bei 180-V-AC-Leitung, 90° Phasenwinkel, 750 V Spitze anlegen, 1,3 ms Impuls

Tabelle A-1 Technische Daten für die Produktreihe S7-200, Fortsetzung

Elektromagnetische Verträglichkeit - Störstrahlung (geleitet und abgestrahlt)² nach EN50081 -1 und -2²	
EN 55011, Klasse A, Gruppe 1, geleitet ¹ 0,15 MHz bis 0,5 MHz 0,5 MHz bis 5 MHz 5 MHz bis 30 MHz	Quasi-Spitze < 79 dB (µV); Mittelwert < 66 dB (µV) Quasi-Spitze < 73 dB (µV); Mittelwert < 60 dB (µV) Quasi-Spitze < 73 dB (µV); Mittelwert < 60 dB (µV)
EN 55011, Klasse A, Gruppe 1, abgestrahlt ¹ 30 MHz bis 230 kHz 230 MHz bis 1 GHz	Quasi-Spitze 30 dB (µV/m); gemessen bei 30 m Quasi-Spitze 37 dB (µV/m); gemessen bei 30 m
EN 55011, Klasse B, Gruppe 1, geleitet ³ 0,15 bis 0,5 MHz 0,5 MHz bis 5 MHz 5 MHz bis 30 MHz	< 66 dB (µV) abnehmende Quasi-Spitze mit log. Freq. bis 56 dB (µV); < 56 dB (µV) abnehmender Mittelwert mit log. Freq. bis 46 dB (µV) Quasi-Spitze < 56 dB (µV); Mittelwert < 46 dB (µV) Quasi-Spitze < 60 dB (µV); Mittelwert < 50 dB (µV)
EN 55011, Klasse B, Gruppe 1, abgestrahlt ³ 30 MHz bis 230 kHz 230 MHz bis 1 GHz	Quasi-Spitze 30 dB (µV/m); gemessen bei 10 m Quasi-Spitze 37 dB (µV/m); gemessen bei 10 m
Hochspannungs-Isolationsprüfung	
Stromkreis mit 24-V/5-V-Nennspannung 115/230-V-Stromkreis an Erde 115/230-V-Stromkr. an 115/230-V-Stromkr. 230-V-Stromkreis an 24/5-V-Stromkreis 115-V-Stromkreis an 24/5-V-Stromkreis	500 V DC (Grenzwerte Potentialtrennung) 1,500 V AC 1,500 V AC 1,500 V AC 1,500 V AC

- 1 Das Gerät muß auf einem geerdeten Metallrahmen montiert sein. Die S7-200 muß direkt über den Metallrahmen geerdet sein. Kabel verlaufen entlang der Metallträger.
- 2 Gültig für alle Geräte mit EC-Kennzeichnung.
- 3 Das Gerät muß in einem geerdeten Metallgehäuse installiert sein. Die AC-Eingangsleitung muß mit einem Filter (Schaffner FN 680-2,5/06 oder vergleichbar) ausgerüstet sein. Maximale Kabellänge von den Filtern zur S7-200: 25 cm. Die Verkabelung der 24-V-DC-Stromquelle und der Geberversorgung muß geschirmt sein.

Lebensdauer eines Relais

Bild A-1 zeigt die typischen Leistungsdaten, die von Relais-Herstellern zur Verfügung gestellt werden. Die tatsächliche Leistungsfähigkeit richtet sich nach der jeweiligen Verwendung.

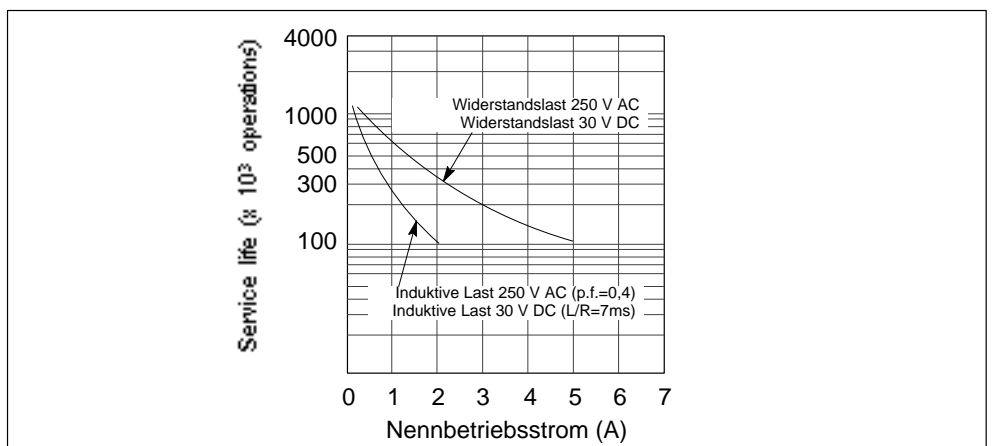


Bild A-1 Lebensdauer eines Relais

A.2 CPU 212, DC-Versorgung, DC-Eingänge, DC-Ausgänge

Bestellnummer: 6ES7 212-1AA01-0XB0

Leistungsmerkmale		Ausgänge, Fortsetzung	
Abmessungen (B x H x T)	160 x 80 x 62 mm	Verzögerung Schaltvorgänge	25 µs EIN, 120 µs AUS
Gewicht	0,3 kg	Stoßstrom	4 A, 100 ms
Stromaufnahme	5 W bei 1,75 A Last	Spannungsabfall	max. 1,8 V bei max. Strom
Größe Anwenderprogramm/ Speicher	512 Wörter/EEPROM	Potentialtrennung	500 V AC, 1 min
Größe Anwenderdaten/ Speicher Datenhaltung	512 Wörter/RAM typ. 50 h (min. 8 h bei 40 °C)	Kurzschlußschutz	keine
Integrierte E/A ¹	8 Eingänge/6 Ausgänge	Eingänge	
Maximale Anzahl Erweiterungsmodule	2	Eingangstyp (IEC 1131-2)	Typ 1, stromziehend
Unterstützte digitale E/A	64 Eingänge/64 Ausgänge	Bereich bei EIN	15 bis 30 V DC, min. 4 mA 35 V DC, 500 ms Spannungsstoß
Unterstützte analoge E/A	16 Eingänge/16 Ausgänge	Nennspannung bei EIN	DC 24 V, 7 mA
Boolesche Ausführungszeiten	1,2 µs/Operation	Maximum bei AUS	DC 5 V, 1 mA
Interne Merker	128	Ansprechzeit E0.0 bis E0.7	max. 0,3 ms
Zeiten	64 Zeiten	Potentialtrennung	500 V AC, 1 min
Zähler	64 Zähler	Stromversorgung	
Schnelle Zähler	1 SW-Zähler (max. 2 kHz)	Spannungsbereich	20,4 bis 28,8 V DC
Analogpotentiometer	1	Eingangsstrom	typ. 60 mA, nur CPU 500 mA max. Laststrom
Normen	UL 508 CSA C22.2 142 FM Klasse I, Kategorie 2 gemäß VDE 0160 gemäß EG-Richtlinie	UL/CSA-Bemessung	50 VA
Ausgänge		Verzögerungszeit	aus 24 V DC min. 10 ms
Ausgangstyp	Transistor, stromliefernd	Einschaltstromstoß	10 A Spitze bei 28,8 V DC
Spannungsbereich	20,4 V DC bis 28,8 V DC	Schmelzsicherung (nicht austauschbar)	1 A, 125 V, träge
Max. Laststrom	<u>0 bis 40° C</u> <u>55° C</u> ²	Verfügbare Strom 5 V DC	260 mA für CPU 340 mA für Erweiterungsmodule
pro einzeltem Ausgang	0,75 A 0,50 A	Elektrisch getrennt	Nein
pro 2 benachbarter	1,00 A 0,75 A	DC-Geberversorgung	
Ausgänge	2,25 A 1,75 A	Spannungsbereich	16,4 bis 28,8 V DC
Ausgänge gesamt		Welligkeit/Störströme (<10 MHz)	wie zugeführte Spannung
Klemmung induktive Last einzelner Impuls	(pro Leiter) 2A L/R = 10 ms 1A L/R = 100 ms	Verfügbare Strom 24 V DC	180 mA
Wiederholung	1 W Energieaufnahme (1/2 Li ² x Schaltfreq. < 1 W)	Kurzschlußstrombegrenzung	< 600 mA
Ableitstrom	100 µA	Elektrisch getrennt	Nein

¹ In der CPU sind 8 Eingänge im Prozeßabbild der Eingänge und 8 Ausgänge im Prozeßabbild der Ausgänge für die integrierten Ein- und Ausgänge vorgesehen.

² Lineare Leistungsminderung 40 bis 55° C, Leistungsminderung bei vertikalem Einbau 10° C.

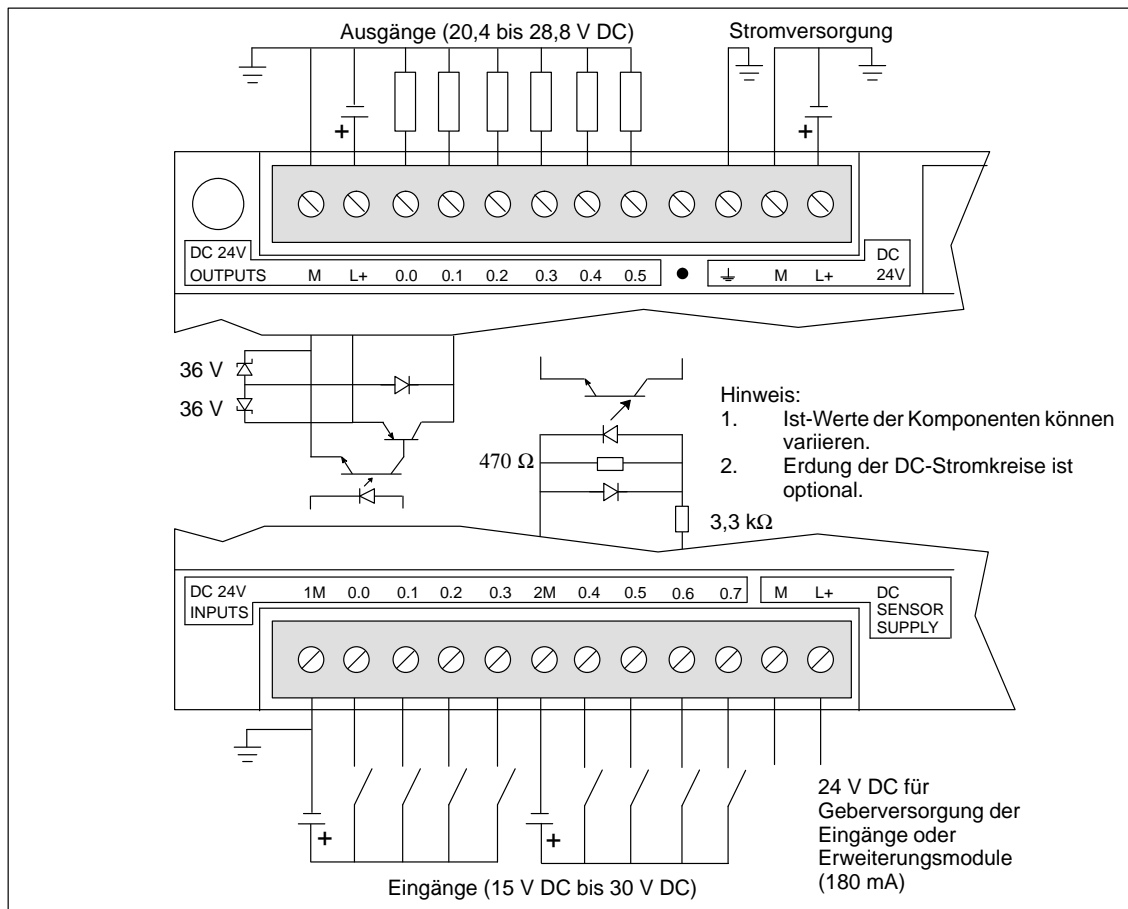


Bild A-2 Kennzeichnung der Anschlüsse bei der CPU 212 DC/DC/DC

A.3 CPU 212, AC-Versorgung, DC-Eingänge, Relaisausgänge

Bestellnummer: 6ES7 212-1BA01-0XB0

Leistungsmerkmale		Eingänge	
Abmessungen (B x H x T)	160 x 80 x 62 mm	Eingangstyp (IEC 1131-2)	Typ 1, stromziehend
Gewicht	0,4 kg	Bereich bei EIN	15 bis 30 V DC, min. 4 mA 35 V DC, 500 ms Sp.stoß
Stromaufnahme	6 W	Nennspannung bei EIN	DC 24 V, 7 mA
Größe Anwenderpr./Speicher	512 Wörter/EEPROM	Maximum bei AUS	DC 5 V, 1 mA
Größe Anw.daten/Speicher	512 Wörter/RAM	Ansprechzeit E0.0 bis E0.7	max. 0,3 ms
Datenhaltung	typ. 50 h (min. 8 h bei 40 °C)	Potentialtrennung	500 V AC, 1 min
Integrierte E/A ¹	8 Eingänge/6 Ausgänge	Stromversorgung	
Maximale Anzahl Erweiterungsmodule	2	Spannungs-/Frequenzbereich	85 - 264 V AC bei 47 - 63 Hz
Unterstützte digitale E/A	64 Eingänge/64 Ausgänge	Eingangsstrom	typ. 4 VA, nur CPU 50 VA max. Laststrom
Unterstützte analoge E/A	16 Eingänge/16 Ausgänge	Verzögerungszeit	aus 110 V AC mind. 20 ms
Boolesche Ausführungszeiten	1,2 µs/Operation	Einschaltstromstoß	20 A Spitze bei 264 V AC
Interne Merker	128	Schmelzsicherung (nicht austauschbar)	2 A, 250 V, träge
Zeiten	64 Zeiten	Strom 5 V DC	260 mA für CPU 340 mA für Erweiter.module
Zähler	64 Zähler	Elektrisch getrennt	Ja, Transformator, 1500 V AC, 1 min
Schnelle Zähler	1 SW-Zähler (max. 2 kHz)	DC-Geberversorgung	
Analogpotentiometer	1	Spannungsbereich	20,4 bis 28,8 V DC
Normen	UL 508 CSA C22.2 142 FM Klasse I, Kategorie 2 gemäß VDE 0160 gemäß EG-Richtlinie	Welligkeit / Störströme (<10 MHz)	max. 1 V Spitze-Spitze
Ausgänge		Verfügbarer Strom 24 V DC	180 mA
Ausgangstyp	Relais, Schwachstromkontakt	Kurzschlußstrombegrenzung	< 600 mA
Spannungsbereich	5 bis 30 V DC/250 V AC	Elektrisch getrennt	Nein
Max. Laststrom	2 A/Ausgang, 6 A/Leiter		
Überspannungsstoß	7 A bei geschl. Kontakten		
Isolationswiderstand	min. 100 MΩ (neu)		
Verzögerung Schaltvorgänge	max. 10 ms		
Lebensdauer	10 000 000 mechanisch 100 000 mit Bemessungslast		
Kontaktwiderstand	max. 200 mΩ (neu)		
Elektrische Trennung			
Spule zu Kontakt	1500 V AC, 1 min		
Kontakt zu Kontakt (zw. geöffn. Kontakten)	750 V AC, 1 min		
Kurzschlußschutz	keine		

1 In der CPU sind 8 Eingänge im Prozeßabbild der Eingänge und 8 Ausgänge im Prozeßabbild der Ausgänge für die integrierten Ein- und Ausgänge vorgesehen.

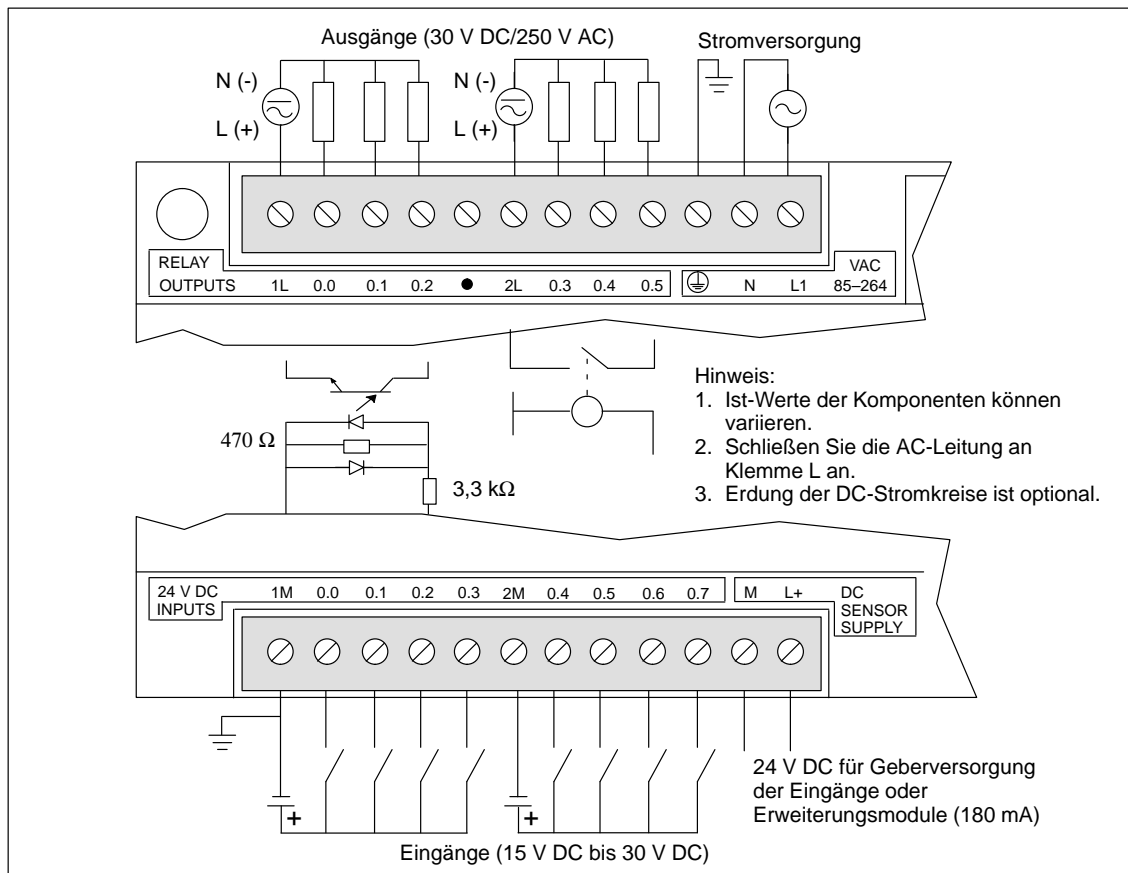


Bild A-3 Kennzeichnung der Anschlüsse bei der CPU 212 AC/DC/Relais

A.4 CPU 212, 24-V-AC-Versorgung, DC-Eingänge, Relaisausgänge

Bestellnummer: **6ES7 212-1FA01-0XB0**

Leistungsmerkmale		Eingänge	
Abmessungen (B x H x T)	160 x 80 x 62 mm	Eingangstyp (IEC 1131-2)	Typ 1, stromziehend
Gewicht	0,4 kg	Bereich bei EIN	15 bis 30 V DC, min. 4 mA 35 V DC, 500 ms Spannungsstoß
Stromaufnahme	6 W	Nennspannung bei EIN	DC 24 V, 7 mA
Größe Anwenderprogramm/ Speicher	512 Wörter/EEPROM	Maximum bei AUS	DC 5 V, 1 mA
Größe Anw.daten/Speicher Datenhaltung	512 Wörter/RAM typ. 50 h (min. 8 h bei 40 °C)	Ansprechzeit E0.0 bis E0.7	max. 0,3 ms
Integrierte E/A ¹	8 Eingänge/6 Ausgänge	Potentialtrennung	500 V AC, 1 min
Maximale Anzahl Erweiterungsmodule	2	Stromversorgung	
Unterstützte digitale E/A	64 Eingänge/64 Ausgänge	Spannungs-/Frequenzbereich	20 - 29 V AC bei 47 - 63 Hz
Unterstützte analoge E/A	16 Eingänge/16 Ausgänge	Eingangsstrom	typ. 4 VA, nur CPU 50 VA max. Laststrom
Boolesche Ausführungszeit	1,2 µs/Operation	Verzögerungszeit	aus 24 V AC mind. 20 ms
Interne Merker	128	Einschaltstromstoß	20 A Spitze bei 29 V AC
Zeiten	64 Zeiten	Schmelzsicherung (nicht austauschbar)	2 A, 250 V, träge
Zähler	64 Zähler	Strom 5 V DC	260 mA für CPU 340 mA für Erweiterungsmodule
Schnelle Zähler	1 SW-Zähler (max. 2 kHz)	Elektrisch getrennt	Ja, Transformator, 500 V AC, 1 min
Analogpotentiometer	1	DC-Geberversorgung	
Normen	UL 508 CSA C22.2 142 FM Klasse I, Kategorie 2 gemäß VDE 0160 gemäß EG-Richtlinie	Spannungsbereich	20,4 bis 28,8 V DC
Ausgänge		Welligkeit/Störströme (<10 MHz)	max. 1 V Spitze-Spitze
Ausgangstyp	Relais, Schwachstromkontakt	Verfügbarer Strom 24 V DC	180 mA
Spannungsbereich	5 bis 30 V DC/250 V AC	Kurzschlußstrombegrenzung	< 600 mA
Max. Laststrom	2 A/Ausgang, 6 A/Leiter	Elektrisch getrennt	Nein
Überspannungsstoß	7 A bei geschl. Kontakten		
Isolationswiderstand	min. 100 MΩ (neu)		
Verzögerung Schaltvorgänge	max. 10 ms		
Lebensdauer	10 000 000 mechanisch 100 000 mit Bemessungslast		
Kontaktwiderstand	max. 200 mΩ (neu)		
Elektrische Trennung			
Spule zu Kontakt	1500 V AC, 1 min		
Kontakt zu Kontakt (zw. geöffn. Kontakten)	750 V AC, 1 min		
Kurzschlußschutz	keine		

1 In der CPU sind 8 Eingänge im Prozeßabbild der Eingänge und 8 Ausgänge im Prozeßabbild der Ausgänge für die integrierten Ein- und Ausgänge vorgesehen.

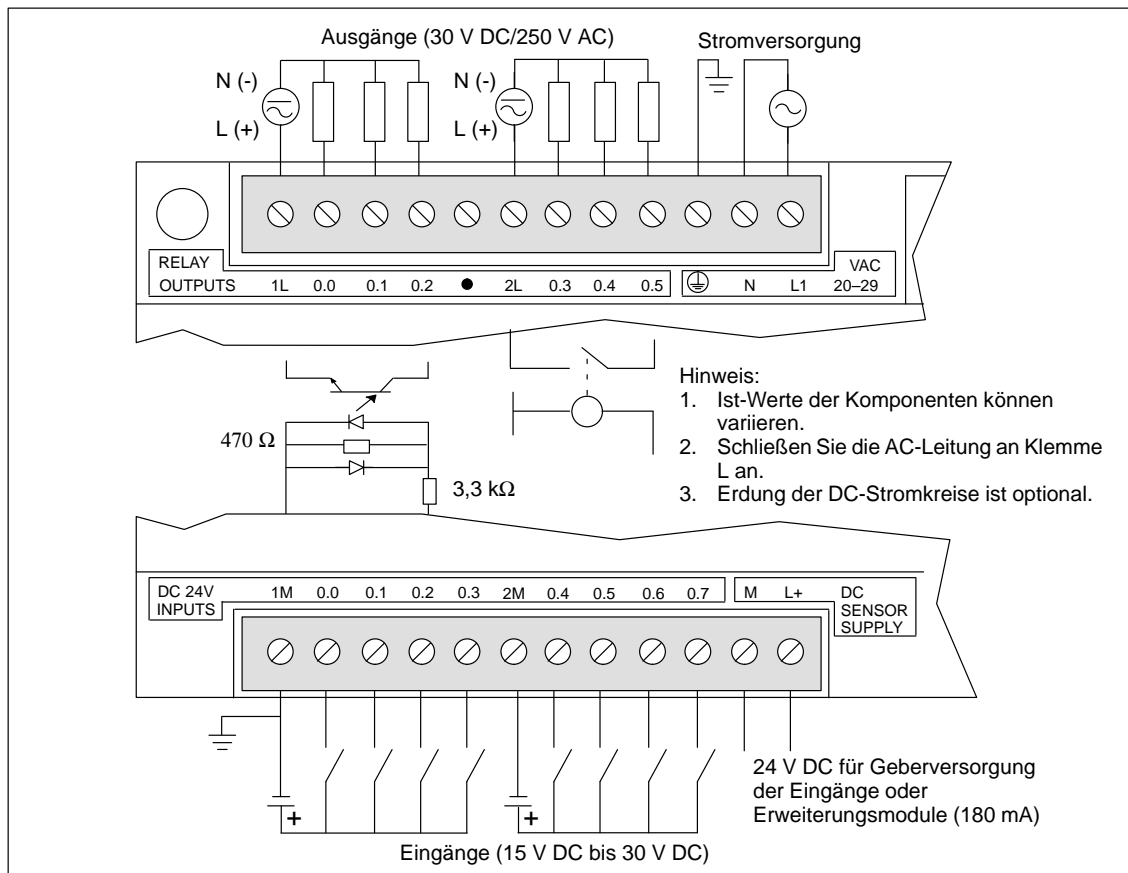


Bild A-4 Kennzeichnung der Anschlüsse bei der CPU 212 24-V-AC/DC/Relais

A.5 CPU 212, AC-Versorgung, AC-Eingänge, AC-Ausgänge

Bestellnummer: 6ES7 212-1CA01-0XB0

Leistungsmerkmale		Ausgänge, Fortsetzung	
Abmessungen (B x H x T)	160 x 80 x 62 mm	Verzögerung Schaltvorgänge	1/2 Zyklus
Gewicht	0,4 kg	Stoßstrom	30 A Spitze, 1 Zyklus/ 10 A Spitze, 5 Zyklen
Stromaufnahme	7 W bei 2,5 A Last	Spannungsabfall	max. 1,5 V bei maximalem Strom
Größe Anwenderprogramm/ Speicher	512 Wörter/EEPROM	Potentialtrennung	1500 V AC, 1 min
Größe Anw.daten/Speicher Datenhaltung	512 Wörter/RAM typ. 50 h (min. 8 h bei 40 °C)	Kurzschlußschutz	keine
Integrierte E/A ¹	8 Eingänge/6 Ausgänge	Eingänge	
Maximale Anzahl Erweiterungsmodule	2	Eingangstyp (IEC 1131-2)	Typ 1, stromziehend
Unterstützte digitale E/A	64 Eingänge/64 Ausgänge	Bereich bei EIN	79 - 135 V AC, 47 - 63 Hz, min. 4 mA
Unterstützte analoge E/A	16 Eingänge/16 Ausgänge	Nennspannung bei EIN	AC 120 V, 60 Hz, 7 mA
Boolesche Ausführungszeiten	1,2 µs/Operation	Maximum bei AUS	AC 20 V, 1 mA
Interne Merker	128	Ansprechzeit	typ. 10 ms, max. 15 ms
Zeiten	64 Zeiten	Potentialtrennung	1500 V AC, 1 min
Zähler	64 Zähler	Stromversorgung	
Schnelle Zähler	1 SW-Zähler (max. 50 Hz)	Spannungs-/Frequenzbereich	85 - 264 V AC bei 47 - 63 Hz
Analogpotentiometer	1	Eingangsstrom	typ. 4 VA, nur CPU 50 VA max. Laststrom
Normen	UL 508 CSA C22.2 142 FM Klasse I, Kategorie 2 gemäß EG-Richtlinie	Verzögerungszeit	aus 110 V AC mind. 20 ms
Ausgänge		Einschaltstromstoß	20 A Spitze bei 264 V AC
Ausgangstyp	Triac, Nulldurchgang	Schmelzsicherung (nicht austauschbar)	2 A, 250 V, träge
Spannungs-/Frequenzbereich	20 - 264 V AC, 47 - 63 Hz	Strom 5 V DC	320 mA für CPU 280 mA für Erweiterungsmodule
Leistungsfaktor Laststromkreis	0,3 bis 1,0	Elektrisch getrennt	Ja, Transformator, 1500 V AC, 1 min
Klemmung induktive Last	MOV 275 V Arbeitsspannung	DC-Geberversorgung	
Max. Laststrom pro einzeltem Ausgang	0 bis 40° C 55° C ² 1,20 A 1,00 A	Spannungsbereich	20,4 bis 28,8 V DC
pro 2 benachb. Ausgänge	1,50 A 1,25 A	Welligkeit/Störströme (<10 MHz)	max. 1 V Spitze-Spitze
Ausgänge gesamt*	3,50 A 2,50 A	Verfügbarer Strom 24 V DC	180 mA
Min. Laststrom	30 mA	Kurzschlußstrombegrenzung	< 600 mA
Ableitstrom	1,5 mA, 120 V AC/2,0 mA, 240 V AC	Elektrisch getrennt	Nein

¹ In der CPU sind 8 Eingänge im Prozeßabbild der Eingänge und 8 Ausgänge im Prozeßabbild der Ausgänge für die integrierten Ein- und Ausgänge vorgesehen.

² Lineare Leistungsminderung 40 bis 55° C. Leistungsminderung bei vertikalem Einbau 10° C.

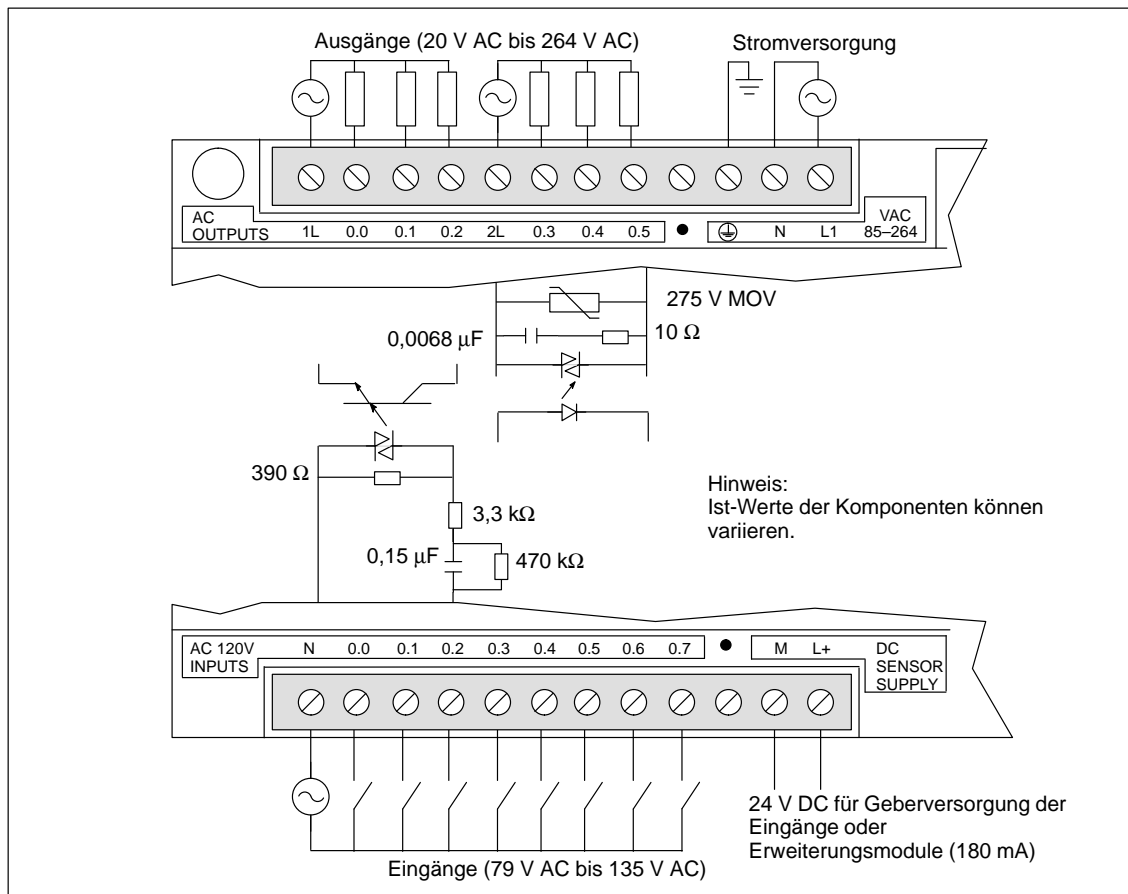


Bild A-5 Kennzeichnung der Anschlüsse bei der CPU 212 AC/AC/AC

A.6 CPU 212 AC-Versorgung, DC-Eingänge (stromliefernd), Relaisausgänge

Bestellnummer: 6ES7 212-1BA10-0XB0

Leistungsmerkmale		Eingänge	
Abmessungen (B x H x T)	160 x 80 x 62 mm	Datentyp	stromliefernd
Gewicht	0,4 kg	Bereich Eingangsspannung	15 - 30 V DC, 35 V DC bei 500 ms
Stromaufnahme	6 W	Nennspannung bei EIN	min. 4 mA
Größe Anwenderprogramm/ Speicher	512 Wörter/EEPROM	Maximum bei AUS	1 mA
Größe Anwenderdaten/ Speicher	512 Wörter/RAM	Ansprechzeit E0.0 bis E0.7	max. 0,3 ms
Datenhaltung	typ. 50 h (min. 8 h bei 40 °C)	Potentialtrennung	500 V AC, 1 min
Integrierte E/A ¹	8 Eingänge/6 Ausgänge	Stromversorgung	
Maximale Anzahl Erweiterungsmodule	2	Spannungs-/Frequenzbereich	85 - 264 V AC bei 47 - 63 Hz
Unterstützte digitale E/A	64 Eingänge/64 Ausgänge	Eingangsstrom	typ. 4 VA, nur CPU 50 VA max. Laststrom
Unterstützte analoge E/A	16 Eingänge/16 Ausgänge	Verzögerungszeit	aus 110 V AC mind. 20 ms
Boolesche Ausführungszeiten	1,2 µs/Operation	Einschaltstromstoß	20 A Spitze bei 264 V AC
Interne Merker	128	Schmelzsicherung (nicht austauschbar)	2 A, 250 V, träge
Zeiten	64 Zeiten	Verfügbarer Strom 5 V DC	260 mA für CPU 340 mA für Erweiterungsmodule
Zähler	64 Zähler	Elektrisch getrennt	Ja, Transformator, 1500 V AC, 1 min
Schnelle Zähler	1 SW-Zähler (max. 2 kHz)	DC-Geberversorgung	
Analogpotentiometer	1	Spannungsbereich	20,4 bis 28,8 V DC
Normen	UL 508 CSA C22.2 142 FM Klasse I, Kategorie 2 gemäß VDE 0160 gemäß EG-Richtlinie	Welligkeit/Störströme (<10 MHz)	max. 1 V Spitze-Spitze
Ausgänge		Verfügbarer Strom 24 V DC	180 mA
Ausgangstyp	Relais, Schwachstromkontakt	Kurzschlußstrombegrenzung	< 600 mA
Spannungsbereich	5 bis 30 V DC/250 V AC	Elektrisch getrennt	Nein
Max. Laststrom	2 A/Ausgang, 6 A/Leiter		
Überspannungsstoß	7 A bei geschl. Kontakten		
Isolationswiderstand	min. 100 MΩ (neu)		
Verzögerung Schaltvorgänge	max. 10 ms		
Lebensdauer	10 000 000 mechanisch 100 000 mit Bemessungslast		
Kontaktwiderstand	max. 200 mΩ (neu)		
Elektrische Trennung			
Spule zu Kontakt	1500 V AC, 1 min		
Kontakt zu Kontakt (zwischen geöffneten Kontakten)	750 V AC, 1 min		
Kurzschlußschutz	keine		

¹ In der CPU sind 8 Eingänge im Prozeßabbild der Eingänge und 8 Ausgänge im Prozeßabbild der Ausgänge für die integrierten Ein- und Ausgänge vorgesehen.

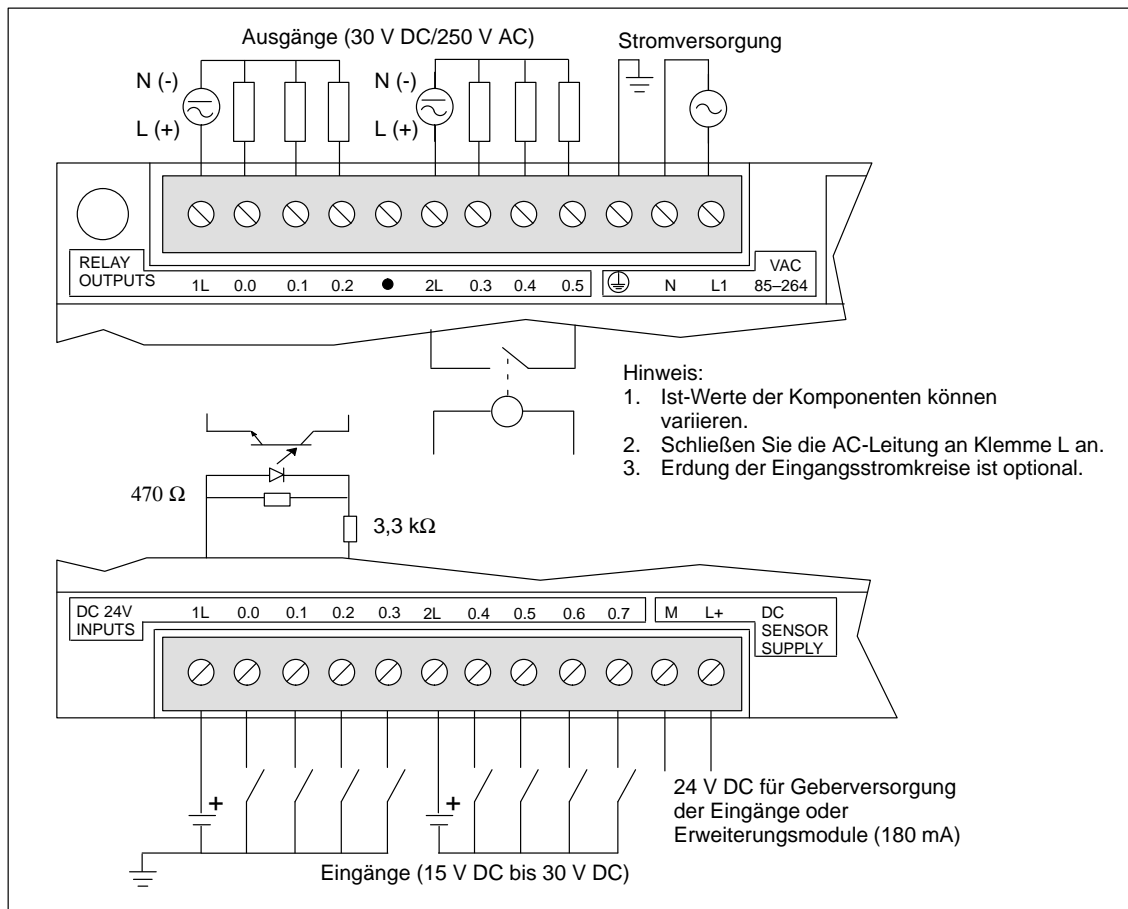


Bild A-6 Kennzeichnung der Anschlüsse bei der CPU 212 AC/DC (stromliefernd)/Relais

A.7 CPU 212, AC-Versorgung, 24-V-AC-Eingänge, AC-Ausgänge

Bestellnummer: 6ES7 212-1DA01-0XB0

Leistungsmerkmale		Ausgänge, Fortsetzung	
Abmessungen (B x H x T)	160 x 80 x 62 mm	Verzögerung Schaltvorgänge	1/2 Zyklus
Gewicht	0,4 kg	Stoßstrom	30 A Spitze, 1 Zyklus/ 10 A Spitze, 5 Zyklen
Stromaufnahme	7 W bei 2,5 A Last	Spannungsabfall	max. 1,5 V bei maximalem Strom
Größe Anwenderprogramm/ Speicher	512 Wörter/EEPROM	Potentialtrennung	1500 V AC, 1 min
Größe Anw.daten/Speicher Datenhaltung	512 Wörter/RAM typ. 50 h (min. 8 h bei 40 °C)	Kurzschlußschutz	keine
Integrierte E/A ¹	8 Eingänge/6 Ausgänge	Eingänge	
Maximale Anzahl Erweiterungsmodule	2	Eingangstyp (IEC 1131-2)	Typ 1, stromziehend
Unterstützte digitale E/A	64 Eingänge/64 Ausgänge	Bereich bei EIN	15 - 30 V DC, 47 - 63 Hz, min. 4 mA
Unterstützte analoge E/A	16 Eingänge/16 Ausgänge	Nennspannung bei EIN	AC 24 V, 60 Hz, 7 mA
Boolesche Ausführungszeiten	1,2 µs/Operation	Maximum bei AUS	AC 5 V, 1 mA
Interne Merker	128	Ansprechzeit	typ. 10 ms, max. 15 ms
Zeiten	64 Zeiten	Potentialtrennung	1500 V AC, 1 min
Zähler	64 Zähler	Stromversorgung	
Schnelle Zähler	1 SW-Zähler (max. 50 Hz)	Spannungs-/Frequenzbereich	85 - 264 V AC bei 47 - 63 Hz
Analogpotentiometer	1	Eingangsstrom	typ. 4 VA, nur CPU 50 VA max. Laststrom
Normen	UL 508 CSA C22.2 142 FM Klasse I, Kategorie 2 gemäß EG-Richtlinie	Verzögerungszeit	aus 110 V AC mind. 20 ms
Ausgänge		Einschaltstromstoß	20 A Spitze bei 264 V AC
Ausgangstyp	Triac, Nulldurchgang	Schmelzsicherung (nicht austauschbar)	2 A, 250 V, träge
Spannungs-/Frequenzbereich	20 - 264 V AC, 47 - 63 Hz	Strom 5 V DC	320 mA für CPU 280 mA für Erweiterungsmodule
Leistungsfaktor Laststromkreis	0,3 bis 1,0	Elektrisch getrennt	Ja, Transformator, 1500 V AC, 1 min
Klemmung induktive Last	MOV 275 V Arbeitsspannung	DC-Geberversorgung	
Max. Laststrom pro einzeltem Ausgang	0 bis 40° C 55° C ² 1,20 A 1,00 A	Spannungsbereich	20,4 bis 28,8 V DC
pro 2 benachb. Ausgänge	1,50 A 1,25 A	Welligkeit/Störströme (<10 MHz)	max. 1 V Spitze-Spitze
Ausgänge gesamt	3,50 A 2,50 A	Verfügbarer Strom 24 V DC	180 mA
Min. Laststrom	30 mA	Kurzschlußstrombegrenzung	< 600 mA
Ableitstrom	1,5 mA, 120 V AC/2,0 mA, 240 V AC	Elektrisch getrennt	Nein

¹ In der CPU sind 8 Eingänge im Prozeßabbild der Eingänge und 8 Ausgänge im Prozeßabbild der Ausgänge für die integrierten Ein- und Ausgänge vorgesehen.

² Lineare Leistungsminderung 40 bis 55° C. Leistungsminderung bei vertikalem Einbau 10° C.

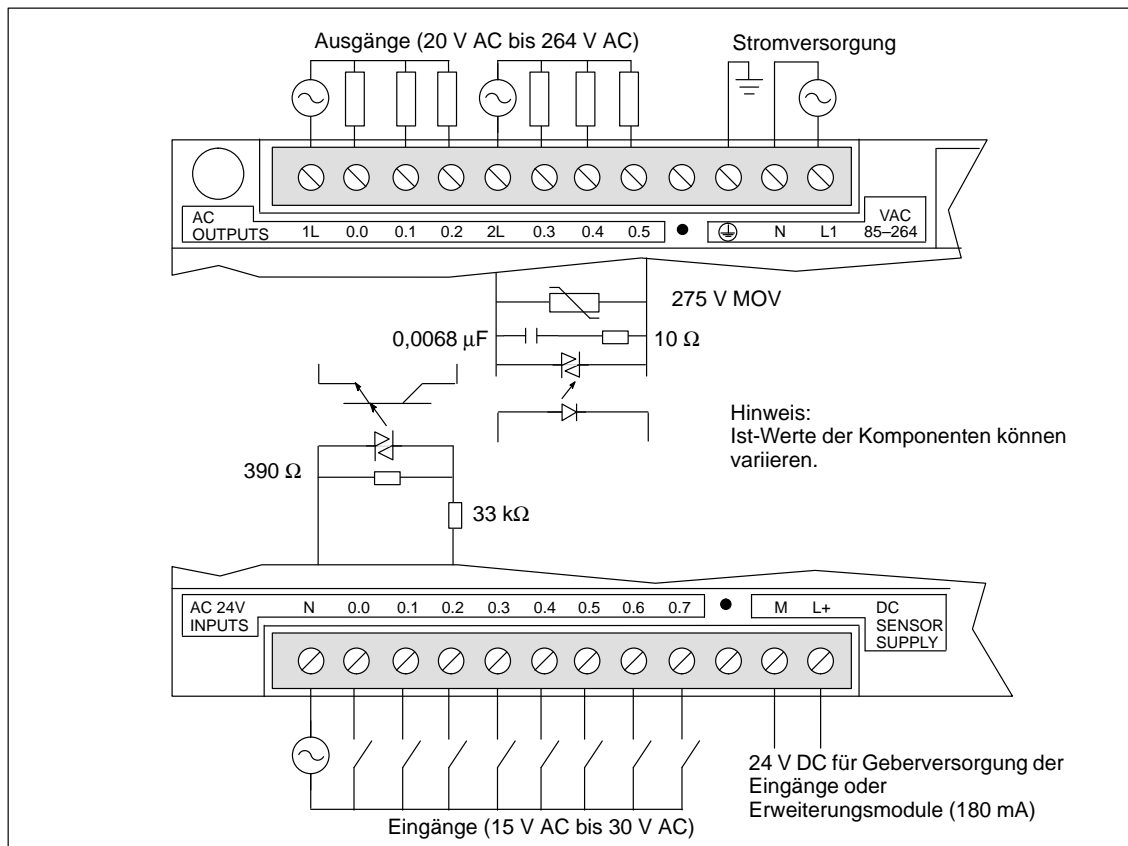


Bild A-7 Kennzeichnung der Anschlüsse bei der CPU 212 AC/AC/AC

A.8 CPU 212, AC-Versorgung, AC-Eingänge, Relaisausgänge

Bestellnummer: 6ES7 212-1GA01-0XB0

Leistungsmerkmale		Eingänge	
Abmessungen (B x H x T)	160 x 80 x 62 mm	Eingangstyp (IEC 1131-2)	Typ 1, stromziehend
Gewicht	0,4 kg	Bereich bei EIN	79 - 135 V AC, 47 - 63 Hz min. 4 mA
Stromaufnahme	6 W	Nennspannung bei EIN	AC 120 V, 60 Hz, 7 mA
Größe Anwenderprogramm/ Speicher	512 Wörter/EEPROM	Maximum bei AUS	AC 20 V, 1 mA
Größe Anw.daten/Speicher Datenhaltung	512 Wörter/RAM typ. 50 h (min. 8 h bei 40 °C)	Ansprechzeit	typ. 10 ms, max. 15 ms
Integrierte E/A ¹	8 Eingänge/6 Ausgänge	Potentialtrennung	AC 1500 V, 1 Minute
Maximale Anzahl Erweiterungsmodule	2	Stromversorgung	
Unterstützte digitale E/A	64 Eingänge/64 Ausgänge	Spannungs-/Frequenzbereich	85 - 264 V AC bei 47 - 63 Hz
Unterstützte analoge E/A	16 Eingänge/16 Ausgänge	Eingangsstrom	typ. 4 VA, nur CPU 50 VA max. Laststrom
Ausführungszeit	1,2 µs/Operation	Verzögerungszeit	aus 110 V AC mind. 20 ms
Interne Merker	128	Einschaltstromstoß	20 A Spitze bei 264 V AC
Zeiten	64 Zeiten	Schmelzsicherung (nicht austauschbar)	2 A, 250 V, träge
Zähler	64 Zähler	Verfügbare Strom 5 V DC	260 mA für CPU 340 mA für Erweiterungsmodule
Schnelle Zähler	1 SW-Zähler (max. 2 kHz)	Elektrisch getrennt	Ja, Transformator, AC 1500 V, 1 Minute
Analogpotentiometer	1	DC-Geberversorgung	
Normen	UL 508 CSA C22.2 142 FM Klasse I, Kategorie 2 gemäß VDE 0160 gemäß EG-Richtlinie	Spannungsbereich	20,4 bis 28,8 V DC
Ausgänge		Welligkeit/Störströme (<10 MHz)	max. 1 V Spitze-Spitze
Ausgangstyp	Relais, Schwachstromkontakt	Verfügbare Strom 24 V DC Kurzschlußstrombegrenzung	180 mA < 600 mA
Spannungsbereich	5 bis 30 V DC/250 V AC	Elektrisch getrennt	Nein
Max. Laststrom	2 A/Ausgang		
Überspannungsstoß	7 A bei geschl. Kontakten		
Isolationswiderstand	min. 100 MΩ (neu)		
Verzögerung Schaltvorgänge	max. 10 ms		
Lebensdauer	10 000 000 mechanisch 100 000 mit Bemessungslast		
Kontaktwiderstand	max. 200 mΩ (neu)		
Elektrische Trennung			
Spule zu Kontakt	AC 1500 V, 1 Minute		
Kontakt zu Kontakt	AC 1000 V, 1 Minute		
Kurzschlußschutz	keine		

¹ In der CPU sind 8 Eingänge im Prozeßabbild der Eingänge und 8 Ausgänge im Prozeßabbild der Ausgänge für die integrierten Ein- und Ausgänge vorgesehen.

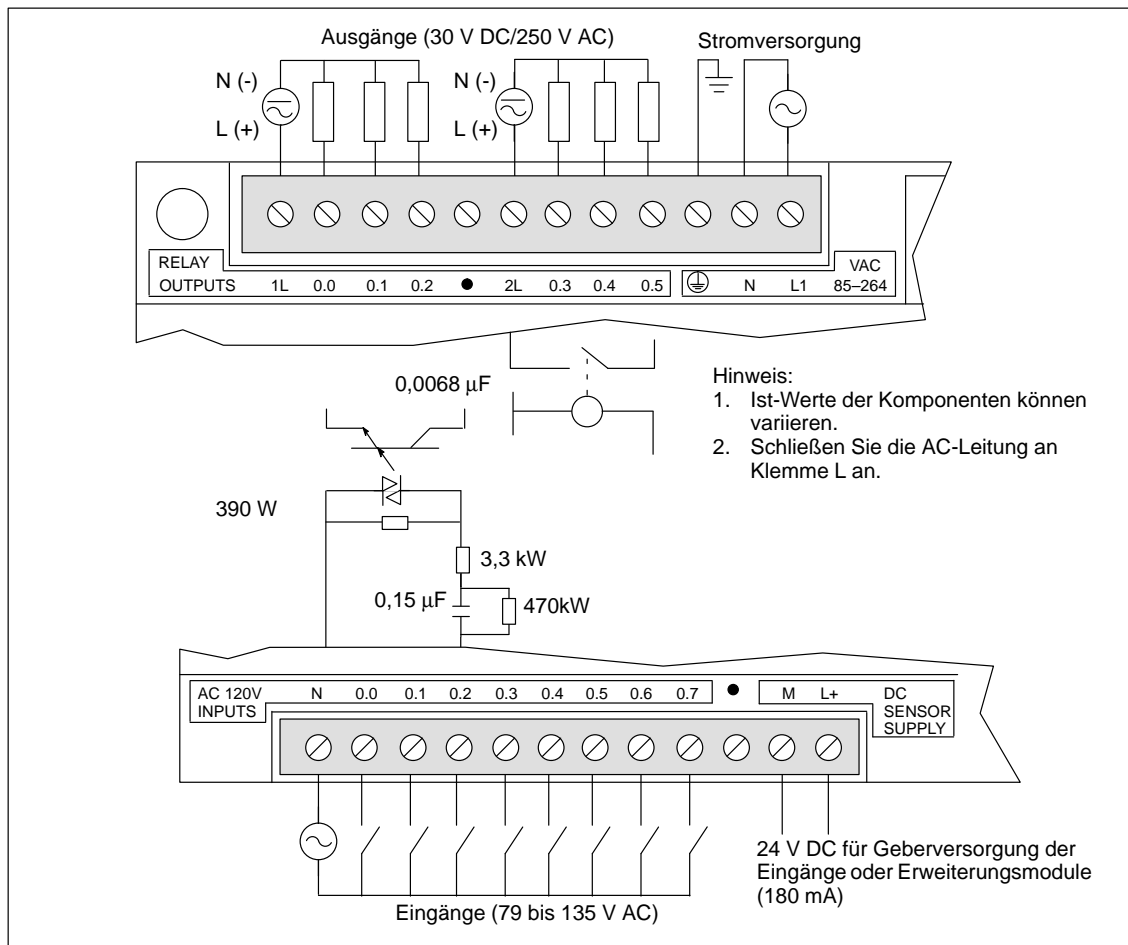


Bild A-8 Kennzeichnung der Anschlüsse bei der CPU 212 AC/AC/Relais

A.9 CPU 214, DC-Versorgung, DC-Eingänge, DC-Ausgänge

Bestellnummer: 6ES7 214-1AC01-0XB0

Leistungsmerkmale	
Abmessungen (B x H x T)	197 x 80 x 62 mm
Gewicht	0,4 kg
Stromaufnahme	8 W bei 3 A Last
Größe Anwenderprogramm / Speicher	2 K Wörter/EEPROM
Größe Anw.daten/Speicher	2 K Wörter/RAM
Pufferung Daten/Echtzeituhr Hochleistungskondensator	typ. 190 h (min. 120 h bei 40° C)
Batteriemodul (optional)	200 Tage bei ständig. Einsatz
Integrierte E/A ¹	14 Eingänge/10 Ausgänge
Max. Anzahl Erweiterungsmodule	7
Unterstützte digitale E/A	64 Eingänge/64 Ausgänge
Unterstützte analoge E/A	16 Eingänge/16 Ausgänge
Boolesche Ausführungszeiten	0,8µs/Operation
Interne Merker	256
Zeiten	128 Zeiten
Zähler	128 Zähler
Schnelle Zähler	1 SW-Zähler (max. 2 kHz) 2 HW-Zähler (max. je 7 kHz)
Toleranz Echtzeituhr	6 Minuten pro Monat
Impulsausgänge	2 (max. jeweils 4 kHz)
Analogpotentiometer	2
Normen	UL 508 CSA C22.2 142 FM Klasse I, Kategorie 2 gemäß VDE 0160/gemäß EG-Richtlinie
Eingänge	
Eingangstyp (IEC 1131-2)	Typ 1, stromziehend
Bereich bei EIN	15 bis 30 V DC, min. 4 mA 35 V DC, 500 ms Spann.stoß
Nennspannung bei EIN	24 V DC, 7 mA
Maximum bei AUS	5 V DC, 1 mA
Maximale Ansprechzeit	
E0.0 bis E1.5	0,2 ms bis 8,7 ms wählbar
E0.6 bis E1.5 wie bei HSC1 und HSC2	0,2 ms standardmäßig typ. 30 µs/max. 70 µs
Potentialtrennung	500 V AC, 1 min
Ausgänge	
Ausgangstyp	Transistor, stromliefernd
Spannungsbereich	20,4 bis 28,8 V DC
Max. Laststrom	0 bis 40° C 55° C ²
pro einzeltem Ausgang	0,75 A 0,50 A
pro 2 benachb. Ausgänge	1,00 A 0,75 A
Ausgänge gesamt	4,00 A 3,00 A
Klemmung induktive Last einzelner Impuls	(pro Leiter) 2A L/R = 10 ms 1A L/R = 100 ms
Wiederholung	1 W Energieaufnahme (1/2 Li ² x Schaltfr. < 1 W)
Ableitstrom	100 µA
Verzögerung Schaltvorgänge	25 µs EIN, 120 µs AUS
Stoßstrom	4 A, 100 ms
Spannungsabfall	max. 1,8 V bei max. Strom
Potentialtrennung	500 V AC, 1 min
Kurzschlußschutz	Nein
Stromversorgung	
Spannungsbereich	20,4 bis 28,8 V DC
Eingangsstrom	typ. 85 mA, nur CPU 900 mA max. Laststrom
UL/CSA-Bemessung	50 VA
Verzögerungszeit	aus 24 V DC min. 10 ms
Einschaltstromstoß	10 A Spitze bei 28,8 V DC
Schmelzsicherung (nicht austauschbar)	1 A, 125 V, träge
Strom 5 V DC	340 mA für CPU; 660 mA für Erweiterungsmodule
Elektrisch getrennt	Nein
DC-Geberversorgung	
Spannungsbereich	16,4 bis 28,8 V DC
Welligkeit/Störströme (<10 MHz)	wie zugeführte Spannung
Verfügbarer Strom 24 V DC	280mA
Kurzschlußstrombegrenzung	< 600 mA
Elektrisch getrennt	Nein

¹ In der CPU sind 16 Eingänge im Prozeßabbild der Eingänge und 16 Ausgänge im Prozeßabbild der Ausgänge für die integrierten Ein- und Ausgänge vorgesehen.

² Lineare Leistungsminderung 40 bis 55° C, Leistungsminderung bei vertikalem Einbau 10° C.

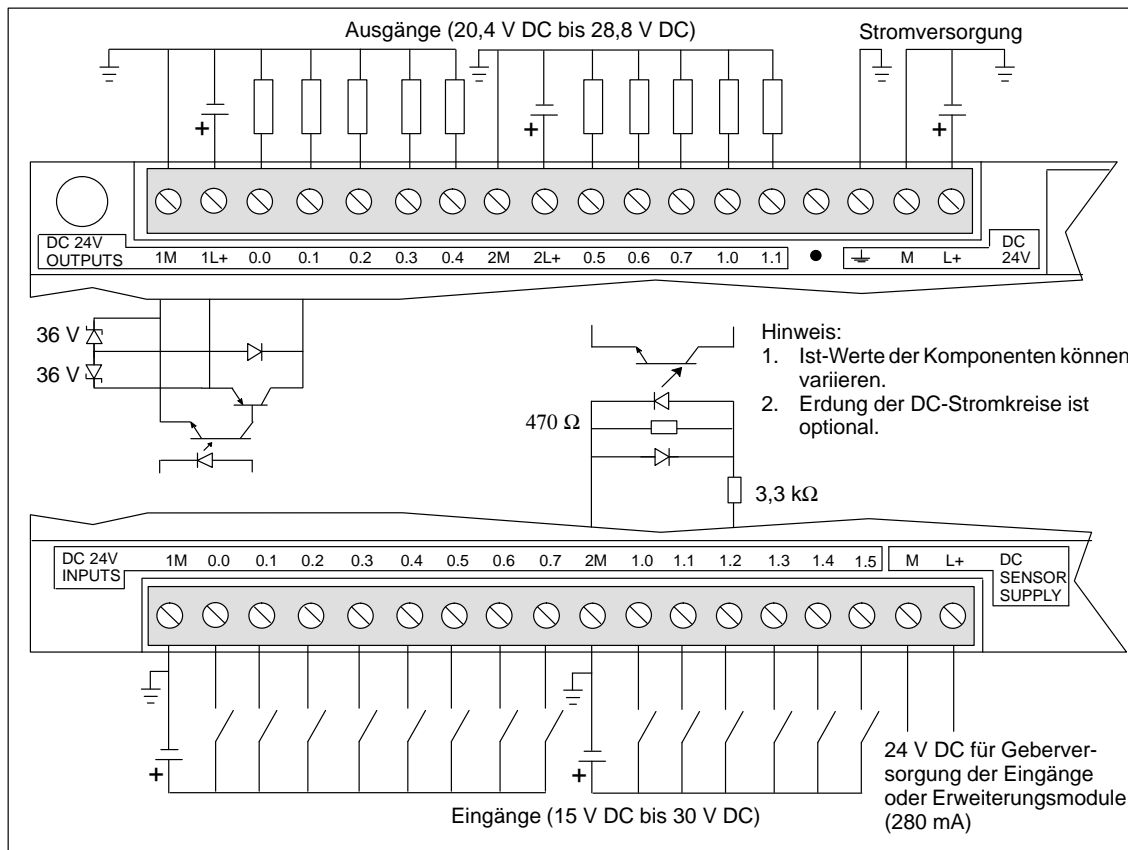


Bild A-9 Kennzeichnung der Anschlüsse bei der CPU 214 DC/DC/DC

A.10 CPU 214, AC-Versorgung, DC-Eingänge, Relaisausgänge

Bestellnummer: **6ES7 214-1BC01-0XB0**

Leistungsmerkmale		Ausgänge	
Abmessungen (B x H x T)	197 x 80 x 62 mm	Ausgangstyp	Relais, Schwachstromkontakt
Gewicht	0,5 kg	Spannungsbereich	5 bis 30 V DC/250 V AC
Stromaufnahme	9 W	Max. Laststrom	2 A/Ausgang, 8 A/Leiter
Größe Anwenderpr./Speicher	2 K Wörter/EEPROM	Überspannungsstoß	7 A bei geschl. Kontakten
Größe Anw.daten/Speicher	2 K Wörter/RAM	Isolationswiderstand	min. 100 MΩ (neu)
Pufferung Daten/Echtzeituhr Hochleistungskondensator	typ. 190 h (min. 120 h bei 40 °C)	Verzögerung Schaltvorgänge	max. 10 ms
Batteriemodul (optional)	200 Tage bei ständig. Einsatz	Lebensdauer	10 000 000 mechanisch 100 000 mit Bemessungslast
Integrierte E/A ¹	14 Eingänge/10 Ausgänge	Kontaktwiderstand	max. 200 mW (neu)
Max. Anzahl Erweiterungsmodule	7	Elektrische Trennung Spule zu Kontakt Kontakt zu Kontakt (zw. geöffn. Kontakten)	1500 V AC, 1 min 750 V AC, 1 min
Unterstützte digitale E/A	64 Eingänge/64 Ausgänge	Kurzschlußschutz	keine
Unterstützte analoge E/A	16 Eingänge/16 Ausgänge	Stromversorgung	
Boolesche Ausführungszeiten	0,8µs/Operation	Spannungs-/Frequenzbereich	85 - 264 V AC bei 47 - 63 Hz
Interne Merker	256	Eingangsstrom	typ. 4,5 VA, nur CPU 50 VA max. Laststrom
Zeiten	128 Zeiten	Verzögerungszeit	aus 110 V AC min. 20 ms
Zähler	128 Zähler	Einschaltstromstoß	20 A Spitze bei 264 V AC
Schnelle Zähler	1 SW-Zähler (max. 2 kHz) 2 HW-Zähler (max. je 7 kHz)	Schmelzsicherung (nicht austauschbar)	2 A, 250 V, träge
Toleranz Echtzeituhr	6 Minuten pro Monat	Strom 5 V DC	340 mA für CPU; 660 mA für Erweiterungsmodule
Impulsausgänge	nicht empfohlen	Elektrisch getrennt	Ja, Transformator, 1500 V AC, 1 min
Analogpotentiometer	2	DC-Geberversorgung	
Normen	UL 508 CSA C22.2 142 FM Kl. I, Kat. 2, gem. VDE 0160 gem. EG-Richtlinie	Spannungsbereich	20,4 bis 28,8 V DC
Eingänge		Welligkeit / Störströme (<10 MHz)	max. 1 V Spitze-Spitze
Eingangstyp (IEC 1131-2)	Typ 1, stromziehend	Verfügbarer Strom 24 V DC	280mA
Bereich bei EIN	15 bis 30 V DC, min. 4 mA 35 V DC, 500 ms Spann.stoß	Kurzschlußstrombegrenzung	< 600 mA
Nennspannung bei EIN	24 V DC, 7 mA	Elektrisch getrennt	Nein
Maximum bei AUS	5 V DC, 1 mA		
Maximale Ansprechzeit E0.0 - E1.5	0,2 ms bis 8,7 ms wählbar 0,2 ms standardmäßig		
E0.6 - E1.5 (wie bei HSC1 und HSC2)	typ. 30 µs/max. 70 µs		
Potentialtrennung	500 V AC, 1 min		

¹ In der CPU sind 16 Eingänge im Prozeßabbild der Eingänge und 16 Ausgänge im Prozeßabbild der Ausgänge für die integrierten Ein- und Ausgänge vorgesehen.

