SIEMENS

SIMATIC

S7-200 Systemhandbuch

Vorwort, Inhaltsverzeichnis	
Produktübersicht	1
Erste Schritte	2
Einbauen der S7-200	3
SPS-Grundlagen	4
Programmiergrundlagen, Konventionen und Funktionen	5
S7-200 Befehlssatz	6
Kommunikation im Netz	7
Hardware-Fehlerbehebung und Werkzeuge für den Software-Test	8
Erstellen eines Programms für das Positioniermodul	9
Erstellen eines Programms für das Modemmodul	10
Steuern eines MicroMaster-Antriebs mit der Bibliothek für das USS-Protokoll	11
Bibliothek für das Modbus-Protokoll	12
Technische Daten	A
Berechnen der Leistungsbilanz	B
Fehlermeldungen	C
Sondermerker	D
S7-200 Bestellnummern	E
Ausführungszeiten von AWL-Operationen	F
S7-200 Kurzinformation	G
Index	

Dieses Handbuch hat die Bestellnummer: **6ES7298-8FA23-8AH0**

Ausgabe 05/2003

A5E00212535-02

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:

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

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

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

Notice
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

Bitte beachten Sie:

Warning
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 HMI® und SIMATIC NET® sind 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 2003 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 Automation and Drives
Geschäftsgebiet Industrial Automation Systems
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 2003
Technische Änderungen bleiben vorbehalten.

Siemens Aktiengesellschaft

6ES7298-8FA23-8AH0

Excellence in Automation & Drives: Siemens

Vorwort

Die Familie S7-200 umfasst verschiedene Kleinsteuerungen (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 S7-200 hervorragend für kleinere Steuerungsanwendungen. Die zahlreichen Ausführungen der S7-200 und die vielfältigen Programmiermöglichkeiten unter Windows bieten Ihnen extrem hohe Flexibilität beim Umsetzen Ihrer Automatisierungslösungen.

Leserkreis

Dieses Handbuch bietet Informationen zum Installieren und Programmieren von S7-200 Kleinsteuerungen und wendet sich an Ingenieure, Programmierer und Wartungspersonal mit allgemeinen Kenntnissen über Automatisierungssysteme.

Umfang des Handbuchs

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

- ❏ Folgende S7-200 CPUs: CPU 221, CPU 222, CPU 224, CPU 226 und CPU 226XM
- ❏ Erweiterungsmodule S7-200 EM 22x
- ❏ STEP 7-Micro/WIN, Version 3.2, eine 32-Bit-Programmiersoftware für die S7-200
- ❏ STEP 7-Micro/WIN Operationsbibliotheken und TP-Designer für das TP070, Version 1.0, Software-Werkzeuge für Kunden, die die S7-200 mit anderen Komponenten, z.B. TP070 Touch Panel, Modbus oder einem MicroMaster-Antrieb einsetzen

Normen

Die Produktreihe SIMATIC S7-200 entspricht folgenden Normen:

- ❏ EG-Richtlinie zu Niederspannungen
  EN 61131-2: Automatisierungssysteme - Anforderungen an die Geräte
- ❏ EG-Richtlinie 89/336/EWG zu elektromagnetischer Verträglichkeit

  Elektromagnetische Aussendung
  EN 61000-6-3: Wohn-, Gewerbe- und Leichtindustrieumgebungen
  EN 61000-6-4: Industrieumgebung

  Elektromagnetische Störfestigkeit
  EN 61000-6-2: Industrieumgebung
- ❏ Underwriters Laboratories, Inc.
  UL 508 Listed (Industriesteuerungsgeräte) Registrierungsnummer E75310
- ❏ Canadian Standards Association: CSA C22.2 Nummer 142 (Prozesssteuerungsgeräte)
- ❏ Factory Mutual Research: FM Klasse I, Division 2, Gruppen A, B, C, & D Explosionsgefährdete Betriebsstätten, T4A und Klasse I, Zone 2, IIC, T4.

Weitere Informationen zu Normen finden Sie im Anhang A.

Tipp

Die Produktreihe SIMATIC S7-200 entspricht der CSA-Norm.

Das cULus-Logo macht kenntlich, dass die S7-200 von Underwriters Laboratories (UL) gemäß den Normen UL 508 und CSA 22.2 Nr. 142 geprüft und zertifiziert wurde.

Zulassungen für das Seewesen

Die S7-200 Produkte werden regelmäßig für die Zulassungen hinsichtlich bestimmter Märkte und Anwendungen bei bestimmten Behörden eingereicht. Diese Tabelle gibt die Behörden und Zertifikatsnummern der S7-200 Produkte an. Nicht alle S7-200 Produkte in diesem Handbuch tragen eine besondere behördliche Zulassung. Wenden Sie sich an Ihre Siemens Vertretung, wenn Sie eine Liste mit den aktuellen Zulassungen für die einzelnen Bestellnummern benötigen.

Behörde	Zertifikatsnummer
Lloyds Register of Shipping (LRS)	99 / 20018(E1)
American Bureau of Shipping (ABS)	01-HG20020-PDA
Germanischer Lloyd (GL)	12 045 - 98 HH
Det Norske Veritas (DNV)	A-8071
Bureau Veritas (BV)	09051 / A2 BV
Nippon Kaiji Kyokai (NK)	A-534

Verwendung des Handbuchs

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

Das *S7-200 Systemhandbuch* umfasst folgende Themen:

- Kapitel 1 (Produktübersicht) bietet einen Überblick über einige Leistungsmerkmale der Kleinsteuerungen der Familie S7-200.
- Kapitel 2 (Erste Schritte) bietet ein Tutorial zum Erstellen und Laden eines Beispiel-Steuerungsprogramms für eine S7-200.
- Kapitel 3 (Einbauen der S7-200) liefert die Abmessungen und grundlegenden Richtlinien für den Einbau von S7-200 CPUs und E/A-Erweiterungsmodulen.
- Kapitel 4 (SPS-Grundlagen) bietet Informationen zum Betrieb der S7-200.
- Kapitel 5 (Programmiergrundlagen, Konventionen und Funktionen) bietet Informationen zu den Funktionen von STEP 7-Micro/WIN, zu den Programm-Editoren und Arten von Operationen (IEC 1131-3 und SIMATIC), zu den S7-200 Datentypen und den Richtlinien zum Erstellen von Programmen.
- Kapitel 6 (S7-200 Befehlssatz) bietet Beschreibungen und Beispiele der Operationen, die von der S7-200 für die Programmierung unterstützt werden.
- Kapitel 7 (Kommunikation im Netz) bietet Informationen zum Einrichten verschiedener, von der S7-200 unterstützter Netzkonfigurationen.
- Kapitel 8 (Hardware-Fehlerbehebung und Werkzeuge zum Software-Test) bietet Informationen zur Fehlerbehebung der S7-200 Hardware und zu den Funktionen von STEP 7-Micro/WIN, mit denen Sie Ihr Programm testen können.
- Kapitel 9 (Anlegen eines Programms für das Positioniermodul) bietet Informationen zu den Operationen und dem Assistenten, mit denen Sie ein Programm für das Positioniermodul EM 253 anlegen können.
- Kapitel 10 (Anlegen eines Programms für das Modemmodul) bietet Informationen zu den Operationen und dem Assistenten, mit denen Sie ein Programm für das Modemmodul EM 241 anlegen können.
- Kapitel 11 (Steuern eines MicroMaster-Antriebs mit der Bibliothek für das USS-Protokoll) bietet Informationen zu den Operationen, mit denen Sie ein Steuerungsprogramm für einen MicroMaster-Antrieb erstellen können. Außerdem liefert es Informationen zur Konfiguration von MicroMaster 3-Antrieben und MicroMaster 4-Antrieben.
- Kapitel 12 (Bibliothek für das Modbus-Protokoll) bietet Informationen zu den Operationen, mit denen Sie ein Programm mit Modbus-Protokoll für die Kommunikation erstellen können.
- Anhang A (Technische Daten) liefert die technischen Daten und Datenblätter zur S7-200 Hardware.

Die anderen Anhänge liefern weitere Referenzinformationen, z.B. Beschreibungen der Fehlercodes, Beschreibungen der Sondermerker (SM), Teilenummern zum Bestellen von S7-200 Geräten und Ausführungszeiten von AWL-Operationen.

Zusätzliche Informationen und Unterstützung

Informationen zu S7-200 und STEP 7-Micro/WIN

Neben diesem Handbuch bietet STEP 7-Micro/WIN eine umfangreiche Online-Hilfe für den Einstieg in die Programmierung der S7-200. Im Lieferumfang der Software STEP 7-Micro/WIN ist eine kostenlose Dokumentations-CD enthalten. Auf dieser CD finden Sie Anwendungstips, eine elektronische Version dieses Handbuchs und weitere Informationen.

Online-Hilfe

Sie erhalten Hilfe per Tastendruck! Mit F1 rufen Sie die umfangreiche Online-Hilfe von STEP 7-Micro/WIN auf. Die Online-Hilfe umfasst nützliche Informationen für den Einstieg in die Programmierung der S7-200 sowie viele andere Themen.

Elektronisches Handbuch

Eine elektronische Version dieses S7-200 Systemhandbuchs ist auf der Dokumentations-CD verfügbar. Sie können das elektronische Handbuch auf Ihrem Computer installieren, so dass Sie schnellen Zugriff auf die Informationen im Handbuch haben, wenn Sie mit der Software STEP 7-Micro/WIN arbeiten.

Tipps für die Programmierung

Die Dokumentations-CD enthält Tipps für die Programmierung mit Anwendungsbeispielen und Beispielprogrammen. Wenn Sie sich diese Beispiele ansehen oder sie ändern, können Sie effiziente und innovative Lösungen für Ihre eigenen Anwendungen entdecken. Die aktuellen Tipps für die Programmierung sind auch auf der S7-200 Internet-Seite verfügbar.

Internet: www.siemens.com/S7-200

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

- www.ad.siemens.de *Allgemeine Informationen zu Siemens*

  Diese Internet-Seite von Siemens Automation & Drives enthält Informationen zur Produktreihe SIMATIC und anderen Siemens Produkten.

- www.siemens.com/S7-200 *S7-200 Produktinformationen*

  Die S7-200 Internet-Seite bietet Antworten auf häufig gestellte Fragen (FAQs), Tipps für die Programmierung (Anwendungsbeispiele und Beispielprogramme), Informationen zu neu erschienenen Produkten sowie Updates und Downloads zu Produkten.

Technische Unterstützung zu und Erwerb von S7-200 Produkten

Lokale Siemens-Vertretung

Haben Sie technische Fragen oder benötigen Sie Informationen zu Schulungen über S7-200 Produkte bzw. zur Bestellung von S7-200 Produkten, wenden Sie sich bitte an Ihre Siemens-Vertretung. Das technisch geschulte Vertriebspersonal verfügt über sehr spezifische Kenntnisse zu Einsatzmöglichkeiten und Prozessen sowie zu den verschiedenen Siemens-Produkten und kann Ihnen deshalb am schnellsten und besten weiterhelfen, wenn Probleme auftreten.

Technische Dienstleistungen

Das spezifisch geschulte Personal des S7-200 Customer Support steht Ihnen auch bei Fragen und Problemen zur Verfügung. Der Customer Support ist 24 Stunden am Tag, 7 Tage die Woche erreichbar.

- *Anrufe aus den Vereinigten Staaten von Amerika*

Ortszeit:	Montag bis Freitag 8.00 bis 19.00 Uhr Eastern Time
Telefon:	+1 800 241-4453
Fax:	+1 (0) 770 740-3699
E-Mail:	drives.support@sea.siemens.com

- *Anrufe aus Amerika außerhalb der USA*

Ortszeit:	Montag bis Freitag 8.00 bis 19.00 Uhr Eastern Time
Telefon:	+1 (0) 770 740-3505
Fax:	+1 (0) 770 740-3699
E-Mail:	drives.support@sea.siemens.com

- *Anrufe aus Europa und Afrika*

Ortszeit (Nürnberg):	Montag bis Freitag 7.00 bis 17.00 Uhr
Telefon:	+49 (0) 180 5050-222
Fax:	+49 (0) 180 5050-223
E-Mail:	techsupport@ad.siemens.de

- *Anrufe aus Asien und Australien*

Ortszeit (Singapur):	Montag bis Freitag 8.30 bis 17.30 Uhr
Telefon:	+65 (0) 740-7000
Fax:	+65 (0) 740-7001
E-Mail:	drives.support@sae.siemens.com.sg

Inhaltsverzeichnis

1 Produktübersicht 1

S7-200 CPU 2
S7-200 Erweiterungsmodule 3
Programmierpaket STEP 7-Micro/WIN 3
Kommunikationsmöglichkeiten 4
Anzeige-Panels 4

2 Erste Schritte 5

Anschließen der S7-200 CPU 6
Anlegen eines Beispielprogramms 9
Laden des Beispielprogramms 12
Versetzen der S7-200 in den Betriebszustand RUN 12

3 Einbauen der S7-200 13

Richtlinien für den Einbau von S7-200 Geräten 14
Einbau und Ausbau der S7-200 Module 15
Richtlinien für Erdung und Verdrahtung 18

4 SPS-Grundlagen 21

Ausführung der Steuerungslogik durch die S7-200 22
Zugreifen auf Daten der S7-200 24
Speichern und Zurückholen von Daten durch die S7-200 34
Speichern Ihres Programms auf einem Speichermodul 36
Einstellen des Betriebszustands der S7-200 CPU 37
Speichern von Variablenspeicher im EEPROM über das Programm 38
Funktionen der S7-200 39

5 Programmiergrundlagen, Konventionen und Funktionen 47

Richtlinien für das Entwerfen einer Automatisierungslösung mit einer Micro-SPS 48
Wesentliche Elemente eines Programms 49
Erstellen Ihrer Programme mit STEP 7-Micro/WIN 51
Befehlssätze SIMATIC und IEC 1131-3 53
Konventionen in den Programm-Editoren 54
Erstellen Ihres Steuerungsprogramms mit Assistenten 56
Fehlerbehebung in der S7-200 56
Zuordnen von Adressen und Anfangswerten im Datenbaustein-Editor 58
Verwenden der Symboltabelle für die symbolische Adressierung von Variablen 58
Lokale Variablen 59
Beobachten Ihres Programms mit der Statustabelle 59
Erstellen einer Operationsbibliothek 60
Funktionen zum Testen Ihres Programms 60

6 S7-200 Befehlssatz 61

Konventionen für die Beschreibung der Operationen 63
Speicherbereiche und Funktionen der S7-200 64
Bitverknüpfungsoperationen 66
Uhroperationen 74
Kommunikationsoperationen 75
Vergleichsoperationen 90
Umwandlungsoperationen 93

Zähloperationen ... 107
Schnelle Zähler ... 112
Operation Impulsausgabe ... 126
Arithmetische Operationen ... 141
Operation PID-Regler (Proportional/Integral/Differential) ... 146
Interruptoperationen ... 156
Verknüpfungsoperationen ... 163
Übertragungsoperationen ... 166
Operationen für die Programmsteuerung ... 169
Schiebe- und Rotieroperationen ... 180
Zeichenkettenoperationen ... 185
Tabellenoperationen ... 190
Zeitoperationen ... 197
Unterprogrammoperationen ... 204

7 Kommunikation im Netz ... 209

Grundlagen der S7-200 Kommunikation im Netz ... 210
Einstellen des Kommunikationsprotokolls für Ihr Netz ... 214
Installieren und Deinstallieren von Kommunikationsschnittstellen ... 220
Aufbauen des Netzes ... 222
Erstellen von anwenderdefinierten Protokollen in der frei programmierbaren Kommunikation ... 227
Modems und STEP 7-Micro/WIN im Netz ... 229
Für erfahrene Anwender ... 235
Konfigurieren des RS-232/PPI-Multi-Master-Kabels für den entfernten Betrieb ... 241

8 Hardware-Fehlerbehebung und Werkzeuge für den Software-Test ... 245

Funktionen zum Testen Ihres Programms ... 246
Anzeigen des Programmstatus ... 248
Beobachten und Ändern von Daten in der S7-200 mit Hilfe einer Statustabelle ... 249
Forcen von Werten ... 250
Ausführen des Programms für eine bestimmte Anzahl von Zyklen ... 250
Hinweise zur Fehlerbehebung der Hardware ... 251

9 Erstellen eines Programms für das Positioniermodul ... 253

Funktionen des Positioniermoduls ... 254
Konfigurieren des Positioniermoduls ... 256
Vom Positionier-Assistenten erzeugte Positionieroperationen ... 267
Beispielprogramme für das Positioniermodul ... 279
Beobachten des Positioniermoduls mit em EM 253 Steuer-Panel ... 284
Fehlercodes für Positioniermodul und Positionieroperationen ... 286
Für erfahrene Anwender ... 288

10 Erstellen eines Programms für das Modemmodul ... 297

Funktionen des Modemmoduls ... 298
Konfigurieren des Modemmoduls EM mit dem erweiterten Modemassistenten ... 304
Übersicht über Modemoperationen und Einschränkungen ... 308
Operationen für das Modemmodul ... 309
Beispielprogramm für das Modemmodul ... 313
S7-200 CPUs, die intelligente Module unterstützen ... 313
Sondermerker für das Modemmodul ... 314
Für erfahrene Anwender ... 316
Format von Telefonnummern für die Nachrichtenübermittlung ... 318

Format von Textnachrichten 319
Format von CPU-Datenübertragungsnachrichten 320

11 Steuern eines MicroMaster-Antriebs mit der Bibliothek für das USS-Protokoll 321

Anforderungen für den Einsatz des USS-Protokolls 322
Berechnen der Zeit für die Kommunikation mit dem Antrieb 323
Verwenden der USS-Operationen 324
Operationen für das USS-Protokoll 325
Beispielprogramme für das USS-Protokoll 332
Fehlercodes für die Ausführung der USS-Operationen 333
Anschließen und Einrichten des MicroMaster-Antriebs der Serie 3 334
Anschließen und Einrichten des MicroMaster-Antriebs der Serie 4 337

12 Bibliothek für das Modbus-Protokoll 339

Anforderungen für den Einsatz des Modbus-Protokolls 340
Initialisierung und Ausführungszeit des Modbus-Protokolls 340
Modbus-Adressierung 341
Verwenden von Operationen für das Modbus-Slave-Protokoll 342
Operationen für das Modbus-Slave-Protokoll 343

A Technische Daten 347
Allgemeine technische Daten 348
Technische Daten der CPUs 351
Technische Daten der digitalen Erweiterungsmodule 357
Technische Daten der analogen Erweiterungsmodule 363
Technische Daten der Thermoelement- und RTD-Erweiterungsmodule 373
Technische Daten des EM 277 PROFIBUS-DP-Moduls 385
Technische Daten des Modemmoduls EM 241 397
Technische Daten des Positioniermoduls EM 253 399
Technische Daten des Ethernet-Moduls (CP 243-1) 405
Technische Daten des Internet-Moduls (CP 243-1 IT) 407
Technische Daten des AS-Interface-Moduls (CP 243-2) 410
Optionale Steckmodule 412
Steckleitung für Erweiterungsmodule 412
RS-232/PPI-Multi-Master-Kabel und USB/PPI-Multi-Master-Kabel 413
Eingangssimulatoren 417

B Berechnen der Leistungsbilanz 419

C Fehlermeldungen 423
Fehlercodes und Meldungen von schweren Fehlern 424
Fehler zur Laufzeit 425
Verletzungen der Übersetzungsregeln 426

D Sondermerker 427
SMB0: Statusbits 428
SMB1: Statusbits 428
SMB2: Empfangene Zeichen in der frei programmierbaren Kommunikation 429
SMB3: Paritätsfehler in der frei programmierbaren Kommunikation 429
SMB4: Überlauf der Warteschlange 429
SMB5: E/A-Status 430
SMB6: Kennregister der CPU 430
SMB7: Reserviert. 430
SMB8 bis SMB21: Kenn- und Fehlerregister des E/A-Moduls 431
SMW22 bis SMW26: Zykluszeiten 432
SMB28 und SMB29: Analogpotentiometer 432

SMB30 und SMB130: Steuerungsregister der frei programmierbaren Kommunikation 432
SMB31 und SMW32: Schreibsteuerung nullspannungsfester Speicher (EEPROM) 433
SMB34 und SMB35: Intervallregister für zeitgesteuerte Interrupts 433
SMB36 bis SMB65: Register HSC0, HSC1 und HSC2 433
SMB66 bis SMB85: PTO/PWM-Register 435
SMB86 bis SMB94 und SMB186 bis SMB194: Steuerung des Meldungsempfangs 436
SMW98: Fehler im Erweiterungsbus 437
SMB130: Steuerungsregister der frei programmierbaren Kommunikation (siehe SMB30) 437
SMB131 bis SMB165: Register HSC3, HSC4 und HSC5 437
SMB166 bis SMB185: Profildefinitionstabelle PTO0, PTO1 438
SMB186 bis SMB194: Steuerung des Meldungsempfangs (siehe SMB86 bis SMB94) 438
SMB200 bis SMB549: Status intelligentes Modul 439

E S7-200 Bestellnummern 441

F Ausführungszeiten von AWL-Operationen 445

G S7-200 Kurzinformation 451

Index 459

Produktübersicht

1

Die Familie S7-200 umfasst verschiedene Kleinsteuerungen (Micro-SPS), mit denen Sie eine breite Palette von Geräten für Ihre Automatisierungslösungen steuern können.

Die S7-200 beobachtet Eingänge und ändert Ausgänge wie vom Anwenderprogramm gesteuert. Das Anwenderprogramm kann boolesche Verknüpfungen, Zähl- und Zeitfunktionen, komplexe arithmetische Operationen und Kommunikation mit anderen intelligenten Geräten umfassen. Durch das kompakte Design, die flexible Konfiguration und einen leistungsstarken Befehlssatz eignet sich die S7-200 hervorragend für eine große Bandbreite von Steuerungsanwendungen.

In diesem Kapitel

S7-200 CPU 2
S7-200 Erweiterungsmodule 3
Programmierpaket STEP 7-Micro/WIN 3
Kommunikationsmöglichkeiten 4
Anzeige-Panels 4

1

S7-200 CPU

Die S7-200 CPU umfasst einen Mikroprozessor, eine integrierte Spannungsversorgung, Eingangskreise und Ausgangskreise in einem kompakten Gehäuse und bildet eine leistungsstarke Micro-SPS (siehe Bild 1-1). Nachdem Sie Ihr Programm geladen haben, enthält die S7-200 die erforderliche Logik, damit Sie die Eingangs- und Ausgangsgeräte in Ihrer Anwendung beobachten und steuern können.

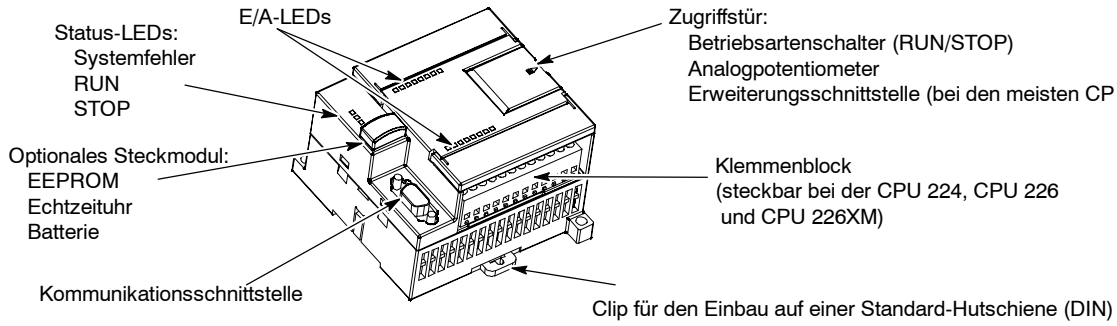


Bild 1-1 Micro-SPS S7-200

Siemens bietet verschiedene Ausführungen der S7-200 CPU mit einer Vielfalt an Leistungsmerkmalen und Funktionen, damit Sie effektive Lösungen für verschiedenste Anwendungen erstellen können. In Tabelle 1-1 werden einige der verschiedenen CPU-Funktionen aufgeführt. Ausführliche Informationen zu bestimmten CPUs finden Sie in Anhang A.

Tabelle 1-1 Vergleich der verschiedenen Ausführungen von S7-200 CPUs

Funktionalität	**CPU 221**	**CPU 222**	**CPU 224**	**CPU 226**	**CPU 226XM**
Abmessungen (mm)	90 x 80 x 62	90 x 80 x 62	120,5 x 80 x 62	190 x 80 x 62	190 x 80 x 62
Programmspeicher	4096 Bytes	4096 Bytes	8192 Bytes	8192 Bytes	16384 Bytes
Datenspeicher	2048 Bytes	2048 Bytes	5120 Bytes	5120 Bytes	10240 Bytes
Speicher-Backup	typ. 50 Std.	typ. 50 Std.	typ. 190 Std.	typ. 190 Std.	typ. 190 Std.
Integrierte E/A	6 Ein-/ 4 Ausgänge	8 Ein-/ 6 Ausgänge	14 Ein-/ 10 Ausgänge	24 Ein-/ 16 Ausgänge	24 Ein-/ 16 Ausgänge
Erweiterungsmodule	0 Module[1]	2 Module[1]	7 Module[1]	7 Module[1]	7 Module[1]
Schnelle Zähler Einphasig Zweiphasig	 4 bei 30 kHz 2 bei 20 kHz	 4 bei 30 kHz 2 bei 20 kHz	 6 bei 30 kHz 4 bei 20 kHz	 6 bei 30 kHz 4 bei 20 kHz	 6 bei 30 kHz 4 bei 20 kHz
Impulsausgänge (DC)	2 bei 20 kHz	2 bei 20 kHz	2 bei 20 kHz	2 bei 20 kHz	2 bei 20 kHz
Analogpotentiometer	1	1	2	2	2
Echtzeituhr	Steckmodul	Steckmodul	Integriert	Integriert	Integriert
Kommunikations-schnittstellen	1 RS-485	1 RS-485	1 RS-485	2 RS-485	2 RS-485
Gleitpunktarithmetik	ja				
Digitale E/A (Prozessabbild)	256 (128 Eingänge, 128 Ausgänge)				
Boolesche Ausführungszeiten	0,37 Mikrosekunden/Operation				

[1] Sie müssen die Leistungsbilanz berechnen, um zu ermitteln, wieviel Leistung (bzw. Strom) die S7-200 CPU für Ihre Konfiguration liefern kann. Wird die Leistungsbilanz der CPU überschritten, können Sie evtl. nicht die maximal zulässige Anzahl Module anschließen. Im Anhang A finden Sie den Leistungsbedarf der CPUs und der Erweiterungsmodule, und im Anhang B können Sie die Leistungsbilanz berechnen.

S7-200 Erweiterungsmodule

Damit Sie die Anforderungen Ihrer Anwendungen besser erfüllen können, gibt es in der S7-200 Familie eine Vielzahl an Erweiterungsmodulen. Mit diesen Erweiterungsmodulen können Sie die S7-200 CPU um zusätzliche Funktionalität erweitern. Tabelle 1-2 zeigt eine Liste der Erweiterungsmodule, die zur Zeit erhältlich sind. Ausführliche Informationen zu bestimmten Modulen finden Sie in Anhang A.

Tabelle 1-2 S7-200 Erweiterungsmodule

Erweiterungsmodule		**Arten**		
Digitale Module	Eingabe	8 x DC-Eingänge	8 x AC-Eingänge	16 x DC-Eingänge
	Ausgang	4 x DC 8 x DC-Ausgänge	4 x Relais 8 x AC-Ausgänge	 8 x Relais
	Kombination	4 x DC-Eing./4 x DC-Ausg. 4 x DC-Eing./4 x Relais	8 x DC-Eing./8 x DC-Ausg. 8 x DC-Eing./8 x Relais	16 x DC-Eing./16 x DC-Ausg. 16 x DC-Eingänge / 16 x Relais
Analogmodule	Eingabe	4 x Analogeingänge	4 x Thermoelement-Eing.	2 x RTD-Eingänge
	Ausgabe	2 x Analogausgänge		
	Kombination	4 x Analogeingänge / 1 Analogausgang		
Intelligente Module		Positionierung Ethernet	Modem Internet	PROFIBUS-DP
Sonstige Module		AS-Interface		

Programmierpaket STEP 7-Micro/WIN

Das Programmierpaket STEP 7-Micro/WIN bietet eine bedienerfreundliche Umgebung zum Entwickeln, Bearbeiten und Beobachten der Logik zur Steuerung Ihrer Anwendung. STEP 7-Micro/WIN verfügt über drei Programm-Editoren, mit denen Sie das Steuerungsprogramm für Ihre Anwendung bequem und effizient entwickeln können. Damit Sie die Informationen finden, die Sie benötigen, bietet STEP 7-Micro/WIN eine umfangreiche Online-Hilfe und eine Dokumentations-CD mit einer elektronischen Version dieses Handbuchs, mit Anwendungstips und anderen nützlichen Informationen.

Voraussetzungen an den PC

STEP 7-Micro/WIN läuft auf PCs und auf Siemens Programmiergeräten, z.B. einem PG 760. Ihr PC bzw. Ihr Programmiergerät muss die folgenden Mindestvoraussetzungen erfüllen:

- ❏ Betriebssystem:
  Windows 95, Windows 98, Windows 2000,
  Windows ME (Millennium Edition),
  Windows NT 4.0 (oder höher),
  Windows XP Professional
- ❏ Mindestens 100 MB freier Speicherplatz auf der Festplatte
- ❏ Maus (empfohlen)

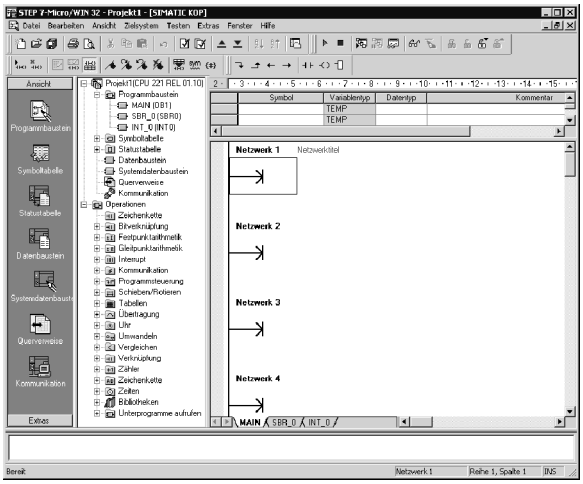


Bild 1-2 STEP 7-Micro/WIN

1

Installieren von STEP 7-Micro/WIN

Legen Sie die STEP 7-Micro/WIN CD in das CD-ROM-Laufwerk Ihres PC ein. Das Installationsprogramm wird automatisch gestartet und führt Sie durch den Installationsvorgang. Weitere Informationen zum Installieren von STEP 7-Micro/WIN finden Sie in der Liesmich-Datei.

Tipp
Zum Installieren von STEP 7-Micro/WIN auf einem PC mit einem der Betriebssysteme Windows NT, Windows 2000 oder Windows XP Professional müssen Sie sich mit Administratorrechten anmelden.

Kommunikationsmöglichkeiten

Siemens bietet zwei Programmiermöglichkeiten, um Ihren PC mit Ihrer S7-200 zu verbinden: eine direkte Verbindung mit einem PPI-Multi-Master-Kabel oder eine Kommunikationsprozessorkarte (CP) mit einem MPI-Kabel.

Das PPI-Multi-Master-Programmierkabel ist die am häufigsten eingesetzte und günstigste Art, Ihren PC an die S7-200 anzuschließen. Dieses Kabel verbindet die Kommunikationsschnittstelle der S7-200 mit der seriellen Kommunikationsschnittstelle Ihres PC. Das PPI-Multi-Master-Programmierkabel kann auch zum Anschließen anderer Kommunikationsgeräte an die S7-200 dienen.

Anzeige-Panels

Textdisplay TD 200

Das TD 200 ist ein zweizeiliges Textdisplay mit 20 Zeichen je Zeile, das an die S7-200 angeschlossen werden kann. Mit dem Assistenten für das TD 200 können Sie Ihre S7-200 auf einfache Weise so programmieren, dass Textmeldungen und andere Daten aus Ihrer Anwendung angezeigt werden.

Das TD 200 ist eine preisgünstige Schnittstelle zu Ihrer Anwendung, mit der Sie die Prozessvariablen Ihrer Anwendung ansehen, beobachten und ändern können.

Die Funktionen und technischen Daten des TD 200 werden in einem getrennten Handbuch beschrieben.

TD 200 Assistent

Mit dem TD 200-Assistent von STEP 7-MicroWIN können Sie Meldungen für das TD 200 schnell und einfach konfigurieren. Zum Aufrufen des Assistenten für das TD 200 wählen Sie den Menübefehl **Extras > TD 200-Assistent**.

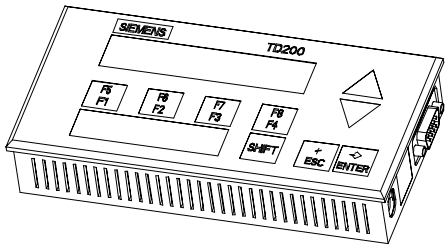


Bild 1-3 Textdisplay TD 200

Touch Panel TP070

Das Touch Panel TP070 ist ein Anzeigegerät, das an die S7-200 angeschlossen werden kann. Mit diesem Touch Panel können Sie Ihre Bedienerschnittstelle benutzerspezifisch einstellen.

Das TP070 kann benutzerspezifische Grafiken, Bildlauffelder, Anwendungsvariablen, Schaltflächen usw. auf einem bedienerfreundlichen Touch Panel anzeigen.

Die optionale CD TP-Designer für TP070, Version 1.0 bietet die Software TP Designer, mit der Sie Ihr TP070 programmieren können.

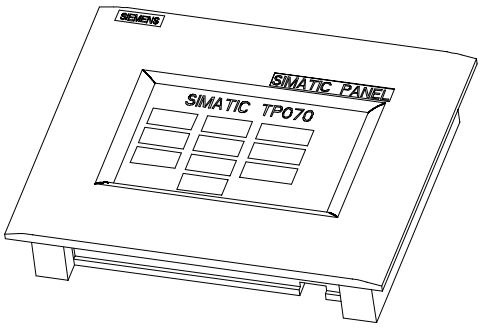


Bild 1-4 TP070 Touch Panel

Erste Schritte

2

Mit STEP 7-Micro/WIN können Sie auf einfache Weise Ihre S7-200 programmieren. Nach nur einigen kurzen Schritten in einem einfachen Beispiel wissen Sie, wie Sie Ihre S7-200 anschließen, programmieren und betreiben.

Für dieses Beispiel benötigen Sie nur ein PPI-Multi-Master-Kabel, eine S7-200 CPU und ein Programmiergerät, auf dem die Programmiersoftware STEP 7-Micro/WIN installiert ist.

In diesem Kapitel

Anschließen der S7-200 CPU 6
Anlegen eines Beispielprogramms 9
Laden des Beispielprogramms 12
Versetzen der S7-200 in den Betriebszustand RUN 12

Anschließen der S7-200 CPU

2

Das Anschließen der S7-200 ist ganz einfach. In diesem Beispiel müssen Sie nur die Spannungsversorgung an die S7-200 CPU anschließen und dann das Kommunikationskabel an das Programmiergerät und an die S7-200 CPU anschließen.

Anschließen der Spannungsversorgung der S7-200 CPU

Als erstes schließen Sie die S7-200 an eine Spannungsversorgung an. Bild 2-1 zeigt die Verdrahtung für eine DC-Ausführung oder eine AC-Ausführung der S7-200 CPU.

Bevor Sie ein elektrisches Gerät einbauen oder ausbauen, müssen Sie sicherstellen, dass die Spannungsversorgung der Geräte ausgeschaltet ist. Treffen Sie alle notwendigen Sicherheitsvorkehrungen und vergewissern Sie sich, dass vor dem Einbau bzw. Ausbau der S7-200 die Spannungsversorgung der S7-200 abgeschaltet ist.

Warnung

Wenn Sie versuchen, die S7-200 oder daran angeschlossene Geräte in eingeschaltetem Zustand einzubauen oder zu verdrahten, kann es passieren, dass Sie einen elektrischen Schlag bekommen oder die Geräte fehlerhaft arbeiten. Ist die Spannungsversorgung der S7-200 und aller daran angeschlossenen Geräte während des Einbaus 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 Einbau bzw. Ausbau der S7-200 oder daran angeschlossener Geräte die Spannungsversorgung der S7-200 abgeschaltet ist.

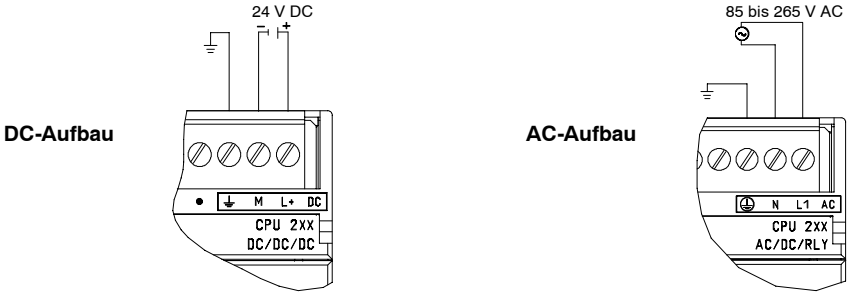


Bild 2-1 Anschließen der Spannungsversorgung der S7-200 CPU

Tipp

In den Beispielen in diesem Handbuch wird das RS-232/PPI-Multi-Master-Kabel verwendet. Das RS-232/PPI-Multi-Master-Kabel ersetzt das frühere PC/PPI-Kabel. Außerdem ist ein USB/PPI-Multi-Master-Kabel erhältlich. Die Bestellnummern finden Sie im Anhang E.

Anschließen des RS-232/PPI-Multi-Master-Kabels

Bild 2-2 zeigt ein RS-232/PPI-Multi-Master-Kabel, das die S7-200 mit dem Programmiergerät verbindet. So schließen Sie das Kabel an:

1. Schließen Sie den RS-232-Stecker (mit “PC” gekennzeichnet) des RS-232/PPI-Multi-Master-Kabels an die Kommunikationsschnittstelle des Programmiergeräts an. (Verwenden Sie in diesem Beispiel COM 1.)
2. Schließen Sie den RS-485-Stecker (mit “PPI” gekennzeichnet) des RS-232/PPI-Multi-Master-Kabels an die Schnittstelle 0 oder an die Schnittstelle 1 der S7-200 an.
3. Achten Sie darauf, dass die DIP-Schalter des RS-232/PPI-Multi-Master-Kabels wie in Bild 2-2 eingestellt sind.

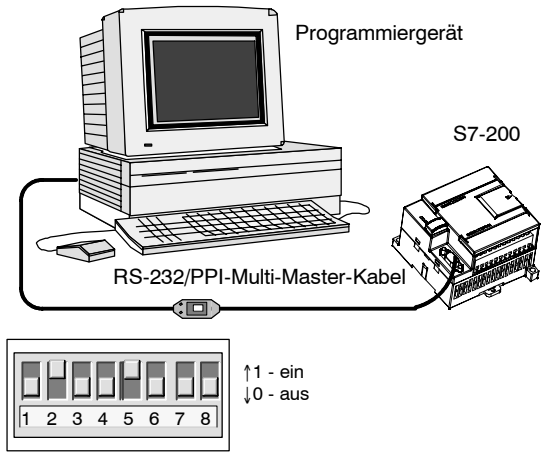


Bild 2-2 Anschließen des RS-232/PPI-Multi-Master-Kabels

Aufrufen von STEP 7-Micro/WIN

Klicken Sie auf das Symbol für STEP 7-Micro/WIN, um ein neues Projekt anzulegen. Bild 2-3 zeigt ein neues Projekt.

Beachten Sie die Navigationsleiste. Mit den Symbolen in der Navigationsleiste können Sie einzelne Elemente des STEP 7-Micro/WIN Projekts öffnen.

Klicken Sie in der Navigationsleiste auf das Symbol für die Kommunikation, um das Dialogfeld ”Kommunikation” aufzurufen. In diesem Dialogfeld richten Sie die Kommunikation für STEP 7-Micro/WIN ein.

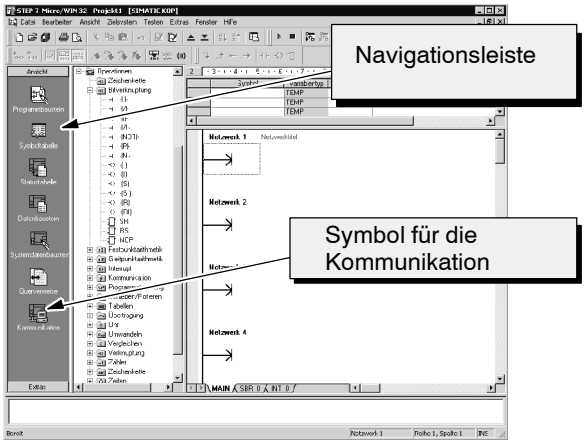


Bild 2-3 Neues STEP 7-Micro/WIN Projekt

2

Prüfen der Kommunikationsparameter für STEP 7-Micro/WIN

Das Beispielprojekt nutzt die Voreinstellungen für STEP 7-Micro/WIN und das RS-232/PPI-Multi-Master-Kabel. So überprüfen Sie diese Einstellungen:

1. Prüfen Sie, dass für die Adresse des PC/PPI-Kabels im Dialogfeld ”Kommunikation” der Wert 0 eingestellt ist.
2. Prüfen Sie, dass als Schnittstelle für die Netzparameter das PC/PPI-Kabel (COM1) eingestellt ist.
3. Prüfen Sie, dass für die Übertragungsgeschwindigkeit 9,6 kBps eingestellt ist.

Wenn Sie die Einstellungen der Kommunikationsparameter ändern müssen, lesen Sie Kapitel 7.

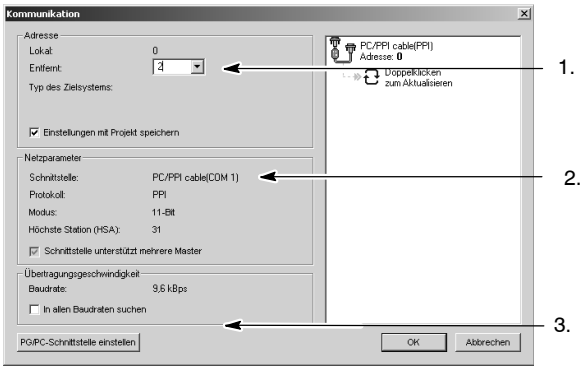


Bild 2-4 Prüfen der Kommunikationsparameter

Aufbauen der Kommunikation mit der S7-200

Im Dialogfeld ”Kommunikation” stellen Sie die Verbindung zur S7-200 CPU her:

1. Doppelklicken Sie im Dialogfeld ”Kommunikation” auf die Schaltfläche ”Aktualisieren”.

   STEP 7-Micro/WIN sucht nach der S7-200 Station und zeigt ein CPU-Symbol für die angeschlossene S7-200 Station an.
2. Wählen Sie die S7-200 und bestätigen Sie mit ”OK”.

Wenn STEP 7-Micro/WIN Ihre S7-200 CPU nicht findet, prüfen Sie die Einstellungen der Kommunikationsparameter und wiederholen Sie diese Schritte.

Nachdem Sie die Kommunikation mit der S7-200 aufgebaut haben, können Sie das Beispielprogramm erstellen und laden.

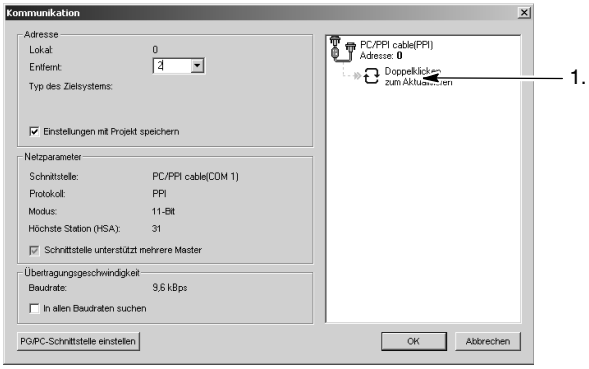


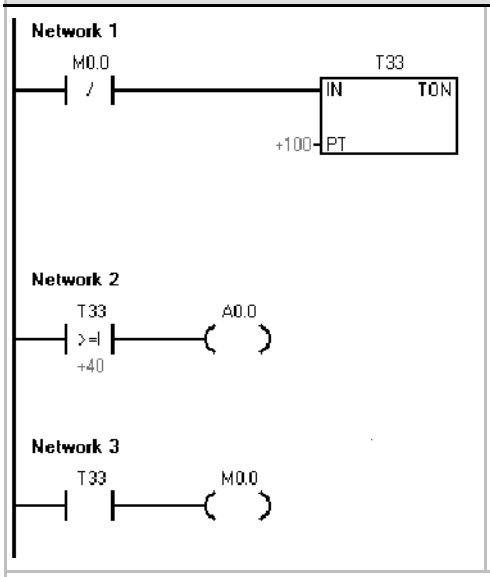
Bild 2-5 Aufbauen der Kommunikation zur S7-200

Anlegen eines Beispielprogramms

Wenn Sie dieses Beispiel für ein Steuerungsprogramm eingeben, werden Sie besser verstehen, wie einfach es ist, in STEP 7-Micro/WIN zu arbeiten. Dieses Programm enthält sechs Operationen in drei Netzwerken und bildet daraus eine sehr einfache Zeit, die von selbst startet und sich selbst zurücksetzt.

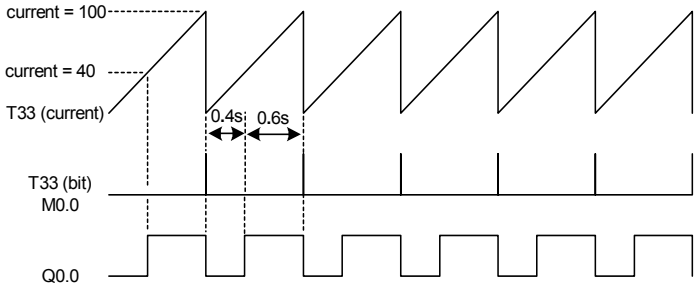
Die Operationen für dieses Beispielprogramm geben Sie im KOP-Editor ein. Das folgende Beispiel zeigt das gesamte Programm im Kontaktplan (KOP) und in der Anweisungsliste (AWL). Die Netzwerkkommentare im AWL-Programm erläutern die Logik der einzelnen Netzwerke. Das Impulsdiagramm zeigt den Betrieb des Programms.

Beispiel: Beispielprogramm für den Einstieg in STEP 7-Micro/WIN




```
Netzwerk 1      //10-ms-Zeit T33 läuft nach (100 x 10 ms = 1 s) ab
                //Impuls M0.0 ist zu schnell für die Beobachtung in der
                //Statusansicht.
LDN     M0.0
TON     T33, +100
Netzwerk 2      //Der Vergleich wird mit einer Geschwindigkeit wahr,
                //die in der Statusansicht sichtbar ist. A0.0 nach
                //(40 x 10 ms = 0,4 s) einschalten, um eine Wellen-
form
                //von 40% aus/60% ein zu erhalten.
LDW>=   T33, +40
=       A0.0
Netzwerk 3      //Impuls T33 (Bit) zu schnell für die Beobachtung in
                //der Statusansicht.
                //Zeit über M0.0 zurücksetzen, nach Ablauf von
                //(100 x 10 ms  = 1 s).
LD      T33
=       M0.0
```

Impulsdiagramm



Aufrufen des Programm-Editors

Klicken Sie auf das Symbol des Programmbausteins, um den Programm-Editor zu öffnen (siehe Bild 2-6).

Beachten Sie den Operationsbaum und den Programm-Editor. Mit dem Operationsbaum fügen Sie KOP-Operationen in die Netzwerke im Programm-Editor ein, indem Sie die Operationen mit der Maus aus dem Operationsbaum in die Netzwerke ziehen.

Die Symbole in der Funktionsleiste bieten Verknüpfungen zu den Menübefehlen.

Nachdem Sie das Programm eingegeben und gespeichert haben, können Sie es in die S7-200 laden.