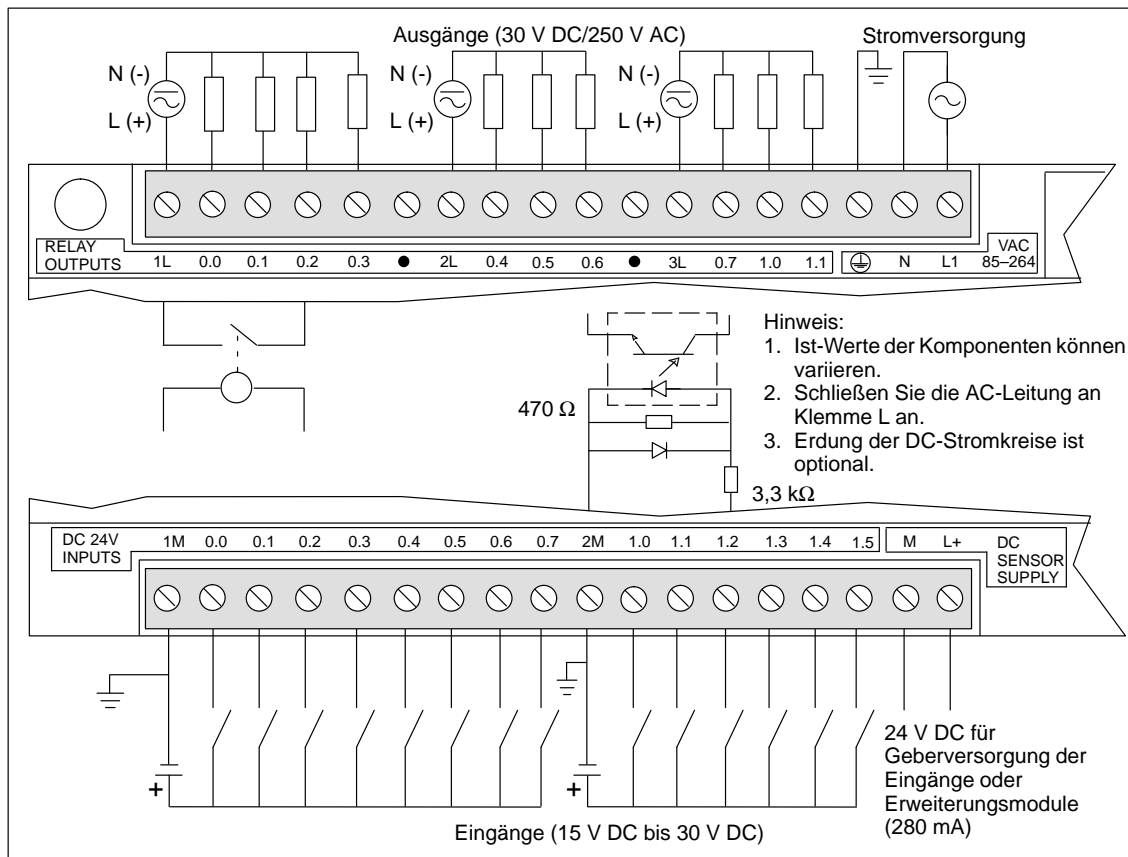


Bild A-10 Kennzeichnung der Anschlüsse bei der CPU 214 AC/DC/Relais

A.11 CPU 214, AC-Versorgung, AC-Eingänge, AC-Ausgänge

Bestellnummer: **6ES7 214-1CC01-0XB0**

Leistungsmerkmale		Ausgänge	
Abmessungen (B x H x T)	197 x 80 x 62 mm	Ausgangstyp	Triac, Nulldurchgang
Gewicht	0,5 kg	Spannungs-/Frequenzbereich	20 - 264 V AC, 47 - 63 Hz
Stromaufnahme	11 W bei 4,25 A Last	Leistungsfaktor	0,3 bis 1,0
Größe Anw.progr./Speicher	2 K Wörter/EEPROM	Laststromkreis	
Größe Anw.daten/Speicher	2 K Wörter/RAM	Klemmung induktive Last	MOV 275 V Arbeitssp.
Pufferung Daten/Echtzeituhr		Max. Laststrom	0 bis 40° C 55° C ²
Hochleistungskondensator	typ. 190 h (min. 120 h bei 40° C)	pro einzeltem Ausgang	1,20 A 1,00 A
Batteriemodul (optional)	200 Tage bei ständig. Einsatz	pro 2 benachb. Ausgänge	1,50 A 1,25 A
Integrierte E/A ¹	14 Eingänge/10 Ausgänge	Ausgänge gesamt	6,00 A 4,25 A
Max. Anz. Erweiterungsmod.	7	Min. Laststrom	30 mA
Unterstützte digitale E/A	64 Eingänge/64 Ausgänge	Ableitstrom	1,5 mA, 120 V AC/2,0 mA, 240 V AC
Unterstützte analoge E/A	16 Eingänge/16 Ausgänge	Verzögerung Schaltvorgänge	1/2 Zyklus
Boolesche Ausführungszeiten	0,8µs/Operation	Stoßstrom	30 A Spitze, 1 Zyklus/ 10 A Spitze, 5 Zyklen
Interne Merker	256	Spannungsabfall	max. 1,5 V bei maximalem Strom
Zeiten	128 Zeiten	Potentialtrennung	1500 V AC, 1 min
Zähler	128 Zähler	Kurzschlußschutz	keine
Schnelle Zähler	1 SW-Zähler (50 Hz) 2 HW-Zähler (je 50 Hz)	Stromversorgung	
Toleranz Echtzeituhr	6 Minuten pro Monat	Spannungs-/Frequenzbereich	85 - 264 V AC bei 47 - 63 Hz
Impulsausgänge	2 (max. jeweils 4 kHz)	Eingangsstrom	typ. 4,5 VA, nur CPU 50 VA max. Laststrom
Analogpotentiometer	2	Verzögerungszeit	aus 110 V AC min. 20 ms
Normen	UL 508 CSA C22.2 142 FM Klasse I, Kategorie 2 gemäß EG-Richtlinie	Einschaltstromstoß	20 A Spitze bei 264 V AC
Eingänge		Schmelzsicherung (nicht austauschbar)	2 A, 250 V, träge
Eingangstyp (IEC 1131-2)	Typ 1, stromziehend	Verfügbare Strom 5 V DC	440 mA für CPU; 560 mA für Erweiterungsmodule
Bereich bei EIN	79 - 135 V DC, 47 - 63 Hz, min. 4 mA	Elektrisch getrennt	Ja, Transformator, 1500 V AC, 1 min
Nennspannung bei EIN	120 V AC, 60 Hz, 7 mA	DC-Geberversorgung	
Maximum bei AUS	20 V AC, 1 mA	Spannungsbereich	20,4 bis 28,8 V DC
Max. Ansprechzeit	0,2 ms bis 8,7 ms wählbar plus 15,0 ms bei festem Filter 15,2 ms standardmäßig	Welligk./Störstr. (<10 MHz)	max. 1 V Spitze-Spitze
Potentialtrennung	1500 V AC, 1 min	Verfügbare Strom 24 V DC	280mA
		Kurzschlußstrombegrenzung	< 600 mA
		Elektrisch getrennt	Nein

¹ In der CPU sind 16 Eingänge im Prozeßabbild der Eingänge und 16 Ausgänge im Prozeßabbild der Ausgänge für die integrierten Ein- und Ausgänge vorgesehen.

² Lineare Leistungsminderung 40 bis 55° C, Leistungsminderung bei vertikalem Einbau 10° C.

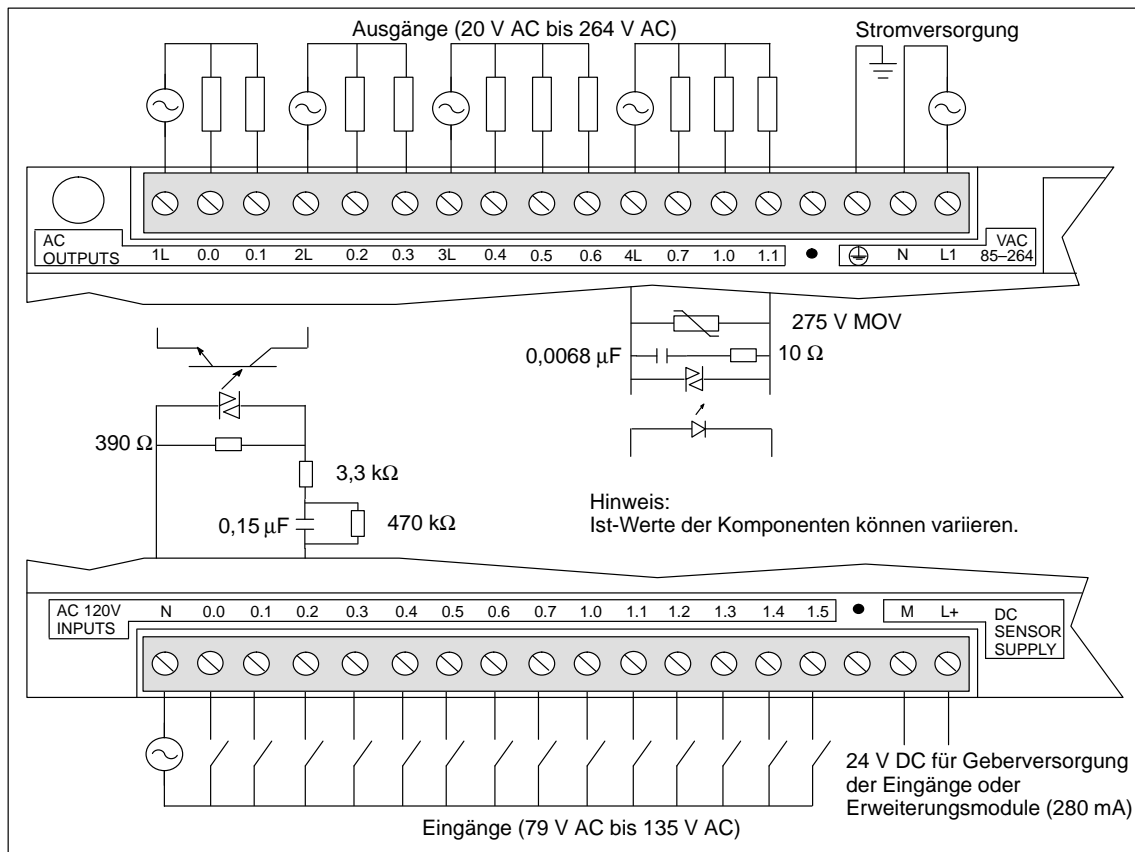


Bild A-11 Kennzeichnung der Anschlüsse bei der CPU 214 AC/AC/AC

A.12 CPU AC-Versorgung, DC-Eingänge (stromliefernd), Relaisausgänge

Bestellnummer: 6ES7 214-1BC10-0XB0

Leistungsmerkmale		Ausgänge	
Abmessungen (B x H x T)	197 x 80 x 62 mm	Ausgangstyp	Relais, Schwachstromkontakt
Gewicht	0,5 kg	Spannungsbereich	5 bis 30 V DC/250 V AC
Stromaufnahme	9 W	Max. Laststrom	2 A/Ausgang, 8 A/Leiter
Größe Anwenderprogramm / Speicher	2 K Wörter/EEPROM	Überspannungsstoß	7 A bei geschl. Kontakten
Größe Anwenderdaten / Speicher	2 K Wörter/RAM	Isolationswiderstand	min. 100 MΩ (neu)
Pufferung Daten/Echtzeituhr Hochleistungskondensator	typ. 190 h (min. 120 h bei 40° C)	Verzögerung Schaltvorgänge	max. 10 ms
Batteriemodul (optional)	200 Tage bei ständig. Einsatz	Lebensdauer	10 000 000 mechanisch 100 000 mit Bemessungslast
Integrierte E/A ¹	14 Eingänge/10 Ausgänge	Kontaktwiderstand	max. 200 mW (neu)
Max. Anzahl Erweiterungsmodule	7	Elektrische Trennung	
Unterstützte digitale E/A	64 Eingänge/ 64 Ausgänge	Spule zu Kontakt	1500 V AC, 1 min
Unterstützte analoge E/A	16 Eingänge/16 Ausgänge	Kontakt zu Kontakt (zwischen geöffneten Kontakten)	750 V AC, 1 min
Boolesche Ausführungszeiten	0,8µs/Operation	Kurzschlußschutz	keine
Interne Merker	256	Stromversorgung	
Zeiten	128 Zeiten	Spannungs-/Frequenzbereich	85 bis 264 V AC bei 47 bis 63 Hz
Zähler	128 Zähler	Eingangsstrom	typ. 4,5 VA, nur CPU 50 VA max. Laststrom
Schnelle Zähler	1 SW-Zähler (max. 2 kHz) 2 HW-Zähler (max. je 7 kHz)	Verzögerungszeit	aus 110 V AC min. 20 ms
Toleranz Echtzeituhr	6 Minuten pro Monat	Einschaltstromstoß	20 A Spitze bei 264 V AC
Impulsausgänge	nicht empfohlen	Schmelzsicherung (nicht austauschbar)	2 A, 250 V, träge
Analogpotentiometer	2	Strom 5 V DC	340 mA für CPU; 660 mA für Erweiterungsmodule
Normen	UL 508 CSA C22.2 142 FM Klasse I, Kategorie 2 gemäß VDE 0160 gemäß EG-Richtlinie	Elektrisch getrennt	Ja, Transformator, 1500 V AC, 1 min
Eingänge		DC-Geberversorgung	
Datentyp	stromliefernd	Spannungsbereich	20,4 bis 28,8 V DC
Bereich Eingangsspannung	15 bis 30 V DC, 35 V DC bei 500 ms	Welligkeit/Störströme (<10 MHz)	max. 1 V Spitze-Spitze
Nennspannung bei EIN	min. 4 mA	Verfügbare Strom 24 V DC	280mA
Maximum bei AUS	1 mA	Kurzschlußstrombegrenzung	< 600 mA
Maximale Ansprechzeit E0.0 bis E1.5	0,2 ms bis 8,7 ms wählbar 0,2 ms standardmäßig	Elektrisch getrennt	Nein
E0.6 bis E1.5 wie bei HSC1 und HSC2	typ. 30 µs/max. 70 µs		
Potentialtrennung	500 V AC, 1 min		

¹ In der CPU sind 16 Eingänge im Prozeßabbild der Eingänge und 16 Ausgänge im Prozeßabbild der Ausgänge für die integrierten Ein- und Ausgänge vorgesehen.

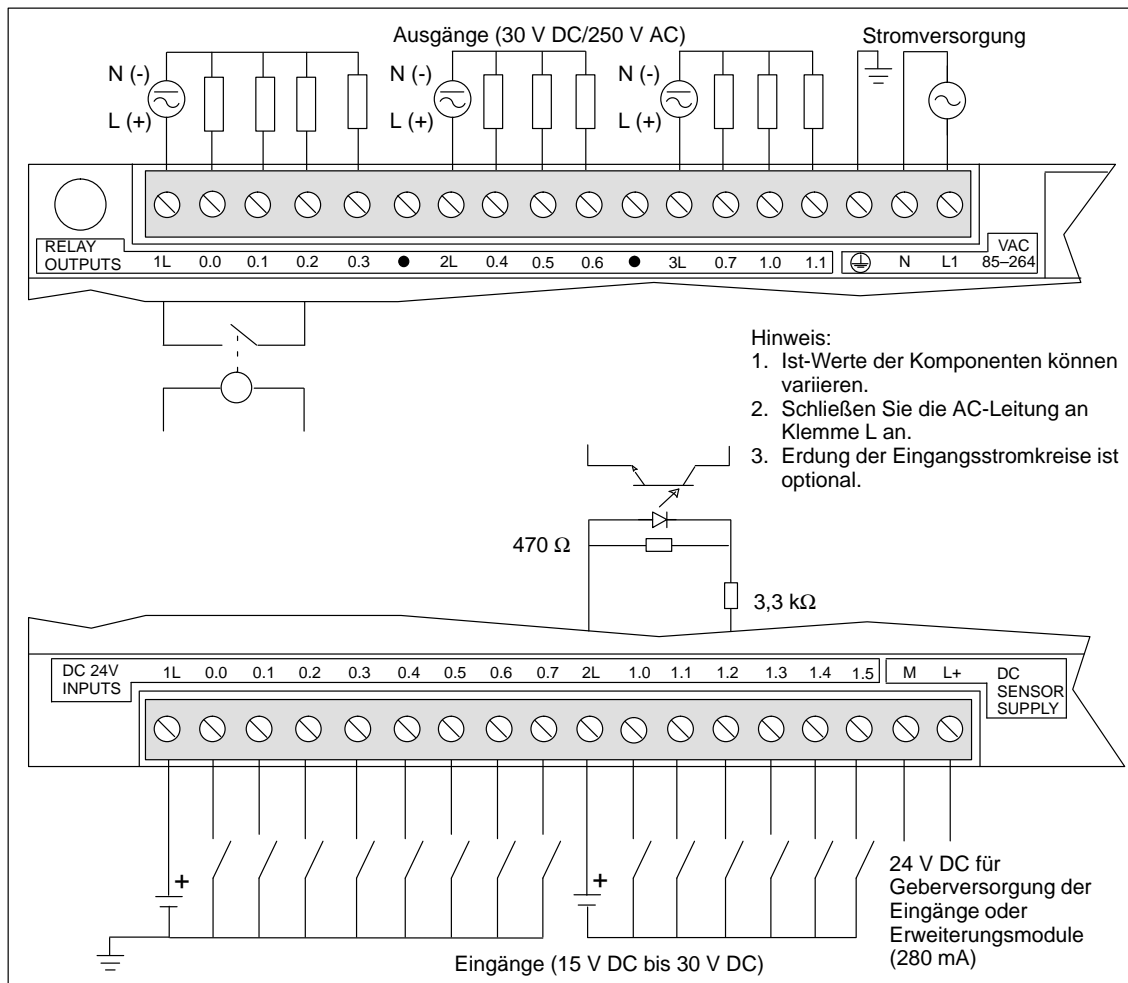


Bild A-12 Kennzeichnung der Anschlüsse bei der CPU 214 AC/DC (stromliefernd)/Relais

A.13 CPU 214, AC-Versorgung, 24-V-AC-Eingänge, AC-Ausgänge

Bestellnummer: **6ES7 214-1DC01-0XB0**

Leistungsmerkmale		Ausgänge	
Abmessungen (B x H x T)	197 x 80 x 62 mm	Ausgangstyp	Triac, Nulldurchgang
Gewicht	0,5 kg	Spannungs-/Frequenzbereich	20 - 264 V AC, 47 - 63 Hz
Stromaufnahme	11 W bei 4,25 A Last	Leistungsfaktor Laststromkreis	0,3 bis 1,0
Größe Anwenderprogramm / Speicher	2 K Wörter/EEPROM	Klemmung induktive Last	MOV 275 V Arbeitsspann.
Größe Anw.daten/Speicher	2 K Wörter/RAM	Max. Laststrom	<u>0 bis 40° C</u> <u>55° C</u> ²
Pufferung Daten/Echtzeituhr Hochleistungskondensator	typ. 190 h (min. 120 h bei 40° C)	pro einzelmem Ausgang	1,20 A 1,00 A
Batteriemodul (optional)	200 Tage bei ständig. Einsatz	pro 2 benachb. Ausgänge	1,50 A 1,25 A
Integrierte E/A ¹	14 Eingänge/10 Ausgänge	Ausgänge gesamt	6,00 A 4,25 A
Max. Anz. Erweiterungsmod.	7	Min. Laststrom	30 mA
Unterstützte digitale E/A	64 Eingänge/64 Ausgänge	Ableitstrom	1,5 mA, 120 V AC/2,0 mA, 240 V AC
Unterstützte analoge E/A	16 Eingänge/16 Ausgänge	Verzögerung Schaltvorgänge	1/2 Zyklus
Boolesche Ausführungszeiten	0,8µs/Operation	Stoßstrom	30 A Spitze, 1 Zyklus/ 10 A Spitze, 5 Zyklen
Interne Merker	256	Spannungsabfall	max. 1,5 V bei maximalem Strom
Zeiten	128 Zeiten	Potentialtrennung	1500 V AC, 1 min
Zähler	128 Zähler	Kurzschlußschutz	keine
Schnelle Zähler	1 SW-Zähler (50 Hz) 2 HW-Zähler (je 50 Hz)	Stromversorgung	
Toleranz Echtzeituhr	6 Minuten pro Monat	Spannungs-/Frequenzbereich	85 - 264 V AC bei 47 - 63 Hz
Impulsausgänge	2 (max. jeweils 4 kHz)	Eingangsstrom	typ. 4,5 VA, nur CPU 50 VA max. Laststrom
Analogpotentiometer	2	Verzögerungszeit	aus 110 V AC min. 20 ms
Normen	UL 508 CSA C22.2 142 FM Klasse I, Kategorie 2 gemäß EG-Richtlinie	Einschaltstromstoß	20 A Spitze bei 264 V AC
Eingänge		Schmelzsicherung (nicht austauschbar)	2 A, 250 V, träge
Eingangstyp (IEC 1131-2)	Typ 1, stromziehend	Verfügbare Strom 5 V DC	440 mA für CPU; 560 mA für Erweiterungsmodule
Bereich bei EIN	15 - 30 V DC, 47 - 63 Hz, min. 4 mA	Elektrisch getrennt	Ja, Transformator, 1500 V AC, 1 min
Nennspannung bei EIN	24 V AC, 60 Hz, 7 mA	DC-Geberversorgung	
Maximum bei AUS	5 V AC, 1 mA	Spannungsbereich	20,4 bis 28,8 V DC
Max. Ansprechzeit	0,2 ms bis 8,7 ms wählbar plus 15,0 ms bei festem Filter 15,2 ms standardmäßig	Welligk./Störstr. (<10 MHz)	max. 1 V Spitze-Spitze
Potentialtrennung	1500 V AC, 1 min	Verfügbare Strom 24 V DC	280mA
		Kurzschlußstrombegrenzung	< 600 mA
		Elektrisch getrennt	Nein

¹ In der CPU sind 16 Eingänge im Prozeßabbild der Eingänge und 16 Ausgänge im Prozeßabbild der Ausgänge für die integrierten Ein- und Ausgänge vorgesehen.

² Lineare Leistungsminderung 40 bis 55° C, Leistungsminderung bei vertikalem Einbau 10° C.

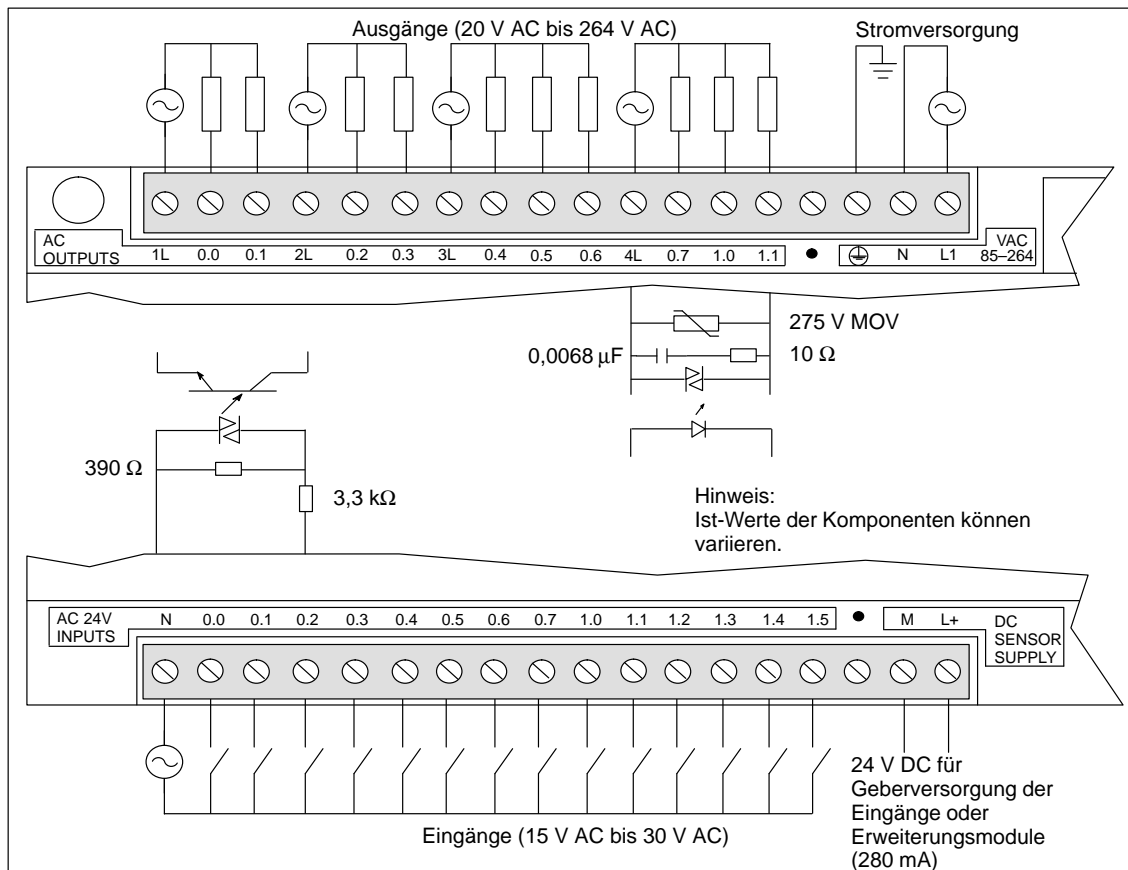


Bild A-13 Kennzeichnung der Anschlüsse bei der CPU 214 AC/AC/AC

A.14 CPU 214, AC-Versorgung, AC-Eingänge, Relaisausgänge

Bestellnummer: 6ES7 214-1GC01-0XB0

Leistungsmerkmale		Ausgänge	
Abmessungen (B x H x T)	197 x 80 x 62 mm	Ausgangstyp	Relais, Schwachstromkontakt
Gewicht	0,5 kg	Spannungsbereich	5 bis 30 V DC/250 V AC
Stromaufnahme	9 W	Max. Laststrom	2 A/Ausgang
Größe Anwenderprogramm/ Speicher	2K Wörter/EEPROM	Überspannungsstoß	7 A bei geschl. Kontakten
Größe Anw.daten/Speicher	2K Wörter/RAM	Isolationswiderstand	min. 100 MΩ (neu)
Pufferung Daten/Echtzeituhr Hochleistungskondensator	typ. 190 h (min. 120 h bei 40° C)	Verzögerung Schaltvorgänge	max. 10 ms
Batteriemodul (optional)	200 Tage bei ständig. Einsatz	Lebensdauer	10 000 000 mechanisch 100 000 mit Bemessungslast
Integrierte E/A ¹	14 Eingänge/10 Ausgänge	Kontaktwiderstand	max. 200 mΩ (neu)
Max. Anzahl Erweiterungsmodule	7	Elektrische Trennung Spule zu Kontakt	AC 1500 V, 1 Minute
Unterstützte digitale E/A	64 Eingänge/64 Ausgänge	Kontakt zu Kontakt	AC 1000 V, 1 Minute
Unterstützte analoge E/A	16 Eingänge/16 Ausgänge	Kurzschlußschutz	keine
Ausführungszeit	0,8µs/Operation	Stromversorgung	
Interne Merker	256	Spannungs-/Frequenzbereich	85 - 264 V AC bei 47 - 63 Hz
Zeiten	128 Zeiten	Eingangsstrom	typ. 4,5 VA, nur CPU 50 VA max. Laststrom
Zähler	128 Zähler	Verzögerungszeit	aus 110 V AC min. 20 ms
Schnelle Zähler	1 SW-Zähler (max. je 2 kHz) 2 HW-Zähler (max. je 7 kHz)	Einschaltstromstoß	20 A Spitze bei 264 V AC
Toleranz Echtzeituhr	6 Minuten pro Monat	Schmelzsicherung (nicht austauschbar)	2 A, 250 V, träge
Impulsausgänge	nicht empfohlen	Verfügbare Strom 5 V DC	340 mA für CPU; 660 mA für Erweiterungsmodule
Analogpotentiometer	2	Elektrisch getrennt	Ja, Transformator, AC 1500 V, 1 Minute
Normen	UL 508 CSA C22.2 142 FM Klasse I, Kategorie 2 gemäß VDE 0160 gemäß EG-Richtlinie	DC-Geberversorgung	
Eingänge		Spannungsbereich	20,4 bis 28,8 V DC
Eingangstyp (IEC 1131-2)	Typ 1, stromziehend	Welligkeit / Störströme (<10 MHz)	max. 1 V Spitze-Spitze
Bereich bei EIN	79 - 135 V AC, 47 - 63 Hz min. 4 mA	Verfügbare Strom 24 V DC	280mA
Nennspannung bei EIN	120 V AC, 60 Hz, 7 mA	Kurzschlußstrombegrenzung	< 600 mA
Maximum bei AUS	20 V AC, 1 mA	Elektrisch getrennt	Nein
Max. Ansprechzeit	0,2 ms bis 8,7 ms wählbar plus 15,0 ms bei festem Filter 15,2 ms standardmäßig		
Potentialtrennung	1500 V AC, 1 Minute		

¹ In der CPU sind 16 Eingänge im Prozeßabbild der Eingänge und 16 Ausgänge im Prozeßabbild der Ausgänge für die integrierten Ein- und Ausgänge vorgesehen.

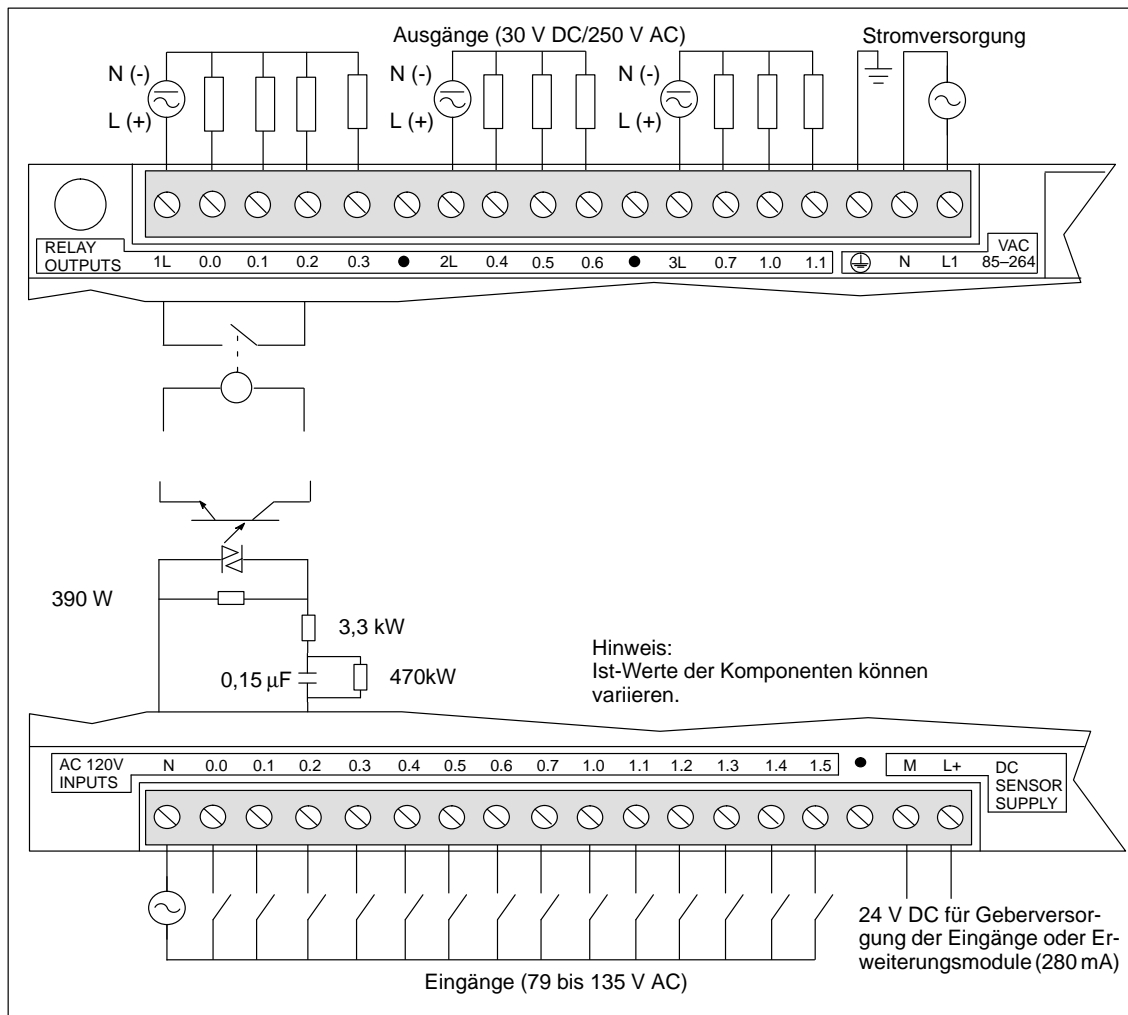


Bild A-14 Kennzeichnung der Anschlüsse bei der CPU 214 AC/AC/Relais

A.15 CPU 215, DC-Versorgung, DC-Eingänge, DC-Ausgänge

Bestellnummer: **6ES7 215-2AD00-0XB0**

Leistungsmerkmale		Ausgänge	
Abmessungen (B x H x T)	217,3 x 80 x 62 mm	Ausgangstyp	MOSFET, stromliefernd
Gewicht	0,5 kg	Spannungsbereich	20,4 bis 28,8 V DC
Stromaufnahme	8 W	Max. Laststrom	0 bis 55° C
Größe Anw.progr./Speicher	4 K Wörter/EEPROM	A0.0 bis A0.7	0,5 A/Ausgang
Größe Anw.daten/Speicher	2,5 K Wörter/RAM	A1.0, A1.1	1,0 A/Ausgang
Pufferung Daten/Echtzeituhr		Für höheren Strom:	
Hochleistungskondensator	typ. 190 h (min. 120 h bei 40° C)	Ausgänge parallel schalten.	
Batteriemodul (optional)	200 Tage bei ständig. Einsatz	Ableitstrom: A0.0 - A0.7	200 µA
Integrierte E/A ¹	14 Eingänge/10 Ausgänge	A1.0, A1.1	400 µA
Max. Anz. Erweiterungsmod.	7	Verzögerung Schaltvorgänge	
Unterstützte digitale E/A	64 Eingänge/64 Ausgänge	A0.0, A0.1	100 µs, EIN/AUS
Unterstützte analoge E/A	16 Eingänge/16 Ausgänge	Alle anderen	150 µs EIN, 400 µs AUS
Boolesche Ausführungszeiten	0,8µs/Operation	Widerstand	max. 400 mΩ
Interne Merker	256	Kurzschlußschutz	
Zeiten	256 Zeiten	A0.0 bis A0.7	0,7 bis 1,5 A/Kanal
Zähler	256 Zähler	A1.0, A1.1	1,5 bis 3 A/Kanal
Schnelle Zähler	1 SW-Zähler (max. 2 kHz) 2 HW-Zähler (max. je 20 kHz)	Potentialtrennung	500 V AC, 1 min
Toleranz Echtzeituhr	6 Minuten pro Monat	Stromversorgung	
Impulsausgänge	2 (max. jeweils 4 kHz)	Spannungsbereich	20,4 bis 28,8 V DC
Analogpotentiometer	2	Eingangsstrom	typ. 120 mA, nur CPU 1,4 A max. Laststrom
Normen	UL 508 CSA C22.2 142 FM Klasse I, Kategorie 2 gemäß VDE 0160 gemäß EG-Richtlinie	UL/CSA-Bemessung	50 VA
Eingänge		Verzögerungszeit	aus 24 V DC min. 10 ms
Eingangstyp	stromziehend/stromliefernd IEC Typ 1 wenn stromzieh.	Einschaltstromstoß	10 A Spitze bei 28,8 V DC
Bereich bei EIN	15 bis 30 V DC, min. 4 mA 35 V DC, 500 ms Spann.stoß	Schmelzsicherung (nicht austauschbar)	2 A, träge
Nennspannung bei EIN	24 V DC, 7 mA	Strom 5 V DC	1000 mA für Erweiter.module
Maximum bei AUS	5 V DC, 1 mA	Elektrisch getrennt	Nein
Maximale Ansprechzeit		DC-Geberversorgung	
E0.0 bis E1.5	0,2 ms bis 8.7 ms wählbar	Spannungsbereich	16,4 V DC bis 28,8 V DC
E 1.5 wie bei	0,2 ms standardmäßig	Welligk./Störstr. (<10 MHz)	wie zugeführte Spannung
HSC1 und HSC2	6 ms EIN, 30 ms AUS	Verfügbare Strom 24 V DC	400 mA
Potentialtrennung	500 V AC, 1 min	Kurzschlußstrombegrenzung	< 600 mA
		Elektrisch getrennt	Nein
		5-V-DP-Kommunikationsversorgung	
		Strom 5 V DC	90 mA, verfügbar an DP-Schnittstelle, Pole 6-5, DP-Busverstärker
		Elektrische Trennung	Transform., 500 V AC, 1 min

¹ In der CPU sind 16 Eingänge im Prozeßabbild der Eingänge und 16 Ausgänge im Prozeßabbild der Ausgänge für die integrierten Ein- und Ausgänge vorgesehen.

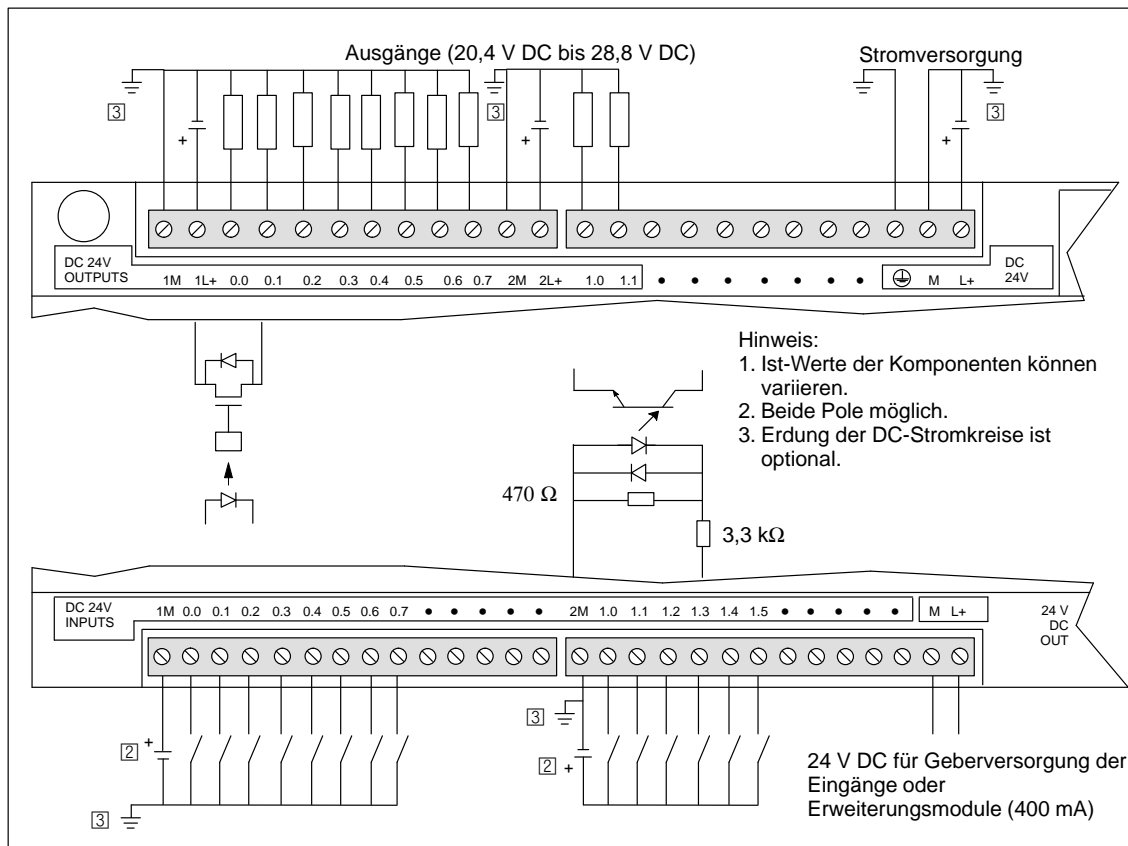


Bild A-15 Kennzeichnung der Anschlüsse bei der CPU 215 DC/DC/DC

A.16 CPU 215, AC-Versorgung, DC-Eingänge, Relaisausgänge

Bestellnummer: **6ES7 215-2BD00-0XB0**

Leistungsmerkmale		Ausgänge	
Abmessungen (B x H x T)	217,3 x 80 x 62 mm	Ausgangstyp	Relais, Schwachstromkontakt
Gewicht	0,6 kg	Spannungsbereich	5 V DC - 30 V DC/250 V AC
Stromaufnahme	9 W	Max. Laststrom	2 A/Ausgang, 6 A/Leiter
Größe Anw.progr./Speicher	4 K Wörter/EEPROM	Überspannungsstoß	7 A bei geschl. Kontakten
Größe Anw.daten/Speicher	2,5 K Wörter/RAM	Isolationswiderstand	min. 100 MΩ (neu)
Pufferung Daten/Echtzeituhr Hochleistungskondensator	typ. 190 h (min. 120 h bei 40°C)	Verzögerung Schaltvorgänge	max. 10 ms
Batteriemodul (optional)	200 Tage bei ständig. Einsatz	Lebensdauer	10 000 000 mechanisch 100 000 mit Bemessungslast
Integrierte E/A ¹	14 Eingänge/10 Ausgänge	Kontaktwiderstand	max. 200 mΩ (neu)
Max. Anzahl Erweiterungsmodule	7	Elektrische Trennung Spule zu Kontakt	1500 V AC, 1 min
Unterstützte digitale E/A	64 Eingänge/64 Ausgänge	Kontakt zu Kontakt (zw. geöffn. Kontakten)	750 V AC, 1 min
Unterstützte analoge E/A	16 Eingänge/16 Ausgänge	Kurzschlußschutz	keine
Boolesche Ausführungszeiten	0,8µs/Operation	Stromversorgung	
Interne Merker	256	Spannungs-/Frequenzbereich	85 - 264 V AC bei 47 - 63 Hz
Zeiten	256 Zeiten	Eingangsstrom	typ. 6 VA, nur CPU 50 VA max. Laststrom
Zähler	256 Zähler	Verzögerungszeit	aus 110 V AC min. 20 ms
Schnelle Zähler	1 SW-Zähler (max. 2 kHz) 2 HW-Zähler (max. je 20 kHz)	Einschaltstromstoß	20 A Spitze bei 264 V AC
Toleranz Echtzeituhr	6 Minuten pro Monat	Schmelzsich. (n. austauschb.)	2 A, 250 V, träge
Impulsausgänge	nicht empfohlen	Strom 5 V DC	1000 mA f. Erweiter.module
Analogpotentiometer	2	Elektrisch getrennt	Ja, Transformator, 1500 V AC, 1 min
Normen	UL 508 CSA C22.2 142 FM Klasse I, Kategorie 2 gemäß VDE 0160 gemäß EG-Richtlinie	DC-Geberversorgung	
Eingänge		Spannungsbereich	19,2 bis 28,8 V DC
Eingangstyp	stromziehend/stromliefernd IEC 1131 Typ 1 wenn stromz.	Welligk./Störstr. (<10 MHz)	max. 1 V Spitze-Spitze
Bereich bei EIN	15 bis 30 V DC, min. 4 mA 35 V DC, 500 ms Spannungsstoß	Verfügbarer Strom 24 V DC	400 mA
Nennspannung bei EIN	24 V DC, 7 mA	Kurzschlußstrombegrenzung	< 600 mA
Maximum bei AUS	5 V DC, 1 mA	Elektrisch getrennt	Nein
Max. Ansprechz.: E0.0 - E1.5	0,2 ms bis 8.7 ms wählbar 0,2 ms standardmäßig	5-V-DP-Kommunikationsversorgung	
E0.6 - E1.5 wie bei HSC1 und HSC2	6 ms EIN, 30 ms AUS mm	Strom 5 V DC	90 mA, verfügbar an DP-Schnittstelle, Pole 6-5, DP-Busverstärker
		Elektrische Trennung	Transformator, 500 V AC, 1 min

¹ In der CPU sind 16 Eingänge im Prozeßabbild der Eingänge und 16 Ausgänge im Prozeßabbild der Ausgänge für die integrierten Ein- und Ausgänge vorgesehen.

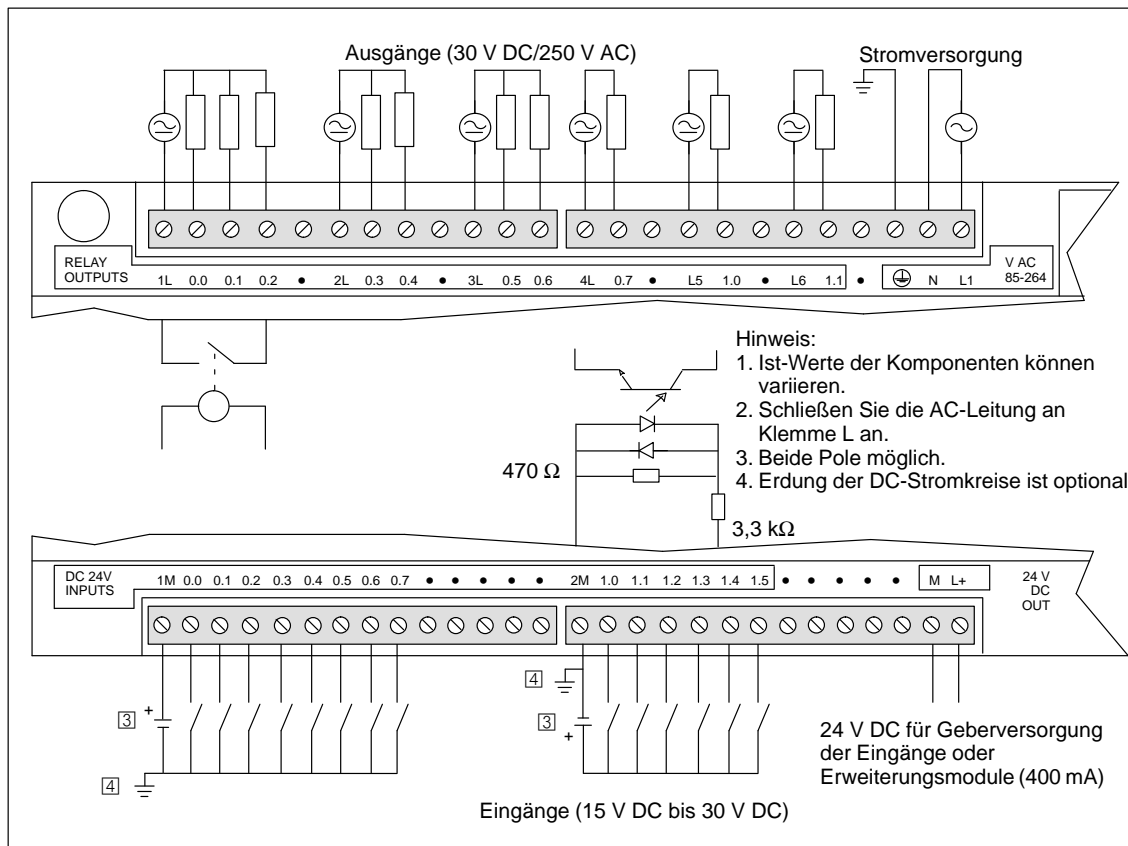


Bild A-16 Kennzeichnung der Anschlüsse bei der CPU 215 AC/DC/Relais

A.17 CPU 216, DC-Versorgung, DC-Eingänge, DC-Ausgänge

Bestellnummer: **6ES7 216-2AD00-0XB0**

Leistungsmerkmale		Ausgänge	
Abmessungen (B x H x T)	217,3 x 80 x 62 mm	Ausgangstyp	MOSFET, stromliefernd
Gewicht	0,5 kg	Spannungsbereich	20,4 V DC bis 28,8 V DC
Stromaufnahme	8 W	Max. Laststrom (Ausgänge können für höheren Strom parallel geschaltet werden)	0 bis 55 C 0,5 A/Ausgang
Größe Anw.progr./Speicher	4 K Wörter/EEPROM	Ableitstrom	200 µA
Größe Anw.erdaten/Speicher	2,5 K Wörter/RAM	Verzögerung Schaltvorgänge	
Pufferung Daten/Echtzeituhr		A0.0, A0.1	100 µs, EIN/AUS
Hochleistungskondensator	typ. 190 h (min. 120 h bei 40° C)	Alle anderen	150 µs EIN, 400 µs AUS
Batteriemodul (optional)	200 Tage bei ständig. Einsatz	Widerstand	max. 400 mΩ
Integrierte E/A ¹	24 Eingänge/16 Ausgänge	Kurzschlußschutz	0,7 bis 1,5 A/Kanal
Max. Anzahl Erweiterungsmodule	7	Potentialtrennung	500 V AC, 1 min
Unterstützte digitale E/A	64 Eingänge/64 Ausgänge	Stromversorgung	
Unterstützte analoge E/A	16 Eingänge/16 Ausgänge	Spannungsbereich	20,4 V DC bis 28,8 V DC
Boolesche Ausführungszeiten	0,8µs/Operation	Eingangsstrom	typ. 100 mA, CPU nur 1,2 A max. Last
Interne Merker	256	UL/CSA-Bemessung	50 VA
Zeiten	256 Zeiten	Verzögerungszeit	aus 24 V DC min. 10 ms
Zähler	256 Zähler	Einschaltstromstoß	10 A Spitze bei 28,8 V DC
Schnelle Zähler	1 SW-Zähler (max. 2 kHz) 2 HW-Zähler (max. je 20 kHz)	Schmelzsich. (n. austauschb.)	2 A, träge
Toleranz Echtzeituhr	6 Minuten pro Monat	Strom 5 V DC	1000 mA für Erweiter.module
Impulsausgänge	2 (max. jeweils 4 kHz)	Elektrisch getrennt	Nein
Analogpotentiometer	2	DC-Geberversorgung	
Normen	UL 508 CSA C22.2 142 FM Klasse I, Kategorie 2 gemäß VDE 0160 gemäß EG-Richtlinie	Spannungsbereich	16,4 V DC bis 28,8 V DC
Eingänge		Welligk./Störstr. (<10 MHz)	wie zugeführte Spannung
Eingangstyp	stromziehend/stromliefernd IEC 1131 Typ 1 wenn stromziehend	Verfügbarer Strom 24 V DC	400 mA
Bereich bei EIN	15 bis 30 V DC, min. 4 mA 35 V DC, 500 ms Spann.st.	Kurzschlußstrombegrenzung	< 600 mA
Nennspannung bei EIN	24 V DC, 7 mA	Elektrisch getrennt	Nein
Maximum bei AUS	5 V DC, 1 mA		
Maximale Ansprechzeit			
E0.0 bis E1.5	0,2 ms bis 8,7 ms wählbar 0,2 ms standardmäßig		
E 1.5 wie bei HSC1 u.HSC2	6 ms EIN, 30 ms AUS		
E 1.6 bis E 2.7	max. 4 ms		
Potentialtrennung	500 V AC, 1 min		

¹ In der CPU sind 24 Eingänge im Prozeßabbild der Eingänge und 16 Ausgänge im Prozeßabbild der Ausgänge für die integrierten Ein- und Ausgänge vorgesehen.

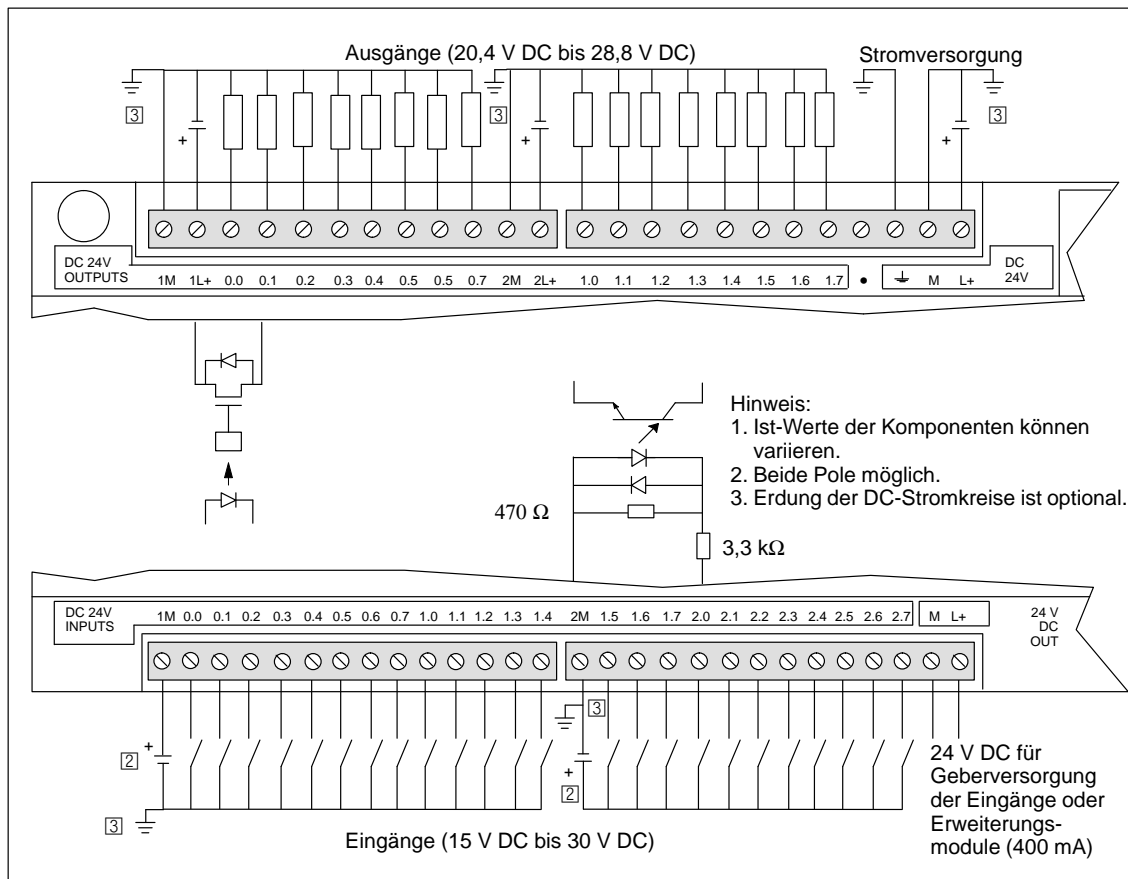


Bild A-17 Kennzeichnung der Anschlüsse bei der CPU 216 DC/DC/DC

A.18 CPU 216, AC-Versorgung, DC-Eingänge, Relaisausgänge

Bestellnummer: **6ES7 216-2BD00-0XB0**

Leistungsmerkmale		Ausgänge	
Abmessungen (B x H x T)	217,3 x 80 x 62 mm	Ausgangstyp	Relais, Schwachstromkontakt
Gewicht	0,6 kg	Spannungsbereich	5 bis 30 V DC/250 V AC
Stromaufnahme	9 W	Max. Laststrom	2 A/Ausgang, 10 A/Leiter
Größe Anw.progr./Speicher	4 K Wörter/EEPROM	Überspannungsstoß	7 A bei geschlossenen Kontakten
Größe Anw.daten/Speicher	2,5 K Wörter/RAM	Isolationswiderstand	min. 100 MΩ (neu)
Puff. Daten/Echtzeituhr		Verzögerung	max. 10 ms
Hochleistungskondensator	typ. 190 h (min. 120h bei 40°C)	Schaltvorgänge	
Batteriemodul (optional)	200 Tage bei ständig. Einsatz	Lebensdauer	10 000 000 mechanisch 100 000 mit Bemessungslast
Integrierte E/A ¹	24 Eingänge/16 Ausgänge	Kontaktwiderstand	max. 200 mΩ (neu)
Max. Anz. Erweiter.module	7	Elektrische Trennung	
Unterstützte digitale E/A	64 Eingänge/64 Ausgänge	Spule zu Kontakt	1500 V AC, 1 min
Unterstützte analoge E/A	16 Eingänge/16 Ausgänge	Kontakt zu Kontakt	750 V AC, 1 min
Boolesche Ausführungszeit	0,8µs/Operation	(zw. geöffn. Kontakten)	
Interne Merker	256	Kurzschlußschutz	keine
Zeiten	256 Zeiten	Stromversorgung	
Zähler	256 Zähler	Spannungs-/Frequenzber.	85 - 264 V AC bei 47 - 63 Hz
Schnelle Zähler	1 SW-Zähler (max. 2 kHz) 2 HW-Zähler (max. je 20 kHz)	Eingangsstrom	typ. 6 VA, nur CPU 50 VA max. Laststrom
Toleranz Echtzeituhr	6 Minuten pro Monat	Verzögerungszeit	aus 110 V AC min. 20 ms
Impulsausgänge	nicht empfohlen	Einschaltstromstoß	20 A Spitze bei 264 V AC
Analogpotentiometer	2	Schmelzsicherung	2 A, 250 V, träge
Normen	UL 508 CSA C22.2 142 FM Klasse I, Kategorie 2 gemäß VDE 0160 gemäß EG-Richtlinie	(nicht austauschbar)	
Eingänge		Strom 5 V DC	1000 mA für Erweiterungsmodule
Eingangstyp	stromziehend/stromliefernd IEC Typ 1131, wenn stromziehend	Elektrisch getrennt	Ja, Transformator, 1500 V AC, 1 min
Bereich bei EIN	15 bis 30 V DC, min. 4 mA 35 V DC, 500 ms Spann.stoß	DC-Geberversorgung	
Nennspannung bei EIN	24 V DC, 7 mA	Spannungsbereich	19,2 V DC bis 28,8 V DC
Maximum bei AUS	5 V DC, 1 mA	Welligkeit / Störströme	max. 1 V Spitze-Spitze (<10 MHz)
Max. Ansprechz.: E0.0 - E1.5	0,2 ms bis 8,7 ms wählbar	Verfügbarer Strom 24 V DC	400 mA
E0.6 - E1.5	0,2 ms standardmäßig	Kurzschlußstrombegrenzung	< 600 mA
wie bei HSC1 und HSC2	6 ms EIN, 30 ms AUS	Elektrisch getrennt	Nein
E1.6 - E2.7	max. 4 ms		
Potentialtrennung	500 V AC, 1 min		

1 In der CPU sind 24 Eingänge im Prozeßabbild der Eingänge und 16 Ausgänge im Prozeßabbild der Ausgänge für die integrierten Ein- und Ausgänge vorgesehen.

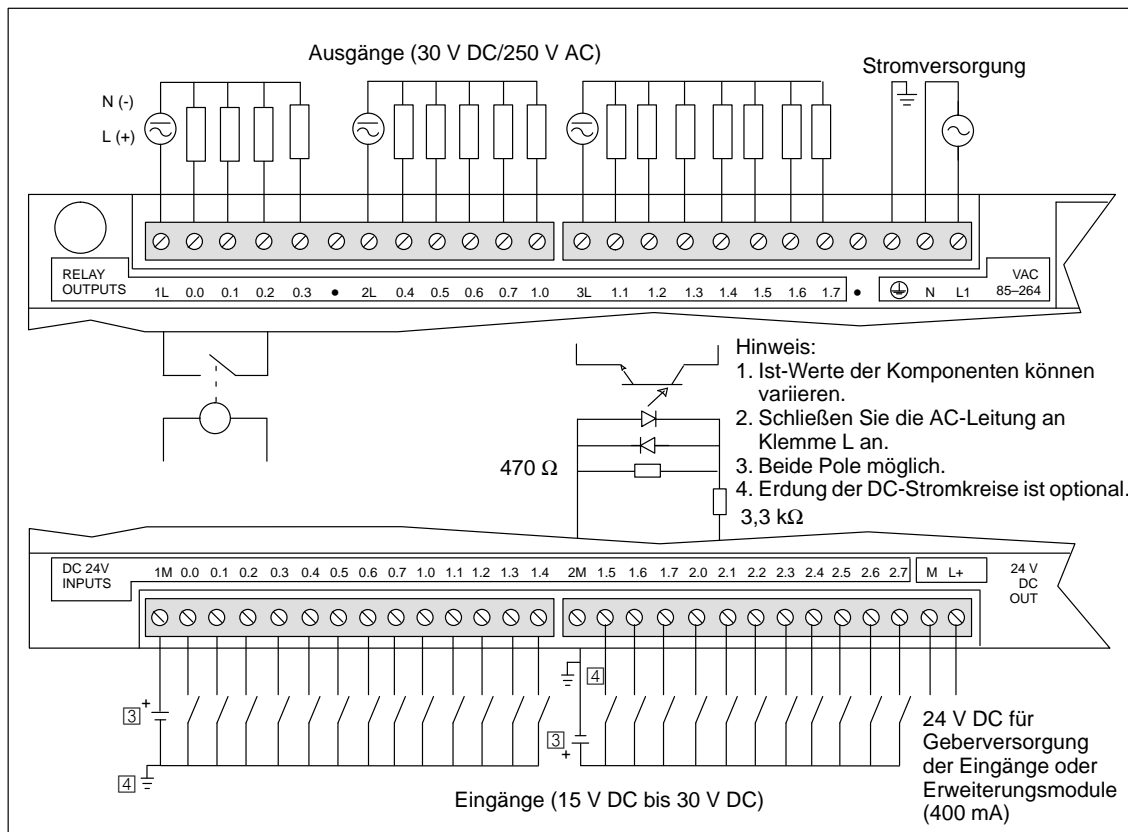


Bild A-18 Kennzeichnung der Anschlüsse bei der CPU 216 AC/DC/Relais

A.19 Erweiterungsmodul EM 221 Digitaleingabe 8 x 24 V DC

Bestellnummer: 6ES7 221-1BF00-0XA0

Leistungsmerkmale		Eingänge	
Abmessungen (B x H x T)	90 x 80 x 62 mm	Eingangstyp	Typ 1, stromziehend gemäß IEC 1131-2
Gewicht	0,2 kg	Bereich bei EIN	15 bis 30 V DC, min. 4 mA 35 V DC, 500 ms Spannungsstoß
Stromaufnahme	2 W	Nennspannung bei EIN	24 V DC, 7 mA
Ein- und Ausgänge ¹	8 digitale Eingänge	Maximum bei AUS	5 V DC, 1 mA
Normen	UL 508 CSA C22.2 142 FM Klasse I, Kategorie 2 gemäß VDE 0160 gemäß EG-Richtlinie	Ansprechzeit	typ. 3,5 ms/max. 4,5 ms
		Potentialtrennung	500 V AC, 1 min
		Strombedarf	
		5-V-DC-logische-Spannung	60 mA von Zentralgerät
		24-V-DC-Geberspannung	60 mA von Zentralgerät oder externer Stromversorgung

¹ In der CPU sind 8 Eingänge im Prozeßabbild der Eingänge für dieses Modul vorgesehen.

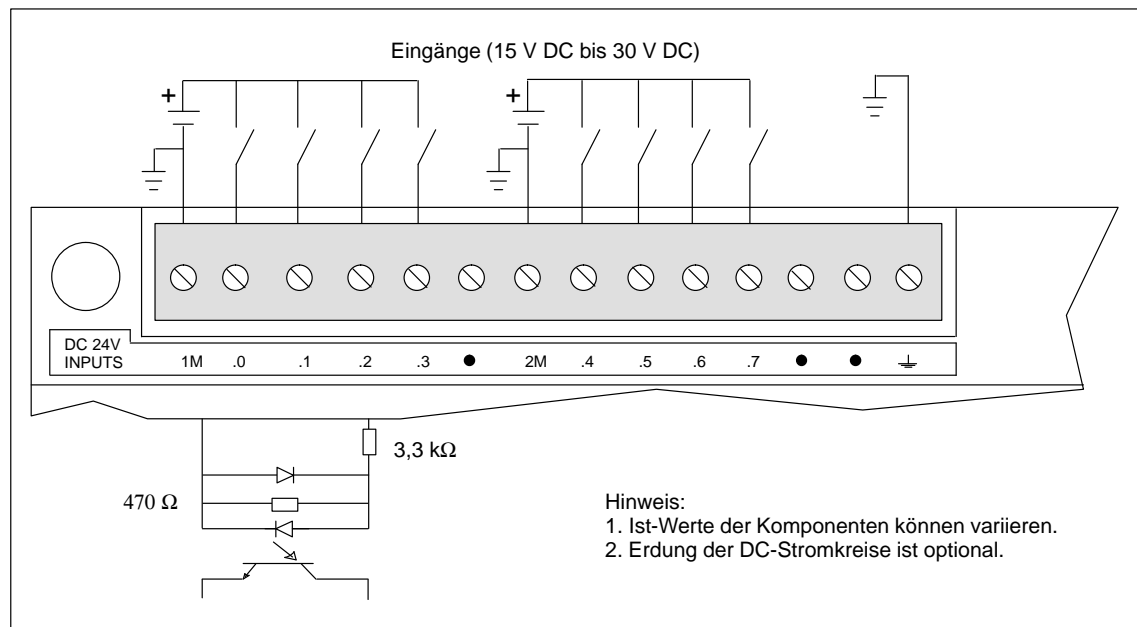


Bild A-19 Kennzeichnung der Anschlüsse beim EM 221 Digitaleingabe 8 x 24 V DC

A.20 Erweiterungsmodul EM 221, Digitaleingabe 8 x 120 V AC

Bestellnummer: **6ES7 221-1EF00-0XA0**

Leistungsmerkmale		Eingänge	
Abmessungen (B x H x T)	90 x 80 x 62 mm	Eingangstyp	Typ 1, stromziehend gemäß IEC 1131-2
Gewicht	0,2 kg	Bereich bei EIN	79 - 135 V AC, 47 - 63 Hz, min. 4 mA
Stromaufnahme	2 W	Nennspannung bei EIN	120 V AC, 60 Hz, 7 mA
Ein- und Ausgänge ¹	8 digitale Eingänge	Maximum bei AUS	20 V AC, 1 mA
Normen	UL 508 CSA C22.2 142 FM Klasse I, Kategorie 2 gemäß EG-Richtlinie	Ansprechzeit	max. 15 ms
		Potentialtrennung	1500 V AC, 1 min
		Strombedarf	
		5-V-DC-logische-Spannung	70 mA aus Zentralgerät

¹ In der CPU sind 8 Eingänge im Prozeßabbild der Eingänge für dieses Modul vorgesehen.

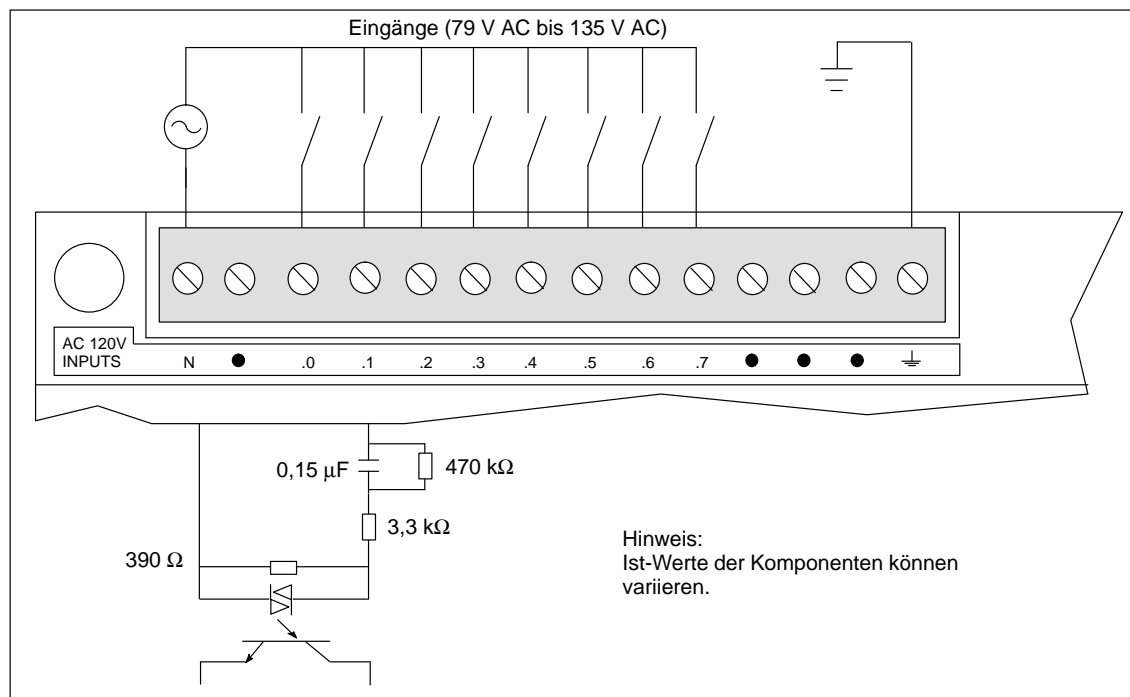


Bild A-20 Kennzeichnung der Anschlüsse beim EM 221 Digitaleingabe 8 x 120 V AC

A.21 Erweiterungsmodul EM 221 Digitaleingabe (stromliefernd) 8 x 24 V DC

Bestellnummer: 6ES7 221-1BF10-0XA0

Leistungsmerkmale		Eingänge	
Abmessungen (B x H x T)	90 x 80 x 62 mm	Datentyp	stromliefernd
Gewicht	0,2 kg	Bereich Eingangsspannung	15 V DC bis 30 V DC, 35 V DC für 500 ms
Stromaufnahme	2 W	Nennspannung bei EIN	min. 4 mA
Ein- und Ausgänge ¹	8 digitale Eingänge	Maximum bei AUS	1 mA
Normen	UL 508 CSA C22.2 142 FM Klasse I, Kategorie 2 gemäß VDE 0160 gemäß EG-Richtlinie	Ansprechzeit	typ. 3,5 ms / max. 4,5 ms
		Potentialtrennung	500 V AC, 1 min
		Strombedarf	
		5-V-DC-logische-Spannung	60 mA von Zentralgerät
		24-V-DC-Geberspannung	60 mA von Zentralgerät oder externer Stromversorgung

¹ In der CPU sind 8 Eingänge im Prozeßabbild der Eingänge für dieses Modul vorgesehen.

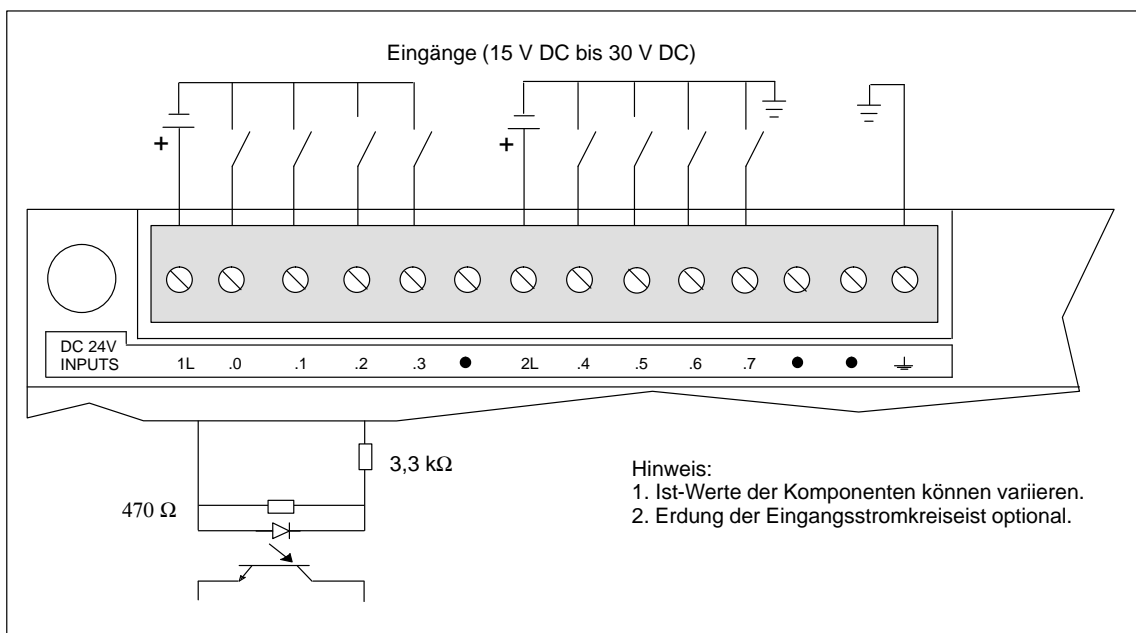


Bild A-21 Kennzeichnung der Anschlüsse beim EM 221 Digitaleingabe (stromliefernd) 8 x 24 V DC

A.22 Erweiterungsmodul EM 221, Digitaleingabe 8 x 24 V AC

Bestellnummer: **6ES7 221-1JF00-0XA0**

Leistungsmerkmale		Eingänge	
Abmessungen (B x H x T)	90 x 80 x 62 mm	Eingangstyp	Typ 1, stromziehend gemäß IEC 1131-2
Gewicht	0,2 kg	Bereich bei EIN	15 - 30 V AC, 47 - 63 Hz, min. 4 mA
Stromaufnahme	2 W	Nennspannung bei EIN	24 V AC, 60 Hz, 7 mA
Ein- und Ausgänge ¹	8 digitale Eingänge	Maximum bei AUS	5 V AC, 1 mA
Normen (beantragt)	UL 508 CSA C22.2 142 FM Klasse I, Kategorie 2 gemäß EG-Richtlinie	Ansprechzeit	max. 15 ms
		Potentialtrennung	1500 V AC, 1 min
		Strombedarf	
		5-V-DC-logische-Spannung	70 mA aus Zentralgerät

¹ In der CPU sind 8 Eingänge im Prozeßabbild der Eingänge für dieses Modul vorgesehen.

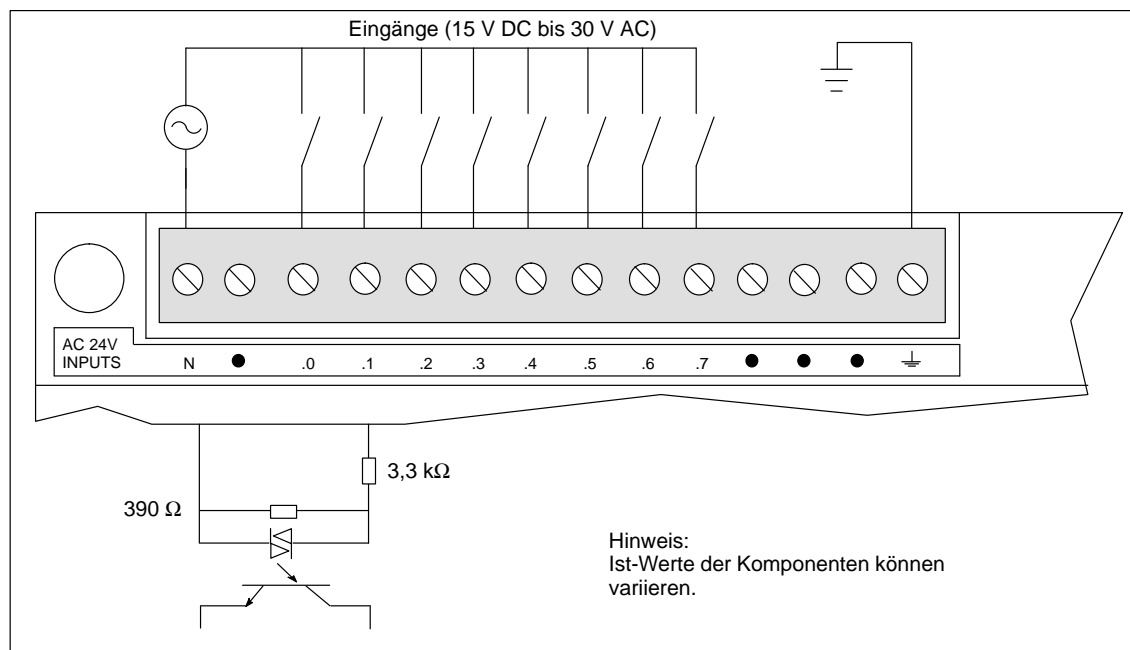


Bild A-22 Kennzeichnung der Anschlüsse beim EM 221 Digitaleingabe 8 x 24 V AC

A.23 Erweiterungsmodul EM 222, Digitalausgabe 8 x 24 V DC

Bestellnummer: **6ES7 222-1BF00-0XA0**

Leistungsmerkmale		Ausgänge, Fortsetzung	
Abmessungen (B x H x T)	90 x 80 x 62 mm	Klemmung induktive Last einzelner Impuls	(pro Leiter) 2A L/R = 10 ms 1A L/R = 100 ms
Gewicht	0,2 kg	Wiederholung	1 W Energieaufnahme (1/2 Li ² x Schaltfreq. < 1 W)
Stromaufnahme	4 W bei 3 A Last	Ableitstrom	100 µA
Ein- und Ausgänge ¹	8 digitale Ausgänge	Verzögerung Schaltvorgänge	50 ms EIN, 200 ms AUS
Normen	UL 508 CSA C22.2 142 FM Klasse I, Kategorie 2 gemäß VDE 0160 gemäß EG-Richtlinie	Stoßstrom	4 A, 100 ms
Ausgänge		Spannungsabfall	max. 1,8 V bei max. Strom
Ausgangstyp	Transistor, stromliefernd	Potentialtrennung	500 V AC, 1 min
Spannungsbereich	20,4 V DC bis 28,8 V DC	Kurzschlußschutz	keine
Max. Laststrom	<u>0 bis 40° C</u> <u>55° C</u> ²	Strombedarf	
pro einzelнем Ausgang	0,75 A 0,50 A	5-V-DC-logische-Spannung	80 mA aus Zentralgerät
pro 2 benachb. Ausgänge	1,00 A 0,75 A	Strom an Ausgängen	Wird vom Anwender am Modulleiter geliefert.
Ausgänge gesamt	4,00 A 3,00 A		

¹ In der CPU sind 8 Eingänge im Prozeßabbild der Eingänge für dieses Modul vorgesehen.

² Lineare Leistungsminderung 40 bis 55° C, Leistungsminderung bei vertikalem Einbau 10° C.

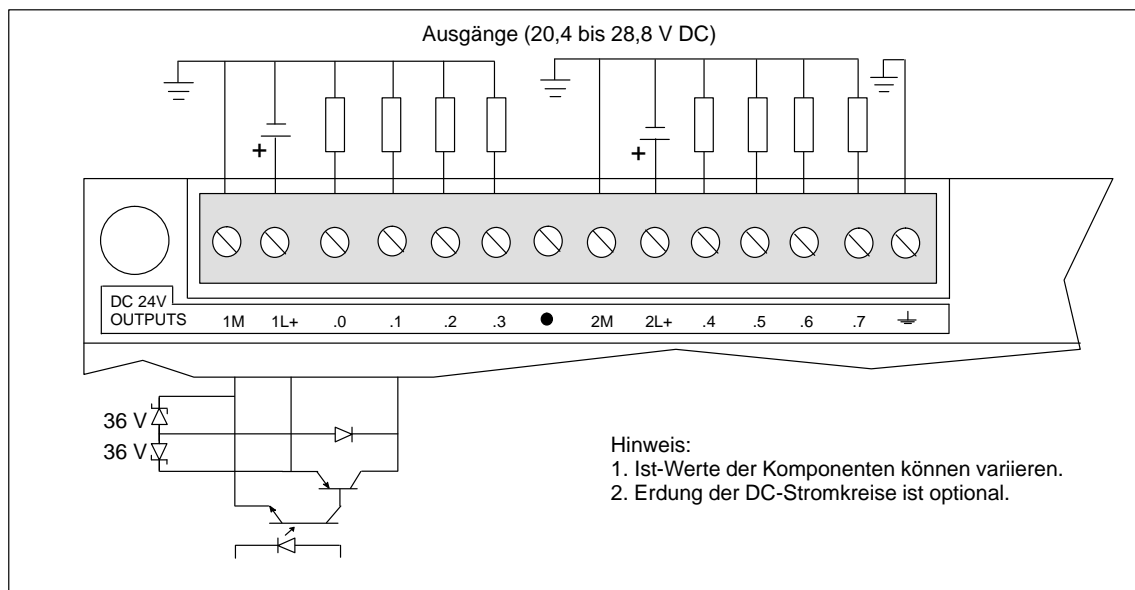


Bild A-23 Kennzeichnung der Anschlüsse beim EM 222 Digitalausgabe 8 x 24 V DC

A.24 Erweiterungsmodul EM 222, Digitalausgabe 8 x Relais

Bestellnummer: 6ES7 222-1HF00-0XA0

Leistungsmerkmale		Ausgänge, Fortsetzung	
Abmessungen (B x H x T)	90 x 80 x 62 mm	Verzögerung Schaltvorgänge	max. 10 ms
Gewicht	0,2 kg	Lebensdauer	10 000 000 mechanisch 100 000 mit Bemessungslast
Stromaufnahme	3 W	Kontaktwiderstand	max. 200 mΩ (neu)
Ein- und Ausgänge ¹	8 digitale Relaisausgänge	Elektrische Trennung	
Normen	UL 508 CSA C22.2 142 FM Klasse I, Kategorie 2 gemäß VDE 0160 gemäß EG-Richtlinie	Spule zu Kontakt	1500 V AC, 1 min
		Kontakt zu Kontakt (zwischen geöffneten Kontakten)	750 V AC, 1 min
		Kurzschlußschutz	keine
Ausgänge		Strombedarf	
Ausgangstyp	Relais, Schwachstromkontakt	5-V-DC-logische-Spannung	80 mA aus Zentralgerät
Spannungsbereich	5 bis 30 V DC/250 V AC	Ausgangsspannung 24 V DC	85 mA von Zentralgerät oder externer Stromversorgung
Max. Laststrom	2 A/Ausgang, 8 A/Leiter	Strom an Ausgängen	Wird vom Anwender am Modulleiter geliefert.
Überspannungsstoß	7 A bei geschl. Kontakten		
Isolationswiderstand	min. 100 MΩ (neu)		

¹ In der CPU sind 8 Eingänge im Prozeßabbild der Eingänge für dieses Modul vorgesehen.

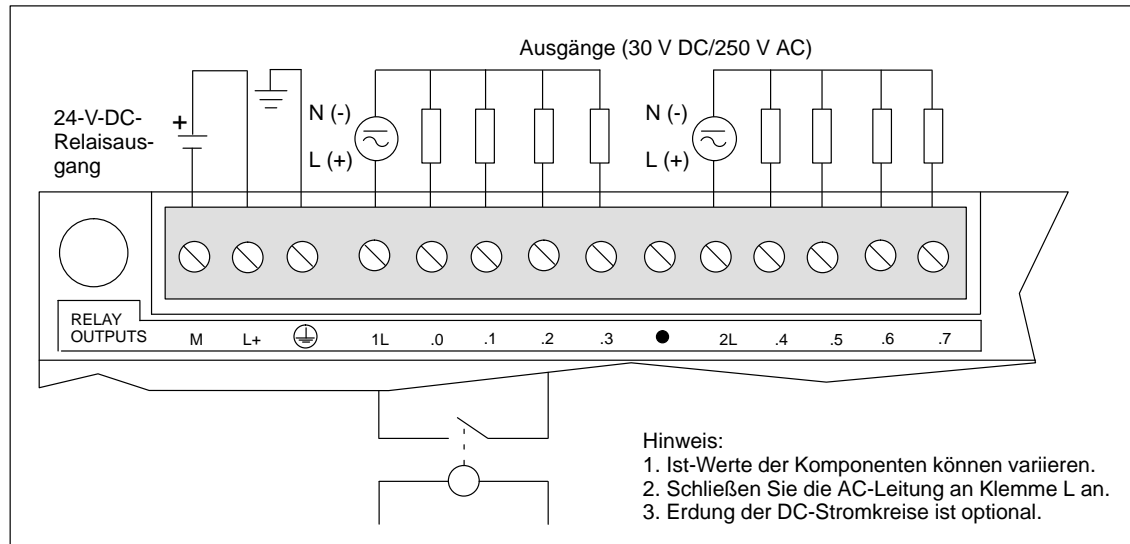


Bild A-24 Kennzeichnung der Anschlüsse beim EM 222 Digitalausgabe 8 x Relais

A.25 Erweiterungsmodul EM 222, Digitalausgabe 8 x 120/230 V AC

Bestellnummer: 6ES7 222-1EF00-0XA0

Leistungsmerkmale		Ausgänge, Fortsetzung	
Abmessungen (B x H x T)	90 x 80 x 62 mm	Min. Laststrom	30 mA
Gewicht	0,2 kg	Ableitstrom	1,5 mA, 120 V AC / 2,0 mA, 240 V AC
Stromaufnahme	5 W bei 3,5 A Last	Verzögerung Schaltvorgänge	1/2 Zyklus
Ein- und Ausgänge ¹	8 digitale Ausgänge	Stoßstrom	30 A Spitze, 1 Zyklus 10 A Spitze, 5 Zyklen
Normen	UL 508 CSA C22.2 142 FM Klasse I, Kategorie 2 gemäß EG-Richtlinie	Spannungsabfall	max. 1,5 V bei maximalem Strom
Ausgänge		Potentialtrennung	1500 V AC, 1 min
Ausgangstyp	Triac, Nulldurchgang einschalten	Kurzschlußschutz	keine
Spannungs-/Frequenzbereich	20 - 264 V AC, 47 - 63 Hz	Strombedarf	
Leistungsfaktor Laststromkreis	0,3 bis 1,0	5-V-DC-logische-Spannung	120 mA von Zentralgerät
Max. Laststrom	<u>0 bis 40° C</u> <u>55° C²</u>	Strom an Ausgängen	Wird vom Anwender am Modulleiter geliefert.
pro einzeltem Ausgang	1,20 A 1,00 A		
pro 2 benachb. Ausgänge	1,50 A 1,25 A		
Ausgänge gesamt	4,75 A 3,50 A		

¹ In der CPU sind 8 Eingänge im Prozeßabbild der Eingänge für dieses Modul vorgesehen.

² Lineare Leistungsminderung 40 bis 55° C, Leistungsminderung bei vertikalem Einbau 10° C.

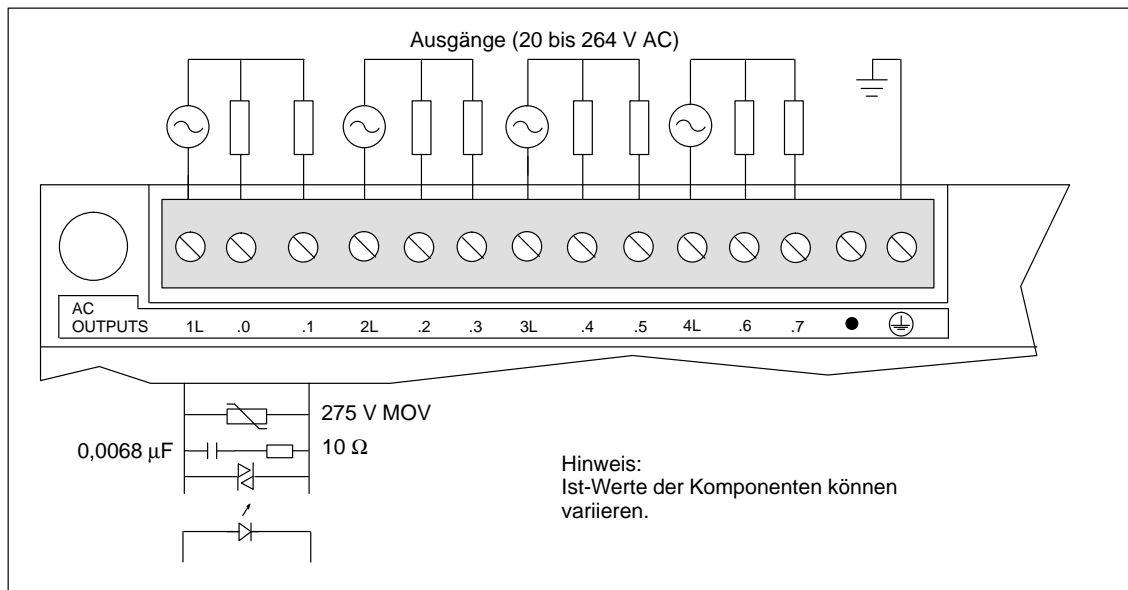


Bild A-25 Kennzeichnung der Anschlüsse beim EM 222 Digitalausgabe 8 x 120/230 V AC

A.26 Erweiterungsmodul EM 223 Digitalein-/Digitalausgabe 4 x 24-V-DC-Eingang/4 x 24-V-DC-Ausgang

Bestellnummer: 6ES7 223-1BF00-0XA0

Leistungsmerkmale		Ausgänge, Fortsetzung	
Abmessungen (B x H x T)	90 x 80 x 62 mm	Ableitstrom	max. 1 µA
Gewicht	0,2 kg	Verzögerung Schaltvorgänge	25 µs EIN, max. 120 µs AUS
Stromaufnahme	3,5 W bei 3 A Last	Stoßstrom	7 A, 100 ms
Ein- und Ausgänge ¹	4 digitale Eingänge 4 digitale Ausgänge	Potentialtrennung	500 V AC, 1 min
Normen	UL 508 CSA C22.2 142 FM Klasse I, Kategorie 2 gemäß VDE 0160 gemäß EG-Richtlinie	Kurzschlußschutz	keine
Ausgänge		Eingänge	
Ausgangstyp	Transistor, stromliefernd (MOSFET, P-Kanal)	Eingangstyp	Typ 1, stromziehend gemäß IEC 1131-2
Spannungsbereich	20,4 bis 28,8 V DC	Bereich bei EIN	15 bis 30 V DC, min. 4 mA 35 V DC, 500 ms Spannungsstoß
Widerstand bei EIN	max. 400 mΩ	Nennspannung bei EIN	24 V DC, 7 mA
Max. Laststrom	<u>0 bis 40° C</u> <u>55° C*</u>	Maximum bei AUS	5 V DC, 1 mA
pro einzeltem Ausgang	2,50 A 2,00 A	Ansprechzeit	typ. 3,5 ms/max. 4,5 ms
Ausgänge gesamt	4,00 A 3,00 A	Potentialtrennung	500 V AC, 1 min
*Lin. Leistungsminderung 40 bis 55° C Leistungsminderung bei vertikalem Einbau 10° C (Zwei Ausgänge können für starke Stromlasten parallel geschaltet werden.)		Strombedarf	
Klemmung induktive Last einzelner Impuls	(pro Leiter) 2A L/R = 10 ms 1A L/R = 100 ms	5-V-DC-logische-Spannung	80 mA aus Zentralgerät
Wiederholung	1 W Energieaufnahme (1/2 Li ² x Schaltfreq. < 1 W)	24-V-DC-Geberspannung	30 mA von Zentralgerät oder externer Stromversorgung
		Strom an Ausgängen	Wird vom Anwender am Modulleiter geliefert.

¹ In der CPU sind 8 Eingänge im Prozeßabbild der Eingänge und 8 Ausgänge im Prozeßabbild der Ausgänge für dieses Modul vorgesehen.

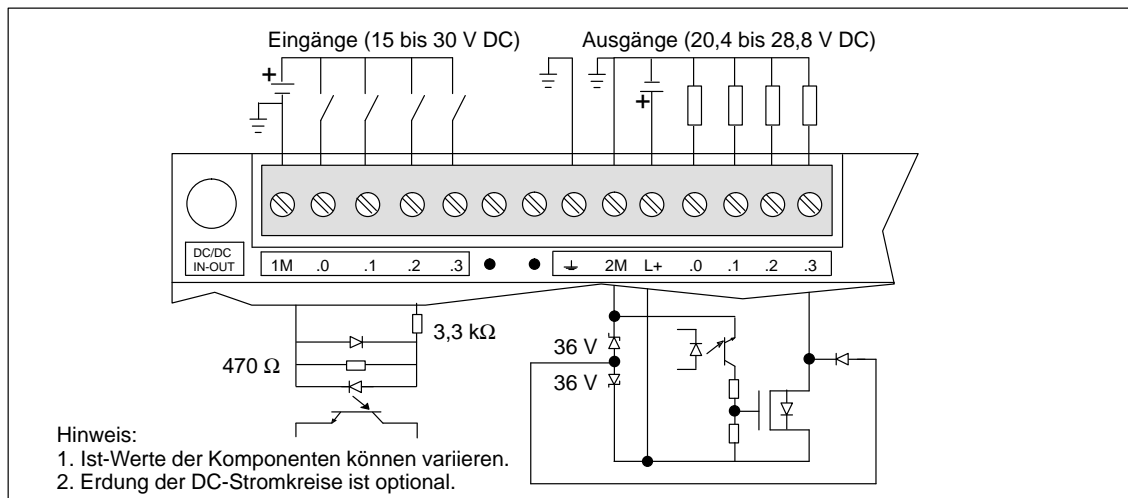


Bild A-26 Kennzeichnung der Anschlüsse beim EM 223 Digitalein-/Digitalausgabe 4 x 24-V-DC-Eingang/4 x 24-V-DC-Ausgang

A.27 Erweiterungsmodul EM 223, Digitalein-/Digitalausgabe 8 x 24-V-DC-Eingang/8 x 24-V-DC-Ausgang

Bestellnummer: 6ES7 223-1BH00-0XA0

Leistungsmerkmale		Eingänge	
Abmessungen (B x H x T)	90 x 80 x 62 mm	Eingangstyp	stromziehend/stromliefernd IEC 1131 Typ 1 wenn stromziehend
Gewicht	0.2 kg	Bereich bei EIN	15 bis 30 V DC, min. 4 mA 35 V DC, 500 ms Spannungstoß
Stromaufnahme	3.0 W	Nennspannung bei EIN	24 V DC, 7 mA
Ein- und Ausgänge ¹	8 digitale Eingänge 8 digitale Ausgänge	Maximum bei AUS	5 V DC, 1 mA
Normen	UL 508 CSA C22.2 142 FM Klasse I, Kategorie 2 gemäß VDE 0160 gemäß EG-Richtlinie	Ansprechzeit	max. 4,0 ms
Ausgänge		Potentialtrennung	500 V AC, 1 min
Ausgangstyp	MOSFET, stromliefernd	Strombedarf	
Spannungsbereich	20,4 V DC bis 28,8 V DC	5-V-DC-logische-Spannung	120 mA von Zentralgerät
Max. Laststrom	0 bis 55° C	24-V-DC-Geberspannung	60 mA von Zentralgerät oder externer Stromversorgung
Ausgänge können für höheren Strom parallel geschaltet werden.	0,5 A/Ausgang	Strom an Ausgängen	Wird vom Anwender am Modulleiter geliefert.
Ableitstrom	200 µA		
Verzögerung Schaltvorgänge	150 µs EIN, 400 µs AUS		
Widerstand	max. 400 mΩ		
Kurzschlußschutz	0,7 bis 1,5 A/Kanal		
Potentialtrennung	500 V AC, 1 min		

¹ In der CPU sind 8 Eingänge im Prozeßabbild der Eingänge und 8 Ausgänge im Prozeßabbild der Ausgänge für dieses Modul vorgesehen.

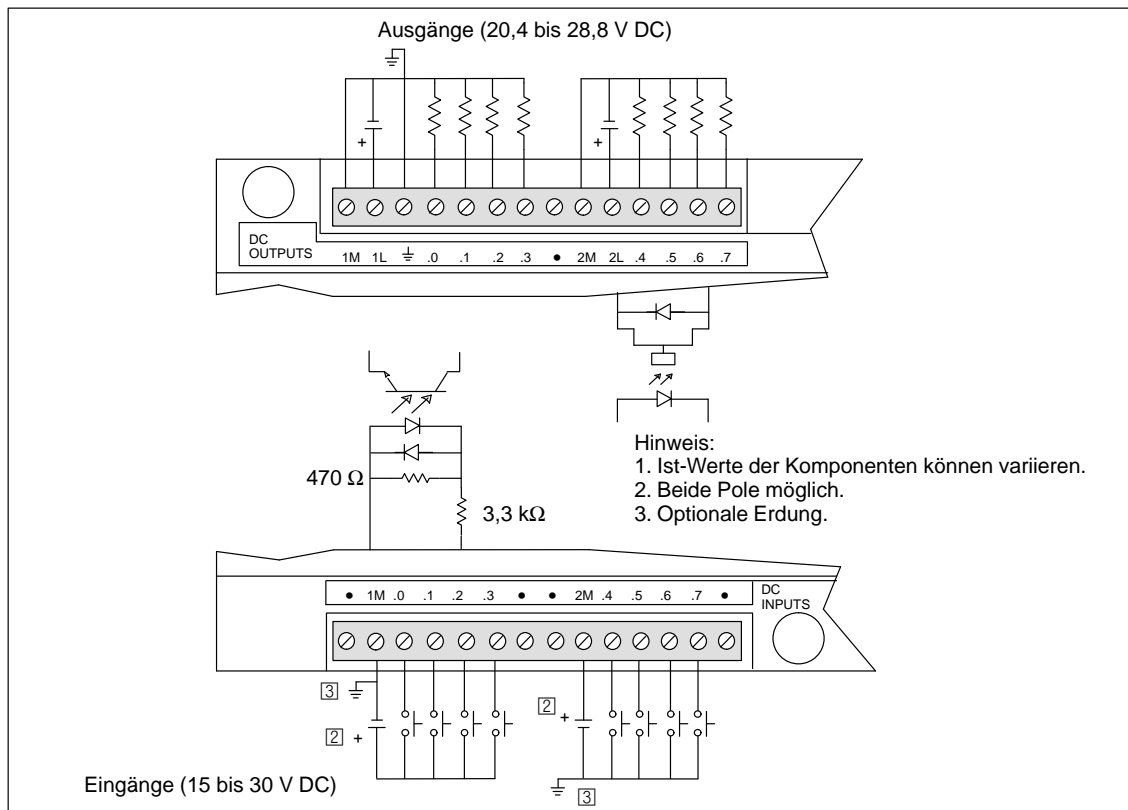


Bild A-27 Kennzeichnung der Anschlüsse beim EM 223, Digitalein-/Digitalausgabe 8 x 24-V-DC-Eingang / 8 x 24-V-DC-Ausgang

A.28 Erweiterungsmodul EM 223, Digitalein-/Digitalausgabe 16 x 24-V-DC-Eingang/16 x 24-V-DC-Ausgang

Bestellnummer: 6ES7 223-1BL00-0XA0

Leistungsmerkmale		Eingänge	
Abmessungen (B x H x T)	160 x 80 x 62 mm	Eingangstyp	stromziehend/stromliefernd IEC 1131 Typ 1 wenn stromziehend
Gewicht	0,4 kg	Bereich bei EIN	15 bis 30 V DC, min. 4 mA Spannungsstoß
Stromaufnahme	5.5 W	Nennspannung bei EIN	24 V DC, 7 mA
Ein- und Ausgänge ¹	16 digitale Eingänge 16 digitale Ausgänge	Maximum bei AUS	5 V DC, 1 mA
Normen	UL 508 CSA C22.2 142 FM Klasse I, Kategorie 2 gemäß VDE 0160 gemäß EG-Richtlinie	Ansprechzeit	max. 4.0 ms
Ausgänge		Potentialtrennung	500 V AC, 1 Minute
Ausgangstyp	MOSFET, stromliefernd	Strombedarf	
Spannungsbereich	20,4 V DC bis 28,8 V DC	5-V-DC-logische-Spannung	210 mA von Zentralgerät
Max. Laststrom	0 bis 55 C	24-V-DC-Geberspannung	120 mA von Zentralgerät oder externer Stromversorgung
Ausgänge können für höheren Strom parallel geschaltet werden.	0,5 A/Ausgang	Strom an Ausgängen	Wird vom Anwender am Modulleiter geliefert.
Ableitstrom	200 µA		
Verzögerung Schaltvorgänge	150 µs EIN, 400 µs AUS		
Widerstand	max. 400 mΩ		
Kurzschlußschutz	0,7 bis 1,5 A/Kanal		
Potentialtrennung	500 V AC, 1 Minute		

¹ In der CPU sind 16 Eingänge im Prozeßabbild der Eingänge und 16 Ausgänge im Prozeßabbild der Ausgänge für dieses Modul vorgesehen.

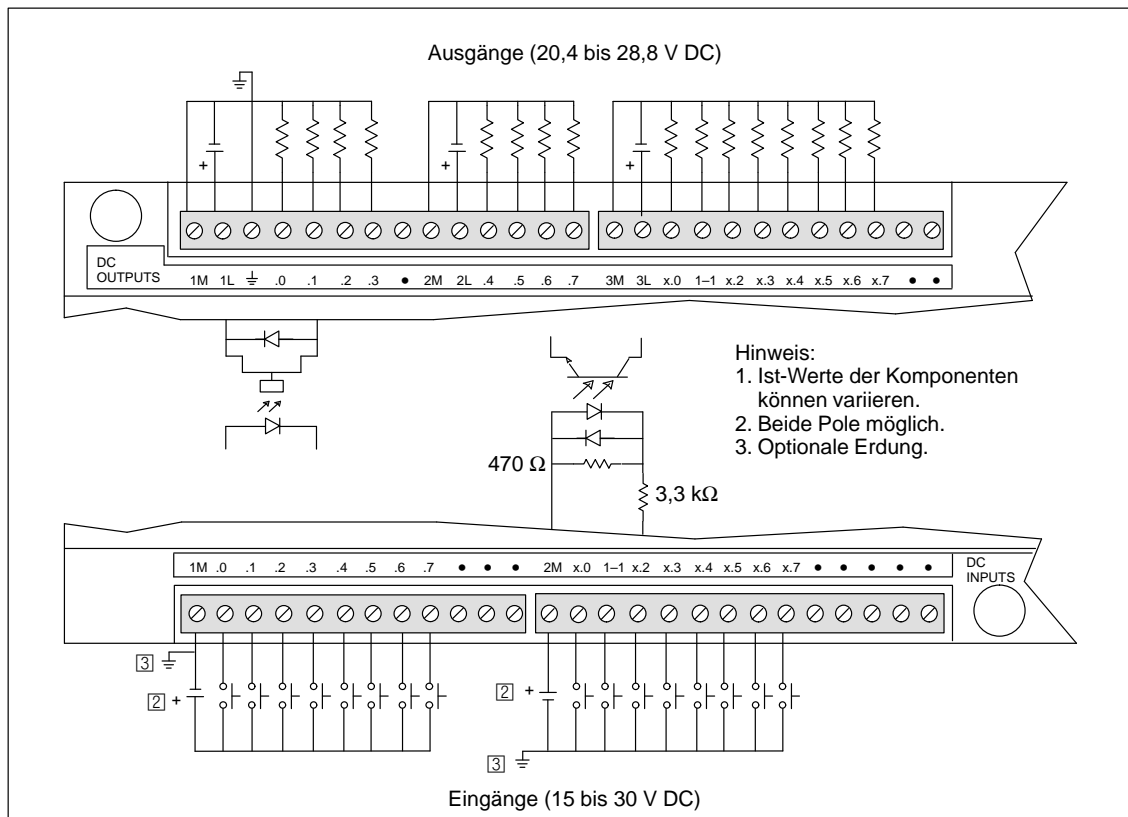


Bild A-28 Kennzeichnung der Anschlüsse beim EM 223 Digitalein-/Digitalausgabe 16 x 24-V-DC-Eingang/16 x 24-V-DC-Ausgang

A.29 Erweiterungsmodul EM 223, Digitalein-/Digitalausgabe 4 x 24-V-DC-Eingang/4 x Relaisausgang

Bestellnummer: 6ES7 223-1HF00-0XA0

Leistungsmerkmale		Ausgänge, Fortsetzung	
Abmessungen (B x H x T)	90 x 80 x 62 mm	Kontaktwiderstand	max. 200 mΩ (neu)
Gewicht	0,2 kg	Kurzschlußschutz	keine
Stromaufnahme	2 W	Eingänge	
Ein- und Ausgänge ¹	4 digitale Eingänge 4 digitale Relaisausgänge	Eingangstyp	Typ 1, stromziehend gemäß IEC 1131-2
Normen	UL 508 CSA C22.2 142 FM Klasse I, Kategorie 2 gemäß VDE 0160 gemäß EG-Richtlinie	Bereich bei EIN	15 bis 30 V DC, min. 4 mA 35 V DC, 500 ms Spannungsstoß
Ausgänge		Nennspannung bei EIN	24 V DC, 7 mA
Ausgangstyp	Relais, Schwachstromkontakt	Maximum bei AUS	5 V DC, 1 mA
Spannungsbereich	5 bis 30 V DC/250 V AC	Ansprechzeit	typ. 3,5 ms./max. 4,5 ms
Max. Laststrom	2 A/Ausgang	Potentialtrennung	500 V AC, 1 min
Isolationswiderstand	min. 100 MΩ (neu)	Strombedarf	
Verzögerung Schaltvorgänge	max. 10 ms	5-V-DC-logische-Spannung	80 mA von Zentralgerät
Lebensdauer	10 000 000 mechanisch 100 000 mit Bemessungslast	24-V-DC-Geberspannung	30 mA von Zentralgerät oder externer Stromversorgung
Elektrische Trennung		Ausgangsspannung 24 V DC	35 mA von Zentralgerät oder externer Stromversorgung
Spule zu Kontakt	1500 V AC, 1 min	Strom an Ausgängen	Wird vom Anwender am Modulleiter geliefert.
Kontakt zu Kontakt (zw. geöffn. Kontakten)	750 V AC, 1 min		

¹ In der CPU sind 8 Eingänge im Prozeßabbild der Eingänge und 8 Ausgänge im Prozeßabbild der Ausgänge für dieses Modul vorgesehen.

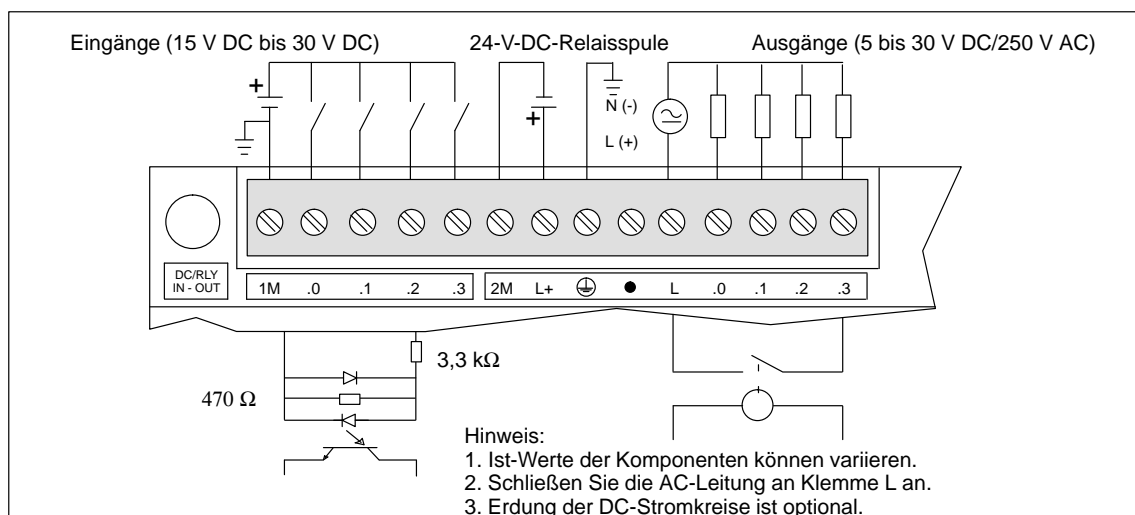


Bild A-29 Kennzeichnung der Anschlüsse beim EM 223, Digitalein-/Digitalausgabe 4 x 24-V-DC-Eingang / 4 x Relaisausgang

A.30 Erweiterungsmodul EM 223, Digitalein-/Digitalausgabe 4 x 120-V-AC-Eingang/4 x 120/230-V-AC-Ausgang

Bestellnummer: 6ES7 223-1EF00-0XA0

Leistungsmerkmale		Ausgänge, Fortsetzung	
Abmessungen (B x H x T)	90 x 80 x 62 mm	Stoßstrom	50 A Spitze, 1 Zyklus 15 A Spitze, 5 Zyklen
Gewicht	0,2 kg	Spannungsabfall	max. 1,8 V bei max. Strom
Stromaufnahme	5,5 W bei 3 A Last	Potentialtrennung	1500 V AC, 1 min
Ein- und Ausgänge ¹	4 digitale Eingänge 4 digitale Ausgänge	Kurzschlußschutz	keine
Normen	UL 508 CSA C22.2 142 FM Klasse I, Kategorie 2 gemäß EG-Richtlinie	Eingänge	
Ausgänge		Eingangstyp	Typ 1, stromziehend gemäß IEC 1131-2
Ausgangstyp	Triac, Nulldurchgang einschalten	Bereich bei EIN	79 bis 135 V AC, 47 bis 63 Hz min. 4 mA
Spannungs-/Frequenzbereich	70 - 264 V AC, 47 - 63 Hz	Nennspannung bei EIN	120 V AC, 60 Hz, 7 mA
Leistungsfaktor	0,3 bis 1,0	Maximum bei AUS	20 V AC, 1 mA
Laststromkreis		Ansprechzeit	max. 15 ms
Max. Laststrom	0 bis 40° C 55° C ² pro einzeltem Ausgang 2,40 A 2,00 A Ausgänge gesamt 4,00 A 3,00 A	Potentialtrennung	1500 V AC, 1 min
Min. Laststrom	10 mA	Strombedarf	
Ableitstrom	2,5 mA, 120 V 4,0 mA, 230 V	5-V-DC-logische-Spannung	100 mA von Zentralgerät
Verzögerung Schaltvorgänge	1/2 Zyklus	Strom an Ausgängen	Wird vom Anwender am Modulleiter geliefert.

¹ In der CPU sind 8 Eingänge im Prozeßabbild der Eingänge und 8 Ausgänge im Prozeßabbild der Ausgänge für dieses Modul vorgesehen.

² Lineare Leistungsminderung 40 bis 55° C, Leistungsminderung bei vertikalem Einbau 10° C.

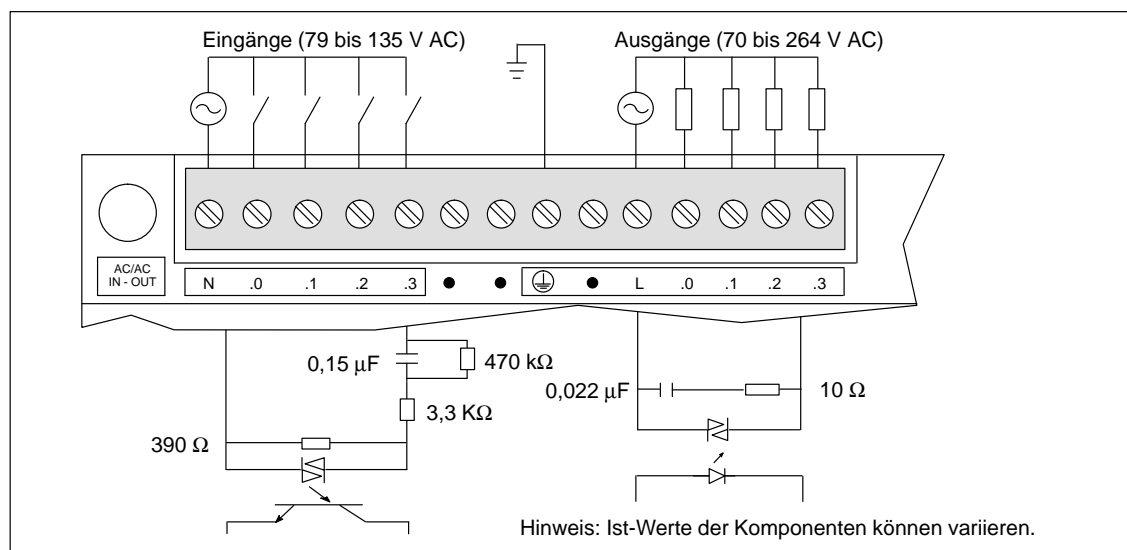


Bild A-30 Kennzeichnung der Anschlüsse beim EM 223, Digitalein-/Digitalausgabe 4 x 120-V-AC-Eingang/
4 x 120/230-V-AC-Ausgang

A.31 Erweiterungsmodul EM 223, Digitalein-/Digitalausgabe 8 x 24-V-DC-Eingang/8 x Relaisausgang

Bestellnummer: 6ES7 223-1PH00-0XA0

Leistungsmerkmale		Eingänge	
Abmessungen (B x H x T)	90 x 80 x 62 mm	Eingangstyp	stromziehend/stromliefernd IEC 1131 Typ 1 wenn stromziehend
Gewicht	0,3 kg	Bereich bei EIN	15 bis 30 V DC, min. 4 mA 35 V DC, 500 ms Spannungstoß
Stromaufnahme	2.5 W	Nennspannung bei EIN	24 V DC, 7 mA
Ein- und Ausgänge ¹	8 digitale Eingänge 8 digitale Ausgänge	Maximum bei AUS	5 V DC, 1 mA
Normen	UL 508 CSA C22.2 142 FM Klasse I, Kategorie 2 gemäß VDE 0160 gemäß EG-Richtlinie	Ansprechzeit	max. 4,0 ms
Ausgänge		Potentialtrennung	500 V AC, 1 min
Ausgangstyp	Relais, Schwachstromkontakt	Strombedarf	
Spannungsbereich	5 bis 30 V DC/250 V AC	5-V-DC-logische-Spannung	100 mA von Zentralgerät
Max. Laststrom	2 A/Ausgang, 8 A/Leiter	24-V-DC-Geberspannung	90 mA von Zentralgerät oder externer Stromversorgung
Isolationswiderstand	max. 100 M Ω (neu)	Strom an Ausgängen	Wird vom Anwender am Modulleiter geliefert.
Verzögerung Schaltvorgänge	max. 10 ms		
Lebensdauer	10 000 000 mechanisch 100 000 mit Bemessungslast		
Kontaktwiderstand	max. 200 m Ω (neu)		
Elektrische Trennung			
Spule zu Kontakt	1500 V AC, 1 min		
Kontakt zu Kontakt (zw. geöffn. Kontakten)	750 V AC, 1 min		
Kurzschlußschutz	keine		

¹ In der CPU sind 8 Eingänge im Prozeßabbild der Eingänge und 8 Ausgänge im Prozeßabbild der Ausgänge für dieses Modul vorgesehen.

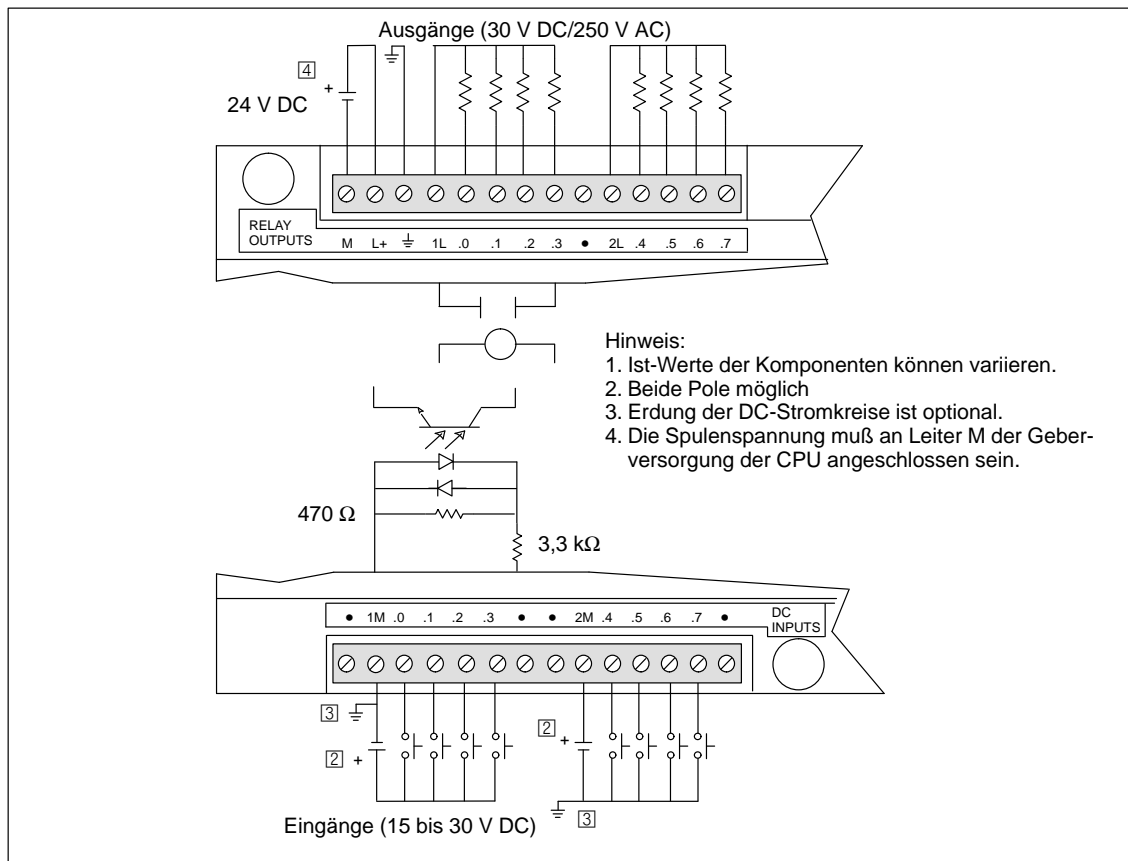


Bild A-31 Kennzeichnung der Anschlüsse beim EM 223 Digitalein-/Digitalausgabe 8 x 24-V-DC-Eingang / 8 x Relaisausgang

A.32 Erweiterungsmodul EM 223, Digitalein-/Digitalausgabe 16 x 24-V-DC-Eingang/16 x Relaisausgang

Bestellnummer: 6ES7 223-1PL00-0XA0

Leistungsmerkmale		Eingänge	
Abmessungen (B x H x T)	160 x 80 x 62 mm	Eingangstyp	stromziehend/stromliefernd IEC 1131 Typ 1 wenn stromziehend
Gewicht	0,45 kg	Bereich bei EIN	15 bis 30 V DC, min. 4 mA Spannungsstoß
Stromaufnahme	7 W	Nennspannung bei EIN	24 V DC, 7 mA
Ein- und Ausgänge ¹	16 digitale Eingänge 16 digitale Ausgänge	Maximum bei AUS	5 V DC, 1 mA
Normen	UL 508 CSA C22.2 142 FM Klasse I, Kategorie 2 gemäß VDE 0160 gemäß EG-Richtlinie	Ansprechzeit	typ. 3,5 ms/max. 4,5 ms
Ausgänge		Potentialtrennung	AC 500 V, 1 Minute
Ausgangstyp	Relais, Schwachstromkontakt	Strombedarf	
Spannungsbereich	5 bis 30 V DC/250 V AC	5-V-DC-logische-Spannung	160 mA von Zentralgerät
Max. Laststrom	2 A/Ausgang, 8 A/Leiter	24-V-DC-Geberspannung	120 mA von Zentralgerät oder externer Stromversorgung
Isolationswiderstand	max. 100 MΩ (neu)	24-V-DC-Ausgangsspannung ²	130 mA von Zentralgerät oder ext. Stromversorgung
Verzögerung Schaltvorgänge	max. 10 ms	Strom an Ausgängen	Wird vom Anwender am Modulleiter geliefert.
Lebensdauer	10 000 000 mechanisch 100 000 mit Bemessungslast		
Kontaktwiderstand	max. 200 mΩ (neu)		
Elektrische Trennung			
Spule zu Kontakt	1500 V AC, 1 min		
Kontakt zu Kontakt (zwischen geöffneten Kontakten)	750 V AC, 1 min		
Kurzschlußschutz	keine		

¹ In der CPU sind 16 Eingänge im Prozeßabbild der Eingänge und 16 Ausgänge im Prozeßabbild der Ausgänge für dieses Modul vorgesehen.