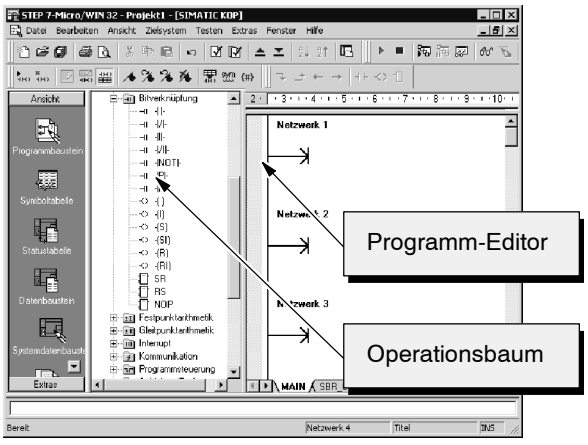


Bild 2-6 Fenster in STEP 7-Micro/WIN

Eingeben von Netzwerk 1: Starten der Zeit

Ist M0.0 ausgeschaltet (0), wird dieser Kontakt eingeschaltet und liefert den Signalfluss zum Starten der Zeit. So geben Sie den Kontakt für M0.0 ein:

1. Doppelklicken Sie entweder auf das Symbol für Bitverknüpfungen oder klicken Sie auf das Pluszeichen (+), um die Bitverknüpfungsoperationen anzuzeigen.
2. Wählen Sie den Öffnerkontakt.
3. Halten Sie die linke Maustaste gedrückt und ziehen Sie den Kontakt ins erste Netzwerk.
4. Klicken Sie auf die drei “???” über dem Kontakt und geben Sie die folgende Adresse ein: M0.0
5. Drücken Sie die Eingabetaste, um die Adresse für den Kontakt einzugeben.

So geben Sie die Zeitoperation für T33 ein:

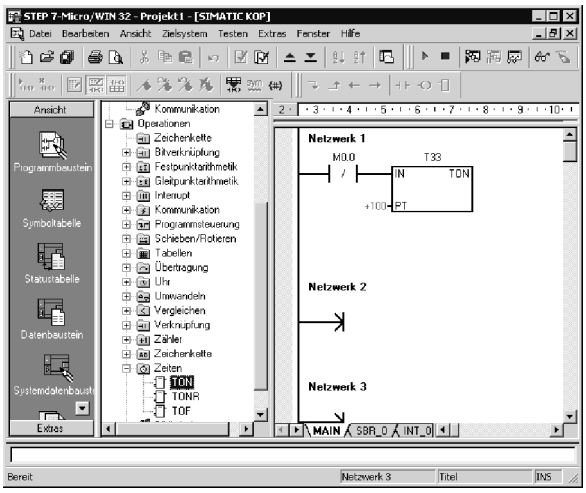


Bild 2-7 Netzwerk 1

1. Doppelklicken Sie auf das Symbol für Zeiten, um die Zeitoperationen anzuzeigen.
2. Wählen Sie TON (Zeit als Einschaltverzögerung starten).
3. Halten Sie die linke Maustaste gedrückt und ziehen Sie die Zeit ins erste Netzwerk.
4. Klicken Sie auf die drei “???” über der Zeit-Box und geben Sie die folgende Nummer für die Zeit ein: T33.
5. Drücken Sie die Eingabetaste, um die Nummer der Zeit einzugeben und den Fokus auf den Parameter (PT) für die Voreinstellung der Zeit zu verschieben.
6. Geben Sie den folgenden Wert für die Voreinstellung der Zeit ein: 100.
7. Drücken Sie die Eingabetaste, um den Wert einzugeben.

Eingeben von Netzwerk 2: Einschalten des Ausgangs

Ist der Zeitwert von T33 größer oder gleich 40 (40 mal 10 Millisekunden bzw. 0,4 Sekunden), liefert der Kontakt den Signalfluss, der den Ausgang A0.0 der S7-200 einschaltet. So geben Sie die Vergleichsoperation ein:

1. Doppelklicken Sie auf das Symbol für Vergleicher, um die Vergleichsoperationen anzuzeigen. Wählen Sie die Operation >=I (Größer oder gleich ganze Zahl).
2. Halten Sie die linke Maustaste gedrückt und ziehen Sie die Vergleichsoperation ins zweite Netzwerk.
3. Klicken Sie auf die drei “???” über dem Kontakt und geben Sie die Adresse für den Zeitwert ein: T33.
4. Drücken Sie die Eingabetaste, um die Nummer der Zeit einzugeben und um den Fokus auf den anderen Wert zu verschieben, der mit dem Zeitwert verglichen werden soll.
5. Geben Sie den folgenden Wert ein, der mit dem Zeitwert verglichen werden soll: 40.
6. Drücken Sie die Eingabetaste, um den Wert einzugeben.

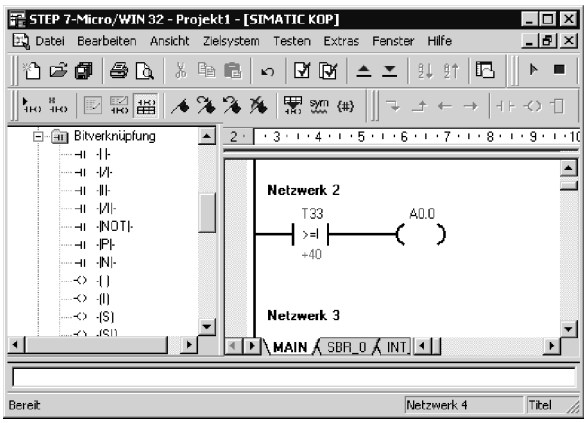


Bild 2-8 Netzwerk 2

So geben Sie die Operation zum Einschalten von Ausgang A0.0 ein:

1. Doppelklicken Sie auf das Symbol für Bitverknüpfungen, um die Bitverknüpfungsoperationen anzuzeigen, und wählen Sie die Ausgangsspule.
2. Halten Sie die linke Maustaste gedrückt und ziehen Sie die Spule ins zweite Netzwerk.
3. Klicken Sie auf die drei “???” über der Spule und geben Sie die folgende Adresse ein: A0.0
4. Drücken Sie die Eingabetaste, um die Adresse für die Spule einzugeben.

Eingeben von Netzwerk 3: Rücksetzen der Zeit

Erreicht die Zeit den voreingestellten Wert (100) und schaltet das Zeitbit ein, wird der Kontakt für T33 eingeschaltet. Der Signalfluss von diesem Kontakt schaltet den Merker M0.0 ein. Weil die Zeit von einem Öffnerkontakt für M0.0 aktiviert wird, wird die Zeit zurückgesetzt, wenn der Zustand von M0.0 von aus (0) nach ein (1) wechselt.

So geben Sie den Kontakt für das Zeitbit von T33 ein:

1. Wählen Sie den Schließerkontakt aus den Bitverknüpfungsoperationen.
2. Halten Sie die linke Maustaste gedrückt und ziehen Sie den Kontakt ins dritte Netzwerk.
3. Klicken Sie auf die drei “???” über dem Kontakt und geben Sie die Adresse des Zeitbits ein: T33.
4. Drücken Sie die Eingabetaste, um die Adresse für den Kontakt einzugeben.

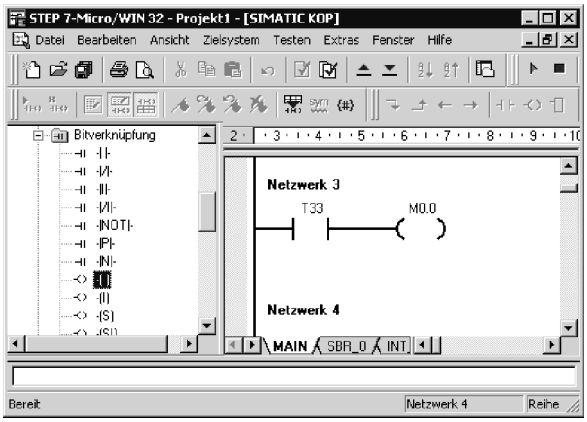


Bild 2-9 Netzwerk 3

So geben Sie die Spule zum Einschalten von M0.0 ein:

1. Wählen Sie die Ausgangsspule aus den Bitverknüpfungsoperationen.
2. Halten Sie die linke Maustaste gedrückt und ziehen Sie die Ausgangsspule ins dritte Netzwerk.
3. Doppelklicken Sie auf die drei “???” über der Spule und geben Sie die folgende Adresse ein: M0.0
4. Drücken Sie die Eingabetaste, um die Adresse für die Spule einzugeben.

Speichern des Beispielprojekts

Nachdem Sie die drei Netzwerke mit Operationen eingegeben haben, haben Sie die Programmeingabe beendet. Wenn Sie das Programm speichern, erstellen Sie ein Projekt, das den S7-200 CPU-Typ und andere Parameter umfasst. So speichern Sie das Projekt:

1. Wählen Sie in der Menüleiste den Menübefehl **Datei > Speichern unter**.
2. Geben Sie im Dialogfeld ”Speichern unter” einen Namen für das Projekt ein.
3. Speichern Sie das Projekt mit ”OK”.

Nachdem Sie das Projekt gespeichert haben, können Sie das Programm in die S7-200 laden.

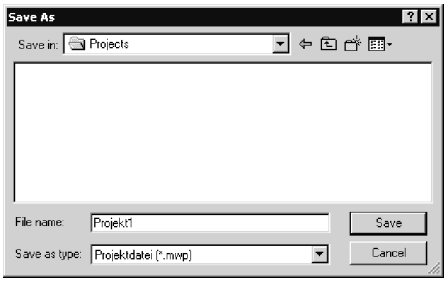


Bild 2-10 Speichern des Beispielprogramms

Laden des Beispielprogramms

Tipp

Jedes Projekt in STEP 7-Micro/WIN ist einem CPU-Modell zugeordnet (CPU 221, CPU 222, CPU 224, CPU 226 oder CPU 226XM). Entspricht die im Projekt eingestellte CPU nicht der angeschlossenen CPU, zeigt STEP 7-Micro/WIN diese Nicht-Übereinstimmung an und fordert Sie auf, entsprechende Maßnahmen einzuleiten. Wählen Sie in diesem Fall bei diesem Beispiel “Ladevorgang fortsetzen”.

1. Klicken Sie in der Funktionsleiste auf das Symbol ”Laden in CPU” oder wählen Sie den Menübefehl **Datei > Laden in CPU**, um das Programm zu laden (siehe Bild 2-11).
2. Klicken Sie auf ”OK”, um die Elemente des Programms in die S7-200 zu laden.

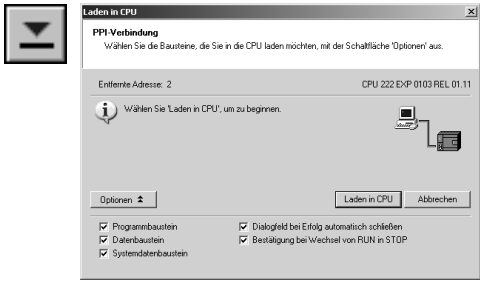


Bild 2-11 Laden des Programms

Befindet sich Ihre S7-200 im Betriebszustand RUN,

werden Sie in einer Meldung aufgefordert, die S7-200 in STOP zu versetzen. Bestätigen Sie mit ”Ja”, um die S7-200 in den Betriebszustand STOP zu versetzen.

Versetzen der S7-200 in den Betriebszustand RUN

Damit STEP 7-Micro/WIN die S7-200 CPU in den Betriebszustand RUN versetzen kann, muss sich der Betriebsartenschalter der S7-200 in Stellung TERM oder RUN befinden. Wenn Sie die S7-200 in den Betriebszustand RUN versetzen, führt die S7-200 das Programm aus:

1. Wählen Sie in der Funktionsleiste das Symbol ”RUN” oder wählen Sie den Menübefehl **Zielsystem > RUN**.
2. Klicken Sie auf ”OK”, um den Betriebszustand der S7-200 zu wechseln.

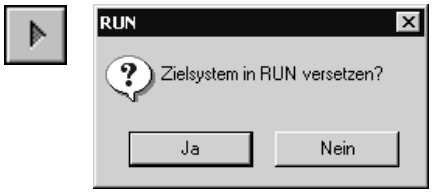


Bild 2-12 Versetzen der S7-200 in den Betriebszustand RUN

Wenn die S7-200 in den Betriebszustand RUN geht, wird die Ausgangs-LED für A0.0 ein- und ausgeschaltet, je nachdem wie die S7-200 das Programm ausführt.

Herzlichen Glückwunsch! Sie haben gerade Ihr erstes S7-200 Programm erstellt.

Sie können das Programm mit dem Menübefehl **Testen > Programmstatus** beobachten.
STEP 7-Micro/WIN zeigt die Werte der Operationen an. Wenn Sie das Programm stoppen möchten, versetzen Sie die S7-200 in den Betriebszustand STOP, indem Sie auf das Symbol ”STOP” klicken oder den Menübefehl **Zielsystem > STOP** wählen.

Einbauen der S7-200

3

Die S7-200 Geräte wurden so ausgelegt, dass sie einfach einzubauen 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 aufbauen und verdrahten.

In diesem Kapitel

Richtlinien für den Einbau von S7-200 Geräten 14
Einbau und Ausbau der S7-200 Module 15
Richtlinien für Erdung und Verdrahtung 18

Richtlinien für den Einbau von S7-200 Geräten

Sie können eine S7-200 entweder in einer Schalttafel oder auf einer Standard-Hutschiene einbauen. Sie können die S7-200 horizontal oder vertikal einbauen.

Halten Sie die S7-200 Geräte fern von Wärme, Hochspannung und elektrischen Störungen

Als allgemeine Regel für die Anordnung von Geräten in Ihrem System gilt, dass Sie Geräte, die Hochspannung oder hohe elektrische Störungen erzeugen, von den elektronischen Niederspannungsgeräten wie der S7-200 fernhalten.

Wenn Sie das Layout der S7-200 in Ihrer Schalttafel planen, berücksichtigen Sie wärmeerzeugende Geräte und ordnen Sie die elektronischen Geräte in den kühleren Bereichen Ihres Schaltschranks an. Wenn Sie ein elektronisches Gerät in einer Umgebung mit hohen Temperaturen betreiben, verringert sich die Zeitspanne bis zum Ausfall.

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

Lassen Sie genügend Abstand für Kühlung und Verdrahtung

Die S7-200 Geräte sind für natürliche Wärmeabfuhr durch Konvektion ausgelegt. Lassen Sie deshalb oberhalb und unterhalb der Geräte jeweils mindestens 25 mm Platz, um die Wärmeabfuhr zu gewährleisten. Sorgen Sie außerdem für eine Mindesteinbautiefe von 75 mm.

Tipp

Bei vertikalem Einbau ist die maximal zulässige Umgebungstemperatur um 10 °C niedriger. Montieren Sie die S7-200 CPU unterhalb der Erweiterungsmodule.

Wenn Sie das Layout für Ihr S7-200 System planen, lassen Sie genügend Abstand für die Verdrahtung und die Kommunikationskabelanschlüsse. Zusätzliche Flexibilität beim Konfigurieren des Layouts Ihres S7-200 Systems erhalten Sie durch den Einsatz von Steckleitungen.

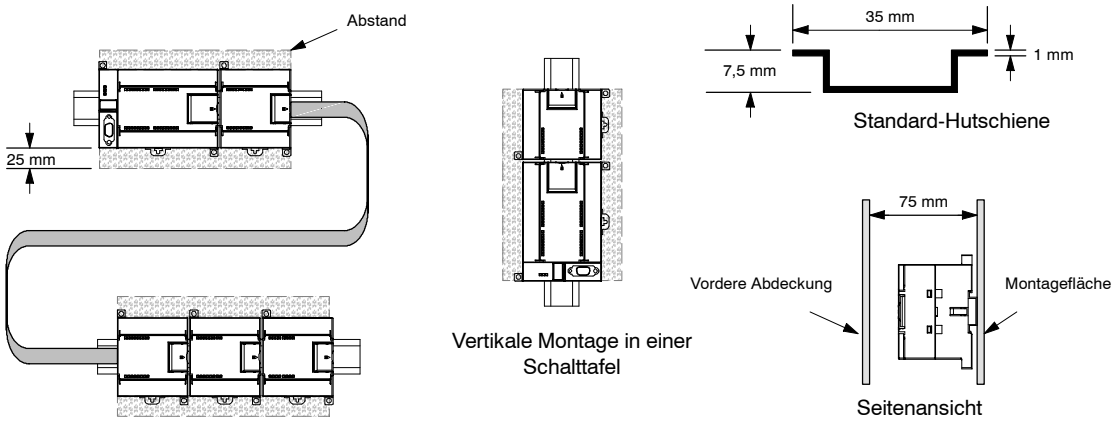


Bild 3-1 Montagemöglichkeiten, Ausrichtung und Abstand

Leistungsbilanz

Alle S7-200 CPUs besitzen eine interne Spannungsversorgung, die neben der CPU die Erweiterungsmodule und andere 24-V-DC-Verbraucher speist.

Die S7-200 CPU liefert eine Spannung von 5 V DC für alle Erweiterungen in Ihrem System. Achten Sie sorgfältig darauf, dass die CPU in Ihrer Systemkonfiguration eine Spannung von 5 V für die Erweiterungsmodule liefern kann. Benötigt Ihre Konfiguration mehr Leistung als die CPU liefern kann, müssen Sie ein Modul entfernen oder eine CPU mit größerer Leistung wählen. Im Anhang A finden Sie Informationen zur 5-V-DC-Leistungsversorgung Ihrer S7-200 CPU und dem 5-V-DC-Leistungsbedarf der Erweiterungsmodule. Mit Hilfe der Informationen in Anhang B können Sie berechnen, wieviel Leistung (bzw. Strom) die CPU für Ihre Konfiguration liefern kann.

Alle S7-200 CPUs verfügen auch über eine 24-V-DC-Geberversorgung, die 24 V DC für Eingänge, für die Versorgung der Relaisspulen der Erweiterungsmodule und für sonstige Verbraucher liefert. Wenn der Leistungsbedarf die Leistung der Geberversorgung übersteigt, dann müssen Sie eine externe 24-V-DC-Spannungsversorgung an Ihr System anschließen. Im Anhang A finden Sie die Leistungsbilanz für die 24-V-DC-Geberversorgung für Ihre jeweilige S7-200 CPU.

Wenn Sie eine externe 24-V-DC-Spannungsversorgung benötigen, müssen Sie darauf achten, dass die Spannungsversorgung nicht parallel zur Geberversorgung der S7-200 CPU angeschlossen ist. Die beste Störfestigkeit erreichen Sie, wenn Sie die Leitungen (M) der jeweiligen Spannungsversorgungen miteinander verbinden.

Warnung

Wenn Sie parallel zu der DC-Geberversorgung der S7-200 eine externe 24-V-DC-Spannungsversorgung 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.

Einbau und Ausbau der S7-200 Module

Die S7-200 kann auf einfache Weise auf einer Standard-Hutschiene oder in einer Schalttafel eingebaut werden.

Voraussetzungen

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

Warnung

Wenn Sie versuchen, eine S7-200 oder andere Geräte in eingeschaltetem Zustand ein- oder auszubauen, kann es passieren, dass Sie einen elektrischen Schlag bekommen oder die Geräte fehlerhaft arbeiten.

Ist die Spannungsversorgung der S7-200 und aller daran angeschlossenen Geräte während des Einbaus 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 Einbau bzw. Ausbau eines Geräts die Spannungsversorgung für die S7-200 abgeschaltet ist.

Achten Sie immer darauf, dass Sie das richtige Modul bzw. das richtige Gerät verwenden, wenn Sie ein S7-200 Gerät einbauen bzw. auswechseln.

Warnung

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

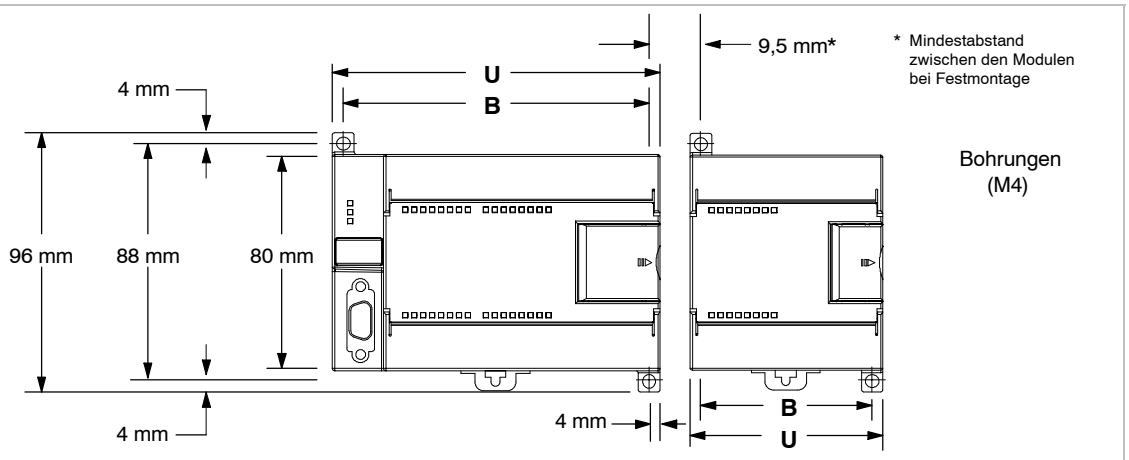
Wird ein S7-200 Gerät durch eine andere Variante ersetzt, nicht richtig ausgerichtet oder in der falschen Reihenfolge eingebaut, so kann dies zu tödlichen oder schweren Verletzungen und/oder Sachschaden führen.

Wechseln Sie ein S7-200 Gerät immer mit der gleichen Ausführung aus, richten Sie das Gerät korrekt aus und bauen Sie es an der richtigen Stelle ein.

Montageabmessungen

Die S7-200 CPUs und die Erweiterungsmodule sind mit Bohrungen versehen, die den Einbau in eine Schalttafel erleichtern. Die Montageabmessungen finden Sie in Tabelle 3-1.

Tabelle 3-1 Montageabmessungen



S7-200 Modul		Breite A	Breite B
CPU 221 und CPU 222		90 mm	82 mm
CPU 224		120,5 mm	112,5 mm
CPU 226 und CPU 226XM		196 mm	188 mm
Erweiterungsmodule:	4 bzw. 8 DC- und Relais-E/A (8E, 4A, 8A, 4E/4A) und Analogausgänge (2 AA)	46 mm	38 mm
Erweiterungsmodule:	16 digitale E/A (16E, 8E/8A), analoge E/A (4AE, 4AE/1AA), RTD, Thermoelement, PROFIBUS, Ethernet, Internet, AS-Interface, 8 AC (8E und 8A), Positionierung und Modem	71,2 mm	63,2 mm
Erweiterungsmodule:	32 digitale E/A (16E/16A)	137,3 mm	129,3 mm

Einbau einer CPU oder eines Erweiterungsmoduls

Der Einbau der S7-200 ist einfach! Gehen Sie wie folgt vor.

Montage in einer Schalttafel

1. Bringen Sie die Bohrungen (M4) mit den in Tabelle 3-1 gezeigten Abmessungen an.
2. Schrauben Sie das Modul bzw. die Module mit den entsprechenden Schrauben in der Schalttafel fest.
3. Wenn Sie ein Erweiterungsmodul einsetzen, schließen Sie das Flachkabel des Erweiterungsmoduls an den Erweiterungsanschluss unter der Fronttür an.

Montage auf einer Standard-Hutschiene

1. Verschrauben Sie die Hutschiene in Abständen von jeweils 75 mm mit der Schalttafel.
2. Öffnen Sie den DIN-Rasthaken (an der Unterseite des Moduls) und hängen Sie das Modul mit der Rückseite auf der DIN-Hutschiene ein.
3. Wenn Sie ein Erweiterungsmodul einsetzen, schließen Sie das Flachkabel des Erweiterungsmoduls an den Erweiterungsanschluss unter der Fronttür an.
4. Drehen Sie das Modul nach unten zur Standard-Hutschiene und schließen Sie den Rasthaken. Achten Sie darauf, dass der Haken richtig einrastet und das Gerät sicher auf der Schiene befestigt ist. Um das Modul nicht zu beschädigen, drücken Sie auf die Bohrung und nicht direkt auf die Vorderseite des Moduls.

Tipp

In Umgebungen, in denen starke Schwingungen auftreten, oder bei vertikalem Einbau der S7-200 kann es nützlich sein, die S7-200 mit Erdungsklemmen auf der Hutschiene zu sichern.

In Umgebungen, in denen starke Schwingungen auftreten, bietet die Schalttafelmontage der S7-200 einen besseren Schutz vor Schwingungen.

Ausbau einer CPU oder eines Erweiterungsmoduls

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

1. Schalten Sie die Spannungsversorgung der S7-200 aus.
2. Lösen Sie alle Kabel und Leitungen vom Gerät. Die meisten S7-200 CPUs und Erweiterungsmodule verfügen über steckbare Klemmenblöcke, die dies erleichtern.
3. Wenn Sie an das Gerät, das Sie ausbauen möchten, Erweiterungsmodule angeschlossen haben, öffnen Sie die Fronttür und ziehen das Flachkabel des benachbarten Erweiterungsmoduls heraus.
4. Lösen Sie die Schrauben oder öffnen Sie den DIN-Rasthaken.
5. Nehmen Sie das Modul von der Schiene.

Ausbau und Wiedereinbau des Klemmenblocks

Die meisten S7-200 Module verfügen über Steckverbindungen, die den Einbau und den Austausch des Moduls erleichtern. Im Anhang A erfahren Sie, ob Ihr S7-200 Modul über einen steckbaren Klemmenblock verfügt. Für Module, die nicht über Steckverbindungen verfügen, können Sie steckbare Klemmenblöcke bestellen. Im Anhang E finden Sie die Bestellnummern.

Ausbau des Klemmenblocks

1. Öffnen Sie die Fronttür, damit Sie Zugriff auf den Klemmenblock haben.
2. Stecken Sie einen kleinen Schraubendreher in die Kerbe in der Mitte des Klemmenblocks.
3. Lösen Sie den Klemmenblock, indem Sie den Schraubendreher vom Gehäuse der S7-200 wegdrücken (siehe Bild 3-2).

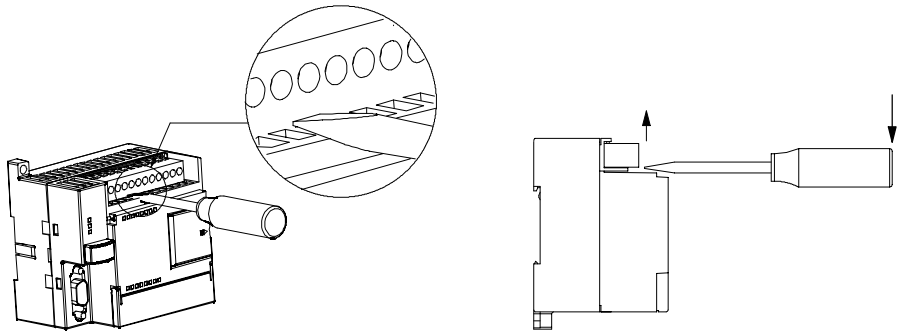


Bild 3-2 Ausbau des Klemmenblocks

Wiedereinbau des Klemmenblocks

1. Öffnen Sie die Fronttür.
2. Richten Sie den Klemmenblock mit den Pins des Geräts und auch die Verdrahtung des Klemmenblocks richtig aus.
3. Drücken Sie den Klemmenblock mit einer Drehbewegung fest nach unten, bis er einrastet. Prüfen Sie, dass der Klemmenblock richtig ausgerichtet ist und fest sitzt.

Richtlinien für Erdung und Verdrahtung

Ordnungsgemäße Erdung und Verdrahtung aller elektrischen Geräte ist wichtig für den optimalen Betrieb Ihres Systems und für zusätzliche Störfestigkeit für Ihre Anwendung und die S7-200.

Voraussetzungen

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

Stellen Sie sicher, dass Sie bei der Verdrahtung der S7-200 und aller angeschlossenen Geräte alle geltenden und verbindlichen Normen befolgen. Beachten Sie beim Einbau 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.

Warnung

Wenn Sie versuchen, die S7-200 oder daran angeschlossene Geräte in eingeschaltetem Zustand einzubauen oder zu verdrahten, kann es passieren, dass Sie einen elektrischen Schlag bekommen oder die Geräte fehlerhaft arbeiten. Ist die Spannungsversorgung der S7-200 und aller daran angeschlossenen Geräte während des Einbaus 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 Einbau bzw. Ausbau eines Geräts die Spannungsversorgung der S7-200 abgeschaltet ist.

Denken Sie beim Planen von Erdung und Verdrahtung Ihres S7-200 Systems immer an die Sicherheit. Elektronische Steuerungsgeräte wie die S7-200 können ausfallen und dadurch unerwarteten Betrieb der gesteuerten oder beobachteten Geräte hervorrufen. Deshalb sollten Sie Sicherheitseinrichtungen implementieren, die von der S7-200 unabhängig sind und vor möglichen Personen- und/oder Sachschäden schützen.

Warnung

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

Sorgen Sie daher für eine NOT-AUS-Funktion, elektromechanische oder andere redundante Sicherheitseinrichtungen, die von Ihrer S7-200 unabhängig sind.

Richtlinien für Potentialtrennung

Die Grenzwerte der AC-Spannungsversorgung und die E/A-Grenzen zu AC-Stromkreisen haben bei der S7-200 eine Nennspannung von 1500 V AC. Diese Potentialtrennungsgrenzen wurden getestet und bieten eine sichere elektrische Trennung zwischen AC-Leitungen und Niederspannungskreisen.

Alle an die S7-200 angeschlossenen Niederspannungskreise, z.B. eine 24-V-Versorgung, müssen aus einer zugelassenen Quelle gespeist werden, die sichere Potentialtrennung von der AC-Leitung und anderen Hochspannungen bietet. Solche Quellen bieten doppelte Isolierung gemäß internationalen elektrischen Sicherheitsstandards und haben Ausgaben ja nach Norm mit der Kennzeichnung SELV, PELV, Klasse 2 oder Begrenzte Spannung.

Warnung

Der Einsatz nicht potentialgetrennter oder einfach isolierter Versorgungen für die Niederspannungskreise aus AC-Leitungen kann zu gefährlichen Spannungen in Kreisen führen, die als berührungssicher gelten, z.B. Kommunikationskreise oder Niederspannungsgeberverdrahtung.

Diese unerwarteten Hochspannungen können zu tödlichen oder schweren Verletzungen und/oder Sachschaden führen.

Verwenden Sie nur solche Hochspannungs-/Niederspannungswandler, die als Quellen berührungssicherer Kreise mit begrenzter Spannung zugelassen sind.

Richtlinien für die Erdung der S7-200

Am besten erden Sie Ihre Anwendung, indem Sie darauf achten, dass alle gemeinsamen Anschlüsse und alle Erdanschlüsse Ihrer S7-200 und aller angeschlossenen Geräte an einer einzigen Stelle geerdet werden. Diese Stelle muss direkt mit Systemerde verbunden werden.

Zur optimalen Störfestigkeit schließen Sie am besten alle gemeinsamen DC-Rückleitungen an der gleichen Stelle an Erde an. Verbinden Sie die 24-V-DC-Geberversorgung (M) mit Erde.

Alle Erdleitungen müssen so kurz wie möglich sein und und über einen großen Leitungsquerschnitt verfügen, z.B. 2 mm^2.

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

Richtlinien für die Verdrahtung der S7-200

Wenn Sie die Verdrahtung Ihrer S7-200 planen, richten Sie einen Einzeltrennschalter ein, der gleichzeitig die Spannung der Spannungsversorgung für die S7-200 CPU, die Spannung aller Eingangskreise und die Spannung aller Ausgangskreise trennt. Sorgen Sie für Überstromschutz, z.B. durch eine Sicherung oder einen Schutzschalter, um Fehlerstrom in der Versorgungsverdrahtung zu begrenzen. Sie können zusätzlichen Schutz durch Sicherungen oder andere Strombegrenzungen in den einzelnen Ausgangskreisen implementieren.

Versehen Sie blitzschlaggefährdete Leitungen mit einem geeigneten Überspannungsschutz.

Vermeiden Sie es, Niederspannungssignalleitungen und Kommunikationskabel in der gleichen Kabelbahn wie AC-Versorgungsleitungen und schnellschaltende Hochgeschwindigkeits-DC-Leitungen zu verlegen. Leitungen sollten Sie paarweise verlegen: den Neutral- oder Nullleiter zusammen mit dem Phasenleiter oder der Signalleitung.

Verlegen Sie Leitungen so kurz wie möglich und achten Sie darauf, dass der Leitungsquerschnitt dem benötigten Strom entspricht. Den Klemmenblock können Sie mit Leitungen mit einem Querschnitt zwischen 2 mm^2 und 0,3 mm^2 verdrahten. Verwenden Sie als optimalen Schutz vor Störfestigkeit geschirmte Leitungen. Typischerweise erreichen Sie durch Erdung des Schirms an der S7-200 die besten Ergebnisse.

Wenn Sie einen Eingangskreis verdrahten, der durch eine externe Spannungsversorgung versorgt wird, implementieren Sie ein Überstromschutzgerät in dem Eingangskreis. Bei Kreisen, die durch die 24-V-DC-Geberversorgung der S7-200 versorgt werden, ist kein externer Schutz erforderlich, weil die Geberversorgung bereits strombegrenzt ist.

Die meisten S7-200 Module verfügen über steckbare Klemmenblöcke für die Anwenderverdrahtung. (Im Anhang A erfahren Sie, ob Ihr Modul über einen steckbaren Klemmenblock verfügt.) Achten Sie darauf, dass der Klemmenblock fest sitzt und die Leitungen fest im Klemmenblock verdrahtet sind, damit es nicht zu losen Anschlüssen kommt. Ziehen Sie die Schrauben jedoch nicht zu fest an, damit Sie den Klemmenblock nicht beschädigen. Das maximale Drehmoment für die Schrauben des Klemmenblocks beträgt 0,56 Nm.

Die S7-200 arbeitet mit Potentialtrennungsgrenzen, wodurch das Entstehen unerwünschter Ströme in Ihrer Anlage verhindert wird. Wenn Sie die Verdrahtung für Ihr System planen, berücksichtigen Sie diese Potentialtrennungsgrenzen. Im Anhang A finden Sie die Werte für die zur Verfügung gestellte Potentialtrennung und die Anordnung der Potentialtrennungsgrenzen. Potentialtrennungsgrenzen mit einer Nennspannung unter 1500 V AC dürfen nicht als einzige Sicherheitsgrenzen verwendet werden.

Tipp

In einem Kommunikationsnetz beträgt die maximale Länge des Kommunikationskabels 50 m ohne Busverstärker. Die Kommunikationsschnittstelle an der S7-200 ist nicht potentialgetrennt. Ausführliche Informationen finden Sie in Kapitel 7.

Richtlinien für Schutzbeschaltungen

Versehen Sie induktive Lasten mit Schutzbeschaltungen, die den Spannungsanstieg beim Ausschalten des Steuerungsausgangs begrenzen. Schutzbeschaltungen schützen Ihre Ausgänge vor frühzeitigem Ausfall aufgrund hoher induktiver Schaltströme. Außerdem begrenzen Schutzbeschaltungen die elektrischen Störungen, die beim Schalten induktiver Lasten entstehen.

Tipp

Die Effektivität einer Schutzbeschaltung hängt von der jeweiligen Anwendung ab und muss immer für den Einzelfall geprüft werden. Es müssen immer alle Komponenten einer Schutzbeschaltung für den Einsatz in der Anwendung bemessen sein.

DC-Ausgänge und Relais, die DC-Lasten steuern

Die DC-Ausgänge verfügen über internen Schutz, der für die meisten Anwendungen angemessen ist. Da die Relais für eine DC-Last oder eine AC-Last verwendet werden können, ist kein interner Schutz vorhanden.

Bild 3-3 zeigt ein Beispiel für eine Schutzbeschaltung für eine DC-Last. In den meisten Anwendungen ist der Einsatz einer Diode (A) parallel zur induktiven Last geeignet, doch wenn Ihre Anwendung schnellere Ausschaltzeiten benötigt, ist der Einsatz einer Zener-Diode (B) empfehlenswert. Bemessen Sie die Zener-Diode gemäß dem Strom im Ausgangskreis.

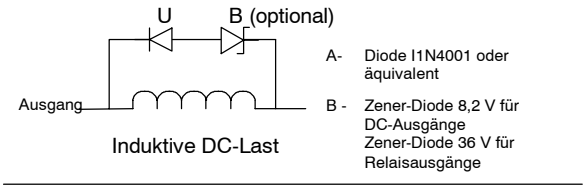


Bild 3-3 Schutzbeschaltung für eine DC-Last

AC-Ausgänge und Relais, die AC-Lasten steuern

Die AC-Ausgänge verfügen über internen Schutz, der für die meisten Anwendungen angemessen ist. Da die Relais für eine DC-Last oder eine AC-Last verwendet werden können, ist kein interner Schutz vorhanden.

Bild 3-4 zeigt ein Beispiel für eine Schutzbeschaltung für eine AC-Last. Wenn Sie mit einem Relais oder einem AC-Ausgang Lasten von 115 V/230 V AC schalten, ordnen Sie die Widerstands-/Kondensatornetzwerke parallel zu den AC-Lasten an (siehe Bild). Sie können die Spitzenspannung auch mit einem Metalloxid-Varistor (MOV) begrenzen. Achten Sie darauf, dass die Arbeitsspannung des Varistors mindstens 20 % höher ist als die Nennspannung.

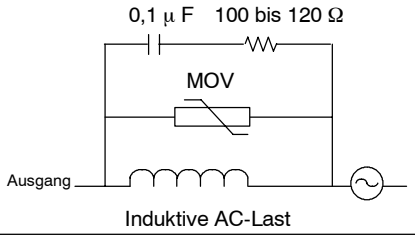


Bild 3-4 Schutzbeschaltung für eine AC-Last

Hinweis

Wenn Sie mit Relaiserweiterungsmodulen induktive Lasten von 230 V AC schalten, muss die externe Widerstands-/Kondensatorschutzbeschaltung parallel zur AC-Last angeordnet werden (siehe Bild 3-4).

SPS-Grundlagen

4

Die wesentliche Funktion der S7-200 ist es, Feldeingänge zu beobachten und die Ausgangsgeräte im Feld entsprechend der Steuerungslogik ein- und auszuschalten. In diesem Kapitel werden die Grundlagen für die Ausführung des Programms, die verschiedenen Arten von Speicher und die Art und Weise der Speicherung erläutert.

In diesem Kapitel

Ausführung der Steuerungslogik durch die S7-200 22
Zugreifen auf Daten der S7-200 24
Speichern und Zurückholen von Daten durch die S7-200 34
Speichern Ihres Programms auf einem Speichermodul 36
Einstellen des Betriebszustands der S7-200 CPU 37
Speichern von Variablenspeicher im EEPROM über das Programm 38
Funktionen der S7-200 39

Ausführung der Steuerungslogik durch die S7-200

Die S7-200 bearbeitet die Steuerungslogik in Ihrem Programm zyklisch, sie liest und schreibt Daten.

Die S7-200 setzt Ihr Programm zu den physikalischen Ein- und Ausgängen in Beziehung

Die grundlegende Funktionsweise der S7-200 ist sehr einfach:

- ❏ Die S7-200 liest den Signalzustand der Eingänge.
- ❏ Das Programm, das in der S7-200 gespeichert ist, wertet mit Hilfe der Eingänge die Logik aus. Während der Bearbeitung des Programms aktualisiert die S7-200 die Daten.
- ❏ Die S7-200 schreibt die Daten in die Ausgänge.

Bild 4-1 zeigt die Verbindung zwischen einem einfachen Schaltplan und der S7-200. In diesem Beispiel wird der Zustand des Schalters zum Starten des Motors mit dem Zustand anderer Eingänge verknüpft. Die Berechnungen dieser Zustände bestimmen dann den Signalzustand des Ausgangs für den Aktor, der den Motor startet.

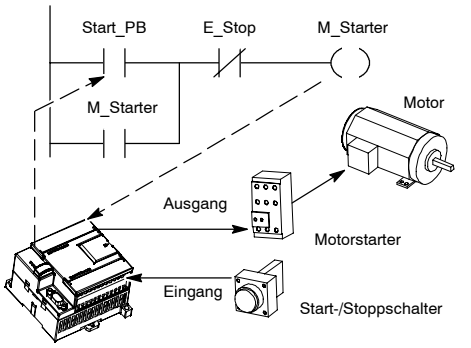


Bild 4-1 Steuern von Eingängen und Ausgängen

Die S7-200 führt alle Aufgaben in einem Zyklus aus

Die S7-200 führt eine Reihe von Aufgaben wiederholt aus. Diese regelmäßige Bearbeitung der Aufgaben wird Zyklus genannt. Die S7-200 führt die meisten bzw. alle der folgenden Aufgaben im Zyklus aus (siehe Bild 4-2):

- ❏ Lesen der Eingänge: Die S7-200 kopiert den Zustand der physikalischen Eingänge in das Prozessabbild der Eingänge.
- ❏ Ausführen der Steuerungslogik im Programm: Die S7-200 führt die Operationen des Programms aus und speichert die Werte in verschiedenen Speicherbereichen.
- ❏ Bearbeiten von Kommunikationsanforderungen: Die S7-200 führt alle für die Kommunikation erforderlichen Aufgaben aus.
- ❏ Selbstdiagnose der CPU: Die S7-200 prüft, dass die Firmware, der Programmspeicher und alle Erweiterungsmodule einwandfrei funktionieren.
- ❏ Schreiben in die Ausgänge: Die im Prozessabbild der Ausgänge gespeicherten Werte werden in die physikalischen Ausgänge geschrieben.

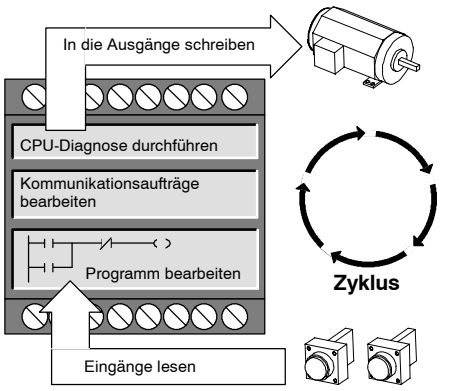


Bild 4-2 Zyklus der S7-200

Die Ausführung des Zyklus richtet sich danach, ob sich die S7-200 im Betriebszustand STOP oder im Betriebszustand RUN befindet. Im Betriebszustand RUN wird Ihr Programm ausgeführt. Im Betriebszustand STOP wird Ihr Programm nicht ausgeführt.

Lesen der Eingänge

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

Analogeingänge: Die S7-200 aktualisiert die Analogeingänge nicht automatisch als Teil des Zyklus, es sei denn, Sie haben das Filtern der Analogeingänge aktiviert. Ein Analogfilter sorgt für stabilere Signale. Sie können für jeden Eingang einen Analogfilter aktivieren.

Ist der Analogeingabefilter für einen Analogeingang aktiviert, aktualisiert die S7-200 den Analogeingang einmal pro Zyklus, führt die Filterfunktion aus und speichert den gefilterten Wert intern. Der gefilterte Wert wird dann jederzeit zur Verfügung gestellt, wenn Ihr Programm auf den Analogeingang zugreift.

Ist der Analogeingabefilter ausgeschaltet, liest die S7-200 den Wert des Analogeingangs jedesmal, wenn Ihr Programm auf einen Analogeingang zugreift, aus dem physikalischen Modul.

Tipp

Der Analogeingabefilter sorgt für stabilere Analogwerte. Der Analogeingabefilter sollte in Anwendungen aktiviert werden, in denen sich das Eingangssignal mit der Zeit langsam verändert. Handelt es sich bei dem Signal um ein schnelles Signal, sollten Sie den Analogfilter nicht aktivieren.

Verwenden Sie den Analogfilter nicht bei Modulen, die digitale Daten oder Alarmanzeigen in Analogwörtern übergeben. Schalten Sie den Analogfilter immer für RTD-, Thermoelement- und AS-Interface-Master-Module aus.

Bearbeiten des Programms

Während dieses Abschnitts im Zyklus bearbeitet die S7-200 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. Die Interruptprogramme werden jedoch nicht als Bestandteil des normalen Zyklus bearbeitet, sondern nur dann, wenn ein Interruptereignis auftritt (dies ist an jeder Stelle im Zyklus möglich).

Bearbeiten von Kommunikationsanforderungen

Während des Abschnitts für die Kommunikationsbearbeitung im Zyklus verarbeitet die S7-200 alle Meldungen, die von der Kommunikationsschnittstelle oder von den intelligenten E/A-Modulen empfangen werden.

Selbstdiagnose der CPU

Während dieses Abschnitts des Zyklus prüft die S7-200 den einwandfreien Betrieb der CPU, die Speicherbereiche und den Zustand der Erweiterungsmodule.

Schreiben in die Digitalausgänge

Am Ende des Zyklus schreibt die S7-200 die Werte aus dem Prozessabbild der Ausgänge in die Digitalausgänge. (Analogausgänge werden sofort aktualisiert, unabhängig vom Zyklus.)

Zugreifen auf Daten der S7-200

Die S7-200 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. Tabelle 4-1 zeigt den Bereich der ganzzahligen Werte, die durch die unterschiedlichen Datengrößen dargestellt werden können.

Tabelle 4-1 Dezimal- und Hexadezimalbereiche für die verschiedenen Datengrößen

Darstellung	Byte (B)	Wort (W)	Doppelwort (D)
Ganze Zahl (16 Bit) ohne Vorzeichen	0 bis 255 0 bis FF	0 bis 65.535 0 bis FFFF	0 bis 4.294.967.295 0 bis FFFF FFFF
Ganze Zahl (16 Bit) mit Vorzeichen	-128 bis +127 80 bis 7F	-32.768 bis +32.767 8000 bis 7FFF	-2.147.483.648 bis +2.147.483.647 8000 0000 bis 7FFF FFFF
IEEE 32-Bit-Gleitpunkt	-/-	-/-	+1.175495E-38 bis +3.402823E+38 (positiv) -1.175495E-38 bis -3.402823E+38 (negativ)

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 4-3 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.

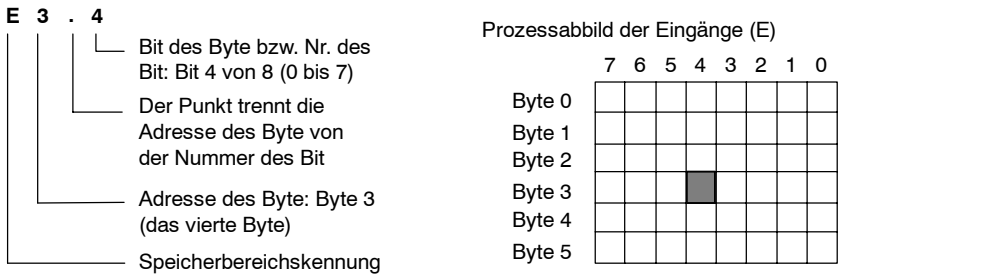


Bild 4-3 Adressierung im Format Byte.Bit

Wenn Sie das Byte.Bit-Format für die Adressierung verwenden, können Sie auf Daten in den meisten Speicherbereichen (V, E, A, M, S, L und SM) als Bytes, Wörter oder Doppelwörter zugreifen. Wenn Sie auf ein Byte, Wort oder Doppelwort an Daten im Speicher 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 4-4).

Auf Daten in anderen Speicherbereichen (z.B. T, Z, HC und Akkumulatoren) greifen Sie zu, indem Sie für die Adresse eine Bereichskennung und die Nummer des Elements angeben.

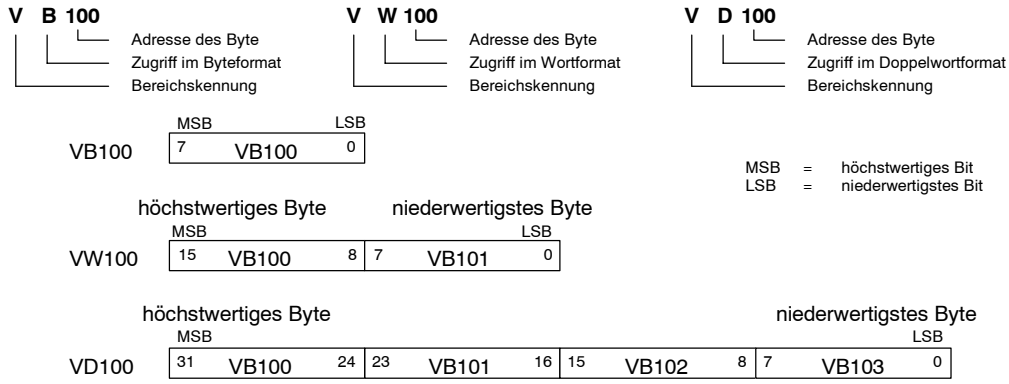


Bild 4-4 Zugriff auf dieselbe Adresse im Byte-, Wort- und Doppelwortformat

Zugreifen auf Daten in den Speicherbereichen

Prozessabbild der Eingänge: E

Die S7-200 fragt 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.

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

Prozessabbild der Ausgänge: A

Am Ende des Zyklus kopiert die S7-200 die Werte des Prozessabbilds der Ausgänge in die physikalischen Ausgänge. Auf das Prozessabbild können Sie im Bit-, Byte-, Wort- und Doppelwortformat zugreifen.