² Die Spulenspannung muß an Leiter M der Geberversorgung der CPU angeschlossen sein.

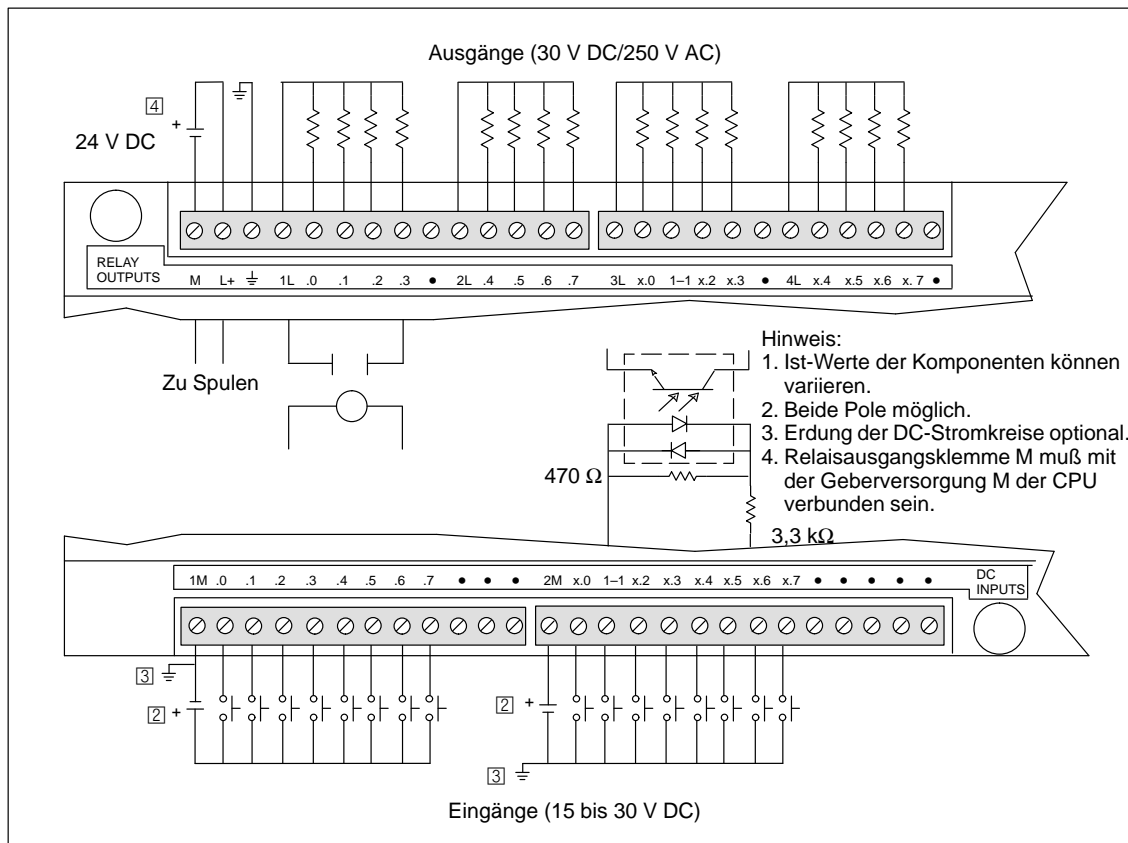


Bild A-32 Kennzeichnung der Anschlüsse beim EM 223 Digitalein-/Digitalausgabe 16 x 24 V-DC-Eingang / 16 x Relaisausgang

A.33 Erweiterungsmodul EM 231, Analogeingabe AE 3 x 12 Bit

Bestellnummer: 6ES7 231-0HC00-0XA0

Leistungsmerkmale		Eingänge, Fortsetzung	
Abmessungen (B x H x T)	90 x 80 x 62 mm	Analog-Digital-Umsetzzeit	< 250 μ s
Gewicht	0,2 kg	Analoge Sprungantwort	1.5 ms bis 95%
Stromaufnahme	2 W	Gleichtaktunterdrückung	40 dB, Nennwert bei 60 Hz
Ein- und Ausgänge ¹	3 Analogeingänge	Gleichtaktspannung	Signalspannung plus Gleichtaktspannung, kleiner gleich 12 V
Normen	UL 508 CSA C22.2 142 FM Klasse I, Kategorie 2 gemäß VDE 0160 gemäß EG-Richtlinie	Datenwortformat ² einpölig, Vollbereich	0 bis 32000
Eingänge		Strombedarf	
Eingangstyp	Differentialeingang	5-V-DC-logische-Spannung	70 mA aus Zentralgerät
Eingangsimpedanz	$\geq 10 \text{ M}\Omega$	Externe Stromversorgung	60 mA aus Zentralgerät oder externer Stromquelle (24-V-DC-Nennwert, Klasse 2 oder DC-Geberversorgung)
Eingabefilterdämpfung	-3 db @ 3,1 kHz	Anzeige LED, EXTf	
Max. Eingangsspannung	30 V	Netzausfall	Spannungsabfall, in externer 24-V-DC-Versorgung
Max. Eingangsstrom	32 mA		
Auflösung	12-Bit-A/D-Umsetzung		
Elektrische Trennung	Nein		

¹ In der CPU sind 4 Analogeingänge für dieses Modul vorgesehen.

² Datenwort: 8 Zählpulse pro Inkrement, linksbündige Werte, siehe Bild A-35.

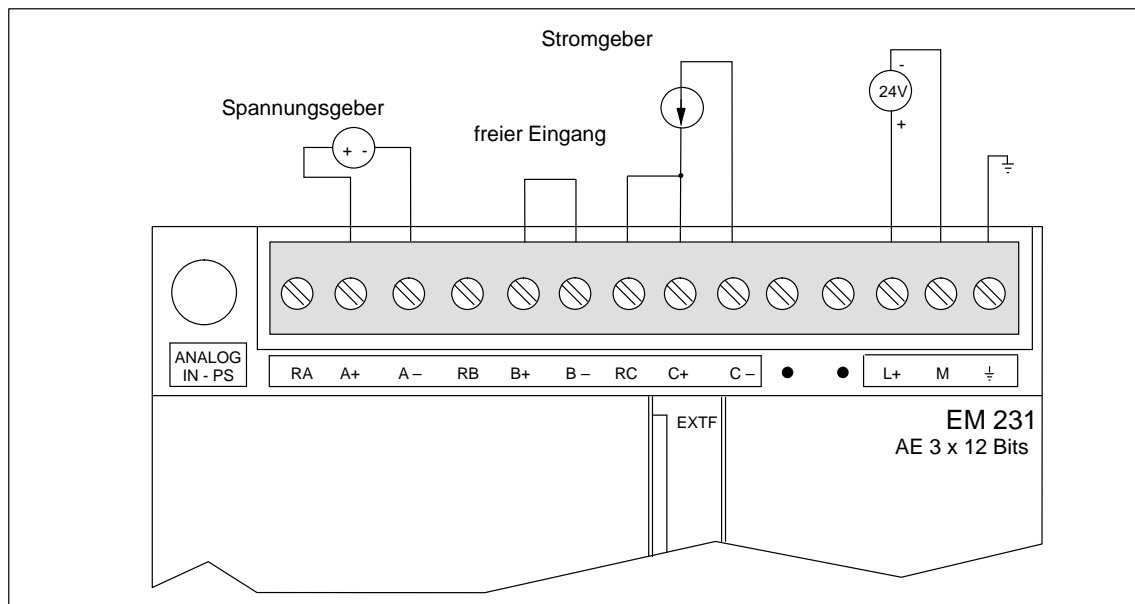


Bild A-33 Kennzeichnung der Anschlüsse beim Erweiterungsmodul EM 231, Analogeingabe AE 3 x 12 Bit

Kalibrierung und Konfiguration

Das Kalibrierungspotentiometer und die DIP-Konfigurationsschalter sind durch die Lüftungsschlitze des Moduls zugänglich (siehe Bild A-34).

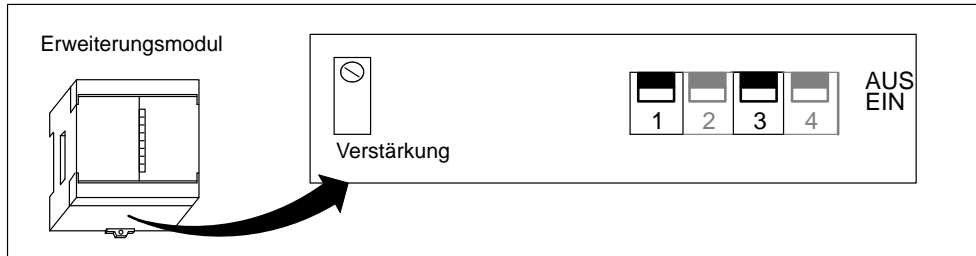


Bild A-34 Kalibrierungspotentiometer und DIP-Konfigurationsschalter

Konfiguration

Tabelle A-2 zeigt, wie Sie das Modul mit den DIP-Schaltern konfigurieren. Die Schalter 1 und 3 wählen den Bereich der Analogeingänge aus. Alle Eingänge werden auf den gleichen Bereich der Analogeingänge gesetzt.

Tabelle A-2 Konfigurationsschalter für das Analogeingabemodul EM 231

Konfigurationsschalter		Vollausschlag	Auflösung
1	3		
EIN	AUS	0 bis 5 V	1,25 mV
EIN	AUS	0 bis 20 mA ¹	5 µA
AUS	EIN	0 bis 10 V	2,5 mV

¹ 0 bis 20 mA nach Messung mit internem in Stromrichtung geschaltetem 250-Ω-Widerstand.

Kalibrieren eines Eingangs

Durch das Kalibrieren eines Moduls können Sie Verstärkungsfehler bei Vollausschlag korrigieren. Versatzfehler werden nicht behoben. Das Kalibrieren wirkt sich auf alle drei Eingangskanäle aus. Es kann nach dem Kalibrieren zu unterschiedlichen Werten der Kanäle kommen.

Um das Modul genau zu kalibrieren, müssen Sie mit einem Programm arbeiten, das aus den aus dem Modul ausgelesenen Werten einen Mittelwert bildet. Arbeiten Sie mit dem Analog-eingabefilter-Assistenten in STEP 7-Micro/WIN, um dieses Programm zu erstellen (siehe Abschnitt 5.3). Bilden Sie den Mittelwert aus mindestens 64 abgefragten Werten.

Zum Kalibrieren eines Eingangs gehen Sie folgendermaßen vor:

1. Schalten Sie die Spannungsversorgung für das Modul aus. Wählen Sie den gewünschten Eingangsbereich.
2. Schalten Sie die Spannungsversorgung für die CPU und das Modul ein. Warten Sie ca. 15 Minuten, damit sich das Modul stabilisieren kann.
3. Legen Sie mittels eines Spannungs- oder Stromgebers an einem der Eingänge ein Nullsignal an.
4. Lesen Sie den Wert, der von dem entsprechenden Eingangskanal an die CPU übermittelt wird. Die Werte für das Eingangssignal "0" zeigen die Größe des Versatzfehlers an. Dieser Fehler kann durch Kalibrieren nicht behoben werden.
5. Legen Sie an einem Eingang den Wert des Vollausschlags an. Lesen Sie den Wert aus, den die CPU empfangen hat.
6. Stellen Sie mit dem Verstärkungspotentiometer den Wert 32.000 bzw. den gewünschten digitalen Datenwert ein.

Datenwortformat

Bild A-35 zeigt die Anordnung des 12-Bit-Datenwerts im Analogeingangswort der CPU.

Weicht die Wiederholgenauigkeit im Vollbereich um nur $\pm 0,45\%$ ab, kann dies eine Abweichung von ± 144 Zählimpulsen bei dem Wert hervorrufen, der aus dem Analogeingang ausgelesen wird.

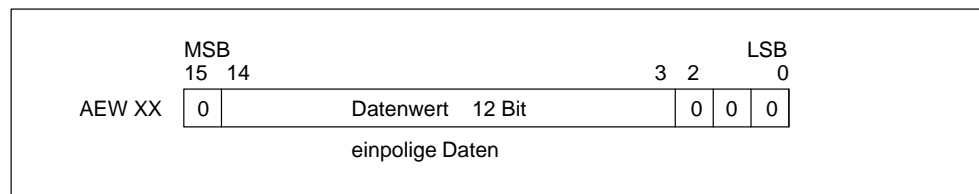


Bild A-35 Datenwortformat

Hinweis

Die 12 Bit eines Werts der Analog-Digital-Umsetzung sind im Datenwortformat linksbündig angeordnet. Das höchstwertige Bit gibt das Vorzeichen an: Null zeigt an, daß der Wert des Datenworts positiv ist. Die drei Nullen am Ende bewirken, daß sich das Datenwort bei jeder Änderung des Zählimpulses im Wert der Analog-Digital-Umsetzung um acht Zählimpulse ändert.

Schaltbild der Eingänge

Bild A-36 zeigt das Schaltbild der Eingänge beim EM 231.

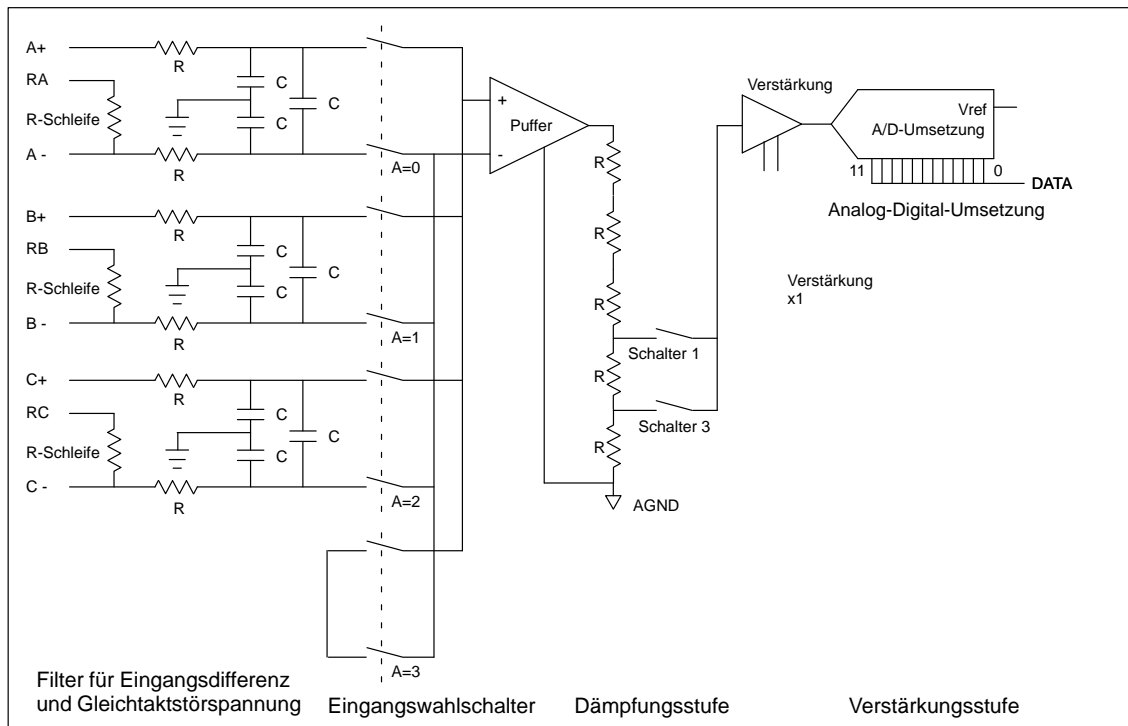


Bild A-36 Schaltbild der Eingänge beim EM 231

Richtlinien für die Installation des EM 231

Beachten Sie die folgenden Richtlinien, damit Genauigkeit und Wiederholbarkeit sichergestellt sind:

- Achten Sie darauf, daß die 24-V-DC-Geberversorgung störfest ist.
- Kalibrieren Sie das Modul.
- Verdrahten Sie die Geberversorgung so kurz wie möglich.
- Verwenden Sie für die Verdrahtung der Geberversorgung geschirmte verdrillte Doppelleitungen.
- Schließen Sie die Schirmung nur an der Geberversorgungsseite ab.
- Überbrücken Sie die Eingänge für freie Kanäle (siehe Bild A-33).
- Vermeiden Sie es, die Leitungen scharf zu knicken.
- Verlegen Sie die Leitungen in Kabelkanälen.
- Achten Sie darauf, daß die Eingangssignale potentialfrei sind oder als Bezugsleiter den externen 24-V-Leiter des Analogmoduls haben.

Arbeiten mit dem Analogeingabemodul: Genauigkeit und Wiederholbarkeit

Das Analogeingabemodul EM 231 ist ein preiswertes, Hochgeschwindigkeits-Analogeingabemodul (12 Bit). Das Modul kann eine Analogeingabe in den entsprechenden Digitalwert in 171 μ s bei der CPU 212 und in 139 μ s bei allen anderen S7-200 CPUs umwandeln. Die Umwandlung des Analogeingabesignals wird jedesmal durchgeführt, wenn Ihr Programm auf den Analogeingang zugreift. Diese Zeitangaben müssen zu der grundlegenden Ausführungszeit der Operation, die auf den Analogeingang zugreift, addiert werden.

Das Erweiterungsmodul EM 231 liefert einen nicht bearbeiteten Digitalwert (keine Linearisierung oder Filterung), der der Analogspannung bzw. dem Analogstrom an den Eingangsklemmen des Moduls entspricht. Da es sich bei dem Modul um ein Hochgeschwindigkeitsmodul handelt, kann es schnellen Änderungen im Analogeingabesignal folgen (einschließlich internem und externem Rauschen). Abweichungen von einem Wert zum nächsten, die durch Störungen in einer Konstante oder in einem sich langsam ändernden Analogeingabesignal verursacht werden, können durch Mittelwertbildung aus einer Reihe von Werten verringert werden. Je weiter Sie die Anzahl der für die Mittelwertbildung herangezogenen Werte erhöhen, desto stärker können Sie eine entsprechend langsamere Reaktionszeit auf Änderungen im Eingabesignal feststellen.

Sie können mit dem Analogeingabefilter-Assistenten in STEP 7-Micro/WIN arbeiten (siehe Abschnitt 5.3), um ein Programm zur Mittelwertbildung in Ihr Programm einzufügen. Beachten Sie, daß ein Mittelwert, der aus einer hohen Zahl abgetasteter Werte gebildet wird, den Wert stabilisiert und gleichzeitig die Reaktion des Werts auf Änderungen im Eingabesignal verlangsamt. Sollen die Analogeingabesignale nur langsam geändert werden, empfehlen wir eine Mindestzahl von 64 abgefragten Werten für das Programm zur Mittelwertbildung.

Die Angaben zur Wiederholbarkeit beschreiben die Abweichungen zwischen den Werten bei gleichbleibendem Eingabesignal. Außerdem definieren die Angaben zur Wiederholbarkeit den Bereich, der 99% aller Werte enthält. Die mittlere Genauigkeit beschreibt den durchschnittlichen Wert des Fehlers (die Differenz zwischen dem Durchschnittswert der einzelnen Werte und dem genauen Wert des tatsächlichen Analogeingabesignals). Die Wiederholbarkeit wird in Bild A-37 durch die Kurve dargestellt. Dieses Bild zeigt die Grenzwerte für die Wiederholbarkeit von 99%, den mittleren bzw. durchschnittlichen Wert aus den einzelnen Werten und die mittlere Genauigkeit in graphischer Form. Tabelle A-3 enthält die Angaben zur Wiederholbarkeit und zur mittleren Genauigkeit in bezug auf die konfigurierbaren Bereiche.

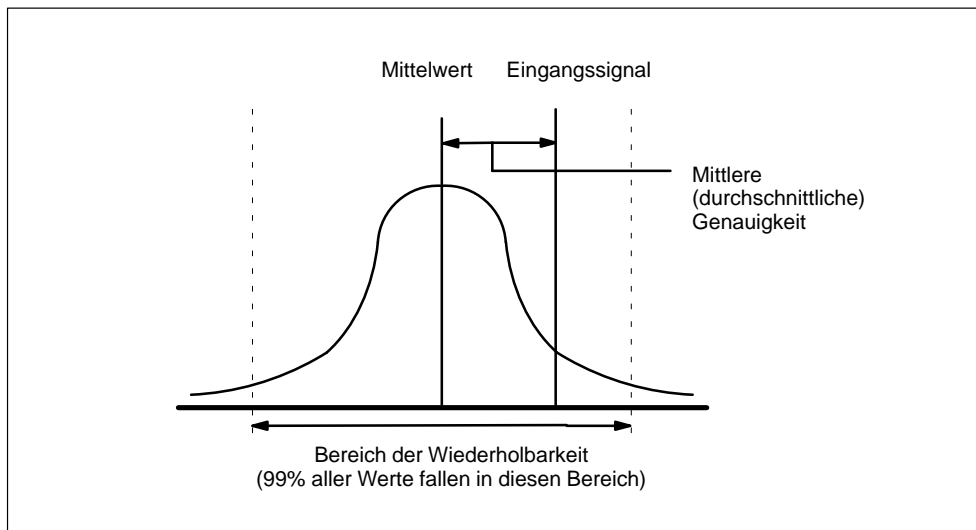


Bild A-37 Definition der Genauigkeit

Tabelle A-3 Angaben für S7-200 CPUs (AC- und DC-Varianten)

Eingangsbereich bei Vollausschlag	Wiederholbarkeit ¹		Mittlere (durchschnittliche) Genauigkeit ^{1, 2, 3, 4}	
	% des Vollausschlags	Zählimpulse	% des Vollausschlags	Zählimpulse
Angaben für DC-Varianten der S7-200 CPUs				
0 bis 5 V	± 0,075%	± 24	± 0,1%	± 32
0 bis 20 mA				
0 bis 10 V				
Angaben für AC-Varianten der S7-200 CPUs				
0 bis 5 V	± 0,15%	± 48	± 0,1%	± 64
0 bis 20 mA				
0 bis 10 V				

¹ Messungen nach durchgeführter Kalibrierung des Eingangsbereichs.

² Versatzfehler bei Signal nahe Null des Analogeingangs werden nicht korrigiert und nicht in den Angaben zur Genauigkeit berücksichtigt.

³ Beim Übertragen von Kanal zu Kanal tritt aufgrund der endlichen Ausregelzeit des Analog-Multiplexers ein Umwandlungsfehler auf. Maximaler Übertragungsfehler ist 0,1% des Unterschieds zwischen den Kanälen.

⁴ Die mittlere Genauigkeit umfaßt Auswirkungen von Nicht-Linearität und Drift zwischen 0 und 55 Grad C.

A.34 Erweiterungsmodul EM 232, Analogausgabe AA 2 x 12 Bit

Bestellnummer: 6ES7 232-0HB00-0XA0

Leistungsmerkmale	
Abmessungen (B x H x T)	90 x 80 x 62 mm
Gewicht	0,2 kg
Stromaufnahme	2 W
Ausgänge ¹	2 Analogausgänge
Normen	UL 508 CSA C22.2 142 FM Klasse I, Kategorie 2 gemäß VDE 0160 gemäß EG-Richtlinie
Ausgänge	
Signalbereich	
Ausgangsspannung	± 10 V
Ausgangsstrom	0 bis 20 mA
Auflösung, Vollausschlag	
Spannung	12 Bit
Strom	11 Bit
Auflösung, Vollausschlag	
Spannung, zweipolig	1 in 2000 Zählimpuls, 0,5% des Vollausschlags pro Zählimpuls
Strom, einpolig	1 in 2000 Zählimpuls, 0,5% des Vollausschlags pro Zählimpuls
Datenwortformat	
Vollbereich	
Spannung, zweipolig	-32768 bis + 32752
Strom, einpolig	0 bis +32752
Vollausschlag	
zweipolig	-32000 bis +32000
einpolig	0 bis + 32000

Genauigkeit	
Ungünstigster Fall, 0 - 55° C	
Ausgangsspannung	± 2% des Vollausschlags
Ausgangsstrom	± 2% des Vollausschlags
Typ., 25° C	
Ausgangsspannung	± 0,5% des Vollausschlags
Ausgangsstrom	± 0,5% des Vollausschlags
Ausregelzeit	
Ausgangsspannung	100 µs
Ausgangsstrom	2 ms
Max. Ansteuerung	
bei 24-V-Versorgung	
Ausgangsspannung	min. 5000Ω
Ausgangsstrom	max. 500Ω
Strombedarf	
5-V-DC-logische-Spannung	70 mA aus Zentralgerät
Externe Stromversorgung	60 mA, plus Ausgangsstrom von 40 mA aus Zentralgerät oder externer Versorgung (24-V-DC-Nennspannung, Klasse 2 oder DC-Geberversorgung)
Anzeige LED, EXTf	
Netzausfall	Spannungsabfall, Bereichsüber- oder -unterschreitung

¹ In der CPU sind 2 Analogausgänge für dieses Modul vorgesehen.

Bild A-38 zeigt die Kennzeichnung der Anschlüsse beim EM 232, Analogausgabe AA 2 x 12 Bit.

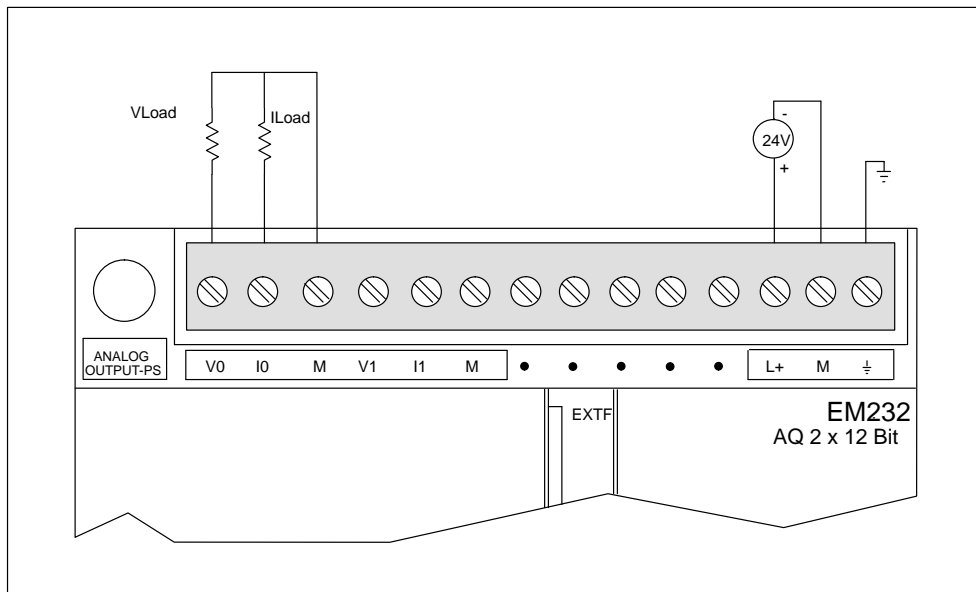


Bild A-38 Connector Terminal Identification for Expansion Module EM232 Analog Output AQ 2 x 12 Bits

Datenwortformat

Bild A-39 zeigt die Anordnung des 12-Bit-Datenwerts im Analogausgangswort der CPU.

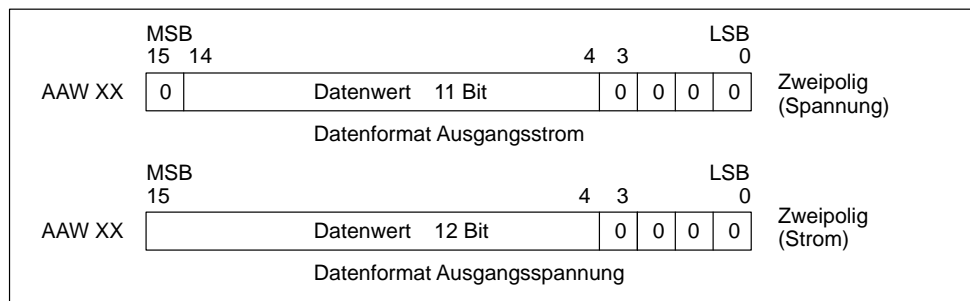


Bild A-39 Datenwortformat

Hinweis

Die 12 Bit eines Werts der Analog-Digital-Umsetzung sind im Datenwortformat linksbündig angeordnet. Das höchstwertige Bit gibt das Vorzeichen an: Null zeigt an, daß der Wert des Datenworts positiv ist. Die vier Nullen am Ende werden vor dem Laden in das Register für den Wert der Analog-Digital-Umsetzung gekürzt. Diese Bits wirken sich nicht auf den Wert des Ausgangssignals aus.

Schaltbild der Ausgänge

Bild A-40 zeigt das Schaltbild der Ausgänge beim EM 232.

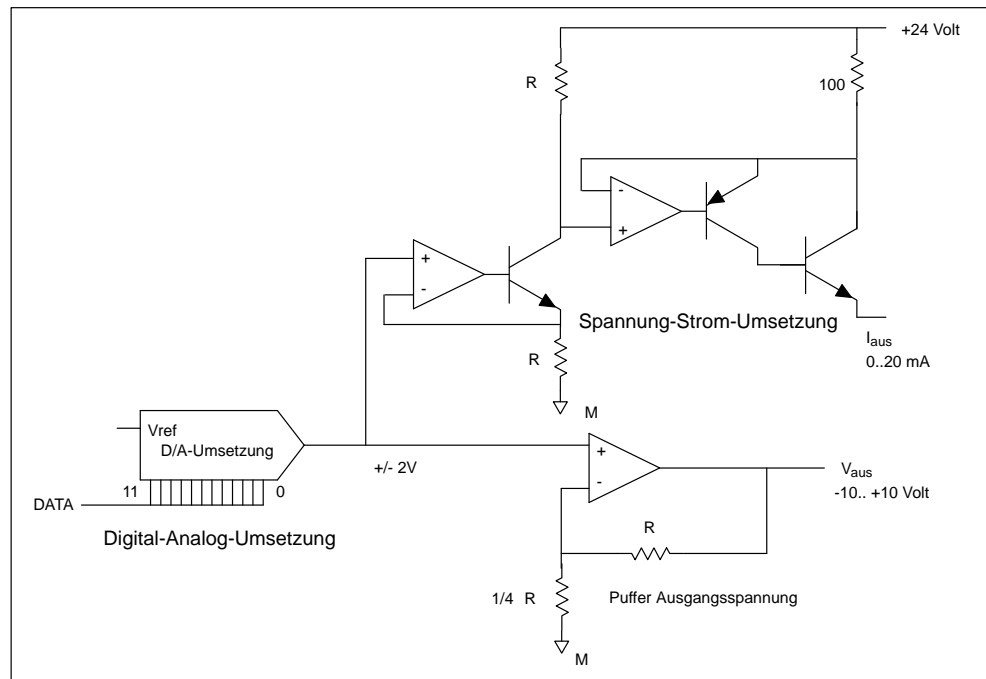


Bild A-40 Schaltbild der Ausgänge beim EM 232

Richtlinien für die Installation des EM 232

Beachten Sie die folgenden Richtlinien, damit Genauigkeit sichergestellt ist:

- Achten Sie darauf, daß die 24-V-DC-Geberversorgung störtest ist.
- Verdrahten Sie die Geberversorgung so kurz wie möglich.
- Verwenden Sie für die Verdrahtung der Geberversorgung geschirmte verdrehte Doppelleitungen.
- Schließen Sie die Schirmung nur an der Geberversorgungsseite ab.
- Vermeiden Sie es, die Leitungen scharf zu knicken.
- Verlegen Sie die Leitungen in Kabelkanälen.
- Verlegen Sie Signalleitungen nicht parallel zu Starkstromleitungen. Kreuzen sich die beiden Leitungen, lassen Sie sie im rechten Winkel kreuzen.

Definitionen der Angaben zu Analogmodulen

- Genauigkeit: Abweichung von einem erwarteten Wert an einem bestimmten Ein- bzw. Ausgang.
- Auflösung: Auswirkungen der Änderung eines niederwertigsten Byte im Ausgang.

A.35 Erweiterungsmodul EM 235, Analogein-/Analogausgabe AE 3 / AA 1 x 12 Bit

Bestellnummer: 6ES7 235-0KD00-0XA0

Leistungsmerkmale		Eingänge	
Abmessungen (B x H x T)	90 x 80 x 62 mm	Eingangstyp	Differentialeingang
Gewicht	0,2 kg	Eingangsimpedanz	$\geq 10 \text{ M}\Omega$
Stromaufnahme	2 W	Eingabefilterdämpfung	-3db bei 3,1 kHz
Ein- und Ausgänge ¹	3 Analogeingänge 1 Analogausgang	Max. Eingangsspannung	30 V
Normen	UL 508 CSA C22.2 142 FM Klasse I, Kategorie 2 gemäß VDE 0160 gemäß EG-Richtlinie	Max. Eingangsstrom	32 mA
Ausgänge		Auflösung	A/D-Umsetzung, 12 Bit
Signalbereich		Elektrische Trennung	Nein
Ausgangsspannung	$\pm 10 \text{ V}$	Analog-Digital-Umsetzzeit	$< 250 \text{ }\mu\text{s}$
Ausgangsstrom	0 bis 20 mA	Analoge Sprungantwort	1,5 ms bis 95%
Auflösung, Vollbereich		Gleichtaktspannung	Signalspannung plus Gleichtaktspannung, kleiner gleich 12 V
Spannung	12 Bit	Gleichtaktunterdrückung	40 dB, DC bis 60 Hz
Strom	11 Bit	Datenwortformat ²	
Datenwortformat ²		Zweipoliger Bereich ³	-32000 bis +32000
Zweipoliger Bereich ³	-32000 bis +32000	Einpoliger Bereich ²	0 bis + 32000
Einpoliger Bereich ²	0 bis + 32000	Strombedarf	
Genauigkeit		5-V-DC-logische-Spannung	70 mA aus Zentralgerät
Ungünstigster Fall, 0 - 60° C		Externe Stromversorgung	60 mA zuzügl. 20-mA-Ausgangsstrom aus Zentralgerät bzw. externer Stromquelle (24-V-DC-Nennwert, Klasse 2 oder DC-Geberversorgung)
Ausgangsspannung	$\pm 2\%$ des Vollausschlags	Anzeige LED, EXTf	
Ausgangsstrom	$\pm 2\%$ des Vollausschlags	Netzausfall	Spannungsabfall, in externer 24-V-DC-Versorgung
Typ., 25° C			
Ausgangsspannung	$\pm 0,5\%$ des Vollausschlags		
Ausgangsstrom	$\pm 0,5\%$ des Vollausschlags		
Ausregelzeit			
Ausgangsspannung	100 μs		
Ausgangsstrom	2 ms		
Max. Ansteuerung bei 24-V-Versorgung			
Ausgangsspannung	min. 5000 Ω		
Ausgangsstrom	max. 500 Ω		

¹ In der CPU sind 4 Analogeingänge und 2 Analogausgänge für dieses Modul vorgesehen.

² Datenwort: 16 Zählimpulse pro Inkrement, linksbündige Werte (siehe Bilder A-43 und A-45).

³ Datenwort: 8 Zählimpulse pro Inkrement, linksbündige Werte (siehe Bild A-43).

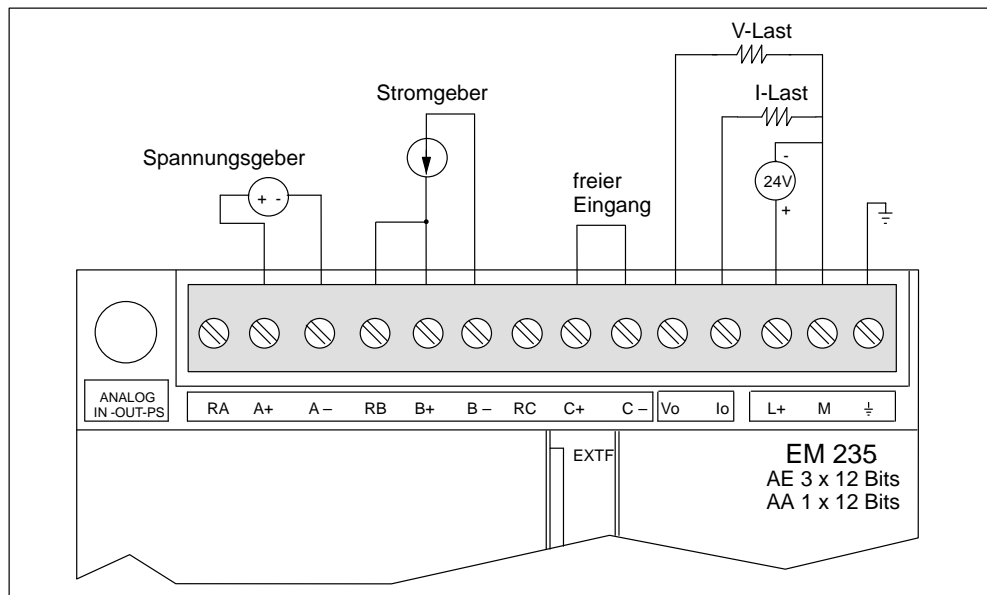


Bild A-41 Kennzeichnung der Anschlüsse beim Erweiterungsmodul EM 235, Analogein-/Analogausgabemodul AE 3/AA 1 x 12 Bit

Kalibrierung und Konfiguration

Das Kalibrierungspotentiometer und die DIP-Konfigurationsschalter sind durch die Lüftungsschlitze des Moduls zugänglich (siehe Bild A-42).

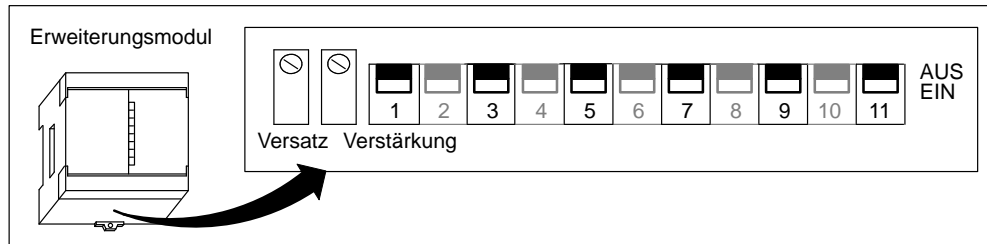


Bild A-42 Kalibrierungspotentiometer und DIP-Konfigurationsschalter

Konfiguration

Tabelle A-4 zeigt, wie Sie das Modul mit den DIP-Schaltern konfigurieren. Die Schalter 1, 3, 5, 7, 9, und 11 wählen den Bereich der Analogeingänge und das Datenformat aus. Alle Eingänge werden auf den gleichen Bereich der Analogeingänge und auf das gleiche Datenformat gesetzt.

Tabelle A-4 Konfigurationsschalter für EM 235, Analogein-/Analogausgabemodul

Konfigurationsschalter						Spannungsbereich	Auflösung
1 ¹	3	5	7	9	11		
EIN	EIN	AUS	EIN	AUS	AUS	0 bis 50 mV	12,5 µV
EIN	EIN	AUS	AUS	EIN	AUS	0 bis 100 mV	25 µV
EIN	AUS	EIN	EIN	AUS	AUS	0 bis 500 mV	125 µV
EIN	AUS	EIN	AUS	EIN	AUS	0 bis 1 V	250 µV
EIN	AUS	AUS	EIN	AUS	AUS	0 bis 5 V	1,25 mV
EIN	AUS	AUS	EIN	AUS	AUS	0 bis 20 mA ²	5 µA
EIN	AUS	AUS	AUS	EIN	AUS	0 bis 10 V	2,5 mV
AUS	EIN	AUS	EIN	AUS	AUS	±25 mV	12,5 µV
AUS	EIN	AUS	AUS	EIN	AUS	±50 mV	25 µV
AUS	EIN	AUS	AUS	AUS	EIN	±100 mV	50 µV
AUS	AUS	EIN	EIN	AUS	AUS	±250 mV	125 µV
AUS	AUS	EIN	AUS	EIN	AUS	±500 mV	250 µV
AUS	AUS	EIN	AUS	AUS	EIN	±1 V	500 µV
AUS	AUS	AUS	EIN	AUS	AUS	±2,5 V	1,25 mV
AUS	AUS	AUS	AUS	EIN	AUS	±5 V	2,5 mV
AUS	AUS	AUS	AUS	AUS	EIN	±10 V	5 mV

- ¹ Schalter 1 dient zum Angeben der Eingangspolarität: EIN für einpolig, AUS für zweipolig. CPU-Netzanschluß beim Umschalten zwischen ein- und zweipoligem Datenformat vorausgesetzt. Schalter 3, 5, 7, 9 und 11 dienen zum Auswählen des Spannungsbereichs.
- ² Messungen zwischen 0 und 20 mA wurden mit internem in Stromrichtung geschaltetem 250-Ohm-Widerstand durchgeführt.

Kalibrieren eines Eingangs

Das Kalibrieren wirkt sich auf alle drei Eingangskanäle aus. Es kann nach dem Kalibrieren zu unterschiedlichen Werten der Kanäle kommen.

Um das Modul genau zu kalibrieren, müssen Sie mit einem Programm arbeiten, das aus den aus dem Modul ausgelesenen Werten einen Mittelwert bildet. Arbeiten Sie mit dem Analog-eingabefilter-Assistenten in STEP 7-Micro/WIN, um dieses Programm zu erstellen (siehe Abschnitt 5.3). Bilden Sie den Mittelwert aus mindestens 64 abgefragten Werten.

Zum Kalibrieren eines Eingangs gehen Sie folgendermaßen vor:

1. Schalten Sie die Spannungsversorgung des Moduls aus. Wählen Sie den gewünschten Eingangsbereich.
2. Schalten Sie die Spannungsversorgung für die CPU und das Modul ein. Warten Sie ca. 15 Minuten, damit sich das Modul stabilisieren kann.
3. Legen Sie mittels eines Spannungs- oder Stromgebers an einem der Eingänge ein Nullsignal an.
4. Lesen Sie den Wert, der der CPU von dem entsprechenden Eingangskanal übermittelt wird.
5. Stellen Sie mit dem Versatzpotentiometer den Wert Null bzw. den gewünschten digitalen Datenwert ein.
6. Legen Sie an einem Eingang den Wert des Vollausschlags an. Lesen Sie den Wert aus, den die CPU empfangen hat.
7. Stellen Sie mit dem Verstärkungspotentiometer den Wert 32000 bzw. den gewünschten digitalen Datenwert ein.
8. Wiederholen Sie die Kalibrierung von Versatz und Verstärkung nach Bedarf.

Datenwortformat

Bild A-43 zeigt die Anordnung des 12-Bit-Datenwerts im Analogeingangswort der CPU.

Weicht die Wiederholgenauigkeit bei Vollausschlag um nur $\pm 0,50\%$ ab, kann dies eine Abweichung von ± 160 Zählimpulsen bei dem Wert hervorrufen, der aus dem Analogeingang ausgelesen wird.

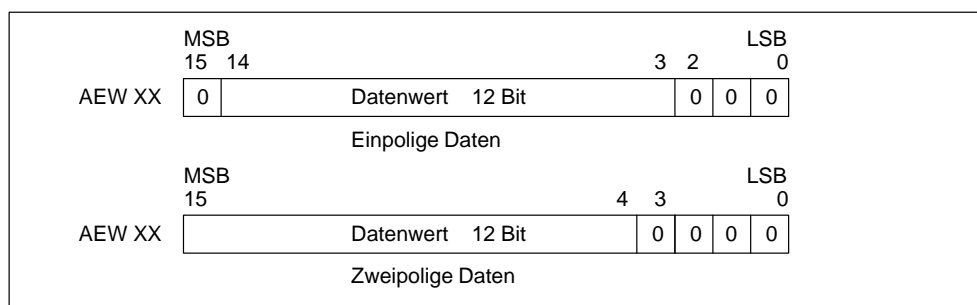


Bild A-43 Datenwortformat

Hinweis

Die 12 Bits eines Werts der Analog-Digital-Umsetzung sind im Datenwortformat linksbündig angeordnet. Das höchstwertige Bit gibt das Vorzeichen an: Null zeigt an, daß der Wert des Datenworts positiv ist. Im einpoligen Format bewirken die drei Nullen am Ende, daß sich das Datenwort bei jeder Änderung des Zählimpulses im Wert der Analog-Digital-Umsetzung um acht Zählimpulse ändert. Beim zweipoligen Format bewirken die vier Nullen am Ende, daß sich das Datenwort bei jeder Änderung des Zählimpulses im Wert der Analog-Digital-Umsetzung um sechzehn Zählimpulse ändert.

Schaltbild der Eingänge

Bild A-44 zeigt das Schaltbild der Eingänge beim EM 235.

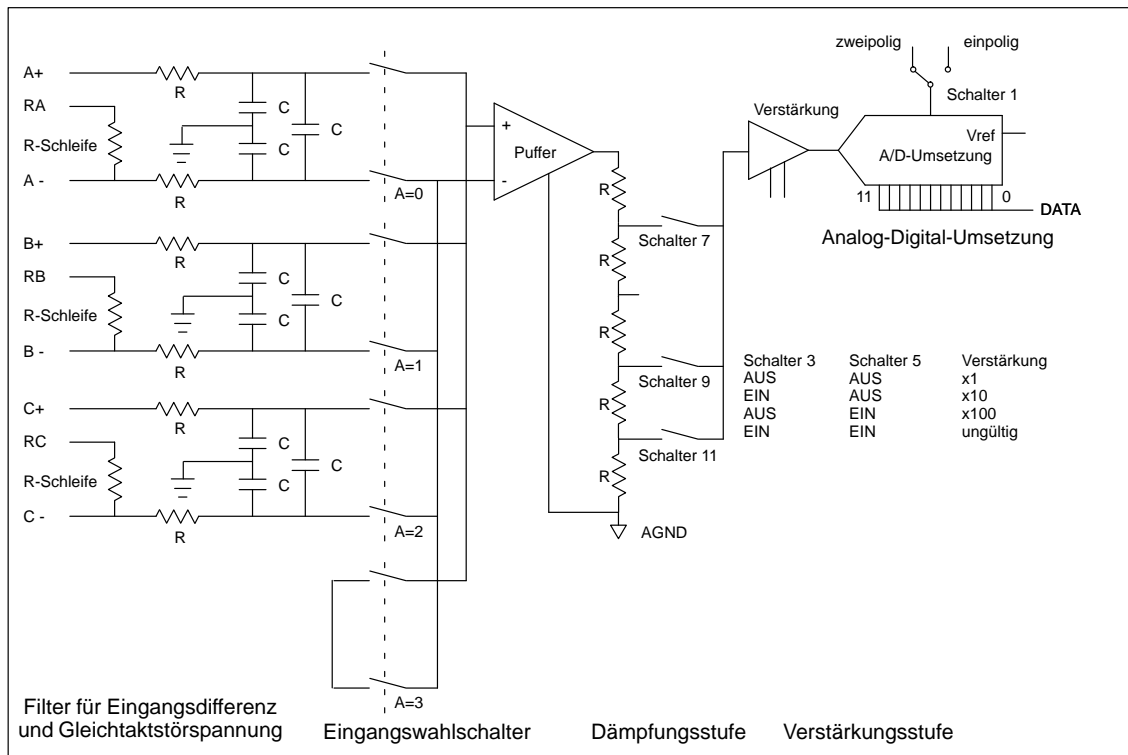


Bild A-44 Schaltbild der Eingänge beim EM 235

Datenwortformat

Bild A-45 zeigt die Anordnung des 12-Bit-Datenwerts im Analogausgangswort der CPU.
 Bild A-46 zeigt das Schaltbild der Ausgänge beim EM 235.

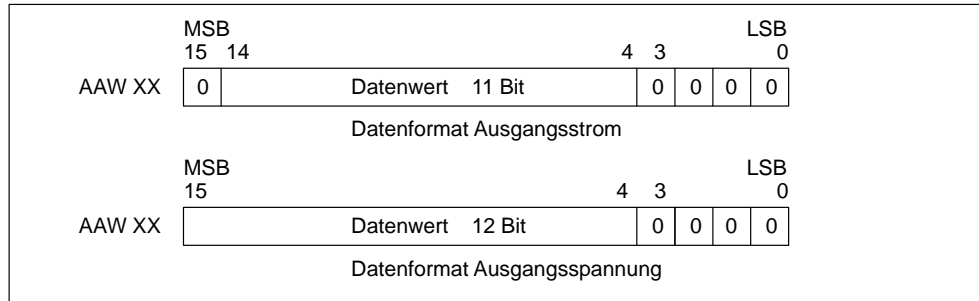


Bild A-45 Datenwortformat

Hinweis

Die 12 Bits eines Werts der Analog-Digital-Umsetzung sind im Datenwortformat linksbündig angeordnet. Das höchstwertige Bit gibt das Vorzeichen an: Null zeigt an, daß der Wert des Datenworts positiv ist. Die vier Nullen am Ende werden vor dem Laden in das Register für den Wert der Analog-Digital-Umsetzung gekürzt. Diese Bit wirken sich nicht auf den Wert des Ausgangssignals aus.

Schaltbild der Ausgänge

Bild A-46 zeigt das Schaltbild der Ausgänge beim EM 235.

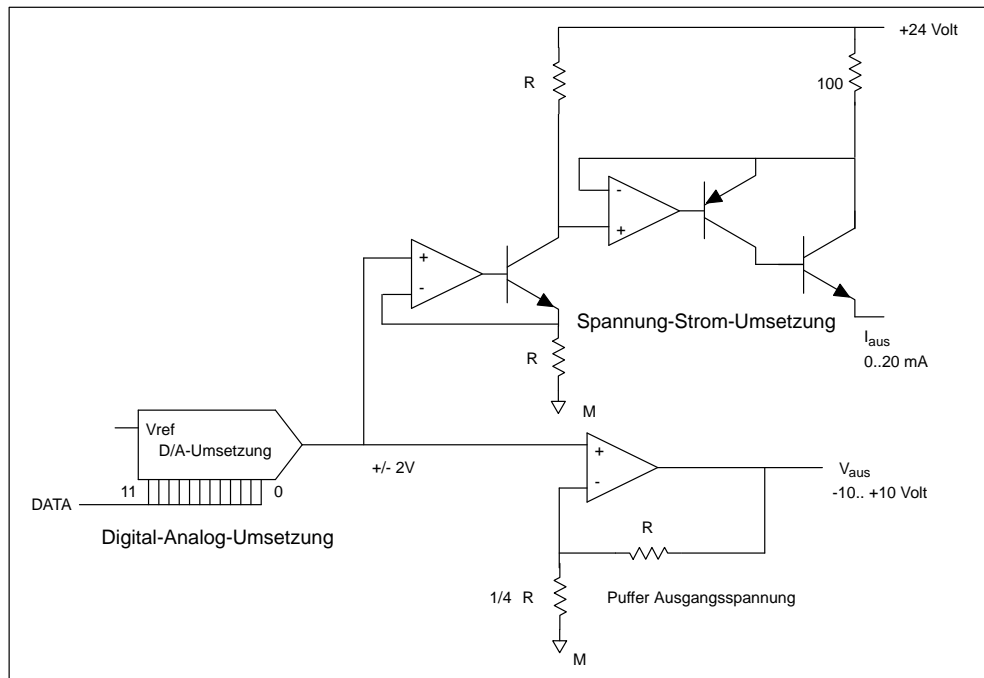


Bild A-46 Schaltbild der Ausgänge beim EM 235

Richtlinien für die Installation des EM 235

Beachten sie die folgenden Richtlinien, damit Genauigkeit und Wiederholbarkeit sichergestellt sind:

- Achten Sie darauf, daß die 24-V-DC-Geberversorgung störfest ist.
- Kalibrieren Sie das Modul.
- Verdrahten Sie die Geberversorgung so kurz wie möglich.
- Verwenden Sie für die Verdrahtung der Geberversorgung geschirmte verdrehte Doppelleitungen.
- Schließen Sie nur die Schirmung an der Geberversorgungsseite ab.
- Überbrücken Sie die Eingänge für freie Kanäle (siehe Bild A-41).
- Vermeiden Sie es, die Leitungen scharf zu knicken.
- Verlegen Sie die Leitungen in Kabelkanälen.
- Verlegen Sie Signalleitungen nicht parallel zu Starkstromleitungen. Kreuzen sich die beiden Leitungen, lassen Sie sie im rechten Winkel kreuzen.
- Achten Sie darauf, daß die Eingangssignale potentialfrei sind oder als Bezugsleiter den externen 24-V-Leiter des Analogmoduls haben.

Hinweis

Dieses Erweiterungsmodul dürfen Sie nicht mit Thermoelementen einsetzen.

Arbeiten mit Analogeingängen: Genauigkeit und Wiederholbarkeit

Das Analogeingabemodul EM 235 ist ein preiswertes, Hochgeschwindigkeits-Analogein-/Analogausgabemodul (12 Bit). Das Modul kann eine Analogeingabe in den entsprechenden Digitalwert in 171 μs bei der CPU 212 und in 139 μs bei allen anderen S7-200 CPUs umwandeln. Die Umwandlung des Analogeingabesignals wird jedesmal durchgeführt, wenn Ihr Programm auf den Analogeingang zugreift. Diese Zeitangaben müssen zu der grundlegenden Ausführungszeit der Operation, die auf den Analogeingang zugreift, addiert werden.

Das Erweiterungsmodul EM 235 liefert einen nicht bearbeiteten Digitalwert (keine Linearisierung oder Filterung), der der Analogspannung bzw. dem Analogstrom an den Eingangsklemmen des Moduls entspricht. Da es sich bei dem Modul um ein Hochgeschwindigkeitsmodul handelt, kann es schnellen Änderungen im Analogeingabesignal folgen (einschließlich internem und externem Rauschen). Abweichungen von einem Wert zum nächsten, die durch Störungen in einer Konstante oder in einem sich langsam ändernden Analogeingabesignal verursacht werden, können durch Mittelwertbildung aus einer Reihe von Werten verringert werden. Je weiter Sie die Anzahl der für die Mittelwertbildung herangezogenen Werte erhöhen, desto stärker können Sie eine entsprechend langsamere Reaktionszeit auf Änderungen im Eingabesignal feststellen.

Sie können mit dem Analogeingabefilter-Assistenten in STEP 7-Micro/WIN arbeiten, um ein Programm zur Mittelwertbildung in Ihr Programm einzufügen. Beachten Sie, daß ein Mittelwert, der aus einer hohen Zahl abgetasteter Werte gebildet wird, den Wert stabilisiert und gleichzeitig die Reaktion des Werts auf Änderungen im Eingabesignal verlangsamt. Sollen die Analogeingabesignale nur langsam geändert werden, empfehlen wir eine Mindestzahl von 64 abgefragten Werten für das Programm zur Mittelwertbildung.

Die Angaben zur Wiederholbarkeit beschreiben die Abweichungen zwischen den Werten bei gleichbleibendem Eingangssignal. Außerdem definieren die Angaben zur Wiederholbarkeit den Bereich, der 99% aller Werte enthält. Die mittlere Genauigkeit beschreibt den durchschnittlichen Wert des Fehlers (die Differenz zwischen dem Durchschnittswert der einzelnen Werte und dem genauen Wert des tatsächlichen Analogeingabesignals). Die Wiederholbarkeit wird in Bild A-47 durch die Kurve dargestellt. Dieses Bild zeigt die Grenzwerte für die Wiederholbarkeit von 99%, den mittleren bzw. durchschnittlichen Wert aus den einzelnen Werten und die mittlere Genauigkeit in graphischer Form. Tabelle A-5 enthält die Angaben zur Wiederholbarkeit und zur mittleren Genauigkeit in bezug auf die konfigurierbaren Bereiche.

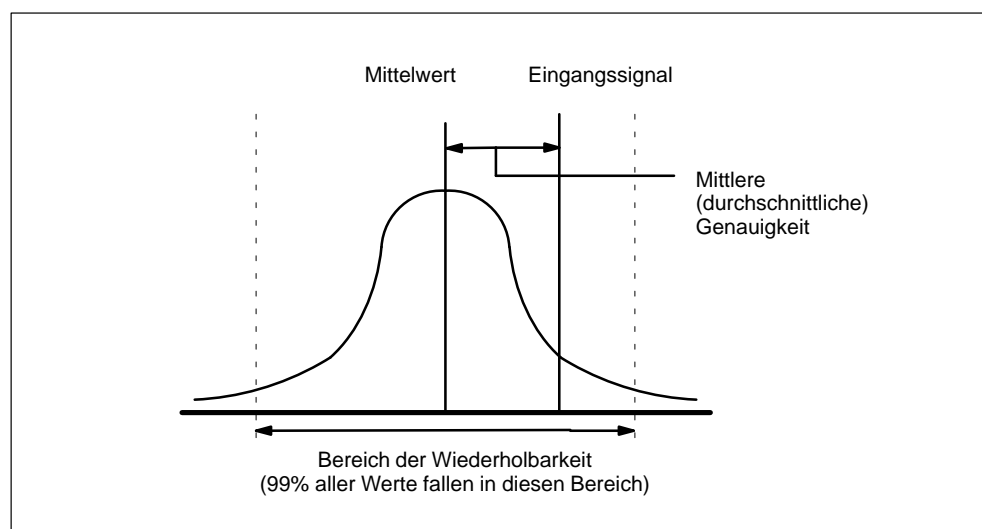


Bild A-47 Definition der Genauigkeit

Tabelle A-5 Angaben für S7-200 CPUs (AC- und DC-Varianten)

Eingangsbereich bei Vollausschlag	Wiederholbarkeit ¹		Mittlere (durchschnittliche) Genauigkeit ^{1, 2, 3, 4}	
	% des Vollausschlags	Zählimpulse	% des Vollausschlags	Zählimpulse
Angaben für DC-Varianten der S7-200 CPUs				
0 bis 50 mV	± 0,075%	± 24	± 0,25%	± 80
0 bis 100 mV			± 0,2%	± 64
0 bis 500 mV			± 0,05%	± 16
0 bis 1 V				
0 bis 5 V				
0 bis 20 mA				
0 bis 10 V				
± 25 mV	± 0,075%	± 48	± 0,25%	± 160
± 50 mV			± 0,2%	± 128
± 100 mV			± 0,1%	± 64
± 250 mV			± 0,05%	± 32
± 500 mV				
± 1 V				
± 2,5 V				
± 5 V				
± 10 V				
Angaben für AC-Varianten der S7-200 CPUs				
0 bis 50 mV	± 0,15%	± 48	± 0,25%	± 80
0 bis 100 mV			± 0,2%	± 64
0 bis 500 mV			± 0,05%	± 16
0 bis 1 V				
0 bis 5 V				
0 bis 20 mA				
0 bis 10 V				
± 25 mV	± 0,15%	± 96	± 0,25%	± 160
± 50 mV			± 0,2%	± 128
± 100 mV			± 0,1%	± 64
± 250 mV			± 0,05%	± 32
± 500 mV				
± 1 V				
± 2,5 V				
± 5 V				
± 10 V				

¹ Messungen nach durchgeführter Kalibrierung des Eingangsbereichs.

² Versatzfehler bei Signal nahe 0 des Analogeingangs werden weder korrigiert noch in den Angaben zur Genauigkeit berücksichtigt.

³ Beim Übertragen von Kanal zu Kanal tritt aufgrund der endlichen Ausregelzeit des Analog-Multiplexers ein Umwandlungsfehler auf. Maximaler Übertragungsfehler ist 0,1% des Unterschieds zwischen den Kanälen.

⁴ Die mittlere Genauigkeit umfaßt Auswirkungen von Nicht-Linearität und Drift zwischen 0 und 55 Grad C.

A.36 Speichermodul 8 K x 8

Bestellnummer: **6ES7 291-8GC00-0XA0**

Leistungsmerkmale	
Abmessungen (B x H x T)	28 x 10 x 16 mm
Gewicht	3,6 g
Stromaufnahme	0,5 mW
Speichertyp	EEPROM
Anwenderspeicher	4096 Byte Anwenderprogramm + 1024 Byte Anwenderdaten + interne Systemdaten
Normen	UL 508 CSA C22.2 142 FM Klasse I, Kategorie 2 gemäß VDE 0160 gemäß EG-Richtlinie

Hinweis

Das Speichermodul 8 K wird als 4polige und 5polige Version produziert. Die beiden Versionen sind vollständig kompatibel.

Sie können das Speichermodul für alle S7-200 CPUs verwenden, doch das Speichermodul 8 K kann die maximale Größe des Programms, das in den CPUs 215 und 216 möglich ist, nicht speichern. Damit durch die Programmgröße keine Probleme auftreten, wird das Speichermodul 8 K nur zur Verwendung für die CPU 214 oder die PDS 210 empfohlen.

Sie können mit Speichermodulen Programme nur zwischen CPUs gleicher Art übertragen. (Beispielsweise können Sie ein mit einer CPU 214 programmiertes Speichermodul nur für andere CPUs 214 einsetzen.)