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

Variablenspeicher: 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.

Bit:	V*[Byteadresse].[Bitadresse]*	V10.2
Byte, Wort oder Doppelwort:	V*[Größe][Anfangsadresse des Byte]*	VW100

Speicherbereich der Merker: M

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

Bit:	M*[Byteadresse].[Bitadresse]*	M26.7
Byte, Wort oder Doppelwort:	M*[Größe][Anfangsadresse des Byte]*	MD20

Zeiten: T

Die S7-200 verfügt über Zeiten, die Zeitinkremente in Auflösungen (Inkrementen der Zeitbasis) von 1 ms, 10 ms oder 100 ms zählen. 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 gesetzt bzw. zurückgesetzt, wenn der aktuelle Wert mit dem voreingestellten Wert verglichen wird. 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 4-5 sehen Sie, dass die Operation Schließerkontakt auf das Zeitbit zugreift, während die Operation Wort übertragen auf den aktuellen Wert der Zeit zugreift.

Format: *T[Nummer der Zeit]* T24

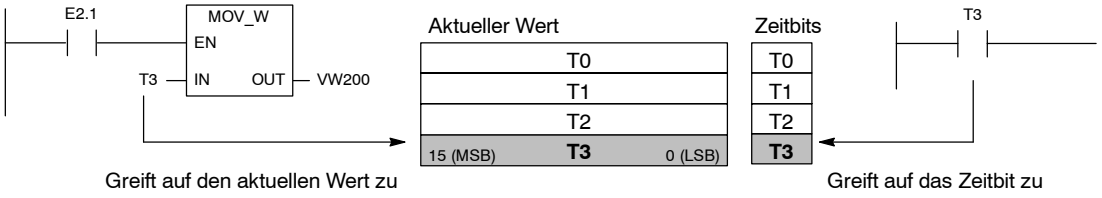


Bild 4-5 Zugreifen auf das Zeitbit und den aktuellen Wert einer Zeit

Zähler: Z

Die S7-200 verfügt über drei Arten von Zählern, die an den Zähleingängen die steigenden Flanken zählen. ein Zähler zählt vorwärts, ein anderer zählt rückwärts und der dritte Zähler zählt sowohl vorwärts als auch rückwärts. 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 gesetzt bzw. zurückgesetzt, wenn der aktuelle Wert mit dem voreingestellten Wert verglichen wird. Der voreingestellte Wert wird zusammen mit der Operation eingegeben.

Sie greifen auf die beiden Variablen über die Adresse des Zählers (Z + Nummer des Zählers) 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 4-6 sehen Sie, dass die Operation Schließerkontakt auf das Zählerbit zugreift, während die Operation Wort übertragen auf den aktuellen Wert des Zählers zugreift.

Format: *Z[Nummer des Zählers]* Z24

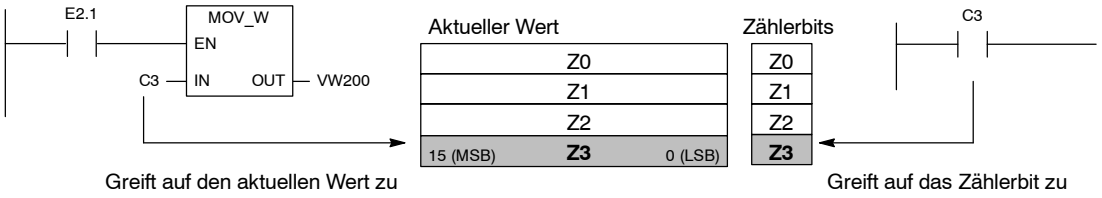


Bild 4-6 Zugreifen auf das Zählerbit und den aktuellen Wert eines Zählers

Schnelle Zähler: HC

Die schnellen Zähler zählen schnelle Ereignisse unabhängig vom Zyklus der CPU. 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.

Format: HC*[Nummer des schnellen Zählers]* HC1

Akkumulatoren: AC

Die 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 S7-200 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.

Die Größe der Daten, auf die zugegriffen wird, richtet sich nach der Operation, mit der Sie auf den Akkumulator zugreifen. Wie Bild 4-7 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.

Informationen zur Verwendung von Akkumulatoren in Interruptprogrammen finden Sie unter den Interruptoperationen im Kapitel 6.

Format: AC*[Nummer des Akkumulators]* AC0

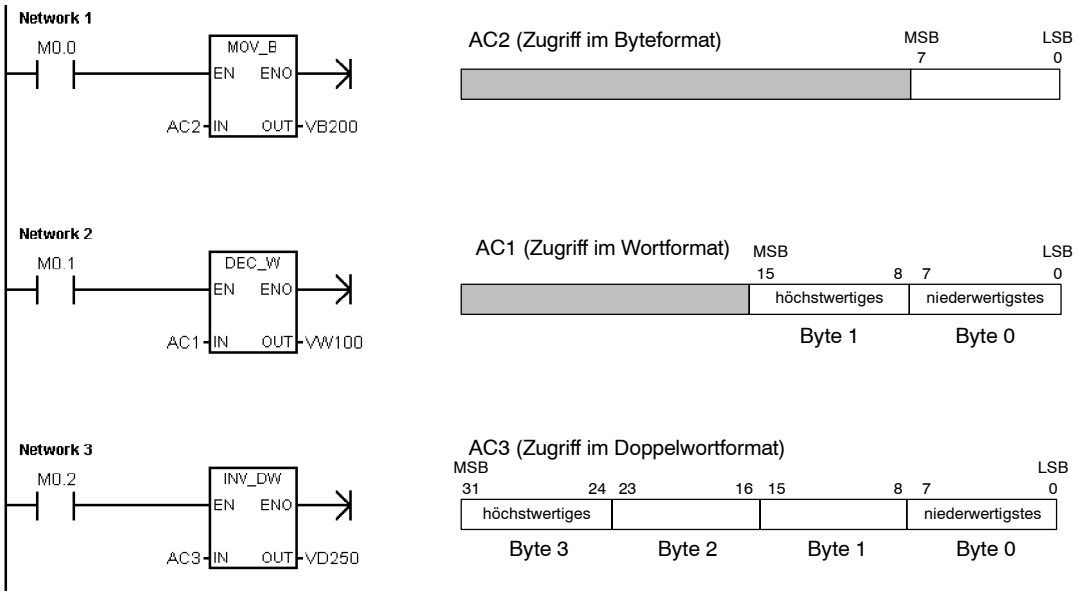


Bild 4-7 Zugreifen auf Akkumulatoren

Sondermerker: 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, ein Bit, das in einem bestimmten Takt ein- und ausgeschaltet wird, oder ein Bit, das den Zustand einer arithmetischen Operation oder einer anderen Operation anzeigt. (Ausführliche Informationen zu Sondermerkern finden Sie in Anhang D.) Auf SM-Bits können Sie im Bit-, Byte-, Wort- und Doppelwortformat zugreifen.

Bit:	SM*[Byteadresse].[Bitadresse]*	SM0.1
Byte, Wort oder Doppelwort:	SM*[Größe][Anfangsadresse des Byte]*	SMB86

Lokaldatenspeicher: L

Die S7-200 verfügt über einen Lokaldatenspeicher von 64 Bytes, von denen 60 Bytes als Zwischenspeicher oder zur Übergabe von Formalparametern an Unterprogramme genutzt werden können.

Tipp

Wenn Sie in KOP oder FUP programmieren, sind die letzten vier Bytes des Lokaldatenbereichs für STEP 7-Micro/WIN reserviert. Wenn Sie in AWL programmieren, stehen alle 64 Bytes des Lokaldatenbereichs zur Verfügung, doch wir empfehlen Ihnen, auch hier die letzten vier Bytes des Lokaldatenbereichs nicht zu verwenden.

Der Lokaldatenspeicher ist dem Variablenspeicher ähnlich. Es gibt jedoch einen wichtigen Unterschied. Der Variablenspeicher hat globalen Geltungsbereich, während der Lokaldatenbereich lokalen Geltungsbereich hat. Der Ausdruck ”globaler Geltungsbereich” bedeutet, dass auf eine Adresse im Speicher von jeder beliebigen Programmorganisationseinheit zugegriffen werden kann (Hauptprogramm, Unterprogramme und Interruptprogramme). Der Ausdruck ”lokaler Geltungsbereich” bedeutet, dass die Speicherzuordnung für eine bestimmte Programmorganisationseinheit gilt. Die S7-200 verfügt über 64 Bytes Lokaldatenspeicher für das Hauptprogramm, 64 Bytes für jede Schachtelungsebene der Unterprogramme und 64 Bytes für Interruptprogramme.

Auf den Lokaldatenbereich, der dem Hauptprogramm zugewiesen ist, haben Unterprogramme und Interruptprogramme keinen Zugriff. Ein Unterprogramm kann nicht auf den Lokaldatenbereich des Hauptprogramms, des Interruptprogramms oder eines anderen Unterprogramms zugreifen. Auch kann ein Interruptprogramm nicht auf den Lokaldatenbereich des Hauptprogramms oder eines Unterprogramms zugreifen.

Die Zuordnung des Lokaldatenbereichs wird von der S7-200 nach Anforderung vorgenommen. D.h. die Zuordnungen des Lokaldatenbereichs für Unter- und Interruptprogramme sind während der Ausführung des Hauptprogramms nicht vorhanden. Tritt ein Interrupt auf oder wird ein Unterprogramm aufgerufen, wird der Lokaldatenbereich nach Bedarf zugeordnet. Diese neue Zuordnung des Lokaldatenbereichs kann die gleichen Adressen wiederverwenden, die zuvor von einem anderen Unterprogramm oder von einem Interruptprogramm verwendet wurden.

Der Lokaldatenbereich wird von der S7-200 nicht zur Zeit der Zuordnung initialisiert und kann beliebige Werte enthalten. Werden beim Aufruf eines Unterprogramms Formalparameter übergeben, werden die Werte der übergebenen Parameter von der S7-200 in den entsprechenden Adressen in dem Lokaldatenbereich, der diesem Unterprogramm zugeordnet ist, abgelegt. Die Adressen im Lokaldatenbereich, die bei der Übergabe der Formalparameter keine Werte aufnehmen, werden nicht initialisiert und können zur Zeit der Zuordnung beliebige Werte enthalten.

Bit:	L*[Byteadresse].[Bitadresse]*	L0.0
Byte, Wort oder Doppelwort:	L*[Größe] [Anfangsadresse des Byte]*	LB33

Analogeingänge: AE

Die S7-200 wandelt 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). Analogeingänge können nur gelesen werden.

Format:	AEW*[Anfangsadresse des Byte]*	AEW4

Analogausgänge: 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.), schreiben Sie die Werte mit den Adressen gerader Bytes (z.B. AAW0, AAW2, AAW4). Analogausgänge können nur geschrieben werden.

Format:	AAW*[Anfangsadresse des Byte]*	AAW4

Ablaufsteuerungsrelais (SCR): S

SCRs bzw. S-Bits gliedern die Funktionsweise einer Anlage in einzelne Schritte bzw. in äquivalente Programmteile. Durch Ablaufsteuerungsrelais wird das Steuerprogramm in logische Segmenten strukturiert. Auf S-Bits können Sie im Bit-, Byte-, Wort- und Doppelwortformat zugreifen.

Bit:	S*[Byteadresse].[Bitadresse]*	S3.1
Byte, Wort oder Doppelwort:	S*[Größe][Anfangsadresse des Byte]*	SB4

Format von Realzahlen

Realzahlen (bzw. Gleitpunktzahlen) werden als einfachgenaue 32-Bit-Zahlen dargestellt, deren Format in der Richtlinie ANSI/IEEE 754-1985 beschrieben ist (siehe Bild 4-8). Auf Realzahlen wird im Doppelwortformat zugegriffen.

Bei der S7-200 sind Gleitpunktzahlen bis zu 6 Dezimalstellen genau. Sie können also bis zu 6 Dezimalstellen angeben, wenn Sie eine Gleitpunktkonstante eingeben.

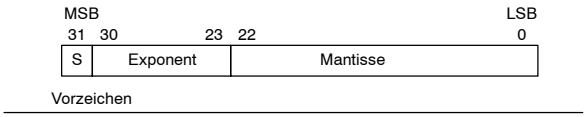


Bild 4-8 Format einer Realzahl

Genauigkeit beim Berechnen von Realzahlen

Bei Berechnungen, die eine lange Reihe von Werten einschließlich sehr großen und sehr kleinen Zahlen benötigten, kann es zu ungenauen Ergebnissen kommen. Dies kann auftreten, wenn sich die Zahlen um 10 hoch *x* unterscheiden, wobei $x > 6$.

Beispiel: 100 000 000 + 1 = 100 000 000

Format von Zeichenketten

Eine Zeichenkette ist eine Folge aus Zeichen, wobei jedes Zeichen als Byte gespeichert wird. Das erste Byte einer Zeichenkette definiert die Länge der Zeichenkette angibt, wobei es sich um die Anzahl der Zeichen handelt. Bild 4-9 zeigt das Format einer Zeichenkette. Eine Zeichenkette kann aus 0 bis 254 Zeichen bestehen, plus Längenbyte, so dass die maximale LÄnge einer Zeichenkette 255 Bytes beträgt.

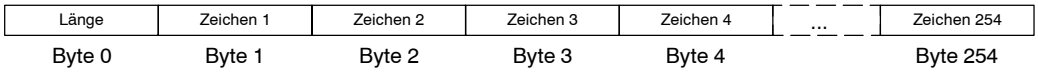


Bild 4-9 Format von Zeichenketten

Angeben eines konstanten Werts für S7-200 Operationen

In vielen Operationen für die S7-200 können Sie Konstanten verwenden. Konstanten können Bytes, Wörter und Doppelwörter sein. Die S7-200 speichert alle Konstanten als Binärwerte, die im Dezimal-, Hexadezimal-, ASCII- und Realzahlen-/Gleitpunktformat dargestellt werden können (siehe Tabelle 4-2).

Tabelle 4-2 Darstellung von konstanten Werten

Darstellung	Format	Beispiel
Dezimal	[Dezimalwert]	20047
Hexadezimal	16#[Hexadezimalwert]	16#4E4F
Binär	2#[Binärzahl]	2#1010_0101_1010_0101
ASCII	'[ASCII-Text]'	'Text in Hochkommata'
Realzahl	ANSI/IEEE 754-1985	+1.175495E-38 (positiv) -1.175495E-38 (negativ)

Tipp

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.

Adressierung der integrierten E/A und der Erweiterungs-E/A

Die integrierten Ein- und Ausgänge auf der Zentraleinheit (CPU) verfügen über feste Adressen. Sie können Ihre S7-200 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.

Tipp

Für digitale Erweiterungsmodule 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. Bei Eingabemodulen werden die freien Bits in jedem Aktualisierungszyklus der Eingänge auf Null gesetzt.

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 zugeordnet werden.

Bild 4-10 zeigt ein Beispiel für die E/A-Nummerierung für eine bestimmte Hardware-Konfiguration. Die Lücken in der Adressierung (grau und kursiv dargestellt) können von Ihrem Programm nicht verwendet werden.

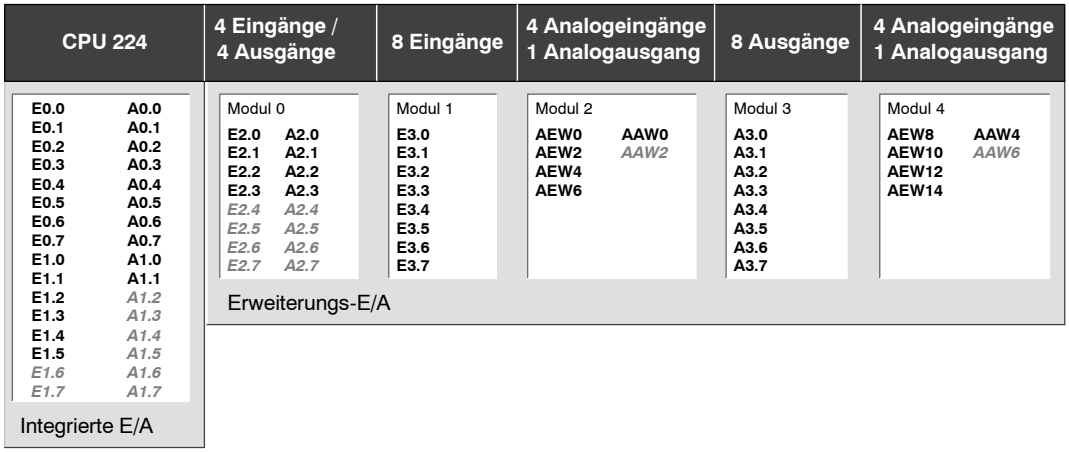


Bild 4-10 Beispiel für die E/A-Adressen von integrierten E/A und Erweiterungs-E/A (CPU 224)

Indirekte Adressierung der S7-200 Speicherbereiche durch Pointer

Die indirekte Adressierung verwendet Pointer, um auf Daten im Speicher zuzugreifen. Pointer sind Doppelwortwerte, die eine Adresse im Speicher enthalten. Als Pointer können Sie nur Adressen des Variablen- und Lokaldatenspeichers bzw. einen der Akkumulatoren (AC1, AC2 oder AC3) angeben. Zum Erstellen eines Pointers verwenden Sie die Operation Doppelwort übertragen. Diese Operation überträgt die indirekt adressierte Adresse an die Adresse des Pointers. Pointer können auch als Parameter an ein Unterprogramm übergeben werden.

Bei der S7-200 können Sie mit einem Pointer auf die folgenden Speicherbereiche zugreifen: E, A, V, M, S, T (nur auf den aktuellen Wert) und Z (nur auf den aktuellen Wert). Mit der indirekten Adressierung können Sie nicht auf einzelne Bits oder auf die Speicherbereiche AE, AA, HC, SM oder L zugreifen.

Wenn Sie indirekt auf Daten in einer Adresse im Speicher zugreifen möchten, können Sie einen Pointer auf die Adresse erstellen, indem Sie das Zeichen & und die entsprechende Adresse eingeben. 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.

Ein Stern (*) vor dem Operanden einer Operation zeigt an, dass es sich bei der Adresse um einen Pointer handelt. In Bild 4-11 gibt *AC1 an, dass AC1 ein Pointer ist, der die Adresse für den Wortwert enthält, der von der Operation Wort übertragen (MOVW) referenziert wird. In diesem Beispiel werden die Werte von VB200 und VB201 in den Akkumulator AC0 übertragen.

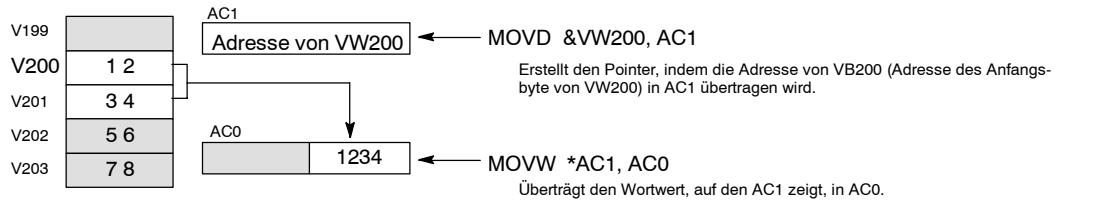


Bild 4-11 Erstellen und Verwenden eines Pointers

Bild 4-12 zeigt, dass Sie den Wert eines Pointers ändern können. 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.

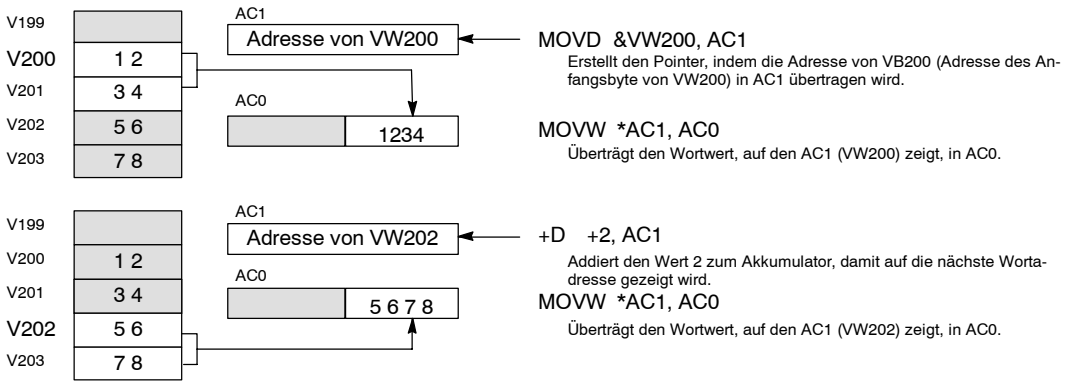


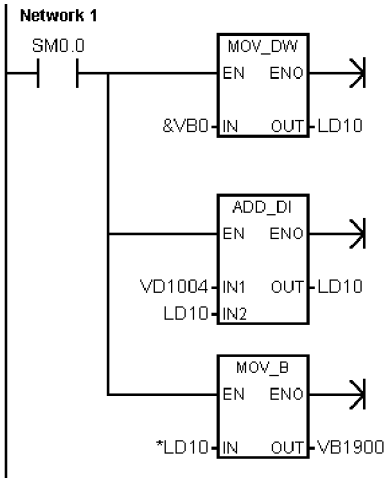
Bild 4-12 Ändern eines Pointers

Tipp

Achten Sie darauf, dass Sie die Größe der Daten angeben, auf die Sie zugreifen möchten: wenn Sie auf ein Byte zugreifen möchten, inkrementieren Sie den Wert des Pointers um 1. Wenn Sie auf ein Wort oder einen aktuellen Wert einer Zeit oder eines Zählers zugreifen möchten, inkrementieren Sie den Wert des Pointers um 2. Und wenn Sie auf ein Doppelwort zugreifen möchten, inkrementieren Sie den Wert des Pointers um 4.

Beispielprogramm für den Zugriff auf Daten im Variablenspeicher mittels Versatz

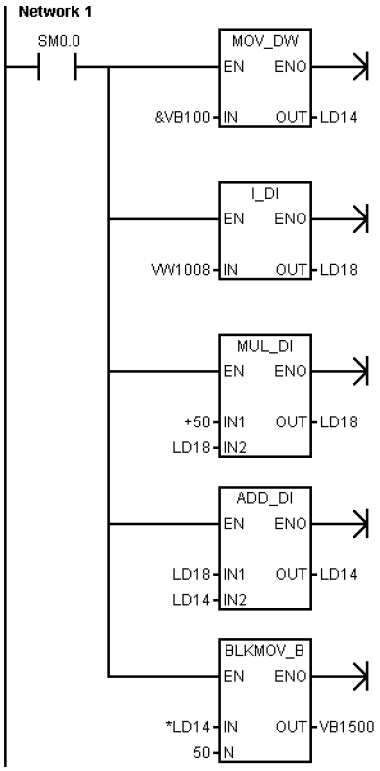
In diesem Beispiel wird LD10 als Pointer auf die Adresse VB0 verwendet. Sie inkrementieren den Pointer dann um den in VD1004 gespeicherten Versatz. LD10 zeigt dann auf eine andere Adresse im Variablenspeicher (VB0 + Versatz). Der in der Adresse im Variablenspeicher gespeicherte Wert, auf den LD10 zeigt, wird dann in VB1900 kopiert. Wenn Sie den Wert in VD1004 ändern, können Sie auf alle Adressen im Variablenspeicher zugreifen.




```
Netzwerk 1  //Lesen eines Werts einer beliebigen VB-Adresse mittels Versatz:
            //
            //1. Anfangsadresse des Variablenspeichers in einen Pointer laden.
            //2. Versatzwert zum Pointer addieren.
            //3. Wert aus der Adresse im Variablenspeicher (Versatz) in VB1900
            //    kopieren.
LD      SM0.0
MOVD    &VB0, LD10
+D      VD1004, LD10
MOVB    *LD10, VB1900
```

Beispielprogramm für den Zugriff auf Daten in einer Tabelle mittels Pointer

In diesem Beispiel wird LD14 als Pointer auf ein Rezept verwendet, das in einer Tabelle mit Rezepten gespeichert ist, die an VB100 beginnt. In diesem Beispiel speichert VW1008 den Index auf ein bestimmtes Rezept in der Tabelle. Wenn jedes Rezept in der Tabelle 50
Bytes lang ist, multiplizieren Sie den Index mit 50, um den Versatz auf die Anfangsadresse eines bestimmten Rezepts zu erhalten. Indem Sie den Versatz zum Pointer addieren, können Sie auf jedes einzelne Rezept in der Tabelle zugreifen. In diesem Beispiel wird das Rezept in die 50 Bytes kopiert, die an VB1500 beginnen.




```
Netzwerk 1  //Übertragen eines Rezepts aus einer Tabelle mit Rezepten:
            // -  Jedes Rezept ist 50 Bytes lang.
            // -  Der Index-Parameter (VW1008) identifiziert das Rezept,
            //     das geladen werden soll.
            //
            //1.  Pointer auf Anfangsadresse der Rezepttabelle erstellen.
            //2.  Index des Rezepts in einen Doppelwortwert umwandeln.
            //3.  Versatz multiplizieren, um die Größe des Rezepts unterzubrin
            //gen.
            //4.  Geänderten Versatz zum Pointer addieren.
            //5.  Gewähltes Rezept VB1500 bis VB1549 übertragen.
LD      SM0.0
MOVD    &VB100, LD14
ITD     VW1008, LD18
*D      +50, LD18
+D      LD18, LD14
BMB     *LD14, VB1500, 50
```

Speichern und Zurückholen von Daten durch die S7-200

Die S7-200 bietet Ihnen verschiedene Methoden, um sicherzustellen, dass Ihr Programm, die Programmdaten und die Konfigurationsdaten Ihrer S7-200 sicher abgelegt sind.

Die S7-200 verfügt über einen Hochleistungskondensator, der die Datensicherheit des RAM-Speichers auch nach dem Abschalten der Spannungsversorgung gewährleistet. Je nach Ausführung der S7-200 kann der Hochleistungskondensator den RAM mehrere Tage lang speichern.

Die S7-200 verfügt über einen EEPROM, in dem Sie Ihr gesamtes Programm, einige Datenbereiche und die Konfigurationsdaten nullspannungsfest ablegen können.

Die S7-200 unterstützt auch ein optionales Batteriemodul, mit dem Sie die Zeit verlängern können, die der RAM-Speicher nach Abschalten der Spannungsversorgung der S7-200 die Daten puffert. Das Batteriemodul übernimmt nach dem Entladen des Hochleistungskondensators das Puffern der Daten.

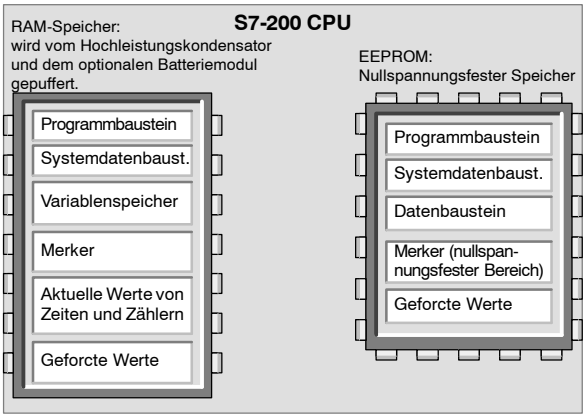


Bild 4-13 Speicherbereiche der S7-200 CPU

Laden von Elementen Ihres Projekts in die und aus der CPU

Ihr Projekt besteht aus drei Komponenten: dem Programmbaustein, dem Datenbaustein (optional) und dem Systemdatenbaustein (optional).

Bild 4-14 zeigt, wie ein Projekt in die S7-200 geladen wird.

Wenn Sie ein Projekt in die CPU laden, werden die Elemente des geladenen Projekts im RAM-Bereich gespeichert. Außerdem kopiert die S7-200 das Anwenderprogramm, den Datenbaustein und den Systemdatenbau-
stein automatisch in den EEPROM zur nullspannungsfesten Speicherung.

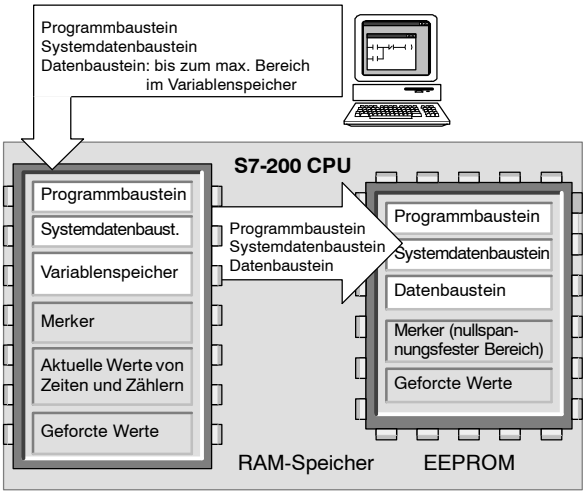


Bild 4-14 Laden eines Projekts in die S7-200

Bild 4-15 zeigt, wie ein Projekt aus der S7-200 geladen wird.

Wenn Sie ein Projekt aus der CPU in Ihren PC laden, lädt die S7-200 den Systemdatenbaustein aus dem RAM-Speicher und den Programmbaustein sowie den Datenbaustein aus dem EEPROM.

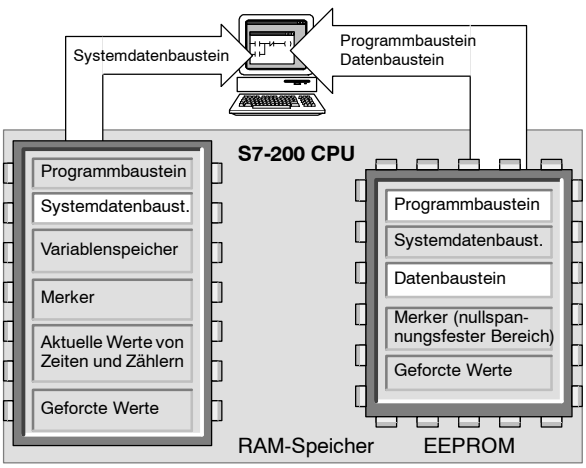


Bild 4-15 Laden eines Projekts aus der S7-200

Speichern des remanenten Speicherbereichs M bei Spannungsausfall

Die ersten 14 Bytes im Speicherbereich der Merker (MB0 bis MB13) werden bei Spannungsverlust der S7-200 nullspannungsfest im EEPROM gespeichert, sofern Sie sie zuvor als remanent definiert haben.

Die S7-200 überträgt die remanenten Bereiche der Merker in den EEPROM (siehe Bild 4-16).

In der Voreinstellung sind die ersten 14 Bytes der Merker nicht remanent. Durch die Voreinstellung ist der Speichervorgang, der üblicherweise beim Ausschalten der S7-200 durchgeführt wird, deaktiviert.

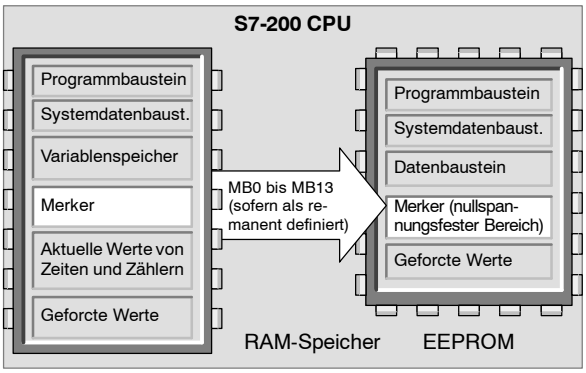
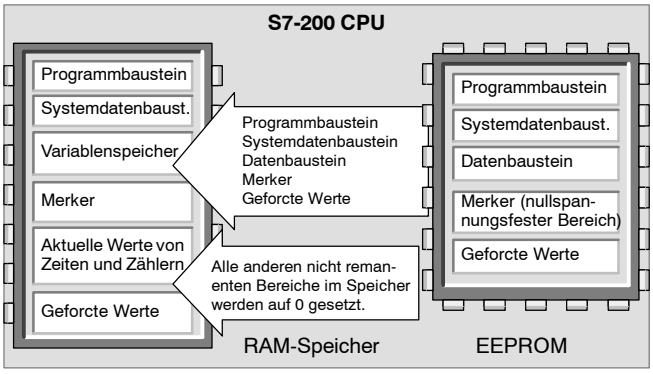


Bild 4-16 Speichern des Speicherbereichs M bei Spannungsausfall

Zurückholen von Daten nach dem Anlauf

Beim Anlauf stellt die S7-200 den Programmbaustein und den Systemdatenbaustein aus dem EEPROM wieder her (siehe Bild 4-17). Die S7-200 prüft beim Anlauf auch 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 remanenten und nicht remanenten Bereiche des Variablenspeichers werden aus dem entsprechenden Datenbaustein im EEPROM wiederhergestellt. Konnte der Inhalt des RAM-Speichers nicht gepuffert werden (z.B. durch extrem langen Spannungsverlust), setzt die S7-200 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). Dann werden die im EEPROM gespeicherten Daten in den RAM-Speicher kopiert.



Wurden die Programmdaten erfolgreich gepuffert, wird der Datenbaustein in die nicht remanenten Bereiche des Variablenspeichers im RAM kopiert.

Wurden die Programmdaten **NICHT** gepuffert, wird der Datenbaustein und der Speicherbereich der Merker (MB0 bis MB13), sofern diese als remanent definiert sind, kopiert.

Bild 4-17 Zurückholen von Daten nach dem Anlauf

Speichern Ihres Programms auf einem Speichermodul

Die S7-200 unterstützt ein optionales Speichermodul, das einen steckbaren EEPROM für Ihr Programm darstellt. Die S7-200 speichert folgende Komponenten im Speichermodul: den Programmbaustein, den Datenbaustein, den Systemdatenbaustein und die geforcten Werte.

Sie können Ihr Programm nur bei betriebsbereiter S7-200 mit gestecktem Speichermodul im Betriebszustand STOP aus dem RAM-Speicher in das Modul kopieren. Sie können das Speichermodul stecken oder ziehen, während die S7-200 eingeschaltet ist.

Vorsicht

Elektrostatische Entladungen können das Speichermodul oder den für das Modul vorgesehenen Schacht in der S7-200 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.

Zum Stecken des Speichermoduls nehmen Sie die Plastikabdeckung von der S7-200 CPU und stecken das Speichermodul in den Schacht. Das Speichermodul ist so geformt, dass es nur in eine Richtung in den Schacht gesteckt werden kann.

Kopieren Ihres Programms ins Speichermodul

Nachdem Sie das Speichermodul gesteckt haben, können Sie das Progamm folgendermaßen in das Modul kopieren:

1. Versetzen Sie die S7-200 CPU in den Betriebszustand STOP.
2. Sofern Sie das Programm noch nicht in die S7-200 geladen haben, laden Sie es jetzt.
3. Wählen Sie den Menübefehl **Zielsystem > Speichermodul**, um das Programm in das Speichermodul zu kopieren. In Bild 4-18 sehen Sie die Komponenten des CPU-Speichers, die im Speichermodul abgelegt werden.
4. Optional: Ziehen Sie das Speichermodul und ersetzen Sie die Abdeckung der S7-200.

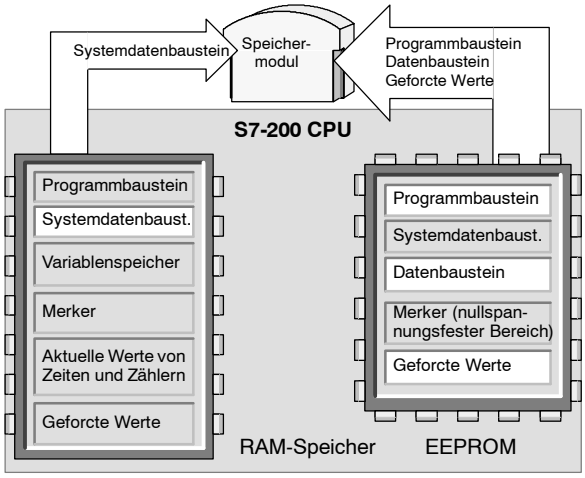


Bild 4-18 Kopieren in ein Speichermodul

Zurückholen des Programms vom Speichermodul

Wenn Sie ein Programm aus dem Speichermodul in die S7-200 laden möchten, müssen Sie die S7-200 ausschalten und mit gestecktem Speichermodul wieder einschalten.

Hinweis

Lassen Sie die S7-200 CPU mit leerem Speichermodul oder mit einem Speichermodul, das von einer anderen S7-200 CPU-Variante programmiert wurde, anlaufen, tritt ein Fehler auf. Speichermodule, die mit einer früheren CPU-Variante programmiert wurden, können von einer späteren CPU-Variante gelesen werden. Der umgekehrte Fall gilt jedoch nicht. Beispielsweise können Speichermodule, die von einer CPU 221 oder einer CPU 222 programmiert wurden, von einer CPU 224 gelesen werden. Doch Speichermodule, die von einer CPU 224 programmiert wurden, können von einer CPU 221 oder einer CPU 222 nicht gelesen werden.

Ziehen Sie das Speichermodul und schalten Sie die Spannungsversorgung der S7-200 ein. Nach dem Anlauf können Sie das Speichermodul bei Bedarf stecken und neu programmieren.

Die S7-200 führt nach dem Einschalten die folgenden Aufgaben aus, sofern ein Speichermodul gesteckt ist (siehe Bild 4-19):

1. Unterscheidet sich der Inhalt des Speichermoduls vom Inhalt des EEPROM, löscht die S7-200 den RAM-Speicher.
2. Die S7-200 kopiert den Inhalt des Speichermoduls in den RAM-Speicher.
3. Die S7-200 kopiert den Programmbaustein, den Systemdatenbaustein und den Datenbaustein in den EEPROM.

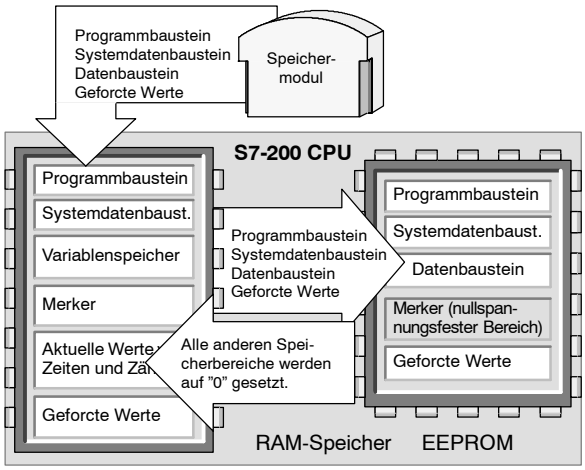


Bild 4-19 Zurückholen vom Speichermodul

Einstellen des Betriebszustands der S7-200 CPU

Die S7-200 verfügt über zwei Betriebsarten: Betriebszustand STOP und Betriebszustand RUN. Die Statusanzeigen auf der Vorderseite der CPU geben den aktuellen Betriebszustand an. Im Betriebszustand STOP führt die S7-200 das Programm nicht aus und Sie können ein Programm oder die CPU-Konfiguration in die CPU laden. Im Betriebszustand RUN bearbeitet die S7-200 das Programm.

- Die S7-200 verfügt über einen Betriebsartenschalter, mit dem Sie den Betriebszustand ändern können. Mit dem Betriebsartenschalter (befindet sich unter der vorderen Abdeckklappe der S7-200) können Sie den Betriebszustand manuell einstellen: Wenn Sie den Betriebsartenschalter in die Stellung STOP versetzen, wird die Programmbearbeitung gestoppt. Wenn Sie den Betriebsartenschalter in die Stellung RUN versetzen, wird die Programmbearbeitung begonnen. Wenn Sie den Betriebsartenschalter in die Stellung TERM (Terminal) versetzen, wird der Betriebszustand nicht verändert.

  Befindet sich der Betriebsartenschalter in einer der Positionen STOP oder TERM und wird die Spannungsversorgung unterbrochen, dann geht die S7-200 bei Spannungsrückkehr automatisch in den Betriebszustand STOP. Befindet sich der Betriebsartenschalter in der Stellung RUN, wenn die Spannungsversorgung unterbrochen wird, dann geht die S7-200 bei Spannungsrückkehr wieder in den Betriebszustand RUN.

- Mit STEP 7-Micro/WIN können Sie den Betriebszustand der Online-S7-200 ändern. Damit die Software den Betriebszustand steuern kann, müssen Sie den Betriebsartenschalter auf der S7-200 manuell in eine der Positionen TERM oder RUN bringen. Mit den Menübefehlen **Zielsystem > STOP** oder **Zielsystem > RUN** oder den entsprechenden Schaltflächen in der Funktionsleiste wechseln Sie den Betriebszustand.

- Sie können in Ihrem Programm die Operation STOP aufnehmen, die die S7-200 in den Betriebszustand STOP versetzt. Auf diese Weise können Sie in Abhängigkeit von Ihrer Logik die Programmbearbeitung unterbrechen. Ausführliche Informationen zur Operation In STOP gehen finden Sie im Kapitel 6.

Speichern von Variablenspeicher im EEPROM über das Programm

Sie können einen Wert (Byte, Wort oder Doppelwort) aus einer beliebigen Adresse im Variablenspeicher im EEPROM speichern. Eine Operation zum Speichern im EEPROM verlängert die Zykluszeit maximal um ca. 5 ms. Wenn Sie einen Wert in den Variablenspeicherbereich im EEPROM schreiben, wird der vorherige Wert an dieser Adresse überschrieben.

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

Tipp

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.

Kopieren des Variablenspeichers in den EEPROM

Sondermerkerbyte 31 (SMB31) weist die S7-200 an, einen Wert des Variablenspeichers in den Bereich des Variablenspeichers im EEPROM zu kopieren. Sondermerkerwort 32 (SMW32) speichert die Adresse des Werts, der kopiert wird. Bild 4-20 zeigt das Format von SMB31 und SMW32.

Gehen Sie folgendermaßen vor, wenn Sie die S7-200 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 4-20).
3. Setzen Sie SM31.7 auf 1.

Am Ende eines jeden Zyklus prüft die S7-200 SM31.7. Ist SM31.7 = 1, dann wird der angegebene Wert im EEPROM gespeichert. Die Operation ist beendet, wenn die S7-200 SM31.7 auf 0 zurücksetzt.

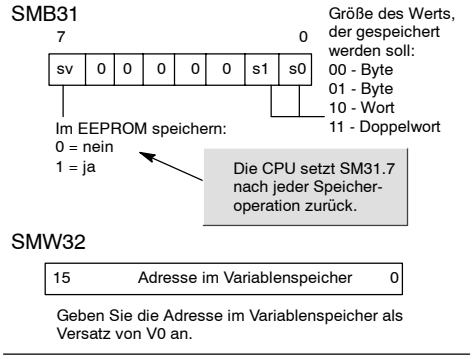


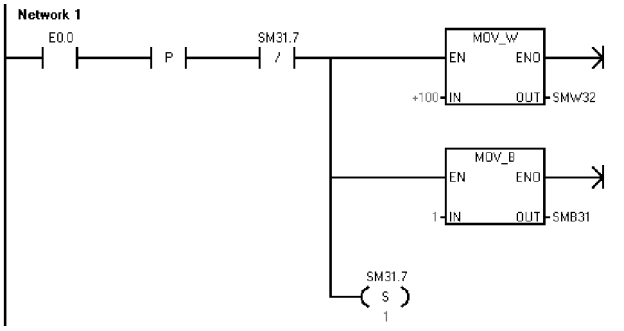
Bild 4-20 SMB31 und SMW32

Ändern Sie den Wert im Variablenspeicher solange nicht, bis die Operation ausgeführt ist.

Beispielprogramm: Kopieren des Variablenspeichers in den EEPROM

Dieses Beispiel überträgt VB100 in den EEPROM. Bei einer steigenden Flanke von E0.0 und wenn keine andere Übertragung in Bearbeitung ist, wird die Adresse im Variablenspeicher für die Übertragung in SMW32 geladen. Die Größe des zu übertragenden Variablenspeichers wird ausgewählt (1 = Byte, 2 = Wort, 3 = Doppelwort oder Realzahl). Dann wird SM31.7 gesetzt, damit die S7-200 die Daten am Ende des Zyklus überträgt.

Die S7-200 setzt SM31.7 automatisch zurück, wenn die Übertragung beendet ist.




```
Netzwerk 1    //Adresse im Variablenspeicher (VB100)
              //in den EEPROM übertragen.
LD      E0.0
EU
UN      SM31.7
MOVW    +100, SMW32
MOVB    1, SMB31
S       SM31.7, 1
```

Funktionen der S7-200

Die S7-200 bietet verschiedene Sonderfunktionen, mit denen Sie den Betrieb Ihrer S7-200 optimal an Ihre Anwendung anpassen können.

Das S7-200 Programm kann die Ein- und Ausgänge direkt lesen und schreiben

Der Befehlssatz der S7-200 verfügt über Operationen, mit denen Sie die physikalischen Ein- und Ausgänge direkt lesen bzw. direkt in die E/A schreiben können. Mit diesen 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 dienen.

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.

Tipp

Die S7-200 behandelt die Werte von Analogein- und Analogausgängen wie direkte Daten, sofern Sie keinen Analogeingabefilter aktiviert haben. Wenn Sie einen Wert in einen Analogausgang schreiben, wird der Ausgang sofort aktualisiert.

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 Bit- oder Byteformat zugegriffen werden muss. Auf die Prozessabbilder können Sie jedoch im Bit-, Byte, Wort- und Doppelwortformat zugreifen. Deshalb bieten die Prozessabbilder zusätzliche Flexibilität.

Das S7-200 Programm kann den Zyklus unterbrechen

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 im Zyklus möglich).

Interrupts werden von der S7-200 in der Reihenfolge ihres Auftretens und unter Beachtung der jeweiligen Prioritäten abgearbeitet. Ausführliche Informationen zu Interruptoperationen finden Sie in Kapitel 6.

Bei der S7-200 können Sie Kommunikationsaufgaben Bearbeitungszeit zuordnen

Sie können den Prozentsatz der Zykluszeit einstellen, der für die Bearbeitung von Kommunikationsaufträgen eingeplant wird, die in Zusammenhang mit Übersetzungen im Betriebszustand RUN oder mit dem Ausführungsstatus stehen. (Die Bearbeitung im Betriebszustand RUN und der Ausführungsstatus wird Ihnen von STEP 7-Micro/WIN geboten, damit Sie Ihr Programm einfacher testen können.) Wenn Sie den Prozentwert erhöhen, der für die Kommunikationsaufträge aufgebracht wird, erhöhen Sie auch die Zykluszeit, wodurch der gesteuerte Prozess länger dauert.

Der voreingestellte Prozentwert der Zykluszeit für die Kommunikationsaufträge beträgt 10 %. Diese Einstellung stellt einen vernünftigen Kompromiss für die Bearbeitung der Übersetzung bzw. des Status dar, während die Beeinträchtigung des gesteuerten Prozesses möglichst gering gehalten wird. Sie können diesen Wert um 5 % bis maximal 50 % anpassen. Wenn Sie die Zeit im Hintergrund des Zyklus für die Kommunikation einstellen möchten, gehen Sie folgendermaßen vor:

1. Wählen Sie den Menübefehl **Ansicht > Komponente > Systemdatenbaustein** und öffnen Sie das Register ”Zeit im Hintergrund”.
2. Ändern Sie die Eigenschaften für die Zeit im Hintergrund für die Kommunikation und bestätigen Sie mit ”OK”.
3. Laden Sie den geänderten Systemdatenbaustein in die S7-200.

Bild 4-21 Zeit im Hintergrund für die Kommunikation

Bei der S7-200 können Sie die Zustände der digitalen Ausgänge im Betriebszustand STOP einstellen

Mit Hilfe der Tabelle der Ausgänge der S7-200 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 vor dem Übergang in STOP befanden. Die Tabelle der Ausgänge ist Teil des Systemdatenbausteins, der in die S7-200 geladen und gespeichert wird und gilt nur für die digitalen Ausgänge.

1. Wählen Sie den Menübefehl **Ansicht > Komponente > Systemdatenbaustein** und öffnen Sie das Register ”Tabelle der Ausgänge”.
2. Aktivieren Sie das Kontrollkästchen ”Ausgänge einfrieren”, wenn Sie die Ausgänge in ihrem letzten Zustand einfrieren möchten.
3. Wenn Sie die Werte aus der Tabelle in die Ausgänge kopieren möchten, geben Sie die Werte in die Tabelle der Ausgänge ein, indem Sie die Kontrollkästchen der Ausgänge aktivieren, die Sie bei einem Wechsel von RUN in STOP auf 1 setzen möchten. (Die Voreinstellungen in der Tabelle sind alle Null.)
4. Bestätigen Sie Ihre Eingaben mit ”OK”.
5. Laden Sie den geänderten Systemdatenbaustein in die S7-200.

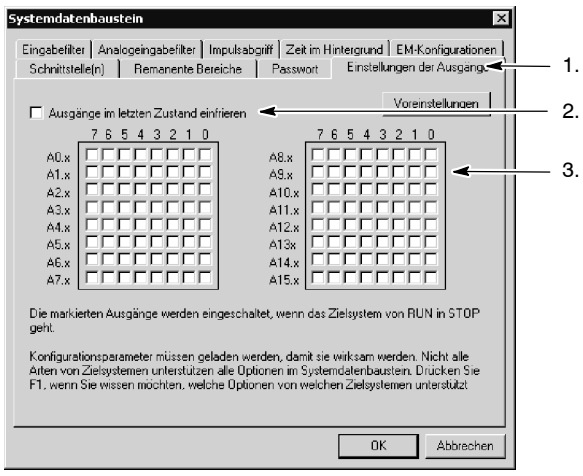


Bild 4-22 Konfigurieren der Tabelle der Ausgänge

Bei der S7-200 können Sie Speicher definieren, der bei Spannungsverlust remanent bleibt

Sie können maximal sechs Bereiche als remanent definieren und die Speicherbereiche auswählen, die Sie bei Spannungsverlust puffern möchten. 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. In der Voreinstellung sind die ersten 14 Bytes der Merker nicht remanent.

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

Tipp
Wenn Sie den Bereich MB0 bis MB13 als remanent definieren, wird eine besondere Funktion aktiviert, die diese Adressen beim Herunterfahren automatisch im EEPROM speichert.

So definieren Sie remanenten Speicher:

1. Wählen Sie den Menübefehl **Ansicht > Komponente > Systemdatenbaustein** und öffnen Sie das Register ”Remanente Bereiche”.
2. Wählen Sie die Bereiche im Speicher, die nach einem Spannungsverlust gepuffert werden sollen. Bestätigen Sie mit ”OK”.
3. Laden Sie den geänderten Systemdatenbaustein in die S7-200.

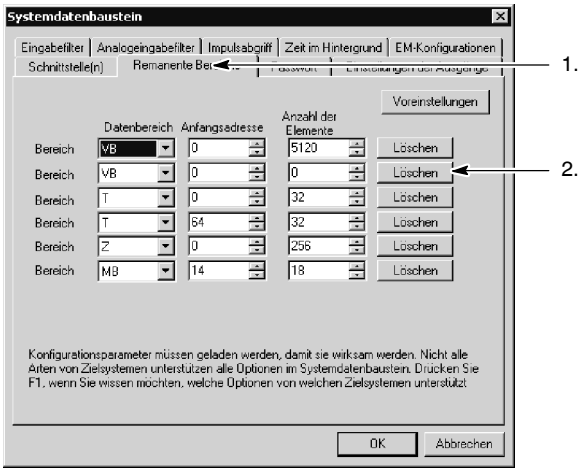


Bild 4-23 Remanenter Speicher

Bei der S7-200 können Sie digitale Eingänge filtern

Sie können für die S7-200 einen Eingabefilter auswählen, der für die physikalischen Eingänge eine Verzögerungszeit (einstellbar zwischen 0,2 ms bis 12,8 ms) definiert. Die Verzögerung dient dazu, ein Rauschen, welches unbeabsichtigte Signalzustandsänderungen an den Eingängen verursachen kann, in der Eingangsverdrahtung zu filtern.

Der Eingabefilter ist Teil des Systemdatenbausteins, der in den Speicher der S7-200 geladen und dort abgelegt wird. Die voreingestellte Filterzeit beträgt 6,4 ms. Wie Sie in Bild 4-24 sehen, gilt jede angegebene Verzögerung für eine Gruppe von Eingängen.

So konfigurieren Sie die Verzögerungszeiten für die Eingabefilter:

1. Wählen Sie den Menübefehl **Ansicht > Komponente > Systemdatenbaustein** und öffnen Sie das Register ”Eingabefilter”.
2. Geben Sie für jede Gruppe von Eingängen die gewünschte Verzögerung ein. Bestätigen Sie mit ”OK”.
3. Laden Sie den geänderten Systemdatenbaustein in die S7-200.

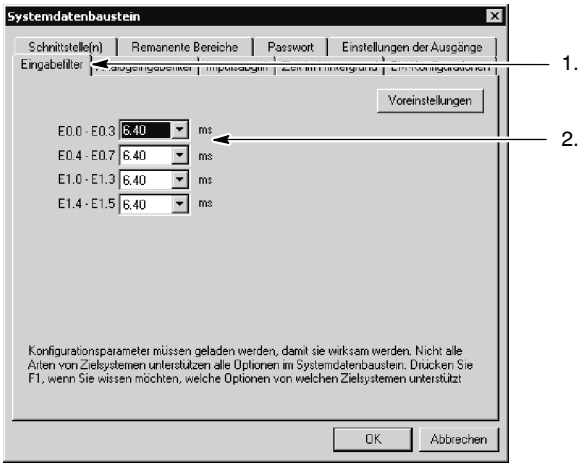


Bild 4-24 Konfigurieren der Eingabefilter

Tipp
Der Filter der digitalen Eingänge wirkt sich auf den Eingangswert aus, der in Werten von Operationen, Eingangs-Interrupts und Impulsabgriffen erscheint. Je nach der Filtereinstellung kann dies dazu führen, dass Ihr Programm ein Interruptereignis oder einen Impulsabgriff verpasst. Die schnellen Zähler zählen die Ereignisse der ungefilterten Eingänge.

Bei der S7-200 können Sie analoge Eingänge filtern

Bei der S7-200 können Sie den Software-Filter für einzelne Analogeingänge einstellen. Der gefilterte Wert entspricht dem Mittelwert aus einer zuvor eingestellten Anzahl von Abfragen der Analogeingänge. Die Angaben zum Filtern (Anzahl Abfragen und Totzeit) gelten für alle Analogeingänge, für die der Filter aktiviert wird.

Der Filter verfügt über eine Schnellreaktionsfunktion, damit sich größere Änderungen schnell auf den Filterwert auswirken. Der Filter verursacht einen Schrittfunktionswechsel zum letzten Analogeingabewert, wenn der Eingang eine bestimmte Änderung des aktuellen Werts überschreitet. Diese Änderung wird Totzeit genannt und wird in der Anzahl der Digitalwerte des Analogeingangs angegeben.

In der Voreinstellung ist der Filter für alle Analogeingänge aktiviert.

1. Wählen Sie den Menübefehl **Ansicht > Komponente > Systemdatenbaustein** und öffnen Sie das Register "Analogeingabefilter".
2. Wählen Sie die Analogeingänge, die Sie filtern möchten, die Anzahl der Abfragen und die Totzeit.
3. Bestätigen Sie mit "OK".
4. Laden Sie den geänderten Systemdatenbaustein in die S7-200.

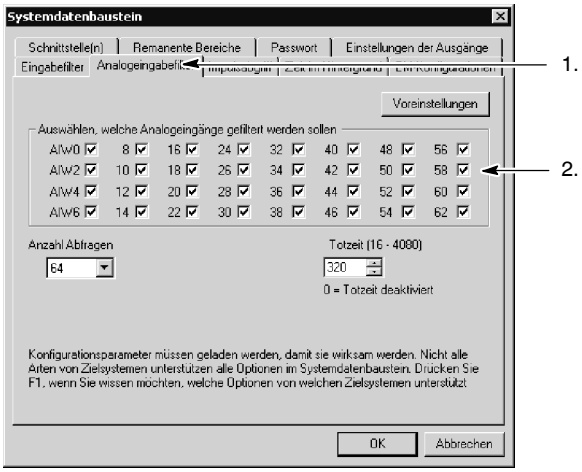


Bild 4-25 Analogeingabefilter

Tipp

Verwenden Sie den Analogfilter nicht bei Modulen, die digitale Daten oder Alarmanzeigen in Analogwörtern übergeben. Schalten Sie den Analogfilter immer für RTD-, Thermoelement- und AS-Interface-Master-Module aus.

Bei der S7-200 können Sie kurze Impulse erfassen

Die S7-200 verfügt über die Funktion "Impulsabgriff" für einige oder alle integrierten digitalen Eingänge. Mit der Funktion Impulsabgriff können Sie hohe oder niedrige Impulse erfassen, die eine so kurze Dauer haben, dass sie von der S7-200 leicht übersehen werden könnten, wenn die Digitaleingänge zu Beginn eines Zyklus gelesen werden. Ist die Funktion Impulsabgriff für einen Eingang aktiviert, wird ein Signalwechsel an dem Eingang so lange gehalten, bis die Aktualisierung des nächsten Zyklus stattfindet. Dadurch wird sichergestellt, dass ein Impuls, der nur kurze Zeit anhält, erfasst und gehalten wird, bis die S7-200 die Eingänge liest.

Sie können den Impulsabgriff für jeden integrierten Digitaleingang einzeln aktivieren.

So rufen Sie das Dialogfeld zum Konfigurieren des Impulsabgriffs auf:

1. Wählen Sie den Menübefehl **Ansicht > Komponente > Systemdatenbaustein** und öffnen Sie das Register "Impulsabgriff".
2. Aktivieren Sie die gewünschten Kontrollkästchen und bestätigen Sie mit "OK".
3. Laden Sie den geänderten Systemdatenbaustein in die S7-200.

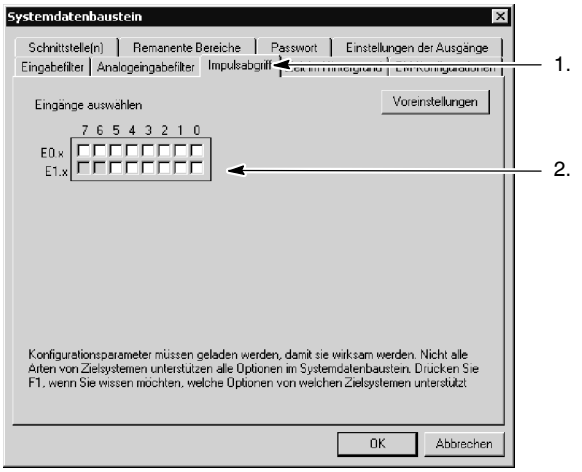


Bild 4-26 Impulsabgriff

Bild 4-27 zeigt die grundlegende Funktionsweise der S7-200 mit und ohne aktiviertem Impulsabgriff.

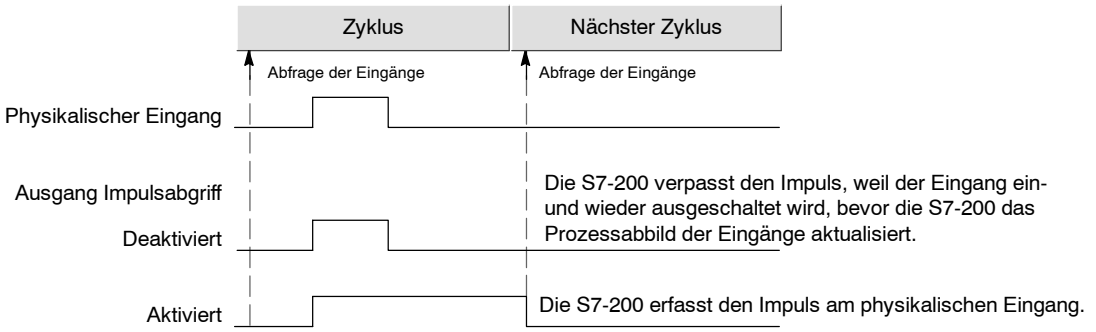


Bild 4-27 Betrieb der S7-200 mit aktiviertem und deaktiviertem Impulsabgriff

Weil sich der Impulsabgriff auf den Eingang auswirkt, nachdem dieser gefiltert wurde, müssen Sie die Zeit für den Eingangsfilter so einstellen, dass der Impuls nicht vom Filter zurückgenommen wird. Bild 4-28 zeigt eine schematische Darstellung des Digitaleingangskreises.

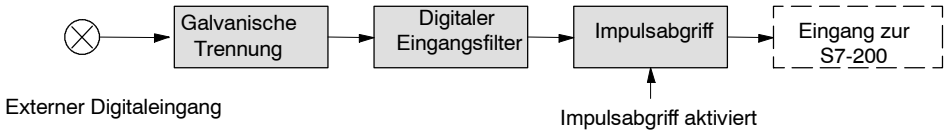


Bild 4-28 Digitaleingangskreis

Bild 4-29 zeigt die Reaktion des aktivierten Impulsabgriffs auf verschiedene Eingangsbedingungen. Gibt es mehr als einen Impuls in einem bestimmten Zyklus, wird nur der erste Impuls gelesen. Bei mehreren Impulsen in einem Zyklus sollten Sie die Interruptereignisse für steigende/fallende Flanken einsetzen. (Eine Auflistung der Interruptereignisse finden Sie in Tabelle 6-45.)

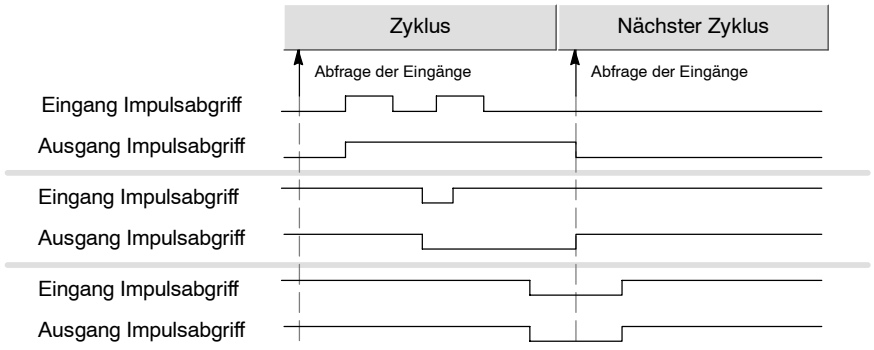


Bild 4-29 Reaktionen des Impulsabgriffs auf verschiedene Eingangsbedingungen

Die S7-200 bietet Ihnen Passwortschutz

Alle Varianten der S7-200 bieten Passwortschutz und schränken dadurch den Zugriff auf bestimmte Funktionen ein.

Durch das Einrichten eines Passworts können nur berechtigte Personen auf bestimmte Funktionen und den Speicher zugreifen. Ohne Passwort ist der uneingeschränkte Zugriff auf die S7-200 möglich. Besteht ein Passwortschutz, begrenzt die S7-200 die Funktionen, die in der Konfiguration des Passworts eingeschränkt sind.

Groß- und Kleinschreibung spielt beim Passwort keine Rolle.

Die S7-200 CPUs bieten drei verschiedene Schutzstufen mit unterschiedlichen Zugriffseinschränkungen (siehe Tabelle 4-3). 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. Die Voreinstellung für die S7-200 ist die Schutzstufe 1 (keine Einschränkungen).

Wenn Sie ein Netzpasswort eingeben, dann wirkt sich dieses Passwort nicht auf den Passwortschutz der S7-200 aus.

Tabelle 4-3 Schutzstufen bei der S7-200 CPU

CPU-Funktion	Schutzstufe 1	Schutzstufe 2	Schutzstufe 3
Lesen und Schreiben von Anwenderdaten	Zugriff erlaubt	Zugriff erlaubt	Zugriff erlaubt
Starten, Stoppen und Neustarten der CPU			
Lesen und Einstellen der Echtzeituhr			
Laden des Anwenderprogramms, der Daten und der CPU-Konfiguration aus der CPU	Zugriff erlaubt	Zugriff erlaubt	Passwort erforderlich
Laden in die CPU	Zugriff **erlaubt**	Passwort **erforderlich**	
Ermitteln des Ausführungsstatus			
Löschen des Programmbausteins, des Datenbausteins und des Systemdatenbausteins			
Forcen von Daten und Ausführen einzelner/mehrerer Zyklen			
Kopieren in das Speichermodul			
Schreiben in die Ausgänge im Betriebszustand STOP			

Ist ein Anwender berechtigt, auf geschützte Funktionen zuzugreifen, so sind deshalb nicht auch andere Anwender berechtigt, auf diese Funktionen zuzugreifen. Es hat immer nur ein Anwender uneingeschränkten Zugriff auf die S7-200.

Tipp

Nachdem Sie das Passwort eingegeben haben, bleibt die Schutzstufe des Passworts nach Trennen des Programmiergeräts von der S7-200 maximal eine Minute lang wirksam. Beenden Sie STEP 7-Micro/WIN immer, bevor Sie das Kabel trennen, damit kein anderer Anwender mit den Rechten auf das Programmiergerät zugreifen kann.

Einrichten eines Passworts für die S7-200

Im Dialogfeld "Systemdatenbaustein" (Bild 4-30) können Sie ein Passwort für die S7-200 einrichten:

1. Wählen Sie den Menübefehl **Ansicht > Komponente > Systemdatenbaustein**, um das Dialogfeld "Systemdatenbaustein" aufzurufen. Öffnen Sie das Register "Passwort".
2. Wählen Sie die gewünschte Schutzstufe für die S7-200.
3. Geben Sie das Passwort ein und bestätigen Sie es.
4. Bestätigen Sie mit "OK".
5. Laden Sie den geänderten Systemdatenbaustein in die S7-200.

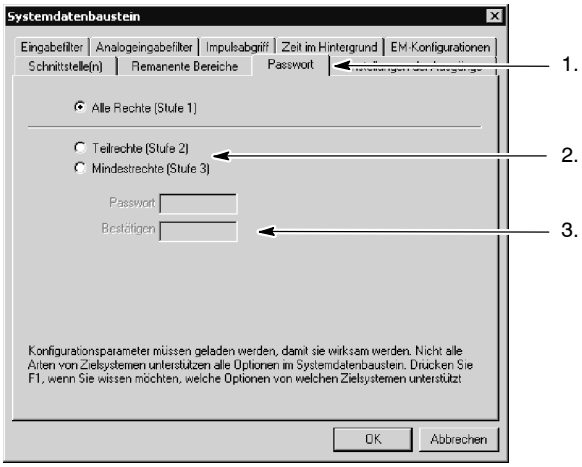


Bild 4-30 Anlegen eines Passworts

Vorgehensweise bei verlorenem Passwort

Wenn Sie Ihr Passwort vergessen haben, müssen Sie den Speicher der S7-200 urlöschen und Ihr Programm erneut in die CPU laden. Beim Urlöschen des Speichers wird die S7-200 zunächst in den Betriebszustand STOP versetzt und anschließend auf die vom Werk voreingestellten Werte zurückgesetzt. Ausnahmen sind die Netzadresse, die Baudrate und die Echtzeituhr, die nicht zurückgesetzt werden. So führen Sie ein Urlöschen Ihres Programms in der S7-200 durch:

1. Wählen Sie den Menübefehl **Zielsystem > Urlöschen**, um das Dialogfeld ”Urlöschen” aufzurufen.
2. Wählen Sie alle drei Bausteine und bestätigen Sie Ihre Eingaben mit ”OK”.
3. Wurde ein Passwort eingerichtet, zeigt STEP 7-Micro/WIN ein Dialogfeld zur Eingabe des Passworts an. Zum Löschen des Passworts geben Sie in diesem Dialogfeld CLEARPLC ein, um das Urlöschen durchzuführen. (Bei der Eingabe des Passworts CLEARPLC spielt die Groß- und Kleinschreibung keine Rolle.)

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 S7-200 werden die Ausgänge ausgeschaltet (Analogausgänge werden mit einem bestimmten Wert eingefroren).

Ist die S7-200 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.

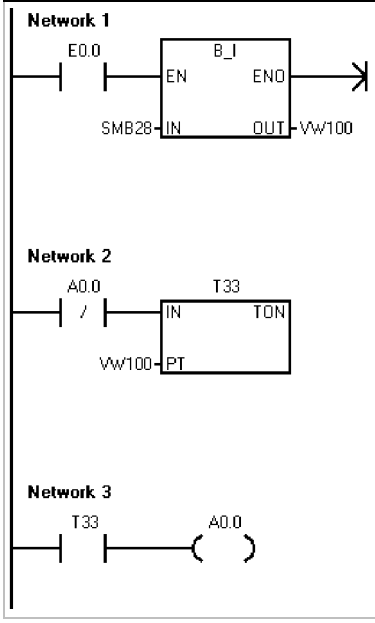
Treffen Sie deshalb alle notwendigen Sicherheitsvorkehrungen und vergewissern Sie sich, dass sich Ihr Prozess in einem sicheren Zustand befindet, bevor Sie die S7-200 urlöschen.

Die S7-200 verfügt über Analogpotentiometer

Die Analogpotentiometer befinden sich unter der vorderen Abdeckklappe der CPU. Mit diesen Potentiometern können Sie Werte, die in Bytes von Sondermerkern (SMB) 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. 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.

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 Wiederholbarkeit von ±2.

Beispielprogramm für das Referenzieren eines über Analogpotentiometer eingegebenen Werts




```
Netzwerk 1    //Analogpotentiometer 0 (SMB28) lesen.
              //Wert als ganze Zahl in VW100 speichern.
LD    E0.0
BTI   SMB28, VW100

Netzwerk 2    //Ganzzahligen Wert (VW100) als Voreinstellung für eine Zeit
              //nutzen.
LDN   A0.0
TON   T33, VW100

Netzwerk 3    //A0.0 einschalten, wenn T33 den voreingestellten Wert er
              //reicht.
LD    T33
=     A0.0
```

Die S7-200 umfasst schnelle Ein-/Ausgänge

Schnelle Zähler

Die S7-200 verfügt über integrierte schnelle Zähler, die externe Hochgeschwindigkeitsereignisse zählen, ohne die Leistungsfähigkeit der S7-200 zu beeinträchtigen. Im Anhang A finden Sie die von Ihrer CPU-Variante unterstützen Geschwindigkeiten. Jeder Zähler verfügt über besondere Eingänge, die Funktionen wie Taktgeber, Richtungssteuerung, Rücksetzen und Starten unterstützen. Sie können die Zählgeschwindigkeit durch Einstellung verschiedener A/B-Zähler variieren. Ausführliche Informationen zu schnellen Zählern finden Sie in Kapitel 6.

Schnelle Impulsausgänge

Die S7-200 unterstützt schnelle Impulsausgänge, wobei die Ausgänge A0.0 und A0.1 entweder eine Impulsfolge (PTO) oder eine Impulsdauermodulation (PWM) erzeugen.

Die Funktion PTO liefert einen Rechteckausgang (50 % relative Einschaltdauer) für eine angegebene Anzahl Impulse (zwischen 1 und 4.294.967.295 Impulsen) und eine angegebene Zykluszeit (in Mikrosekunden- oder Millisekunden-Inkrementen von 50 µs bis 65.535 µs oder von 2 ms bis 65.535 ms). Sie können die Funktion PTO so programmieren, dass entweder eine Impulsfolge oder eine Impulsprofil erzeugt wird, das aus mehreren Impulsfolgen besteht. Sie können beispielsweise mit einem Impulsprofil einen Schrittmotor über eine einfache Abfolge von Hochlauf, Betrieb und Herunterfahren oder über andere komplexere Folgen steuern. Das Impulsprofil kann aus maximal 255 Segmenten bestehen, wobei ein Segment jeweils einem der Arbeitsschritte Hochlauf, Betrieb oder Herunterfahren entspricht.

Die Funktion PWM liefert eine feste Zykluszeit mit einer variablen relativen Einschaltdauer, wobei die Zykluszeit und die Impulsdauer in Mikrosekunden- oder in Millisekunden-Inkrementen angegeben werden. Die Zykluszeit liegt zwischen 50 µs und 65.535 µs oder zwischen 2 ms und 65.535 ms. Die Impulsdauer liegt zwischen 0 µs und 65.535 µs 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 schnellen Impulsausgängen finden Sie im Kapitel 6.

Programmiergrundlagen, Konventionen und Funktionen

5

Die S7-200 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 S7-200. STEP 7-Micro/WIN stellt Ihnen verschiedene Werkzeuge und Funktionen zum Entwerfen, Implementieren und Testen Ihres Programms zur Verfügung.

In diesem Kapitel

Richtlinien für das Entwerfen einer Automatisierungslösung mit einer Micro-SPS 48
Wesentliche Elemente eines Programms 49
Erstellen Ihrer Programme mit STEP 7-Micro/WIN 51
Befehlssätze SIMATIC und IEC 1131-3 53
Konventionen in den Programm-Editoren 54
Erstellen Ihres Steuerungsprogramms mit Assistenten 56
Fehlerbehebung in der S7-200 56
Zuordnen von Adressen und Anfangswerten im Datenbaustein-Editor 58
Verwenden der Symboltabelle für die symbolische Adressierung von Variablen 58
Lokale Variablen 59
Beobachten Ihres Programms mit der Statustabelle 59
Erstellen einer Operationsbibliothek 60
Funktionen zum Testen Ihres Programms 60

Richtlinien für das Entwerfen einer Automatisierungslösung mit einer Micro-SPS

Es gibt viele Methoden, ein Automatisierungssystem zu entwerfen. Die folgenden allgemeinen Richtlinien können sich auf viele Projekte beziehen. Dabei sollten Sie sich selbstverständlich an die Verfahrensanweisungen in Ihrem Unternehmen halten und Ihre eigenen Erfahrungen berücksichtigen.

Gliedern Ihres Prozesses bzw. Ihrer Anlage

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

Beschreiben der Funktionsbereiche

Beschreiben Sie die Funktionsweise jedes Prozess- bzw. Anlagenabschnitts. Zeichnen Sie folgende Punkte auf: Eingänge und Ausgänge, Beschreibung der Funktionsweise, Zustände, die erreicht sein müssen, bevor die Bedienung von Aktoren (z.B. Magnetschalter, Motoren und Antriebe) möglich wird, Beschreibung der Bedienerschnittstelle und alle Schnittstellen zu anderen Abschnitten des Prozesses bzw. der Maschine.

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 Programmeingriffen, die unabhängig von der S7-200 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 S7-200 erkannt werden.
- ❏ Definieren Sie, wie die S7-200 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 S7-200 gesperrt werden.
- ❏ Lassen Sie von den unabhängigen Stromkreisen Statusinformationen an die S7-200 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, z.B. Display, Schalter und Lampen, der Operator-Station
- ❏ Verdrahtungspläne mit den zugehörigen Ein- und Ausgängen der S7-200 oder der Erweiterungsmodule

Zeichnen der Konfigurationspläne

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

- ❑ Position der einzelnen S7-200 in Bezug zum Prozess bzw. zur Anlage
- ❑ Mechanische Anordnung der S7-200 und Erweiterungsmodule (einschließlich Schränke usw.)
- ❑ Verdrahtungspläne für alle S7-200 und Erweiterungsmodule (einschließlich Gerätenummern, Kommunikationsadressen und Adressen der Ein- und Ausgänge)

Auflisten der symbolischen Namen (optional)

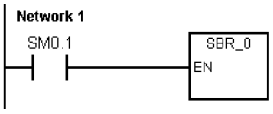
Wenn Sie sich für die symbolische Adressierung entscheiden, müssen Sie den absoluten Adressen symbolische Namen zuordnen. Geben Sie nicht nur die physikalischen Ein- und Ausgänge, sondern auch alle anderen Elemente an, die Sie in Ihrem Programm verwenden.

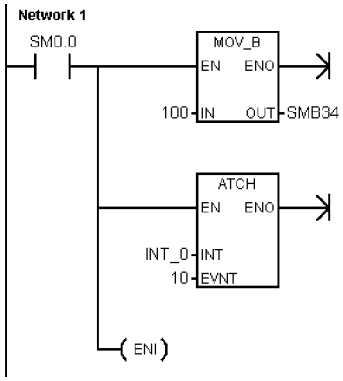
Wesentliche Elemente eines Programms

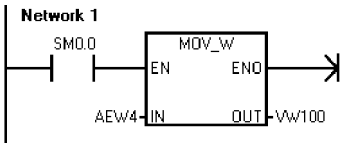
Ein Programmbaustein besteht aus aus ausführbarem Code und Kommentaren. Der ausführbare Code besteht aus einem Hauptprogramm (OB1) sowie Unter- und Interruptprogrammen. Der Code wird übersetzt und in die S7-200 geladen. Die Kommentare werden nicht übersetzt und nicht geladen. Mit diesen Organisationseinheiten (Hauptprogramm, Unterprogramme und Interruptprogramme) können Sie Ihr Steuerungsprogramm strukturieren.

Das folgende Beispiel zeigt ein Programm, das ein Unterprogramm und ein Interruptprogramm umfasst. Dieses Beispielprogramm liest mit Hilfe eines zeitgesteuerten Interrupts alle 100 ms den Wert eines Analogeingangs.

Beispiel: Wesentliche Elemente eines Programms

MAIN		Netzwerk 1 //Im ersten Zyklus Unterprogramm 0 aufrufen. LD SM0.1 CALL SBR0

SBR0		Netzwerk 1 //Intervall für zeitgesteuerten Interrupt
		//auf 100 ms setzen.
		//Interrupt 0 freigeben.
		LD SM0.0
		MOVB 100, SMB34
		ATCH INT_0, 10
		ENI

INT0		Netzwerk 1 //Analogeingang 4 abfragen. LD SM0.0 MOVW AEW4, VW100

Hauptprogramm

Dieser Hauptteil des Programms enthält die Operationen, die Ihre Anwendung steuern. Die S7-200 führt diese Operationen der Reihe nach einmal je Zyklus aus. Das Hauptprogramm wird auch OB1 genannt.

Unterprogramme

Diese optionalen Elemente Ihres Programms werden nur ausgeführt, wenn sie aufgerufen werden: und zwar vom Hauptprogramm, von einem Interruptprogramm oder von einem anderen Unterprogramm. Unterprogramme sind nützlich, wenn Sie eine Funktion mehrfach ausführen möchten. Damit Sie die Logik nicht an jeder Stelle, an der Sie die Funktion ausführen möchten, ins Hauptprogramm aufnehmen müssen, schreiben Sie die Logik einmal in ein Unterprogramm und rufen dieses Unterprogramm so oft auf, wie es während der Bearbeitung des Hauptprogramms erforderlich ist. Unterprogramme bieten mehrere Vorteile:

- ❏ Durch Unterprogramme wird die Gesamtgröße des Programms verringert.
- ❏ Der Einsatz von Unterprogrammen verkürzt die Zykluszeit, weil Sie den Code aus dem Hauptprogramm entfernt haben. Die S7-200 wertet den Code im Hauptprogramm in jedem Zyklus aus, ganz gleich, ob der Code ausgeführt wird oder nicht. Der Code im Unterprogramm wird von der S7-200 jedoch nur dann ausgewertet, wenn das Unterprogramm aufgerufen wird. In Zyklen, in denen das Unterprogramm nicht aufgerufen wird, wird auch der Code nicht ausgewertet.
- ❏ Durch den Einsatz von Unterprogrammen erstellen Sie portierbaren Code. Sie können den Code für eine Funktion in einem Unterprogramm ausgrenzen und dann das Unterprogramm ohne viel Aufwand in andere Programme kopieren.

Tipp

Die Verwendung von Adressen im Variablenspeicher schränkt die Portabilität von Unterprogrammen ein, weil die Zuordnung von Adressen im Variablenspeicher des einen Programms mit der Zuordnung der Adressen im anderen Programm in Konflikt stehen kann. Unterprogramme, die alle Adressen in der lokalen Variablentabelle (Lokaldaten) zuordnen, sind dagegen gut portierbar, weil es nicht zu Adresskonflikten zwischen dem Unterprogramm und anderen Teilen des Programms kommen kann, wenn lokale Variablen verwendet werden.

Interruptprogramme

Diese optionalen Komponenten eines Programms reagieren auf bestimmte Interruptereignisse. Sie entwerfen ein Interruptprogramm, damit es ein bestimmtes vordefiniertes Interruptereignis bearbeitet. Wenn das jeweilige Ereignis auftritt, führt die S7-200 das Interruptprogramm aus.

Die Interruptprogramme werden nicht vom Hauptprogramm aufgerufen. Sie ordnen das Interruptprogramm einem Interruptereignis zu. Die S7-200 führt die Operationen im Interruptprogramm nur aus, wenn das Interruptereignis auftritt.

Tipp

Weil es nicht vorhersehbar ist, wann die S7-200 einen Interrupt erzeugt, ist es empfehlenswert, die Anzahl der Variablen zu begrenzen, die sowohl im Interruptprogramm als auch an anderen Stellen im Programm verwendet werden.

Wenn Sie die lokale Variablentabelle des Interruptprogramms verwenden, stellen Sie sicher, dass Ihr Interruptprogramm nur den temporären Speicher verwendet und keine Daten an anderen Stellen im Programm überschreibt.

Es gibt eine Reihe von Programmiertechniken, mit denen Sie sicherstellen, dass Fehler beim gemeinsamen Nutzen von Daten im Hauptprogramm und im Interruptprogramm vermieden werden. Diese Techniken werden im Kapitel 6 bei den Interruptoperationen beschrieben.

Sonstige Elemente des Programms

Andere Bausteine enthalten Informationen für die S7-200. Sie können diese Bausteine laden, wenn Sie Ihr Programm laden.

System-datenbaustein

Systemdatenbaustein

Im Systemdatenbaustein können Sie verschiedene Hardware-Optionen für die S7-200 konfigurieren.

Daten-baustein

Datenbaustein

Der Datenbaustein speichert die Werte für verschiedene Variablen (Variablenspeicher), die Ihr Programm verwendet. Im Datenbaustein können Sie Anfangswerte für die Daten eingeben.

Erstellen Ihrer Programme mit STEP 7-Micro/WIN

Zum Aufrufen von STEP 7-Micro/WIN doppelklicken Sie auf das Symbol von STEP 7-Micro/WIN oder wählen den Menübefehl **Start > SIMATIC > STEP-7 MicroWIN 3.2**. Wie Sie in Bild 5-1 sehen, bietet das Projektfenster in STEP 7-Micro/WIN einen komfortablen Arbeitsbereich zum Erstellen von Steuerungsprogrammen.

Die Funktionsleisten bieten Schaltflächen für häufig verwendete Menübefehle. Sie können die Funktionsleisten einzeln anzeigen oder ausblenden.

Die Navigationsleiste bietet Symbole für den Zugriff auf verschiedene Programmierfunktionen von STEP 7-Micro/WIN.

Der Operationsbaum zeigt alle Objekte des Projekts und die Operationen an, mit denen Sie Ihr Steuerungsprogramm erstellen können. Sie können die Operationen mit der Maus aus dem Baum in Ihr Programm ziehen oder Sie können auf eine Operation doppelklicken, um sie an der Cursorposition in den Programm-Editor einzufügen.

Der Programm-Editor enthält die Programmlogik und eine lokale Variablentabelle, in der Sie temporären lokalen Variablen symbolische Namen zuordnen können. Unterprogramme und Interruptprogramme werden als Register am unteren Rand des Programm-Editors dargestellt. Sie können auf die Laschen der Register klicken, um die Unterprogramme, die Interruptprogramme oder das Hauptprogramm anzuzeigen.

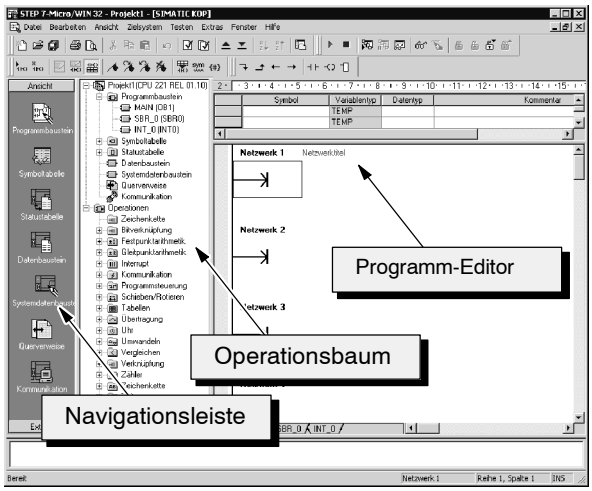


Bild 5-1 STEP 7-Micro/WIN

Programm -Editor

STEP 7-Micro/WIN verfügt über drei Editoren, mit denen Sie Ihr Programm anlegen können: Kontaktplan (KOP), Anweisungsliste (AWL) und Funktionsplan (FUP). Programme, die in diesen Programm-Editoren geschrieben wurden, können mit einigen Einschränkungen mit anderen Programm-Editoren angezeigt und bearbeitet werden.

Funktionen des AWL-Editors

Der AWL-Editor zeigt das Programm als textbasierte Sprache an. Mit dem AWL-Editor können Sie Programme erstellen, indem Sie die Mnemonik der Operationen eingeben. Im AWL-Editor können Sie Programme erstellen, die Sie im KOP- oder FUP-Editor nicht erstellen können. Dies ist darauf zurückzuführen, dass Sie in AWL in der "Muttersprache" der S7-200 schreiben und keinen grafischen Editor einsetzen, in dem es einige Einschränkungen gibt, damit die Schaltpläne korrekt gezeichnet werden können. Wie Sie in Bild 5-2 sehen, ist diese Programmierung im Text-Editor der Programmierung mit Assemblersprachen sehr ähnlich.

Die S7-200 führt die Operationen in der vom Programm vorgegebenen Reihenfolge, von oben nach unten, aus und beginnt dann von vorne.

In AWL wird die Steuerungslogik mit Hilfe eines logischen Stack gelöst. In AWL müssen Sie die Operationen für die Stackbearbeitung eingeben.

```
LD    E0.0    //Einen Eingang lesen
U     E0.1    //mit anderem Eingang durch
              //UND verknüpfen
=     A1.0    //Wert in Ausgang 1 schreiben
```

Bild 5-2 Beispiel für ein AWL-Programm

Wenn Sie mit dem AWL-Editor programmieren möchten, sollten Sie folgendes beachten:

- AWL ist am besten für erfahrene Programmierer geeignet.
- In AWL können Sie einige Programmieraufgaben lösen, die sich in KOP und FUP nicht so leicht lösen lassen.
- Im AWL-Editor können Sie nur den SIMATIC-Befehlssatz verwenden.
- Sie können sich in AWL jederzeit ein Programm, das Sie im KOP- oder FUP-Editor erstellt haben, ansehen und es auch bearbeiten, während dies umgekehrt nicht immer möglich ist. Sie können sich ein Programm, das Sie im AWL-Editor geschrieben haben, nicht immer im KOP- oder FUP-Editor anzeigen lassen.

Funktionen des KOP-Editors

Der KOP-Editor zeigt das Programm als grafische Darstellung wie in elektrischen Schaltplänen an. KOP-Programme ermöglichen dem Programm, den elektrischen Stromfluss von einer Spannungsquelle über eine Reihe von logischen Eingangsbedingungen, die wiederum logische Ausgangsbedingungen aktivieren, nachzubilden. Ein KOP-Programm verfügt über eine linke Stromschiene mit Signalfluss. Bei geschlossenen Kontakten kann der Signalfluss über die Kontakte zu den nächsten Elementen fließen. Geöffnete Kontakte verhindern den Signalfluss.

Die Logik wird in Netzwerke unterteilt. Das Programm wird Netzwerk für Netzwerk ausgeführt, und zwar von links nach rechts und von oben nach unten. Bild 5-3 zeigt ein Beispiel für ein KOP-Programm. Die verschiedenen Operationen werden von grafischen Symbolen dargestellt und umfassen drei wesentliche Arten.

5

Kontakte stellen logische Eingangsbedingungen wie Schalter, Taster oder interne Bedingungen dar.

Spulen stellen logische Ausgangsbedingungen wie Lampen, Motorstarter, zwischengeschaltete Relais oder interne Ausgangsbedingungen dar.

Boxen stellen zusätzliche Funktionen wie Zeiten, Zähler oder arithmetische Operationen dar.

Wenn Sie mit dem KOP-Editor programmieren möchten, sollten Sie folgendes beachten:

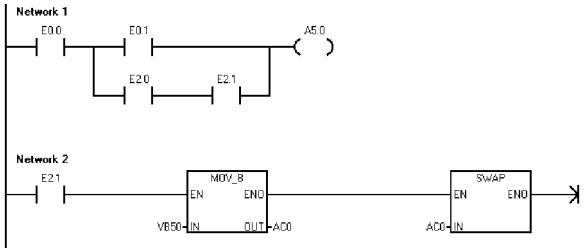


Bild 5-3 Beispiel für ein KOP-Programm

- ❏ Der Kontaktplan eignet sich für unerfahrene Programmierer.
- ❏ Grafische Darstellungen sind leichter zu verstehen und weltweit verbreitet.
- ❏ Im KOP-Editor können Sie sowohl den SIMATIC als auch den IEC 1131-3 Befehlssatz verwenden.
- ❏ Sie können sich ein Programm, das Sie im SIMATIC KOP-Editor geschrieben haben, immer im AWL-Editor anzeigen lassen.

Funktionen des FUP-Editors

Der FUP-Editor zeigt das Programm als grafische Darstellung wie Verknüpfungsglieder in Funktionsschaltplänen an. Es gibt keine Kontakte und Spulen wie im KOP-Editor, sondern äquivalente Operationen, die als Boxen dargestellt werden.

Bild 5-4 zeigt ein Beispiel für ein FUP-Programm.

In einem Funktionsplan gibt es keine linke und rechte Stromschiene, deshalb drückt der Begriff ”Signalfluss” den Fluss der Steuerung durch die FUP-Funktionsbausteine aus.

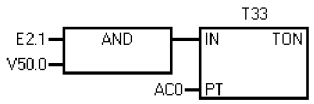


Bild 5-4 Beispiel für ein FUP-Programm

Aus diesem Grund wird der Pfad des Zustands 1 durch die FUP-Elemente als Signalfluss bezeichnet. Die Herkunft eines Eingangs für Signalfluss sowie das Ziel eines Ausgangs für Signalfluss können direkt einem Operanden zugewiesen werden.

Die Programmlogik entsteht aus den Verbindungen zwischen diesen Boxen. D.h. der Ausgang einer Operation (z.B. einer UND-Box) gibt eine weitere Operation frei (z.B. eine Zeit), um die erforderliche Logik zu erstellen. Durch dieses Konzept können Sie eine große Bandbreite von Steuerungsaufgaben lösen.

Wenn Sie mit dem FUP-Editor programmieren möchten, sollten Sie folgendes beachten:

- ❏ Die grafische Darstellung eines Funktionsplans ist gut geeignet, um den Programmablauf hervorzuheben.
- ❏ Im FUP-Editor können Sie sowohl den SIMATIC als auch den IEC 1131-3 Befehlssatz verwenden.
- ❏ Sie können sich ein Programm, das Sie im SIMATIC FUP-Editor geschrieben haben, immer im AWL-Editor anzeigen lassen.

Befehlssätze SIMATIC und IEC 1131-3

Die meisten Automatisierungssysteme bieten im wesentlichen die gleichen Arten von Operationen, doch es gibt geringfügige Unterschiede in Darstellung, Funktion usw. zwischen den verschiedenen Herstellern. In den letzten Jahren hat die Internationale Elektrotechnische Kommission (IEC) globale Normen entwickelt, die sich auf viele Aspekte der Programmierung von Automatisierungssystemen beziehen. Im wesentlichen fordert die Norm die verschiedenen Hersteller von Automatisierungssystemen dazu auf, Operationen zur Verfügung zu stellen, die in Darstellungsart und Funktionsweise gleich sind.

Ihre S7-200 bietet Ihnen zwei verschiedene Befehlssätze, mit denen Sie eine Bandbreite von Automatisierungslösungen programmieren können. Der IEC-Befehlssatz entspricht der Norm IEC 1131-3 für die Programmierung von Automatisierungssystemen. Der SIMATIC-Befehlssatz ist spezifisch für die S7-200 ausgelegt.

Tipp
Im IEC-Modus zeigt STEP 7-Micro/WIN im Operationsbaum eine rote Raute ⟨♦) neben den Operationen an, die nicht in der Norm IEC 1131-3 definiert sind.

Es gibt einige wesentliche Unterschiede zwischen den Befehlssätzen SIMATIC und IEC.

- Der IEC-Befehlssatz umfasst nur die Operationen, die bei allen Herstellern von Automatisierungssystemen genormt sind. Einige Operationen, die sich standardmäßig im SIMATIC-Befehlssatz befinden, sind keine genormten IEC 1131-3 Operationen. Sie stehen als nicht genormte Operationen zur Verfügung, doch wenn Sie diese Operationen in einem Programm verwenden, ist das Programm nicht mehr kompatibel mit IEC 1131-3.
- Bei einigen IEC-Boxen kann mit mehreren Datenformaten gearbeitet werden. Dieses Verhalten wird oft als ”Überladen” bezeichnet. Beispiel: Anstatt für die Operationen ADD_I (Ganze Zahlen (16 Bit) addieren) und ADD_R (Realzahlen addieren) einzelne Boxen zur Verfügung zu stellen, prüft die IEC-Operation ADD das Format der Daten, die addiert werden sollen, und wählt automatisch die korrekte Operation für die S7-200 aus. Hierdurch sparen Sie sich einigen Programmieraufwand.
- Wenn Sie mit IEC-Operationen arbeiten, werden die Parameter der Operation automatisch auf das richtige Datenformat geprüft, z.B. vorzeichenlose ganze Zahl im Vergleich zu ganze Zahl mit Vorzeichen. Wenn Sie beispielsweise versuchen, einen ganzzahligen Wert für eine Operation einzugeben, die einen Bitwert benötigt (ein/aus), wird Ihnen ein Fehler angezeigt. Diese Funktion verringert Syntaxfehler beim Programmieren.

Wenn Sie den SIMATIC-Befehlssatz oder den IEC-Befehlssatz auswählen, sollten Sie folgendes beachten:

- Die SIMATIC-Operationen haben im allgemeinen die kürzesten Ausführungszeiten. Einige IEC-Operationen haben evtl. längere Ausführungszeiten.
- Einige IEC-Operationen, z.B. Zeiten, Zähler, Multiplizieren und Dividieren, funktionieren anders als die entsprechenden SIMATIC-Operationen.
- Alle drei Programm-Editoren (KOP, AWL, FUP) arbeiten mit dem SIMATIC-Befehlssatz. Mit IEC-Operationen können Sie nur im KOP- und im FUP-Editor arbeiten.
- Die Funktionsweise von IEC-Operationen ist bei allen Herstellern von Automatisierungssystemen genormt, d.h. IEC-konforme Programme können Sie unabhängig vom Automatisierungssystem erstellen.
- Der SIMATIC-Befehlssatz enthält mehr Operationen als in der IEC-Norm definiert sind. Sie können deshalb jederzeit SIMATIC-Operationen in Ihr IEC-Programm aufnehmen.
- IEC 1131-3 verlangt, dass für Variablen ein Typ deklariert wird, und es unterstützt das Prüfen des Datentyps durch das System.

Konventionen in den Programm-Editoren

In STEP 7-Micro/WIN gelten folgende Konventionen für alle Editoren:

- Das Zeichen # vor einem symbolischen Namen (#var1) zeigt an, dass das Symbol lokalen Geltungsbereich hat.
- Bei IEC-Operationen kennzeichnet das Symbol % eine direkte Adresse.
- Das Operandensymbol “?.?” oder “????” bedeutet, dass der Operand konfiguriert werden muss.

KOP-Programme sind in Strompfade unterteilt, die Netzwerke genannt werden. Ein Netzwerk ist eine geordnete Anordnung von Kontakten, Spulen und Boxen, die miteinander verbunden sind und einen geschlossenen Stromkreis bilden. Es sind keine Kurzschlüsse, offenen Stromkreise oder umgekehrter Signalfluss vorhanden. In STEP 7-Micro/WIN können Sie in Ihren KOP-Programmen für jedes einzelne Netzwerk Kommentare erstellen. In FUP unterteilen und kommentieren Sie Ihr Programm auf der Basis von Netzwerken.

5

In AWL-Programmen gibt es keine Netzwerke, doch Sie können das Schlüsselwort NETWORK verwenden, um Ihre Programme zu strukturieren.

Spezifische Konventionen des KOP-Editors

Im KOP-Editor können Sie mit den Funktionstasten F4, F6 und F9 auf Ihrer Tastatur Kontakte, Boxen und Spulen einfügen. Im KOP-Editor gibt es die folgenden Konventionen:

- Das Symbol “--->>” ist ein offener Schaltkreis bzw. benötigt einen Anschluss für Signalfluss.
- Das Symbol “→|” zeigt an, dass der Ausgang für eine Operation, die in Kaskadenschaltung angeschlossen oder in Reihe geschaltet werden kann, optionalen Signalfluss darstellt.
- Das Symbol “>>” zeigt an, dass Sie Signalfluss verwenden können.