Abmessungen des Speichermoduls

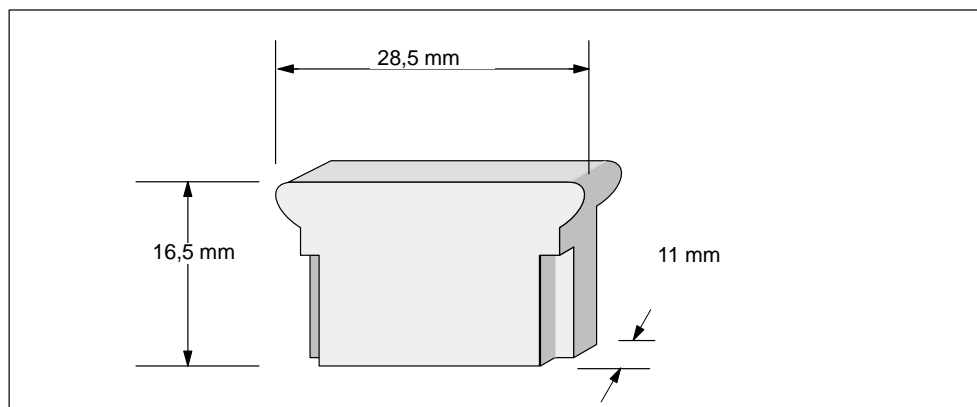


Bild A-48 Abmessungen des Speichermoduls - 8 K x 8

A.37 Speichermodul 16 K x 8

Bestellnummer: **6ES7 291-8GD00-0XA0**

Leistungsmerkmale	
Abmessungen (B x H x T)	28 x 10 x 16 mm
Gewicht	3,6 g
Stromaufnahme	0,5 mW
Speichertyp	EEPROM
Anwenderspeicher	8192 Byte Anwenderprogramm + 5120 Byte Anwenderdaten + interne Systemdaten
Normen	UL 508 CSA C22.2 142 FM Klasse I, Kategorie 2 gemäß VDE 0160 gemäß EG-Richtlinie

Hinweis

Das Speichermodul 16 K können Sie für die PDS 210, die CPUs 214, 215 und 216 verwenden.

Sie können mit Speichermodulen Programme nur zwischen CPUs gleicher Art übertragen. (Beispielsweise können Sie ein mit einer CPU 214 programmiertes Speichermodul nur für andere CPUs 214 einsetzen.)

Abmessungen des Speichermoduls

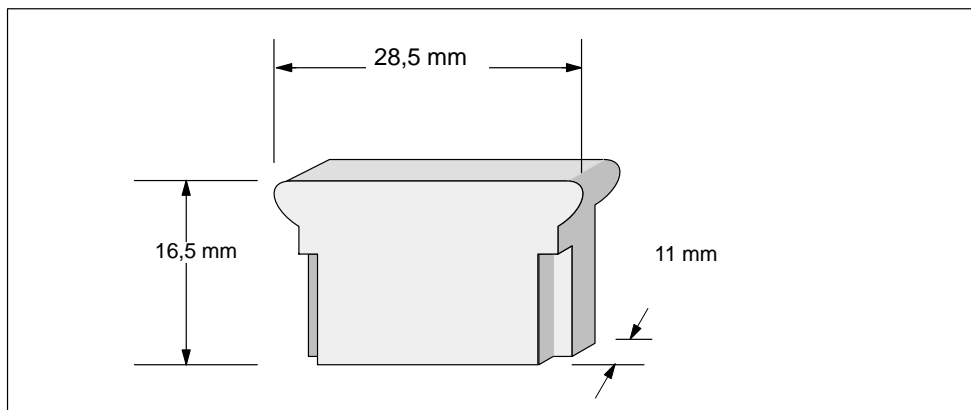


Bild A-49 Abmessungen des Speichermoduls - 16 K x 8

A.38 Batteriemodul

Bestellnummer: **6ES7 291-8BA00-0XA0**

Leistungsmerkmale	
Abmessungen (B x H x T)	28 x 10 x 16 mm
Gewicht	3,6 g
Batterie	
Größe (Durchmesser x Höhe)	9,9 x 2,5 mm
Typ.	Lithium (< 0,6 g)
Haltbarkeit	10 Jahre
Typ. Lebensdauer	200 Tage bei ständigem Einsatz*
Austausch	3V 30 mA/h (Renata CR 1025) empfohlen nach 1 Jahr
Normen	UL 508 CSA C22.2 142 FM Klasse I, Kategorie 2 gemäß VDE 0160 gemäß EG-Richtlinie
*Die Batterie wird erst aktiviert, nachdem sich der Hochleistungskondensator entladen hat. Stromausfälle, die die Zeit nicht überschreiten, die der Hochleistungskondensator die Daten puffert, verkürzen die Lebensdauer der Batterie nicht.	

Abmessungen des Batteriemoduls

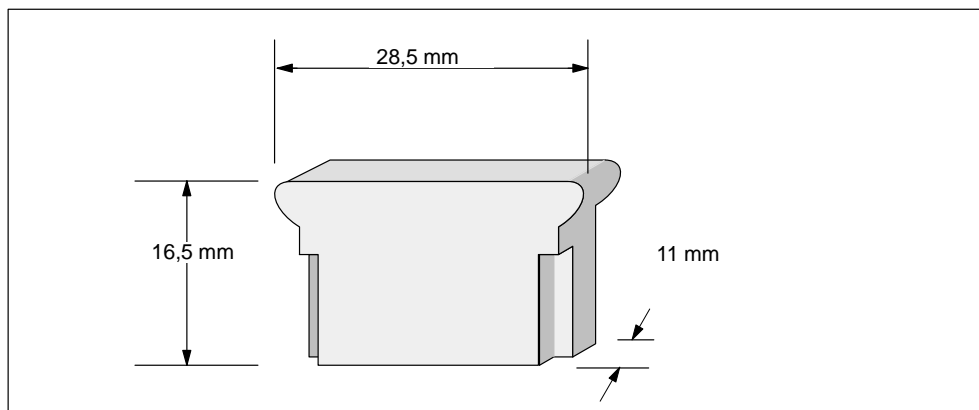


Bild A-50 Abmessungen des Batteriemoduls

A.39 Steckleitung für Erweiterungsmodule

Bestellnummer: **6ES7 290-6BC50-0XA0**

Leistungsmerkmale	
Kabellänge	0,8 m
Gewicht	0,2 kg
Stecker	Kartenstecker

Typische Installation der Steckleitung für Erweiterungsmodule

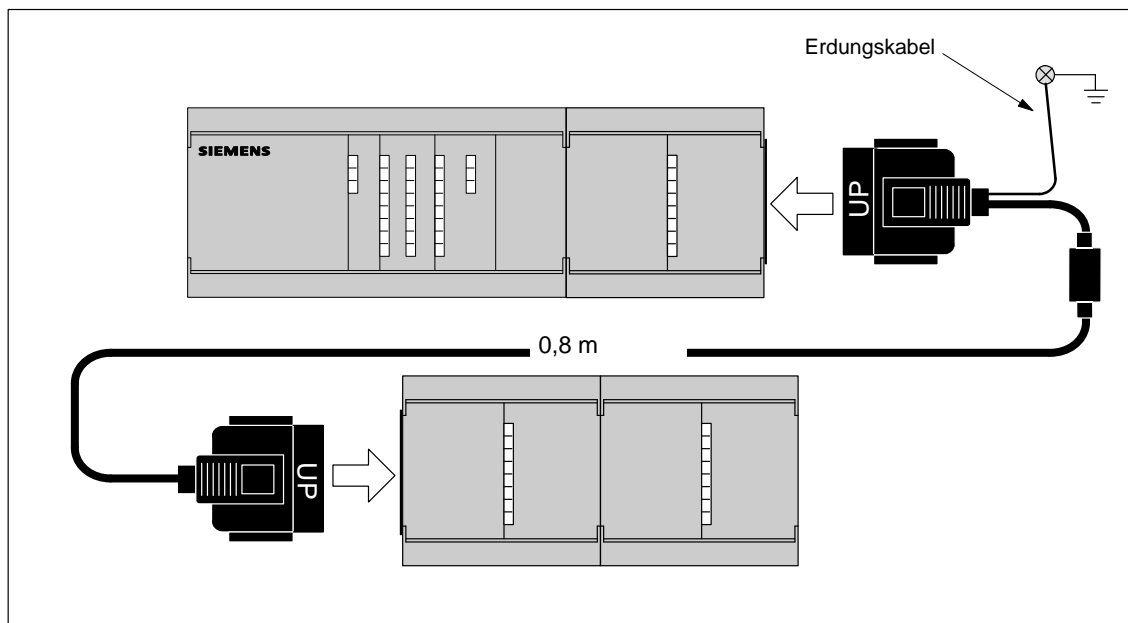


Bild A-51 Typische Installation der Steckleitung für Erweiterungsmodule



Vorsicht

Wenn Sie die Steckleitung für Erweiterungsmodule nicht sachgemäß installieren, kann Sachschaden an Ihren Geräten entstehen.

Bei unsachgemäßer Installation der Steckleitung kann der durch die Steckleitung fließende Strom das Erweiterungsmodul beschädigen.

Richten Sie die Steckleitung so aus, daß die Oberseite des Kabels beim Anschließen an das Erweiterungsmodul nach vorne zeigt (siehe Bild A-51).

A.40 PC/PPI-Kabel

Bestellnummer: **6ES7 901-3BF00-0XA0**

Leistungsmerkmale															
Kabellänge	5 m														
Gewicht	0,3 kg														
Stromaufnahme	0,5 W														
Anschlußstecker PC AS	9-poliger Sub-D-Stecker (Sockel) 9-poliger Sub-D-Stecker (Pole)														
Kabelart	RS-232 in RS-485, nicht elektrisch getrennt														
Umschaltzeit Empfangen/Senden	Dauer 2 Zeichen														
Unterstützte Baudrate (durch DIP-Schalter auszuwählen)	<table border="0"> <tr> <td></td> <td><u>Schalter</u></td> </tr> <tr> <td>38,4 k</td> <td>0000</td> </tr> <tr> <td>19,2 k</td> <td>0010</td> </tr> <tr> <td>9,6 k</td> <td>0100</td> </tr> <tr> <td>2,4 k</td> <td>1000</td> </tr> <tr> <td>1,2 k</td> <td>1010</td> </tr> <tr> <td>600</td> <td>1100</td> </tr> </table>		<u>Schalter</u>	38,4 k	0000	19,2 k	0010	9,6 k	0100	2,4 k	1000	1,2 k	1010	600	1100
	<u>Schalter</u>														
38,4 k	0000														
19,2 k	0010														
9,6 k	0100														
2,4 k	1000														
1,2 k	1010														
600	1100														
Normen	UL 508 CSA C22.2 142 FM Klasse I, Kategorie 2 gemäß VDE 0160 gemäß EG-Richtlinie														

Tabelle A-6 Anschlußbelegung der Pole des Kabels

RS-232 Pol	Funktion beim PC	RS-485 Pol	Funktion bei der S7-200 CPU
2	Empfangsdaten (PC empfängt)	8	Signal A
3	Sendedaten (PC sendet)	3	Signal B
5	Signalmasse	7	+24 V
		2	+24-V-Rückleiter (log. Masse AS)
		1	Schirm (logische Masse AS)

**Vorsicht**

Wenn Sie Geräte mit unterschiedlichen Bezugsspannungen miteinander verbinden, kann dies zu unerwünschten Strömen im Verbindungskabel führen.

Diese unerwünschten Ströme können Kommunikationsfehler verursachen oder Schäden in den Geräten hervorrufen.

Achten Sie deshalb darauf, daß alle Geräte, die Sie über ein Kommunikationskabel miteinander verbinden, entweder einen gemeinsamen Bezugsstromkreis haben oder elektrisch voneinander getrennt sind, um unerwünschte Ströme zu vermeiden. Weitere Hinweise hierzu entnehmen Sie den Richtlinien für Erdung und Bezugsspannung galvanisch getrennter Stromkreise in Abschnitt 2.3.

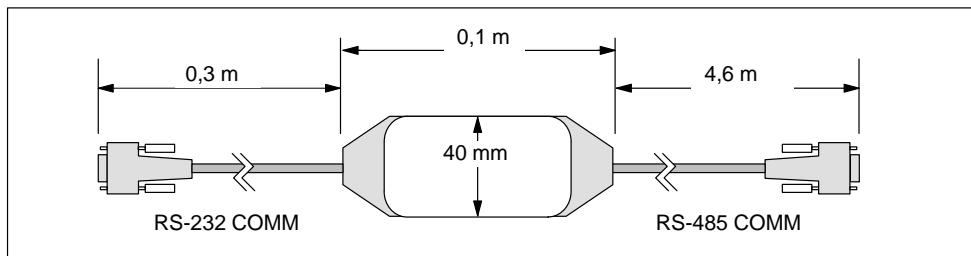
Abmessungen des PC/PPI-Kabels

Bild A-52 Abmessungen des PC/PPI-Kabels

A.41 DC-Eingangssimulator für die CPU 212

Bestellnummer: **6ES7 274-1XF00-0XA0**

Leistungsmerkmale	
Abmessungen (B x H x T)	61 x 36 x 22 mm
Gewicht	0,02 kg
Eingänge	8

Installation

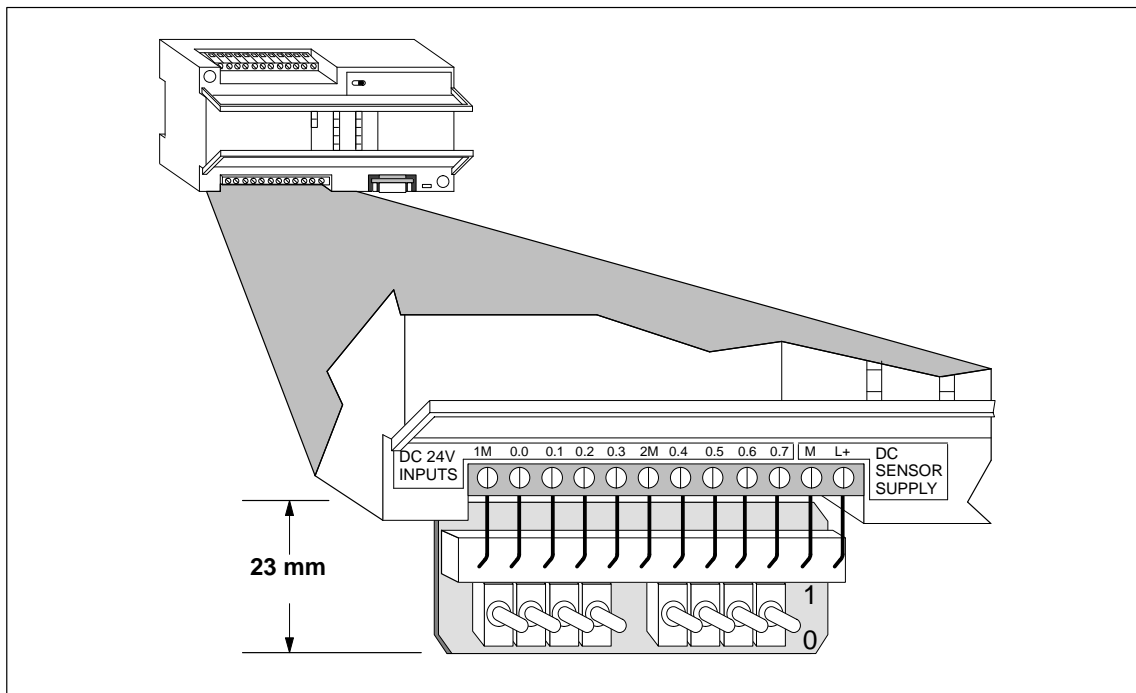


Bild A-53 Installation des DC-Eingangssimulators für die CPU 212

A.42 DC-Eingangssimulator für die CPU 214

Bestellnummer: **6ES7 274-1XH00-0XA0**

Leistungsmerkmale	
Abmessungen (B x H x T)	91 x 36 x 22 mm
Gewicht	0,03 kg
Eingänge	14

Installation

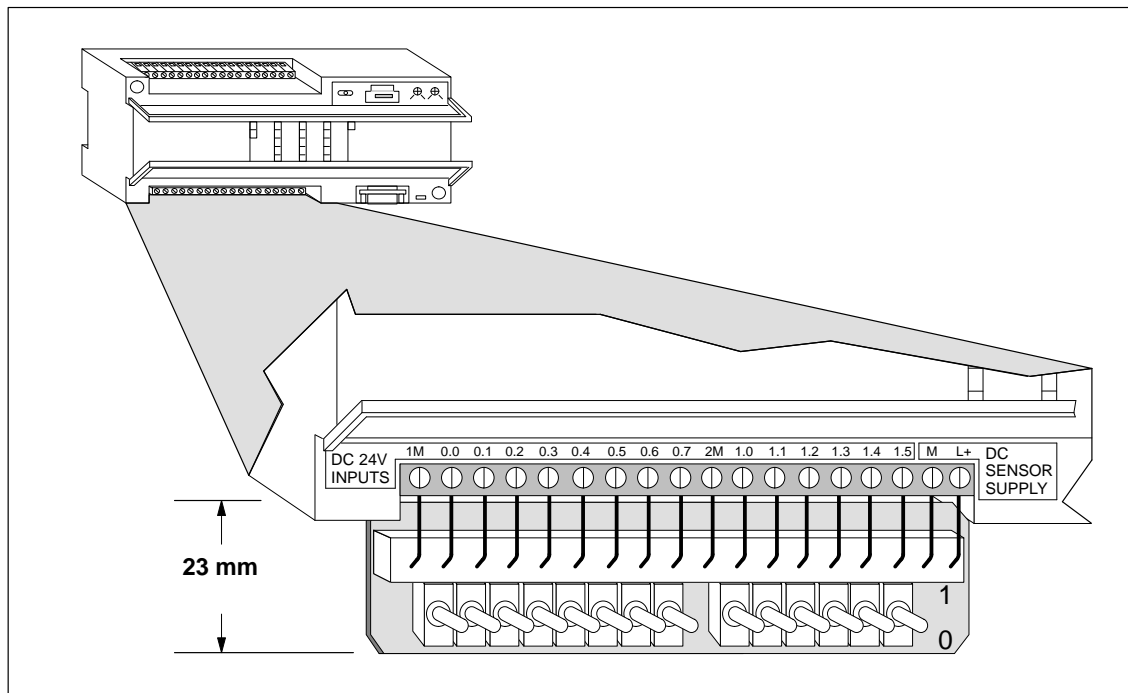


Bild A-54 Installation des DC-Eingangssimulators für die CPU 214

A.43 DC-Eingangssimulator für die CPU 215/216

Bestellnummer: **6ES7 274-1XK00-0XA0**

Leistungsmerkmale	
Abmessungen (B x H x T)	147 x 36 x 25 mm
Gewicht	0,04 kg
Eingänge	24

Installation

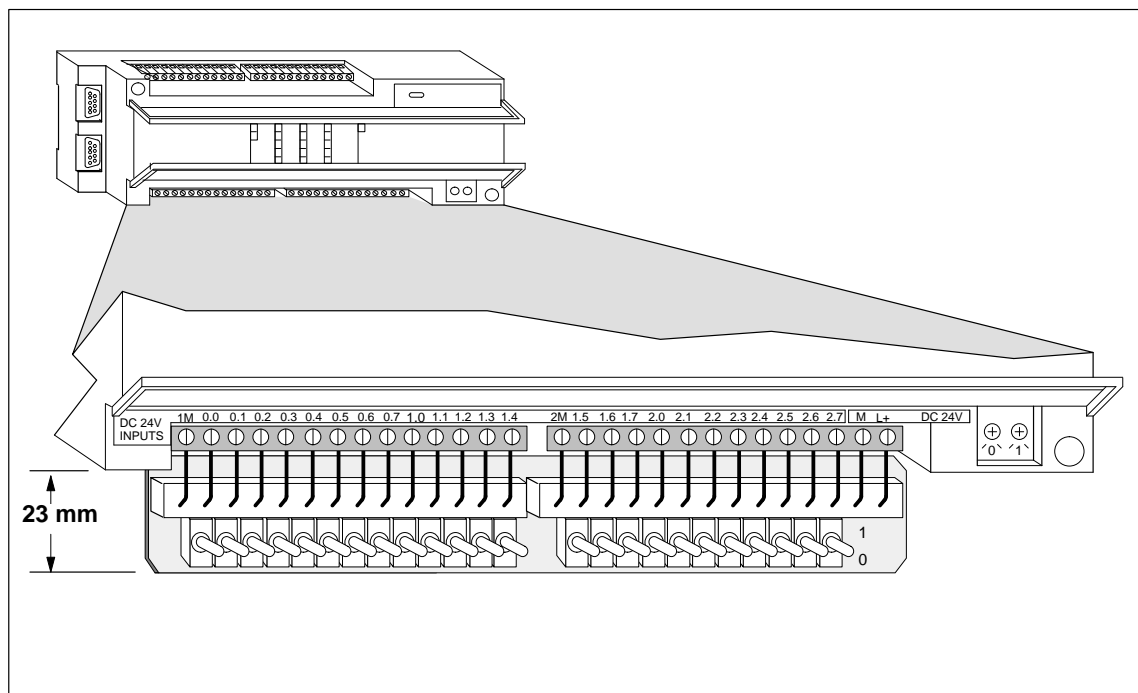


Bild A-55 Installation des DC-Eingangssimulators für die CPU 215/216

B

Berechnungstabelle Strombilanz

Jede S7-200 CPU (Zentralgerät) liefert Gleichstrom von 5 V und 24 V für die Erweiterungsmodule.

- Mit dem 5-V-Gleichstrom werden automatisch die Erweiterungsmodule über den Erweiterungsbus versorgt.
- Jede CPU verfügt über eine 24-V-DC-Geberversorgung, die die Eingänge sowie die Relaisspulen der Erweiterungsmodule versorgt. Sie müssen die 24-V-DC-Versorgung manuell an die Eingänge und die Relaisspulen anschließen.

Mit Hilfe der folgenden Tabelle können Sie berechnen, wieviel Leistung bzw. Strom die CPU für Ihre Konfiguration liefern kann. Die Datenblätter in Anhang A bieten Informationen zu der verfügbaren Leistung der einzelnen CPUs und zum Leistungsbedarf der Erweiterungsmodule. Abschnitt 2.5 zeigt Ihnen ein Beispiel für die Berechnung einer Strombilanz.

Leistung der CPU	5 V DC	24 V DC

minus

Strombedarf des Systems	5 V DC	24 V DC
	Zentralgerät	
Gesamtbedarf		

gleich

Strombilanz	5 V DC	24 V DC
Gesamt-Strombilanz		

Fehlermeldungen

C

Die folgenden Informationen zu möglichen Fehlern sollen Ihnen bei der Fehlerbehebung mit Ihrer S7-200 CPU behilflich sein.

Kapitelübersicht

Abschnitt	Beschreibung	Seite
C.1	Fehlercodes und Meldungen von schweren Fehlern	C-2
C.2	Fehler zur Laufzeit	C-3
C.3	Verletzungen der Übersetzungsregeln	C-4

C.1 Fehlercodes und Meldungen von schweren Fehlern

Tritt ein schwerer Fehler auf, beendet Ihre CPU die Bearbeitung des Programms. Je nach dem Schweregrad des Fehlers kann die CPU einige oder auch gar keine Funktionen mehr ausführen. Ziel der Behebung von schweren Fehlern ist es, die CPU in einen sicheren Zustand zu bringen, so daß Informationen zu der Fehlerbedingung in der CPU abgefragt werden können.

Die CPU führt bei Auftreten eines schweren Fehlers die folgenden Funktionen aus:

- Die CPU geht in den Betriebszustand STOP.
- Die LED-Anzeigen für Systemfehler (SF) und für den Betriebszustand STOP an der CPU leuchten auf.
- Die Ausgänge werden ausgeschaltet.

Die CPU bleibt in diesem Zustand, bis der schwere Fehler behoben ist. Tabelle C-1 führt für alle Fehlercodes, die die CPU zu schweren Fehlern ausgibt, eine Beschreibung auf.

Tabelle C-1 Fehlercodes und Meldungen von schweren Fehlern in der CPU

Fehlercode	Beschreibung
0000	Kein schwerer Fehler aufgetreten.
0001	Prüfsummenfehler im Anwenderprogramm.
0002	Prüfsummenfehler im übersetzten KOP-Programm.
0003	Zeitfehler in Zykluszeitüberwachung.
0004	Interner EEPROM-Fehler.
0005	Interner EEPROM-Prüfsummenfehler im Anwenderprogramm.
0006	Interner EEPROM-Prüfsummenfehler in Konfigurationsparametern.
0007	Interner EEPROM-Prüfsummenfehler in geforcten Daten.
0008	Interner EEPROM-Prüfsummenfehler in den Defaultwerten der Tabelle der Ausgänge.
0009	Interner EEPROM-Prüfsummenfehler in Anwenderdaten, DB1.
000A	Speichermodul-Fehler.
000B	Speichermodul-Prüfsummenfehler im Anwenderprogramm.
000C	Speichermodul-Prüfsummenfehler in Konfigurationsparametern.
000D	Speichermodul-Prüfsummenfehler in geforcten Daten.
000E	Speichermodul-Prüfsummenfehler in den Defaultwerten der Tabelle der Ausgänge.
000F	Speichermodul-Prüfsummenfehler in Anwenderdaten, DB1.
0010	Interner Software-Fehler.
0011	Fehler bei indirekter Adressierung des Vergleichskontakts.
0012	Ungültiger Wert bei Vergleichskontakt.
0013	Speichermodul ist leer oder das Programm wird von dieser CPU nicht unterstützt.

C.2 Fehler zur Laufzeit

Während der Bearbeitung Ihres Programms können leichte Fehlerbedingungen auftreten (z. B. Adressierungsfehler). In diesem Fall gibt die CPU einen Fehlercode für einen Laufzeitfehler aus. Tabelle C-2 listet die Beschreibungen zu den Fehlercodes der leichten Fehler auf.

Tabelle C-2 Fehler zur Laufzeit

Fehlercode	Fehler zur Laufzeit (keine schweren Fehler)
0000	Kein Fehler aufgetreten.
0001	Box HSC vor Bearbeitung der Box HDEF freigegeben.
0002	Eingangsinterrupt wurde einem Eingang zugewiesen, der bereits einem HSC zugeordnet ist.
0003	Eingänge, die bereits einem Eingangsinterrupt zugeordnet sind, wurden einem HSC zugewiesen.
0004	Versuch, eine der Operationen ENI, DISI oder HDEF im Interruptprogramm auszuführen.
0005	Versuch, vor Beenden des ersten HSC einen zweiten HSC mit der gleichen Nummer auszuführen (HSC im Interruptprogramm in Konflikt mit HSC im Hauptprogramm).
0006	Fehler bei indirekter Adressierung.
0007	Fehler in Daten für Operation TODW (Echtzeituhr schreiben).
0008	Maximale Schachtelungstiefe für Unterprogramme überschritten.
0009	Bearbeitung einer Operation XMT oder RCV, während eine andere Operation XMT oder RCV gerade bearbeitet wird.
000A	Versuch, HSC durch zweite Operation HDEF für denselben HSC neu zu definieren.
0091	Bereichsfehler (mit Adreßinformationen): Prüfen Sie die Operandenbereiche.
0092	Fehler im Zählfeld einer Operation (mit Zählinformationen): Prüfen Sie den maximalen Zählwert.
0094	Bereichsfehler beim Schreiben in nullspannungsfesten Speicher (mit Adreßinformationen).

C.3 Verletzungen der Übersetzungsregeln

Wenn Sie ein Programm in die CPU laden, übersetzt die CPU das Programm. Erkennt die CPU, daß das Programm eine Übersetzungsregel verletzt (z. B. eine Operation ungültig ist), dann bricht die CPU den Ladevorgang ab und gibt einen Übersetzungsfehler aus. Tabelle C-3 beschreibt die Fehlercodes bei Verletzungen der Übersetzungsregeln.

Tabelle C-3 Verletzungen der Übersetzungsregeln

Fehlercode	Übersetzungsfehler (keine schweren Fehler)
0080	Programm zum Übersetzen zu groß: Kürzen Sie das Programm.
0081	Unterschreitung des Stack: Teilen Sie das Netzwerk in mehrere Netzwerke.
0082	Unzulässige Operation: Prüfen Sie die Mnemonik.
0083	MEND fehlt bzw. ungültige Operation im Hauptprogramm: Geben Sie die Operation MEND ein bzw. löschen Sie die ungültige Operation.
0084	Reserviert.
0085	FOR fehlt: Ergänzen Sie die Operation FOR oder löschen Sie die Operation NEXT.
0086	NEXT fehlt: Ergänzen Sie die Operation NEXT oder löschen Sie die Operation FOR.
0087	Sprungmarke fehlt (LBL, INT, SBR): Ergänzen Sie die entsprechende Sprungmarke.
0088	RET fehlt bzw. ungültige Operation im Unterprogramm: Geben Sie die Operation RET am Ende des Unterprogramms ein bzw. löschen Sie die ungültige Operation.
0089	RETI fehlt bzw. ungültige Operation im Interruptprogramm: Geben Sie die Operation RETI am Ende des Interruptprogramms ein bzw. löschen Sie die ungültige Operation.
008A	Reserviert.
008B	Reserviert.
008C	Doppelte Sprungmarke (LBL, INT, SBR): Benennen Sie eine der Sprungmarken um.
008D	Unzulässige Sprungmarke (LBL, INT, SBR): Stellen Sie sicher, daß die zulässige Anzahl an Sprungmarken nicht überschritten wurde.
0090	Unzulässiger Parameter: Prüfen Sie, ob die Parameter für die Operation zulässig sind.
0091	Bereichsfehler (mit Adreßinformationen): Prüfen Sie die Operandenbereiche.
0092	Fehler im Zählfeld einer Operation (mit Zählinformationen): Prüfen Sie den maximalen Zählwert.
0093	Die Schachtelungstiefe FOR/NEXT wurde überschritten.
0095	Operation LSCR (Ablaufsteuerungsrelais laden) fehlt.
0096	Operation SCRE (Ablaufsteuerungsrelais beenden) fehlt bzw. ungültige Operation vor SCRE.

Sondermerker

D

Sondermerker bieten verschiedene Status- und Steuerungsfunktionen und dienen dazu, Informationen zwischen dem Automatisierungssystem und Ihrem Programm auszutauschen. Sondermerker können als Bits, Bytes, Wörter und Doppelwörter verwendet werden.

SMB0: Statusbits

Wie in Tabelle D-1 beschrieben, enthält SMB0 acht Statusbits, die von der S7-200 CPU am Ende eines jeden Zyklus aktualisiert werden.

Tabelle D-1 Sondermerker SMB0 (SM0.0 bis SM0.7)

Sondermerker	Beschreibung
SM0.0	Dieses Bit ist immer eingeschaltet.
SM0.1	Dieses Bit ist im ersten Zyklus eingeschaltet. Es wird z. B. zum Aufrufen eines Initialisierungs-Unterprogramms verwendet.
SM0.2	Dieses Bit wird für die Dauer eines Zyklus eingeschaltet, wenn remanente Daten verloren gegangen sind. Es kann entweder als Fehlermerker oder als Mechanismus zum Aufrufen von besonderen Anlaufsequenzen verwendet werden.
SM0.3	Dieses Bit wird für die Dauer eines Zyklus eingeschaltet, wenn die Betriebsart RUN von einer Einschaltbedingung aus eingestellt wird. Damit kann vor dem Betrieb für eine Aufwärmzeit der Anlage gesorgt werden.
SM0.4	Dieses Bit sorgt für einen Takt, der 30 Sekunden eingeschaltet und 30 Sekunden ausgeschaltet ist, und zwar für eine Zykluszeit von 1 Minute. Damit verfügen Sie über eine leicht zu programmierende Verzögerungszeit bzw. eine Taktzeit von 1 Minute.
SM0.5	Dieses Bit sorgt für einen Takt, der 0,5 Sekunden eingeschaltet und 0,5 Sekunden ausgeschaltet ist, und zwar für eine Zykluszeit von 1 Sekunde. Damit verfügen Sie über eine leicht zu programmierende Verzögerungszeit bzw. eine Taktzeit von 1 Sekunde.
SM0.6	Dieses Bit stellt einen Zyklustakt dar. Es ist einen Zyklus eingeschaltet, den nächsten Zyklus ausgeschaltet. Sie können dieses Bit als Zykluszähleingang verwenden.
SM0.7	Dieses Bit zeigt die Position des Betriebsartenschalters an (TERM - Aus, RUN - Ein). Wird mit diesem Bit die frei programmierbare Kommunikation freigegeben, wenn der Schalter auf RUN steht, dann können Sie die normale Kommunikation mit dem Programmiergerät freigegeben, indem Sie den Schalter auf TERM stellen.

SMB1: Statusbits

Wie in Tabelle D-2 beschrieben, enthält SMB1 verschiedene Indikatoren möglicher Fehler. Diese Bits werden während der Bearbeitung von Operationen gesetzt und zurückgesetzt.

Tabelle D-2 Sondermerker SMB1 (SM1.0 bis SM1.7)

Sondermerker	Beschreibung
SM1.0	Dieses Bit wird bei der Bearbeitung bestimmter Operationen aktiviert, wenn das Ergebnis Null beträgt.
SM1.1	Dieses Bit wird bei der Bearbeitung bestimmter Operationen aktiviert, wenn sich ein Überlauf ereignet oder wenn ein ungültiger numerischer Wert erkannt wird.
SM1.2	Dieses Bit wird aktiviert, wenn das Ergebnis einer arithmetischen Operation negativ ist.
SM1.3	Dieses Bit wird bei Division durch Null aktiviert.
SM1.4	Dieses Bit wird aktiviert, wenn durch die Operation Wert in Tabelle eintragen die Tabelle überläuft.
SM1.5	Dieses Bit wird aktiviert, wenn mit einer der Operationen LIFO oder FIFO versucht wird, einen Wert aus einer leeren Tabelle auszulesen.
SM1.6	Dieses Bit wird aktiviert, wenn ein Wert, der nicht im BCD-Format vorliegt, in einen Binärwert umgewandelt werden soll.
SM1.7	Dieses Bit wird aktiviert, wenn ein ASCII-Wert nicht in einen gültigen Hexadezimalwert umgewandelt werden kann.

SMB2: Empfang bei der frei programmierbaren Kommunikation

SMB2 dient als Puffer für empfangene Zeichen bei der frei programmierbaren Kommunikation. Wie in Tabelle D-3 beschrieben, werden die Zeichen, die während der frei programmierbaren Kommunikation empfangen werden, in diesem Speicher abgelegt, damit das KOP-Programm schnell auf die Werte zugreifen kann.

Tabelle D-3 Sondermerker SMB2

Sondermerker	Beschreibung
SMB2	Dieses Byte enthält alle Zeichen, die in der frei programmierbaren Kommunikation über Schnittstelle 0 oder Schnittstelle 1 empfangen werden.

SMB3: Paritätsfehler bei der frei programmierbaren Kommunikation

SMB3 wird bei der frei programmierbaren Kommunikation verwendet und enthält ein Bit, das gesetzt wird, wenn bei einem der empfangenen Zeichen ein Paritätsfehler erkannt wird. Wie in Tabelle D-4 beschrieben, wird SM3.0 eingeschaltet, wenn ein Paritätsfehler erkannt wird. Mit diesem Bit verwerfen Sie die Meldung.

Tabelle D-4 Sondermerker SMB3 (SM3.0 bis SM3.7)

Sondermerker	Beschreibung
SM3.0	Paritätsfehler an Schnittstelle 0 oder Schnittstelle 1 (0 = kein Fehler; 1 = Fehler)
SM3.1 bis SM3.7	Reserviert.

SMB4: Überlauf der Warteschlange

Wie in Tabelle D-5 beschrieben, enthält SMB4 die Überlaufbits für die Interrupt-Warteschlange, einen Statusindikator, der angibt, ob die Interrupts freigegeben oder gesperrt sind sowie einen Merker der anzeigt, ob sich der Sender im Leerlauf befindet. Die Bits für den Warteschlangenüberlauf zeigen an, daß sich die Interrupts entweder in einer höheren Geschwindigkeit ereignen, als sie bearbeitet werden können oder daß sie durch die Operation Alle Interruptereignisse sperren gesperrt wurden.

Tabelle D-5 Sondermerker SMB4 (SM4.0 bis SM4.7)

Sondermerker	Beschreibung
SM4.0 ¹	Wird aktiviert, wenn in der Warteschlange für Kommunikationsinterrupts ein Überlauf auftritt.
SM4.1 ¹	Wird aktiviert, wenn in der Warteschlange für Eingang interrupts ein Überlauf auftritt.
SM4.2 ¹	Wird aktiviert, wenn in der Warteschlange für zeitgesteuerte Interrupts ein Überlauf auftritt.
SM4.3	Wird aktiviert, wenn zur Laufzeit ein Programmierfehler erkannt wird.
SM4.4	Zeigt den Freigabestatus der Interrupts an. Wird aktiviert, wenn alle Interrupts freigegeben sind.
SM4.5	Wird aktiviert, wenn sich der Sender im Leerlauf befindet (Schnittstelle 0).
SM4.6	Wird aktiviert, wenn sich der Sender im Leerlauf befindet (Schnittstelle 1).
SM4.7	Reserviert.

¹ Verwenden Sie die Statusbits 4.0, 4.1 und 4.2 nur in Interruptprogrammen. Diese Statusbits werden zurückgesetzt, wenn die Warteschlange abgearbeitet ist und die Bearbeitung des Hauptprogramms wieder aufgenommen wird.

SMB5: Status der Ein- und Ausgänge

Wie in Tabelle D-6 beschrieben, enthält SMB5 Statusbits zu den Fehlerbedingungen, die bei den Ein- und Ausgängen erkannt wurden. Diese Bits bieten einen Überblick über die aufgetretenen E/A-Fehler.

Tabelle D-6 Sondermerker SMB5 (SM5.0 bis SM5.7)

Sondermerker	Beschreibung
SM5.0	Dieses Bit wird eingeschaltet, wenn ein E/A-Fehler auftritt.
SM5.1	Dieses Bit wird eingeschaltet, wenn zu viele digitale Ein- und Ausgänge an den E/A-Bus angeschlossen sind.
SM5.2	Dieses Bit wird aktiviert, wenn zu viele Analogein- und Analogausgänge an den E/A-Bus angeschlossen sind.
SM5.3 bis SM5.7	Reserviert.

SMB6: Kennregister der CPU

Wie in Tabelle D-7 beschrieben, ist SMB6 das Kennregister der CPU. SM6.4 bis SM6.7 enthalten die Kennung der CPU. SM6.0 bis SM6.3 sind für zukünftige Funktionen reserviert.

Tabelle D-7 Sondermerker SMB6

Sondermerker	Beschreibung
Format	<div style="display: flex; justify-content: space-between; align-items: center;"> <div style="text-align: center;"> <small>MSB</small> 7 x x x x </div> <div style="text-align: center;"> <small>LSB</small> 0 r r r r </div> <div style="margin-left: 20px;">Kennregister der CPU</div> </div>
SM6.4 bis SM6.7	xxxx = 0000 = CPU 212 0010 = CPU 214 1000 = CPU 215 1001 = CPU 216
SM6.0 bis SM6.3	Reserviert.

SMB7: Reserviert

SMB7 ist für zukünftige Funktionen reserviert.

SMB8 bis SMB21: Kenn- und Fehlerregister der Erweiterungsmodule

SMB8 bis SMB21 sind für die Erweiterungsmodule 0 bis 6 in Bytepaaren gegliedert. Wie in Tabelle D-8 beschrieben, ist das Byte mit der geraden Nummer in einem Paar das Kennregister des Erweiterungsmoduls. Dieses Byte kennzeichnet den Modultyp sowie die Art und Anzahl der Ein- und Ausgänge. Das Byte mit der ungeraden Nummer in einem Paar ist das Fehlerregister des Moduls. Dieses Byte zeigt jeden in den Ein- und Ausgängen des Moduls erkannten Fehler an.

Tabelle D-8 Sondermerker SMB8 bis SMB21

Sondermerker	Beschreibung
Format	<div style="display: flex; justify-content: space-between;"> <div style="width: 45%;"> <p>Byte mit gerader Nummer: Kennregister des Moduls</p> <div style="display: flex; justify-content: space-between; align-items: center;"> <div style="text-align: center;"> <small>MSB</small> 7 M t t A i i </div> <div style="text-align: center;"> <small>LSB</small> 0 Q Q </div> </div> <p>M: Modul vorhanden 0 = Vorhanden 1 = Nicht vorhanden</p> <p>tt: 00 Erweiterungsmodul 01 Reserviert 10 Reserviert 11 Reserviert</p> <p>A Art E/A 0 = Digital 1 = Analog</p> <p>ii 00 Keine Eingänge 01 2 AE oder 8 DE 10 4 AE oder 16 DE 11 8 AE oder 32 DE</p> </div> <div style="width: 45%;"> <p>Byte mit ungerader Nummer: Fehlerregister des Moduls</p> <div style="display: flex; justify-content: space-between; align-items: center;"> <div style="text-align: center;"> <small>MSB</small> 7 C 0 0 0 </div> <div style="text-align: center;"> <small>LSB</small> 0 R P r r </div> </div> <p>C: Konfigurationsfehler R: Bereichsüberschreitung P: Kein Anwenderstrom rr: Reserviert</p> <p>QQ 00 Keine Ausgänge 01 2 AA oder 8 DA 10 4 AA oder 16 DA 11 8 AA oder 32 DA</p> </div> </div>
SMB8	Kennregister Modul 0
SMB9	Fehlerregister Modul 0
SMB10	Kennregister Modul 1
SMB11	Fehlerregister Modul 1

Tabelle D-8 Sondermerker SMB8 bis SMB21, Fortsetzung

Sondermerker	Beschreibung
SMB12 SMB13	Kennregister Modul 2 Fehlerregister Modul 2
SMB14 SMB15	Kennregister Modul 3 Fehlerregister Modul 3
SMB16 SMB17	Kennregister Modul 4 Fehlerregister Modul 4
SMB18 SMB19	Kennregister Modul 5 Fehlerregister Modul 5
SMB20 SMB21	Kennregister Modul 6 Fehlerregister Modul 6

SMW22 bis SMW26: Zykluszeiten

Wie in Tabelle D-9 beschrieben, liefern SMW22, SMW24 und SMW26 Informationen zur Zykluszeit: kürzeste Zykluszeit, längste Zykluszeit und letzte Zykluszeit in Millisekunden.

Tabelle D-9 Sondermerker SMW22 bis SMW26

Sondermerker	Beschreibung
SMW22	Dieses Wort speichert die Zykluszeit des letzten Zyklus.
SMW24	Dieses Wort speichert die kürzeste Zykluszeit seit Beginn des Betriebszustands RUN.
SMW26	Dieses Wort speichert die längste Zykluszeit seit Beginn des Betriebszustands RUN.

SMB28 und SMB29: Analogpotentiometer

Wie in Tabelle D-10 beschrieben, enthält SMB28 den Digitalwert, der die Position des Analogpotentiometers 0 darstellt. SMB29 speichert den Digitalwert, der die Position des Analogpotentiometers 1 darstellt.

Tabelle D-10 Sondermerker SMB28 und SMB29

Sondermerker	Beschreibung
SMB28	Dieses Byte speichert den Wert, der mit dem Analogpotentiometer 0 eingegeben wird. Dieser Wert wird einmal pro Zyklus in STOP/RUN aktualisiert.
SMB29	Dieses Byte speichert den Wert, der mit dem Analogpotentiometer 1 eingegeben wird. Dieser Wert wird einmal pro Zyklus in STOP/RUN aktualisiert.

SMB30 und SMB130: Steuerungsregister der frei programmierbaren Kommunikation

SMB30 steuert die frei programmierbare Kommunikation für die Schnittstelle 0. SMB130 steuert die frei programmierbare Kommunikation für die Schnittstelle 1. Sie können SMB30 und SMB130 lesen und schreiben. Wie in Tabelle D-11 beschrieben, konfigurieren diese Bytes die entsprechende Kommunikationsschnittstelle für die frei programmierbare Kommunikation. Ausgewählt werden kann die Unterstützung des Protokolls für die frei programmierbare Kommunikation sowie das Systemprotokoll.

Tabelle D-11 Sondermerker SMB30 und SMB130

Schnittstelle 0	Schnittstelle 1	Beschreibung								
Format von SMB30	Format von SMB130	<div style="display: flex; justify-content: space-between; align-items: center;"> MSB 7 <div style="border: 1px solid black; padding: 2px;"> <table style="border-collapse: collapse; text-align: center;"> <tr> <td style="border: 1px solid black; padding: 2px;">p</td> <td style="border: 1px solid black; padding: 2px;">p</td> <td style="border: 1px solid black; padding: 2px;">d</td> <td style="border: 1px solid black; padding: 2px;">b</td> <td style="border: 1px solid black; padding: 2px;">b</td> <td style="border: 1px solid black; padding: 2px;">b</td> <td style="border: 1px solid black; padding: 2px;">m</td> <td style="border: 1px solid black; padding: 2px;">m</td> </tr> </table> </div> LSB 0 </div> Steuerbyte für die frei programmierbare Kommunikation	p	p	d	b	b	b	m	m
p	p	d	b	b	b	m	m			
SM30.6 und SM30.7	SM130.6 und SM130.7	pp Auswahl der Parität 00 = keine Parität 01 = gerade Parität 10 = keine Parität 11 = ungerade Parität								
SM30.5	SM130.5	d Datenbits pro Zeichen 0 = 8 Bits pro Zeichen 1 = 7 Bits pro Zeichen								
SM30.2 bis SM30.4	SM130.2 bis SM130.4	bbb Baudrate für die frei programmierbare Kommunikation 000 = 38.400 Baud (bei der CPU 212: 19.200 Baud) 001 = 19.200 Baud 010 = 9.600 Baud 011 = 4.800 Baud 100 = 2.400 Baud 101 = 1.200 Baud 110 = 600 Baud 111 = 300 Baud								
SM30.0 und SM30.1	SM130.0 und SM130.1	mm Auswahl des Protokolls 00 = Protokoll der Punkt-zu-Punkt-Schnittstelle (PPI/Slave-Modus) 01 = Protokoll der frei programmierbaren Kommunikation 10 = PPI/Master-Modus 11 = Reserviert (Voreinstellung PPI/Slave-Modus)								

SMB31 und SMW32: Schreibsteuerung nullspannungsfester Speicher (EEPROM)

Sie können über Ihr Programm einen Wert, der sich im Variablenspeicher befindet, im nullspannungsfesten Speicher (EEPROM) ablegen. Hierzu laden Sie die Adresse, die nullspannungsfest gespeichert werden soll, in SMW32. Dann laden Sie SMB31 mit dem Befehl, den Wert zu speichern. Nachdem Sie den Befehl zum Speichern des Werts abgesetzt haben, dürfen Sie den Wert im Variablenspeicher solange nicht mehr ändern, bis die CPU den Sondermerker SM31.7 zurücksetzt und dadurch das Ende der Speicheroperation anzeigt.

Am Ende eines jeden Zyklus prüft die CPU, ob ein Wert im nullspannungsfesten Speicher abgelegt werden soll. Ist dies der Fall, dann wird der angegebene Wert im nullspannungsfesten Speicher abgelegt.

Wie in Tabelle D-12 beschrieben, definiert SMB31 die Größe der Daten, die im nullspannungsfesten Speicher abgelegt werden sollen, und liefert außerdem den Befehl, der die Bearbeitung der Speicheroperation auslöst. SMW32 speichert die Anfangsadresse der Daten im Variablenspeicher, die nullspannungsfest gespeichert werden sollen.

Tabelle D-12 Sondermerker SMB31 und SMW32

Sondermerker	Beschreibung									
Format	<p>SMB31: Software-Befehl</p> <p style="text-align: center;">MSB LSB</p> <p style="text-align: center;">7 0</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td style="width: 15px; text-align: center;">c</td> <td style="width: 15px; text-align: center;">0</td> <td style="width: 15px; text-align: center;">0</td> <td style="width: 15px; text-align: center;">0</td> <td style="width: 15px; text-align: center;">0</td> <td style="width: 15px; text-align: center;">0</td> <td style="width: 15px; text-align: center;">s</td> <td style="width: 15px; text-align: center;">s</td> </tr> </table> <p>SMW32 Adresse im Variablenspeicher</p> <p style="text-align: center;">MSB LSB</p> <p style="text-align: center;">15 0</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td style="width: 100px; text-align: center;">Adresse im Variablenspeicher</td> </tr> </table>	c	0	0	0	0	0	s	s	Adresse im Variablenspeicher
c	0	0	0	0	0	s	s			
Adresse im Variablenspeicher										
SM31.0 und SM31.1	<p>ss: Größe des zu speichernden Werts</p> <p>00 = Byte</p> <p>01 = Byte</p> <p>10 = Wort</p> <p>11 = Doppelwort</p>									
SM30.7	<p>c: Im nullspannungsfesten Speicher (EEPROM) ablegen</p> <p>0 = Keine Anforderung zu speichern</p> <p>1 = Anwenderprogramm fordert die CPU auf, den Wert im nullspannungsfesten Speicher abzulegen.</p> <p>Die CPU setzt dieses Bit nach jeder Speicheroperation zurück.</p>									
SMW32	<p>Die Adresse im Variablenspeicher von den Daten, die nullspannungsfest abgelegt werden sollen, ist in SMW32 gespeichert. Der Wert wird als Versatz von V0 angegeben. Wird eine Speicheroperation ausgeführt, dann wird der Wert dieser Adresse im Variablenspeicher an der entsprechenden Adresse des Variablenspeichers im nullspannungsfesten EEPROM abgelegt.</p>									

SMB34 und SMB35: Intervallregister für zeitgesteuerte Interrupts

Wie in Tabelle D-13 beschrieben, gibt SMB34 das Intervall für den zeitgesteuerten Interrupt 0 und SMB35 das Intervall für den zeitgesteuerten Interrupt 1 an. Sie können das Intervall (in Inkrementen von 1 ms) zwischen 5 und 255 ms angeben. Die CPU erhält den Wert des Intervalls, wenn der entsprechende zeitgesteuerte Interrupt einem Interruptprogramm zugeordnet wird. Um den Wert des Intervalls zu ändern, müssen Sie den zeitgesteuerten Interrupt entweder demselben oder einem anderen Interruptprogramm erneut zuordnen. Sie können das zeitgesteuerte Interruptereignis beenden, indem Sie das Ereignis trennen.

Tabelle D-13 Sondermerker SMB34 und SMB35

Sondermerker	Beschreibung
SMB34	Dieses Byte gibt das Intervall (in Inkrementen von 1 ms zwischen 5 und 255 ms) für den zeitgesteuerten Interrupt 0 an.
SMB35	Dieses Byte gibt das Intervall (in Inkrementen von 1 ms zwischen 5 und 255 ms) für den zeitgesteuerten Interrupt 1 an.

SMB36 bis SMB65: HSC-Register

Wie in Tabelle D-14 beschrieben, werden SMB36 bis SMB65 dazu verwendet, den Betrieb der schnellen Zähler zu überwachen und zu steuern.

Tabelle D-14 Sondermerker SMB36 bis SMB65

Sondermerker	Beschreibung
SM36.0 bis SM36.4	Reserviert.
SM36.5	Statusbit HSC0 - Aktuelle Zählrichtung: 1 = Vorwärtszählen
SM36.6	Statusbit HSC0 - Aktueller Wert gleich voreingestellter Wert: 1 = gleich
SM36.7	Statusbit HSC0 - Aktueller Wert ist größer als voreingestellter Wert: 1 = größer als
SM37.0 bis SM37.2	Reserviert.
SM37.3	Steuerbit HSC0 - Zählrichtung: 1 = Vorwärtszählen
SM37.4	HSC0 - Richtung aktualisieren: 1 = Richtung aktualisieren
SM37.5	HSC0 - Voreingestellten Wert aktualisieren: 1 = neuen Wert in die Voreinstellung von HSC0 schreiben
SM37.6	HSC0 - Aktuellen Wert aktualisieren: 1 = neuen Wert in aktuellen Wert von HSC0 schreiben
SM37.7	HSC0 - Freigabebit: 1 = Freigeben
SMB38 SMB39 SMB40 SMB41	HSC0 - Neuer aktueller Wert SMB38 ist das höchstwertige Byte und SMB41 ist das niederwertigste Byte.
SMB42 SMB43 SMB44 SMB45	HSC0 - Neuer voreingestellter Wert SMB42 ist das höchstwertige Byte und SMB45 ist das niederwertigste Byte.
SM46.0 bis SM46.4	Reserviert.
SM46.5	Statusbit HSC1 - Aktuelle Zählrichtung: 1 = Vorwärtszählen
SM46.6	Statusbit HSC1 - Aktueller Wert gleich voreingestellter Wert: 1 = gleich
SM46.7	Statusbit HSC1 - Aktueller Wert ist größer als voreingestellter Wert: 1 = größer als
SM47.0	Steuerbit HSC1 - Aktivitätsstufe zum Rücksetzen: 0 = hoch; 1 = niedrig
SM47.1	Steuerbit HSC1 - Aktivitätsstufe zum Starten: 0 = hoch; 1 = niedrig
SM47.2	HSC1 - Zählgeschwindigkeit A/B-Zähler: 0 = vierfache Zählgeschwindigkeit; 1 = einfache Zählgeschwindigkeit
SM47.3	Steuerbit HSC1 - Zählrichtung: 1 = Vorwärtszählen
SM47.4	HSC1 - Richtung aktualisieren: 1 = Richtung aktualisieren
SM47.5	HSC1 - Voreingestellten Wert aktualisieren: 1 = neuen Wert in die Voreinstellung von HSC1 schreiben
SM47.6	HSC1 - Aktuellen Wert aktualisieren: 1 = neuen Wert in aktuellen Wert von HSC1 schreiben
SM47.7	HSC1 - Freigabebit: 1 = Freigeben

Tabelle D-14 Sondermerker SMB36 bis SMB65, Fortsetzung

Sondermerker	Beschreibung
SMB48 SMB49 SMB50 SMB51	HSC1 - Neuer aktueller Wert SMB48 ist das höchstwertige Byte und SMB51 ist das niederwertigste Byte.
SMB52 bis SMB55	HSC1 - Neuer voreingestellter Wert SMB52 ist das höchstwertige Byte und SMB55 ist das niederwertigste Byte.
SM56.0 bis SM56.4	Reserviert.
SM56.5	Statusbit HSC2 - Aktuelle Zählrichtung: 1 = Vorwärtszählen
SM56.6	Statusbit HSC2 - Aktueller Wert gleich voreingestellter Wert: 1 = gleich
SM56.7	Statusbit HSC2 - Aktueller Wert ist größer als voreingestellter Wert: 1 = größer als
SM57.0	Steuerbit HSC2 - Aktivitätsstufe zum Rücksetzen: 0 = hoch; 1 = niedrig
SM57.1	Steuerbit HSC2 - Aktivitätsstufe zum Starten: 0 = hoch; 1 = niedrig
SM57.2	HSC2 - Zählgeschwindigkeit A/B-Zähler: 0 = vierfache Zählgeschwindigkeit; 1 = einfache Zählgeschwindigkeit
SM57.3	Steuerbit HSC2 - Zählrichtung: 1 = Vorwärtszählen
SM57.4	HSC2 - Richtung aktualisieren: 1 = Richtung aktualisieren
SM57.5	HSC2 - Voreingestellten Wert aktualisieren: 1 = neuen Wert in die Voreinstellung von HSC2 schreiben
SM57.6	HSC2 - Aktuellen Wert aktualisieren: 1 = neuen Wert in aktuellen Wert von HSC2 schreiben
SM57.7	HSC2 - Freigabebit: 1 = Freigeben
SMB58 SMB59 SMB60 SMB61	HSC2 - Neuer aktueller Wert SMB58 ist das höchstwertige Byte und SMB61 ist das niederwertigste Byte.
SMB62 SMB63 SMB64 SMB65	HSC2 - Neuer voreingestellter Wert SMB62 ist das höchstwertige Byte und SMB65 ist das niederwertigste Byte.

SMB66 bis SMB85: PTO/PWM-Register

Wie in Tabelle D-15 beschrieben, dienen SMB66 bis SMB85 zum Überwachen und Steuern der Operationen Impulsfolge und Impulsdauermodulation. Sie entnehmen der Beschreibung der schnellen Ausgangsoperationen in Kapitel 10 ausführliche Informationen zu diesen Bits.

Tabelle D-15 Sondermerker SMB66 bis SMB85

Sondermerker	Beschreibung
SM66.0 bis SM66.5	Reserviert.
SM66.6	PTO0 - Pipeline-Überlauf: 0 = kein Überlauf; 1 = Überlauf
SM66.7	PTO0 - Leerlaufbit: 0 = PTO wird bearbeitet; 1 = PTO im Leerlauf

Tabelle D-15 Sondermerker SMB66 bis SMB85, Fortsetzung

Sondermerker	Beschreibung
SM67.0	PTO0/PWM0 - Zykluszeit aktualisieren: 1 = neue Zykluszeit schreiben
SM67.1	PWM0 - Impulsdauer aktualisieren: 1 = neue Impulsdauer schreiben
SM67.2	PTO0 - Impulswert aktualisieren: 1 = neuen Impulswert schreiben
SM67.3	PTO0/PWM0 - Zeitbasis: 0 = 1 µs/Takt, 1 = 1 ms/Takt
SM67.4 und SM67.5	Reserviert.
SM67.6	PTO0/PWM0 - Modus auswählen: 0 = PTO; 1 = PWM
SM67.7	PTO0/PWM0 - Freigabebit: 1 = freigeben
SMB68 SMB69	PTO0/PWM0 - Zykluszeit SMB68 ist das höchstwertige Byte und SMB69 ist das niederwertigste Byte.
SMB70 SMB71	PWM0 - Impulsdauer SMB70 ist das höchstwertige Byte und SMB71 ist das niederwertigste Byte.
SMB72 SMB73 SMB74 SMB75	PTO0 - Impulswert SMB72 ist das höchstwertige Byte und SMB75 ist das niederwertigste Byte.
SM76.0 bis SM76.5	Reserviert.
SM76.6	PTO1 - Pipeline-Überlauf: 0 = kein Überlauf; 1 = Überlauf
SM76.7	PTO1 - Leerlaufbit: 0 = PTO wird bearbeitet; 1 = PTO im Leerlauf
SM77.0	PTO1/PWM1 - Zykluszeit aktualisieren: 1 = neue Zykluszeit schreiben
SM77.1	PWM1 - Impulsdauer aktualisieren: 1 = neue Impulsdauer schreiben
SM77.2	PTO1 - Impulswert aktualisieren: 1 = neuen Impulswert schreiben
SM77.3	PTO1/PWM1 - Zeitbasis: 0 = 1 µs/Takt, 1 = 1 ms/Takt
SM77.4 und SM77.5	Reserviert.
SM77.6	PTO1/PWM1 - Modus auswählen: 0 = PTO; 1 = PWM
SM77.7	PTO1/PWM1 - Freigabebit: 1 = Freigeben
SMB78 SMB79	PTO1/PWM1 - Zykluszeit SMB78 ist das höchstwertige Byte und SMB79 ist das niederwertigste Byte.
SMB80 SMB81	PWM1 - Impulsdauer SMB80 ist das höchstwertige Byte und SMB81 ist das niederwertigste Byte.
SMB82 SMB83 SMB84 SMB85	PTO1 - Impulswert SMB82 ist das höchstwertige Byte und SMB85 ist das niederwertigste Byte.

SMB86 bis SMB94 und SMB186 bis SMB194: Steuerung des Meldungsempfangs

Wie in Tabelle D-16 beschrieben, werden SMB86 bis SMB94 und SMB186 bis SMB194 dazu verwendet, den Status der Funktion zum Empfangen von Meldungen zu steuern und zu lesen.