Spezifische Konventionen des FUP-Editors

Im FUP-Editor können Sie mit den Funktionstasten F4, F6 und F9 auf Ihrer Tastatur die Operationen UND und ODER sowie Boxen einfügen. Im FUP-Editor gibt es die folgenden Konventionen:

- Das Symbol “--->>” zeigt bei einem Operanden EN Signalfluss oder einen Operanden an. Es kann auch einen offenen Schaltkreis bzw. einen benötigten Signalfluss-Anschluss darstellen.
- Das Symbol “→|” zeigt an, dass der Ausgang für eine Operation, die in Kaskadenschaltung angeschlossen oder in Reihe geschaltet werden kann, optionalen Signalfluss darstellt.
- Die Symbole “<<” und “>>” zeigen an, dass ein Wert oder Signalfluss verwendet werden kann.
- Kennzeichen für Negationen: Die logische Bedingung NOT oder der umgekehrte Zustand des Operanden bzw. des Signalflusses wird durch einen kleinen Kreis vor dem jeweiligen Eingang dargestellt. Im Bild 5-5 ist A0.0 gleich der Negation von E0.0 UND E0.1. Negationen sind nur bei booleschen Signalen gültig, die als Parameter oder Signalfluss angegeben werden können.

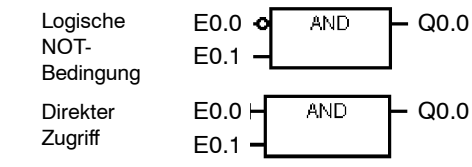


Bild 5-5 FUP-Konventionen

- Kennzeichen für direkte Ansteuerung: Im FUP-Editor wird durch die senkrechte Linie am Eingang der FUP-Operation angezeigt, dass es sich bei einem booleschen Operanden um einen direkt angesteuerten Wert handelt (siehe Bild 5-5). Ist ein Eingang für die direkte Ansteuerung gekennzeichnet, wird der angegebene physikalische Eingang direkt gelesen. Direkt angesteuerte Operatoren sind nur bei physikalischen Eingängen gültig.
- Box ohne Eingang oder Ausgang: Eine Box ohne Eingang kennzeichnet eine Operation, die vom Signalfluss unabhängig ist.

Tipp

Die Anzahl der Operanden kann auf maximal 32 Eingänge bei den Operationen UND und ODER erweitert werden. Sie ergänzen oder löschen die Kennzeichen für Operanden mit den Tasten “+” und “-” auf Ihrer Tastatur.

Allgemeine Konventionen zum Programmieren einer S7-200

Definition von EN/ENO

EN (Enable IN = Freigabeeingang) ist ein boolescher Eingang der Boxen in KOP und FUP. An diesem Eingang muss Signalfluss vorhanden sein, damit die Box ausgeführt werden kann. AWL-Operationen haben keinen EN-Eingang. Der oberste Stackwert muss logisch 1 sein, damit die entsprechende AWL-Operation ausgeführt wird.

EN (Enable OUT = Freigabeausgang) ist ein boolescher Ausgang der Boxen in KOP und FUP. Liegt an einer Box am Eingang EN Signalfluss an und die Box wird fehlerfrei ausgeführt, dann leitet der Ausgang ENO den Signalfluss zum nächsten Element weiter. Tritt während der Ausführung der Box ein Fehler auf, dann wird der Signalfluss an der Box, die den Fehler verursacht hat, beendet.

In AWL gibt es keinen ENO-Ausgang, doch die AWL-Operationen, die den KOP- und FUP-Operationen mit ENO-Ausgängen entsprechen, setzen auch ein besonderes ENO-Bit. Auf dieses Bit kann mit der Operation UENO (UND ENO) zugegriffen werden und es kann verwendet werden, um die gleiche Wirkung wie mit dem ENO-Bit einer Box zu erzielen.

Tipp

Die Operanden EN/ENO und deren Datentypen werden nicht in der Tabelle der gültigen Operanden zu den einzelnen Operationen aufgeführt, weil diese Operanden für alle KOP- und FUP-Operationen identisch sind. Tabelle 5-1 führt die Operanden und Datentypen für KOP und FUP auf. Diese Operanden gelten für alle KOP- und FUP-Operationen in diesem Handbuch.

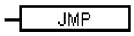
Tabelle 5-1 EN/ENO-Operanden und Datentypen in KOP und FUP

Programm-Editor	Eingänge / Ausgänge	Operanden	Datentypen
KOP	EN, ENO	Signalfluss	BOOL
FUP	EN, ENO	E, A, V, M, SM, S, T, Z, L	BOOL

Bedingte/absolute Eingänge

In KOP und FUP wird eine Box oder eine Spule, die vom Signalfluss abhängig ist, mit Verbindung zu einem anderen Element auf der linken Seite dargestellt. Eine Spule oder eine Box, die nicht vom Signalfluss abhängig ist, wird mit direkter Verbindung zur linken Stromschiene dargestellt. Tabelle 5-2 zeigt ein Beispiel für einen bedingten und einen absoluten Eingang.

Tabelle 5-2 Darstellung von bedingten und absoluten Eingängen

Signalfluss	KOP	FUP
Operation, die vom Signalfluss abhängig ist (bedingt)		
Operation, die nicht vom Signalfluss abhängig ist (absolut)		

Operationen ohne Ausgänge

Boxen, die keine Kaskadenschaltung zulassen, werden ohne boolesche Ausgänge gezeichnet. Dies sind die Operationen Unterprogramm aufrufen, Zu Sprungmarke springen und Unterprogramm bedingt beenden. Es gibt auch Spulen in KOP, die nur an die linke Stromschiene angeschlossen werden können. Hierzu gehören die Operationen Sprungmarke definieren, Ende Programmschleife mit NEXT, Ablaufsteuerungsrelais laden, Ablaufsteuerungsrelais bedingt beenden und Ende Ablaufsteuerungsrelais. Diese Operationen werden in FUP als Boxen dargestellt und durch nicht gekennzeichnete Eingänge und ohne Ausgänge hervorgehoben.

Vergleichsoperationen

Die Vergleichsoperationen werden unabhängig vom Zustand des Signalflusses ausgeführt. Ist der Signalfluss FALSE, ist der Ausgang FALSE. Ist der Signalfluss TRUE, wird der Ausgang entsprechend dem Ergebnis des Vergleichs gesetzt. Die SIMATIC-Vergleichsoperationen in FUP und die IEC-Vergleichsoperationen in FBS werden als Boxen dargestellt, obwohl die Operationen wie Kontakte ausgeführt werden.

Erstellen Ihres Steuerungsprogramms mit Assistenten

STEP 7-Micro/WIN verfügt über Assistenten, die bestimmte Aspekte der Programmierung automatisieren und dadurch vereinfachen. Im Kapitel 6 werden Operationen, die über einen Assistenten verfügen, mit dem folgenden Symbol gekennzeichnet:

Operations-
Assistent

Fehlerbehebung in der S7-200

Die S7-200 unterteilt aufgetretene Fehler in schwere und leichte Fehler. Sie können sich die von einem Fehler erzeugten Fehlercodes mit dem Menübefehl **Zielsystem > Informationen** anzeigen lassen.

Bild 5-6 zeigt das Dialogfeld ”Informationen”, in dem der Fehlercode und die Beschreibung des Fehlers angezeigt werden.

Das Feld ”Letzter schwerer” führt den vorherigen von der S7-200 generierten schweren Fehler auf. Dieser Wert wird nach dem Ausschalten gespeichert, wenn der RAM-Speicher gepuffert wird. Der Wert wird gelöscht, wenn für die S7-200 Urlöschen durchgeführt wird oder wenn der RAM-Speicher nach längerer Spannungsunterbrechung nicht mehr gepuffert wird.

Das Feld ”Gesamt” enthält die Gesamtzahl der Fehler, die seit dem letzten Urlöschen aller Speicherbereiche der S7-200 von der S7-200 generiert wurden. Dieser Wert wird nach dem Ausschalten gespeichert, wenn der RAM-Speicher gepuffert wird. Der Wert wird gelöscht, wenn für die S7-200 Urlöschen durchgeführt wird oder wenn der RAM-Speicher nach längerer Spannungsunterbrechung nicht mehr gepuffert wird.

Anhang C führt die Fehlercodes der S7-200 auf und Anhang D beschreibt die Sondermerker (SM) für die Fehlerüberwachung.

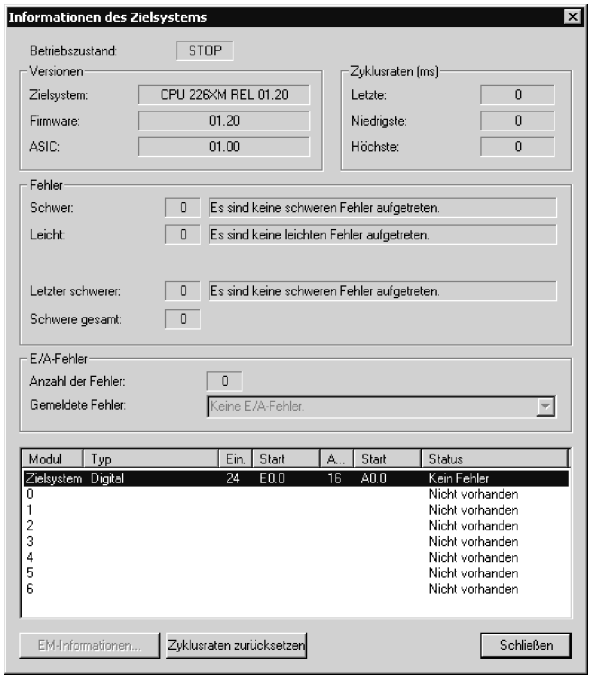


Bild 5-6 Dialogfeld ”Informationen”

Leichte Fehler

Bei leichten Fehlern handelt es sich um Fehler im Aufbau des Anwenderprogramms, um Fehler bei der Ausführung einer Operation im Anwenderprogramm und um Fehler bei den Erweiterungsmodulen. Mit STEP 7-Micro/WIN können Sie sich die Fehlercodes anzeigen lassen, die von leichten Fehlern erzeugt wurden. Es gibt drei wesentliche Gruppen leichter Fehler:

Fehler beim Übersetzen des Programms

Wenn Sie ein Programm laden, übersetzt die S7-200 das Programm. Erkennt die S7-200, dass das Programm eine Übersetzungsregel verletzt, dann bricht die S7-200 den Ladevorgang ab und erzeugt einen Fehlercode. (War bereits ein Programm in die S7-200 geladen, ist dieses Programm im EEPROM noch vorhanden, es geht nicht verloren.) Nachdem Sie Ihr Programm korrigiert haben, können Sie es erneut laden. Im Anhang C finden Sie eine Liste mit Verletzungen der Übersetzungsregeln.

E/A-Fehler

Beim Anlauf liest die S7-200 die E/A-Konfiguration der einzelnen Module. Im normalen Betrieb prüft die S7-200 regelmäßig den Zustand der einzelnen Module und vergleicht diesen Zustand mit der während des Anlaufs erhaltenen Konfiguration. Erkennt die S7-200 einen Unterschied, setzt die S7-200 das Konfigurationsfehlerbit im Fehlerregister des Moduls. Die S7-200 liest erst dann wieder Eingangsdaten aus dem Modul und schreibt erst dann wieder Ausgangsdaten in das Modul, wenn die Modulkonfiguration wieder der beim Anlauf gespeicherten Konfiguration entspricht.

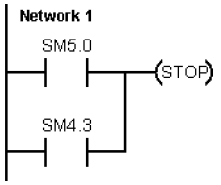
Die Statusinformationen eines Moduls werden in den Sondermerkern (SM) gespeichert. Ihr Programm kann diese Sondermerker überwachen und auswerten. Ausführliche Informationen zu Sondermerkern, die E/A-Fehler anzeigen, finden Sie in Anhang D. SM5.0 ist das globale E/A-Fehlerbit und bleibt solange gesetzt, wie der Fehlerzustand in einem Erweiterungsmodul vorhanden ist.

Fehler beim Ausführen des Programms

Ihr Programm kann während der Ausführung Fehlerbedingungen hervorrufen. Diese Fehler können durch fehlerhaften Einsatz einer Operation oder durch die Bearbeitung ungültiger Daten durch eine Operation ausgelöst werden. Ein Pointer auf eine indirekte Adresse 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 ist ein Beispiel für einen Programmierfehler zur Laufzeit. SM4.3 wird bei Auftreten des Programmierfehlers zur Laufzeit gesetzt und bleibt gesetzt, solange sich die S7-200 im Betriebszustand RUN befindet. (Im Anhang C finden Sie eine Liste mit Programmierfehlern zur Laufzeit.) Informationen zu Programmausführungsfehlern werden in den Sondermerkern (SM) gespeichert. Ihr Programm kann diese Sondermerker überwachen und auswerten. Ausführliche Informationen zu Sondermerkern, die Programmausführungsfehler anzeigen, finden Sie in Anhang D.

Die S7-200 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 der S7-200 in den Betriebszustand STOP erzwungen wird. Das folgende Beispielprogramm zeigt ein Netzwerk eines Programms, das zwei der globalen Bits für leichte Fehler überwacht und die S7-200 in den Betriebszustand STOP versetzt, wenn eines dieser Bits eingeschaltet wird.

Beispielprogramm: Logik zum Erkennen eines leichten Fehlers




```
Netzwerk 1      //Wenn ein E/A-Fehler oder ein Laufzeitfehler auftritt, in STOP gehen
LD      SM5.0
O       SM4.3
STOP
```

Schwere Fehler

Tritt ein schwerer Fehler auf, beendet die S7-200 die Bearbeitung des Programms. Je nach dem Schweregrad des Fehlers kann die S7-200 einige oder auch gar keine Funktionen mehr ausführen. Ziel der Behebung von schweren Fehlern ist es, die S7-200 in einen sicheren Zustand zu bringen, so dass Informationen zu der Fehlerbedingung in der S7-200 abgefragt werden können. Erkennt die S7-200 einen schweren Fehler, geht sie in die Betriebsart STOP, schaltet die LED-Anzeige für Systemfehler (SF) und die LED-Anzeige für STOP ein, übersteuert die Tabelle der Ausgänge und schaltet die Ausgänge aus. Die S7-200 bleibt in diesem Zustand, bis der schwere Fehler behoben ist.

Nachdem Sie die Änderungen zur Behebung des schweren Fehlers vorgenommen haben, starten Sie die S7-200 auf eine der folgenden Arten neu:

- Schalten Sie die Spannungsversorgung aus und wieder ein.
- Bringen Sie den Betriebsartenschalter von TERM bzw. RUN in STOP.
- Wählen Sie in STEP 7-Micro/WIN den Menübefehl **Zielsystem > Rücksetzen beim Anlauf**, um die S7-200 neu zu starten. Hierdurch wird die S7-200 zurückgesetzt und alle schweren Fehler werden gelöscht.

Durch das Neustarten der S7-200 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 S7-200 auf und deutet damit an, dass noch immer ein Fehler vorhanden ist. Andernfalls beginnt die S7-200 ihren normalen Betrieb.

Durch einige Fehlerzustände wird die S7-200 kommunikationsunfähig. In diesen Fällen können Sie sich den Fehlercode der S7-200 nicht anzeigen lassen. Diese Arten von Fehlern deuten meist auf Hardware-Fehler hin, die nur durch Reparatur der S7-200 behoben werden können. Solche Fehlerbedingungen können nicht durch Änderungen im Programm oder Urlöschen des Speichers der S7-200 beseitigt werden.

Zuordnen von Adressen und Anfangswerten im Datenbaustein-Editor

Daten-baustein

Im Datenbaustein-Editor können Sie Zuordnungen von Anfangswerten im Variablenspeicher vornehmen. Sie können Bytes, Wörter und Doppelwörter im Variablenspeicher zuordnen. Kommentare sind optional.

Beim Datenbaustein-Editor handelt es sich um einen Texteditor, mit dem das Format frei wählbar ist, d.h. es sind für bestimmte Informationen keine vorgegebenen Felder definiert. Nachdem Sie eine Zeile eingetragen haben und die Eingabetaste drücken, formatiert der Datenbaustein-Editor die Zeile (Adressen, Daten, Kommentare werden in Spalten ausgerichtet; Adressen im Variablenspeicher werden in Großbuchstaben dargestellt) und zeigt sie erneut an. Der Datenbaustein-Editor ordnet eine entsprechende Menge Speicherplatz im Variablenspeicher gemäß Ihren vorherigen Adresszuordnungen und der Größe (Byte, Wort oder Doppelwort) der Datenwerte zu.

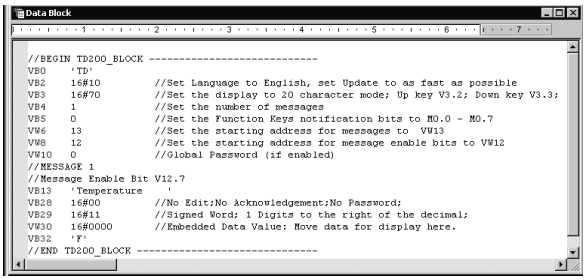


Bild 5-7 Datenbaustein-Editor

Die erste Zeile im Datenbaustein muss eine explizite Adresszuordnung enthalten. Die folgenden Zeilen können explizite oder implizite Adresszuordnungen enthalten. Eine implizite Adresszuordnung wird vom Editor erstellt, wenn Sie mehrere Datenwerte im Anschluss an eine Adresszuordnung eintragen oder wenn Sie eine Zeile eingeben, die nur Datenwerte enthält.

Sie können im Datenbaustein-Editor Groß- und Kleinbuchstaben eingeben und Kommas, Tabulatoren und Leerzeichen als Trennungszeichen zwischen Adressen und Datenwerten verwenden.

Verwenden der Symboltabelle für die symbolische Adressierung von Variablen

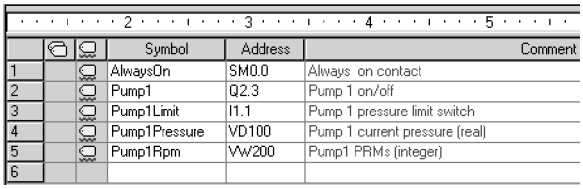
Symbol-tabelle

In der Symboltabelle können Sie Symbole definieren und bearbeiten, auf die über den symbolischen Namen im gesamten Programm zugegriffen werden kann. Sie können mehrere Symboltabellen anlegen. Die Symboltabelle enthält außerdem ein Register für systemdefinierte Symbole, die Sie in Ihrem Programm verwenden können. Die Symboltabelle wird auch globale Variablentabelle genannt.

Sie können die Operanden in den Anweisungen in Ihrem Programm absolut oder symbolisch angeben. Ein absoluter Verweis gibt den Speicherbereich sowie das Bit oder Byte der Adresse an. Ein symbolischer Verweis gibt die Adresse mittels einer Kombination aus alphanumerischen Zeichen an.

Für SIMATIC-Programme ordnen Sie globale Symbole in der Symboltabelle zu. Für IEC-Programme ordnen Sie globale Symbole in der globalen Variablentabelle zu.

Zum Zuordnen eines symbolischen Namens zu einer Adresse gehen Sie folgendermaßen vor:



	Symbol	Address	Comment
1	AlwaysOn	SM0.0	Always on contact
2	Pump1	Q2.3	Pump 1 on/off
3	Pump1Limit	I1.1	Pump 1 pressure limit switch
4	Pump1Pressure	VD100	Pump 1 current pressure (real)
5	Pump1Rpm	VW200	Pump1 PRMs (integer)
6			

Bild 5-8 Symboltabelle

1. Klicken Sie in der Navigationsleiste auf die Schaltfläche der Symboltabelle, um die Symboltabelle aufzurufen.
2. Geben Sie den symbolischen Namen (z.B. Eingang1) in die Spalte ”Symbolischer Name” ein. Die maximale Länge für Symbole beträgt 23 Zeichen.
3. Geben Sie die Adresse (z.B. E0.0) in die Spalte ”Adresse” ein.
4. Wenn Sie in IEC in der globalen Variablentabelle arbeiten, tragen Sie einen Wert in die Spalte ”Datentyp” ein oder wählen einen Wert im aufklappbaren Listenfeld aus.

Sie können mehrere Symboltabellen anlegen. Sie können jedoch die gleiche Zeichenkette nicht mehr als einmal als globales Symbol zuordnen, weder in einer einzigen Tabelle noch in verschiedenen Tabellen.

Lokale Variablen

In der lokalen Variablentabelle im Programm-Editor können Sie Variablen zuordnen, die nur in einem einzelnen Unterprogramm oder in einem einzelnen Interruptprogramm vorkommen (siehe Bild 5-9).

Lokale Variablen können als Parameter verwendet werden, die an ein Unterprogramm übergeben werden. Lokale Variablen erhöhen die Portabilität bzw. die Wiederverwendbarkeit eines Unterprogramms.

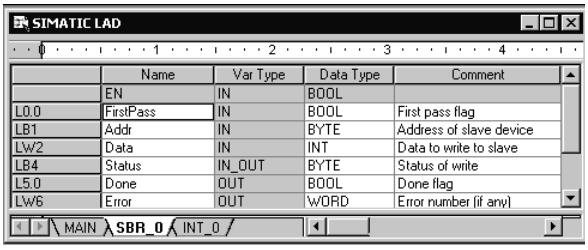


Bild 5-9 Lokale Variablentabelle

Beobachten Ihres Programms mit der Statustabelle

Status-tabelle

Mit einer Statustabelle können Sie die Werte der Prozessvariablen beobachten und ändern, während die S7-200 das Steuerungsprogramm ausführt. Sie können den Zustand von Eingängen, Ausgängen und Variablen im Programm verfolgen, indem Sie die aktuellen Werte anzeigen. In der Statustabelle können Sie außerdem Werte von Prozessvariablen forcen oder ändern.

Sie können mehrere Statustabellen anlegen, damit Sie die Elemente aus verschiedenen Teilen Ihres Programms beobachten können.

Zum Aufrufen der Statustabelle wählen Sie den Menübefehl **Ansicht > Komponente > Statustabelle** oder klicken in der Navigationsleiste auf das Symbol der Statustabelle.

Wenn Sie eine Statustabelle anlegen, geben Sie Adressen von Prozessvariablen ein, die Sie beobachten möchten. Den Status von Konstanten, Akkumulatoren und lokalen Variablen können Sie nicht anzeigen. Die Werte von Zeiten und Zählern können Sie sich als Bit oder als Wort anzeigen lassen. Wenn Sie sich den Wert als Bit anzeigen lassen, wird der Status des Zeit- oder Zählerbits angezeigt. Wenn Sie sich den Wert als Wort anzeigen lassen, wird der Wert der Zeit bzw. des Zählers angezeigt.

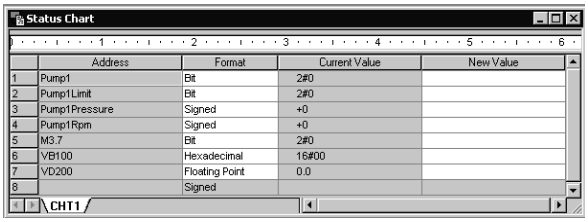


Bild 5-10 Statustabelle

So erstellen Sie eine Statustabelle und beobachten Variablen:

1. Tragen Sie in das Adressfeld die Adressen der gewünschten Werte ein.
2. Wählen Sie in der Spalte ”Format” den Datentyp aus.
3. Zum Anzeigen des Status der Prozessvariablen Ihrer S7-200 wählen Sie den Menübefehl **Testen > Tabellenstatus**.
4. Wenn Sie die Werte kontinuierlich abfragen möchten oder wenn Sie den Status einmal auslesen möchten, klicken Sie auf die Symbole in der Funktionsleiste. In der Statustabelle können Sie die Werte der unterschiedlichen Prozessvariablen auch forcen oder ändern.

Mit dem Menübefehl **Bearbeiten > Einfügen > Reihe** können Sie weitere Reihen in die Statustabelle einfügen.

Tipp

Sie können mehrere Statustabellen anlegen, um die Variablen in logische Gruppen aufzuteilen. So können Sie sich jede Gruppe in einer eigenen, kurzen Statustabelle ansehen.

Erstellen einer Operationsbibliothek

In STEP 7-Micro/WIN können Sie eine anwenderspezifische Bibliothek mit Operationen anlegen, oder Sie können eine von einer anderen Person erstellte Bibliothek nutzen. (Siehe Bild 5-11).

Zum Erstellen einer Bibliothek mit Operationen legen Sie in STEP 7-Micro/WIN Unterprogramme und Interruptprogramme an und gruppieren diese. Sie können den Code in diesen Unterprogrammen und Interruptprogrammen ausblenden, um versehentliche Änderungen zu verhindern und um die Technologie bzw. das Know-how des Autors zu schützen.

Zum Erstellen einer Operationsbibliothek gehen Sie folgendermaßen vor:

1. Schreiben Sie das Programm als normales STEP 7-Micro/WIN Projekt und legen Sie die Funktionen, die in die Bibliothek aufgenommen werden sollen, in Unterprogrammen und Interruptprogrammen ab.
2. Stellen Sie sicher, dass Sie allen Adressen im Variablenspeicher in den Unterprogrammen und Interruptprogrammen einen symbolischen Namen zugeordnet haben. Verwenden Sie aufeinander folgende Adressen im Variablenspeicher, um den von der Bibliothek verwendeten Variablenspeicher möglichst gering zu halten.
3. Benennen Sie die Unterprogramme und Interruptprogramme so um, wie sie in der Operationsbibliothek heißen sollen.
4. Zum Übersetzen der neuen Operationsbibliothek wählen Sie den Menübefehl **Datei > Bibliothek erstellen**.

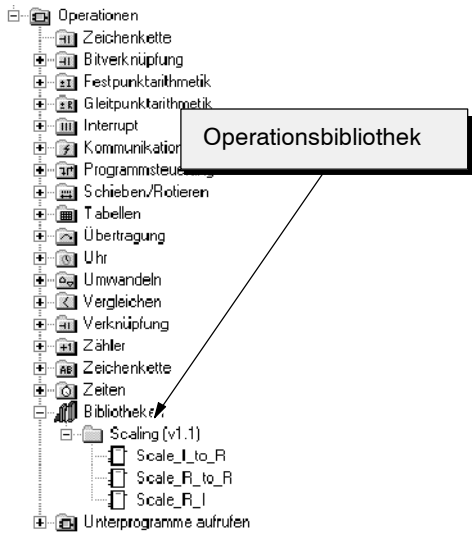


Bild 5-11 Operationsbaum mit Bibliotheken

Weitere Informationen zum Erstellen von Bibliotheken finden Sie in der Online-Hilfe von STEP 7-Micro/WIN.

Gehen Sie folgendermaßen vor, um auf eine Operation in einer Operationsbibliothek zuzugreifen:

1. Nehmen Sie das Verzeichnis ”Bibliotheken” mit dem Menübefehl **Datei > Bibliotheken ergänzen** in den Operationsbaum auf.
2. Wählen Sie die gewünschte Operation und fügen Sie sie in Ihr Programm ein (wie Sie alle Operationen einfügen).

   Benötigt das Programm aus der Bibliothek Variablenspeicher, werden Sie von STEP 7-Micro/WIN nach dem Übersetzen des Projekts aufgefordert, einen Speicherbereich zuzuweisen. Speicherbereiche weisen Sie im Dialogfeld ”Speicher für Bibliothek” zu.

Funktionen zum Testen Ihres Programms

STEP 7-Micro/WIN bietet die folgenden Funktionen, mit denen Sie Ihr Programm testen können:

- ❏ Sie können in Ihrem Programm Lesezeichen setzen, um bestimmte Zeilen in einem Programm leichter wiederfinden zu können.
- ❏ In der Tabelle der Querverweise können Sie die in Ihrem Programm verwendeten Referenzen prüfen.
- ❏ Wenn Sie das Programm im Betriebszustand RUN bearbeiten, können Sie kleinere Änderungen am Anwenderprogramm vornehmen, ohne den vom Programm gesteuerten Prozess zu stören. Sie können auch den Programmbaustein laden, wenn Sie das Programm im Betriebszustand RUN ändern.

Weitere Informationen zum Testen Ihres Programms finden Sie in Kapitel 8.

S7-200 Befehlssatz

6

In diesem Kapitel werden der SIMATIC-Befehlssatz und der IEC 1131-Befehlssatz für die S7-200 Micro-SPS beschrieben.

In diesem Kapitel

Konventionen für die Beschreibung der Operationen 63
Speicherbereiche und Funktionen der S7-200 64
Bitverknüpfungsoperationen 66
Kontakte 66
Spulen 69
Stackoperationen 71
Operationen Bistabiler Funktionsbaustein: Vorrangig Setzen und Vorrangig Rücksetzen 73
Uhroperationen 74
Kommunikationsoperationen 75
Operationen Aus Netz lesen und In Netz schreiben 75
Operationen Meldung aus Zwischenspeicher übertragen und Meldung in Zwischenspeicher empfangen (frei programmierbare Kommunikation) 80
Operationen Schnittstellenadresse holen und Schnittstellenadresse einstellen 89
Vergleichsoperationen 90
Vergleich von numerischen Werten 90
Zeichenkettenvergleich 92
Umwandlungsoperationen 93
Genormte Umwandlungsoperationen 93
ASCII-Umwandlungsoperationen 97
Zeichenketten-Umwandlungsoperationen 101
Operationen Hexadezimalzahl in Bit wandeln und Bit in Hexadezimalzahl wandeln 106
Zähloperationen 107
SIMATIC: Zähler 107
IEC: Zähler 110
Schnelle Zähler 112
Operation Impulsausgabe 126
Arithmetische Operationen 141
Operationen Addieren, Subtrahieren, Multiplizieren und Dividieren 141
Ganze Zahlen (16 Bit) in ganze Zahl (32 Bit) multiplizieren und Ganze Zahlen (16 Bit) dividieren mit Divisionsrest 143
Numerische Funktionen 144
Operationen Inkrementieren und Dekrementieren 145
Operation PID-Regler (Proportional/Integral/Differential) 146
Interruptoperationen 156
Verknüpfungsoperationen 163
Invertieroperationen 163
Operationen Bitwert durch UND, ODER oder EXKLUSIV ODER verknüpfen 164
Übertragungsoperationen 166
Byte, Wort, Doppelwort oder Realzahl übertragen 166
Byte direkt lesen bzw. schreiben und übertragen 167
Operationen Wertebereich übertragen 168

Operationen für die Programmsteuerung 169
Bearbeitung bedingt beenden 169
STOP 169
Überwachungszeit rücksetzen 169
Operationen Programmschleife mit FOR und Ende Programmschleife mit NEXT 171
Sprungoperationen 173
Operationen für das Ablaufsteuerungsrelais (SCR) 174
Schiebe- und Rotieroperationen 180
Operationen Rechts schieben und Links schieben 180
Operationen Rechts rotieren und Links rotieren 180
Operation Wert in Schieberegister schieben 182
Operation Bytes im Wort tauschen 184
Zeichenkettenoperationen 185
Tabellenoperationen 190
Wert in Tabelle eintragen 190
Ersten Wert aus Tabelle löschen und Letzten Wert aus Tabelle löschen 191
Speicher mit Bitmuster belegen 193
Wert in Tabelle suchen 194
Zeitoperationen 197
SIMATIC: Zeitoperationen 197
IEC: Zeiten 202
Unterprogrammoperationen 204

Konventionen für die Beschreibung der Operationen

Bild 6-1 zeigt eine typische Beschreibung einer Operation und verweist auf die verschiedenen Bereiche, in denen die Operation und ihre Funktionsweise beschrieben wird. Die Abbildung der Operation zeigt das Format in KOP, FUP und AWL. Die Operandentabelle führt die Operanden für die Operation auf und zeigt die gültigen Datentypen, Speicherbereiche und Größe der Operanden.

Die Operanden EN/ENO und deren Datentypen werden in der Operandentabelle der Operation nicht aufgeführt, weil diese Operanden für alle KOP- und FUP-Operationen identisch sind.

- *Für KOP:* EN und ENO sind Signalfluss und vom Datentyp BOOL.
- *Für FUP:* EN und ENO sind E, A, V, M, SM, S, T, Z, L oder Signalfluss und vom Datentyp BOOL.

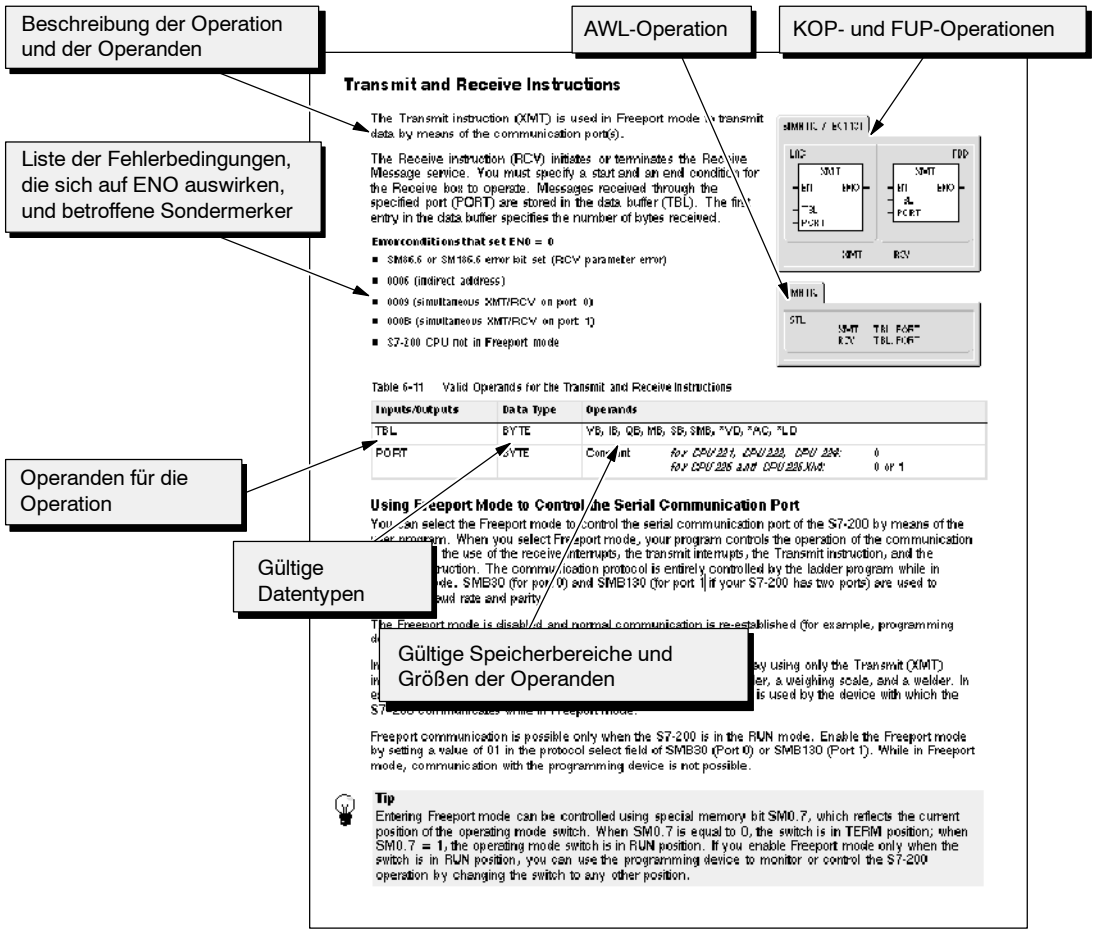


Bild 6-1 Beschreibung der Operation

Speicherbereiche und Funktionen der S7-200

Tabelle 6-1 Speicherbereiche und Funktionen der S7-200 CPUs

Beschreibung	CPU 221	CPU 222	CPU 224	CPU 226	CPU 226XM
Größe Anwenderprogramm	4096 Bytes	4096 Bytes	8192 Bytes	8192 Bytes	16384 Bytes
Größe Anwenderdaten	2048 Bytes	2048 Bytes	5120 Bytes	5120 Bytes	10240 Bytes
Prozessabbild der Eingänge	E0.0 bis E15.7	E0.0 bis E15.7	E0.0 bis E15.7	E0.0 bis E15.7	E0.0 bis E15.7
Prozessabbild der Ausgänge	A0.0 bis A15.7	A0.0 bis A15.7	A0.0 bis A15.7	A0.0 bis A15.7	A0.0 bis A15.7
Analogeingänge (nur Lesen)	--	AEW0 bis AEW30	AEW0 bis AEW62	AEW0 bis AEW62	AEW0 bis AEW62
Analogausgänge (nur Schreiben)	--	AAW0 bis AAW30	AAW0 bis AAW62	AAW0 bis AAW62	AAW0 bis AAW62
Variablenspeicher (V)	VB0 bis VB2047	VB0 bis VB2047	VB0 bis VB5119	VB0 bis VB5119	VB0 bis VB10239
Lokaldatenspeicher (L)[1]	LB0 bis LB63	LB0 bis LB63	LB0 bis LB63	LB0 bis LB63	LB0 bis LB63
Merker (M)	M0.0 bis M31.7	M0.0 bis M31.7	M0.0 bis M31.7	M0.0 bis M31.7	M0.0 bis M31.7
Sondermerker (SM)	SM0.0 bis SM179.7	SM0.0 bis SM299.7	SM0.0 bis SM549.7	SM0.0 bis SM549.7	SM0.0 bis SM549.7
Schreibgeschützt	SM0.0 bis SM29.7	SM0.0 bis SM29.7	SM0.0 bis SM29.7	SM0.0 bis SM29.7	SM0.0 bis SM29.7
Zeiten	256 (T0 bis T255)	256 (T0 bis T255)	256 (T0 bis T255)	256 (T0 bis T255)	256 (T0 bis T255)
Speichernde Einschaltverzögerung 1 ms	T0, T64	T0, T64	T0, T64	T0, T64	T0, T64
10 ms	T1 bis T4 und T65 bis T68	T1 bis T4 und T65 bis T68	T1 bis T4 und T65 bis T68	T1 bis T4 und T65 bis T68	T1 bis T4 und T65 bis T68
100 ms	T5 bis T31 und T69 bis T95	T5 bis T31 und T69 bis T95	T5 bis T31 und T69 bis T95	T5 bis T31 und T69 bis T95	T5 bis T31 und T69 bis T95
Ein-/Ausschaltverzög. 1 ms	T32, T96	T32, T96	T32, T96	T32, T96	T32, T96
10 ms	T33 bis T36 und T97 bis T100	T33 bis T36 und T97 bis T100	T33 bis T36 und T97 bis T100	T33 bis T36 und T97 bis T100	T33 bis T36 und T97 bis T100
100 ms	T37 bis T63 und T101 bis T255	T37 bis T63 und T101 bis T255	T37 bis T63 und T101 bis T255	T37 bis T63 und T101 bis T255	T37 bis T63 und T101 bis T255
Zähler	Z0 bis Z255	Z0 bis Z255	Z0 bis Z255	Z0 bis Z255	Z0 bis Z255
Schnelle Zähler	HC0, HC3, HC4 und HC5	HC0, HC3, HC4 und HC5	HC0 bis HC5	HC0 bis HC5	HC0 bis HC5
Ablaufsteuerungsrelais (S)	S0.0 bis S31.7	S0.0 bis S31.7	S0.0 bis S31.7	S0.0 bis S31.7	S0.0 bis S31.7
Akkumulatoren	AC0 bis AC3	AC0 bis AC3	AC0 bis AC3	AC0 bis AC3	AC0 bis AC3
Sprünge/Sprungmarken	0 bis 255	0 bis 255	0 bis 255	0 bis 255	0 bis 255
Aufrufe/Unterprogramme	0 bis 63	0 bis 63	0 bis 63	0 bis 63	0 bis 127
Interruptprogramme	0 bis 127	0 bis 127	0 bis 127	0 bis 127	0 bis 127
Steigende/fallende Flanken	256	256	256	256	256
PID-Regler	0 bis 7	0 bis 7	0 bis 7	0 bis 7	0 bis 7
Anschlüsse	Schnittstelle 0	Schnittstelle 0	Schnittstelle 0	Schnittstelle 0, Schnittstelle 1	Schnittstelle 0, Schnittstelle 1

[1] LB60 bis LB63 sind für STEP 7-Micro/WIN ab Version 3.0 reserviert.

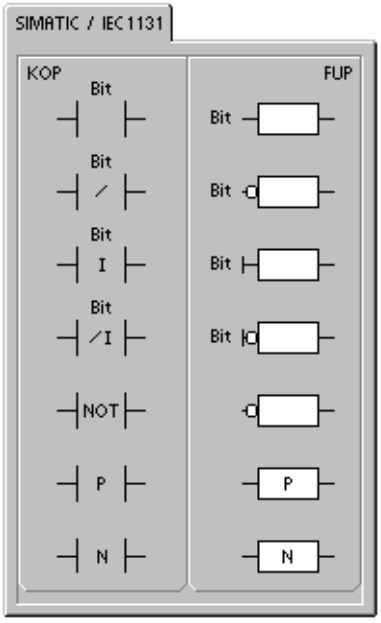
Tabelle 6-2 Operandenbereiche der S7-200 CPUs

Zugriffsformat		CPU 221	CPU 222	CPU 224, CPU 226	CPU 226XM
Bit (Byte.Bit)	E	0.0 bis 15.7	0.0 bis 15.7	0.0 bis 15.7	0.0 bis 15.7
	A	0.0 bis 15.7	0.0 bis 15.7	0.0 bis 15.7	0.0 bis 15.7
	V	0.0 bis 2047.7	0.0 bis 2047.7	0.0 bis 5119.7	0.0 bis 10239.7
	M	0.0 bis 31.7	0.0 bis 31.7	0.0 bis 31.7	0.0 bis 31.7
	SM	0.0 bis 179.7	0.0 bis 299.7	0.0 bis 549.7	0.0 bis 549.7
	S	0.0 bis 31.7	0.0 bis 31.7	0.0 bis 31.7	0.0 bis 31.7
	T	0 bis 255	0 bis 255	0 bis 255	0 bis 255
	Z	0 bis 255	0 bis 255	0 bis 255	0 bis 255
	L	0.0 bis 59.7	0.0 bis 59.7	0.0 bis 59.7	0.0 bis 59.7
Byte	EB	0 bis 15	0 bis 15	0 bis 15	0 bis 15
	AB	0 bis 15	0 bis 15	0 bis 15	0 bis 15
	VB	0 bis 2047	0 bis 2047	0 bis 5119	0 bis 10239
	MB	0 bis 31	0 bis 31	0 bis 31	0 bis 31
	SMB	0 bis 179	0 bis 299	0 bis 549	0 bis 549
	SB	0 bis 31	0 bis 31	0 bis 31	0 bis 31
	L	0 bis 63	0 bis 63	0 bis 63	0 bis 255
	AC	0 bis 3	0 bis 3	0 bis 3	0 bis 255
Wort	EW	0 bis 14	0 bis 14	0 bis 14	0 bis 14
	AW	0 bis 14	0 bis 14	0 bis 14	0 bis 14
	VW	0 bis 2046	0 bis 2046	0 bis 5118	0 bis 10238
	MW	0 bis 30	0 bis 30	0 bis 30	0 bis 30
	SMW	0 bis 178	0 bis 298	0 bis 548	0 bis 548
	SW	0 bis 30	0 bis 30	0 bis 30	0 bis 30
	T	0 bis 255	0 bis 255	0 bis 255	0 bis 255
	Z	0 bis 255	0 bis 255	0 bis 255	0 bis 255
	LW	0 bis 58	0 bis 58	0 bis 58	0 bis 58
	AC	0 bis 3	0 bis 3	0 bis 3	0 bis 3
	AEW	*keine*	0 bis 30	0 bis 62	0 bis 62
	AAW	*keine*	0 bis 30	0 bis 62	0 bis 62
Doppelwort	ED	0 bis 12	0 bis 12	0 bis 12	0 bis 12
	AD	0 bis 12	0 bis 12	0 bis 12	0 bis 12
	VD	0 bis 2044	0 bis 2044	0 bis 5116	0 bis 10236
	MD	0 bis 28	0 bis 28	0 bis 28	0 bis 28
	SMD	0 bis 176	0 bis 296	0 bis 546	0 bis 546
	SD	0 bis 28	0 bis 28	0 bis 28	0 bis 28
	LD	0 bis 56	0 bis 56	0 bis 56	0 bis 56
	AC	0 bis 3	0 bis 3	0 bis 3	0 bis 3
	HC	0, 3, 4, 5	0, 3, 4, 5	0 bis 5	0 bis 5

Bitverknüpfungsoperationen

Kontakte

Standardkontakte

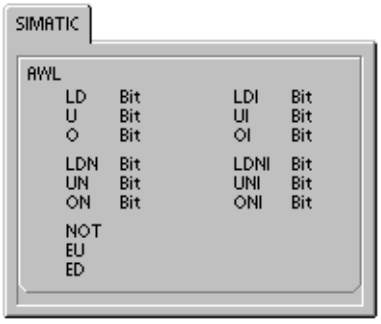


Die Operationen Schließerkontakt (LD, U und O) und Öffnerkontakt (LDN, UN, ON) erhalten den referenzierten Wert aus dem Speicher oder aus dem Prozessabbild. Die Standardkontakte erhalten den referenzierten Wert aus dem Speicher (oder aus dem Prozessabbild, wenn es sich um den Datentyp E oder A handelt).

Der Schließerkontakt ist geschlossen (ein), wenn das Bit gleich 1 ist, und der Öffnerkontakt ist geschlossen (ein), wenn das Bit gleich 0 ist. In FUP können maximal 32 Eingänge an die Boxen UND oder ODER angeschlossen werden. In AWL laden die Schließerkontakte den Bitwert des Adressbits als obersten Stackwert bzw. sie verknüpfen den Bitwert mit dem obersten Stackwert durch UND oder ODER. Die Öffnerkontakte laden den negierten Bitwert als obersten Stackwert bzw. sie verknüpfen den negierten Bitwert mit dem obersten Stackwert durch UND oder ODER.

Direkte Kontakte

Ein direkter Kontakt richtet sich zur Aktualisierung nicht nach dem Zyklus der S7-200. Er wird sofort aktualisiert. Die Operationen Direkter Schließerkontakt (LDI, UI und OI) und Direkter Öffnerkontakt (LDNI, UNI und ONI) erhalten den physikalischen Eingangswert, wenn die Operation ausgeführt wird, doch das Prozessabbild wird nicht aktualisiert.



Der direkte Schließerkontakt ist geschlossen (ein), wenn der physikalische Eingang (Bit) 1 ist, und der direkte Öffnerkontakt ist geschlossen (ein), wenn der physikalische Eingang (Bit) 0 ist. Die Schließerkontakte laden den Wert des physikalischen Eingangs direkt als obersten Stackwert bzw. sie verknüpfen den Wert des Eingangs mit dem obersten Stackwert durch UND oder ODER. Die Öffnerkontakte laden den negierten Wert des physikalischen Eingangs direkt als obersten Stackwert bzw. sie verknüpfen den negierten Wert des Eingangs mit dem obersten Stackwert durch UND oder ODER.

Operation NOT

Die Operation NOT ändert den Zustand des Signalflusseingangs (d.h. sie ändert den obersten Stackwert von 0 nach 1 oder von 1 nach 0).

Operationen Steigende Flanke und Fallende Flanke

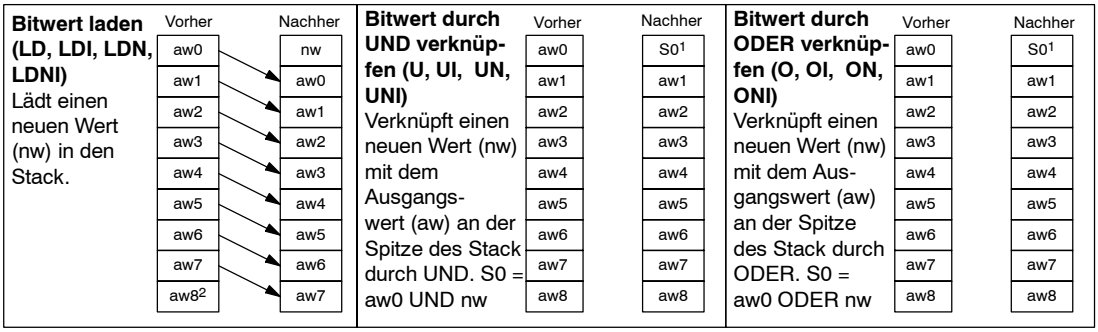
Die Operation Steigende Flanke (EU) lässt den Signalfluss bei jeder steigenden Flanke einen Zyklus lang zu. Die Operation Fallende Flanke (ED) lässt den Signalfluss bei jeder fallenden Flanke einen Zyklus lang zu. Wird bei der Operation Steigende Flanke im obersten Stackwert eine steigende Flanke (Wechsel von 0 nach 1) erkannt, dann wird der oberste Stackwert auf 1 gesetzt. Wird keine steigende Flanke erkannt, wird der Wert auf 0 gesetzt. Wird bei der Operation Fallende Flanke im obersten Stackwert eine fallende Flanke (Wechsel von 1 nach 0) erkannt, dann wird der oberste Stackwert auf 1 gesetzt. Wird keine fallende Flanke erkannt, wird der Wert auf 0 gesetzt.

Wenn Sie Ihr Programm im Betriebszustand RUN bearbeiten, müssen Sie einen Parameter für die Operationen Steigende Flanke und Fallende Flanke eingeben. Ausführliche Informationen zur Bearbeitung im Betriebszustand RUN finden Sie im Kapitel 5.

Tabelle 6-3 Gültige Operanden für die Bitverknüpfungseingangsoperationen

Eingänge/Ausgänge	Datentyp	Operanden
Bit	BOOL	E, A, V, M, SM, S, T, Z, L, Signalfluss
Bit (direkt)	BOOL	E

Die S7-200 löst die Steuerungslogik mit Hilfe eines logischen Stack (siehe Bild 6-2). In diesem Beispiel kennzeichnen ”aw0” bis ”aw7” die Ausgangswerte des logischen Stack, ”nw” kennzeichnet einen neuen Wert, der von der Operation bereitgestellt wird, und S0 kennzeichnet den errechneten Wert, der in dem logischen Stack gespeichert wird.



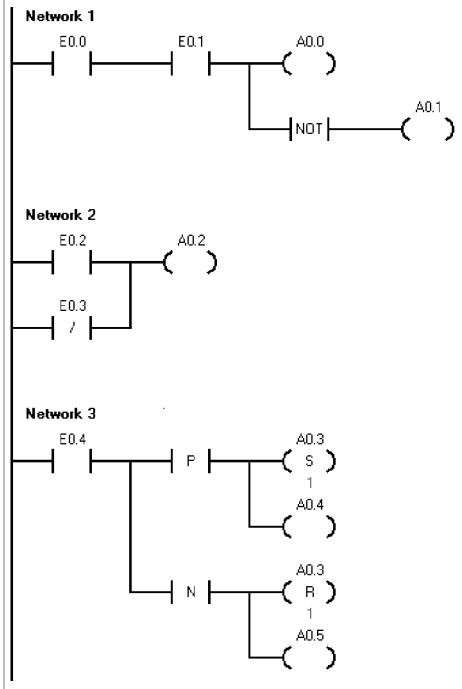
1 S0 ist der berechnete Wert, der im logischen Stack gespeichert wird.
2 Nach der Ausführung einer Operation Bitwert laden geht der Wert aw8 verloren.

Bild 6-2 Funktionsweise von Kontakten

Tipp

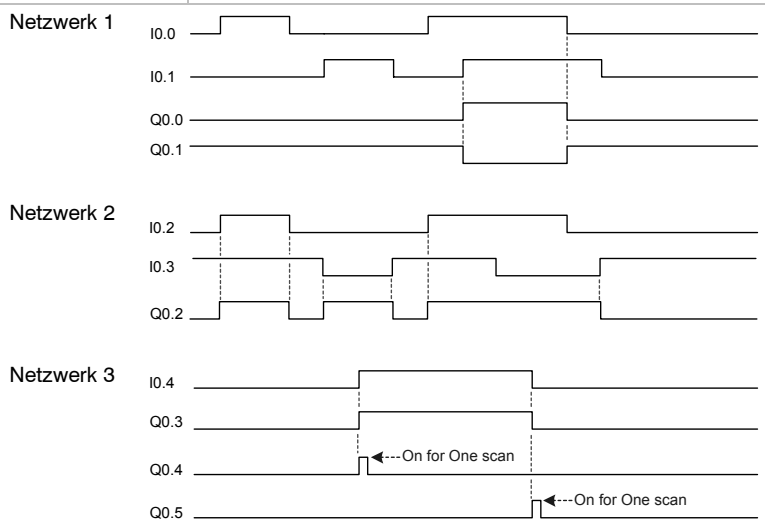
Weil die Operationen Steigende Flanke und Fallende Flanke einen Wechsel von 1 nach 0 oder von 0 nach 1 benötigen, können Sie im ersten Zyklus keine steigenden Flanken oder fallenden Flanken erkennen. Im ersten Zyklus setzt die S7-200 den Zustand des von diesen Operationen angegebenen Bit. In nachfolgenden Zyklen können diese Operationen dann Flanken für das angegebene Bit erkennen.

Beispiel: Kontakte




```
Netzwerk 1  //Schließerkontakte E0.0 UND E0.1 müssen geschlossen (ein) sein,
            //um A0.0 zu aktivieren. Die Operation NOT dient zum Invertieren.
            //Im Betriebszustand RUN haben A0.0 und A0.1 gegensätzliche
            // Signalzustände.
LD    E0.0
U     E0.1
=     A0.0
NOT
=     A0.1
Netzwerk 2  //Schließerkontakt E0.2 muss EP sein oder Öffnerkontakt E0.3 muss

            //AUS sein, damit A0.2 aktiviert wird.
            //Ein oder mehrere parallele KOP-Zweige (ODER-Eingänge)
            //müssen wahr sein, damit der Ausgang aktiviert wird.
LD    E0.2
ON    E0.3
=     A0.2
Netzwerk 3  //Eine steigende Flanke am Kontakt P oder
            //eine fallende Flanke am Kontakt N
            //gibt einen Impuls mit einer Dauer von 1 Zyklus aus. In RUN sind
            //die
            //Signalzustandsänderungen an A0.4 und A0.5 zu schnell,
            //als dass sie im Programm in der Statusansicht sichtbar wären.
            //Die Ausgänge Setzen und Rücksetzen speichern Impuls in A0.3
            //und
            //machen die Zustandsänderung in der Statusansicht des Pro
            //gramms
            // sichtbar.
LD    E0.4
LPS
EU
S     A0.3, 1
=     A0.4
LPP
ED
R     A0.3, 1
=     A0.5
```

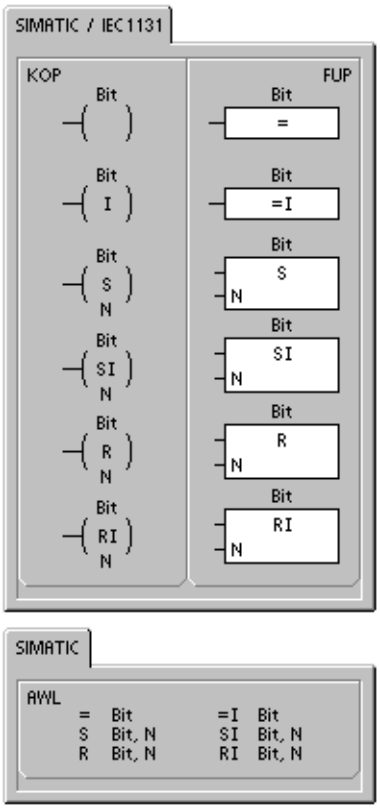
Impulsdiagramm



Spulen

Zuweisen

Die Operation Zuweisen (=) schreibt den neuen Wert für das Ausgangsbit in das Prozessabbild. Wird die Operation Zuweisen ausgeführt, dann schaltet die S7-200 das Ausgangsbit im Prozessabbild ein oder aus. In KOP und FUP wird das angegebene Bit entsprechend dem Zustand des Signalflusses gesetzt. In AWL wird der oberste Stackwert in das angegebene Bit kopiert.



Bitwert direkt zuweisen

Die Operation Bitwert direkt zuweisen (=I) schreibt den neuen Wert bei Ausführung der Operation sowohl in den physikalischen Ausgang als auch in das Prozessabbild.

Wird die Operation Bitwert direkt zuweisen ausgeführt, dann wird der physikalische Ausgang (Bit) sofort entsprechend dem Zustand des Signalflusses gesetzt. In AWL kopiert die Operation Bitwert direkt zuweisen den obersten Stackwert direkt in den angegebenen physikalischen Ausgang (AWL). 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 Eingang bzw. Ausgang nur in das Prozessabbild geschrieben wird.

Setzen und Rücksetzen

Die Operationen Setzen (S) und Rücksetzen (R) setzen (schalten ein) oder rücksetzen (schalten aus) die angegebene Anzahl Eingänge bzw. Ausgänge (N) mit Beginn an der angegebenen Adresse (Bit). Sie können zwischen 1 und 255 Ein- und Ausgänge setzen bzw. rücksetzen.

Wenn die Operation Rücksetzen ein Zeitbit (T) oder Zählerbit (Z) angibt, dann setzt die Operation das Zeit- oder Zählerbit zurück und löscht den aktuellen Wert der Zeit bzw. des Zählers.

Fehlerbedingungen, die ENO = 0 setzen

- 0006 (Indirekte Adresse)
- 0091 (Operand außerhalb des Bereichs)

Bitwert direkt setzen und Bitwert direkt rücksetzen

Die Operationen Bitwert direkt setzen und Bitwert direkt rücksetzen setzen (schalten ein) oder rücksetzen (schalten aus) die angegebene Anzahl Eingänge bzw. Ausgänge (N) mit Beginn an der angegebenen Adresse (Bit) direkt/sofort. Sie können zwischen 1 und 128 Ein- und Ausgänge direkt setzen bzw. rücksetzen.

Das "I" zeigt direkten Zugriff an. Der neue Wert wird bei Ausführung der Operation sowohl direkt in den physikalischen Ausgang als auch in das Prozessabbild geschrieben. Hierin unterscheidet sich eine direkte Operation von anderen Operationen, bei denen der Wert für den angesprochenen Eingang bzw. Ausgang nur in das Prozessabbild geschrieben wird.

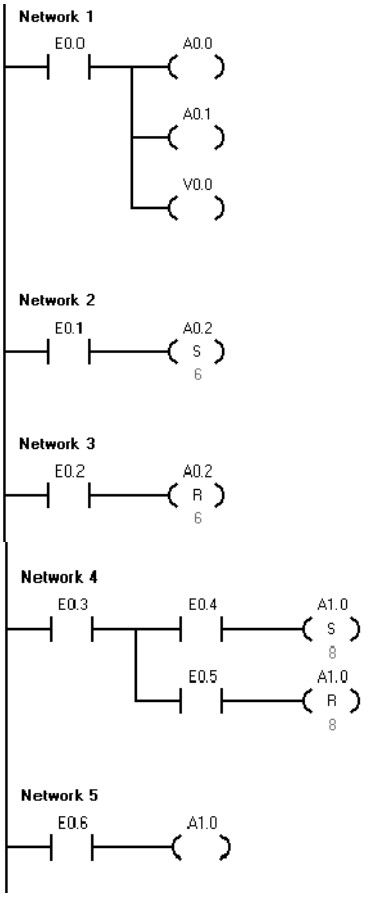
Fehlerbedingungen, die ENO = 0 setzen

- 0006 (Indirekte Adresse)
- 0091 (Operand außerhalb des Bereichs)

Tabelle 6-4 Gültige Operanden für die Bitverknüpfungsausgangsoperationen

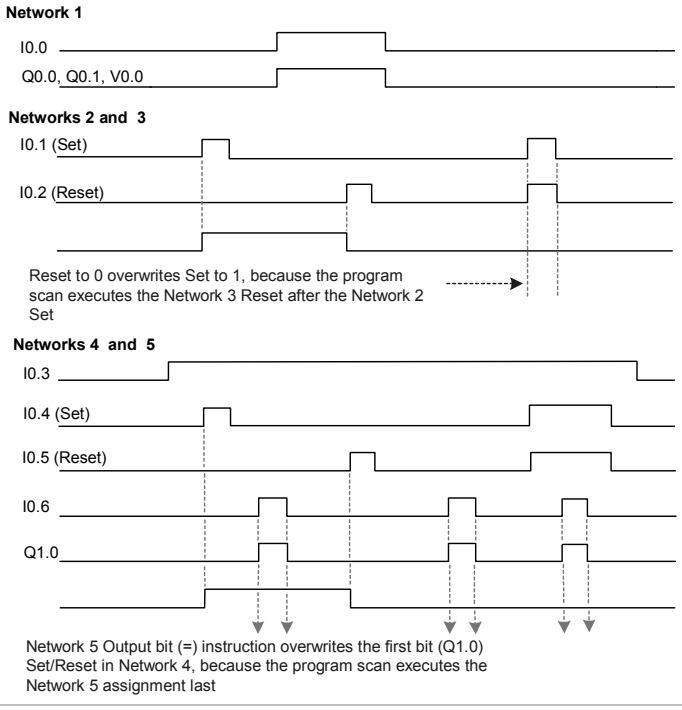
Eingänge/Ausgänge	Datentyp	Operanden
Bit	BOOL	E, A, V, M, SM, S, T, Z, L
Bit (direkt)	BOOL	A
N	BYTE	EB, AB, VB, MB, SMB, SB, LB, AC, *VD, *LD, *AC, Konstante

Beispiel: Spulen



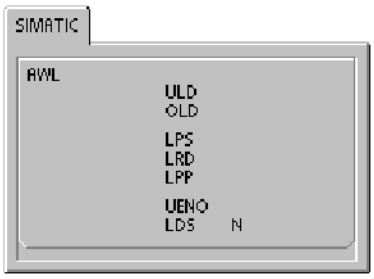

```
Netzwerk 1  //Die Operationen Zuweisen weisen Bitwerte zu externen Ein- und
            //Ausgängen (E, A) und internem Speicher (M, SM, T, Z, V, S, L) zu.
LD    E0.0
=     A0.0
=     A0.1
=     V0.0
Netzwerk 2  //Gruppe aus 6 zusammenhängenden Bits auf den Wert 1 setzen.
            //Adresse des Anfangsbit und Anzahl der zu setzenden Bits angeben.
            //Die Programmstatusanzeige für Setzen
            //ist EIN, wenn der Wert des ersten Bit (A0.2) 1 ist.
LD    E0.1
S     A0.2, 6
Netzwerk 3  //Gruppe aus 6 zusammenhängenden Bits auf den Wert 0 zurücksetzen.
            //Adresse des Anfangsbit und Anzahl der zurückzusetzenden Bits angeben.
            //Die Programmstatusanzeige für Rücksetzen
            //ist EIN, wenn der Wert des ersten Bit (A0.2) 0 ist.
LD    E0.2
R     A0.2, 6
Netzwerk 4  //Acht Ausgangsbits (A1.0 bis A1.7) als Gruppe setzen und rücksetzen.
LD    E0.3
LPS
U     E0.4
S     A1.0, 8
LPP
U     E0.5
R     A1.0, 8
Netzwerk 5  //Die Operationen Setzen und Rücksetzen führen die Funktion eines Haltere
            //lais aus. Stellen Sie sicher, dass die Bits Setzen/Rücksetzen nicht von
            //einer an deren Operation Zuweisen überschrieben werden.
            //In diesem Beispiel dient Netzwerk 4 dazu,
            //acht Ausgangsbits (A1.0 bis A1.7) als Gruppe zu setzen bzw. zurückzuset
            //zen. In RUN kann Netzwerk 5 den Bitwert von A1.0 überschreiben und
            //die Programmstatusanzeigen für Setzen/Rücksetzen in Netzwerk 4
            //steuern.
LD    E0.6
=     A1.0
```

Impulsdiagramm



Stackoperationen

Erste und zweite Stackebene durch UND verknüpfen



Die Operation Erste und zweite Stackebene durch UND verknüpfen (ULD) verknüpft die Werte der ersten und 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.

Erste und zweite Stackebene durch ODER verknüpfen

Die Operation Erste und zweite Stackebene durch ODER verknüpfen (OLD) verknüpft die Werte der ersten und 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.

Obersten Stackwert duplizieren

Die Operation Obersten Stackwert duplizieren (LPS) dupliziert den obersten Stackwert und schiebt ihn in den Stack. Der unterste Stackwert wird aus dem Stack geschoben und geht verloren.

Zweiten Stackwert kopieren

Die Operation Zweiten Stackwert kopieren (LRD) 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.

Obersten Stackwert aus Stack schieben

Die Operation Obersten Stackwert aus Stack schieben (LPP) schiebt den obersten Wert aus dem Stack. Der zweite Stackwert wird in die Spitze des Stack geschoben.

UND ENO

Die Operation UND ENO (UENO) verknüpft das ENO-Bit mit der Spitze des Stack durch UND, um das gleiche Ergebnis zu erzielen wie das ENO-Bit einer Box in KOP oder FUP. Das Ergebnis der UND-Verknüpfung ist der neue Wert an der Spitze des Stack.

ENO ist ein boolescher Ausgang von KOP- und FUP-Boxen. Liegt an einer Box am Eingang EN Signalfluss an und die Box wird fehlerfrei ausgeführt, leitet der Ausgang ENO den Signalfluss zum nächsten Element weiter. Sie können ENO als Freigabebit verwenden, das die erfolgreiche Ausführung einer Operation anzeigt. Das ENO-Bit wird zusammen mit dem obersten Stackwert eingesetzt, um den Signalfluss für die Ausführung nachfolgender Operationen zu beeinflussen. AWL-Operationen verfügen nicht über einen Eingang EN. Die Spitze des Stack muss 1 sein, damit bedingte Operationen ausgeführt werden. In AWL gibt es auch keinen Ausgang ENO. Die AWL-Operationen, jedoch, die den KOP- und FUP-Operationen mit ENO-Ausgängen entsprechen, setzen ein besonderes ENO-Bit. Auf dieses Bit kann mit der Operation UND ENO (UENO) zugegriffen werden.

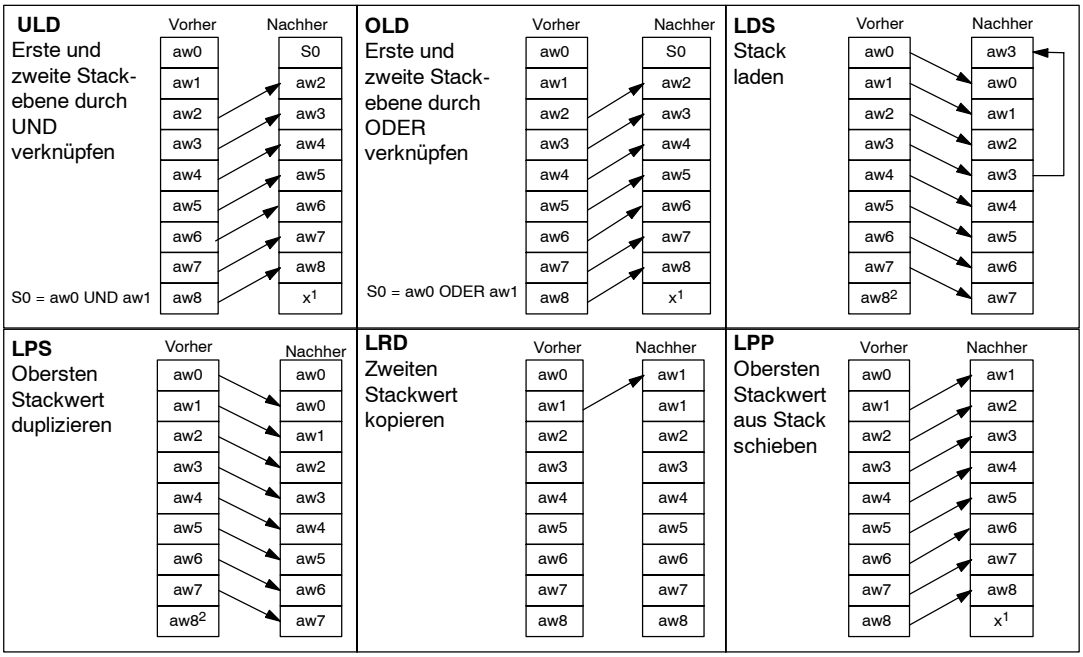
Stack laden

Die Operation Stack laden (LDS) dupliziert das Stackbit (N) im Stack und legt es in den obersten Stackwert. Der unterste Stackwert wird aus dem Stack geschoben und geht verloren.

Tabelle 6-5 Gültige Operanden für die Operation Stack laden

Eingänge/Ausgänge	Datentyp	Operanden
N	BYTE	Konstante (0 bis 8)

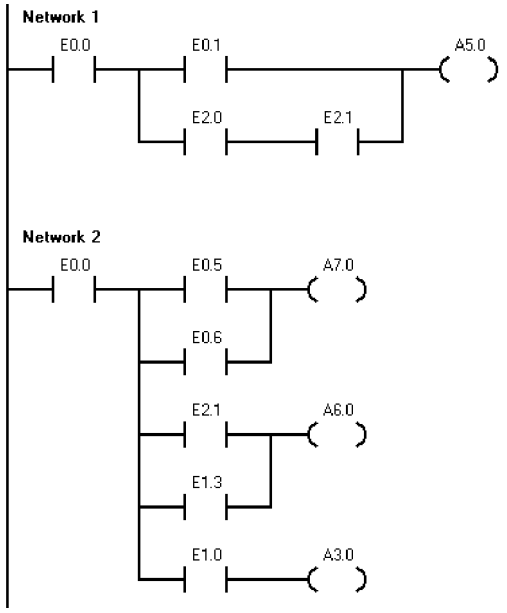
Die S7-200 löst die Steuerungslogik mit Hilfe eines logischen Stack (siehe Bild 6-3). In diesem Beispiel kennzeichnen ”aw0” bis ”aw7” die Ausgangswerte des logischen Stack, ”nw” kennzeichnet einen neuen Wert, der von der Operation bereitgestellt wird, und S0 kennzeichnet den errechneten Wert, der in dem logischen Stack gespeichert wird.



1 Der Wert ist unbekannt (es kann sich um den Wert 0 oder 1 handeln).
2 Nach der Ausführung einer Operation Obersten Stackwert duplizieren oder Stack laden geht der Wert aw8 verloren.

Bild 6-3 Funktionsweise der Stackoperationen

Beispiel: Stackoperationen




```
Netzwerk 1
LD    E0.0
LD    E0.1
LD    E2.0
U     E2.1
OLD
ULD
=     A5.0
Netzwerk 2
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
```

Operationen Bistabiler Funktionsbaustein: Vorrangig Setzen und Vorrangig Rücksetzen

Die Operation Bistabiler Funktionsbaustein: Vorrangig Setzen ist ein Flipflop, bei dem das Setzen Vorrang hat. Sind beide Signale Setzen (S1) und Rücksetzen (R) wahr, ist der Ausgang (OUT) wahr.

Die Operation Bistabiler Funktionsbaustein: Vorrangig Rücksetzen ist ein Flipflop, bei dem das Rücksetzen Vorrang hat. Sind beide Signale Setzen (S) und Rücksetzen (R) wahr, ist der Ausgang (OUT) falsch.

Der Parameter Bit gibt den booleschen Parameter an, der gesetzt bzw. zurückgesetzt wird. Der optionale Ausgang gibt den Signalzustand des Parameters Bit an.

Tabelle 6-7 zeigt die Zustandstabellen für das Beispielprogramm.

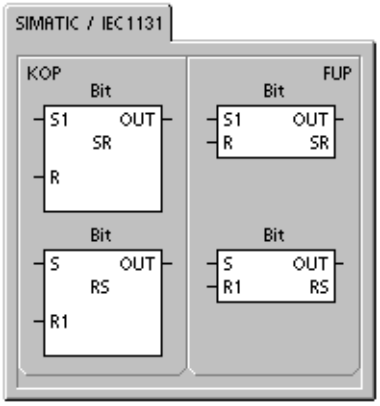


Tabelle 6-6 Gültige Operanden für die Operationen Bistabiler Funktionsbaustein: Vorrangig Setzen und Bistabiler Funktionsbaustein: Vorrangig Rücksetzen

Eingänge/Ausgänge	Datentypen	Operanden
S1, R	BOOL	E, A, V, M, SM, S, T, Z, Signalfluss
S, R1, OUT	BOOL	E, A, V, M, SM, S, T, Z, L, Signalfluss
Bit	BOOL	E, A, V, M, S

Beispiel: Operationen Bistabiler Funktionsbaustein: Vorrangig Setzen und Vorrangig Rücksetzen

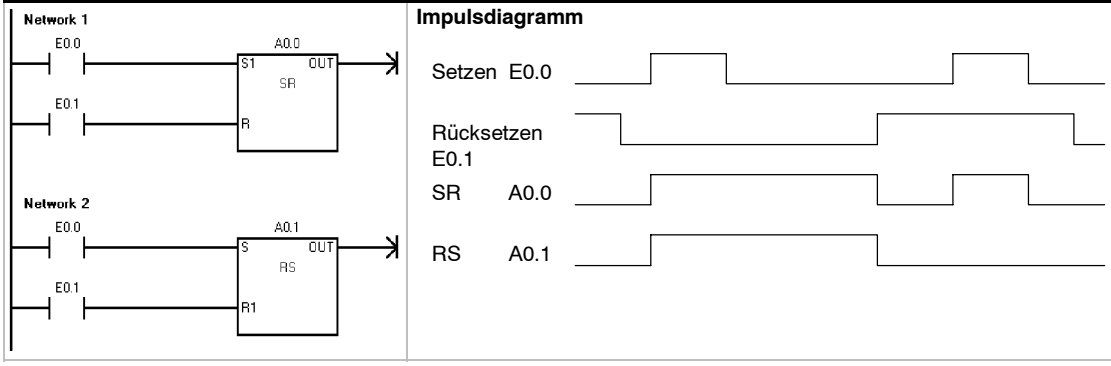
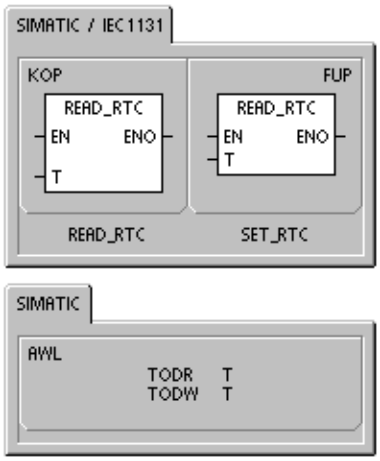


Tabelle 6-7 Zustandstabelle für die Operationen Bistabiler Funktionsbaustein: Vorrangig Setzen und Bistabiler Funktionsbaustein: Vorrangig Rücksetzen

Operation	S1	R	Out (Bit)
Bistabiler Funktionsbaustein: Vorrangig Setzen (SR)	0	0	Vorheriger Zustand
	0	1	0
	1	0	1
	1	1	1
Operation	**S**	**R1**	**Out (Bit)**
Bistabiler Funktionsbaustein: Vorrangig Rücksetzen (RS)	0	0	Vorheriger Zustand
	0	1	0
	1	0	1
	1	1	0

Uhroperationen

Echtzeituhr lesen und Echtzeituhr schreiben



Die Operation Echtzeituhr lesen (TODR) liest die aktuelle Uhrzeit und das aktuelle Datum aus der Hardware-Uhr und lädt beide in einen 8-Byte-Zeitpuffer mit Beginn an Adresse T. Die Operation Echtzeituhr schreiben (TODW) schreibt die aktuelle Uhrzeit und das aktuelle Datum der Hardware-Uhr in den 8-Byte-Zeitpuffer mit Beginn an der von T angegebenen Adresse.

Sie müssen alle Datum- und Zeitwerte im BCD-Format codieren (z.B. 16#97 für das Jahr 1997). Bild 6-4 zeigt das Format des Zeitpuffers (T).

Die Echtzeituhr (TOD) startet nach längerem Stromausfall oder nach Speicherverlust mit folgendem Datum und folgender Zeit:

Datum: 01-Jan-90
Zeit: 00:00:00
Wochentag: Sonntag

Fehlerbedingungen, die ENO = 0 setzen

- 0006 (Indirekte Adresse)
- 0007 (TOD-Datenfehler) *Nur Operation Echtzeituhr schreiben*
- 000C (keine Uhr vorhanden)

Tabelle 6-8 Gültige Operanden für die Uhroperationen

Eingänge/Ausgänge	Datentypen	Operanden
T	BYTE	EB, AB, VB, MB, SMB, SB, LB, *VD, *LD, *AC

T	T+1	T+2	T+3	T+4	T+5	T+6	T+7
Jahr: 00 bis 99	Monat: 01 bis 12	Tag: 01 bis 31	Stunden: 00 bis 23	Minuten: 00 bis 59	Sekunden: 00 bis 59	0	Wochentag: 0 bis 7*

*T+7 1=Sonntag, 7=Samstag
0 deaktiviert den Wochentag.

Bild 6-4 Format des 8-Byte-Zeitpuffers (T)

Tipp

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. SM4.3 wird gesetzt und zeigt dadurch an, dass von zwei verschiedenen Operationen gleichzeitig versucht wurde, auf die Uhr zuzugreifen (leichter Fehler 0007).

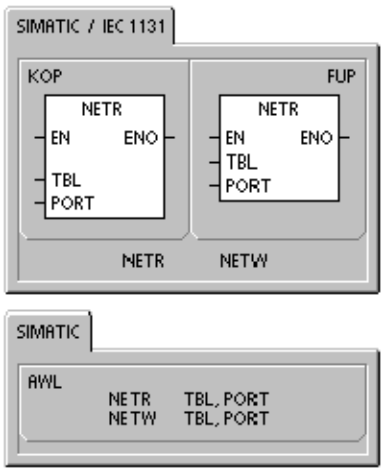
Die Echtzeituhr der S7-200 verwendet die beiden niederwertigsten Ziffern für die Jahresangabe. Deshalb wird das Jahr 2000 als das Jahr 00 dargestellt. Das Automatisierungssystem S7-200 nutzt die Informationen zum Jahr in keinster Weise. Anwenderprogramme jedoch, deren arithmetische Operationen oder Vergleichsoperationen den Jahreswert einsetzen, müssen die zweistellige Darstellung und den Jahrtausendwechsel beachten.

Schaltjahre werden bis zum Jahr 2096 korrekt behandelt.

Kommunikationsoperationen

Operationen Aus Netz lesen und In Netz schreiben

Die Operation Aus Netz lesen (NETR) löst eine Kommunikationsoperation aus, die entsprechend der Definition in der Tabelle (TBL) über die angegebene Schnittstelle (PORT) aus einem fernen Gerät Daten liest. Die Operation In Netz schreiben (NETW) löst eine Kommunikationsoperation aus, die entsprechend der Definition in der Tabelle (TBL) über die angegebene Schnittstelle (PORT) in ein fernes Gerät Daten schreibt.



Fehlerbedingungen, die ENO = 0 setzen:

- 0006 (Indirekte Adresse)
- Die Funktion gibt einen Fehler aus und setzt das E-Bit des Byte für den Tabellen-
  status (siehe Bild 6-5).

Mit der Operation Aus Netz lesen können Sie maximal 16 Bytes an Daten aus einer fernen Station lesen. Mit der Operation In Netz schreiben können Sie maximal 16 Bytes an Daten in eine ferne Station schreiben.

Sie können beliebig viele Operationen Aus Netz lesen und In Netz schreiben in Ihr Programm einbauen, doch es können maximal acht Operationen Aus Netz lesen und In Netz schreiben gleichzeitig bearbeitet werden. Es können beispielsweise 4 Operationen Aus Netz lesen und 4 Operationen In Netz schreiben oder 2 Operationen Aus Netz lesen und 6 Operationen In Netz schreiben gleichzeitig in der S7-200 aktiv sein.

Operations-Assistent

Mit dem Operations-Assistent für die Operationen Aus Netz lesen/In Netz schreiben richten Sie den Zähler ein. Sie rufen den Operations-Assistenten Aus Netz lesen/In Netz schreiben mit dem Menübefehl **Extras > Operations-Assistent** auf. Dann wählen Sie im Fenster des Operations-Assistenten die Option Aus Netz lesen/In Netz schreiben.

Tabelle 6-9 Gültige Operanden für die Operationen Aus Netz lesen und In Netz schreiben

Eingänge/Ausgänge	Datentyp	Operanden		
TBL	BYTE	VB, MB, *VD, *LD, *AC		
PORT	BYTE	Konstante	*für CPU 221, CPU 222, CPU 224:* *für CPU 226 und CPU 226XM:*	0 0 oder 1

Bild 6-5 beschreibt die Tabelle, auf die sich der Parameter TBL bezieht und Tabelle 6-10 führt die Fehlercodes auf.

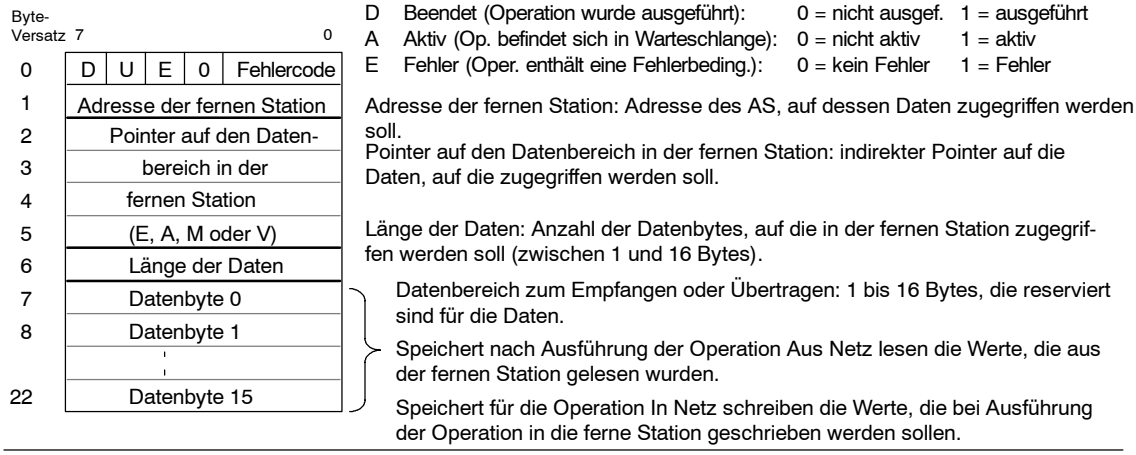
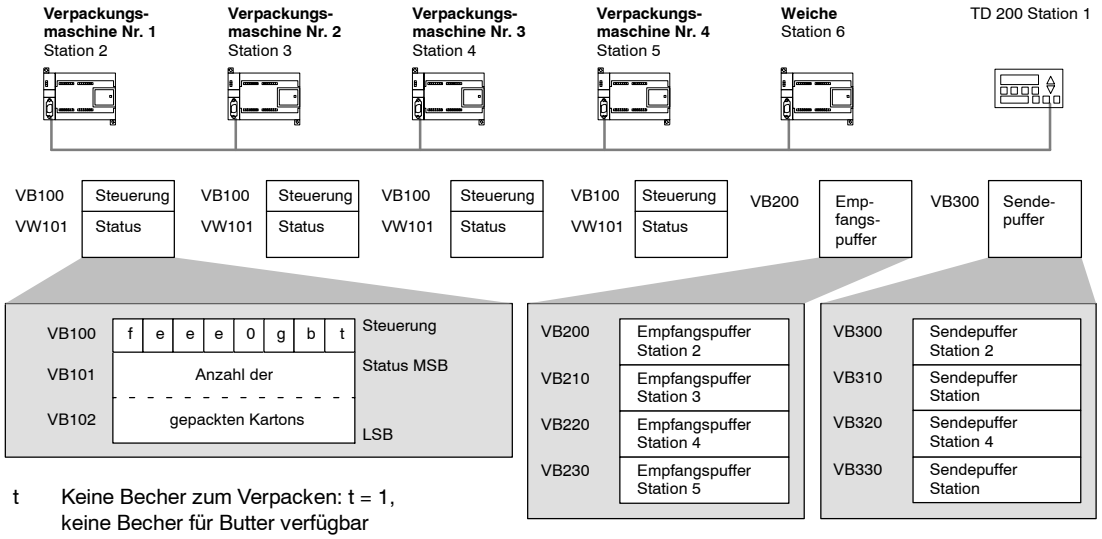


Bild 6-5 Parameter TBL für die Operationen Aus Netz lesen und In Netz schreiben

Tabelle 6-10 Fehlercodes für den Parameter TBL

Code	Definition
0	Kein Fehler
1	Timeout-Fehler: Ferne Station reagiert nicht.
2	Empfangsfehler: Parität, Rahmen oder Prüfsumme in Antwort.
3	Offline-Fehler: Konflikt durch doppelte Stationsadresse oder fehlerhafte Hardware.
4	Warteschlangenüberlauf: Mehr als 8 Operationen Aus Netz lesen oder In Netz schreiben wurden aktiviert.
5	Protokollverletzung: Versuchte Ausführung von Aus Netz lesen oder In Netz schreiben ohne Aktivierung des PPI-Mastermodus in SMB30 oder SMB130.
6	Unzulässiger Parameter: Parameter TBL enthält einen unzulässigen oder ungültigen Wert.
7	Keine Ressource: Ferne Station ist belegt. (Ein Ladevorgang ist in Bearbeitung.)
8	Fehler Ebene 7: Anwendungsprotokollverletzung.
9	Meldungsfehler: Adresse oder Länge der Daten falsch.
A bis F	Nicht verwendet. (Reserviert).

Bild 6-6 zeigt ein Beispiel für den Einsatz der Operationen Aus Netz lesen und In Netz schreiben. 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 S7-200 Geräte steuern die Verpackungsmaschinen und eine S7-200 mit einer Bedienerschnittstelle TD 200 steuert die Weiche.



t Keine Becher zum Verpacken: t = 1, keine Becher für Butter verfügbar

b Wenig Kartons vorhanden: b = 1, innerhalb der nächsten 30 Minuten müssen Kartons aufgefüllt werden

g Wenig Klebstoff vorhanden: g = 1, innerhalb der nächsten 30 Minuten muss Klebstoff nachgefüllt werden

eee Fehlercode, gibt die jeweilige Fehlerart an

f Fehleranzeige: f = 1, an der Verpackungsmaschine ist ein Fehler aufgetreten

Bild 6-6 Beispiel für die Operationen Aus Netz lesen und In Netz schreiben

Bild 6-7 zeigt den Empfangspuffer (VB200) und den Sendepuffer (VB300) für den Zugriff auf die Daten in Station 2. Die S7-200 liest mit der Operation Aus Netz lesen regelmäßig die Steuerungs- und Statusinformationen aus den einzelnen Verpackungsmaschinen aus. Jedesmal, wenn eine Verpackungsmaschine 100 Kartons gepackt hat, sendet die Weiche mit der Operation In Netz schreiben eine Meldung, um das Statuswort zurückzusetzen.

Empfangspuffer *zum Lesen aus Verpackungsmaschine Nr. 1*

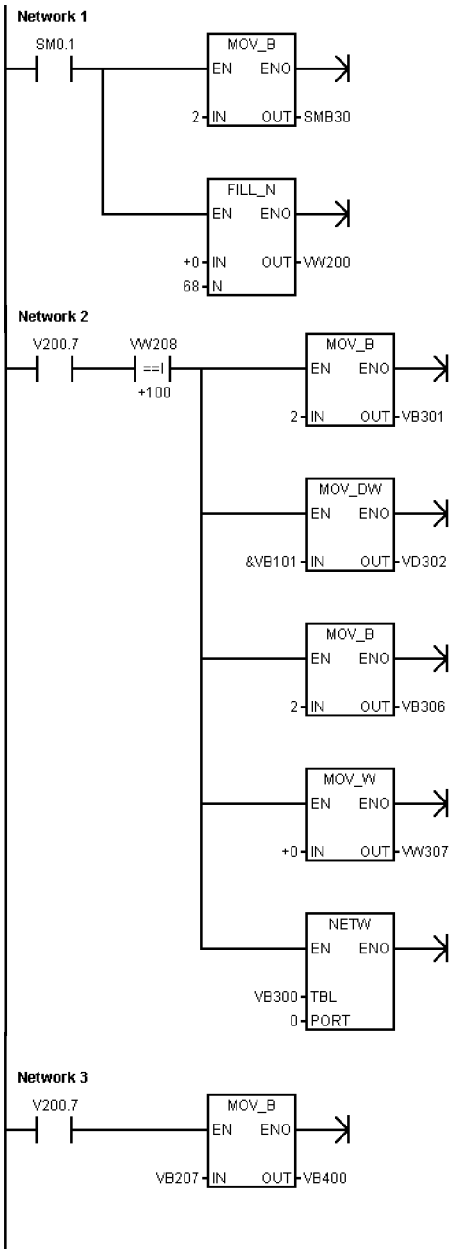
	7				0
VB200	D	U	E	0	Fehlercode
VB201	Adresse der fernen Station = 2				
VB202	Pointer auf den				
VB203	Datenbereich				
VB204	in der fernen				
VB205	Station = (VB100)				
VB206	Datenlänge = 3 Bytes				
VB207	Steuerung				
VB208	Status (MSB)				
VB209	Status (LSB)				

Sendepuffer *zum Löschen des Zählwerts von Verpackungsmaschine Nr. 1*

	7				0
VB300	D	U	E	0	Fehlercode
VB301	Adresse der fernen Station = 2				
VB302	Pointer auf den				
VB303	Datenbereich				
VB304	in der fernen				
VB305	Station = (&VB101)				
VB306	Datenlänge = 2 Bytes				
VB307	0				
VB308	0				

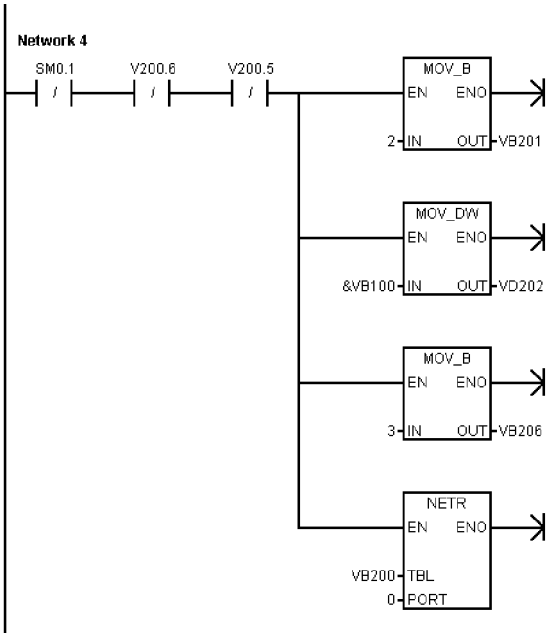
Bild 6-7 TBL-Beispieldaten für das Beispiel Aus Netz lesen/In Netz schreiben

Beispiel: Operationen Aus Netz lesen und In Netz schreiben



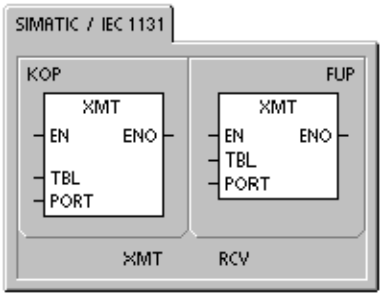

```
Netzwerk 1      //Im ersten Zyklus PPI-Mastermodus freigeben
                //und alle Empfangs- und Sendepuffer löschen.
LD      SM0.1
MOVB    2, SMB30
FILL    +0, VW200, 68

Netzwerk 2      //Wenn das Bit NETR beendet (V200.7) gesetzt ist
                //und 100 Kartons gepackt wurden:
                //1. Stationsadresse von Verpackungsmaschine Nr. 1
                //laden.
                //2. Pointer auf die Daten in der fernen Station laden.
                //3. Länge der zu sendenden Daten laden.
                //4. Die zu sendenden Daten laden.
                //5. Anzahl gepackter Kartons
                //von Verpackungsmaschine Nr. 1 zurücksetzen.
LD      V200.7
UW=     VW208, +100
MOVB    2, VB301
MOVD    &VB101, VD302
MOVB    2, VB306
MOVW    +0, VW307
NETW    VB300, 0

Netzwerk 3      //Wird das Bit NETR beendet gesetzt,
                //Steuerungsinformationen von
                //Verpackungsmaschine Nr. 1 laden.
LD      V200.7
MOVB    VB207, VB400
```

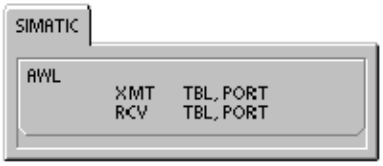
Beispiel: Operationen Aus Netz lesen und In Netz schreiben, Fortsetzung




```
Netzwerk 4    //Wenn nicht erster Zyklus und wenn keine Fehler:
              //1. Stationsadresse von Verpackungsmaschine Nr. 1
              //laden.
              //2. Pointer auf die Daten in der fernen Station laden.
              //3. Länge der zu empfangenen Daten laden.
              //4. Steuerungs- und Statusbyte
              //   von Verpackungsmaschine Nr. 1 lesen.
LDN     SM0.1
UN      V200.6
UN      V200.5
MOVB    2, VB201
MOVD    &VB100, VD202
MOVB    3, VB206
NETR    VB200, 0
```

Operationen Meldung aus Zwischenspeicher übertragen und Meldung in Zwischenspeicher empfangen (frei programmierbare Kommunikation)





Die Operation Meldung aus Zwischenspeicher übertragen (XMT) wird in der frei programmierbaren Kommunikation zum Übertragen von Daten über die Kommunikationsschnittstelle(n) verwendet.

Die Operation Meldung in Zwischenspeicher empfangen (RCV) veranlasst und beendet die Funktion zum Meldungsempfang. 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 (TBL) abgelegt. Der erste Eintrag im Puffer gibt die Anzahl der empfangenen Bytes an.

Fehlerbedingungen, die ENO = 0 setzen

- 0006 (Indirekte Adresse)
- 0009 (Gleichzeitiges Senden/Empfangen an Schnittstelle 0)
- 000B (Gleichzeitiges Senden/Empfangen an Schnittstelle 1)
- Empfangsparameterfehler setzt SM86.6 oder SM186.6
- Die S7-200 CPU ist nicht im Modus für frei programmierbare Kommunikation.

Tabelle 6-11 Gültige Operanden für die Operationen Meldung aus Zwischenspeicher übertragen und Meldung in Zwischenspeicher empfangen

Eingänge/Ausgänge	Datentyp	Operanden		
TBL	BYTE	EB, AB, VB, MB, SMB, SB, *VD, *LD, *AC		
PORT	BYTE	Konstante	*für CPU 221, CPU 222, CPU 224:* *für CPU 226 und CPU 226XM:*	0 0 oder 1

Ausführliche Informationen zur Verwendung der frei programmierbaren Kommunikation finden Sie im Abschnitt ”Erstellen von anwenderdefinierten Protokollen mit der frei programmierbaren Kommunikation” auf Seite 227 in Kapitel 7.

Steuern der seriellen Kommunikationsschnittstelle in der frei programmierbaren Kommunikation

In der frei programmierbaren Kommunikation kann die serielle Kommunikationsschnittstelle der S7-200 mit Hilfe des Anwenderprogramms gesteuert werden. Haben Sie die frei programmierbare Kommunikation gewählt, steuert das 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 S7-200 über zwei Schnittstellen verfügt) dienen zum Auswählen von Baudrate und Parität.

Befindet sich die S7-200 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 Anzeigegerät 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 S7-200 in der frei programmierbaren Kommunikation kommunizieren soll.

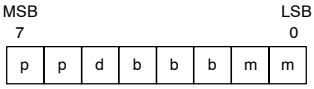
Die frei programmierbare Kommunikation kann nur im Betriebszustand RUN der S7-200 aktiviert werden. Sie geben die frei programmierbare Kommunikation frei, indem Sie in dem Feld zum Auswählen des Protokolls in SMB30 (Schnittstelle 0) oder in SMB130 (Schnittstelle 1) den Wert ”01” einstellen. Während der frei programmierbaren Kommunikation können Sie nicht mit dem Programmiergerät kommunizieren.

Tipp

Die frei programmierbare Kommunikation kann über SM0.7 gesteuert werden, der die aktuelle Position des Betriebsartenschalters wiedergibt. Ist SM0.7 gleich 0, befindet sich der Schalter in der Position TERM. Ist SM0.7 gleich 1, befindet sich der Schalter in der Position RUN. Wenn Sie die frei programmierbare Kommunikation nur freigeben, wenn der Schalter auf RUN steht, dann können Sie mit dem Programmiergerät die Funktion der S7-200 überwachen und steuern, indem Sie den Schalter in eine andere Position versetzen.

Wechseln von der PPI-Kommunikation zur 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. Bild 6-8 beschreibt das Steuerbyte für die frei programmierbare Kommunikation. Es wird ein Stoppbit für alle Konfigurationen erzeugt.