Tabelle D-16 Sondermerker SMB86 bis SMB94 und SMB186 bis SMB194

Schnittstelle 0	Schnittstelle 1	Beschreibung								
SMB86	SMB186	<div style="display: flex; justify-content: space-between; align-items: center;"> MSB 7 LSB 0 </div> <div style="display: flex; align-items: center; margin-top: 5px;"> <table border="1" style="border-collapse: collapse; text-align: center;"> <tr> <td style="padding: 2px 5px;">n</td> <td style="padding: 2px 5px;">r</td> <td style="padding: 2px 5px;">e</td> <td style="padding: 2px 5px; background-color: #cccccc;">0</td> <td style="padding: 2px 5px; background-color: #cccccc;">0</td> <td style="padding: 2px 5px;">t</td> <td style="padding: 2px 5px;">c</td> <td style="padding: 2px 5px;">p</td> </tr> </table> <div style="margin-left: 10px;"> <p>Statusbyte zum Empfangen von Meldungen</p> <p>n: 1 = Meldungsempfang vom Anwender gesperrt</p> <p>r: 1 = Meldungsempfang beendet: Fehler in Eingangsparametern oder fehlendes Kriterium zum Starten bzw. Beenden</p> <p>e: 1 = Endezeichen empfangen</p> <p>t: 1 = Meldungsempfang beendet: Zeit abgelaufen</p> <p>c: 1 = Meldungsempfang beendet: maximale Zeichenzahl erreicht</p> <p>p: 1 = Meldungsempfang beendet: Paritätsfehler</p> </div> </div>	n	r	e	0	0	t	c	p
n	r	e	0	0	t	c	p			
SMB87	SMB187	<div style="display: flex; justify-content: space-between; align-items: center;"> MSB 7 LSB 0 </div> <div style="display: flex; align-items: center; margin-top: 5px;"> <table border="1" style="border-collapse: collapse; text-align: center;"> <tr> <td style="padding: 2px 5px;">n</td> <td style="padding: 2px 5px;">x</td> <td style="padding: 2px 5px;">y</td> <td style="padding: 2px 5px;">z</td> <td style="padding: 2px 5px;">m</td> <td style="padding: 2px 5px;">t</td> <td style="padding: 2px 5px; background-color: #cccccc;">0</td> <td style="padding: 2px 5px; background-color: #cccccc;">0</td> </tr> </table> <div style="margin-left: 10px;"> <p>Steuerbyte zum Empfangen von Meldungen</p> <p>n: 0 = Funktion zum Empfangen von Meldungen ist gesperrt. 1 = Funktion zum Empfangen von Meldungen ist aktiviert. Das Bit zum Sperren/Freigeben des Meldungsempfangs wird jedesmal, wenn die Operation RCV bearbeitet wird, geprüft.</p> <p>x: 0 = SMB88 oder SMB188 ignorieren. 1 = Mit dem Wert von SMB88 oder von SMB188 den Beginn der Meldung erkennen.</p> <p>y: 0 = SMB89 oder SMB189 ignorieren. 1 = Mit dem Wert von SMB89 oder von SMB189 den Beginn der Meldung erkennen.</p> <p>z: 0 = SMW90 oder SMB190 ignorieren. 1 = Mit dem Wert von SMW90 den Beginn der Meldung erkennen.</p> <p>m: 0 = Taktgeber mißt die Zeit zwischen den Zeichen. 1 = Taktgeber mißt die Zeit der Meldungen</p> <p>t: 0 = SMW92 oder SMW192 ignorieren. 1 = Empfang beenden, wenn die Zeitdauer von SMW92 oder SMW192 überschritten ist.</p> <p>Diese Bits definieren die Kriterien zum Kennzeichnen einer Meldung (einschließlich der Kriterien für den Beginn und das Ende einer Meldung). Um den Beginn einer Meldung zu erkennen, werden die freigegebenen Kriterien für den Beginn einer Meldung logisch durch UND verknüpft und müssen in Reihe auftreten (Leerlaufnlinie gefolgt von einem Startzeichen). Um das Ende einer Meldung zu erkennen, werden die freigegebenen Kriterien für das Ende einer Meldung logisch durch ODER verknüpft.</p> <p>Gleichungen für die Kriterien für den Beginn und das Ende einer Meldung: Beginn der Meldung = z * x Ende der Meldung = y + t + maximale Zeichenzahl erreicht</p> <p>Hinweis: Eine freigegebene Funktion zum Empfangen von Meldungen wird sofort automatisch beendet, wenn ein Überlauf- oder Paritätsfehler auftritt. Sie müssen ein Kriterium für den Beginn (x oder z) und ein Kriterium für das Ende (y, t oder maximale Zeichenzahl erreicht) definieren, damit die Funktion zum Empfangen von Meldungen fehlerfrei arbeiten kann.</p> </div> </div>	n	x	y	z	m	t	0	0
n	x	y	z	m	t	0	0			
SMB88	SMB188	Zeichen für den Beginn einer Meldung								
SMB89	SMB189	Zeichen für das Ende einer Meldung								

Tabelle D-16 Sondermerker SMB86 bis SMB94 und SMB186 bis SMB194, Fortsetzung

Schnittstelle 0	Schnittstelle 1	Beschreibung
SMB90 SMB91	SMB190 SMB191	Dauer der Leerlauflinie in Millisekunden. Das ersten Zeichen, das nach Ablauf der Zeit für die Leerlauflinie empfangen wird, kennzeichnet den Beginn einer neuen Meldung. SM90 (oder SM190) ist das höchstwertige Byte und SM91 (oder SM191) ist das niederwertigste Byte.
SMB92 SMB93	SMB192 SMB193	Wert für die Zeitüberwachung beim Messen der Zeit zwischen den Zeichen und der Zeit der Meldungen (in ms). Ist die Zeit überschritten, wird das Empfangen von Meldungen beendet. SM92 (oder SM192) ist das höchstwertige Byte und SM93 (oder SM193) ist das niederwertigste Byte.
SMB94	SMB194	Maximale Zeichenzahl, die empfangen werden kann (1 bis 255 Bytes). Hinweis: Dieser Bereich muß auf die maximal zu erwartende Puffergröße eingestellt werden, auch wenn der Empfangsabbruch durch die Zeichenzählfunktion nicht eingesetzt wird.

SMB110 bis SMB115: Status des DP-Standardprotokolls

Wie in Tabelle D-17 beschrieben, werden SMB110 bis SMB115 dazu verwendet, den Status des DP-Standardprotokolls zu überwachen.

Hinweis

Diese Adressen werden nur für den Status verwendet. Schreiben Sie nicht in diese Adressen. Die Adressen geben Werte an, die von dem DP-Master während des Konfigurationsvorgangs eingerichtet wurden.

Tabelle D-17 Sondermerker SMB110 bis SMB115

Sondermerker	Beschreibung								
SMB110	<div style="display: flex; justify-content: space-between; align-items: center;"> <div style="text-align: center;"> <small>MSB</small> 7 <table border="1" style="border-collapse: collapse; text-align: center;"> <tr> <td style="width: 20px;">0</td> <td style="width: 20px;">0</td> <td style="width: 20px;">0</td> <td style="width: 20px;">0</td> <td style="width: 20px;">0</td> <td style="width: 20px;">0</td> <td style="width: 20px;">s</td> <td style="width: 20px;">s</td> </tr> </table> </div> <div style="text-align: center;"> <small>LSB</small> 0 </div> </div> <p style="margin-left: 100px;">Schnittstelle 1: Statusbyte des DP-Standardprotokolls</p> <p>ss: Statusbyte DP-Standardprotokoll 00 = DP-Kommunikation nach Anlauf nicht initiiert 01 = Fehler in Konfiguration oder Parametrierung 10 = Modus für Datenaustausch aktiviert 11 = Modus für Datenaustausch nicht aktiviert</p> <p>SM111 bis SM115 werden jedesmal aktualisiert, wenn die CPU Konfigurations- bzw. Parametrierungsdaten aufnimmt. Diese Adressen werden aktualisiert, auch wenn ein Konfigurations- bzw. Parametrierungsfehler erkannt wird. Die Adressen werden bei jedem Einschalten der CPU gelöscht.</p>	0	0	0	0	0	0	s	s
0	0	0	0	0	0	s	s		
SMB111	Dieses Byte definiert die Adresse vom Master des Slave (0 bis 126).								
SMB112 SMB113	Dieses Byte definiert die Adresse im Variablenspeicher des Ausgabepuffers (Versatz von VB0). SM112 ist das höchstwertige Byte (MSB) und SMB113 ist das niederwertigste Byte (LSB).								
SMB114	Dieses Byte definiert die Anzahl der Ausgabebytes.								
SMB115	Dieses Byte definiert die Anzahl der Eingabebytes.								

Einsetzen von STEP 7-Micro/WIN mit STEP 7 und STEP 7-Micro/DOS

E

STEP 7-Micro/WIN 32 wird in Verbindung mit STEP 7 als integriertes Produkt betrieben. Sie können STEP 7-Micro/WIN in STEP 7 genauso aufrufen wie andere STEP 7-Tools, z.B. den Symbol-Editor oder den Programm-Editor. Ausführliche Informationen zur Programmiersoftware STEP 7 entnehmen Sie der Online-Hilfe von STEP 7 und dem *SIMATIC STEP 7 Benutzerhandbuch*.

Sie können auch Programmdateien, die Sie in STEP 7-Micro/DOS erstellt haben, in STEP 7-Micro/WIN importieren. Diese Dateien können dann bearbeitet und von STEP 7-Micro/WIN in die CPU geladen werden. Ausführliche Informationen zu STEP 7-Micro/DOS entnehmen Sie der Online-Hilfe und dem *SIMATIC STEP 7-Micro/DOS Benutzerhandbuch*.

Kapitelübersicht

Abschnitt	Beschreibung	Seite
E.1	Arbeiten mit STEP 7-Micro/WIN und STEP 7	E-2
E.2	Importieren von Dateien aus STEP 7-Micro/DOS	E-4

E.1 Arbeiten mit STEP 7-Micro/WIN und STEP 7

Sie können in STEP 7 über die Software STEP 7-Micro/WIN Ihre S7-200-Programme aufrufen:

- Offline: Sie können eine SIMATIC 200-Station in ein STEP 7-Projekt einfügen.
- Online: Sie können über eine Liste der im Netz aktiven Stationen auf die S7-200 CPU zugreifen.

Wenn Sie die Programmiersoftware STEP 7-Micro/WIN aus STEP 7 heraus betreiben, kann sich das Erscheinungsbild der Software STEP 7-Micro/WIN geringfügig von dem Erscheinungsbild der Einzelapplikation unterscheiden:

- Dialogfelder zum Auswählen von Objekten: Läuft STEP 7-Micro/WIN innerhalb der Software STEP 7, verwenden Sie die Dialogfelder von STEP 7 zum Auswählen der S7-200-Stationen innerhalb der Hierarchie von STEP 7. Sie können nur zu den S7-200-Objekten navigieren, die sich innerhalb der STEP 7-Hierarchie befinden. Sie können keine Objekte öffnen (Projekte, Programme, Datenbausteine oder Statustabellen), die in der Projekthierarchie von STEP 7-Micro/WIN abgelegt sind.
- Sprache und Mnemonik: Läuft STEP 7-Micro/WIN in der Software STEP 7, gelten die Einstellungen für Sprache und Mnemonik von STEP 7.

Erstellen einer S7-200 CPU in einem Projekt in STEP 7

Zum Anlegen einer S7-200 CPU in der Programmiersoftware STEP 7 fügen Sie eine SIMATIC 200-Station in ein STEP 7-Projekt ein. Daraufhin erstellt STEP 7 die S7-200-Station. Im Gegensatz zu Stationen der S7-300 und S7-400 gehören zu einer S7-200-Station keine anderen Objekte (z.B. CPUs oder Netze). Eine einzige S7-200-Station stellt ein komplettes STEP 7-Micro/WIN-Projekt dar. Dies umfaßt das Programm, den Datenbaustein, die Symboltabelle und die Statustabelle.

Mit der Programmiersoftware STEP 7 können Sie das S7-200-Projekt kopieren, verschieben, löschen oder umbenennen.

Hinweis

Sie können eine S7-200 CPU ("SIMATIC 200-Station") nur in der Wurzel eines STEP 7-Projekts einfügen. Sie können die SIMATIC 200-Station unter keinem anderen Objekt anlegen. Es gibt keine Wechselwirkungen zwischen der SIMATIC 200-Station und anderen Objekten in STEP 7.

Zum Erstellen einer S7-200-Station gehen Sie folgendermaßen vor:

1. Legen Sie mit dem Menübefehl **Datei ► Neu** ein neues Projekt im Projektfenster des SIMATIC Managers an.
2. Wählen Sie den Menübefehl **Einfügen ► Station ► SIMATIC 200-Station**, um ein S7-200-Objekt anzulegen.
3. Zum Bearbeiten der S7-200-Station doppelklicken Sie auf das S7-200-Objekt. Daraufhin öffnet sich die Station und STEP 7 ruft die Programmiersoftware STEP 7-Micro/WIN auf.

Hinweis

Sie können nie mehr als ein Fenster geöffnet haben, in dem die Programmiersoftware STEP 7-Micro/WIN läuft. Haben Sie bereits ein anderes Projekt für die S7-200 geöffnet, müssen Sie dieses Projekt zunächst schließen, bevor Sie ein anderes S7-200-Projekt öffnen können.

Bearbeiten einer S7-200 CPU im Online-Betrieb in STEP 7

Der SIMATIC Manager bietet Ihnen im Online-Betrieb eine Liste der S7-Teilnehmer und Stationen, die sich im Netz befinden. Diese Liste führt auch die S7-200-Teilnehmer (Stationen) auf, die an das Netz angeschlossen sind. Wenn Sie in der Liste einen S7-200-Teilnehmer wählen, ruft STEP 7 die Programmiersoftware STEP 7-Micro/WIN auf. STEP 7-Micro/WIN öffnet ein leeres (namenloses) Projekt und lädt das Anwenderprogramm, den Datenbaustein und die CPU-Konfiguration aus der S7-200 CPU.

Hinweis

Es können auch mehrere Netze vorhanden sein, auf die Sie nur über STEP 7 oder nur über STEP 7-Micro/WIN zugreifen können. Läuft STEP 7-Micro/WIN innerhalb der Software STEP 7, dann werden in der Liste der Netzteilnehmer nur die Stationen aufgeführt, die über STEP 7 zugänglich sind.

Aufrufen eines STEP 7-Projekts in STEP 7-Micro/WIN

Sie können auf ein Anwenderprogramm für eine S7-200-Station, das in einem STEP 7-Projekt abgelegt ist, zugreifen, auch wenn die Software STEP 7-Micro/WIN nicht in STEP 7 läuft. Zum Bearbeiten des Anwenderprogramms gehen Sie folgendermaßen vor:

1. Wählen Sie in der Programmiersoftware STEP 7-Micro/WIN den Menübefehl **Projekt ► Neu** zum Anlegen eines neuen Projekts.
2. Wählen Sie den Menübefehl **Projekt ► Importieren ► STEP 7-Projekt** (siehe Bild E-1).
3. Im Dialogfeld zum Auswählen eines Projekts in STEP 7 wählen Sie die S7-200-Station im STEP 7-Projekt und klicken auf die Schaltfläche "OK".

Daraufhin werden das Anwenderprogramm und andere Elemente (Datenbaustein, Status-tabelle, Symboltabelle) im Projekt in STEP 7-Micro/WIN geöffnet (siehe Bild E-1).

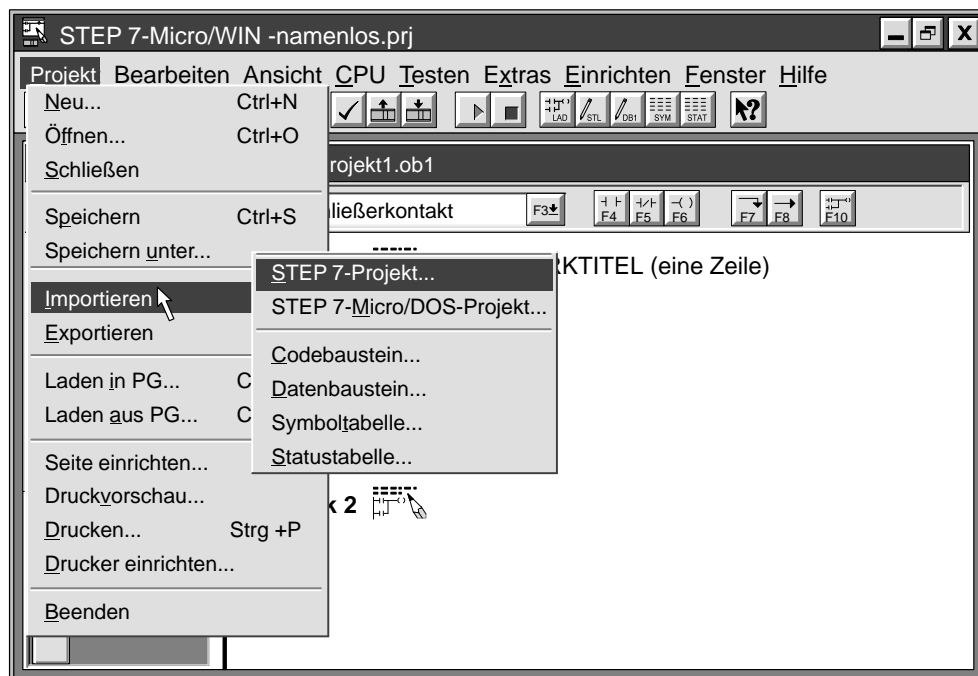


Bild E-1 Aufrufen eines STEP 7-Projekts in STEP 7-Micro/WIN

E.2 Importieren von Dateien aus STEP 7-Micro/DOS

Mit STEP 7-Micro/WIN können Sie Programme, die Sie mit der Software STEP 7-Micro/DOS erstellt haben, in Projekte in STEP 7-Micro/WIN importieren.

Importieren von Programmen aus STEP 7-Micro/DOS

Zum Importieren eines Programms aus STEP 7-Micro/DOS in ein Projekt in STEP 7-Micro/WIN gehen Sie folgendermaßen vor:

1. Mit dem Menübefehl **Projekt ► Neu** legen Sie ein neues Projekt an.
2. Wählen Sie den Menübefehl **Projekt ► Importieren ► STEP 7-Micro/DOS-Projekt...** (siehe Bild E-2).

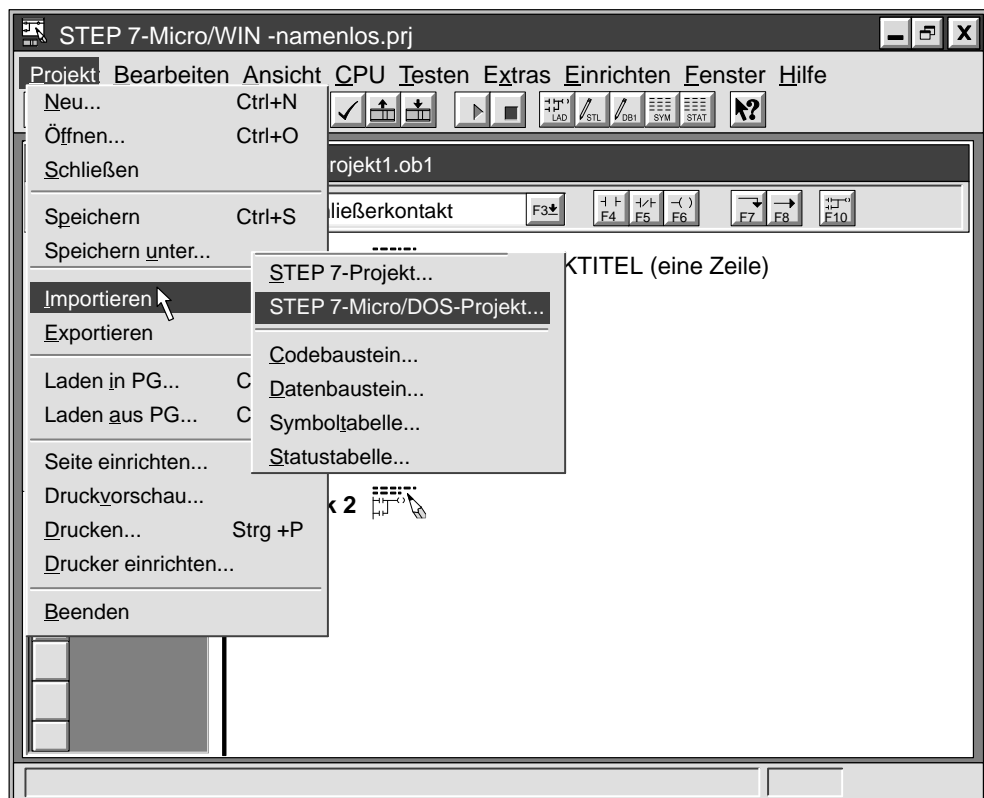


Bild E-2 Importieren von Dateien aus STEP 7-Micro/DOS

3. Beantworten Sie die Meldung (die darauf hinweist, daß durch das Importieren des Programms aus STEP 7-Micro/DOS das gesamte Programm überschrieben wird) mit "Ja". (Das neue Projekt enthält noch kein Programm.) Mit der Schaltfläche "Nein" können Sie die Funktion abbrechen.
4. Wählen Sie im Dialogfeld "Programm aus STEP 7-Micro/DOS importieren" (siehe Bild E-3) das Verzeichnis aus, in dem sich das Programm befindet, das Sie aus STEP 7-Micro/DOS importieren möchten.

5. Doppelklicken Sie auf die Datei aus STEP 7-Micro/DOS oder tragen Sie den Dateinamen ein (siehe Bild E-3).
6. Wählen Sie die Schaltfläche "Öffnen". Daraufhin werden das Programm aus STEP 7-Micro/DOS und die zugehörigen Dateien als namenloses Projekt geöffnet.

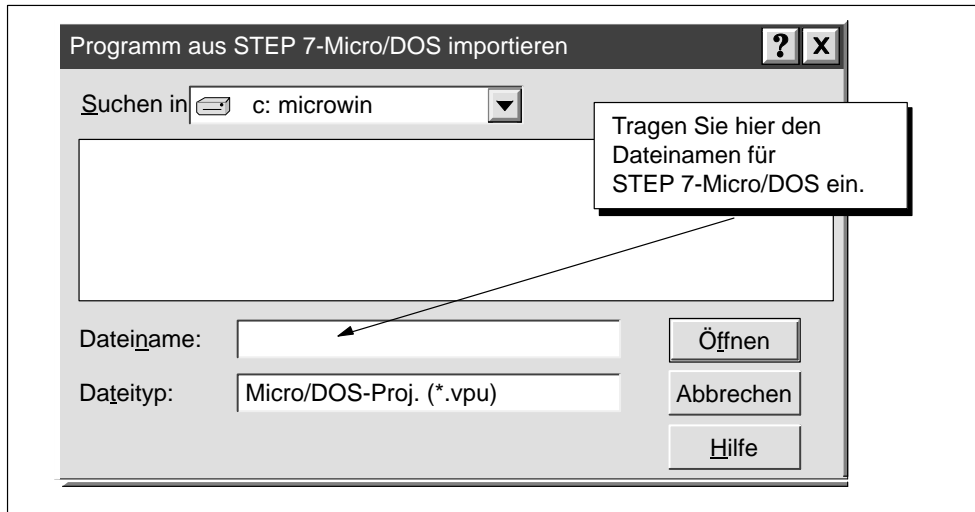


Bild E-3 Auswählen eines Programms aus STEP 7-Micro/DOS

Richtlinien und Einschränkungen beim Importieren

Wenn Sie eine Programmdatei STEP 7-Micro/DOS.vpu importieren, wird nach dem Speichern eine Kopie der folgenden Dateien aus STEP 7-Micro/DOS in das Format von STEP 7-Micro/WIN umgewandelt:

- Programmdateien
- Variablenspeicher und Daten
- Synonyme und Deskriptoren
- Statustabelle mit dem gleichen Namen wie das Projekt

Beim Importieren von Programmen aus STEP 7-Micro/DOS in ein Projekt in STEP 7-Micro/WIN geschieht folgendes:

- Konstanten, die im Variablenspeicher definiert wurden, bleiben erhalten.
- Synonyme aus STEP 7-Micro/DOS werden in Symbole in STEP 7-Micro/WIN umgewandelt und, sofern erforderlich, auf die maximale Länge von 23 Zeichen gekürzt. Die Deskriptoren der Synonyme, die bis zu 144 Zeichen umfassen können, werden in STEP 7-Micro/WIN auf eine maximale Länge von 79 Zeichen pro Symbolkommentar gekürzt.
- Netzwerkkommentare in STEP 7-Micro/DOS (maximal 16 Zeilen mit 60 Zeichen) bleiben in den AWL- und KOP-Editoren erhalten.
- Eine Statustabelle aus STEP 7-Micro/DOS, die den gleichen Namen wie ein Programm in STEP 7-Micro/DOS hat, wird in eine Statustabelle in STEP 7-Micro/WIN umgewandelt. Gibt es beispielsweise ein Programm mit dem Namen TEST.VPU und dazu die beiden Statustabellen TEST.CH2 und TEST2.CH2, dann wird die Statustabelle mit dem Namen TEST importiert, die Statustabelle mit dem Namen TEST2 wird dagegen nicht importiert.
- Netzadresse, Paßwort, Schutzstufe, Einstellungen der Ausgänge und remanente Bereiche werden entsprechend der Dateien aus STEP 7-Micro/DOS eingestellt. Sie können sich diese Parameter mit Hilfe des Menübefehls **CPU ► Konfigurieren...** anzeigen lassen.

Speichern des umgewandelten Programms

Wenn Sie das importierte Programm in dem gleichen Verzeichnis wie Ihre aktuellen Projekte in STEP 7-Micro/WIN speichern möchten, gehen Sie folgendermaßen vor:

1. Wählen Sie den Menübefehl **Projekt ► Speichern unter...** und wählen Sie im Dialogfeld "Speichern unter" das aktuelle Verzeichnis in STEP 7-Micro/WIN aus.
2. In dem Feld "Dateiname" geben Sie den Namen an, den Sie den importierten Programmdateien zuordnen möchten. Verwenden Sie die Erweiterung .prj.
3. Klicken Sie auf die Schaltfläche "OK".

Hinweis

Haben Sie das importierte Programm gespeichert oder geändert, dann können Sie es nicht wieder in das Format von STEP 7-Micro/DOS exportieren. Die ursprünglichen Dateien in STEP 7-Micro/DOS sind jedoch nicht geändert worden. Sie können Sie weiterhin in STEP 7-Micro/DOS verwenden.

Ausführungszeiten von AWL-Operationen

F

Auswirkungen des Signalflusses auf die Ausführungszeiten

Die Grundausführungszeiten der AWL-Operationen (siehe Tabelle F-4) zeigen die Zeiten, die für die Ausführung der Logik bzw. der Funktion der Operation erforderlich sind, wenn Signalfluß vorhanden ist (wenn der oberste Stackwert EIN bzw. "1" ist). Die Ausführung einiger Operationen ist davon abhängig, ob Signalfluß vorhanden ist, d.h. die CPU bearbeitet die Funktion nur dann, wenn der Signalfluß die Operation aktiviert (wenn der oberste Stackwert EIN bzw. "1" ist). Ist für die Operation kein Signalfluß vorhanden (der oberste Stackwert ist AUS bzw. "0"), dann müssen Sie die Ausführungszeiten "ohne Signalfluß" verwenden, um die Bearbeitungszeit für Ihr Programm zu berechnen. Tabelle F-1 zeigt die Ausführungszeit von AWL-Operationen ohne Signalfluß (wenn der oberste Stackwert AUS bzw. "0" ist) für die einzelnen S7-200 CPUs.

Tabelle F-1 Ausführungszeiten für Operationen ohne Signalfluß

Operation ohne Signalfluß	CPU 212	CPU 214/215/216
Alle AWL-Operationen	10 µs	6 µs

Auswirkungen der indirekten Adressierung auf die Ausführungszeiten

Die Grundausführungszeiten der AWL-Operationen (siehe Tabelle F-4) zeigen die Zeiten, die für die Ausführung der Operationen erforderlich sind, wenn Operanden oder Konstanten direkt adressiert werden. Verwenden Sie in Ihrem Programm indirekte Adressierung, müssen Sie pro indirekt adressiertem Operand den in Tabelle F-2 angegebenen Wert zu der Ausführungszeit addieren.

Tabelle F-2 Zusätzliche Zeit, die bei indirekter Adressierung addiert werden muß

Operation mit indirekter Adressierung	CPU 212	CPU 214/215/216
Alle Operationen mit Ausnahme von R, RI, S und SI	76 µs	47 µs
R, RI, S und SI	185,3 µs	120,2 µs

Auswirkungen von Analogein- und Analogausgängen auf die Ausführungszeiten

Das Zugreifen auf Analogeingänge und Analogausgänge wirkt sich auf die Ausführungszeiten einer Operation aus. Tabelle F-3 zeigt den Faktor, der zu der Grundausführungszeit einer Operation addiert werden muß, wenn die Operation auf einen Analogwert zugreift.

Tabelle F-3 Auswirkung von Analogein- und Analogausgängen auf die Ausführungszeiten

	Variante	CPU 212	CPU 214/215/216
Analogeingänge	EM231, EM235	171 µs	139 µs
Analogausgänge	EM232, EM235	99 µs	66 µs

Grundaussführungszeiten der AWL-Operationen

Tabelle F-4 führt die Grundaussführungszeiten der einzelnen AWL-Operationen bei den verschiedenen S7-200 CPUs auf.

Tabelle F-4 Ausführungszeiten für AWL-Operationen (in μ s)

Operation	Beschreibung	CPU 212 (in μ s)	CPU 214 (in μ s)	CPU 215 (in μ s)	CPU 216 (in μ s)
=	Grundaussführungszeit: E, A M SM, T, Z, V, S	1,2 4,8 6,0	0,8 3,2 4,0	0,8 3,2 4,0	0,8 3,2 4,0
+D	Grundaussführungszeit	143	95	95	95
-D	Grundaussführungszeit	144	96	96	96
+I	Grundaussführungszeit	110	73	73	73
-I	Grundaussführungszeit	111	74	74	74
=I	Grundaussführungszeit	63	42	42	42
+R	Grundaussführungszeit Maximale Ausführungszeit	- -	220 350	220 350	220 350
-R	Grundaussführungszeit Maximale Ausführungszeit	- -	225 355	225 355	225 355
*R	Grundaussführungszeit Maximale Ausführungszeit	- -	255 320	255 320	255 320
/R	Grundaussführungszeit Maximale Ausführungszeit	- -	810 870	810 870	810 870
A	Grundaussführungszeit: E, A M SM, T, Z, V, S	1,2 3,0 4,8	0,8 2,0 3,2	0,8 2,0 3,2	0,8 2,0 3,2
AE	Grundaussführungszeit	54	36	36	36
ATCH	Grundaussführungszeit	48	32	32	32
ATH	Gesamt = Grundaussführungszeit + (Länge) * (Längenmultiplikator) Grundaussführungszeit Längenmultiplikator (LM)	729 62	486 41	486 41	486 41
ATT	Grundaussführungszeit	-	268	268	268
BCDI	Grundaussführungszeit	249	166	166	166
BMB	Gesamt = Grundaussführungszeit + (Länge) * (LM) Grundaussführungszeit Längenmultiplikator (LM)	633 32	422 21	422 21	422 21
BMD	Gesamt = Grundaussführungszeit + (Länge) * (LM) Grundaussführungszeit Längenmultiplikator (LM)	- -	- -	446 43	446 43
BMW	Gesamt = Grundaussführungszeit + (Länge) * (LM) Grundaussführungszeit Längenmultiplikator (LM)	636 51	424 34	424 34	424 34
CALL	Grundaussführungszeit	35	23	23	23
CRET	Grundaussführungszeit	26	17	17	17

Tabelle F-4 Ausführungszeiten für AWL-Operationen (in μ s), Fortsetzung

Operation	Beschreibung	CPU 212 (in μ s)	CPU 214 (in μ s)	CPU 215 (in μ s)	CPU 216 (in μ s)
CRETI	Grundaussführungszeit	75	50	50	50
DECB	Grundaussführungszeit	-	-	37	37
DECD	Grundaussführungszeit	98	65	65	65
DECO	Grundaussführungszeit	84	56	56	56
DECW	Grundaussführungszeit	83	55	55	55
DISI	Grundaussführungszeit	36	24	24	24
DIV	Grundaussführungszeit	410	273	273	273
DTCH	Grundaussführungszeit	39	26	26	26
DTR	Grundaussführungszeit Maximale Ausführungszeit	-	108 135	108 135	108 135
ED	Grundaussführungszeit	32	21	21	21
EIN	Grundaussführungszeit: E, A M SM, T, Z, V, S	1,2 3,0 4,8	0,8 2,0 3,2	0,8 2,0 3,2	0,8 2,0 3,2
ENCO	Minimale Ausführungszeit Maximale Ausführungszeit	75 93	50 62	50 62	50 62
END	Grundaussführungszeit	1.8	1.2	1.2	1.2
ENI	Grundaussführungszeit	36	24	24	24
EU	Grundaussführungszeit	32	21	21	21
FIFO	Gesamt = Grundaussführungszeit + (LM) * (Länge) Grundaussführungszeit Längenmultiplikator (LM)	-	234 29	234 29	234 29
FILL	Gesamt = Grundaussführungszeit + (LM) * (Länge) Grundaussführungszeit Längenmultiplikator (LM)	578 18	385 12	385 12	385 12
FND<	Gesamt = Grundaussführungszeit + (LM) * (Länge) Grundaussführungszeit Längenmultiplikator (LM)	-	424 28	424 28	424 28
FND<>	Gesamt = Grundaussführungszeit + (LM) * (Länge) Grundaussführungszeit Längenmultiplikator (LM)	-	423 29	423 29	423 29
FND=	Gesamt = Grundaussführungszeit + (LM) * (Länge) Grundaussführungszeit Längenmultiplikator (LM)	-	431 25	431 25	431 25
FND>	Gesamt = Grundaussführungszeit + (LM) * (Länge) Grundaussführungszeit Längenmultiplikator (LM)	-	428 28	428 28	428 28
FOR	Gesamt = Grundaussführungszeit + (LM) * (Anzahl der Wiederholungen) Grundaussführungszeit Schleifenmultiplikator (LM)	-	135 129	135 129	135 129
HDEF	Grundaussführungszeit	80	53	53	53

Tabelle F-4 Ausführungszeiten für AWL-Operationen (in µs), Fortsetzung

Operation	Beschreibung	CPU 212 (in µs)	CPU 214 (in µs)	CPU 215 (in µs)	CPU 216 (in µs)
HSC	Grundausführungszeit	101	67	67	67
HTA	Gesamt = Grundausführungszeit + (LM) * (Länge)	714	476	476	476
	Grundausführungszeit Längenmultiplikator (LM)	35	23	23	23
IBCD	Grundausführungszeit	186	124	124	124
INCB	Grundausführungszeit	-	-	34	34
INCD	Grundausführungszeit	96	64	64	64
INCW	Grundausführungszeit	81	54	54	54
INT	Typische Ausführungszeit mit einem Interrupt	180	120	120	120
INVB	Grundausführungszeit	-	-	40	40
INVD	Grundausführungszeit	99	66	66	66
INVW	Grundausführungszeit	84	56	56	56
JMP	Grundausführungszeit	1,2	0,8	0,8	0,8
LBL	Grundausführungszeit	0	0	0	0
LD	Grundausführungszeit: E, A	1,2	0,8	0,8	0,8
	M	3,0	2,0	2,0	2,0
	SM, T, Z, V, S	4,8	3,2	3,2	3,2
LDB<=	Ausführungszeit, wenn Vergleich wahr ist	63	42	42	42
	Ausführungszeit, wenn Vergleich falsch ist	66	44	44	44
LDB=	Ausführungszeit, wenn Vergleich wahr ist	63	42	42	42
	Ausführungszeit, wenn Vergleich falsch ist	66	44	44	44
LDB>=	Ausführungszeit, wenn Vergleich wahr ist	63	42	42	42
	Ausführungszeit, wenn Vergleich falsch ist	66	44	44	44
LDD<=	Ausführungszeit, wenn Vergleich wahr ist	135	90	90	90
	Ausführungszeit, wenn Vergleich falsch ist	138	92	92	92
LDD=	Ausführungszeit, wenn Vergleich wahr ist	135	90	90	90
	Ausführungszeit, wenn Vergleich falsch ist	138	92	92	92
LDD>=	Ausführungszeit, wenn Vergleich wahr ist	135	90	90	90
	Ausführungszeit, wenn Vergleich falsch ist	138	92	92	92
LDI	Grundausführungszeit	50	33	33	33
LDN	Grundausführungszeit: E, A	1,8	1,2	1,2	1,2
	M	3,6	2,4	2,4	2,4
	SM, T, Z, V, S	5,4	3,6	3,6	3,6
LDNI	Grundausführungszeit	50	33	33	33
LDR=	Grundausführungszeit	-	98	98	98
LDR<=	Grundausführungszeit	-	98	98	98
LDR>=	Grundausführungszeit	-	98	98	98
LDW<=	Ausführungszeit, wenn Vergleich wahr ist	108	72	72	72
	Ausführungszeit, wenn Vergleich falsch ist	111	74	74	74
LDW=	Ausführungszeit, wenn Vergleich wahr ist	108	72	72	72
	Ausführungszeit, wenn Vergleich falsch ist	111	74	74	74

Tabelle F-4 Ausführungszeiten für AWL-Operationen (in µs), Fortsetzung

Operation	Beschreibung	CPU 212 (in µs)	CPU 214 (in µs)	CPU 215 (in µs)	CPU 216 (in µs)
LDW>=	Ausführungszeit, wenn Vergleich wahr ist Ausführungszeit, wenn Vergleich falsch ist	108 111	72 74	72 74	72 74
LIFO	Grundausführungszeit	-	261	261	261
LPP	Grundausführungszeit	0.6	0.4	0.4	0.4
LPS	Grundausführungszeit	1.2	0.8	0.8	0.8
LRD	Grundausführungszeit	0.6	0.4	0.4	0.4
LSCR	Grundausführungszeit	18	12	12	12
MEND	Grundausführungszeit	1.2	0.8	0.8	0.8
MOVB	Grundausführungszeit	45	30	30	30
MOVD	Grundausführungszeit	81	54	54	54
MOVR	Grundausführungszeit	81	54	54	54
MOVW	Grundausführungszeit	66	44	44	44
MUL	Grundausführungszeit	210	140	140	140
NETR	Grundausführungszeit	-	478	478	478
NETW	Gesamt = Grundausführungszeit + (LM) * (Länge) Grundausführungszeit Längenmultiplikator (LM)	-	460 16,8	460 16,8	460 16,8
NEXT	Grundausführungszeit	-	0	0	0
NOP	Grundausführungszeit	0	0	0	0
NOT	Grundausführungszeit	1,2	0,8	0,8	0,8
O	Grundausführungszeit: E, A M SM, T, Z, V, S	1,2 3,0 4,8	0,8 2,0 3,2	0,8 2,0 3,2	0,8 2,0 3,2
OB<=	Ausführungszeit, wenn Vergleich wahr ist Ausführungszeit, wenn Vergleich falsch ist	65 68	43 45	43 45	43 45
OB=	Ausführungszeit, wenn Vergleich wahr ist Ausführungszeit, wenn Vergleich falsch ist	65 68	43 45	43 45	43 45
OB>=	Ausführungszeit, wenn Vergleich wahr ist Ausführungszeit, wenn Vergleich falsch ist	65 68	43 45	43 45	43 45
OD<=	Ausführungszeit, wenn Vergleich wahr ist Ausführungszeit, wenn Vergleich falsch ist	138 140	92 93	92 93	92 93
OD=	Ausführungszeit, wenn Vergleich wahr ist Ausführungszeit, wenn Vergleich falsch ist	138 140	92 93	92 93	92 93
OD>=	Ausführungszeit, wenn Vergleich wahr ist Ausführungszeit, wenn Vergleich falsch ist	138 140	92 93	92 93	92 93
OI	Grundausführungszeit	54	36	36	36
OLD	Grundausführungszeit	1,2	0,8	0,8	0,8
ONI	Grundausführungszeit	54	36	36	36
OR=	Grundausführungszeit	-	98	98	98
OR<=	Grundausführungszeit	-	98	98	98

Tabelle F-4 Ausführungszeiten für AWL-Operationen (in μs), Fortsetzung

Operation	Beschreibung	CPU 212 (in μs)	CPU 214 (in μs)	CPU 215 (in μs)	CPU 216 (in μs)
OR>=	Grundaussführungszeit	-	98	98	98
ORB	Grundaussführungszeit	-	-	49	49
ORD	Grundaussführungszeit	137	91	91	91
ORW	Grundaussführungszeit	110	73	73	73
OW<=	Ausführungszeit, wenn Vergleich wahr ist Ausführungszeit, wenn Vergleich falsch ist	108 111	72 74	72 74	72 74
OW=	Ausführungszeit, wenn Vergleich wahr ist Ausführungszeit, wenn Vergleich falsch ist	108 111	72 74	72 74	72 74
OW>=	Ausführungszeit, wenn Vergleich wahr ist Ausführungszeit, wenn Vergleich falsch ist	108 111	72 74	72 74	72 74
PID	Grundaussführungszeit Addierer zum Berechnen von $(K_c * T_s/T_i)$ und $(K_c * T_d/T_s)$ vor der Berechnung des PID-Reglers. Es wird erneut gerechnet, wenn sich der Wert von K_c , T_s , T_i oder T_s nach der vorherigen Ausführung der Operation geändert hat oder wenn in den automatischen Betrieb gewechselt wurde.	- -	- -	2000 2600	2000 2600
PLS	Grundaussführungszeit	-	153	153	153
R	Gesamt = Zeit für Operanden + (LM) * (Länge) Ausführungszeit für Zähler Ausführungszeit für Zeit Ausführungszeit für anderen Operanden Längenmultiplikator für Zähler (LM) Längenmultiplikator für Zeit (LM) Längenmultiplikator für anderen Operanden (LM) Handelt es sich bei der Länge um eine Variable anstelle einer Konstanten, erhöhen Sie die Grundaussführungszeit durch Addieren von:	33,9 32,9 39,9 28,8 49,7 5,6 109,8	23 21 27 19,2 33,1 3,7 73,2	23 22 27 19,2 33,1 3,7 73,2	23 22 27 19,2 33,1 3,7 73,2
RCV	Grundaussführungszeit	-	-	126	126
RET	Grundaussführungszeit	27	18	18	18
RETI	Grundaussführungszeit	75	50	50	50
RI	Gesamt = Grundaussführungszeit + (LM) * (Länge) Grundaussführungszeit Längenmultiplikator (LM) Handelt es sich bei der Länge um eine Variable anstelle einer Konstanten, erhöhen Sie die Grundaussführungszeit durch Addieren von:	31,5 60 110	21 40 73	21 40 73	21 40 73
RLB	Gesamt = Grundaussführungszeit + (LM) * (Länge) Grundaussführungszeit Längenmultiplikator (LM)	- -	- -	62 1,2	62 1,2
RLD	Gesamt = Grundaussführungszeit + (LM) * (Länge) Grundaussführungszeit Längenmultiplikator (LM)	129 10,7	86 7,1	86 7,1	86 7,1

Tabelle F-4 Ausführungszeiten für AWL-Operationen (in μ s), Fortsetzung

Operation	Beschreibung	CPU 212 (in μ s)	CPU 214 (in μ s)	CPU 215 (in μ s)	CPU 216 (in μ s)
RLW	Gesamt = Grundaussführungszeit + (LM) * (Länge) Grundaussführungszeit Längenmultiplikator (LM)	116 6,9	77 4,6	77 4,6	77 4,6
RRB	Gesamt = Grundaussführungszeit + (LM) * (Länge) Grundaussführungszeit Längenmultiplikator (LM)	- -	- -	62 1,2	62 1,2
RRD	Gesamt = Grundaussführungszeit + (LM) * (Länge) Grundaussführungszeit Längenmultiplikator (LM)	135 10,4	90 6,9	90 6,9	90 6,9
RRW	Gesamt = Grundaussführungszeit + (LM) * (Länge) Grundaussführungszeit Längenmultiplikator (LM)	117 6,6	78 4,4	78 4,4	78 4,4
S	Gesamt = Grundaussführungszeit + (LM) * (Länge) Grundaussführungszeit Längenmultiplikator (LM) Handelt es sich bei der Länge um eine Variable an- stelle einer Konstanten, erhöhen Sie die Grundauss- führungszeit durch Addieren von:	38 5,6 110	25 3,7 74	25 3,7 74	25 3,7 74
SBR	Grundaussführungszeit	0	0	0	0
SCRE	Grundaussführungszeit	0	0	0	0
SCRT	Grundaussführungszeit	31	21	21	21
SEG	Grundaussführungszeit	47	31	31	31
SHRB	Gesamt = Grundaussführungszeit + (LM) * (Länge) Grundaussführungszeit Längenmultiplikator (LM)	449 2,3	299 1,5	299 1,5	299 1,5
SI	Gesamt = Grundaussführungszeit + (LM) * (Länge) Grundaussführungszeit Längenmultiplikator (LM) Handelt es sich bei der Länge um eine Variable an- stelle einer Konstanten, erhöhen Sie die Grundauss- führungszeit durch Addieren von:	32 58 110	21 38 73	21 38 73	21 38 73
SLB	Gesamt = Grundaussführungszeit + (LM) * (Länge) Grundaussführungszeit Längenmultiplikator (LM)	- -	- -	64 1,6	64 1,6
SLD	Gesamt = Grundaussführungszeit + (LM) * (Länge) Grundaussführungszeit Längenmultiplikator (LM)	131 8,9	87 5,9	87 5,9	87 5,9
SLW	Gesamt = Grundaussführungszeit + (LM) * (Länge) Grundaussführungszeit Längenmultiplikator (LM)	119 5,1	79 3,4	79 3,4	79 3,4
SQRT	Grundaussführungszeit Maximale Ausführungszeit	-	1830 2110	1830 2110	1830 2110
SRB	Gesamt = Grundaussführungszeit + (LM) * (Länge) Grundaussführungszeit Längenmultiplikator (LM)	- -	- -	64 1,6	64 1,6

Tabelle F-4 Ausführungszeiten für AWL-Operationen (in µs), Fortsetzung

Operation	Beschreibung	CPU 212 (in µs)	CPU 214 (in µs)	CPU 215 (in µs)	CPU 216 (in µs)
SRD	Gesamt = Grundausführungszeit + (LM) * (Länge) Grundausführungszeit Längenmultiplikator (LM)	137 8,6	91 5,7	91 5,7	91 5,7
SRW	Gesamt = Grundausführungszeit + (LM) * (Länge) Grundausführungszeit Längenmultiplikator (LM)	120 5,0	80 3,3	80 3,3	80 3,3
STOP	Grundausführungszeit	13	9	9	9
SWAP	Grundausführungszeit	65	43	43	43
TODR	Grundausführungszeit	-	282	282	282
TODW	Grundausführungszeit	-	489	489	489
TON	Grundausführungszeit	48	32	32	32
TONR	Grundausführungszeit	74	49	49	49
TRUNC	Grundausführungszeit Maximale Ausführungszeit	-	258 420	258 420	258 420
UB<=	Ausführungszeit, wenn Vergleich wahr ist Ausführungszeit, wenn Vergleich falsch ist	65 68	43 45	43 45	43 45
UB=	Ausführungszeit, wenn Vergleich wahr ist Ausführungszeit, wenn Vergleich falsch ist	65 68	43 45	43 45	43 45
UB>=	Ausführungszeit, wenn Vergleich wahr ist Ausführungszeit, wenn Vergleich falsch ist	65 68	43 45	43 45	43 45
UD<=	Ausführungszeit, wenn Vergleich wahr ist Ausführungszeit, wenn Vergleich falsch ist	137 140	91 93	91 93	91 93
UD=	Ausführungszeit, wenn Vergleich wahr ist Ausführungszeit, wenn Vergleich falsch ist	137 140	91 93	91 93	91 93
UD>=	Ausführungszeit, wenn Vergleich wahr ist Ausführungszeit, wenn Vergleich falsch ist	137 140	91 93	91 93	91 93
ULD	Grundausführungszeit	1,2	0,8	0,8	0,8
UN	Grundausführungszeit: E, A M SM, T, Z, V, S	1,2 3,0 4,8	0,8 2,0 3,2	0,8 2,0 3,2	0,8 2,0 3,2
UNDB	Grundausführungszeit	-	-	49	49
UNDD	Grundausführungszeit	137	91	91	91
UNDW	Grundausführungszeit	110	73	73	73
UNI	Grundausführungszeit	54	36	36	36
UR=	Grundausführungszeit	-	98	98	98
UR<=	Grundausführungszeit	-	98	98	98
UR>=	Grundausführungszeit	-	98	98	98
UW<=	Ausführungszeit, wenn Vergleich wahr ist Ausführungszeit, wenn Vergleich falsch ist	110 113	73 75	73 75	73 75
UW=	Ausführungszeit, wenn Vergleich wahr ist Ausführungszeit, wenn Vergleich falsch ist	110 113	73 75	73 75	73 75

Tabelle F-4 Ausführungszeiten für AWL-Operationen (in μ s), Fortsetzung

Operation	Beschreibung	CPU 212 (in μs)	CPU 214 (in μs)	CPU 215 (in μs)	CPU 216 (in μs)
UW>=	Ausführungszeit, wenn Vergleich wahr ist	110	73	73	73
	Ausführungszeit, wenn Vergleich falsch ist	113	75	75	75
WDR	Grundausführungszeit	21	14	14	14
XMT	Grundausführungszeit	272	181	181	181
XORB	Grundausführungszeit	-	-	49	49
XORD	Grundausführungszeit	137	91	91	91
XORW	Grundausführungszeit	110	73	73	73
ZV	Grundausführungszeit	78	52	52	52
ZVR	Grundausführungszeit	105	70	70	70

S7-200 Bestellnummern



CPU	Bestellnummer
CPU 212, DC-Versorgung, DC-Eingänge, DC-Ausgänge	6ES7 212-1AA01-0XB0
CPU 212, AC-Versorgung, DC-Eingänge, Relaisausgänge	6ES7 212-1BA01-0XB0
CPU 212, AC-Versorgung, AC-Eingänge, AC-Ausgänge	6ES7 212-1CA01-0XB0
CPU 212, AC-Versorgung, DC-Eingänge (stromliefernd), Relaisausgänge	6ES7 212-1BA10-0XB0
CPU 212, AC-Versorgung, 24-V-AC-Eingänge, AC-Ausgänge	6ES7 212-1DA01-0XB0
CPU 212, 24-V-AC-Versorgung, DC-Eingänge, Relaisausgänge	6ES7 212-1FA01-0XB0
CPU 212, AC-Versorgung, AC-Eingänge, Relaisausgänge	6ES7 212-1GA01-0XB0
CPU 214, DC-Versorgung, DC-Eingänge, DC-Ausgänge	6ES7 214-1AC01-0XB0
CPU 214, AC-Versorgung, DC-Eingänge, Relaisausgänge	6ES7 214-1BC01-0XB0
CPU 214, AC-Versorgung, AC-Eingänge, AC-Ausgänge	6ES7 214-1CC01-0XB0
CPU 214, AC-Versorgung, DC-Eingänge (stromliefernd), Relaisausgänge	6ES7 214-1BC10-0XB0
CPU 214, AC-Versorgung, 24-V-AC-Eingänge, AC-Ausgänge	6ES7 214-1DC01-0XB0
CPU 214, AC-Versorgung, AC-Eingänge, Relaisausgänge	6ES7 214-1GC01-0XB0
CPU 215, DC-Versorgung, DC-Eingänge, DC-Ausgänge	6ES7 215-2AD00-0XB0
CPU 215, AC-Versorgung, DC-Eingänge, Relaisausgänge	6ES7 215-2BD00-0XB0
CPU 216, DC-Versorgung, DC-Eingänge, DC-Ausgänge	6ES7 216-2AD00-0XB0
CPU 216, AC-Versorgung, DC-Eingänge, Relaisausgänge	6ES7 216-2BD00-0XB0

Erweiterungsmodul	Bestellnummer
Erweiterungsmodul EM221, Digitaleingabe 8 x 24 V DC	6ES7 221-1BF00-0XA0
Erweiterungsmodul EM221, Digitaleingabe 8 x 120 V AC	6ES7 221-1EF00-0XA0
Erweiterungsmodul EM221, Digitaleingabe 8 x 24 V DC	6ES7 221-1BF10-0XA0
Erweiterungsmodul EM221, Digitaleingabe 8 x 24 V AC	6ES7 221-1JF00-0XA0
Erweiterungsmodul EM222, Digitalausgabe 8 x 24 V DC	6ES7 222-1BF00-0XA0
Erweiterungsmodul EM222, Digitalausgabe 8 x Relais	6ES7 222-1HF00-0XA0
Erweiterungsmodul EM222, Digitalausgabe 8 x 120/230 V AC	6ES7 222-1EF00-0XA0
Erweiterungsmodul EM223, Digitalein-/Digitalausgabe 4 x 24-V-DC-Eingang / 4 x 24-V-DC-Ausgang	6ES7 223-1BF00-0XA0
Erweiterungsmodul EM223, Digitalein-/Digitalausgabe 4 x 24-V-DC-Eingang / 4 x Relaisausgang	6ES7 223-1HF00-0XA0

Erweiterungsmodul	Bestellnummer
Erweiterungsmodul EM 223, Digitalein-/Digitalausgabe 4 x 120-V-AC-Eingang / 4 x 120/230-V-AC-Ausgang	6ES7 223-1EF00-0XA0
Erweiterungsmodul EM223, Digitalein-/Digitalausgabe 8 x 24-V-DC-Eingang / 8 x Relaisausgang	6ES7 223-1PH00-0XA0
Erweiterungsmodul EM223, Digitalein-/Digitalausgabe 8 x 24-V-DC-Eingang / 8 x 24-V-DC-Ausgang	6ES7 223-1BH00-0XA0
Erweiterungsmodul EM223, Digitalein-/Digitalausgabe 16 x 24-V-DC-Eingang / 16 x Relaisausgang	6ES7 223-1PL00-0XA0
Erweiterungsmodul EM223, Digitalein-/Digitalausgabe 16 x 24-V-DC-Eingang / 16 x 24-V-DC-Ausgang	6ES7 223-1BL00-0XA0
Erweiterungsmodul EM 231, Analogeingabe AE 3 x 12 Bit	6ES7 231-0HC00-0XA0
Erweiterungsmodul EM232, Analogausgabe AA 2 x 12 Bit	6ES7 232-0HB00-0XA0
Erweiterungsmodul EM 235, Analogein-/Analogausgabe AE 3/AA 1 x 12 Bit	6ES7 235-0KD00-0XA0
CP 242-2 AS-Interface Mastermodul für S7-200	6GK7 242-2AX00-0XA0

Kabel, Netzverbinder, Busverstärker	Bestellnummer
Steckleitung für Erweiterungsmodule	6ES7 290-6BC50-0XA0
MPI-Kabel	6ES7 901-0BF00-0AA0
PC/PPI-Kabel	6ES7 901-3BF00-0XA0
PROFIBUS-Netzkabel	6XV1 830-0AH10
Busanschlußstecker mit Stecker für die Programmierschnittstelle, axialer Kabelabgang	6ES7 972-0BB10-0XA0
Busanschlußstecker ohne Stecker für die Programmierschnittstelle, axialer Kabelabgang	6ES7 972-0BA10-0XA0
RS 485-Busverbinder mit axialem Kabelabgang	6GK1 500-0EA00
RS 485-Busverbinder mit 30°-Kabelabgang	6ES7 972-0BA30-0XA0
RS 485 IP 20 Busverstärker	6ES7 972-0AA00-0XA0

Kommunikationsbaugruppen	Bestellnummer
MPI-Baugruppe: Short AT ISA	6ES7 793-2AA01-0AA0
CP 5411: Short AT ISA	6GK1 541-1AA00
CP 5511: PCMCIA, Typ II, Plug & Play-Hardware	6GK1 551-1AA00
CP 5611: Short PCI, Plug & Play-Hardware	6GK1 561-1AA00

Bedienerschnittstellen	Bestellnummer
Textdisplay TD 200	6ES7 272-0AA00-0YA0
Operator Panel OP3	6AV3 503-1DB10
Operator Panel OP7	6AV3 607-IJC20-0AX0
Operator Panel OP17	6AV3 617-IJC20-0AX0

Allgemein	Bestellnummer
Speichermodul 8 K x 8	6ES7 291-8GC00-0XA0
Speichermodul 16 K x 8	6ES7 291-8GD00-0XA0
Batterieminodul	6ES7 291-8BA00-0XA0
Erdungsklemmen für Hutschiene	6ES5 728-8MAll
Klemmenblock mit 12 Anschlußklemmen für Feldverdrahtung (CPU 212/215/216) 10 Stück	6ES7 290-2AA00-0XA0
Klemmenblock mit 14 Anschlußklemmen für Feldverdrahtung (CPU 215/216 und Erweiterungsmodul) 10 Stück	6ES7 290-2CA00-0XA0
Klemmenblock mit 18 Anschlußklemmen für Feldverdrahtung (CPU 214) 10 Stück	6ES7 290-2BA00-0XA0
DC-Eingangssimulator für die CPU 212	6ES7 274-1XF00-0XA0
DC-Eingangssimulator für die CPU 214	6ES7 274-1XH00-0XA0
DC-Eingangssimulator für die CPU 215/216	6ES7 274-1XK00-0XA0

Programmiersoftware	Bestellnummer
STEP 7-Micro/WIN 16 (V 2.1) Einzellizenz	6ES7 810-2AA01-0YX0
STEP 7-Micro/WIN 16 (V 2.1) Kopierlizenz	6ES7 810-2AA01-0YX1
STEP 7-Micro/WIN 16 (V 2.1) Update	6ES7 810-2AA01-0YX3
STEP 7-Micro/WIN 32 (V 2.1) Einzellizenz	6ES7 810-2AA11-0YX0
STEP 7-Micro/WIN 32 (V 2.1) Kopierlizenz	6ES7 810-2AA11-0YX1
STEP 7-Micro/WIN 32 (V 2.1) Update	6ES7 810-2AA11-0YX3
STEP 7-Micro/DOS Einzellizenz	6ES7 810-2DA00-0YX0

Handbücher	Bestellnummer
Dezentrales Peripheriegerät ET 200, Systemhandbuch	6ES5 998-3ES22
Programmiergerät PG 702, Handbuch	6ES7 702-0AA00-8BA0
Textdisplay TD 200, Benutzerhandbuch	6ES7 272 0AA00-8BA0
CP242-2 AS-Interface Mastermodul, Handbuch	6GK7 242-2AX00-8BA0
STEP 7-Micro/DOS, Benutzerhandbuch	6ES7 810-2DA10-8BA0

H

S7-200 Fehlerbehebung

Tabelle H-1 S7-200 Fehlerbehebung

Fehler	Mögliche Ursachen	Abhilfe
Die Ausgänge arbeiten nicht mehr.	Im gesteuerten Gerät ist Überspannung aufgetreten, wodurch der Ausgang beschädigt wurde.	Beim Anschließen an induktive Lasten (z.B. Motoren oder Relais) müssen entsprechende Schutzbeschaltungen eingesetzt werden. Ausführliche Informationen hierzu entnehmen Sie dem Abschnitt 2.4.
Die Anzeige SF (Systemfehler) auf der CPU leuchtet auf.	Folgende Liste führt die häufigsten Ursachen auf: <ul style="list-style-type: none"> • Fehler im Anwenderprogramm <ul style="list-style-type: none"> – 0003 Zeitüberwachungsfehler – 0011 Indirekte Adressierung – 0012 Unzulässiger Vergleich • Elektrische Störungen <ul style="list-style-type: none"> – 0001 bis 0009 • Defektes Gerät <ul style="list-style-type: none"> – 0001 bis 0010 	Lesen Sie den Fehlercode des schweren Fehlers und entnehmen Sie weitere Hinweise aus Anhang C.1: <ul style="list-style-type: none"> • Bei Programmierfehlern prüfen Sie die Verwendung der Operationen FOR, NEXT, JMP, LBL und CMP. • Bei elektrischen Störungen: <ul style="list-style-type: none"> – Lesen Sie die Verdrahtungsrichtlinien in Abschnitt 2.3. Es ist äußerst wichtig, daß die Schalttafel geerdet ist und daß die Hochspannungsleitungen nicht parallel zu Niederspannungsleitungen verlaufen. – Erden Sie die Klemme M der 24-V-DC-Geberversorgung.
Die Werte der Analogeingänge sind in jedem Zyklus unterschiedlich, obwohl das Eingangssignal konstant bleibt.	Hierfür gibt es verschiedene Ursachen: <ul style="list-style-type: none"> • Elektrische Störungen von der Spannungsversorgung • Elektrische Störungen am Eingangssignal • Schlechte Erdung • Die Formatierung des ausgegebenen Werts ist anders als erwartet • Bei dem Modul handelt es sich um ein schnelles Modul, bei dem kein Filter für 50/60 Hz vorhanden ist 	<ul style="list-style-type: none"> • Der vom Modul ausgegebene Wert ist ungefiltert. Sie können ein einfaches Filterprogramm in das Anwenderprogramm einbauen. Ausführliche Informationen zum Analogeingabefilter-Assistenten entnehmen Sie Kapitel 5. • Vergleichen Sie die tatsächliche Wiederholbarkeit des vom Modul ausgegebenen Werts mit den technischen Daten in Anhang A. Die Module der S7-200 geben ungefilterte, linksbündige Werte aus. Dies bedeutet, daß jede Variation des Schritts von 1 Einheit den Wert im S7-200 Modul um einen Schritt von 8 erhöht. • Um die Ursache der elektrischen Störung zu ermitteln, können Sie versuchen, einen unbenutzten Analogeingang kurzzuschließen. Variiert der Wert aus dem kurzgeschlossenen Eingang genauso wie der Eingang der Geberversorgung, dann liegt die Ursache der Störungen in den Leitungen. Ansonsten verursachen die Geberversorgung bzw. deren Leitungen die Störung. <ul style="list-style-type: none"> – Bei Störung in den Leitungen der Geberversorgung entnehmen Sie den Installationsrichtlinien ausführliche Informationen zum EM231 (Anhang A.33) oder zum EM235 (Anhang A.35). – Bei Störung in der Spannungsversorgung entnehmen Sie den Verdrahtungsrichtlinien in Abschnitt 2.3 ausführliche Informationen. Sie können auch versuchen, die Klemmen M des Analogmoduls und die Geberversorgung der CPU zu erden.