SMB30 = Schnittstelle 0
SMB130 = Schnittstelle 1

pp: Auswahl der Parität
- 00 = keine Parität
- 01 = gerade Parität
- 10 = keine Parität
- 11 = ungerade Parität

d: Datenbits pro Zeichen
- 0 = 8 Bits pro Zeichen
- 1 = 7 Bits pro Zeichen

bbb: Baudrate für die frei programmierbare Kommunikation
- 000 = 38.400 Baud
- 001 = 19.200 Baud
- 010 = 9.600 Baud
- 011 = 4.800 Baud
- 100 = 2.400 Baud
- 101 = 1.200 Baud
- 110 = 115,2 kBaud[1]
- 111 = 57,6 kBaud[1]

[1] S7-200 CPUs ab Version 1.2 unterstützen die Baudraten 57,6 kBaud und 115,2 kBaud.

mm: Auswahl des Protokolls
- 00 = PPI-Slavemodus
- 01 = Protokoll der frei programmierbaren Kommunikation
- 10 = PPI-Mastermodus
- 11 = Reserviert (Voreinstellung PPI/Slave-Modus)

Bild 6-8 Steuerbyte für die frei programmierbare Kommunikation (SMB30 oder SMB130)

Senden von Daten

Mit der Operation Meldung aus Zwischenspeicher übertragen können Sie einen Puffer mit maximal 255 Zeichen senden.

Bild 6-9 zeigt das Format des Sendepuffers.

Wenn dem Ereignis Übertragung beendet ein Interruptprogramm zugeordnet ist, erzeugt die S7-200 einen Interrupt (Interruptereignis 9 für Schnittstelle 0 und Interruptereignis 26 für Schnittstelle 1), nachdem das letzte Zeichen des Puffers gesendet wurde.

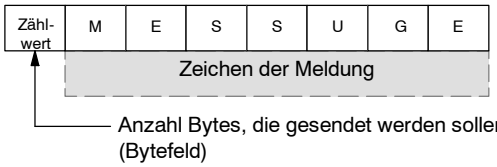


Bild 6-9 Format des Sendepuffers

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.

Mit der Operation Meldung aus Zwischenspeicher übertragen können Sie eine BREAK-Bedingung erzeugen, indem Sie die Anzahl der Zeichen auf Null setzen und anschließend die Operation Meldung aus Zwischenspeicher übertragen ausführen. Dadurch wird in der Zeile für 16-Bit-Zeiten bei der aktuellen Baudrate eine BREAK-Bedingung erzeugt. Das Übertragen einer BREAK-Bedingung läuft genauso ab wie das Übertragen einer anderen Meldung. Es wird ein Sende-Interrupt erzeugt, wenn die BREAK-Bedingung übertragen ist, und SM4.5 oder SM4.6 zeigt den aktuellen Status der Übertragung an.

Empfangen von Daten

Mit der Operation Meldung in Zwischenspeicher empfangen können Sie einen Puffer mit maximal 255 Zeichen empfangen.

Bild 6-10 zeigt das Format des Empfangspuffers.

Wenn dem Ereignis Empfangen beendet ein Interruptprogramm zugeordnet ist, erzeugt die S7-200 einen Interrupt (Interruptereignis 23 für Schnittstelle 0 und Interruptereignis 24 für Schnittstelle 1), nachdem das letzte Zeichen im Puffer empfangen wurde.

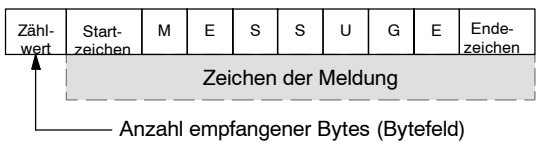


Bild 6-10 Format des Empfangspuffers

Sie können durch Beobachten von SMB86 (Schnittstelle 0) und SMB186 (Schnittstelle 1) Meldungen ohne Interrupts empfangen. SMB86 bzw. SMB186 ist ungleich Null, wenn die Operation Meldung in Zwischenspeicher empfangen inaktiv ist oder beendet wurde. SMB86 bzw. SMB186 ist gleich Null, wenn Daten empfangen werden.

Bei der Operation Meldung in Zwischenspeicher empfangen können Sie die Bedingungen für den Start und das Ende der Meldung auswählen (siehe Tabelle 6-12). Hierzu verwenden Sie SMB86 bis SMB94 für Schnittstelle 0 und SMB186 bis SMB194 für Schnittstelle 1.

Tipp

Eine freigegebene Funktion zum Meldungsempfang wird sofort automatisch beendet, wenn ein Überlauf- oder Paritätsfehler auftritt. Sie müssen eine Bedingung für den Beginn und eine Bedingung für das Ende (maximale Zeichenzahl) definieren, damit die Funktion zum Meldungsempfang fehlerfrei arbeiten kann.

Tabelle 6-12 Bytes des Empfangspuffers (SMB86 bis SMB94 und SM1B86 bis SMB194)

Schnitt-stelle 0	Schnitt-stelle 1	Beschreibung
SMB86	SMB186	Statusbyte zum Empfangen von Meldungen MSB 7 … LSB 0: n r e 0 0 t c p n: 1 = Funktion zum Meldungsempfang beendet: Anwender hat Befehl zum Deaktivieren abgesetzt r: 1 = Funktion zum Meldungsempfang beendet: Fehler in Eingabeparameter oder fehlende Start- oder Endebedingung e: 1 = Endezeichen empfangen t: 1 = Funktion zum Meldungsempfang beendet: Zeit abgelaufen c: 1 = Funktion zum Meldungsempfang beendet: maximale Zeichenzahl erreicht p 1 = Funktion zum Meldungsempfang beendet: Paritätsfehler
SMB87	SMB187	Steuerbyte zum Empfangen von Meldungen MSB 7 … LSB 0: en sc ec il c/m tmr bk 0 en: 0 = Funktion zum Meldungsempfang ist gesperrt. 1 = Funktion zum Meldungsempfang ist aktiviert. Das Bit zum Sperren/Freigeben des Meldungsempfangs wird jedesmal, wenn die Operation RCV bearbeitet wird, geprüft. sc: 0 = SMB88 oder SMB188 ignorieren. 1 = Mit dem Wert von SMB88 oder von SMB188 den Beginn der Meldung erkennen. ec: 0 = SMB89 oder SMB189 ignorieren. 1 = Mit dem Wert von SMB89 oder von SMB189 das Ende der Meldung erkennen. il: 0 = SMW90 oder SMW190 ignorieren. 1 = Mit dem Wert von SMW90 oder SMW190 Leerlaufbedingung erkennen. c/m: 0 = Zeit misst die Zeit zwischen den Zeichen. 1 = Zeit misst die Zeit der Meldungen. tmr: 0 = SMW92 oder SMW192 ignorieren. 1 = Empfang beenden, wenn die Zeitdauer von SMW92 bzw. SMW192 überschritten ist. bk: 0 = Pausenbedingungen ignorieren. 1 = Mit Pausenbedingung Meldungsbeginn erkennen.
SMB88	SMB188	Zeichen für den Beginn einer Meldung
SMB89	SMB189	Zeichen für das Ende einer Meldung
SMW90	SMW190	Leerlaufzeit in Millisekunden. Das erste Zeichen, das nach Ablauf der Leerlaufzeit empfangen wird, kennzeichnet den Beginn einer neuen Meldung.
SMW92	SMW192	Wert für die Zeitüberwachung beim Messen der Zeit zwischen den Zeichen und der Zeit der Meldungen in Millisekunden. Ist die Zeit überschritten, wird die Funktion zum Meldungsempfang beendet.
SMB94	SMB194	Maximale Zeichenzahl, die empfangen werden kann (1 bis 255 Bytes). Dieser Bereich muss auf die maximal zu erwartende Puffergröße eingestellt werden, auch wenn der Empfangsabbruch durch die Zeichenzählfunktion nicht eingesetzt wird.

Start- und Endebedingungen der Operation Meldung in Zwischenspeicher empfangen

Die Operation Meldung in Zwischenspeicher empfangen definiert mit den Bits des Steuerbyte für den Meldungsempfang (SMB87 oder SMB187) die Bedingungen für den Meldungsbeginn und das Meldungsende.

Tipp

Ist die Kommunikationsschnittstelle durch andere Geräte besetzt, wenn die Operation Meldung in Zwischenspeicher empfangen ausgeführt wird, kann die Funktion zum Empfangen ein Zeichen in der Mitte des Zeichens empfangen, was möglicherweise zu einem Paritätsfehler und zur Beendigung des Meldungsempfangs führt. Ist Parität nicht aktiviert, kann die empfangene Meldung falsche Zeichen enthalten. Diese Situation kann auftreten, wenn als Startbedingung ein bestimmtes Startzeichen oder ein beliebiges Zeichen definiert ist, wie es unter 2. und 6. im folgenden beschrieben ist.

Die Operation Meldung in Zwischenspeicher empfangen unterstützt verschiedene Startbedingungen für Meldungen. Wenn Sie eine Startbedingung angeben, die eine Pause oder eine Leerlauferkennung enthält, vermeiden Sie dieses Problem, indem die Funktion zum Meldungsempfang dazu gezwungen wird, den Meldungsbeginn mit dem Beginn eines Zeichens zu synchronisieren, bevor Zeichen im Meldungspuffer abgelegt werden.

Die Operation Meldung in Zwischenspeicher empfangen unterstützt verschiedene Startbedingungen.

1. *Leerlauferkennung:* Die Leerlaufbedingung ist definiert als Ruhe- oder Leerlaufzeit in der Übertragungsleitung. Der Empfang wird begonnen, wenn die Kommunikationsleitung während der in SMW90 oder SMW190 angegebenen Anzahl Millisekunden im Ruhe- oder Leerlaufzustand war. Wenn die Operation Meldung in Zwischenspeicher empfangen in Ihrem Programm ausgeführt wird, beginnt die Funktion zum Meldungsempfang die Suche nach der Leerlaufbedingung. Werden Zeichen empfangen, bevor die Leerlaufzeit abgelaufen ist, ignoriert die Funktion zum Meldungsempfang diese Zeichen und startet die Leerlaufzeit mit dem Wert aus SMW90 oder SMW190 neu (siehe Bild 6-11). Nachdem die Leerlaufzeit abgelaufen ist, speichert die Funktion zum Meldungsempfang alle nachfolgend empfangenen Zeichen im Meldungspuffer.

   Die Leerlaufzeit muss immer größer sein als die Zeit für die Übertragung eines Zeichens (Startbit, Datenbits, Paritäts- und Stoppbits) bei der angegebenen Baudrate. Ein typischer Wert für die Leerlaufzeit ist die Zeit für drei Zeichen bei der angegebenen Baudrate.

   Die Leerlauferkennung verwenden Sie als Startbedingung für Binärprotokolle, für Protokolle, die über kein bestimmtes Startzeichen verfügen, oder für den Fall, dass ein Protokoll eine Mindestzeit zwischen den Meldungen fordert.

   Einrichtung: il = 1, sc = 0, bk = 0, SMW90/SMW190 = Leerlauf-Timeout in Millisekunden

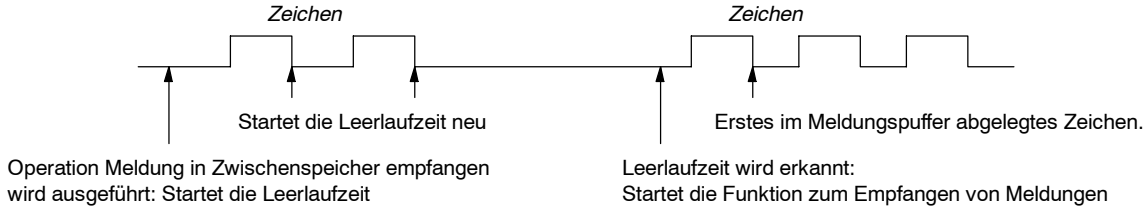


Bild 6-11 Starten der Operation Meldung in Zwischenspeicher empfangen mit Leerlauferkennung

2. *Erkennung Startzeichen:* Das Startzeichen ist ein beliebiges Zeichen, das als erstes Zeichen einer Meldung verwendet wird. Eine Meldung beginnt, wenn das in SMB88 oder SMB188 angegebene Startzeichen empfangen wird. Die Funktion zum Meldungsempfang speichert das Startzeichen im Empfangspuffer als erstes Zeichen der Meldung. Die Funktion zum Meldungsempfang ignoriert beliebige anderen Zeichen, die vor dem Startzeichen empfangen werden. Das Startzeichen und alle Zeichen, die nach dem Startzeichen empfangen werden, werden im Meldungspuffer abgelegt.

   Üblicherweise verwenden Sie die Startzeichenerkennung bei ASCII-Protokollen, in denen alle Meldungen mit dem gleichen Zeichen beginnen.

   Einrichtung: il = 0, sc = 1, bk = 0, SMW90/SMW190 = nicht relevant, SMB88/SMB188 = Startzeichen

3. *Leerlauf und Startzeichen:* Die Operation Meldung in Zwischenspeicher empfangen kann eine Meldung mit einer Kombination aus Leerlauf und Startzeichen beginnen. Wenn die Operation Meldung in Zwischenspeicher empfangen ausgeführt wird, sucht die Funktion zum Meldungsempfang nach einer Leerlaufbedingung. Nachdem die Leerlaufbedingung gefunden wurde, wartet die Funktion zum Meldungsempfang auf das angegebene Startzeichen. Werden beliebige andere Zeichen, nur nicht das Startzeichen, empfangen, beginnt die Funktion zum Meldungsempfang die Suche nach einer Leerlaufbedingung erneut. Alle Zeichen, die vor der Leerlaufbedingung und vor dem Startzeichen empfangen werden, werden ignoriert. Das Startzeichen wird mit allen nachfolgenden Zeichen im Meldungspuffer abgelegt.

   Die Leerlaufzeit muss immer größer sein als die Zeit für die Übertragung eines Zeichens (Startbit, Datenbits, Paritäts- und Stoppbits) bei der angegebenen Baudrate. Ein typischer Wert für die Leerlaufzeit ist die Zeit für drei Zeichen bei der angegebenen Baudrate.

   Üblicherweise verwenden Sie diese Art von Startbedingung bei einem Protokoll, das eine Mindest- und eine Höchstzeit zwischen den Meldungen angibt, und wenn das erste Zeichen der Meldung eine Adresse oder dergleichen ist, die ein bestimmtes Gerät angibt. Dies ist sehr nützlich, wenn Sie ein Protokoll implementieren, bei dem sich mehrere Geräte an der Kommunikation beteiligen. In diesem Fall löst die Operation Meldung in Zwischenspeicher empfangen nur dann einen Interrupt aus, wenn eine Meldung für die spezifische Adresse oder die spezifischen Geräte empfangen wird, die vom Startzeichen angegeben werden.

   Einrichtung: il = 1, sc = 1, bk = 0, SMW90/SMW190 > 0, SMB88/SMB188 = Startzeichen

4. *Pauseerkennung:* Eine Pause wird angezeigt, wenn die empfangenen Daten für einen Zeitraum, der länger ist als die Übertragungszeit für ein komplettes Zeichen, auf dem Wert Null gehalten werden. Die Übertragungszeit für ein komplettes Zeichen ist definiert als die Gesamtzeit für Start-, Daten, Paritäts- und Stoppbits. Wurde die Operation Meldung in Zwischenspeicher empfangen so konfiguriert, dass die Meldung beim Empfang einer Pausebedingung begonnen werden soll, werden alle Zeichen, die nach der Pausebedingung empfangen werden, im Meldungspuffer abgelegt. Alle Zeichen, die vor der Pausebedingung empfangen wurden, werden ignoriert.

   Üblicherweise verwenden Sie eine Pauseerkennung als Startbedingung nur dann, wenn dies für das Protokoll erforderlich ist.

   Einrichtung: il = 0, sc = 0, bk = 1, SMW90/SMW190 = nicht relevant, SMB88/SMB188 = nicht relevant

5. *Pause und Startzeichen:* Die Operation Meldung in Zwischenspeicher empfangen kann so konfiguriert werden, dass nach dem Empfang einer Pausebedingung und eines bestimmten Startzeichens Zeichen empfangen werden. Nach der Pausebedingung wartet die Funktion zum Meldungsempfang auf das angegebene Startzeichen. Werden beliebige andere Zeichen, nur nicht das Startzeichen, empfangen, beginnt die Funktion zum Meldungsempfang die Suche nach einer Pausebedingung erneut. Alle Zeichen, die vor der Pausebedingung und vor dem Startzeichen empfangen werden, werden ignoriert. Das Startzeichen wird mit allen nachfolgenden Zeichen im Meldungspuffer abgelegt.

   Einrichtung: il = 0, sc = 1, bk = 1, SMW90/SMW190 = nicht relevant, SMB88/SMB188 = Startzeichen

6. *Beliebiges Zeichen:* Die Operation Meldung in Zwischenspeicher empfangen kann so konfiguriert werden, dass sofort alle Zeichen empfangen und im Meldungspuffer abgelegt werden. Dies ist ein Sonderfall der Leerlauferkennung. In diesem Fall wird die Leerlaufzeit (SMW90 oder SMW190) auf Null gesetzt. So wird die Operation Meldung in Zwischenspeicher empfangen gezwungen, sofort nach der Ausführung Zeichen zu empfangen.

   Einrichtung: il = 1, sc = 0, bk = 0, SMW90/SMW190 = 0, SMB88/SMB188 = nicht relevant

   Bei Meldungsbeginn mit einem beliebigen Zeichen kann der Meldungs-Timer für die Zeitüberwachung des Meldungsempfangs eingesetzt werden. Dies ist nützlich, wenn mittels frei programmierbarer Kommunikation der Master- oder Host-Anteil eines Protokolls implementiert wird und es erforderlich ist, eine Zeitüberwachung für den Fall zu haben, dass von einem Slave innerhalb einer bestimmten Zeit keine Antwort empfangen wird. Der Meldungs-Timer beginnt zu laufen, wenn die Operation Meldung in Zwischenspeicher empfangen ausgeführt wird, weil die Leerlaufzeit auf Null gesetzt wurde. Der Meldungs-Timer läuft ab und beendet die Funktion zum Meldungsempfang, wenn keine andere Endebedingung erfüllt wird.

   Einrichtung: il = 1, sc = 0, bk = 0, SMW90/SMW190 = 0, SMB88/SMB188 = nicht relevant
   c/m = 1, tmr = 1, SMW92 = Meldungs-Timeout in Millisekunden

Die Operation Meldung in Zwischenspeicher empfangen unterstützt verschiedene Arten, um eine Meldung zu beenden. Die Meldung kann mit einer oder mehreren der folgenden Bedingungen beendet werden:

1. *Endezeichenerkennung:* Das Endezeichen ist ein beliebiges Zeichen, das das Ende einer Meldung kennzeichnet. Nachdem die Operation Meldung in Zwischenspeicher empfangen die Startbedingung erkannt hat, prüft sie jedes Zeichen daraufhin, ob es dem Endezeichen entspricht. Wenn das Endezeichen empfangen wird, wird es im Meldungspuffer abgelegt und der Empfang wird beendet.

   Üblicherweise verwenden Sie die Endezeichenerkennung bei ASCII-Protokollen, in denen alle Meldungen mit einem bestimmten Zeichen enden. Sie können die Endezeichenerkennung zusammen mit dem Timer für die Zeit zwischen den Zeichen, mit dem Meldungs-Timer oder mit dem Zählwert für die Höchstzeichenzahl verwenden, um eine Meldung zu beenden.

   Einrichtung: ec = 1, SMB89/SMB189 = Endezeichen

2. *Timer für die Zeit zwischen den Zeichen:* Die Zeit zwischen den Zeichen ist die Zeit vom Ende eines Zeichens (dem Stoppbit) bis zum Ende des nächsten Zeichens (dem Stoppbit). Überschreitet die Zeit zwischen den Zeichen (einschließlich dem zweiten Zeichen) die Anzahl der Millisekunden, die in SMW92 oder SMW192 angegeben ist, dann wird die Funktion zum Meldungsempfang beendet. Der Timer für die Zeit zwischen den Zeichen wird bei jedem empfangenen Zeichen neu gestartet (siehe Bild 6-12).

   Mit dem Timer für die Zeit zwischen den Zeichen können Sie eine Meldung bei Protokollen beenden, die über kein bestimmtes Zeichen für das Ende einer Meldung verfügen. Dieser Timer muss auf einen Wert gesetzt werden, der größer ist als die Zeit für ein Zeichen bei der eingestellten Baudrate, weil der Timer immer die Zeit umfasst, die für den Empfang eines vollständigen Zeichens (Startbit, Datenbits, Paritäts- und Stoppbit) erforderlich ist.

   Sie können den Timer für die Zeit zwischen den Zeichen zusammen mit der Endezeichenerkennung und mit dem Zählwert für die Höchstzeichenzahl verwenden, um eine Meldung zu beenden.

   Einrichtung: c/m = 0, tmr = 1, SMW92/SMW192 = Timeout in Millisekunden

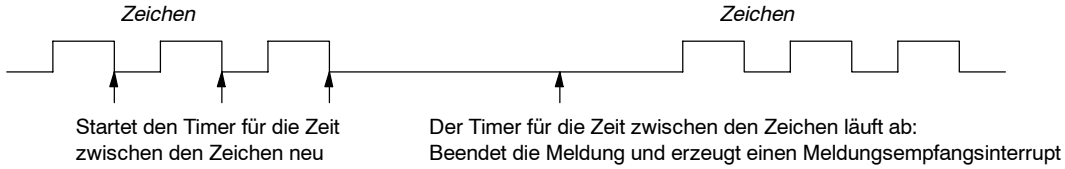


Bild 6-12 Beenden der Operation Meldung in Zwischenspeicher empfangen mit dem Timer für die Zeit zwischen den Zeichen

3. *Meldungs-Timer:* Der Meldungs-Timer beendet eine Meldung nach einer bestimmten Zeit nach dem Beginn der Meldung. Der Meldungs-Timer beginnt zu zählen, sowie die Startbedingung(en) für die Funktion zum Meldungsempfang erfüllt ist/sind. Der Meldungs-Timer läuft ab, wenn die Anzahl an Millisekunden, die in SMW92 oder SMW192 angegeben wurde, abgelaufen ist (siehe Bild 6-13).

   Üblicherweise verwenden Sie einen Meldungs-Timer, wenn die Kommunikationsgeräte nicht sicherstellen können, dass zwischen den Zeichen keine Zeitlücken auftreten, oder wenn Sie im Modembetrieb arbeiten. Bei Modems können Sie einen Meldungs-Timer verwenden, um eine Höchstzeit anzugeben, während der nach dem Meldungsbeginn die Meldung empfangen werden darf. Ein typischer Wert für einen Meldungs-Timer ist ungefähr 1,5 mal die Zeit, die zum Empfang der längstmöglichen Meldung bei der eingestellten Baudrate erforderlich ist.

   Sie können den Meldungs-Timer zusammen mit der Endezeichenerkennung und mit dem Zählwert für die Höchstzeichenzahl verwenden, um eine Meldung zu beenden.

   Einrichtung: c/m = 1, tmr = 1, SMW92/SMW192 = Timeout in Millisekunden

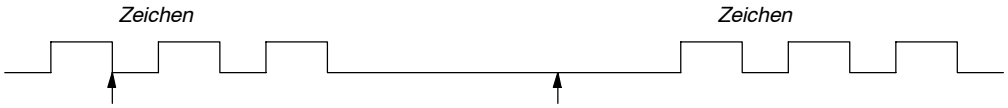


Bild 6-13 Beenden der Operation Meldung in Zwischenspeicher empfangen mit dem Meldungs-Timer

4. *Zählwert für die Höchstzeichenzahl:* Der Operation Meldung in Zwischenspeicher empfangen muss die maximale Anzahl Zeichen mitgeteilt werden, die empfangen werden soll (SMB94 oder SMB194). Wenn dieser Wert erreicht oder überschritten wird, wird die Funktion zum Meldungsempfang beendet. Für die Operation Meldung in Zwischenspeicher empfangen muss eine maximale Zeichenzahl angegeben werden, auch wenn dieser Wert nicht unbedingt als Endebedingung genutzt wird. Dies ist deshalb erforderlich, weil die Operation Meldung in Zwischenspeicher empfangen die maximale Größe der empfangenen Meldung kennen muss, damit Anwenderdaten, die sich im Anschluss an den Meldungspuffer befinden, nicht überschrieben werden.

   Der Zählwert für die Höchstzeichenzahl kann zum Beenden von Meldungen eingesetzt werden, wenn beim Protokoll die Meldungslänge bekannt und immer gleich ist. Der Zählwert für die Höchstzeichenzahl wird immer zusammen mit der Endezeichenerkennung, mit dem Timer für die Zeit zwischen den Zeichen oder mit dem Meldungs-Timer verwendet.

5. *Paritätsfehler:* Die Operation Meldung in Zwischenspeicher empfangen wird automatisch beendet, wenn die Hardware einen Paritätsfehler an einem empfangenen Zeichen meldet. Paritätsfehler sind nur möglich, wenn in SMB30 oder SMB130 die Parität aktiviert ist. Diese Funktion kann nicht ausgeschaltet werden.

6. *Anwenderabbruch:* Das Anwenderprogramm kann die Funktion zum Meldungsempfang beenden, indem eine weitere Operation Meldung in Zwischenspeicher empfangen ausgeführt wird und das Freigabebit (EN) in SMB87 oder SMB187 auf Null gesetzt ist. Daraufhin wird die Funktion zum Meldungsempfang sofort beendet.

Interruptgesteuerter Empfang von Daten

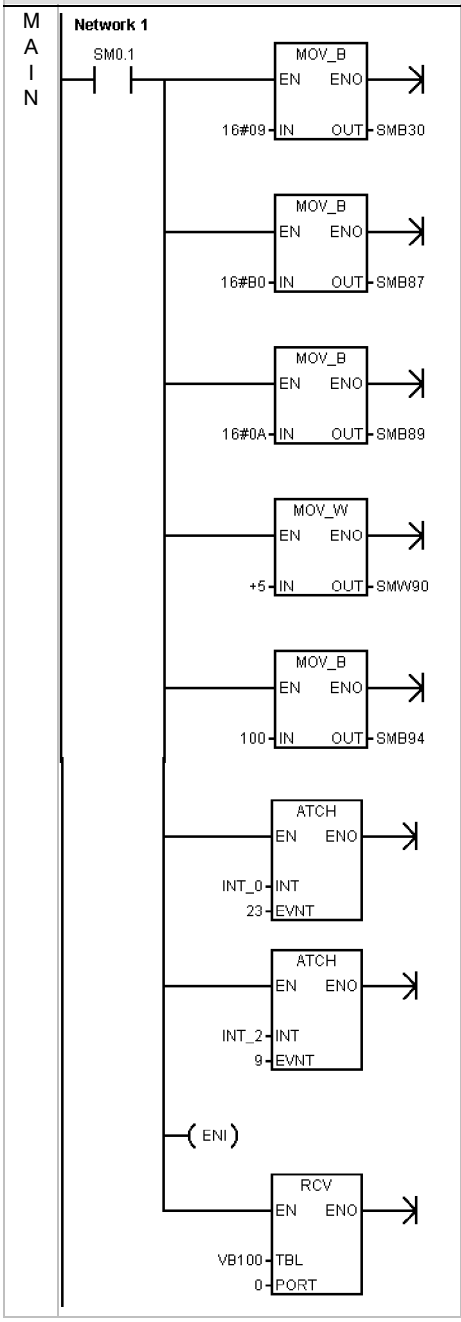
Damit Sie bei den unterstützten Protokollen eine größere Flexibilität haben, können Sie auch interruptgesteuert Daten empfangen. Hierbei erzeugt jedes empfangene Zeichen einen Interrupt. Das empfangene Zeichen wird in SMB2 abgelegt und der Status der Parität (sofern aktiviert) wird in SM3.0 abgelegt. 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 dem Paritätsbit können Sie die Meldung verwerfen oder eine negative Quittierung erzeugen.

Wenn Sie die Interruptsteuerung bei hohen Baudraten (38,4 kBaud bis 115,2 kBaud) verwenden, ist die Zeit zwischen den Interrupts sehr kurz. Der Interrupt für 38,4 kBaud beträgt beispielsweise 260 Mikrosekunden, der Interrupt für 57,6 kBaud beträgt 173 Mikrosekunden und der Interrupt für 115,2 kBaud beträgt 86 Mikrosekunden. Stellen Sie sicher, dass Sie die Interruptprogramme sehr kurz halten, damit keine Zeichen verloren gehen, oder verwenden Sie die Operation Meldung in Zwischenspeicher empfangen.

Tipp

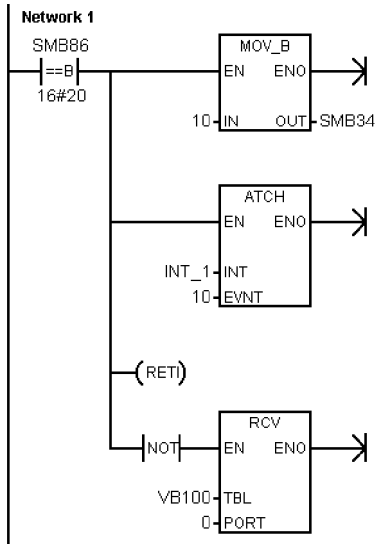
SMB2 und SMB3 werden von Schnittstelle 0 und Schnittstelle 1 gemeinsam genutzt. Löst der Empfang eines Zeichens an Schnittstelle 0 die Ausführung des Interruptprogramms aus, das dem Ereignis (Interruptereignis 8) zugeordnet ist, dann enthält SMB2 das an Schnittstelle 0 empfangene Zeichen und SMB3 enthält den Paritätsstatus des Zeichens. Löst der Empfang eines Zeichens an Schnittstelle 1 die Ausführung des Interruptprogramms aus, das dem Ereignis (Interruptereignis 25) zugeordnet ist, dann enthält SMB2 das an Schnittstelle 1 empfangene Zeichen und SMB3 enthält den Paritätsstatus des Zeichens.

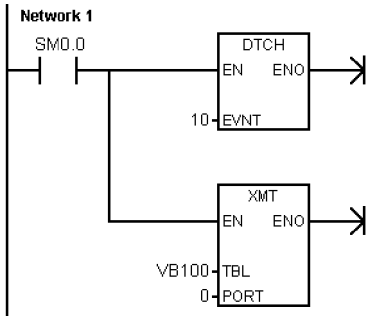
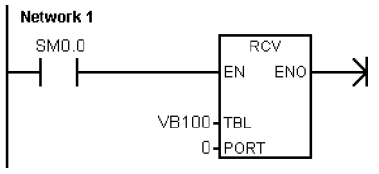
Beispiel: Operationen Meldung aus Zwischenspeicher übertragen und Meldung in Zwischenspeicher empfangen




```
Netzwerk 1      //Das Programm empfängt eine Zeichenkette, bis
                //das Zeichen für Zeilenschaltung empfangen wird.
                //Daraufhin wird die Meldung an den Sender zurückgeschickt.
LD      SM0.1           //Im ersten Zyklus:
MOVB    16#09, SMB30    //1.  Frei programmierbare Kommunikation initiali
                        //sieren:
                        //    - 9600 Baud wählen.
                        //    - 8 Datenbits wählen.
                        //    - Keine Parität wählen.
MOVB    16#B0, SMB87    //2.  Steuerbyte für Meldungsempfang initialisieren:
                        //    - Meldungsempfang freigeben.
                        //    - Zeichen für Meldungsende erkennen.
                        //    - Leerlaufbedingung als Bedingung für Meldungs
                        //beginn
                        //      erkennen.
MOVB    16#0A, SMB89    //3.  Zeichen für Meldungsende
                        //    auf Hex OA (Zeilenschaltung) setzen.
MOVW    +5, SMW90       //4.  Timeout für Leerlauf auf
                        //    5 ms setzen.
MOVB    100, SMB94      //5.  Maximale Zeichenzahl auf
                        //    100 setzen.
ATCH    INT_0, 23       //6.  Interrupt 0
                        //    dem Ereignis Empfang beendet zuordnen.
ATCH    INT_2, 9        //7.  Interrupt 2
                        //    dem Ereignis Senden beendet zuordnen.
ENI                     //8.  Anwenderinterrupts freigeben.
RCV     VB100, 0        //9.  Empfangsbox mit Puffer an VB100 freigeben.
```

Beispiel: Operationen Meldung aus Zwischenspeicher übertragen und Meldung in Zwischenspeicher empfangen, Fortsetzung

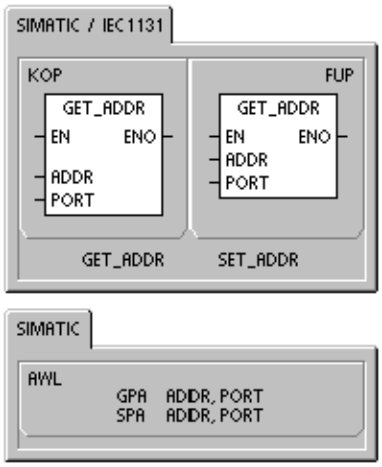
INT 0		Netzwerk 1 //Interruptprogramm für Empfang vollständig:
		//1. Zeigt der Empfangsstatus den Empfang des Endezeichens,
		// dann eine 10-ms-Zeit zuweisen, um Senden und Rückkehr
		// auszulösen.
		//2. Wurde der Empfang aus anderen Gründen beendet,
		// dann neuen Empfang starten.
		LDB = SMB86, 16#20
		MOVB 10, SMB34
		ATCH INT_1, 10
		CRETI
		NOT
		RCV VB100, 0

INT 1		Netzwerk 1 //Zeitgesteuerter Interrupt von 10 ms: //1. Zeitgesteuerten Interrupt trennen //2. Meldung zurück zum Anwender senden. LD SM0.0 DTCH 10 XMT VB100, 0
INT 2		Netzwerk 1 //Interrupt Senden beendet: //Neuen Empfang freigeben. LD SM0.0 RCV VB100, 0

Operationen Schnittstellenadresse holen und Schnittstellenadresse einstellen

Die Operation Schnittstellenadresse holen (GPA) liest die Stationsadresse der S7-200 CPU-Schnittstelle, die in PORT angegeben wird, und legt den Wert in der von ADDR angegebenen Adresse ab.

Die Operation Schnittstellenadresse einstellen (SPA) stellt die Stationsadresse der Schnittstelle (PORT) auf den in ADDR angegebenen Wert ein. Die neue Adresse wird nicht dauerhaft gespeichert. Nach einem Neustart wird die entsprechende Schnittstelle wieder auf die letzte Adresse zurückgesetzt (auf die Adresse, die mit dem Systemdatenbaustein geladen wurde).



Fehlerbedingungen, die ENO = 0 setzen:

- 0006 (Indirekte Adresse)
- 0004 (Operation Schnittstellenadresse einstellen in Interruptprogramm)

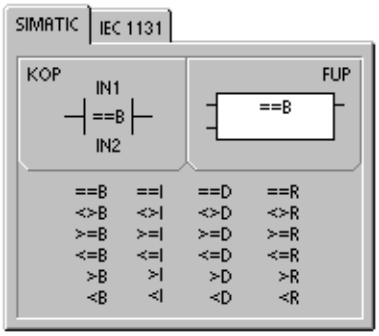
6

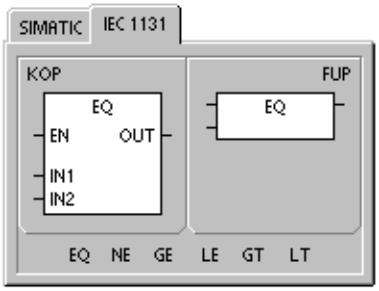
Tabelle 6-13 Gültige Operanden für die Operationen Schnittstellenadresse holen und Schnittstellenadresse einstellen

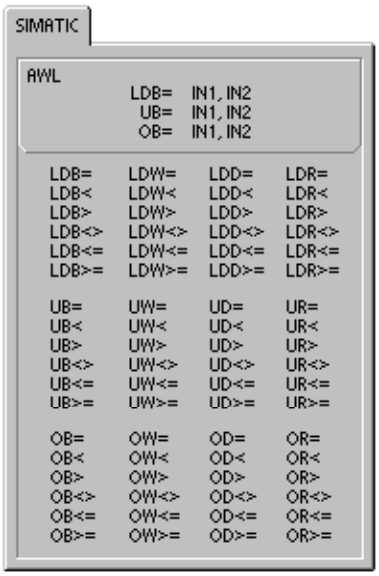
Eingänge/Ausgänge	Datentyp	Operanden		
ADDR	BYTE	EB, AB, VB, MB, SMB, SB, LB, AC, *VD, *LD, *AC, Konstante *(Konstanten sind nur gültig für die Operation Schnittstellenadresse einstellen.)*		
PORT	BYTE	Konstante	*für CPU 221, CPU 222, CPU 224:* *für CPU 226 und CPU 226XM:*	0 0 oder 1

Vergleichsoperationen

Vergleich von numerischen Werten







Die Vergleichsoperationen vergleichen zwei Werte miteinander:

IN1 = IN2 IN1 >= IN2 IN1 <= IN2
IN1 > IN2 IN1 < IN2 IN1 <> IN2

Bytevergleiche haben kein Vorzeichen.
Ganzzahlenvergleiche haben ein Vorzeichen.
Doppelwortvergleiche haben ein Vorzeichen.
Realzahlenvergleiche haben ein Vorzeichen.

In KOP und FUP: Ist der Vergleich wahr, dann schaltet die Vergleichsoperation den Kontakt (KOP) bzw. den Ausgang (FUP) ein.

In AWL: Ist der Vergleich wahr, dann lädt die Vergleichsoperation den Wert 1 in die Spitze des Stack bzw. sie verknüpft den Wert 1 mit der Spitze des Stack durch UND oder ODER (AWL).

Wenn Sie mit den IEC-Vergleichsoperationen arbeiten, können Sie unterschiedliche Datentypen für die Eingänge verwenden. Beide Eingangswerte müssen jedoch vom gleichen Datentyp sein.

Hinweis

Bei den folgenden Bedingungen handelt es sich um schwere Fehler, die bewirken, dass die S7-200 die Programmbearbeitung sofort stoppt:

- Unzulässige indirekte Adresse erkannt (in beliebiger Vergleichsoperation)
- Unzulässige Realzahl (z.B. NAN) erkannt (Realzahlenvergleich)

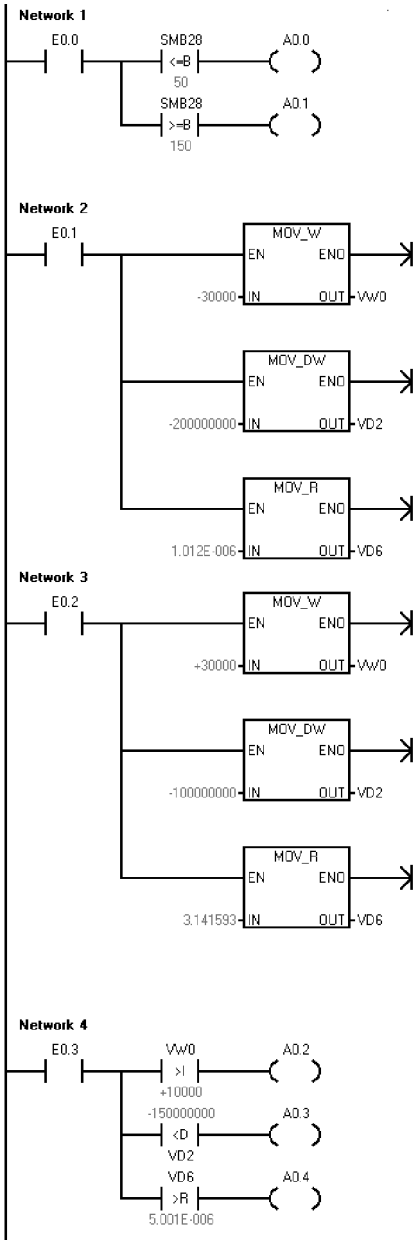
Damit diese Fehlerzustände nicht auftreten, müssen Sie darauf achten, dass Sie alle Pointer und Werte, die Realzahlen enthalten, vor dem Ausführen der Vergleichsoperationen, die diese Werte verwenden, richtig initialisieren.

Vergleichsoperationen werden unabhängig vom Zustand des Signalflusses ausgeführt.

Tabelle 6-14 Gültige Operanden für die Vergleichsoperationen

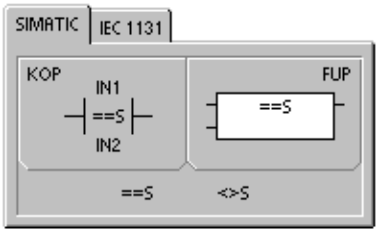
Eingänge/ Ausgänge	Typ	Operanden
IN1, IN2	BYTE	EB, AB, VB, MB, SMB, SB, LB, AC, *VD, *LD, *AC, Konstante
	INT	EW, AW, VW, MW, SMW, SW, T, Z, LW, AC, AEW, *VD, *LD, *AC, Konstante
	DINT	ED, AD, VD, MD, SMD, SD, LD, AC, HC, *VD, *LD, *AC, Konstante
	REAL	ED, AD, VD, MD, SMD, SD, LD, AC, *VD, *LD, *AC, Konstante
Ausgang (oder OUT)	BOOL	E, A, V, M, SM, S, T, Z, L, Signalfluss

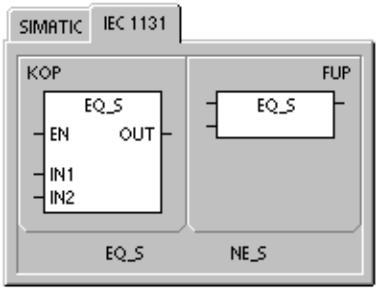
Beispiel: Vergleichsoperationen

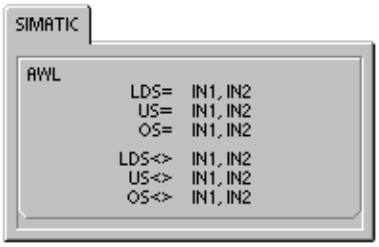



```
Netzwerk 1      //Analogpotentiometer 0 drehen, um den
                //Bytewert von SMB28 zu verändern.
                //A0.0 ist aktiv, wenn der Wert von SMB28 kleiner als oder
                //gleich 50 ist.
                //A0.1 ist aktiv, wenn der Wert von SMB28 größer als oder
                //gleich 150 ist.
                //Die Statusanzeige ist eingeschaltet, wenn der Vergleich
                //wahr ist.
LD      E0.0
LPS
UB<=    SMB28, 50
=       A0.0
LPP
UB>=    SMB28, 150
=       A0.1

Netzwerk 2      //Adressen aus Variablenspeicher mit geringen Werten
                //laden,
                //so dass die Vergleiche falsch sind und
                //die Statusanzeigen ausgeschaltet werden.
LD      E0.1
MOVW    -30000, VW0
MOVD    -200000000, VD2
MOVR    1.012E-006, VD6

Netzwerk 3      //Adressen aus Variablenspeicher mit hohen Werten laden,

                //so dass die Vergleiche wahr sind und
                //die Statusanzeigen eingeschaltet werden.
LD      E0.2
MOVW    +30000, VW0
MOVD    -100000000, VD2
MOVR    3.141593, VD6

Netzwerk 4      //Der Ganzzahlenvergleich prüft, ob
                //VW0 > +10000 wahr ist.
                //Mit Programmkonstanten werden die unterschiedlichen
                //Datentypen gezeigt. Sie können auch zwei Werte verglei
                //chen,
                //die im programmierbaren Speicher abgelegt sind, z.B.:
                //VW0 > VW100.
LD      E0.3
LPS
UW>     VW0, +10000
=       A0.2
LRD
UD<     -150000000, VD2
=       A0.3
LPP
UR>     VD6, 5.001E-006
=       A0.4
```

Zeichenkettenvergleich







Die Operation Zeichenkettenvergleich vergleicht zwei Zeichenketten aus ASCII-Zeichen miteinander:

IN1 = IN2 IN1 <> IN2

Ist der Vergleich wahr, schaltet die Vergleichsoperation den Kontakt (KOP) bzw. den Ausgang (FUP) ein. In AWL lädt die Vergleichsoperation den Wert 1 in die Spitze des Stack bzw. sie verknüpft den Wert 1 mit der Spitze des Stack durch UND oder ODER.

Hinweis

Bei den folgenden Bedingungen handelt es sich um schwere Fehler, die bewirken, dass die S7-200 die Programmbearbeitung sofort stoppt:

- Unzulässige indirekte Adresse erkannt (in beliebiger Vergleichsoperation)
- Zeichenkette mit einer Länge von mehr als 254 Zeichen erkannt (Zeichenkettenvergleich)
- Zeichenkette, deren Anfangsadresse und Länge so angegeben sind, dass sie nicht in den angegebenen Speicherbereich passt (Zeichenkettenvergleich)

Damit diese Fehlerzustände nicht auftreten, müssen Sie darauf achten, dass Sie alle Pointer und Adressen im Speicher, die ASCII-Zeichenketten enthalten sollen, vor dem Ausführen der Vergleichsoperationen, die diese Werte verwenden, richtig initialisieren. Stellen Sie sicher, dass der für eine ASCII-Zeichenkette reservierte Puffer komplett in den angegebenen Speicherbereich passt.

Vergleichsoperationen werden unabhängig vom Zustand des Signalflusses ausgeführt.

Tabelle 6-15 Gültige Operanden für die Operation Zeichenkettenvergleich

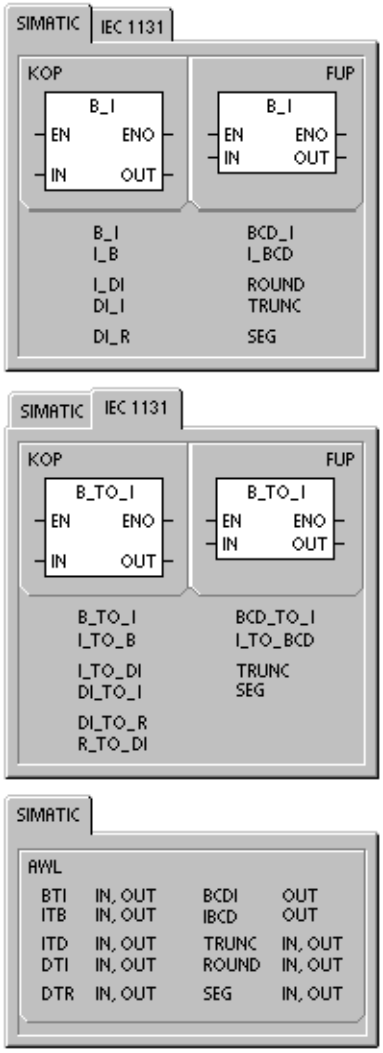
Eingänge/ Ausgänge	Typ	Operanden
IN1, IN2	BYTE (Zeichenkette)	VB, LB, *VD, *LD, *AC
Ausgang (OUT)	BOOL	E, A, V, M, SM, S, T, Z, L, Signalfluss

Umwandlungsoperationen

Genormte Umwandlungsoperationen

Numerische Umwandlungen

Die Operationen Byte in ganze Zahl wandeln (BTI), Ganze Zahl in Byte wandeln (ITB), Ganze Zahl (16 Bit) in ganze Zahl (32 Bit) wandeln (ITD), Ganze Zahl (32 Bit) in ganze Zahl (16 Bit) wandeln (DTI), Ganze Zahl (32 Bit) in Realzahl wandeln (DTR), BCD in ganze Zahl wandeln (BCDI) und Ganze Zahl in BCD wandeln (IBCD) wandeln einen Eingangswert IN in das angegebene Format um und speichern den Ausgangswert in der von OUT angegebenen Adresse im Speicher. Sie können beispielsweise eine ganze Zahl (32 Bit) in eine Realzahl wandeln. Sie können auch Ganzzahlen- und BCD-Formate umwandeln.



Zahl runden und Realzahl in ganze Zahl (32 Bit) wandeln

Die Operation Zahl runden (ROUND) wandelt eine Realzahl (IN) in einen ganzzahligen Wert (32 Bit) um und lädt das gerundete Ergebnis in die von OUT angegebene Variable.

Die Operation Realzahl in ganze Zahl (32 Bit) wandeln (TRUNC) wandelt eine Realzahl (IN) in einen ganzzahligen Wert (32 Bit) um und lädt den ganzzahligen Teil des Ergebnisses in die von OUT angegebene Variable.

Bitmuster für Sieben-Segment-Anzeige erzeugen

Mit der Operation Bitmuster für Sieben-Segment-Anzeige erzeugen (SEG) können Sie ein Bitmuster erzeugen, das die Segmente einer siebenteiligen Anzeige beleuchtet.