Tabelle H-1 S7-200 Fehlerbehebung, Fortsetzung

Fehler	Mögliche Ursachen	Abhilfe
Defekte Spannungsversorgung	Überspannung in den Leitungen, die das Gerät speisen.	Schließen Sie ein Meßgerät an das System an, um die Größe und Dauer der Überspannungsspitzen zu prüfen. Mit Hilfe dieser Informationen schließen Sie einen entsprechenden Ableiter an Ihr System an. Ausführliche Informationen zum Installieren der Feldverdrahtung entnehmen Sie den Verdrahtungsrichtlinien in Abschnitt 2.3.
Elektrische Störung	<ul style="list-style-type: none"> • Schlechte Erdung • Weiterleitung in der Verdrahtung innerhalb des Schaltschranks 	Lesen Sie die Verdrahtungsrichtlinien in Abschnitt 2.3. Es ist äußerst wichtig, daß die Schalttafel geerdet ist und daß die Hochspannungsleitungen nicht parallel zu Niederspannungsleitungen verlaufen. Erden Sie die Klemme M der 24-V-DC-Geberversorgung.
Zwischenwerte aus den Erweiterungsmodulen	Übermäßige Schwingungen	Die Grenzwerte für sinusförmige Schwingungen entnehmen Sie dem Anhang A.1.
	Unsachgemäße Montage der Standard-Hutschiene (DIN)	Ist Ihr System auf einer Standard-Hutschiene (DIN) montiert, entnehmen Sie dem Abschnitt 2.2 ausführliche Informationen hierzu.
	Die Kunststoffreste wurden nach Herausbrechen der Abdeckung für den Buserweiterungsanschluß nicht vollständig entfernt.	Informationen zum Montieren der Erweiterungsmodule entnehmen Sie dem Abschnitt 2.2.
	Defekter Busverbinder	Tauschen Sie den Busverbinder aus.
Das Kommunikationsnetz ist beim Anschließen eines externen Geräts nicht mehr funktionsfähig. (Entweder ist die Schnittstelle am Computer, die Schnittstelle am Automatisierungssystem oder das PC/PPI-Kabel defekt.)	Die Schnittstelle RS-485 der S7-200 CPU und das PC/PPI-Kable sind nicht galvanisch getrennt (sofern im Datenblatt nicht anders angegeben). Das Kommunikationskabel kann zu einem unerwünschten Strompfad werden, wenn die Geräte, die nicht galvanisch getrennt sind (wie Automatisierungssysteme, Computer oder andere Geräte) und an das Kabel angeschlossen sind, nicht den gleichen Bezugsleiter im Stromkreis haben. Diese unerwünschten Ströme können Kommunikationsfehler verursachen oder Schäden in den Stromkreisen hervorrufen.	<ul style="list-style-type: none"> • Ausführliche Informationen hierzu entnehmen Sie den Verdrahtungsrichtlinien in Abschnitt 2.3 und den Richtlinien für Kommunikationsnetze in Kapitel 9. • Verwenden Sie einen galvanisch getrennten RS-485/RS-232-Adapter (nicht von Siemens erhältlich) anstelle des PC/PPI-Kabels. • Verwenden Sie einen galvanisch getrennten RS-485/RS-232-Busverstärker, wenn Sie Geräte anschließen, die nicht den gleichen Bezugsleiter im Stromkreis haben.
Kommunikationsfehler in STEP 7-Micro/WIN		Ausführliche Informationen zur Kommunikation im Netz entnehmen Sie dem Kapitel 9.
Fehlerbehebung		Informationen zu Fehlercodes entnehmen Sie dem Anhang C.

Index

A

- Ablaufsteuerungsrelais
 - Adressierung, 7-4
 - CPU 212/214/215/216, 10-2
- Abmessungen
 - Batteriemodul, A-80
 - CPU 212, 2-3
 - CPU 214, 2-3
 - CPU 215, 2-4
 - CPU 216, 2-4
 - Erweiterungsmodule, 2-4
 - PC/PPI-Kabel, A-83
 - Schraubengröße für den Einbau, 2-3–2-5
 - Speichermodul, A-78
- Abschließen, Netz, 9-7
- Abschluß, Netz, 9-7
- Abtasten von Analogeingängen, 5-14–5-16
- AC-Aufbau, Richtlinien, 2-10
- AC-Ausgänge, 2-14
- Adapter, Null-Modem, 3-19–3-20, 9-12
- Adressen
 - Absolute, 6-4
 - MPI-Kommunikation, 3-17
 - Statustabelle, 3-35
 - Symbolische, 6-4
 - Überwachen, 5-17, 5-18
- Adreßformat für Bytes, 7-2
- Adressierung
 - Ablaufsteuerungsrelais, 7-4
 - Akkumulatoren, 7-6
 - Analogausgänge, 7-6
 - Analogeingänge, 7-6
 - Arbeiten mit symbolischer Adressierung, 3-36
 - Bereich, Anzeigen, 5-18
 - Byte.Bit-Adresse, 7-2
 - Erweiterte Ein- und Ausgänge, 8-2
 - Indirekt (Pointer), 7-9–7-11
 - & und *, 7-9
 - Ändern eines Pointers, 7-10
 - Integrierte Ein- und Ausgänge, 8-2
 - Prozeßabbild der Ausgänge, 7-3
 - Prozeßabbild der Eingänge, 7-3
 - Schnelle Zähler, 7-7
 - Sondermerker, 7-4
 - Speicherbereich der Merker, 7-3
 - Speicherbereiche, 7-2
 - Teilnehmer im Netz, 9-2
 - Variablenspeicher, 7-3
 - Verwendete Elemente, 5-18
 - Zähler, 7-5
 - Zeiten, 7-4
- Akkumulatoren, Adressierung, 7-6
- Aktualisieren, Zeiten, 10-14
- Aktualisierungsrate für die Anzeige, Auswählen, 5-5
- Aktuelle Werte von Zeiten, Aktualisieren, 10-16
- Algorithmus für den PID-Regelkreis, 10-55–10-59
- Alle Interruptereignisse freigeben, 10-116
- Alle Interruptereignisse sperren, 10-116
- Amerikanischer Aufbau, Richtlinien, 2-12
- Analogausgänge
 - Adressierung, 7-6
 - Zugreifen, 6-11
- Analogein- und Analogausgänge, Auswirkungen auf die Ausführungszeiten, F-1
- Analogeingabefilter-Assistent, 5-14–5-16

- Analogeingänge
 - Adressierung, 7-6
 - Interruptprogramm zum Lesen von Werten, 10-123
 - Zugreifen, 6-10
- Analogerweiterungsmodul, Adressierung, 8-2
- Analogpotentiometer, 8-8
 - EM231, A-61
- Analogpotentiometer SMB28, SMB29, D-5
- Ändern eines Pointers, 7-10
- Ändern eines Pointers (indirekte Adressierung), 7-10
- Anforderungen an die Ausrüstung
 - S7-200, 1-2
 - STEP 7-Micro/WIN, 3-1
- Anlauf, Datenhaltung, 7-13–7-17
- Anschlußbelegung, Kommunikationsschnittstelle, 9-6
- Anschlußbelegung der Pole des Kabels, PC/PPI, A-82
- Anweisungsliste, 6-5
 - Anzeigen eines Programms in STEP 7-Micro/WIN, 3-31
 - Anzeigen im Kontaktplan, 3-29
 - Ausführungszeiten, F-1–F-11
 - Editor, 3-29
 - Eingeben von Programmen, 5-21
 - Grundlegende Elemente, 6-6
 - Programm
 - Drucken, 5-23
 - Eingeben in STEP 7-Micro/WIN, 3-29
 - Programmierbeispiel, 4-4
 - Wechseln in den Kontaktplan, 3-31
- Anwenderdefiniertes Protokoll, Frei programmierbare Kommunikation, 9-5
- Anwenderprogramm (OB1), 3-27
- Anzahl an Bytes übertragen, 10-69
- Anzahl an Doppelwörtern übertragen, 10-69
- Anzahl an Wörtern übertragen, 10-69
- Anzeigen, Programm, 3-31
- Approbationen, A-3
- Arbeiten mit Unterprogrammen, 10-88
- Arithmetische Operationen, 10-50–10-65
 - Beispiel, 10-54
 - Ganze Zahlen (16 Bit) addieren, 10-50
 - Ganze Zahlen (16 Bit) subtrahieren, 10-50
 - Ganze Zahlen (32 Bit) addieren, 10-50
 - Ganze Zahlen (32 Bit) subtrahieren, 10-50
 - Ganze Zahlen dividieren, 10-52
 - Ganze Zahlen multiplizieren, 10-52
 - Quadratwurzel einer Realzahl ziehen, 10-53
 - Realzahlen addieren, 10-51
 - Realzahlen dividieren, 10-53
 - Realzahlen multiplizieren, 10-53
 - Realzahlen subtrahieren, 10-51
- ASCII-Zeichenkette in Hexadezimalzahl wandeln, 10-112
- Assistent, TD 200, 5-3
 - Internationale Zeichen und Sonderzeichen, 5-9
- Aus Netz lesen, 10-133
 - Beispiel, 10-134–10-136
 - Fehler, 10-133
- Ausbau
 - Abmessungen
 - CPU 212, 2-3
 - CPU 214, 2-3
 - CPU 215, 2-4
 - CPU 216, 2-4
 - Erweiterungsmodule, 2-4
 - Schraubengröße für den Einbau, 2-3–2-5
 - CPU, 2-7
 - Erweiterungsmodul, 2-7
 - Platzbedarf, 2-2
 - Richtige Ausrichtung des Moduls, 2-7
 - Schraubengröße für den Einbau, 2-3–2-5
- Ausführungszeiten
 - Auswirkungen der Analogein- und Analogausgänge, F-1
 - Auswirkungen der indirekten Adressierung, F-1
 - Auswirkungen des Signalflusses, F-1
 - Operationen der Anweisungsliste, F-1–F-11
- Ausgabepuffer, CPU 215, 9-18, 9-21

- Ausgänge
 Einfrieren, 8-6
 Funktionsweise, 6-4
 Schnelle Impulsausgänge, 8-7
 Ausrichtung des Moduls, 2-5–2-8
- B**
- Batteriemodul
 Abmessungen, A-80
 Bestellnummer, G-3
 Technische Daten, A-80
 Baudrate, PC/PPI-Kabel, A-82
 Baudraten
 CPUs, 9-2
 Kommunikationsschnittstellen, 9-2
 Schalterstellungen am PC/PPI-Kabel, 3-7, 9-10
 Baugruppenparametrierung
 Auswählen, 3-12–3-13
 MPI-Baugruppe (MPI), 3-16–3-17
 MPI-Baugruppe (PPI), 3-14
 PC/PPI-Kabel (PPI), 3-12–3-13
 BCD in ganze Zahl wandeln, 10-108
 Bearbeitung beenden, 10-84
 Beispiele
 Ablaufsteuerungsrelais, 10-93–10-98
 Teilung von Ablaufketten, 10-94
 Weiterschaltbedingungen, 10-98
 Zusammenführung von Ablaufketten, 10-96–10-99
 Adressen von Ein- und Ausgängen, 8-2, 8-3
 Analogpotentiometer, 8-8
 Anzahl an Bytes übertragen, 10-71–10-73
 Arithmetik, 10-54
 ASCII-Zeichenkette in Hexadezimalzahl wandeln, 10-113
 Aus Netz lesen/In Netz schreiben, 10-134–10-136
 Berechnen des Leistungsbedarfs, 2-15
 Bit in Hexadezimalzahl wandeln/Hexadezimalzahl in Bit wandeln, 10-111
 Bitmuster für Sieben-Segment-Anzeige erzeugen, 10-111
 Datenbaustein, 3-32
 Dekrementieren, 10-67
 Einerkomplement, 10-107–10-109
 Ersten Wert aus Tabelle löschen, 10-75
 GSD, 9-24
 Impulsdauermodulation, 10-47
 Impulsfolge, 10-45
 Initialisierung von HSC1, 10-21
 Inkrementieren, 10-67
 Interruptoperationen, 10-122
 Kontakte, 10-6
 Letzten Wert aus Tabelle löschen, 10-74
 Logischer Stack, 10-101–10-103
 Meldung aus Zwischenspeicher übertragen, 10-130
 MPI-Baugruppe mit Master/Slave, 3-9
 Netz mit Token-Passing, 9-28
 Operationen mit Ausgängen, 10-12
 Operationen STOP, END und WDR, 10-86–10-88
 Parameterbaustein, 5-11
 PID-Regler, 10-63–10-65
 Programmierbeispiel, 4-2
 Programmierbeispiel für die DP-Kommunikation, 9-26
 Programmschleife mit FOR/Ende Programmschleife mit NEXT, 10-91–10-93
 Realzahl in ganze Zahl (32 Bit) wandeln, 10-109
 Schiebe- und Rotieroperationen, 10-83–10-85
 Schnelle Zähler, 10-36
 HSC1 bzw. HSC2 in den Zählerarten 9, 10 oder 11, 10-25
 Zähler mit Rücksetz- und Starteingang, 10-23
 Schnellen Zähler aktivieren
 Betrieb von HSC0 in Zählerart 0 und HSC1 oder HSC2 in einer der Zählerarten 0, 1 oder 2, 10-23
 Betrieb von HSC1 oder HSC2 in einer der Zählerarten 3, 4 oder 5, 10-24
 Mit Rücksetzen und ohne Starten, 10-22
 Speicher mit Bitmuster belegen, 10-72–10-74
 Statustabelle, 3-34
 Symboltabelle, 3-36
 Textdisplays TD 200 im Netz, 9-14
 Übertragungsoperationen und Bytes im Wort tauschen, 10-70–10-72
 Umwandlung von Realzahlen, 10-109
 Umwandlungsoperationen, 10-109
 UND, ODER, EXKLUSIV ODER, 10-105–10-107
 Unterprogramm aufrufen, 10-89–10-91
 Vergleichskontakte, 10-9
 Wert in Schieberegister schieben, 10-79–10-81
 Wert in Tabelle eintragen, 10-73
 Wert in Tabelle suchen, 10-77
 Zähler, 10-20

- Zeit als Einschaltverzögerung straten, 10-17
 - Zeit als speichernde Einschaltverzögerung starten, 10-18
 - Zeitgesteuerten Interrupt einrichten, 6-9
 - Zu Sprungmarke springen, 10-87–10-89
 - Beobachten, Programmstatus, 6-17
 - Berechnen des Leistungsbedarfs, 2-15
 - Beschreibung, Funktionsbereiche, 6-2
 - Bestellnummern, G-1
 - Betriebsarten
 - Siehe auch* Operation modes
 - Frei programmierbare Kommunikation, 10-124
 - Funktion Forcen, 6-17
 - Statusbits, D-1
 - Wechseln, 6-13
 - Betriebsarten, PID-Regler, 10-61
 - Betriebsartenschalter, Funktionsweise, 6-13
 - Betriebszustand RUN, 6-13
 - Betriebszustand STOP, 6-13
 - Betriebszustand TERM, 6-13
 - Betriebszustände, Versetzen der CPU in RUN im Programmierbeispiel, 4-15
 - Bias
 - Anpassen, PID-Regler, 10-61
 - PID-Algorithmus, 10-57
 - Bit in Hexadezimalzahl wandeln, 10-110
 - Bitmuster für Sieben-Segment-Anzeige erzeugen (Umwandlungsoperationen), 10-110
 - Bits, 7-2
 - Bits, Sondermerker, D-1–D-13
 - Bitwert direkt rücksetzen, 10-11
 - Bitwert direkt setzen, 10-11
 - Bitwert direkt zuweisen, 10-10
 - Busanschluß, Stecker, 9-7
 - Buserweiterungsanschluß, Herausbrechen der Abdeckung, 2-5–2-7
 - Busverbinder, 2-5–2-7
 - Ausbauen von Erweiterungsmodulen, 2-7
 - Busverstärker
 - Bestellnummer, G-2
 - PROFIBUS-Netz, 9-8
 - Byte, Ganzzahliger Bereich, 7-3
 - Byte links rotieren, 10-81
 - Byte links schieben, 10-80
 - Byte rechts rotieren, 10-81
 - Byte rechts schieben, 10-80
 - Byte übertragen, 10-68
 - Byte um 1 erhöhen, 10-66
 - Byte um 1 vermindern instruction, 10-66
 - Bytekonsistenz, 9-20
 - Bytes, 7-2
 - Bytes durch EXKLUSIV ODER verknüpfen, 10-102
 - Bytes durch ODER verknüpfen, 10-102
 - Bytes durch UND verknüpfen, 10-102
 - Bytes im Wort tauschen, 10-70
 - Bytevergleich, 10-7
- C**
- CP (Kommunikationsprozessor), 9-13
 - Konfiguration mit PC, 9-14
 - Vorgehensweise zum Anschließen, 3-8
 - CP 5411, 9-13
 - Bestellnummer, G-2
 - Einrichten der Parameter für die MPI-Baugruppe (MPI), 3-16–3-17
 - Einrichten der Parameter für die MPI-Baugruppe (PPI), 3-14
 - CP 5511, 9-13
 - Bestellnummer, G-2
 - Einrichten der Parameter für die MPI-Baugruppe (MPI), 3-16–3-17
 - Einrichten der Parameter für die MPI-Baugruppe (PPI), 3-14
 - CP 5611, 9-13
 - Bestellnummer, G-2
 - Einrichten der Parameter für die MPI-Baugruppe (MPI), 3-16–3-17
 - Einrichten der Parameter für die MPI-Baugruppe (PPI), 3-14
 - CPU
 - Allgemeine technische Daten, A-4
 - Bestellnummern, G-1
 - Einstellen der Betriebsart, 6-13
 - Fehlerbehebung, 6-19
 - Funktionsweise, 6-4
 - Kennregister (SMB6), D-4
 - Kommunikationsfähigkeiten, 9-2
 - Laden eines Programms aus STEP 7-Micro/WIN, 3-30
 - Logischer Stack, 6-6
 - Modemanschluß, 3-19–3-24
 - Operandenbereiche, 10-3
 - Paßwort, 6-14
 - Schwere Fehler, C-2
 - Speicher urlöschen, 6-15
 - Speicherbereiche, 7-2
 - Zyklus, 6-10

- CPU 212
- Beispiele für Adressen von Ein- und Ausgängen, 8-3
 - Bestellnummer, G-1
 - E/A, 1-3
 - Eingabefilter, 1-3
 - Erweiterungsmodule, 1-3
 - Funktionen, 10-2
 - Hochleistungskondensator, 1-3
 - Interruptereignisse, 10-117
 - Interrupts, Maximum, 10-120
 - Kommunikation, 9-2
 - Kommunikationsschnittstellen, 1-3
 - Modul, 1-5
 - Operandenbereiche, 10-3
 - Operationen, Ausführungszeiten, F-1–F-10
 - Speicher, 1-3
 - Bereiche, 10-2
 - Technische Daten, A-6–A-15
 - Eingangssimulator, A-84
 - Überblick, 1-3
 - Unterstützte Baudraten, 9-2
 - Unterstützte Hardware zur Kommunikation im Netz, 3-4
 - Unterstützte Interrupts, 1-3, 10-118
 - Unterstützte Operationen, 1-3
 - Ablaufsteuerungsrelais laden, To \square Next, 10-92
 - Alle Interruptereignisse freigeben/Alle Interruptereignisse sperren, 10-116
 - Anzahl an Bytes übertragen, 10-69
 - Anzahl an Wörtern übertragen, 10-69
 - ASCII-Zeichenkette in Hexadezimalzahl wandeln, 10-112
 - BCD in ganze Zahl wandeln, 10-108
 - Bearbeitung bedingt beenden/Bearbeitung absolut beenden, 10-84
 - Bit in Hexadezimalzahl wandeln, 10-110
 - Bitmuster für Sieben-Segment-Anzeige erzeugen, 10-110
 - Bitwert direkt durch ODER verknüpfen/Negierten Bitwert direkt durch ODER verknüpfen, 10-4
 - Bitwert direkt durch UND verknüpfen/Negierten Bitwert direkt durch UND verknüpfen, 10-4
 - Bitwert direkt laden/Negierten Bitwert direkt laden, 10-4
 - Bitwert direkt setzen/Bitwert direkt rücksetzen, 10-11
 - Bitwert direkt zuweisen, 10-10
 - Bitwert durch ODER verknüpfen/Negierten Bitwert durch ODER verknüpfen, 10-4
 - Bitwert durch UND verknüpfen/Negierten Bitwert durch UND verknüpfen, 10-4
 - Bitwert laden/Negierten Bitwert laden, 10-4
 - Byte übertragen, 10-68
 - Bytes im Wort tauschen, 10-70
 - Bytevergleich, 10-7
 - Call, 10-88
 - Direkte Kontakte, 10-4
 - Doppelwort rechts rotieren/Doppelwort links rotieren, 10-82
 - Doppelwort rechts schieben/Doppelwort links schieben, 10-81
 - Doppelwort übertragen, 10-68
 - Doppelwort um 1 erhöhen, 10-67
 - Doppelwort um 1 vermindern, 10-67
 - Doppelwörter durch EXKLUSIV ODER verknüpfen, 10-104
 - Doppelwörter durch ODER verknüpfen, 10-104
 - Doppelwörter durch UND verknüpfen, 10-104
 - Doppelwortvergleich, 10-8
 - Einerkomplement von ganzer Zahl (16 Bit) erzeugen, 10-106
 - Einerkomplement von ganzer Zahl (32 Bit) erzeugen, 10-106
 - END/MEND, 10-84
 - Ende Ablaufsteuerungsrelais, 10-92
 - Erkennung steigende Flanke/Erkennung fallende Flanke, 10-5
 - Erste und zweite Stackebene durch ODER verknüpfen, 10-99
 - Erste und zweite Stackebene durch UND verknüpfen, 10-99
 - Flanke, 10-5
 - Flanke Ablaufsteuerungsrelais, 10-92
 - Ganze Zahl in BCD wandeln, 10-108
 - Ganze Zahlen (16 Bit) addieren, 10-50
 - Ganze Zahlen (16 Bit) subtrahieren, 10-50
 - Ganze Zahlen (32 Bit) addieren, 10-50
 - Ganze Zahlen (32 Bit) subtrahieren, 10-50
 - Ganze Zahlen dividieren, 10-52
 - Ganze Zahlen multiplizieren, 10-52
 - Hexadezimalzahl in ASCII-Zeichenkette wandeln, 10-112
 - Hexadezimalzahl in Bit wandeln, 10-110
 - In STOP gehen, 10-84
 - Interrupt zuordnen/Interrupt trennen,

- 10-116
- Interruptprogramm bedingt/absolut beenden, 10-114
- Interruptprogramm beginnen, 10-114
- Meldung aus Zwischenspeicher übertragen, 10-124
- Modus für schnellen Zähler definieren, 10-21
- NOT, 10-5
- Nulloperation, 10-11
- Obersten Stackwert aus Stack schieben, 10-99
- Obersten Stackwert duplizieren, 10-99
- Setzen/Rücksetzen, 10-10
- Speicher mit Bitmuster belegen, 10-72
- Standardkontakte, 10-4
- Steigende Flanke/Fallende Flanke, 10-5
- Überwachungszeit rücksetzen, 10-85
- Unterprogramm bedingt/absolut beenden, 10-88
- Unterprogramm beginnen, 10-88
- Vorwärts-/Rückwärtszählen, 10-19
- Vorwärtszählen, 10-19
- Wert in Schieberegister schieben, 10-78
- Wort rechts rotieren/Wort links rotieren, 10-82
- Wort rechts schieben/Wort links schieben, 10-80
- Wort übertragen, 10-68
- Wort um 1 erhöhen, 10-66
- Wort um 1 vermindern, 10-66
- Wörter durch EXKLUSIV ODER verknüpfen, 10-103
- Wörter durch ODER verknüpfen, 10-103
- Wörter durch UND verknüpfen, 10-103
- Wortvergleich, 10-7
- Zeit als Einschaltverzögerung starten, 10-13
- Zeit als speichernde Einschaltverzögerung starten, 10-13
- Zu Sprungmarke springen/Sprungmarke definieren, 10-87
- Zuweisen, 10-10
- Zweiten Stackwert kopieren, 10-99
- Unterstützte Protokolle, 1-3
- CPU 214
 - Beispiele für Adressen von Ein- und Ausgängen, 8-3
 - Bestellnummer, G-1
 - E/A, 1-3
 - Eingabefilter, 1-3
 - Erweiterungsmodule, 1-3
 - Funktionen, 10-2
 - Hochleistungskondensator, 1-3
 - Interruptereignisse, 10-117
 - Interrupts, Maximum, 10-120
 - Kommunikation, 9-2
 - Kommunikationsschnittstellen, 1-3
 - Modul, 1-5
 - Operandenbereiche, 10-3
 - Operationen, Ausführungszeiten, F-1–F-10
 - Speicher, 1-3
 - Bereiche, 10-2
 - Technische Daten, A-20–A-29
 - Eingangssimulator, A-85
 - Überblick, 1-3
 - Unterstützte Baudraten, 9-2
 - Unterstützte Hardware zur Kommunikation im Netz, 3-4
 - Unterstützte Interrupts, 1-3, 10-118
 - Unterstützte Operationen, 1-3
 - Ablaufsteuerungsrelais laden, To \square Next, 10-92
 - Alle Interruptereignisse freigeben/Alle Interruptereignisse sperren, 10-116
 - Anzahl an Bytes übertragen, 10-69
 - Anzahl an Wörtern übertragen, 10-69
 - ASCII-Zeichenkette in Hexadezimalzahl wandeln, 10-112
 - Aus Netz lesen/In Netz schreiben, 10-133
 - BCD in ganze Zahl wandeln, 10-108
 - Bearbeitung bedingt beenden/Bearbeitung absolut beenden, 10-84
 - Bit in Hexadezimalzahl wandeln, 10-110
 - Bitmuster für Sieben-Segment-Anzeige erzeugen, 10-110
 - Bitwert direkt durch ODER verknüpfen/Negierten Bitwert direkt durch ODER verknüpfen, 10-4
 - Bitwert direkt durch UND verknüpfen/Negierten Bitwert direkt durch UND verknüpfen, 10-4
 - Bitwert direkt laden/Negierten Bitwert direkt laden, 10-4
 - Bitwert direkt setzen/Bitwert direkt rücksetzen, 10-11
 - Bitwert direkt zuweisen, 10-10
 - Bitwert durch ODER verknüpfen/Negierten Bitwert durch ODER verknüpfen, 10-4
 - Bitwert durch UND verknüpfen/Negierten Bitwert durch UND verknüpfen, 10-4
 - Bitwert laden/Negierten Bitwert laden, 10-4
 - Byte übertragen, 10-68

- Bytes im Wort tauschen, 10-70
- Bytevergleich, 10-7
- Call, 10-88
- Direkte Kontakte, 10-4
- Doppelwort rechts rotieren/Doppelwort links rotieren, 10-82
- Doppelwort rechts schieben/Doppelwort links schieben, 10-81
- Doppelwort übertragen, 10-68
- Doppelwort um 1 erhöhen, 10-67
- Doppelwort um 1 vermindern, 10-67
- Doppelwörter durch EXKLUSIV ODER verknüpfen, 10-104
- Doppelwörter durch ODER verknüpfen, 10-104
- Doppelwörter durch UND verknüpfen, 10-104
- Doppelwortvergleich, 10-8
- Echtzeituhr lesen, 10-49
- Echtzeituhr schreiben, 10-49
- Einerkomplement von ganzer Zahl (16 Bit) erzeugen, 10-106
- Einerkomplement von ganzer Zahl (32 Bit) erzeugen, 10-106
- END/MEND, 10-84
- Ende Ablaufsteuerungsrelais, 10-92
- Ende Programmschleife mit NEXT, 10-90
- Erkennung steigende Flanke/Edge Down, 10-5
- Erste und zweite Stackebene durch ODER verknüpfen, 10-99
- Erste und zweite Stackebene durch UND verknüpfen, 10-99
- Ersten Wert aus Tabelle löschen, 10-75
- Flanke, 10-5
- Flanke Ablaufsteuerungsrelais, 10-92
- Ganze Zahl (32 Bit) in Realzahl wandeln, 10-108
- Ganze Zahl in BCD wandeln, 10-108
- Ganze Zahlen (16 Bit) addieren, 10-50
- Ganze Zahlen (16 Bit) subtrahieren, 10-50
- Ganze Zahlen (32 Bit) addieren, 10-50
- Ganze Zahlen (32 Bit) subtrahieren, 10-50
- Ganze Zahlen dividieren, 10-52
- Ganze Zahlen multiplizieren, 10-52
- Hexadezimalzahl in ASCII-Zeichenkette wandeln, 10-112
- Hexadezimalzahl in Bit wandeln, 10-110
- Impulsausgabe, 10-37
- In STOP gehen, 10-84
- Interrupt zuordnen/Interrupt trennen, 10-116
- Interruptprogramm bedingt/absolut beenden, 10-114
- Interruptprogramm beginnen, 10-114
- Letzten Wert aus Tabelle löschen, 10-74
- Meldung aus Zwischenspeicher übertragen, 10-124
- Modus für schnellen Zähler definieren, 10-21
- NOT, 10-5
- Nulloperation, 10-11
- Obersten Stackwert aus Stack schieben, 10-99
- Obersten Stackwert duplizieren, 10-99
- Programmschleife mit FOR/Ende Programmschleife mit NEXT, 10-90
- Quadratwurzel einer Realzahl ziehen, 10-53
- Realzahl in ganze Zahl (32 Bit) wandeln, 10-108
- Realzahl übertragen, 10-68
- Realzahlen addieren, 10-51
- Realzahlen dividieren, 10-53
- Realzahlen multiplizieren, 10-53
- Realzahlen subtrahieren, 10-51
- Realzahlenvergleich, 10-8
- Setzen/Rücksetzen, 10-10
- Speicher mit Bitmuster belegen, 10-72
- Standardkontakte, 10-4
- Steigende Flanke/Fallende Flanke, 10-5
- Überwachungszeit rücksetzen, 10-85
- Unterprogramm bedingt/absolut beenden, 10-88
- Unterprogramm beginnen, 10-88
- Vorwärts-/Rückwärtszählen, 10-19
- Vorwärtszählen, 10-19
- Wert in Schieberegister schieben, 10-78
- Wert in Tabelle eintragen, 10-73
- Wert in Tabelle suchen, 10-76
- Wort rechts rotieren/Wort links rotieren, 10-82
- Wort rechts schieben/Wort links schieben, 10-80
- Wort übertragen, 10-68
- Wort um 1 erhöhen, 10-66
- Wort um 1 vermindern, 10-66
- Wörter durch EXKLUSIV ODER verknüpfen, 10-103
- Wörter durch ODER verknüpfen, 10-103
- Wörter durch UND verknüpfen, 10-103
- Wortvergleich, 10-7
- Zeit als Einschaltverzögerung starten,

- 10-13
- Zeit als speichernde Einschaltverzögerung starten, 10-13
- Zu Sprungmarke springen/Sprungmarke definieren, 10-87
- Zuweisen, 10-10
- Zweiten Stackwert kopieren, 10-99
- Unterstützte Protokolle, 1-3
- CPU 215
 - als dezentrale Peripherie, 3-19
 - als DP-Slave, 3-19
 - Ausgabepuffer, 9-18, 9-21
 - Beispiele für Adressen von Ein- und Ausgängen, 8-3
 - Bestellnummer, G-1
 - Datenaustausch mit DP-Master, 9-21
 - Datenkonsistenz, 9-20
 - DP-Schnittstelle, 3-19
 - DP-Slave, 9-15
 - E/A, 1-3
 - Eingabefilter, 1-3
 - Eingabepuffer, 9-18, 9-21
 - Erweiterungsmodule, 1-3
 - Funktionen, 10-2
 - Größe des Datenpuffers, 9-19
 - Hochleistungskondensator, 1-3
 - Interruptereignisse, 10-117
 - Interrupts, Maximum, 10-120
 - Kommunikation, 9-2
 - Kommunikationsschnittstellen, 1-3
 - Konfiguration als DP-Slave, 9-17–9-19
 - Konfigurationsrichtlinien, 9-19
 - LED-Statusanzeige für die DP-Kommunikation, 9-22
 - Modul, 1-5
 - Operandenbereiche, 10-3
 - Operationen, Ausführungszeiten, F-1–F-10
 - Programmierbeispiel für einen DP-Slave, 9-26
 - Speicher, 1-3
 - Bereiche, 10-2
 - Statusinformationen als DP-Slave, 9-21
 - Technische Daten, A-32–A-35
 - Eingangssimulator, A-86
 - Überblick, 1-3
 - Unterstützte Baudraten, 9-2
 - Unterstützte E/-A-Konfigurationen, 9-19
 - Unterstützte Hardware zur Kommunikation im Netz, 3-4
 - Unterstützte Interrupts, 1-3, 10-118
 - Unterstützte Operationen, 1-3
 - Ablaufsteuerungsrelais laden, 10-92
 - Alle Interruptereignisse freigeben/Alle Interruptereignisse sperren, 10-116
- Anzahl an Bytes übertragen, 10-69
- Anzahl an Doppelwörtern übertragen, 10-69
- Anzahl an Wörtern übertragen, 10-69
- ASCII-Zeichenkette in Hexadezimalzahl wandeln, 10-112
- Aus Netz lesen/In Netz schreiben, 10-133
- BCD in ganze Zahl wandeln, 10-108
- Bearbeitung bedingt beenden/Bearbeitung absolut beenden, 10-84
- Bit in Hexadezimalzahl wandeln, 10-110
- Bitmuster für Sieben-Segment-Anzeige erzeugen, 10-110
- Bitwert direkt durch ODER verknüpfen/Negierten Bitwert direkt durch ODER verknüpfen, 10-4
- Bitwert direkt durch UND verknüpfen/Negierten Bitwert direkt durch UND verknüpfen, 10-4
- Bitwert direkt ladent/Negierten Bitwert direkt laden, 10-4
- Bitwert direkt setzen/Bitwert direkt rücksetzen, 10-11
- Bitwert direkt zuweisen, 10-10
- Bitwert durch ODER verknüpfen/Negierten Bitwert durch ODER verknüpfen, 10-4
- Bitwert durch UND verknüpfen/Negierten Bitwert durch UND verknüpfen, 10-4
- Bitwert laden/Negierten Bitwert laden, 10-4
- Byte rechts rotieren/Byte links rotieren, 10-81
- Byte rechts schieben/Byte links schieben, 10-80
- Byte übertragen, 10-68
- Byte um 1 erhöhen, 10-66
- Byte um 1 vermindern, 10-66
- Bytes durch EXKLUSIV ODER verknüpfen, 10-102
- Bytes durch ODER verknüpfen, 10-102
- Bytes durch UND verknüpfen, 10-102
- Bytes im Wort tauschen, 10-70
- Bytevergleich, 10-7
- Call, 10-88
- Direkte Kontakte, 10-4
- Doppelwort rechts rotieren/Doppelwort links rotieren, 10-82
- Doppelwort rechts schieben/Doppelwort links schieben, 10-81
- Doppelwort übertragen, 10-68

- Doppelwort um 1 erhöhen, 10-67
 Doppelwort um 1 vermindern, 10-67
 Doppelwörter durch EXKLUSIV ODER verknüpfen, 10-104
 Doppelwörter durch ODER verknüpfen, 10-104
 Doppelwörter durch UND verknüpfen, 10-104
 Doppelwortvergleich, 10-8
 Echtzeituhr lesen, 10-49
 Echtzeituhr schreiben, 10-49
 Einerkomplement von Byte erzeugen, 10-106
 Einerkomplement von ganzer Zahl (16 Bit) erzeugen, 10-106
 Einerkomplement von ganzer Zahl (32 Bit) erzeugen, 10-106
 END/MEND, 10-84
 Ende Ablaufsteuerungsrelais, To \bar{Q} Next, 10-92
 Ende Programmschleife mit NEXT, 10-90
 Erkennung steigende Flanke/Erkennung fallende Flanke, 10-5
 Erste und zweite Stackebene durch ODER verknüpfen, 10-99
 Erste und zweite Stackebene durch UND verknüpfen, 10-99
 Ersten Wert aus Tabelle löschen, 10-75
 Flanke, 10-5
 Flanke Ablaufsteuerungsrelais, 10-92
 Ganze Zahl (32 Bit) in Realzahl wandeln, 10-108
 Ganze Zahl in BCD wandeln, 10-108
 Ganze Zahlen (16 Bit) addieren, 10-50
 Ganze Zahlen (16 Bit) subtrahieren, 10-50
 Ganze Zahlen (32 Bit) addieren, 10-50
 Ganze Zahlen (32 Bit) subtrahieren, 10-50
 Ganze Zahlen dividieren, 10-52
 Ganze Zahlen multiplizieren, 10-52
 Hexadezimalzahl in ASCII-Zeichenkette wandeln, 10-112
 Hexadezimalzahl in Bit wandeln, 10-110
 Impulsausgabe, 10-37
 In STOP gehen, 10-84
 Interrupt zuordnen/Interrupt trennen, 10-116
 Interruptprogramm bedingt/absolut beenden, 10-114
 Interruptprogramm beginnen, 10-114
 Letzten Wert aus Tabelle löschen, 10-74
 Meldung aus Zwischenspeicher übertragen, 10-124
 Meldung in Zwischenspeicher empfangen, 10-124
 Not, 10-5
 Nulloperation, 10-11
 Obersten Stackwert aus Stack schieben, 10-99
 Obersten Stackwert duplizieren, 10-99
 PID-Regler, 10-55
 Programmschleife mit FOR/Ende Programmschleife mit NEXT, 10-90
 Quadratwurzel einer Realzahl ziehen, 10-53
 Realzahl in ganze Zahl (32 Bit) wandeln, 10-108
 Realzahl übertragen, 10-68
 Realzahlen addieren, 10-51
 Realzahlen dividieren, 10-53
 Realzahlen multiplizieren, 10-53
 Realzahlen subtrahieren, 10-51
 Realzahlenvergleich, 10-8
 Schnellen Zähler aktivieren, 10-21
 Setzen/Rücksetzen, 10-10
 Speicher mit Bitmuster belegen, 10-72
 Standardkontakte, 10-4
 Steigende Flanke/Fallende Flanke, 10-5
 Überwachungszeit rücksetzen, 10-85
 Unterprogramm bedingt/absolut beenden, 10-88
 Unterprogramm beginnen, 10-88
 Vorwärts-/Rückwärtszählen, 10-19
 Vorwärtszählen, 10-19
 Wert in Schieberegister schieben, 10-78
 Wert in Tabelle eintragen, 10-73
 Wert in Tabelle suchen, 10-76
 Wort rechts rotieren/Wort links rotieren, 10-82
 Wort rechts schieben/Wort links schieben, 10-80
 Wort übertragen, 10-68
 Wort um 1 erhöhen, 10-66
 Wort um 1 vermindern, 10-66
 Wörter durch EXKLUSIV ODER verknüpfen, 10-103
 Wörter durch ODER verknüpfen, 10-103
 Wörter durch UND verknüpfen, 10-103
 Wortvergleich, 10-7
 Zeit als Einschaltverzögerung starten, 10-13
 Zeit als speichernde Einschaltverzögerung starten, 10-13
 Zu Sprungmarke springen/Sprungmarke

- definieren, 10-87
- Zuweisen, 10-10
- Zweiten Stackwert kopieren, 10-99
- Unterstützte Protokolle, 1-3
- CPU 216
 - Beispiele für Adressen von Ein- und Ausgängen, 8-4
 - Bestellnummer, G-1
 - E/A, 1-3
 - Eingabefilter, 1-3
 - Erweiterungsmodule, 1-3
 - Funktionen, 10-2
 - Hochleistungskondensator, 1-3
 - Interruptereignisse, 10-117
 - Interrupts, Maximum, 10-120
 - Kommunikation, 9-2
 - Kommunikationsschnittstellen, 1-3
 - Modul, 1-5
 - Operandenbereiche, 10-3
 - Operationen, Ausführungszeiten, F-1–F-10
 - Speicher, 1-3
 - Bereiche, 10-2
 - Technische Daten, A-36–A-39
 - Eingangssimulator, A-86
 - Überblick, 1-3
 - Unterstützte Baudraten, 9-2
 - Unterstützte Hardware zur Kommunikation im Netz, 3-4
 - Unterstützte Interrupts, 1-3, 10-118
 - Unterstützte Operationen, 1-3
 - Ablaufsteuerungsrelais laden, 10-92
 - Alle Interruptereignisse freigeben/Alle Interruptereignisse sperren, 10-116
 - Anzahl an Bytes übertragen, 10-69
 - Anzahl an Doppelwörtern übertragen, 10-69
 - Anzahl an Wörtern übertragen, 10-69
 - ASCII-Zeichenkette in Hexadezimalzahl wandeln, 10-112
 - Aus Netz lesen/In Netz schreiben, 10-133
 - BCD in ganze Zahl wandeln, 10-108
 - Bearbeitung bedingt beenden/Bearbeitung absolut beenden, 10-84
 - Bit in Hexadezimalzahl wandeln, 10-110
 - Bitmuster für Sieben-Segment-Anzeige erzeugen, 10-110
 - Bitwert direkt durch ODER verknüpfen/Negierten Bitwert direkt durch ODER verknüpfen, 10-4
 - Bitwert direkt durch UND verknüpfen/Negierten Bitwert direkt durch UND verknüpfen, 10-4
 - Bitwert direkt laden/Negierten Bitwert direkt laden, 10-4
 - Bitwert direkt setzen/Bitwert direkt rücksetzen, 10-11
 - Bitwert direkt zuweisen, 10-10
 - Bitwert durch ODER verknüpfen/Negierten Bitwert durch ODER verknüpfen, 10-4
 - Bitwert durch UND verknüpfen/Negierten Bitwert durch UND verknüpfen, 10-4
 - Bitwert laden/Negierten Bitwert laden, 10-4
 - Byte rechts rotieren/Byte links rotieren, 10-81
 - Byte rechts schieben/Byte links schieben, 10-80
 - Byte übertragen, 10-68
 - Byte um 1 erhöhen, 10-66
 - Byte um 1 vermindern, 10-66
 - Bytes durch EXKLUSIV ODER verknüpfen, 10-102
 - Bytes durch ODER verknüpfen, 10-102
 - Bytes durch UND verknüpfen, 10-102
 - Bytes im Wort tauschen, 10-70
 - Bytevergleich, 10-7
 - Call, 10-88
 - Direkte Kontakte, 10-4
 - Doppelwort rechts rotieren/Doppelwort links rotieren, 10-82
 - Doppelwort rechts schieben/Doppelwort links schieben, 10-81
 - Doppelwort übertragen, 10-68
 - Doppelwort um 1 erhöhen, 10-67
 - Doppelwort um 1 vermindern, 10-67
 - Doppelwörter durch EXKLUSIV ODER verknüpfen, 10-104
 - Doppelwörter durch ODER verknüpfen, 10-104
 - Doppelwörter durch UND verknüpfen, 10-104
 - Doppelwortvergleich, 10-8
 - Echtzeituhr lesen, 10-49
 - Echtzeituhr schreiben, 10-49
 - Einerkomplement von Byte erzeugen, 10-106
 - Einerkomplement von ganzer Zahl (16 Bit) erzeugen, 10-106
 - Einerkomplement von ganzer Zahl (32 Bit) erzeugen, 10-106
 - END/MEND, 10-84
 - Ende Ablaufsteuerungsrelais, To \square Next, 10-92
 - Ende Programmschleife mit NEXT,

- 10-90
- Erkennung steigende Flanke/Erkennung fallende Flanke, 10-5
- Erste und zweite Stackebene durch ODER verknüpfen, 10-99
- Erste und zweite Stackebene durch UND verknüpfen, 10-99
- Ersten Wert aus Tabelle löschen, 10-75
- Flanke, 10-5
- Flanke Ablaufsteuerungsrelais, 10-92
- Ganze Zahl (32 Bit) in Realzahl wandeln, 10-108
- Ganze Zahl in BCD wandeln, 10-108
- Ganze Zahlen (16 Bit) addieren, 10-50
- Ganze Zahlen (16 Bit) subtrahieren, 10-50
- Ganze Zahlen (32 Bit) addieren, 10-50
- Ganze Zahlen (32 Bit) subtrahieren, 10-50
- Ganze Zahlen dividieren, 10-52
- Ganze Zahlen multiplizieren, 10-52
- Hexadezimalzahl in ASCII-Zeichenkette wandeln, 10-112
- Hexadezimalzahl in Bit wandeln, 10-110
- Impulsausgabe, 10-37
- In STOP gehen, 10-84
- Interrupt zuordnen/Interrupt trennen, 10-116
- Interruptprogramm bedingt/absolut beenden, 10-114
- Interruptprogramm beginnen, 10-114
- Letzten Wert aus Tabelle löschen, 10-74
- Meldung aus Zwischenspeicher übertragen, 10-124
- Meldung in Zwischenspeicher empfangen, 10-124
- Modus für schnellen Zähler definieren, 10-21
- NOT, 10-5
- Nulloperation, 10-11
- Obersten Stackwert aus Stack schieben, 10-99
- Obersten Stackwert duplizieren, 10-99
- PID-Regler, 10-55
- Programmschleife mit FOR/Ende Programmschleife mit NEXT, 10-90
- Quadratwurzel einer Realzahl ziehen, 10-53
- Realzahl in ganze Zahl (32 Bit) wandeln, 10-108
- Realzahl übertragen, 10-68
- Realzahlen addieren, 10-51
- Realzahlen dividieren, 10-53
- Realzahlen multiplizieren, 10-53
- Realzahlen subtrahieren, 10-51
- Realzahlenvergleich, 10-8
- Setzen/Rücksetzen, 10-10
- Speicher mit Bitmuster belegen, 10-72
- Standardkontakte, 10-4
- Steigende Flanke/Fallende Flanke, 10-5
- Überwachungszeit rücksetzen, 10-85
- Unterprogramm bedingt/absolut beenden, 10-88
- Unterprogramm beginnen, 10-88
- Vorwärts-/Rückwärtszählen, 10-19
- Vorwärtszählen, 10-19
- Wert in Schieberegister schieben, 10-78
- Wert in Tabelle eintragen, 10-73
- Wert in Tabelle suchen, 10-76
- Wort rechts rotieren/Wort links rotieren, 10-82
- Wort rechts schieben/Wort links schieben, 10-80
- Wort übertragen, 10-68
- Wort um 1 erhöhen, 10-66
- Wort um 1 vermindern, 10-66
- Wörter durch EXKLUSIV ODER verknüpfen, 10-103
- Wörter durch ODER verknüpfen, 10-103
- Wörter durch UND verknüpfen, 10-103
- Wortvergleich, 10-7
- Zeit als Einschaltverzögerung starten, 10-13
- Zeit als speichernde Einschaltverzögerung starten, 10-13
- Zu Sprungmarke springen/Sprungmarke definieren, 10-87
- Zuweisen, 10-10
- Zweiten Stackwert kopieren, 10-99
- Unterstützte Protokolle, 1-3

CPUs

- Abmessungen
 - CPU 212, 2-3
 - CPU 214, 2-3
 - CPU 215, 2-4
 - CPU 216, 2-4
 - Erweiterungsmodule, 2-4
 - Schraubengröße für den Einbau, 2-3–2-5
- Ausbau, 2-7
- Einbau
 - Hutschiene, 2-6
 - Richtige Ausrichtung des Moduls, 2-5–2-8
 - Schalttafel, 2-5
 - Steckleitung, 2-5–2-7
- Leistungsbedarf, 2-15
- Platzbedarf, 2-2
- Schraubengröße für den Einbau, 2-3–2-5
- Vorgehensweise, Ausbau, 2-7

D

- Daten speichern, 7-11–7-16
- Datenaustausch, DP-Master und CPU 215, 9-21
- Datenbaustein
 - Beispiele, 3-32
 - Datentyp, 3-33
 - Erstellen in STEP 7-Micro/WIN, 3-32
 - Gültige Kennzeichen für die Datengröße, 3-33
- Datenbaustein-Editor, 3-32
- Datenblätter. *Siehe* Specifications
- Datenhaltung, 7-11–7-16
 - Anlauf, 7-13–7-17
 - Batteriemodul (optional), 7-11
 - Bereiche, 7-15
 - EEPROM, 7-11, 7-13, 7-16
 - Hochleistungskondensator, 7-11
- Datenkonsistenz, CPU 215, 9-20
- Datentypen, 7-8
- Datenwortformat
 - EM231, A-62
 - EM235, A-72, A-74
- Datum, Einstellen, 10-49
- DC-Aufbau, Richtlinien, 2-11
- DC-Eingangssimulator, Installation, A-84, A-85, A-86
- DC-Eingangssimulator für die CPU 212, Installation, A-84
- DC-Eingangssimulator für die CPU 214, Installation, A-85
- DC-Eingangssimulator für die CPU 215/216, Installation, A-86
- DC-Relais, 2-14
- DC-Transistor, Schutzbeschaltung, 2-13
- Definieren von Meldungen (TD 200), 5-8
- Dekrementieren, 10-50–10-65
 - Beispiel, 10-67
 - Byte um 1 vermindern, 10-66
 - Doppelwort um 1 vermindern, 10-67
 - Ganze Zahlen (16 Bit) subtrahieren, 10-50
 - Ganze Zahlen (32 Bit) subtrahieren, 10-50
 - Realzahlen subtrahieren, 10-51
 - Wort um 1 vermindern, 10-66
- Dezentrale Peripherie
 - CPU 215, 3-19
 - Kommunikation, 3-19, 9-15
- Dezentrale Peripherie (DP): StandardKommunikation, 9-15–9-26
- Diagramm zur Token-Umlaufzeit, 9-30
- Dialogfeld 'PG/PC-Schnittstelle einstellen', 3-10
- Dialogfeld 'Ressourcen' unter Windows NT, 3-6
- Dialogfeld zum Installieren und Deinstallieren von Baugruppen, 3-3
- Differentialanteil, PID-Algorithmus, 10-58
- Digitalausgänge, Schreiben, 6-11
- Digitaleingänge, Lesen, 6-10
- Digitale Erweiterungsmodul, Adressierung, 8-2
- DIN-Hutschiene, Platzbedarf, 2-2–2-4
- DIP-Schalter
 - EM231 Konfiguration, A-61
 - Konfiguration des EM235, A-70, A-71
- Direkte Adressierung, 7-2
- Direkte Kontakte, 10-4
- Direktes Ansteuern der Ein- und Ausgänge, 6-12
- Doppelwort, Ganzzahliger Bereich, 7-3
- Doppelwort links rotieren, 10-82
- Doppelwort links schieben, 10-81
- Doppelwort rechts rotieren, 10-82
- Doppelwort rechts schieben, 10-81
- Doppelwort übertragen, 10-68
- Doppelwort um 1 erhöhen instruction, 10-67
- Doppelwort um 1 vermindern, 10-67
- Doppelwörter durch EXKLUSIV ODER verknüpfen, 10-104
- Doppelwörter durch ODER verknüpfen, 10-104
- Doppelwörter durch UND verknüpfen, 10-104
- Doppelwortvergleich, 10-8
 - Doppelwortvergleich, 10-8

- DP-Kommunikation (dezentrale Peripherie), Programmierbeispiel, 9-26
 - DP-Kommunikation (dezentrale Peripherie)
 - Siehe auch* Remote I/O
 - Arbeiten mit der CPU 215 als Slave, 9-15
 - Verwenden der CPU 215 als Slave, 3-19
 - DP-Kommunikation (dezentrale Peripherie) , 9-15–9-26
 - DP-Master
 - Datenaustausch mit CPU 215, 9-21
 - Konfigurationswerkzeuge, 9-19
 - DP-Schnittstelle, CPU 215, 3-19
 - DP-Standard, Überwachungsstatus, D-12
 - DP-Statusinformationen, CPU 215 als DP-Slave, 9-21
 - Drucken, Programm in AWL oder KOP, 5-23
- E**
- E/A-Adressen, PROFIBUS-DP-Master, 9-18
 - E/A-Konfigurationen, die von der CPU 215 unterstützt werden, 9-19
 - Echtzeituhr (TOD), Aktivieren, 5-4
 - Echtzeituhr lesen, 10-49
 - Echtzeituhr schreiben, 10-49
 - Echtzeituhr-Operationen, 10-49
 - Echtzeituhr lesen, 10-49
 - Echtzeituhr schreiben, 10-49
 - EEPROM, 7-11, 7-13
 - Fehlercodes, C-2
 - Kopieren von Variablenspeicher, 7-16
 - Speichern aus Variablenspeicher, D-6
 - EG-Richtlinien, A-3
 - Einbau
 - Abmessungen, Standard-Hutschiene, 2-3
 - Umgebung mit starken Schwingungen, Verwendung von Erdungsklemmen auf der Hutschiene, 2-6
 - Vorgehensweise
 - Erweiterungsmodul, 2-5–2-7
 - Richtige Ausrichtung des Moduls, 2-5–2-8
 - Einerkomplement von Byte erzeugen, 10-106
 - Einerkomplement von ganzer Zahl (16 Bit) erzeugen, 10-106
 - Einerkomplement von ganzer Zahl (32 Bit) erzeugen, 10-106
 - Einfrieren der Ausgänge, 8-6
 - Eingabefilter, Rauschunterdrückung, 8-5
 - Eingabepuffer, CPU 215, 9-18, 9-21
 - Eingänge, Funktionsweise, 6-4
 - Eingangssimulator
 - Bestellnummer, G-3
 - CPU 212, A-84
 - CPU 214, A-85
 - CPU 215/216, A-86
 - Eingebettete Datenwerte (Textmeldungen), 5-8
 - Formatieren, 5-10
 - Einrichten
 - Kommunikation, 3-7–3-24
 - Kommunikation in Windows-Systemsteuerung, 3-11
 - Kommunikation während der Installation, 3-12
 - Kommunikationsparameter, 3-9
 - Voreinstellungen in STEP 7-Micro/WIN, 3-25
 - Einstellungen der Ausgänge, Konfigurieren der Signalzustände der Ausgänge, 8-6
 - Einstellungen im Dialogfeld 'PG/PC-Schnittstelle einstellen', 3-10
 - Elektromagnetische Verträglichkeit, S7-200, A-5
 - EM221, Technische Daten, A-40–A-43
 - EM222, Technische Daten, A-44–A-46
 - EM223, Technische Daten, A-48–A-54
 - EM231
 - Datenwortformat, A-62
 - DIP-Schalter, A-61
 - , A-61
 - Installationsrichtlinien, A-64
 - Kalibrierung, A-61
 - Konfiguration, Analogeingabebereich, A-61
 - Schaltbild der Eingänge, A-63
 - Technische Daten, A-60–A-64
 - EM235
 - Datenwortformat, A-72, A-74
 - DIP-Schalter
 - , A-70
 - Einstellen, A-71
 - Installationsrichtlinien, A-75
 - Kalibrierung, A-70
 - Konfiguration, Analogeingangsbereich, A-71
 - Schaltbild der Ausgänge, A-74
 - Schaltbild der Eingänge, A-73
 - Technische Daten, A-69–A-75
 - Empfangen von Meldungen, SMB86-SMB94, SMB186-SMB194, D-10
 - Ende Programmschleife mit NEXT, 10-90
 - Entwerfen einer Automatisierungslösung, 6-2

- Erdung und Bezugsspannung, Verdrahtungsregeln, 2-9
 - Ersetzen, 5-19
 - Erste und zweite Stackebene durch ODER verknüpfen, 10-99–10-101
 - Erste und zweite Stackebene durch UND verknüpfen, 10-99–10-101
 - Erstellen, STEP 7-Micro/WIN-Projekt, 3-26
 - Erstellen eines Pointers, 7-9
 - Erstellen eines Programms, Beispiel: Zeitgesteuerten Interrupt einrichten, 6-9
 - Ersten Wert aus Tabelle löschen, 10-75
 - Erweiterungsmodul. *Siehe* EM231, etc.
 - Erweiterungsmodule, 1-4
 - Abmessungen
 - CPU 212, 2-3
 - CPU 214, 2-3
 - CPU 215, 2-4
 - CPU 216, 2-4
 - Module mit 8-, 16- und 32-Ein- und Ausgängen, 2-4
 - Schraubengröße für den Einbau, 2-3–2-5
 - Adressierung der Ein- und Ausgänge, 8-2
 - Ausbau, 2-7
 - Bestellnummern, G-1
 - Einbau
 - Herausziehen des Buserweiterungsanschlusses, 2-5–2-7
 - Hutschiene, 2-6
 - Richtige Ausrichtung des Moduls, 2-5–2-8
 - Schalttafel, 2-5
 - Steckleitung, 2-5–2-7
 - Kenn- und Fehlerregister (SMB8 bis SMB21), D-4
 - Leistungsbedarf, 2-15
 - Platzbedarf, 2-2
 - Schraubengröße für den Einbau, 2-3–2-5
 - Steckleitung, Einbau, 2-5–2-7
 - ET 200, Handbuch, G-3
 - Examples, high-speed counter, operation of HSC1 or HSC2, Modes 6, 7 or 8, 10-24
- F**
- Fallende Flanke, 10-5
 - Fehler
 - Aus Netz lesen/In Netz schreiben, 10-133
 - Leichte, C-3, C-4
 - PID-Regler, 10-62
 - Schwere, C-2
 - SMB1, Bearbeitungsfehler, D-2
 - Verletzungen der Übersetzungsregeln, C-4
 - Zur Laufzeit, C-3
 - Fehler zur Laufzeit, C-3
 - Systemreaktion, 6-20
 - Fehlerbehebung
 - Aus Netz lesen/In Netz schreiben, 10-133
 - Fehler zur Laufzeit, C-3
 - Fehlerbehebung, 6-19
 - Leichte Fehler, 6-20
 - MPI-Kommunikation, 3-17
 - Neustarten der CPU nach schwerem Fehler, 6-19
 - PID-Regler, 10-62
 - Reagieren auf Fehler, 6-19
 - Schwere Fehler, 6-19, C-2
 - STEP 7-Micro/WIN installieren, 3-2
 - Übersetzungsfehler, C-4
 - Vergessenes Paßwort, 6-15
 - Feldverdrahtung
 - Einbau, 2-8
 - Leiterquerschnitte, 2-8
 - Optionalen Klemmenblock, 2-10
 - Filtern von Analogeingängen, 5-14–5-16
 - Forcen von Variablen, Statustabelle, 3-35
 - Formatieren, Datenwert im Text, 5-10
 - Frei programmierbare Kommunikation
 - Anwenderdefiniertes Protokoll, 9-5
 - Definition, 10-118
 - Einsetzen des PC/PPI-Kabels, 9-10–9-11
 - Freigeben, 10-125
 - SMB2, Empfang bei der frei programmierbaren Kommunikation, D-2
 - SMB3, Paritätsfehler bei der frei programmierbaren Kommunikation, D-2
 - SMB30, SMB130: Steuerungsregister der frei programmierbaren Kommunikation, D-6
 - SMB30, SMB130: Steuerungsregister für die frei programmierbare Kommunikation, 10-126

Zeichen-Interrupts, 10-129
 frei programmierbare Kommunikation
 Betriebsarten, 10-124
 Funktionsweise, 10-124
 Initialisieren, 10-126
 Freigabemerker (TD 200), 5-7
 Funktion Forcen, 6-17
 Aktivieren, 5-4
 Funktionen PTO/PWM
 Auswirkungen auf die Ausgänge, 10-43
 Impulsdauer/Impulszahlwert, 10-39
 Initialisierung, 10-40
 Prozeßabbilder, 10-44
 Referenztabelle für Hexadezimalwerte,
 10-40
 Statusbit, 10-39
 Steuerbits, 10-39
 Steuerregister, 10-40
 Zykluszeit, 10-39
 Funktionstasten, Aktivieren, 5-5

G

Ganze Zahl (32 Bit) in Realzahl wandeln,
 10-108
 Ganze Zahl in BCD wandeln, 10-108
 Ganze Zahlen, Umwandeln in Realzahlen,
 10-59
 Ganze Zahlen (16 Bit) addieren, 10-50
 Ganze Zahlen (16 Bit) subtrahieren, 10-50
 Ganze Zahlen (32 Bit) addieren, 10-50
 Ganze Zahlen (32 Bit) subtrahieren, 10-50
 Ganze Zahlen dividieren, 10-52
 Ganze Zahlen multiplizieren, 10-52
 GAP-Aktualisierungsfaktor, 9-31
 Geräte, Master nicht von SIMATIC, 9-24
 Gerätestammdaten (GSD-Datei), 9-23–9-25
 Gerätestammdaten-Datei (GSD-Datei), Einset-
 zen für Master nicht von SIMATIC, 9-24
 Gleitpunktwerte, PID-Regler, 10-59
 Gleitpunktzahlen, Darstellung, 7-3
 GSD, Einsetzen für Master nicht von SIMATIC,
 9-24
 GSD-Datei, 9-23
Siehe auch Device database file
 Gültige Bereiche für die CPUs, 10-2

H

Handbücher, Bestellnummer, G-3

Herausbrechen, Abdeckung des Buserweite-
 rungsanschlusses, 2-5–2-7
 Hexadezimalzahl in ASCII-Zeichenkette wan-
 deln, 10-112
 Hexadezimalzahl in Bit wandeln, 10-110
 Hilfe. *Siehe* Online help
 Hochleistungskondensator, 7-11
 Hochspannungs-Isolationsprüfung, A-5
 Höchste Teilnehmeradresse (HSA), 9-31
 HSA. *Siehe* Highest station address
 HSC-Register, D-8
 Hutschiene
 Abmessungen, 2-3
 Ausbau, 2-7
 Einbau, 2-6
 Einbau bei starken Schwingungen, 2-6
 Platzbedarf, 2-2–2-4
 Vertikaler Einbau, 2-6
 Verwendung von Erdungsklemmen auf der
 Hutschiene, 2-6
 Hutschiene (DIN)
 Abmessungen, 2-3
 Bestellnummer, G-3
 Einbau, 2-6
 Einbau bei starken Schwingungen, 2-6
 Vertikaler Einbau, 2-6
 Verwendung von Erdungsklemmen auf der
 Hutschiene, 2-6