Tabelle 6-16 Gültige Operanden für die genormten Umwandlungsoperationen

Eingänge/ Ausgänge	Datentyp	Operanden
IN	BYTE	EB, AB, VB, MB, SMB, SB, LB, AC, *VD, *LD, *AC, Konstante
	WORD, INT	EW, AW, VW, MW, SMW, SW, T, Z, LW, AEW, AC, *VD, *LD, *AC, Konstante
	DINT	ED, AD, VD, MD, SMD, SD, LD, HC, AC, *VD, *LD, *AC, Konstante
	REAL	ED, AD, VD, MD, SMD, SD, LD, AC, *VD, *LD, *AC, Konstante
OUT	BYTE	EB, AB, VB, MB, SMB, SB, LB, AC, *VD, *LD, *AC
	WORD, INT	EW, AW, VW, MW, SMW, SW, T, Z, LW, AC, *VD, *LD, *AC
	DINT, REAL	ED, AD, VD, MD, SMD, SD, LD, AC, *VD, *LD, *AC

Funktionsweise der Operationen BCD in ganze Zahl wandeln und Ganze Zahl in BCD wandeln

Die Operation BCD in ganze Zahl wandeln (BCDI) wandelt einen binärcodierten Dezimalwert IN in einen ganzzahligen Wert um und lädt das Ergebnis in die von OUT angegebene Variable. Der gültige Bereich für IN liegt zwischen 0 und 9999 (BCD).

Die Operation BCD in ganze Zahl wandeln (IBCD) wandelt einen binärcodierten Dezimalwert IN in einen ganzzahligen Wert um und lädt das Ergebnis in die von OUT angegebene Variable. Der gültige Bereich für IN liegt zwischen 0 und 9999 (ganze Zahl).

Fehlerbedingungen, die ENO = 0 setzen

- SM1.6 (ungültiger BCD-Wert)
- 0006 (Indirekte Adresse)

Betroffene Sondermerker:

- SM1.6 (ungültiger BCD-Wert)

Funktionsweise der Operation Ganze Zahl (32 Bit) in Realzahl wandeln

Die Operation Ganze Zahl (32 Bit) in Realzahl wandeln (DTR) wandelt eine ganze Zahl (32 Bit) mit Vorzeichen IN in eine Realzahl (32 Bit) um und lädt das Ergebnis in die von OUT angegebene Variable.

Fehlerbedingungen, die ENO = 0 setzen

- 0006 (Indirekte Adresse)

Funktionsweise der Operation Ganze Zahl (32 Bit) in ganze Zahl (16 Bit) wandeln

Die Operation Ganze Zahl (32 Bit) in ganze Zahl (16 Bit) wandeln (DTI) wandelt einen ganzzahligen Wert (32 Bit) IN in einen ganzzahligen Wert (16 Bit) um und lädt das Ergebnis in die von OUT angegebene Variable.

Ist der umgewandelte Wert zu groß, um im Ausgang dargestellt zu werden, wird das Überlaufbit gesetzt und der Ausgang nicht verändert.

Fehlerbedingungen, die ENO = 0 setzen

- SM1.1 (Überlauf)
- 0006 (Indirekte Adresse)

Betroffene Sondermerker:

- SM1.1 (Überlauf)

Funktionsweise der Operation Ganze Zahl (16 Bit) in ganze Zahl (32 Bit) wandeln

Die Operation Ganze Zahl (16 Bit) in ganze Zahl (32 Bit) wandeln (ITD) wandelt einen ganzzahligen Wert (16 Bit) IN in einen ganzzahligen Wert (32 Bit) um und lädt das Ergebnis in die von OUT angegebene Variable. Das Vorzeichen wird erweitert.

Fehlerbedingungen, die ENO = 0 setzen

- 0006 (Indirekte Adresse)

Funktionsweise der Operation Byte in ganze Zahl wandeln

Die Operation Byte in ganze Zahl wandeln (BTI) wandelt einen Bytewert IN in einen ganzzahligen Wert um und lädt das Ergebnis in die von OUT angegebene Variable. Das Byte ist vorzeichenlos, deswegen gibt es keine Vorzeichenerweiterung.

Fehlerbedingungen, die ENO = 0 setzen

- 0006 (Indirekte Adresse)

Funktionsweise der Operation Ganze Zahl in Byte wandeln

Die Operation Ganze Zahl in Byte wandeln (ITB) wandelt einen Wortwert IN in einen Bytewert um und lädt das Ergebnis in die von OUT angegebene Variable. Es werden die Werte 0 bis 255 umgewandelt. Alle anderen Werte liefern Überlauf und der Ausgang wird nicht verändert.

Fehlerbedingungen, die ENO = 0 setzen

- SM1.1 (Überlauf)
- 0006 (Indirekte Adresse)

Betroffene Sondermerker:

- SM1.1 (Überlauf)

Tipp

Wenn Sie eine ganze Zahl in eine Realzahl umwandeln möchten, verwenden Sie die Operation Ganze Zahl (16 Bit) in ganze Zahl (32 Bit) wandeln und dann die Operation Ganze Zahl (32 Bit) in Realzahl wandeln.

Funktionsweise der Operationen Zahl runden und Realzahl in ganze Zahl (32 Bit) wandeln

Die Operation Zahl runden (ROUND) wandelt eine Realzahl IN in einen ganzzahligen Wert (32 Bit) um und lädt das Ergebnis in die von OUT angegebene Variable. Ist der gebrochene Anteil 0,5 oder höher, wird die Zahl aufgerundet.

Die Operation Realzahl in ganze Zahl (32 Bit) wandeln (TRUNC) wandelt eine Realzahl IN in einen ganzzahligen Wert (32 Bit) mit Vorzeichen um und lädt das Ergebnis in die von OUT angegebene Variable. Nur der ganzzahlige Teil der Realzahl wird umgewandelt (und der Rest wird verworfen).

Fehlerbedingungen, die ENO = 0 setzen

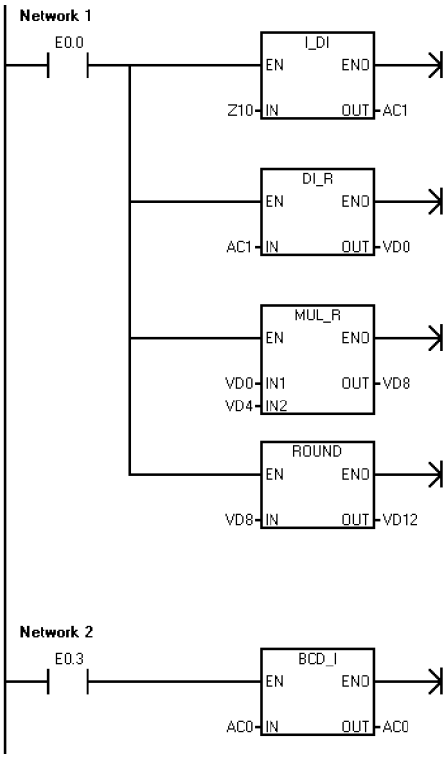
- SM1.1 (Überlauf)
- 0006 (Indirekte Adresse)

Betroffene Sondermerker:

- SM1.1 (Überlauf)

Ist der Wert, den Sie umwandeln möchten, keine gültige Realzahl oder zu groß, um im Ausgang dargestellt zu werden, wird das Überlaufbit gesetzt und der Ausgang nicht verändert.

Beispiel: Genormte Umwandlungsoperationen




```
Netzwerk 1      //Zoll in Zentimeter umwandeln:
                //1. Zählerwert (Zoll) in Akkumulator 1 laden.
                //2. Wert in Realzahl wandeln.
                //3. Mit 2,54 multiplizieren (um in Zentimeter umzuwan
                //deln).
                //4. Wert zurück in ganze Zahl wandeln.
LD      E0.0
ITD     Z10, AC1
DTR     AC1, VD0
MOVR    VD0, VD8
*R      VD4, VD8
ROUND   VD8, VD12

Netzwerk 2      //BCD-Wert in ganze Zahl wandeln.
LD      E0.3
BCDI    AC0
```

Ganze Zahl (32 Bit) in Realzahl wandeln und Zahl runden

Z10	101	Zählwert = 101 Zoll
VD0	101.0	Zählwert (als Realzahl)
VD4	2.54	Konstante 2,54 (Zoll in Zentimeter)
VD8	256.54	256,54 Zentimeter als Realzahl
VD12	257	257 cm als ganze Zahl (32 Bit)

BCD in ganze Zahl wandeln

AC0	1234
	BCDI
AC0	04D2

Funktionsweise der Operation Bitmuster für Sieben-Segment-Anzeige erzeugen

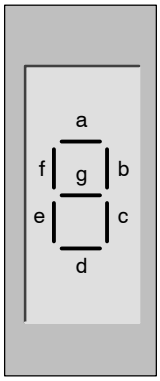
Zum Beleuchten der Segmente einer siebenteiligen Anzeige wandelt die Operation Bitmuster für Sieben-Segment-Anzeige erzeugen (SEG) das in IN angegebenen Zeichen (Byte) in ein Bitmuster (Byte) um, das in der von OUT angegebenen Adresse abgelegt wird.

Die beleuchteten Segmente stellen das Zeichen in der niederwertigsten Ziffer des Eingangsbytes dar. Bild 6-14 zeigt die Codierung, die von der Operation für die Beleuchtung einer Sieben-Segment-Anzeige verwendet wird.

Fehlerbedingungen, die ENO = 0 setzen

- 0006 (Indirekte Adresse)

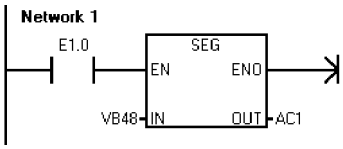
(IN) LSD	Segment-Anzeige	(OUT) - g f e d c b a
0		0 0 1 1 1 1 1 1
1		0 0 0 0 0 1 1 0
2		0 1 0 1 1 0 1 1
3		0 1 0 0 1 1 1 1
4		0 1 1 0 0 1 1 0
5		0 1 1 0 1 1 0 1
6		0 1 1 1 1 1 0 1
7		0 0 0 0 0 1 1 1



(IN) LSD	Segment-Anzeige	(OUT) - g f e d c b a
8		0 1 1 1 1 1 1 1
9		0 1 1 0 0 1 1 1
U		0 1 1 1 0 1 1 1
B		0 1 1 1 1 1 0 0
C		0 0 1 1 1 0 0 1
D		0 1 0 1 1 1 1 0
E		0 1 1 1 1 0 0 1
F		0 1 1 1 0 0 0 1

Bild 6-14 Codierung einer Sieben-Segment-Anzeige

Beispiel: Operation Bitmuster für Sieben-Segment-Anzeige erzeugen



Netzwerk 1

```
LD      E1.0
SEG     VB48, AC1
```

05 SEG 6D

VB48 AC1

(Zeichen in der Anzeige)

ASCII-Umwandlungsoperationen

Zulässige ASCII-Zeichen sind die Hexadezimalwerte 30 bis 39 und 41 bis 46.

Umwandeln zwischen ASCII- und Hexadezimalwerten

Die Operation ASCII-Zeichenkette in Hexadezimalzahl wandeln (ATH) wandelt eine Anzahl ASCII-Zeichen, die bei IN beginnt, in Hexadezimalziffern um, die an OUT beginnen. Die Operation Hexadezimalzahl in ASCII-Zeichenkette wandeln (HTA) wandelt die Hexadezimalziffern, die an Eingangsbyte IN beginnen, in ASCII-Zeichen um, die an OUT beginnen. Wieviele Hexadezimalziffern umgewandelt werden sollen, wird durch die Länge (LEN) angegeben.

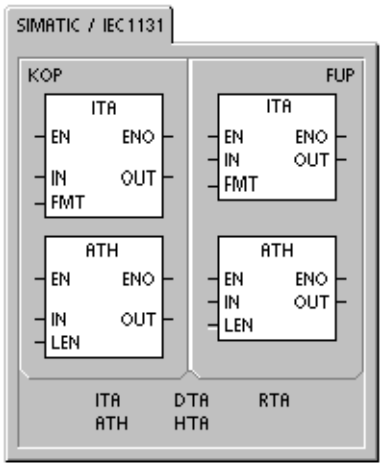
Die maximale Anzahl ASCII-Zeichen oder Hexadezimalziffern, die umgewandelt werden kann, beträgt 255.

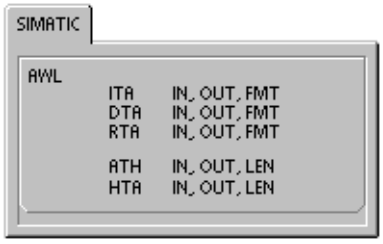
Fehlerbedingungen, die ENO = 0 setzen

- SM1.7 (unzulässige ASCII-Zeichen) *nur bei der Operation ASCII-Zeichenkette in Hexadezimalzahl wandeln*
- 0006 (Indirekte Adresse)
- 0091 (Operand außerhalb des Bereichs)

Betroffene Sondermerker:

- SM1.7 (unzulässiges ASCII-Zeichen)





Umwandeln von numerischen Werten in ASCII

Die Operationen Ganze Zahl in ASCII-Zeichenkette wandeln (ITA), Ganze Zahl (32 Bit) in ASCII-Zeichenkette wandeln (DTA) und Realzahl in ASCII-Zeichenkette wandeln (RTA) wandeln eine ganze Zahl (16 Bit), eine ganze Zahl (32 Bit) oder eine Realzahl in ASCII-Zeichen um.

Tabelle 6-17 Gültige Operanden für die ASCII-Umwandlungsoperationen

Eingänge/ Ausgänge	Datentyp	Operanden
IN	BYTE	EB, AB, VB, MB, SMB, SB, LB, *VD, *LD, *AC
	INT	EW, AW, VW, MW, SMW, SW, T, Z, LW, AC, AEW, *VD, *LD, *AC, Konstante
	DINT	ED, AD, VD, MD, SMD, SD, LD, AC, HC, *VD, *LD, *AC, Konstante
	REAL	ED, AD, VD, MD, SMD, SD, LD, AC, *VD, *LD, *AC, Konstante
LEN, FMT	BYTE	EB, AB, VB, MB, SMB, SB, LB, AC, *VD, *LD, *AC, Konstante
OUT	BYTE	EB, AB, VB, MB, SMB, SB, LB, *VD, *LD, *AC

Funktionsweise der Operation Ganze Zahl in ASCII-Zeichenkette wandeln

Die Operation Ganze Zahl in ASCII-Zeichenkette wandeln (ITA) wandelt eine ganze Zahl (IN) in ein Feld aus ASCII-Zeichen um. Das Format (FMT) gibt die Umwandlungsgenauigkeit rechts von der Dezimalzahl an und ob der Dezimalpunkt als Punkt oder als Komma dargestellt wird. Das Ergebnis der Umwandlung wird in 8 aufeinanderfolgende Bytes mit Beginn an OUT abgelegt.

Fehlerbedingungen, die ENO = 0 setzen

- 0006 (Indirekte Adresse)
- Unzulässiges Format
- *nnn* > 5

Das Feld aus ASCII-Zeichen umfasst immer 8 Zeichen.

Bild 6-15 beschreibt den Formatoperanden der Operation Ganze Zahl in ASCII-Zeichenkette wandeln. Die Größe des Ausgabepuffers ist immer 8 Bytes. Die Anzahl der Ziffern rechts vom Dezimalpunkt im Ausgabepuffer wird vom Feld *nnn* angegeben. Der gültige Bereich für das Feld *nnn* liegt zwischen 0 und 5. Werden 0 Ziffern rechts vom Dezimalpunkt angegeben, wird der Wert ohne Dezimalpunkt angezeigt. Bei Werten für *nnn* größer als 5, wird der Ausgabepuffer mit ASCII-Leerzeichen belegt. Das Bit *c* gibt an, ob ein Komma (c = 1) oder ein Dezimalpunkt (c = 0) als Trennzeichen zwischen der ganzen Zahl und dem Bruch verwendet werden soll. Die oberen 4 Bits müssen Null sein.

Bild 6-15 zeigt Beispiele für Werte, die mit Dezimalpunkt (c = 0) und mit drei Ziffern rechts vom Dezimalpunkt (nnn = 011) formatiert wurden. Der Ausgabepuffer wird entsprechend den folgenden Richtlinien formatiert:

- ❏ Positive Werte werden ohne Vorzeichen in den Ausgabepuffer geschrieben.
- ❏ Negative Werte werden mit einem Minuszeichen (-) vor dem Wert in den Ausgabepuffer geschrieben.
- ❏ Führende Nullen links vom Dezimalpunkt (mit Ausnahme der Ziffer direkt neben dem Dezimalpunkt) werden unterdrückt.
- ❏ Die Werte werden rechtsbündig im Ausgabepuffer angeordnet.

FMT

MSB 7	6	5	4	3	2	1	LSB 0
0	0	0	0	c	n	n	n

c = Komma (1) oder Dezimalpunkt (0)
nnn = Ziffern rechts vom Dezimalpunkt

	OUT	OUT +1	OUT +2	OUT +3	OUT +4	OUT +5	OUT +6	OUT +7
IN = 12				0	.	0	1	2
in = -123			-	0	.	1	2	3
IN = 1234				1	.	2	3	4
IN = -12345		-	1	2	.	3	4	5

Bild 6-15 Operand FMT der Operation Ganze Zahl in ASCII-Zeichenkette wandeln (ITA)

Funktionsweise der Operation Ganze Zahl (32 Bit) in ASCII-Zeichenkette wandeln

Die Operation Ganze Zahl (32 Bit) in ASCII-Zeichenkette wandeln (DTA) wandelt eine ganze Zahl (32 Bit) (IN) in ein Feld aus ASCII-Zeichen um. Der Formatoperand (FMT) gibt die Umwandlungsgenauigkeit rechts vom Dezimalpunkt an. Das Ergebnis der Umwandlung wird in 12 aufeinanderfolgende Bytes mit Beginn an OUT abgelegt.

Fehlerbedingungen, die ENO = 0 setzen

- 0006 (Indirekte Adresse)
- Unzulässiges Format
- *nnn* > 5

Die Größe des Ausgabepuffers ist immer 12 Bytes.

Bild 6-16 beschreibt den Formatoperanden der Operation Ganze Zahl (32 Bit) in ASCII-Zeichenkette wandeln. Die Anzahl der Ziffern rechts vom Dezimalpunkt im Ausgabepuffer wird vom Feld *nnn* angegeben. Der gültige Bereich für das Feld *nnn* liegt zwischen 0 und 5. Werden 0 Ziffern rechts vom Dezimalpunkt angegeben, wird der Wert ohne Dezimalpunkt angezeigt. Bei Werten für *nnn* größer als 5, wird der Ausgabepuffer mit ASCII-Leerzeichen belegt. Das Bit *c* gibt an, ob ein Komma (c = 1) oder ein Dezimalpunkt (c = 0) als Trennzeichen zwischen der ganzen Zahl und dem Bruch verwendet werden soll. Die oberen 4 Bits müssen Null sein.

Bild 6-16 zeigt Beispiele für Werte, die mit Dezimalpunkt (c = 0) und mit vier Ziffern rechts vom Dezimalpunkt (nnn = 100) formatiert wurden. Der Ausgabepuffer wird entsprechend den folgenden Richtlinien formatiert:

- ❏ Positive Werte werden ohne Vorzeichen in den Ausgabepuffer geschrieben.
- ❏ Negative Werte werden mit einem Minuszeichen (-) vor dem Wert in den Ausgabepuffer geschrieben.
- ❏ Führende Nullen links vom Dezimalpunkt (mit Ausnahme der Ziffer direkt neben dem Dezimalpunkt) werden unterdrückt.
- ❏ Die Werte werden rechtsbündig im Ausgabepuffer angeordnet.

FMT

MSB 7	6	5	4	3	2	1	LSB 0
0	0	0	0	c	n	n	n

c = Komma (1) oder Dezimalpunkt (0)
nnn = Ziffern rechts vom Dezimalpunkt

	OUT	OUT +1	OUT +2	OUT +3	OUT +4	OUT +5	OUT +6	OUT +7	OUT +8	OUT +9	OUT +10	OUT +11
IN = -12						-	0	.	0	0	1	2
IN = 1234567					1	2	3	.	4	5	6	7

Bild 6-16 Operand FMT der Operation Ganze Zahl (32 Bit) in ASCII-Zeichenkette wandeln (DTA)

Funktionsweise der Operation Realzahl in ASCII-Zeichenkette wandeln

Die Operation Realzahl in ASCII-Zeichenkette wandeln (RTA) wandelt eine Realzahl IN in ASCII-Zeichen um. Das Format (FMT) gibt die Umwandlungsgenauigkeit rechts von der Dezimalzahl an, es gibt an, ob der Dezimalpunkt als Punkt oder als Komma dargestellt wird und es gibt die Größe des Ausgabepuffers an.

Das Ergebnis der Umwandlung wird in einen Ausgabepuffer mit Beginn an OUT abgelegt.

Fehlerbedingungen, die ENO = 0 setzen

- 0006 (Indirekte Adresse)
- nnn > 5
- ssss < 3
- ssss < Anzahl der Zeichen in OUT

Die Anzahl (oder Länge) der resultierenden ASCII-Zeichen entspricht der Größe des Ausgabepuffers und kann in einem Bereich zwischen 3 und 15 Bytes oder Zeichen angegeben werden.

Das von den S7-200 verwendete Realzahlenformat unterstützt maximal 7 signifikante Ziffern. Sollen mehr als 7 signifikante Ziffern angezeigt werden, wird ein Rundungsfehler ausgegeben.

Bild 6-17 beschreibt den Formatoperanden (FMT) der Operation RTA. Die Größe des Ausgabepuffers wird von dem Feld ssss angegeben. Die Größen 0, 1 oder 2 Bytes sind ungültig. Die Anzahl der Ziffern rechts vom Dezimalpunkt im Ausgabepuffer wird vom Feld nnn angegeben. Der gültige Bereich für das Feld nnn liegt zwischen 0 und 5. Werden 0 Ziffern rechts vom Dezimalpunkt angegeben, wird der Wert ohne Dezimalpunkt angezeigt. Der Ausgabepuffer wird mit ASCII-Leerzeichen belegt, wenn der Wert nnn größer als 5 ist oder wenn der angegebene Ausgabepuffer zu klein ist, um den umgewandelten Wert zu speichern. Das Bit c gibt an, ob ein Komma (c = 1) oder ein Dezimalpunkt (c = 0) als Trennzeichen zwischen der ganzen Zahl und dem Bruch verwendet werden soll.

Bild 6-17 zeigt Beispiele für Werte, die mit Dezimalpunkt (c = 0), mit einer Ziffer rechts vom Dezimalpunkt (nnn = 001) und einer Puffergröße von sechs Bytes (ssss = 0110) formatiert wurden. Der Ausgabepuffer wird entsprechend den folgenden Richtlinien formatiert:

- ❏ Positive Werte werden ohne Vorzeichen in den Ausgabepuffer geschrieben.
- ❏ Negative Werte werden mit einem Minuszeichen (-) vor dem Wert in den Ausgabepuffer geschrieben.
- ❏ Führende Nullen links vom Dezimalpunkt (mit Ausnahme der Ziffer direkt neben dem Dezimalpunkt) werden unterdrückt.
- ❏ Der Wert rechts vom Dezimalpunkt wird gerundet, damit er der Anzahl der Ziffern entspricht, die rechts vom Dezimalpunkt angegeben werden sollen.
- ❏ Die Größe des Ausgabepuffers muss mindestens drei Bytes mehr umfassen als die Anzahl der Ziffern rechts vom Dezimalpunkt.
- ❏ Die Werte werden rechtsbündig im Ausgabepuffer angeordnet.

FMT

MSB LSB

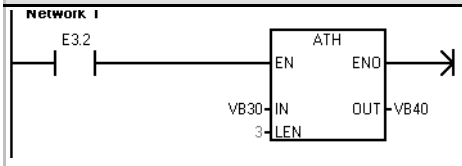
7	6	5	4	3	2	1	0
s	s	s	s	c	n	n	n

ssss = Größe des Ausgabepuffers
c = Komma (1) oder Dezimalpunkt (0)
nnn = Ziffern rechts vom Dezimalpunkt

	OUT	OUT +1	OUT +2	OUT +3	OUT +4	OUT +5
IN = 1234.5	1	2	3	4	.	5
IN = -0.0004				0	.	0
IN = -3.67526			-	3	.	7
IN = 1.95				2	.	0

Bild 6-17 Operand FMT der Operation Realzahl in ASCII-Zeichenkette wandeln (RTA)

Beispiel: Operation ASCII-Zeichenkette in Hexadezimalzahl wandeln



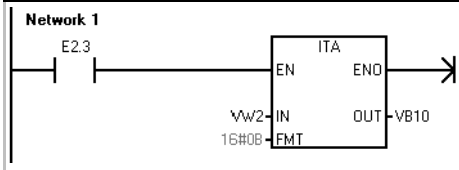
Netzwerk 1

LD E3.2
ATH VB30, VB40, 3

'3'	'E'	'A'			
33	45	41	ATH	3E	AX
VB40				VB40	

Hinweis:
Das X zeigt an, dass das Halb-Byte nicht verändert wurde.

Beispiel: Operation Ganze Zahl in ASCII-Zeichenkette wandeln

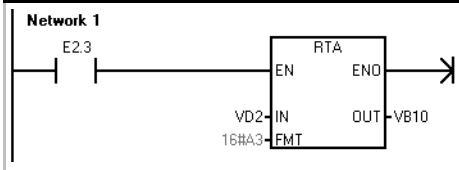


Netzwerk 1 //Ganzzahligen Wert aus VW2
//in 8 ASCII-Zeichen wandeln mit Beginn an VB10,
//mit Format von 16#0B
//(Komma statt Dezimalpunkt,
//gefolgt von 3 Ziffern).

LD E2.3
ITA VW2, VB10, 16#0B

		' '	' '	'1'	'2'	','	'3'	'4'	'5'
12345	ITA	20	20	31	32	2C	33	34	35
VW2		VB10	VB11	...					

Beispiel: Operation Realzahl in ASCII-Zeichenkette wandeln



Netzwerk 1 //Realzahl aus VD2
//in 10 ASCII-Zeichen wandeln mit Beginn an VB10,
//mit Format von 16#A3
//(Komma statt Dezimalpunkt,
//gefolgt von 3 Ziffern).

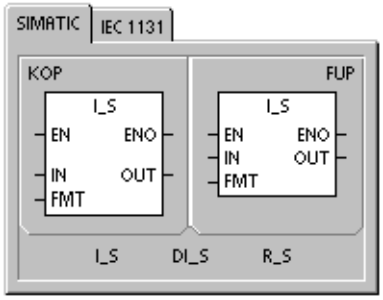
LD E2.3
RTA VD2, VB10, 16#A3

		' '	' '	' '	'1'	'2'	'3'	'.'	'4'	'5'	'0'
123.45	RTA	20	20	20	31	32	33	2E	34	35	30
VD2		VB10	VB11	...							

Zeichenketten-Umwandlungsoperationen

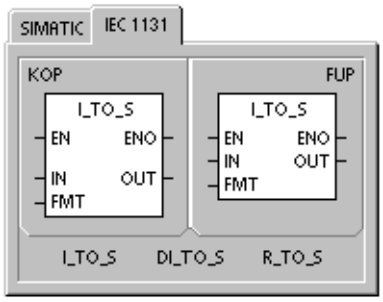
Umwandeln von numerischen Werten in Zeichenketten

Die Operationen Ganze Zahl in Zeichenkette wandeln (ITS), Ganze Zahl (32 Bit) in Zeichenkette wandeln (DTS) und Realzahl in Zeichenkette wandeln (RTS) wandeln eine ganze Zahl (16 Bit), eine ganze Zahl (32 Bit) oder eine Realzahl in eine ASCII-Zeichenkette (OUT) um.



Funktionsweise der Operation Ganze Zahl in Zeichenkette wandeln

Die Operation Ganze Zahl in Zeichenkette wandeln (ITS) wandelt eine ganze Zahl IN in eine ASCII-Zeichenkette mit einer Länge von 8 Zeichen um. Das Format (FMT) gibt die Umwandlungsgenauigkeit rechts von der Dezimalzahl an und ob der Dezimalpunkt als Punkt oder als Komma dargestellt wird. Die sich ergebende Zeichenkette wird in 9 aufeinander folgende Bytes mit Beginn an OUT geschrieben. Ausführliche Informationen zu Formaten von Zeichenketten finden Sie in Kapitel 4.



Fehlerbedingungen, die ENO = 0 setzen

- 0006 (Indirekte Adresse)
- 0091 (Operand außerhalb des Bereichs)
- Unzulässiges Format (nnn > 5)

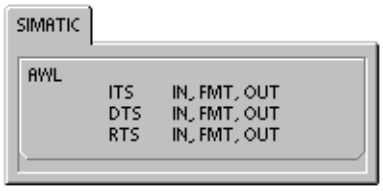


Bild 6-18 beschreibt den Formatoperanden der Operation Ganze Zahl in Zeichenkette wandeln. Die Länge der ausgegebenen Zeichenkette beträgt immer 8 Zeichen. Die Anzahl der Ziffern rechts vom Dezimalpunkt im Ausgabepuffer wird vom Feld nnn angegeben. Der gültige Bereich für das Feld nnn liegt zwischen 0 und 5. Werden 0 Ziffern rechts vom Dezimalpunkt angegeben, wird der Wert ohne Dezimalpunkt angezeigt. Bei Werten von nnn größer als 5 wird eine Zeichenkette aus 8 ASCII-Leerzeichen ausgegeben. Das Bit c gibt an, ob ein Komma (c = 1) oder ein Dezimalpunkt (c = 0) als Trennzeichen zwischen der ganzen Zahl und dem Bruch verwendet werden soll. Die oberen 4 Bits des Formats müssen Null sein.

Bild 6-18 zeigt auch Beispiele für Werte, die mit Dezimalpunkt (c = 0) und mit drei Ziffern rechts vom Dezimalpunkt (nnn = 011) formatiert wurden. Der Wert von OUT ist die Länge der Zeichenkette.

Die ausgegebene Zeichenkette wird entsprechend den folgenden Richtlinien formatiert:

- ❏ Positive Werte werden ohne Vorzeichen in den Ausgabepuffer geschrieben.
- ❏ Negative Werte werden mit einem Minuszeichen (-) vor dem Wert in den Ausgabepuffer geschrieben.
- ❏ Führende Nullen links vom Dezimalpunkt (mit Ausnahme der Ziffer direkt neben dem Dezimalpunkt) werden unterdrückt.
- ❏ Die Werte werden rechtsbündig in der ausgegebenen Zeichenkette angeordnet.

Tabelle 6-18 Gültige Operanden für die Operationen, die numerische Werte in Zeichenketten umwandeln

Eingänge/ Ausgänge	Datentyp	Operanden
IN	BYTE (Zeichenkette)	VB, LB, *VD, *LD, *AC
	INT	EW, AW, VW, MW, SMW, SW, T, Z, LW, AEW, *VD, *LD, *AC, Konstante
	DINT	ED, AD, VD, MD, SMD, SD, LD, AC, HC, *VD, *LD, *AC, Konstante
	REAL	ED, AD, VD, MD, SMD, SD, LD, AC, *VD, *LD, *AC, Konstante
FMT	BYTE	EB, AB, VB, MB, SMB, SB, LB, AC, *VD, *LD, *AC, Konstante
OUT	BYTE (Zeichenkette)	VB, LB, *VD, *LD, *AC
	INT	EW, AW, VW, MW, SMW, SW, T, Z, LW, AC, AAW, *VD, *LD, *AC
	DINT, REAL	ED, AD, VD, MD, SMD, SD, LD, AC, *VD, *LD, *AC

FMT

MSB							LSB
7	6	5	4	3	2	1	0
0	0	0	0	c	n	n	n

c = Komma (1) oder Dezimalpunkt (0)
nnn = Ziffern rechts vom Dezimalpunkt

	OUT	OUT +1	OUT +2	OUT +3	OUT +4	OUT +5	OUT +6	OUT +7	OUT +8
IN = 12	8				0	.	0	1	2
IN = -123	8				0	.	1	2	3
IN = 1234	8				1	.	2	3	4
IN = -12345	8		-	1	2	.	3	4	5

Bild 6-18 Operand FMT der Operation Ganze Zahl in Zeichenkette wandeln

Funktionsweise der Operation Ganze Zahl (32 Bit) in Zeichenkette wandeln

Die Operation Ganze Zahl (32 Bit) in Zeichenkette wandeln (DTS) wandelt eine ganze Zahl (32 Bit) IN in eine ASCII-Zeichenkette mit einer Länge von 12 Zeichen um. Das Format (FMT) gibt die Umwandlungsgenauigkeit rechts von der Dezimalzahl an und ob der Dezimalpunkt als Punkt oder als Komma dargestellt wird. Die sich ergebende Zeichenkette wird in 13 aufeinander folgende Bytes mit Beginn an OUT geschrieben. Weitere Informationen finden Sie in dem Abschnitt in Kapitel 4, der das Format von Zeichenketten beschreibt.

Fehlerbedingungen, die ENO = 0 setzen

- 0006 (Indirekte Adresse)
- 0091 (Operand außerhalb des Bereichs)
- Unzulässiges Format (nnn > 5)

Bild 6-19 beschreibt den Formatoperanden der Operation Ganze Zahl in Zeichenkette wandeln. Die Länge der ausgegebenen Zeichenkette beträgt immer 8 Zeichen. Die Anzahl der Ziffern rechts vom Dezimalpunkt im Ausgabepuffer wird vom Feld nnn angegeben. Der gültige Bereich für das Feld nnn liegt zwischen 0 und 5. Werden 0 Ziffern rechts vom Dezimalpunkt angegeben, wird der Wert ohne Dezimalpunkt angezeigt. Bei Werten von nnn größer als 5 wird eine Zeichenkette aus 12 ASCII-Leerzeichen ausgegeben. Das Bit c gibt an, ob ein Komma (c = 1) oder ein Dezimalpunkt (c = 0) als Trennzeichen zwischen der ganzen Zahl und dem Bruch verwendet werden soll. Die oberen 4 Bits des Formats müssen Null sein.

Bild 6-19 zeigt auch Beispiele für Werte, die mit Dezimalpunkt (c = 0) und mit vier Ziffern rechts vom Dezimalpunkt (nnn = 100) formatiert wurden. Der Wert in OUT ist die Länge der Zeichenkette. Die ausgegebene Zeichenkette wird entsprechend den folgenden Richtlinien formatiert:

- ❏ Positive Werte werden ohne Vorzeichen in den Ausgabepuffer geschrieben.
- ❏ Negative Werte werden mit einem Minuszeichen (-) vor dem Wert in den Ausgabepuffer geschrieben.
- ❏ Führende Nullen links vom Dezimalpunkt (mit Ausnahme der Ziffer direkt neben dem Dezimalpunkt) werden unterdrückt.
- ❏ Die Werte werden rechtsbündig in der ausgegebenen Zeichenkette angeordnet.

FMT

MSB							LSB
7	6	5	4	3	2	1	0
0	0	0	0	c	n	n	n

c = Komma (1) oder Dezimalpunkt (0)
nnn = Ziffern rechts vom Dezimalpunkt

	OUT	OUT +1	OUT +2	OUT +3	OUT +4	OUT +5	OUT +6	OUT +7	OUT +8	OUT +9	OUT +10	OUT +11	OUT +12
IN = 12	12					.	-	0	.	0	0	1	2
IN = -1234567	12					1	2	3	.	4	5	6	7

Bild 6-19 Operand FMT der Operation Ganze Zahl (32 Bit) in Zeichenkette wandeln

Funktionsweise der Operation Realzahl in Zeichenkette wandeln

Die Operation Realzahl in Zeichenkette wandeln (RTS) wandelt eine Realzahl IN in eine ASCII-Zeichenkette um. Das Format (FMT) gibt die Umwandlungsgenauigkeit rechts von der Dezimalzahl an, es gibt an, ob der Dezimalpunkt als Punkt oder als Komma dargestellt wird und es gibt die Länge der ausgegebenen Zeichenkette an.

Das Ergebnis der Umwandlung wird in einer Zeichenkette mit Beginn an OUT abgelegt. Die Länge der sich ergebenden Zeichenkette wird im Format angegeben und kann 3 bis 15 Zeichen umfassen. Weitere Informationen finden Sie in dem Abschnitt in Kapitel 4, der das Format von Zeichenketten beschreibt.

Fehlerbedingungen, die ENO = 0 setzen

- 0006 (Indirekte Adresse)
- 0091 (Operand außerhalb des Bereichs)
- Unzulässiges Format:
  nnn > 5
  ssss < 3
  ssss < Anzahl der erforderlichen Zeichen

Das von den S7-200 verwendete Realzahlenformat unterstützt maximal 7 signifikante Ziffern. Sollen mehr als 7 signifikante Ziffern angezeigt werden, wird ein Rundungsfehler ausgegeben.

Bild 6-20 beschreibt den Formatoperanden der Operation Realzahl in Zeichenkette wandeln. Die Länge der ausgegebenen Zeichenkette wird von dem Feld ssss angegeben. Die Größen 0, 1 oder 2 Bytes sind ungültig. Die Anzahl der Ziffern rechts vom Dezimalpunkt im Ausgabepuffer wird vom Feld nnn angegeben. Der gültige Bereich für das Feld nnn liegt zwischen 0 und 5. Werden 0 Ziffern rechts vom Dezimalpunkt angegeben, wird der Wert ohne Dezimalpunkt angezeigt. Die ausgegebene Zeichenkette wird mit ASCII-Leerzeichen belegt, wenn der Wert nnn größer als 5 ist oder wenn die angegebene Länge der ausgegebenen Zeichenkette zu klein ist, um den umgewandelten Wert zu speichern. Das Bit c gibt an, ob ein Komma (c = 1) oder ein Dezimalpunkt (c = 0) als Trennzeichen zwischen der ganzen Zahl und dem Bruch verwendet werden soll.

Bild 6-20 zeigt auch Beispiele für Werte, die mit Dezimalpunkt (c = 0) mit einer Ziffer rechts vom Dezimalpunkt (nnn = 001) und einer Länge der ausgegebenen Zeichenkette von 6 Zeichen (ssss = 0110) formatiert wurden. Der Wert in OUT ist die Länge der Zeichenkette. Die ausgegebene Zeichenkette wird entsprechend den folgenden Richtlinien formatiert:

- ❏ Positive Werte werden ohne Vorzeichen in den Ausgabepuffer geschrieben.
- ❏ Negative Werte werden mit einem Minuszeichen (-) vor dem Wert in den Ausgabepuffer geschrieben.
- ❏ Führende Nullen links vom Dezimalpunkt (mit Ausnahme der Ziffer direkt neben dem Dezimalpunkt) werden unterdrückt.
- ❏ Der Wert rechts vom Dezimalpunkt wird gerundet, damit er der Anzahl der Ziffern entspricht, die rechts vom Dezimalpunkt angegeben werden sollen.
- ❏ Die Größe der ausgegebenen Zeichenkette muss mindestens drei Bytes mehr umfassen als die Anzahl der Ziffern rechts vom Dezimalpunkt.
- ❏ Die Werte werden rechtsbündig in der ausgegebenen Zeichenkette angeordnet.

FMT

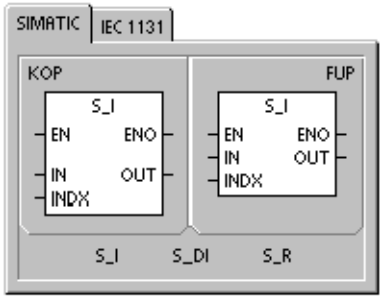
MSB							LSB
7	6	5	4	3	2	1	0
s	s	s	s	c	n	n	n

ssss = Länge der ausgegebenen Zeichenkette
c = Komma (1) oder Dezimalpunkt (0)
nnn = Ziffern rechts vom Dezimalpunkt

	OUT	OUT +1	OUT +2	OUT +3	OUT +4	OUT +5	OUT +6
IN = 1234.5	6	1	2	3	4	.	5
IN = -0.0004	6				0	.	0
IN = -3.67526	6			-	3	.	7
IN = 1.95	6				2	.	0

Bild 6-20 Operand FMT der Operation Realzahl in Zeichenkette wandeln

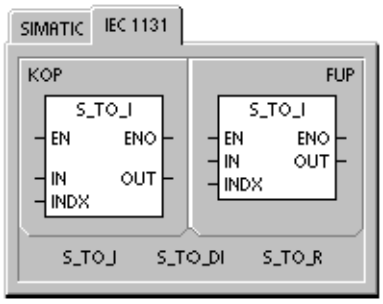
Umwandeln von Teilzeichenketten in numerische Werte



Die Operationen Teilzeichenkette in ganze Zahl wandeln (STI), Teilzeichenkette in ganze Zahl (32 Bit) wandeln (STD) und Teilzeichenkette in Realzahl wandeln (STR) wandeln die Zeichenkette in IN mit Beginn am Versatz INDX in eine ganze Zahl (16 Bit), ganze Zahl (32 Bit) oder in eine Realzahl in OUT um.

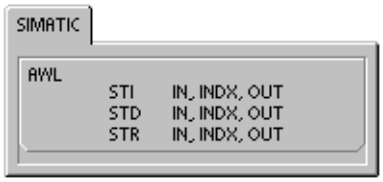
Fehlerbedingungen, die ENO = 0 setzen

- 0006 (Indirekte Adresse)
- 0091 (Operand außerhalb des Bereichs)
- 009B (Index = 0)
- SM1.1 (Überlauf)



Die Operationen Teilzeichenkette in ganze Zahl wandeln und Teilzeichenkette in ganze Zahl (32 Bit) wandeln wandeln Zeichenketten mit dem folgenden Format um:
[Leerzeichen] [+ oder -] [Ziffern 0 - 9]

Die Operation Teilzeichenkette in Realzahl wandeln wandelt Zeichenketten mit dem folgenden Format um:
[Leerzeichen] [+ oder -] [Ziffern 0 - 9] [. oder ,] [Ziffern 0 - 9]



Der Wert INDX wird üblicherweise auf 1 gesetzt, so dass die Umwandlung mit dem ersten Zeichen der Zeichenkette beginnt. Für den Wert INDX können andere Werte eingestellt werden, so dass die Umwandlung an ver-
schiedenen Stellen in der Zeichenkette beginnt. Dies ist beispielsweise dann nützlich, wenn die eingegebene Zeichenkette Text enthält, der nicht Teil der umzuwandelnden Nummer ist. Beispiel: die eingegebene Zeichenkette lautet “Temperature: 77.8”. Sie setzen INDX auf den Wert 13, um das Wort “Temperature:” zu Beginn der Zeichenkette zu überspringen.

Die Operation Teilzeichenkette in Realzahl wandeln wandelt keine Zeichenketten mit wissenschaftlicher Notierung oder Exponentialfunktionen von Realzahlen um. Die Operation erzeugt keinen Überlauffehler (SM1.1), aber sie wandelt die Zeichenkette in eine Realzahl bis zur Exponentialfunktion um und beendet dann die Umwandlung. Beispiel: Die Zeichenkette ‘1.234E6’ wird fehlerfrei in die Realzahl 1.234 umgewandelt.

Die Umwandlung wird beendet, wenn das Ende der Zeichenkette erreicht ist oder wenn das erste ungültige Zeichen erkannt wird. Ungültig sind alle Zeichen, die keine Ziffern sind (0 - 9).

Der Überlauffehler (SM1.1) wird gesetzt, wenn die Umwandlung einen ganzzahligen Wert erzeugt, der für den Ausgabewert zu lang ist. Beispiel: Die Operation Teilzeichenkette in ganze Zahl wandeln setzt den Überlauffehler, wenn die eingegebene Zeichenkette einen Wert erzeugt, der größer als 32767 oder kleiner als -32768 ist.

Der Überlauffehler (SM1.1) wird auch gesetzt, wenn keine Umwandlung möglich ist, wenn die eingegebene Zeichenkette keinen gültigen Wert enthält. Beispiel: Enthält die eingegebene Zeichenkette ‘A123’, setzt die Umwandlungsoperation SM1.1 (Überlauf) und der Ausgabewert bleibt unverändert.

Tabelle 6-19 Gültige Operanden für die Operationen, die Teilzeichenketten in numerische Werte umwandeln

Eingänge/ Ausgänge	Datentyp	Operanden
IN	BYTE (Zeichenkette)	EB, AB, VB, MB, SMB, SB, LB, *VD, *LD, *AC, Konstante
INDX	BYTE	VB, EB, AB, MB, SMB, SB, LB, AC, *VD, *LD, *AC, Konstante
OUT	BYTE (Zeichenkette)	VB, EB, AB, MB, SMB, SB, LB, *VD, *LD, *AC, Konstante
	INT	VW, EW, AW, MW, SMW, SW, T, Z, LW, AC, AAW, *VD, *LD, *AC
	DINT, REAL	VD, ED, AD, MD, SMD, SD, LD, AC, *VD, *LD, *AC

Gültige eingegebene Zeichenketten für Ganzzahlen (16 Bit) und Ganzzahlen (32 Bit)

Eingeg. Zeichenkette	Ausg. ganze Zahl
‘123’	123
‘-00456’	-456
‘123.45’	123
‘+2345’	2345
‘000000123ABCD’	123

Gültige eingegebene Zeichenketten für Realzahlen

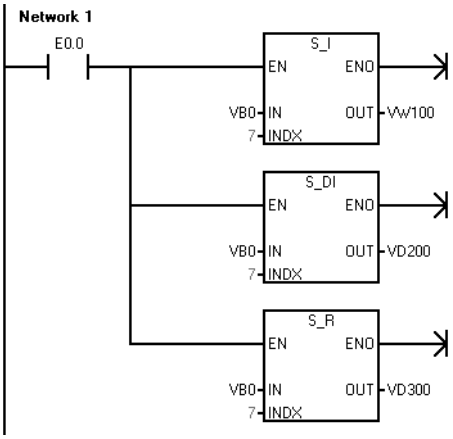
Eingeg. Zeichenkette	Ausg. Realzahl
‘123’	123.0
‘-00456’	-456.0
‘123.45’	123.45
‘+2345’	2345.0
‘00.000000123’	0.000000123

Ungültige eingegebene Zeichenketten

Eingeg. Zeichenkette
‘A123’
‘ ’
‘++123’
‘+-123
‘+ 123’

Bild 6-21 Beispiele für gültige und ungültige eingegebene Zeichenketten

Beispiel: Zeichenkettenumwandlung: Teilzeichenkette in ganze Zahl (16 Bit), ganze Zahl (32 Bit) und Realzahl wandeln




```
Netzwerk 1    //Numerische Zeichenkette in ganze Zahl wandeln.
              //Numerische Zeichenkette in ganze Zahl (32 Bit) wan
              //deln.
              //Numerische Zeichenkette in Realzahl wandeln.
LD    E0.0
STI   VB0,7,VW100
STD   VB0,7,VD200
STR   VB0,7,VD300
```

VB0											VB11
11	’T’	’e’	’m’	’P’	’ ’	’ ’	’9’	’8’	’.’	’6’	’f’

Nach Ausführung des Netzwerks:

VW100 (Ganzzahl, 16 Bit) = 98

VD200 (Ganzzahl, 32 Bit) = 98

VD300 (Realzahl) = 98,6

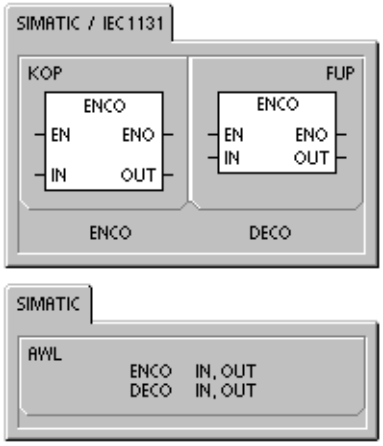
Operationen Hexadezimalzahl in Bit wandeln und Bit in Hexadezimalzahl wandeln

Hexadezimalzahl in Bit wandeln

Die Operation Hexadezimalzahl in Bit wandeln (ENCO) schreibt die Bitnummer des niederwertigsten Bit im Eingangswort IN in das niederwertigste Halb-Byte (4 Bit) des Ausgangsbytes (OUT).

Bit in Hexadezimalzahl wandeln

Die Operation Bit in Hexadezimalzahl wandeln (DECO) 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 Augangsworts werden auf 0 gesetzt.



Sondermerker und ENO

Bei beiden Operationen Hexadezimalzahl in Bit wandeln und Bit in Hexadezimalzahl wandeln wirken sich die folgenden Bedingungen auf ENO aus.

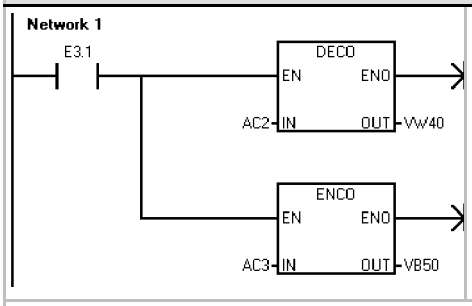
Fehlerbedingungen, die ENO = 0 setzen

- 0006 (Indirekte Adresse)