I

Import, Richtlinien und Einschränkungen, E-5
 Importieren, Dateien aus STEP 7-Micro/DOS,
 E-4
 Impulsausgabe, 8-7, 10-13, 10-37
 Impulsausgabe (PLS), 8-7, 10-37
 Impulsdauermodulation (PWM), 8-7, 10-37
 Ändern der Impulsdauer, 10-38, 10-41
 Beispiel, 10-47
 Initialisieren, 10-41
 Impulsfolge (PTO), 8-7, 10-37
 Ändern der Zykluszeit, 10-42
 Ändern der Zykluszeit und des Impulszahl-
 werts, 10-43
 Ändern des Impulszahlwerts, 10-42
 Beispiel, 10-45
 Initialisieren, 10-42
 In Netz schreiben, 10-133
 Beispiel, 10-134–10-136
 Fehler, 10-133
 In STOP gehen, 10-84
 Indirekte Adressierung, 7-9–7-11

- & und *, 7-9
 - Ändern eines Pointers, 7-10
 - Auswirkungen auf die Ausführungszeiten, F-1
 - Initialisieren, Schnelle Zähler, 10-31–10-34
 - Initialisierung
 - Frei programmierbare Kommunikation, 10-126
 - Funktion PWM, 10-41
 - Funktionen PTO/PWM, 10-40
 - Impulsfolge (PTO), 10-42
 - Inkrementieren, 10-50–10-65
 - Beispiel, 10-67
 - Byte um 1 erhöhen, 10-66
 - Doppelwort um 1 erhöhen, 10-67
 - Ganze Zahlen (16 Bit) addieren, 10-50
 - Ganze Zahlen (32 Bit) addieren, 10-50
 - Realzahlen addieren, 10-51
 - Wort um 1 erhöhen, 10-66
 - Inkrementieren eines Pointers, 7-10
 - Installation
 - Ausbau, 2-7
 - DC-Eingangssimulator für die CPU 212, A-84
 - DC-Eingangssimulator für die CPU 214, A-85
 - DC-Eingangssimulator für die CPU 215/216, A-86
 - EM231, A-64
 - EM235, A-75
 - Geräteanordnungen, 2-2
 - Kommunikationshardware, 3-4–3-6
 - Besondere Anweisungen für Windows NT, 3-6
 - Steckleitung für Erweiterungsmodule, A-81
 - STEP 7-Micro/WIN
 - Windows 3.1, 3-2
 - Windows 95, 3-2
 - Windows NT, 3-2
 - Integralanteil, PID-Algorithmus, 10-57
 - Integrierte Ein- und Ausgänge, Adressierung, 8-2
 - Internationale Zeichen, TD 200-Assistent, 5-9
 - Interrupt trennen, 10-116
 - Interrupt zuordnen instruction, 10-116
 - Interruptoperationen, 10-114–10-136
 - Alle Interruptereignisse freigeben, 10-116
 - Alle Interruptereignisse sperren, 10-116
 - Beispiel, 10-122
 - Funktionsweise, 10-116
 - Interrupt trennen, 10-116
 - Interrupt zuordnen, 10-116
 - Interruptprogramm beenden, 10-114
 - Interruptprogramm beginnen, 10-114
 - Interruptprogramm beenden, 10-114
 - Interruptprogramm beginnen, 10-114
 - Interruptprogramme, Richtlinien, 6-8
 - Interrupts
 - CPU 212/214/215/216, 10-2
 - Definitionen für Warteschlangen-Überlauf, 10-120
 - E/A, 10-118
 - Einrichten, 10-116
 - Ereignisse
 - CPU 212/214/215/216, 10-117
 - Prioritäten, 10-121
 - Freigeben und Sperren, 10-116
 - Gemeinsames Nutzen von Daten mit dem Hauptprogramm, 10-115
 - Priorität, To Next, 10-120
 - Programme, 10-114
 - restrictions for using, 10-114
 - Schnelle Zähler, 10-30
 - Steigende/fallende Flanke, 10-118
 - system support, 10-114
 - Warteschlangen, 10-120
 - Zeitgesteuert, 10-119
 - Einrichten zum Lesen eines Analogeingangs, 10-123
 - zeitgesteuerte, D-7
 - Zyklus, 6-11
- ## K
- Kabel
 - Ausbauen von Modulen, 2-7
 - Bestellnummer, G-2
 - Installation der Steckleitung, 2-5–2-7
 - MPI, 3-8
 - PC/PPI, 9-9–9-11
 - Anschlußbelegung der Pole des Kabels, A-82
 - Baudrate, A-82
 - Einrichten der Parameter, 3-12
 - Technische Daten, A-82
 - PROFIBUS-Netz, 9-8
 - Steckleitung für Erweiterungsmodule, Technische Daten, A-81
 - Kalibrieren eines Eingangs
 - EM231, A-62
 - EM235, A-72
 - Kalibrierung
 - EM231, A-61
 - EM235, A-70, A-72
 - Kennzeichnung der Anschlüsse

- CPU 212 24-V-AC/DC/Relais, A-11
- CPU 212 AC/AC/AC, A-13, A-17
- CPU 212 AC/DC/Relais, A-9, A-15
- CPU 212 DC/DC/DC, A-7
- CPU 214 AC/AC/AC, A-25, A-29
- CPU 214 AC/DC (stromliefernd)/Relais, A-27
- CPU 214 AC/DC/Relais, A-23
- CPU 214 DC/DC/DC, A-21
- CPU 215 AC/DC/Relais, A-35
- CPU 215 DC/DC/DC, A-33
- CPU 216 AC/DC/Relais, A-39
- CPU 216 DC/DC/DC, A-37
- EM221 Digitaleingabe (stromliefernd) 8 x 24 V DC, A-42
- EM221 Digitaleingabe 8 x 120 V AC, A-41
- EM221 Digitaleingabe 8 x 24 V AC, A-43
- EM221 Digitaleingabe 8 x 24 V DC, A-40
- EM222 Digitalausgabe 8 x 120/230 V AC, A-47
- EM222 Digitalausgabe 8 x 24 V DC, A-44
- EM222 Digitalausgabe 8 x Relais, A-45
- EM223 Digitalein-/Digitalausgabe 16 x 24 V DC/16 x Relais, A-59
- EM223 Digitalein-/Digitalausgabe 4 x 120 V AC/4 x 120/230 V AC, A-55
- EM223 Digitalein-/Digitalausgabe 4 x 24 V DC/4 x 24 V DC, A-49
- EM223 Digitalein-/Digitalausgabe 4 x 24 V DC/4 x Relais, A-54
- EM223 Digitalein-/Digitalausgabe 4 x 24 V DC/8 x Relais, A-57
- EM231 Analogeingabe AE 3 x 12 Bit, A-60
- EM235 Analogein-/Analogausgabemodul AE 3/AA 1 x 12 Bit, A-70
- Kommunikation
 - Anforderungen, 6-11
 - Anschließen eines PC, 3-7
 - Auswählen der Baugruppenparametrierung, 3-12–3-13
 - Baudraten, 9-2
 - Dezentrale Peripherie, 3-19, 9-15
 - DP-Standard (dezentrale Peripherie), 9-15–9-26
 - Einsetzen der CPU 215 als Slave, 9-15
 - Verwenden der CPU 215 als Slave, 3-19
 - Einrichten, 3-7–3-24
 - Einrichten der Parameter, 3-9
 - Einrichten in der Systemsteuerung von Windows, 3-11
 - Einrichten während der Installation, 3-12
 - Fähigkeiten, 9-2
 - Fehlerbehebung, 3-17
 - Frei programmierbare Kommunikation, 10-124, D-6
 - Hardware
 - Installieren unter NT, 3-6
 - Installieren/Deinstallieren, 3-4–3-6
 - Konfiguration der CPU 215 als DP-Slave, 9-17–9-19
 - Konfigurationen, 9-2
 - Master/Slaves, 9-9
 - Modem, 3-19–3-24
 - MPI, 3-8, 9-3
 - Netzkomponenten, 9-6
 - PPI, 3-7, 9-3
 - PROFIBUS-DP-Protokoll, 9-4
 - Programmierbeispiel für die CPU 215 als DP-Slave, 9-26
 - Prüfen der Einstellungen, 3-9
 - Über CP, 3-8
 - Über CP-Karte, 9-13–9-14
 - Über MPI-Baugruppe, 3-8, 9-13–9-14
 - Über PC/PPI-Kabel, 9-9–9-11
 - Unterstützte Protokolle, 9-2
- Kommunikationsoperationen, 10-124–10-136
 - Aus Netz lesen, 10-133
 - In Netz schreiben, 10-133
 - Meldung aus Zwischenspeicher übertragen, 10-124
 - Meldung in Zwischenspeicher empfangen, 10-124
- Kommunikationsprozessor (CP), Bestellnummer, G-2
- Kommunikationsschnittstelle
 - Anschlußbelegung, 9-6
 - Interrupts, 10-118
- Konfiguration
 - Berechnen des Leistungsbedarfs, B-1
 - CPU 215 als DP-Slave, 9-17–9-19
 - DP-Master, 9-19
 - EM231, A-61
 - EM235, A-71
 - Gerätstammdaten (GSD-Datei) für PROFIBUS, 9-23–9-25
 - Kommunikationshardware, 3-4
 - Meldungen (TD 200), 5-3, 5-6–5-10
 - Parameterbaustein, 5-3
 - PC mit CP-Karte und Programmiergerät, 9-14
 - PC mit MPI-Baugruppe und Programmiergerät, 9-14
 - Remanente Bereiche im Speicher, 7-15
 - Signalzustände der Ausgänge, 8-6
 - Voreinstellungen, 3-25
 - zeichnen der Konfigurationspläne, 6-3

Konsistenz, Daten, 9-20
 Konstanten, 7-8
 Kontakte, 10-4–10-6
 Beispiel, 10-6
 Direkte Kontakte, 10-4
 Fallende Flanke, 10-5
 NOT, 10-5
 Standardkontakte, 10-4
 Steigende Flanke, 10-5
 Kontaktplan
 Anzeigen eines Programms in STEP 7-Micro/WIN, 3-31
 Drucken von Programmen, 5-23
 Editor, 3-27
 Eingeben von Programmen, 5-21
 Grundlegende Elemente, 6-5
 Programm, Eingeben in STEP 7-Micro/WIN, 3-27
 Programmierbeispiel, 4-5, 4-10
 Wechseln in die Anweisungsliste, 3-31
 Kontrollkästchen 'Multi Master Netzwerk', 3-13
 Kontrollkästchen 'Wird nicht als einziger Master aktiv', 3-17
 KOP, Programmstatus, 6-17

L

Laden
 Erforderlicher Betriebszustand, 6-13
 Fehlermeldung, 4-15
 Programmierbeispiel, 4-15
 Voraussetzungen, 4-15
 Laden aus der CPU, Programm, 7-11
 Laden in die CPU, Programm, 3-30, 7-11
 LED-Statusanzeige für die DP-Kommunikation, CPU 215 als DP-Slave, 9-22
 Leichte Fehler
 Betrieb der CPU, 6-20
 Systemreaktion, 6-20
 Leistungsbedarf
 Berechnen, 2-15
 Berechnungstabelle, B-1
 CPU, 2-15
 Erweiterungsmodul, 2-15
 Letzten Wert aus Tabelle löschen, 10-74
 Logische MPI-Verbindungen, 9-4

Logische Verbindungen, MPI, 9-3, 9-4
 Logischer Stack
 Ablaufsteuerungsrelais (SCRs), 10-92
 Operation, 6-6

M

Master
 GSD, 9-24
 Kommunikation, 9-9
 Modem, 3-19
 MPI-Protokoll, 9-3, 9-13
 Nicht von SIMATIC, 9-24
 PPI-Protokoll, 9-3
 PROFIBUS-DP-Protokoll, 9-4
 Meldung aus Zwischenspeicher übertragen, 10-124, 10-127
 Beispiel, 10-130
 Meldung in Zwischenspeicher empfangen, 10-124, 10-127
 Meldungen
 Adressen, 5-7
 Definieren, 5-8
 Einbetten von Datenwerten, 5-8
 Formatieren von eingebetteten Datenwerten, 5-10
 Freigabemerker, TD 200, 5-7
 Größe/Anzahl, 5-6
 Netz mit Token-Passing, 9-29
 Merker, Adressierung, 7-3
 Modem
 Einsetzen des PC/PPI-Kabels, 9-12
 Kabel, 3-19
 Kommunikation im Netz, 3-19–3-24
 Null-modem-Adapter, 9-12
 Verbinden von PC/PG und CPU, 3-19–3-20
 Modulgröße
 CPU 212, 2-3
 CPU 214, 2-3
 CPU 215, 2-4
 CPU 216, 2-4
 Erweiterungsmodule, 2-4
 Schraubengröße für den Einbau, 2-3–2-5
 Modus für schnellen Zähler definieren, 10-21
 Modus für schnellen Zähler definieren (HDEF), 10-21

- Modus für schnellen Zähler definieren instruction, Zählerarten, 10-28
- Montage
- Abmessungen
 - CPU 212, 2-3
 - CPU 214, 2-3
 - CPU 215, 2-4
 - CPU 216, 2-4
 - Erweiterungsmodule, 2-4
 - Schraubengröße für den Einbau, 2-3-2-5
 - Standard-Hutschiene, 2-3
 - Ausbau, 2-7
 - Platzbedarf, 2-2
 - Schraubengröße für den Einbau, 2-3-2-5
 - Umgebung mit starken Schwingungen, Verwendung von Erdungsklemmen auf der Hutschiene, 2-6
 - Vertikaler Einbau, Verwendung von Erdungsklemmen auf der Hutschiene, 2-6
 - Vorgehensweise
 - Erweiterungsmodul, 2-5-2-7
 - Hutschiene, 2-6
 - Richtige Ausrichtung des Moduls, 2-5-2-8
 - Schalttafel, 2-5
- MPI (mehrpunktfähige Schnittstelle), Protokoll, 9-3
- Baudrate, 9-13
- MPI-Baugruppe, 3-8, 9-13
- Bestellnummer, G-2
 - Einrichten der Parameter für die MPI-Baugruppe (MPI), 3-16-3-17
 - Einrichten der Parameter für die MPI-Baugruppe (PPI), 3-14
 - Konfiguration mit PC, 9-14
 - MPI-Parameter, 3-16
 - PPI-Parameter, 3-14
 - Vorgehensweise zum Anschließen, 3-8
- MPI-Kabel, 3-8
- MPI-Kommunikation, 3-8, 9-3
- CP-Karte, 9-13
 - Fehlerbehebung, 3-17
 - Voreingestellte Adressen, 3-17
- MPI-Verbindungen, 9-3
- N**
- Netz
- Abschließen, 9-7
 - Abschluß, 9-7
 - Arbeiten mit Mastern nicht von SIMATIC, 9-24
 - Auswählen der Parametrierung, 3-12
 - Busverstärker, 9-8
 - Einrichten der Kommunikation, 3-7-3-24
 - Einschränkungen, 9-28
 - GAP-Aktualisierungsfaktor, 9-31
 - Höchste Teilnehmeradresse (HSA), 9-31
 - Installieren der Kommunikationshardware, 3-4-3-6
 - Kabelverbindungen, 9-9
 - Kommunikationsschnittstelle, 9-6
 - Komponenten, 9-6
 - Leistungsfähigkeit, 9-28
 - Master, 9-2
 - Mehrere Master, 9-13
 - Optimieren der Leistungsfähigkeit, 9-31
 - Segmente, 9-2
 - Senden von Meldungen, 9-29
 - Slaves, 9-2
 - Technische Daten des Kabels, 9-8
 - Teilnehmeradressen, 9-2
 - Token-Umlaufzeit, 9-29-9-32
- Netz mit mehreren Mastern, 9-13
- CP-Karte, 9-13
 - MPI-Baugruppe, 9-13
- Netz mit Token-Passing, Beispiel, 9-28
- Netzwerk, Suchen/Ersetzen, 5-19
- Neustarten der CPU, Nach schwerem Fehler, 6-19
- Normen, Nationale und internationale, A-3
- Null-Modem-Adapter, 3-19-3-20, 9-12
- Nulloperation, 10-11
- Nullspannungsfestes Speichern, 7-16
- Nummern, Verwenden von Konstanten, 7-8
- O**
- OB1 (Anwenderprogramm), 3-27
- Obersten Stackwert aus Stack schieben, 10-99-10-101
- Obersten Stackwert duplizieren, 10-99-10-101
- Online-Hilfe, STEP 7-Micro/WIN, 3-1
- Operandenbereiche, CPU 212/214/215/216, 10-3
- Operation NOT, 10-5
- Operationen
- Ablaufsteuerungsrelais, 10-92
 - Alle Interruptereignisse freigeben, 10-116
 - Alle Interruptereignisse sperren, 10-116
 - Ändern eines Pointers, 7-10
 - Anzahl an Bytes übertragen, 10-69
 - Anzahl an Doppelwörtern übertragen, 10-69
 - Anzahl an Wörtern übertragen, 10-69

- Arithmetik, 10-50–10-65
- ASCII-Zeichenkette in Hexadezimalzahl wandeln, 10-112
- Aus Netz lesen, 10-133
- Ausführungszeiten, F-1–F-9
- Ausgänge, 10-10–10-12
- BCD in ganze Zahl wandeln, 10-108
- Bearbeitung beenden, 10-84
- Bit in Hexadezimalzahl wandeln, 10-110
- Bitmuster für Sieben-Segment-Anzeige erzeugen, 10-110
- Bitwert direkt rücksetzen, 10-11
- Bitwert direkt setzen, 10-11
- Bitwert direkt zuweisen, 10-10
- Byte links rotieren, 10-81
- Byte links schieben, 10-80
- Byte rechts rotieren, 10-81
- Byte rechts schieben, 10-80
- Byte übertragen, 10-68
- Byte um 1 erhöhen, 10-66
- Byte um 1 vermindern, 10-66
- Bytes durch EXKLUSIV ODER verknüpfen, 10-102
- Bytes durch ODER verknüpfen, 10-102
- Bytes durch UND verknüpfen, 10-102
- Bytes im Wort tauschen, 10-70
- Bytevergleich, 10-7
- Dekrementieren, 10-50–10-65
- Direkte Kontakte, 10-4
- Doppelwort links rotieren, 10-82
- Doppelwort links schieben, 10-81
- Doppelwort rechts rotieren, 10-82
- Doppelwort rechts schieben, 10-81
- Doppelwort übertragen, 10-68
- Doppelwort um 1 erhöhen, 10-67
- Doppelwort um 1 vermindern, 10-67
- Doppelwörter durch EXKLUSIV ODER verknüpfen, 10-104
- Doppelwörter durch ODER verknüpfen, 10-104
- Doppelwörter durch UND verknüpfen, 10-104
- Doppelwortvergleich, 10-8
- Echtzeituhr lesen, 10-49
- Echtzeituhr schreiben, 10-49
- Einerkomplement von Byte erzeugen, 10-106
- Einerkomplement von ganzer Zahl (16 Bit) erzeugen, 10-106
- Einerkomplement von ganzer Zahl (32 Bit) erzeugen, 10-106
- Ende Programmschleife mit NEXT, 10-90
- Erste und zweite Stackebene durch ODER verknüpfen, 10-99–10-101
- Erste und zweite Stackebene durch UND verknüpfen, 10-99–10-101
- Ersten Wert aus Tabelle löschen, 10-75
- Fallende Flanke, 10-5
- Ganze Zahl (32 Bit) in Realzahl wandeln, 10-108
- Ganze Zahl in BCD wandeln, 10-108
- Ganze Zahlen (16 Bit) addieren, 10-50
- Ganze Zahlen (16 Bit) subtrahieren, 10-50
- Ganze Zahlen (32 Bit) addieren, 10-50
- Ganze Zahlen (32 Bit) subtrahieren, 10-50
- Ganze Zahlen dividieren, 10-52
- Ganze Zahlen multiplizieren, 10-52
- Hexadezimalzahl in ASCII-Zeichenkette wandeln, 10-112
- Hexadezimalzahl in Bit wandeln, 10-110
- Impulsausgabe, 10-13, 10-37
- Impulsausgabe (PLS), 8-7, 10-37
- In Netz schreiben, 10-133
- In STOP gehen, 10-84
- Inkrementieren, 10-50–10-65
- Inkrementieren eines Pointers, 7-10
- Interrupt trennen, 10-116
- Interrupt zuordnen, 10-116
- Interruptprogramm beenden, 10-114
- Interruptprogramm beginnen, 10-114
- Interrupts, 10-114–10-136
- Kommunikation, 10-124–10-136
- Kontakte, 10-4–10-6
- Letzten Wert aus Tabelle löschen, 10-74
- Logischer Stack, 10-99–10-101
- Meldung aus Zwischenspeicher übertragen, 10-124
- Meldung in Zwischenspeicher empfangen, 10-124
- Modus für schnellen Zähler definieren, 10-21
- Modus für schnellen Zähler definieren (HDEF), 10-21
- NOT, 10-5
- Nulloperation, 10-11
- Obersten Stackwert aus Stack schieben, 10-99–10-101
- Obersten Stackwert duplizieren, 10-99–10-101
- PID-Regler, 10-55–10-65
- Programmschleife mit FOR, 10-90
- Programmsteuerung, 10-84–10-98
- Quadratwurzel einer Realzahl ziehen, 10-53
- Realzahl in ganze Zahl (32 Bit) wandeln, 10-108
- Realzahl übertragen, 10-68

Realzahlen addieren, 10-51
 Realzahlen dividieren, 10-53
 Realzahlen multiplizieren, 10-53
 Realzahlen subtrahieren, 10-51
 Realzahlenvergleich, 10-8
 Rotieren, 10-68–10-77
 Rücksetzen, 10-10
 Schieben, 10-68–10-77
 Schnelle Ausgänge, 8-7, 10-37–10-49
 Schnelle Zähler, 10-13
 Schnellen Zähler aktivieren, 10-21–10-49
 Schnellen Zähler aktivieren (HSC), 10-21
 Schneller Zähler, 8-7
 Setzen, 10-10
 Speicher mit Bitmuster belegen,
 10-68–10-77
 Standardkontakte, 10-4
 Steigende Flanke, 10-5
 Suchen, 10-73–10-77
 Suchen/Ersetzen, 5-19
 Tabellenoperationen, 10-73–10-77
 Übertragen, 10-68–10-77
 Überwachungszeit rücksetzen, 10-85–10-87
 Umwandlung, 10-108–10-113
 Unterprogramm aufrufen, 10-88
 Unterprogramm beenden, 10-88
 Unterprogramm beginnen, 10-88
 Verknüpfungsoperationen, 10-102–10-107
 Vorwärts-/Rückwärtszählen, 10-19
 Vorwärtszählen, 10-19
 Wert in Schieberegister schieben, 10-78
 Wert in Schieberegister schieben (SHRB),
 10-78
 Wert in Tabelle eintragen, 10-73
 Wert in Tabelle suchen, 10-76
 Wort links rotieren, 10-82
 Wort links schieben, 10-80
 Wort rechts rotieren, 10-82
 Wort rechts schieben, 10-80
 Wort übertragen, 10-68
 Wort um 1 erhöhen, 10-66
 Wort um 1 vermindern, 10-66
 Wörter durch EXKLUSIV ODER verknüpfen,
 10-103
 Wörter durch ODER verknüpfen, 10-103
 Wörter durch UND verknüpfen, 10-103
 Wortvergleich, 10-7
 Zähler, 10-13–10-49
 Zeit als Einschaltverzögerung starten, 10-13
 Zeit als speichernde Einschaltverzögerung
 starten, 10-13
 Zeiten, 10-13–10-49
 Zu Sprungmarke springen, 10-87

Zuweisen (Spule), 10-10
 Zweiten Stackwert kopieren, 10-99–10-101
 Operationen für Ablaufsteuerungsrelais, Bei-
 spiele, 10-93–10-97
 Operationen für das Ablaufsteuerungsrelais,
 10-92
 Operationen für das Ablaufsteuerungsrelais
 (SCR), 10-93
 Operationen mit Ausgängen, 10-10–10-12
 Beispiel, 10-12
 Bitwert direkt rücksetzen, 10-11
 Bitwert direkt setzen, 10-11
 Bitwert direkt zuweisen, 10-10
 Nulloperation, 10-11
 Rücksetzen, 10-10
 Setzen, 10-10
 Zuweisen (Spule), 10-10
 Operationen mit schnellen Ausgängen,
 10-37–10-49
 Siehe auch PTO/PWM functions
 Impulsausgabe, 10-37
 Operations-Assistent, S7-200, Analogeingabefil-
 ter, 5-14–5-16
 Operations-Assistent-200, Aufrufen/Einsetzen,
 5-12–5-14
 Operator-Stationen, definieren, 6-3

P

Parameter, Suchen/Ersetzen, 5-19
 Parameterbaustein (TD 200), 5-2
 Adresse, 5-7
 Beispiel, 5-11
 Konfigurieren, 5-3
 Speichern/Anzeigen, 5-11
 Parametrierung, Baugruppe
 Auswählen, 3-12–3-13
 MPI-Baugruppe (MPI), 3-16–3-17
 MPI-Baugruppe (PPI), 3-14
 PC/PPI-Kabel (PPI), 3-12–3-13
 Paßwort
 Aktivieren des Paßwortschutzes (TD 200),
 5-4
 CPU, 6-14
 Einrichten, 6-14
 Einschränken des Zugriffs, 6-14
 Schutzstufen, 6-14
 Urlöschen, 6-15
 Vergessen, 6-15
 PC/PPI-Kabel, 9-9–9-11
 Abmessungen, A-83
 Arbeiten in der frei programmierbaren Kom-

- munikation, 9-10–9-11
- Einrichten der Parameter, 3-12
- Einsetzen eines Modems, 9-12
- Pole der RS-232-Schnittstelle, 9-10
- Schalterstellungen, 3-7
- Schalterstellungen für Baudraten, 9-10
- Technische Daten, A-82
- Verwenden mit Modem, 3-19–3-20
- Vorgehensweise zum Anschließen, 3-7
- PC/PPI-Netz, 9-9
- Physikalische Größe
 - CPU 212, 2-3
 - CPU 214, 2-3
 - CPU 215, 2-4
 - CPU 216, 2-4
 - Erweiterungsmodule, 2-4
 - Schraubengröße für den Einbau, 2-3–2-5
- PID-Algorithmus, 10-55–10-59
- PID-Regler, 10-55–10-65
 - Auswählen, 10-58
 - Auswählen des Regelkreises, 10-58
 - Beispiel, 10-63–10-65
 - Bereiche, Variablen, 10-60
 - Bereiche/Variablen, 10-60
 - Betriebsarten, 10-61
 - Bias, 10-61
 - CPU 212/214/215/216, 10-2
 - Fehlerbedingungen, 10-62
 - Programmierbeispiel, 10-63–10-65
 - Tabelle für den Regelkreis, 10-62
 - Umwandeln der Eingabewerte, 10-59
 - Umwandeln von Stellgrößen, 10-60
 - Verlaufsbits, 10-61
 - Vorwärts-/Rückwärtsverhalten, 10-60
- Platzbedarf, 2-2
- Pointer, 7-9–7-11
 - & und *, 7-9
 - Ändern eines Pointers, 7-10
- Potentiometer
 - EM235, A-70
 - SMB28, SMB29, 8-8
- PPI (Punkt-zu-Punkt-Schnittstelle)
 - Kabelverbindungen, 9-9
 - Kommunikation, 3-7
 - Netzverbindung, 9-9
 - Protokoll, 9-3
- PPI-Kommunikation, 9-3
- PROFIBUS
 - Busverstärker, 9-8
 - Datenkonsistenz, 9-20
 - Gerätstammdaten (GSD-Datei), 9-23–9-25
 - Netzkabel, Technische Daten, 9-8
 - PROFIBUS-Standard, Anschlußbelegung, 9-6
 - PROFIBUS-DP, 9-17
 - Siehe auch DP (distributed peripheral) standard*
 - Protokoll, 9-4
 - PROFIBUS-DP-Kommunikation, 9-4
 - PROFIBUS-DP-Master, E/A-Adreßbereich, 9-18
 - PROFIBUS-DP-Standard, 9-15
 - Program, Ein- und Ausgänge, 6-4
 - Programm
 - Analogeingänge, 6-10
 - Anzeigen in STEP 7-Micro/WIN, 3-31
 - Arbeiten mit der Statustabelle, 6-16
 - Arbeiten mit Unterprogrammen, 10-88
 - Aufbau, 6-8
 - Bearbeiten, 6-11
 - Beobachten des Status, 6-17
 - Drucken, 5-23
 - Eingeben, 5-21
 - Eingeben von Kommentaren, 5-21
 - Erstellen in STEP 7-Micro/WIN, 3-27–3-31
 - Grundlegende Elemente, 6-8
 - Importieren aus STEP 7-Micro/DOS, E-4
 - Laden aus der CPU, 7-11
 - Laden in die CPU, 7-11
 - Laden in die CPU aus STEP 7-Micro/WIN, 3-30
 - Nullspannungsfestes Speichern, 7-16
 - Programmierbeispiel, 4-2–4-19
 - Richtlinien und Einschränkungen beim Importieren, E-5
 - Speichern, 7-11–7-14, 7-17
 - testen, 6-16–6-18
 - Übersetzen in STEP 7-Micro/WIN, 3-29
 - überwachen, 6-16–6-18
 - Zurückholen aus dem Speichermodul, 7-18
 - Programmierbeispiel
 - Anlegen eines Projekts, 4-6
 - Anweisungsliste, 4-4
 - Aufgaben, 4-3
 - Eingeben im Kontaktplan, 4-10–4-14
 - Erstellen einer Statustabelle, 4-14
 - Erstellen einer Symboltabelle, 4-8
 - Kontaktplan, 4-5
 - Laden, 4-15
 - Speichern, 4-13
 - Systemanforderungen, 4-2
 - Übersetzen, 4-13
 - Überwachen, 4-16
 - Wechseln des Betriebszustands, 4-15
 - Programmierkonzept, 6-4
 - Programmiersoftware, Bestellnummer, G-3

Programmiersprache, Konzept, 6-5
 Programmschleife mit FOR, 10-90
 Programmsteuerungsoperationen, 10-84–10-98
 Ablaufsteuerungsrelais, 10-92
 Bearbeitung beenden, 10-84
 Beispiel, 10-86–10-88
 Ende Programmschleife mit NEXT, 10-90
 In STOP gehen, 10-84
 Beispiel, 10-86–10-88
 Programmschleife mit FOR, 10-90
 Programmschleife mit FOR/Ende Pro-
 grammschleife mit NEXT, Beispiel,
 10-91–10-93
 Überwachungszeit rücksetzen, 10-85–10-87
 Beispiel, 10-86–10-88
 Unterprogramm aufrufen, 10-88
 Beispiel, 10-89–10-91
 Unterprogramm beenden, 10-88
 Unterprogramm beginnen, 10-88
 Zu Sprungmarke springen, 10-87
 Beispiel, 10-87–10-89
 Projekt
 Anlegen, 4-6
 Erstellen in STEP 7-Micro/WIN, 3-26
 Komponenten, 3-30
 Laden in die CPU, 3-30
 Programmierbeispiel, 4-6
 Speichern in STEP 7-Micro/WIN, 3-26
 Proportionalanteil, PID-Algorithmus, 10-57
 Protokolle. *Siehe* Communications, protocols;
 Module parameter set
 Prozeßabbild der Ausgänge, 6-11, 6-12
 Adressierung, 7-3
 Funktionen PTO/PWM, 10-44
 Prozeßabbild der Eingänge, 6-12
 Adressierung, 7-3
 Funktion, 6-10
 Prozeßvariable, Umwandeln, 10-59
 Prüfen der Datentypen, 7-8
 PTO/PWM, 10-38–10-44
 PTO-Pipeline, 10-38
 Steuerbyte, 10-38
 PTO/PWM-Funktionen, Steuerungsregister,
 SMB66-SMB85, D-9
 Pufferkonsistenz, 9-20
 Punkt-zu-Punkt-Kommunikation, 1-3

Q

Quadratwurzel einer Realzahl ziehen, 10-53
 Querverweise, 5-17
 Querverweisliste, Drucken, 5-23

R

Rauschunterdrückung, Eingabefilter, 8-5
 Realzahl in ganze Zahl (32 Bit) wandeln,
 10-108
 Realzahl übertragen, 10-68
 Realzahlen addieren, 10-51
 Realzahlen dividieren, 10-53
 Realzahlen multiplizieren, 10-53
 Realzahlen subtrahieren, 10-51
 Realzahlenvergleich, 10-8
 Relais, Widerstände/Kondensatoren, 2-14
 Remanente Bereiche im Speicher, Definieren,
 7-15
 Reserviert SMB7, D-4
 Richtige Ausrichtung des Moduls, 2-5–2-8
 Richtlinien
 AC-Aufbau, 2-10
 Amerikanischer Aufbau, 2-12
 Ändern eines Pointers für die indirekte
 Adressierung, 7-10
 DC-Aufbau, 2-11
 Eingeben von symbolischen Adressen, 3-36
 Entwerfen einer Automatisierungslösung,
 6-2–6-4
 Erdung und Bezugsspannung, 2-9
 Hardware-Installation, 2-2–2-4
 Installieren des EM235, A-75
 Schutzbeschaltungen, 2-13
 AC-Ausgang, 2-14
 DC-Relais, 2-14
 Umgebungen mit starken Schwingungen, 2-6
 Verdrahtung, 2-8
 Galvanische Trennung, 2-9
 Vertikaler Einbau, 2-6
 Verwenden der Operation Überwachungszeit
 rücksetzen, 10-85
 Verwendung von Erdungsklemmen auf der
 Hutschiene, 2-6
 Richtlinien der Europäischen Gemeinschaft
 (EG), A-3
 Rotieroperationen, 10-68–10-77
 Beispiel für Schiebe- und Rotieroperationen,
 10-83–10-85
 Byte links rotieren, 10-81
 Byte rechts rotieren, 10-81
 Doppelwort links rotieren, 10-82
 Doppelwort rechts rotieren, 10-82
 Wort links rotieren, 10-82
 Wort rechts rotieren, 10-82
 Rücksetzen, 10-10

S**S7-200**

Abmessungen

CPU 212, 2-3

CPU 214, 2-3

CPU 215, 2-4

CPU 216, 2-4

Erweiterungsmodule, 2-4

Schraubengröße für den Einbau, 2-3–2-5

Ausbau, 2-7

CPU-Überblick, 1-3

CPUs, Ausbau, 2-7

Einbau

Hutschiene, 2-6

Richtige Ausrichtung des Moduls,
2-5–2-8

Schalttafel, 2-5

Steckleitung, 2-5–2-7

Elektromagnetische Verträglichkeit, A-5

Erweiterungsmodule, 1-4

Ausbau, 2-7

Komponenten, 1-4

Operations-Assistent, 5-12–5-16

Analogeingabefilter, 5-14–5-16

Platzbedarf, 2-2

Schraubengröße für den Einbau, 2-3–2-5

Systemkomponenten, 1-2

Technische Daten, A-4

Umgebungsbedingungen, A-4

Schaltbild der Ausgänge, EM235, A-74

Schaltbild der Eingänge, EM231, A-63, A-73

Schalterstellungen, PC/PPI-Kabel, 3-7

Schaltplan

CPU 212 24-V-AC/DC/Relais, A-11

CPU 212 AC/AC/AC, A-13, A-17

CPU 212 AC/DC (stromliefernd)/Relais,
A-15

CPU 212 AC/DC/Relais, A-9

CPU 212 DC/DC/DC, A-7

CPU 214 AC/AC/AC, A-25, A-29

CPU 214 AC/DC (stromliefernd)/Relais,
A-27

CPU 214 AC/DC/Relais, A-23

CPU 214 DC/DC/DC, A-21

CPU 215 AC/DC/Relais, A-35

CPU 215 DC/DC/DC, A-33

CPU 216 AC/DC/Relais, A-39

CPU 216 DC/DC/DC, A-37

EM221 Digitaleingabe (stromliefernd) 8 x
24 V DC, A-42

EM221 Digitaleingabe 8 x 120 V AC, A-41

EM221 Digitaleingabe 8 x 24 V AC, A-43

EM221 Digitaleingabe 8 x 24 V DC, A-40
EM222 Digitalausgabe 8 x 120/230 V AC,
A-47

EM222 Digitalausgabe 8 x 24 V DC, A-44

EM222 Digitalausgabe 8 x Relais, A-45

EM223 Digitalein-/Digitalausgabe 16 x 24 V
DC/16 x Relais, A-59EM223 Digitalein-/Digitalausgabe 4 x 120 V
AC/4 x 120/230 V AC, A-55EM223 Digitalein-/Digitalausgabe 4 x 24 V
DC/4 x 24 V DC, A-49EM223 Digitalein-/Digitalausgabe 4 x 24 V
DC/4 x Relais, A-54EM223 Digitalein-/Digitalausgabe 4 x 24 V
DC/8 x Relais, A-57

EM231 Analogeingabe AE 3 x 12 Bit, A-60

EM235 Analogein-/Analogausgabemodul
AE 3/AA 1 x 12 Bit, A-70

Schalttafel

Abmessungen

CPU 212, 2-3

CPU 214, 2-3

CPU 215, 2-4

CPU 216, 2-4

Erweiterungsmodule, 2-4

Ausbau, 2-7

Einbau, 2-5

Steckleitung, 2-5–2-7

Schiebeoperation, Beispiel für die Operation

Wert in Schieberegister schieben,
10-79–10-81

Schiebeoperationen, 10-68–10-77

Beispiel für Schiebe- und Rotieroperationen,
10-83–10-85

Byte links schieben, 10-80

Byte rechts schieben, 10-80

Doppelwort links schieben, 10-81

Doppelwort rechts schieben, 10-81

Wert in Schieberegister schieben, 10-78

Wort links schieben, 10-80

Wort rechts schieben, 10-80

Schnelle Ausgänge

Ändern der Impulsdauer, 8-7, 10-38

Funktionsweise, 10-37

Funktionsweise von PTO/PWM,
10-38–10-44Operation PTO/PWM, SMB66-SMB85:
Sondermerker, D-9

Schnelle Ein- und Ausgänge, 8-7

Schnelle Impulsausgabe, 8-7

- Schnelle Zähler, 10-13
 Adressierung, 7-7
 Auswählen der Aktivitätsstufe, 10-28
 Beispiele, 10-36
 Einstellen von aktuellen und voreingestellten Werten, 10-29
 HSC-Interrupts, 10-30
 Initialisieren der Zählerarten, 10-31–10-34
 Laden von neuen aktuellen/voreingestellten Werten, 10-35
 Modus für schnellen Zähler definieren, 10-21
 Richtungswechsel, 10-35
 SMB36-SMB 65: HSC-Register, D-8
 Sperren, 10-35
 Statusbyte, 10-30
 Steuerbyte, 10-28
 Verdrahtung der Eingänge, 10-26
 Zählerarten, 10-27
- Schnellen Zähler aktivieren, 10-21–10-49
 Beispiele, 10-22–10-25
 Funktionsweise, 10-22
 Impulsdigramme, 10-22–10-25
 Schnellen Zähler aktivieren, 10-21
- Schnellen Zähler aktivieren (HSC), 10-21
- Schneller Zähler, 8-7
- Schraubengröße (zum Einbau), 2-3–2-5
- Schreibsteuerung, D-6
- Schutzbeschaltungen, Richtlinien
 AC-Ausgang, 2-14
 DC-Relais, 2-14
 DC-Transistor, 2-13
- Schwere Fehler, C-2
 Betrieb der CPU, 6-19
- Schwingungspotential beim Einbau, Verwendung von Erdungsklemmen auf der Hut-schiene, 2-6
- Segmente, Netz, 9-2
- Setzen, 10-10
- Sicherheitsstromkreise, entwerfen, 6-3
- Signalfuß, Auswirkungen auf die Ausführungszeiten, F-1
- Simulator. *Siehe* Input simulator
- Skalieren von Stellgrößen im Regelkreis, 10-60
- Slaves
 CPU 215 als DP-Slave, 3-19, 9-15
 Kommunikation, 9-9
- SM0.2: Remanente Daten verloren, 7-14
- SMB0: Statusbits, D-1
- SMB1: Statusbits, D-2
- SMB110-SMB115: Status des DP-Standardprotokolls, D-12
- SMB186-SMB194: Steuerung des Meldungsempfangs, To \bar{Q} Next, D-10
- SMB2: Empfang bei der frei programmierbaren Kommunikation, D-2
- SMB2: In frei programmierbarer Kommunikation empfangene Zeichen, Zeichen-Interrupts, 10-129
- SMB28, SMB29 Analogpotentiometer, 8-8
- SMB3: Paritätsfehler bei der frei programmierbaren Kommunikation, D-2
- SMB3: Paritätsfehler frei programmierbare Kommunikation, Zeichen-Interrupts, 10-129
- SMB30, SMB130: Steuerungsregister der frei programmierbaren Kommunikation, 10-126, D-6
- SMB34/SMB35: Intervallregister für zeitgesteuerte Interrupts, D-7
- SMB36-SMB65: HSC-Register, D-8
- SMB6: Kennregister der CPU, D-4
- SMB66-SMB85: PTO/PWM-Register, D-9
- SMB8-SMB21: Kenn- und Fehlerregister der Erweiterungsmodule, D-4
- SMB86-SMB94: Steuerung des Meldungsempfangs, D-10
- SMW22-SMW26: Zykluszeiten, D-5
- Sollwert, Umwandeln, 10-59

- Sondermerker, D-1–D-13
 - Adressierung, 7-4
 - Analogpotentiometer SMB28, SMB29, D-5
 - Reserviert SMB7, D-4
 - SMB0: Statusbits, D-1
 - SMB1: Statusbits, D-2
 - SMB110-SMB115: Status des DP-Standardprotokolls, D-12
 - SMB186-SMB194: Steuerung des Meldungsempfangs, D-10
 - SMB2: Empfang bei der frei programmierbaren Kommunikation, D-2
 - SMB3: Paritätsfehler bei der frei programmierbaren Kommunikation, D-2
 - SMB30, SMB130: Steuerungsregister der frei programmierbaren Kommunikation, 10-126, D-6
 - SMB31: Schreibsteuerung nullspannungsfester Speicher (EEPROM), D-6
 - SMB34/SMB35: Intervallregister für zeitgesteuerte Interrupts, D-7
 - SMB36-SMB65: HSC-Register, D-8
 - SMB6: Kennregister der CPU, D-4
 - SMB66-SMB85: PTO/PWM-Register, D-9
 - SMB8-SMB21: Kenn- und Fehlerregister der Erweiterungsmodule, D-4
 - SMB86-SMB94: Steuerung des Meldungsempfangs, D-10
 - SMW22-SMW26: Zykluszeiten, D-5
 - SMW32: Schreibsteuerung nullspannungsfester Speicher (EEPROM), D-6
 - Status der Ein- und Ausgänge SMB5, D-3
 - Überlauf der Warteschlange SMB4, D-3
- Speicher
 - Urlöschen, 6-15
 - Verwendete Elemente, 5-18
- Speicher mit Bitmuster belegen, 10-68–10-77
 - Beispiel, 10-72–10-74
 - Speicher mit Bitmuster belegen, 10-72
- Speicherbereiche, 6-4
 - Bits, 7-2
 - Bytes, 7-2
 - CPU, 7-2
 - CPU 212/214/215/216, 10-2
 - Operandenbereiche, 10-3
 - Zugreifen auf Daten, 6-4, 7-2
- Speichermodul, 7-11
 - Abmessungen, A-78
 - Bestellnummer, G-3
 - Fehlercodes, C-2
 - Kopieren in, 7-17
 - Stecken, 7-17
 - Technische Daten, A-78
 - Verwenden, 7-17
 - Ziehen, 7-17
 - Zurückholen des Programms, 7-18
- Speichern
 - Programm, 7-16
 - STEP 7-Micro/WIN-Projekt, 3-26
 - Wert im EEPROM, D-6
- Sprache, Textdisplay, 5-4
- Sprungmarke definieren, 10-87
- StackOperationen, Erste und zweite Stackebene durch UND verknüpfen, 10-99–10-101
- Stackoperationen, 10-99–10-101
 - Beispiel, 10-101–10-103
 - Erste und zweite Stackebene durch ODER verknüpfen, 10-99–10-101
 - Funktionsweise, 10-100
 - Obersten Stackwert aus Stack schieben, 10-99–10-101
 - Obersten Stackwert duplizieren, 10-99–10-101
 - Zweiten Stackwert kopieren, 10-99–10-101
- Standard-Hutschiene
 - Abmessungen, 2-3
 - Ausbau, 2-7
 - Einbau, 2-6
 - Einbau bei starken Schwingungen, 2-6
 - Platzbedarf, 2-2–2-4
 - Vertikaler Einbau, 2-6
 - Verwendung von Erdungsklemmen auf der Hutschiene, 2-6
- Standardkontakte, 10-4
- Status der Ein- und Ausgänge SMB5, D-3
- Statusanzeige, CPU 215 als DP-Slave, 9-22
- Statusbits (SMB0), D-1
- Statusbyte, schnelle Zähler, 10-30
- Statusinformationen, CPU 215 als DP-Slave, 9-21
- Statustabelle
 - Ändern des Programms, 6-16
 - Bearbeiten von Adressen, 3-35
 - Erstellen für das Programmierbeispiel, 4-14
 - Forcen von Variablen, 3-35
 - Lesen und Schreiben von Variablen, 3-34
 - Programmierbeispiel, 4-14
 - STEP 7-Micro/WIN, 3-34
 - Werte beobachten/bedienen, 4-17

- Statustabelle/Tabelle zum Forcen, Zyklus, 6-17
- Stecken, Speichermodul, 7-17
- Stecker, Busanschluß, 9-7
- Steckleitung
Siehe auch I/O expansion cable
 Installation, 2-5-2-7
- Steckleitung für Erweiterungsmodule
 Installation, A-81
 Technische Daten, A-81
- Steckverbinder, Bestellnummern, G-2
- Steigende Flanke, 10-5
- STEP 7-Micro/DOS
 Importieren von Dateien, E-4
 Umwandeln von Dateien, E-4
- STEP 7-Micro/WIN
 Anforderungen an die Ausrüstung, 3-1
 Anzeigen eines Programms, 3-31
 Bestellnummer, G-3
 Bestellnummer Kopierlizenz, G-3
 Bestellnummer Update, G-3
 Datenbaustein-Editor, 3-32
 Einrichten der Kommunikation, 3-10
 Erstellen eines Datenbausteins, 3-32
 Erstellen eines Programms, 3-27-3-31
 Erstellen eines Projekts, 3-26
 Fehlerbehebung bei der Installation, 3-2
 Hardware zur Kommunikation im Netz, 3-4
 Installieren, 3-2
 Installieren der Kommunikationshardware,
 3-4-3-6
 Kommunikation per Modem, 3-19-3-24
 Laden eines Programms in die CPU, 3-30
 Online-Hilfe, 3-1
 Speichern eines Projekts, 3-26
 Statustabelle, 3-34
 Übersetzen eines Programms, 3-29
 Umwandeln von Dateien aus STEP 7-Micro/
 DOS, E-4
 Voreinstellungen, 3-25
- Steuerbits, Schnelle Zähler, 10-28
- Suchen/Ersetzen, 5-19
- Suchoperationen, 10-73-10-77
 Ersten Wert aus Tabelle löschen, 10-75
 Letzten Wert aus Tabelle löschen, 10-74
 Wert in Tabelle eintragen, 10-73
 Wert in Tabelle suchen, 10-76
- Symbol, Suchen/Ersetzen, 5-19
- Symbolische Adressierung, 3-36
- Symbolischen Adressen, erstellen, 6-3
- Symboltabelle
 Bearbeitungsfunktionen, 3-37
 Erstellen, 4-8
 Programmierbeispiel, 4-8
 Sortieren nach Namen/Adressen, 3-37
 STEP 7-Micro/WIN, 3-36
- Synchrone Aktualisierung, Funktion PWM,
 10-41
- Systementwurf, Micro-SPS, 6-2
- ## T
- Tabelle für den Regelkreis, 10-62
- Tabelle für den Regelkreis beim PID-Regler,
 10-62
- Tabelle für die Funktionen PTO/PWM, 10-40
- Tabellenoperationen, 10-73-10-77
 Ersten Wert aus Tabelle löschen, 10-75
 Letzten Wert aus Tabelle löschen, 10-74
 Wert in Tabelle eintragen, 10-73
 Wert in Tabelle suchen, 10-76
- Takt, Statusbits, D-1
- Tastenkombinationen mit der ALT-Taste, 5-9
- TD 200, 5-2-5-9
 Aktualisierungsrate für die Anzeige, 5-5
 Assistent zum Konfigurieren, 5-3
 Erstellen von Meldungen, 5-8
 Funktion Forcen, 5-4
 Funktionstasten, 5-5
 Konfigurieren des Parameterbausteins, 5-3
 Meldungen, 5-6-5-10
 Parameterbaustein, 5-2
 Paßwortschutz, 5-4
 Sprache, 5-4
 Zeichensatz für Balkenanzeigen, 5-4
- Technische Daten
 Batteriemodul, A-80
 CPU 212, A-6-A-15
 CPU 214, A-20-A-29
 CPU 215, A-32-A-35
 CPU 216, A-36-A-39
 Eingangssimulator
 CPU 212, A-84
 CPU 214, A-85
 CPU 215/216, A-86
 EM221, A-40-A-43
 EM222, A-44-A-46
 EM223, A-48-A-54
 EM231, A-60-A-64
 EM235, A-69-A-75
 Familie S7-200, A-4
 PC/PPI-Kabel, A-82
 Speichermodul, A-78
 Steckleitung für Erweiterungsmodule, A-81
- Testen, Programm, 6-16-6-18
- Textdisplay, TD 200, 5-2

Token-Umlaufzeit, 9-29–9-32

U

Überblick über die S7-200 CPUs

Funktionen, 1-3

Operandenbereiche, 10-3

Überblick über S7-200 CPUs, Speicherbereiche,
10-2

Überlauf der Warteschlange SMB4, D-3

Übersetzen

Fehler

Systemreaktion, 6-20

Verletzungen der Übersetzungsregeln,
C-4

Programm in STEP 7-Micro/WIN, 3-29

Überspannungsschutz Diode, 2-13

Übertragungsoperationen, 10-68–10-77

Anzahl an Bytes übertragen, 10-69

Anzahl an Doppelwörtern übertragen, 10-69

Anzahl an Wörtern übertragen, 10-69

Beispiel, 10-71–10-73

Beispiel für Übertragungsoperationen und
die Operatione Bytes im Wort tauschen,
10-70–10-72

Byte übertragen, 10-68

Bytes im Wort tauschen, 10-70

Doppelwort übertragen, 10-68

Realzahl übertragen, 10-68

Wort übertragen, 10-68

Überwachen

Adressen, 5-17

Adressen/Bereiche, 5-18

Programm, 6-16–6-18

Programmierbeispiel, 4-16

Überwachungszeit rücksetzen, 10-85–10-87

Richtlinien, 10-85

Uhr, Aktivieren, 5-4

Uhr, Echtzeit-, 10-49

Uhrzeit, Einstellen, 10-49

Umgebung mit starken Schwingungen, Verwen-
dung von Erdungsklemmen auf der Hut-
schiene, 2-6

Umgebungsbedingungen, A-4

Umwandeln

Dateien aus STEP7-Micro/DOS, E-4

Ganze Zahl in Realzahl, 10-59

Prozeßvariablen, 10-59

Realzahlen in normalisierte Werte, 10-59

Speichern eines umgewandelten Programms,
E-6

Umwandlungsoperationen, 10-108–10-113

ASCII-Zeichenkette in Hexadezimalzahl
wandeln, 10-112

BCD in ganze Zahl wandeln, 10-108

Bit in Hexadezimalzahl wandeln, 10-110

Bitmuster für Sieben-Segment-Anzeige er-
zeugen, 10-110

Ganze Zahl (32 Bit) in Realzahl wandeln,
10-108

Ganze Zahl in BCD wandeln, 10-108

Hexadezimalzahl in ASCII-Zeichenkette
wandeln, 10-112

Hexadezimalzahl in Bit wandeln, 10-110

Realzahl in ganze Zahl (32 Bit) wandeln,
10-108

Unterprogramm

Beispiel, 6-9

Richtlinien, 6-8

Unterprogramm aufrufen, 10-88

Unterprogramm beenden, 10-88

Unterprogramm beginnen, 10-88

V

Variablen, Forcen, 3-35, 6-17

Variablenspeicher

Adressierung, 7-3

Kopieren in den EEPROM, 7-16

Verbinder, Buserweiterungsanschluß, 2-5–2-7

Herausbrechen der Abdeckung, 2-7

Verbindungen, MPI, 9-3, 9-4

Verdrahtung

Ausbauen von Modulen, 2-7

Eingänge, schnelle Zähler, 10-26

Optionaler Klemmenblock für die Feldver-
drahtung, 2-10

Richtlinien, 2-8–2-13

AC-Aufbau, 2-10

Amerikanischer Aufbau, 2-12

DC-Aufbau, 2-11

Schutzbeschaltungen, 2-13–2-14

Verdrahtungsrichtlinien für AC-Aufbau, 2-10

Verdrahtungsrichtlinien für amerikanischen Auf-
bau, 2-12

Verdrahtungsrichtlinien für DC-Aufbau, 2-11

Vergleich, S7-200 CPUs, 1-3

Vergleichskontakt, Beispiel, 10-9

- Vergleichskontakte
 Beispiel, 10-9
 Bytevergleich, 10-7
 Realzahlenvergleich, 10-8
 Wortvergleich, 10-7
- Verknüpfungsoperationen, 10-102–10-107
 Beispiel
 Einerkomplement, 10-107–10-109
 UND, ODER, EXKLUSIV ODER,
 10-105–10-107
 Bytes durch EXKLUSIV ODER verknüpfen,
 10-102
 Bytes durch ODER verknüpfen, 10-102
 Bytes durch UND verknüpfen, 10-102
 Doppelwörter durch EXKLUSIV ODER ver-
 knüpfen, 10-104
 Doppelwörter durch ODER verknüpfen,
 10-104
 Doppelwörter durch UND verknüpfen,
 10-104
 Einerkomplement von Byte erzeugen,
 10-106
 Einerkomplement von ganzer Zahl (16 Bit)
 erzeugen, 10-106
 Einerkomplement von ganzer Zahl (32 Bit)
 erzeugen, 10-106
 Wörter durch EXKLUSIV ODER verknüp-
 fen, 10-103
 Wörter durch ODER verknüpfen, 10-103
 Wörter durch UND verknüpfen, 10-103
- Vertikaler Einbau, Verwendung von Erdungs-
 klemmen auf der Hutschiene, 2-6
- Verwenden eines Pointers
 & und *, 7-9
 Ändern eines Pointers, 7-10
- Verwendete Elemente, 5-18
 Drucken, 5-23
- Voreinstellungen, Einrichten, 3-25
 Vorwärts-/Rückwärtszählen, 10-19
 Vorwärtszählen, 10-19
- W**
- Wert in Schieberegister schieben, 10-78
 Wert in Schieberegister schieben (SHRB), 10-78
 Wert in Tabelle eintragen, 10-73
 Wert in Tabelle suchen, 10-76
- Werte
 Datenbaustein, 3-33
 In Textmeldungen, 5-8
- Widerstände/Kondensatoren, Relais, 2-14
- Windows 3.1
 Fehlerbehebung beim Einrichten der MPI-
 Kommunikation, 3-17
 Installieren von STEP 7-Micro/WIN, 3-2
- Windows 95, Installieren von STEP 7-Micro/
 WIN, 3-2
- Windows NT
 Fehlerbehebung beim Einrichten der MPI-
 Kommunikation, 3-18
 Installieren von Hardware, 3-6
 Installieren von STEP 7-Micro/WIN, 3-2
- Wort, Ganzzahliger Bereich, 7-3
 Wort links rotieren, 10-82
 Wort links schieben, 10-80
 Wort rechts rotieren, 10-82
 Wort rechts schieben, 10-80
 Wort übertragen, 10-68
 Wort um 1 erhöhen, 10-66
 Wort um 1 vermindern, 10-66
- Wörter durch EXKLUSIV ODER verknüpfen,
 10-103
 Wörter durch ODER verknüpfen, 10-103
 Wörter durch UND verknüpfen, 10-103
- Wortkonsistenz, 9-20
 Wortvergleich, 10-7
- Z**
- Zahlen, Darstellung, 7-3
- Zähler, 10-13–10-49
 Adressierung, 7-5
 Arten, 7-5
 Beispiel, 10-20
 CPU 212/214/215/216, 10-2
 Funktionweise, 10-19
 Variablen, 7-5
 Vorwärts-/Rückwärtszählen, 10-19
 Vorwärtszählen, 10-19
- Zählerarten, Schnelle Zähler, 10-27
- Zeichen, TD 200-Assistent, 5-9
 Zeichen-Interrupts, 10-129
 Zeichensatz für Balkenanzeigen, TD 200, 5-4
 Zeit als Einschaltverzögerung starten, 10-13

- Zeit als speichernde Einschaltverzögerung starten, 10-13
- Zeiten, 10-13–10-49
 - Adressierung, 7-4
 - Aktualisieren, 10-14–10-18
 - Auflösung, 10-13
 - Beispiel für die Operation Zeit als Einschaltverzögerung starten, 10-17
 - Beispiel für die Operation Zeit als speichernde Einschaltverzögerung starten, 10-18
 - CPU 212/214/215/216, 10-2
 - Nummer, 10-13
 - Operation, 10-13
 - Zeit als Einschaltverzögerung starten, 10-13
 - Zeit als speichernde Einschaltverzögerung starten, 10-13
- Zeiten T32/T96, Interrupts, 10-119
- Zeitgesteuerte Interrupts, 10-119
 - SMB34, SMB35, D-7
- Zeitgesteuerter Interrupt, Beispiel, 6-9, 10-123
- Ziehen, Speichermodul, 7-17
- Zu Sprungmarke springen, 10-87
- Zugreifen
 - Direkte Adressierung, 7-2
 - Operandenbereiche, 10-3
 - Speicherbereiche
 - & und *, 7-9
 - Ändern eines Pointers, 7-10
 - Indirekte Adressierung, 7-9–7-11
 - Zugriff im Bitformat, 7-2
 - CPU 212/214/215/216, 10-3
 - Zugriff im Byteformat, 7-2
 - CPU 212/214/215/216, 10-3
 - Verwenden eines Pointers, 7-10
 - Zugriff im Doppelwortformat, CPU 212/214/215/216, 10-3
 - Zugriff im Wortformat, 7-2
 - CPU 212/214/215/216, 10-3
 - Verwenden eines Pointers, 7-10
 - Zugriffsbeschränkung. *Siehe* Password
 - Zuweisen (Spule), 10-10
 - Zweiten Stackwert kopieren, 10-99–10-101
- Zyklus
 - Aufgaben, 6-10
 - Funktion Forcen, 6-18
 - Statusbits, D-1
 - Statustabelle/Tabelle zum Forcen, 6-17
 - Unterbrechen, 6-11
- Zykluszeit, Impulsfolge (PTO), 10-42
- Zykluszeit, SMW22 bis SMW26), D-5

An
Siemens AG
A&D AS E 46
Östliche Rheinbrückenstr. 50
76181 Karlsruhe

Absender:

Ihr Name: _____
Ihre Funktion: _____
Ihre Firma: _____
Straße: _____
Ort: _____
Telefon: _____

Bitte kreuzen Sie Ihren zutreffenden Industriezweig an:

- | | |
|--|--|
| <input type="checkbox"/> Automobilindustrie | <input type="checkbox"/> Pharmazeutische Industrie |
| <input type="checkbox"/> Chemische Industrie | <input type="checkbox"/> Kunststoffverarbeitung |
| <input type="checkbox"/> Elektroindustrie | <input type="checkbox"/> Papierindustrie |
| <input type="checkbox"/> Nahrungsmittel | <input type="checkbox"/> Textilindustrie |
| <input type="checkbox"/> Leittechnik | <input type="checkbox"/> Transportwesen |
| <input type="checkbox"/> Maschinenbau | <input type="checkbox"/> Andere _____ |
| <input type="checkbox"/> Petrochemie | |



Vorschläge und Anmerkungen zur Anwenderdokumentation

Ihre Anmerkungen und Vorschläge helfen uns, die Qualität und Benutzbarkeit unserer Dokumentation zu verbessern. Bitte füllen Sie diesen Fragebogen bei der nächsten Gelegenheit aus und senden Sie ihn an Siemens zurück.

Geben Sie bitte bei den folgenden Fragen Ihre persönliche Bewertung mit Werten von 1 = gut bis 5 = schlecht an.

1. Entspricht der Inhalt Ihren Anforderungen?
2. Sind die benötigten Informationen leicht zu finden?
3. Sind die Texte leicht verständlich?
4. Entspricht der Grad der technischen Einzelheiten Ihren Anforderungen?
5. Wie bewerten Sie die Qualität der Abbildungen und Tabellen?

Falls Sie auf konkrete Probleme gestoßen sind, erläutern Sie diese bitte in den folgenden Zeilen:

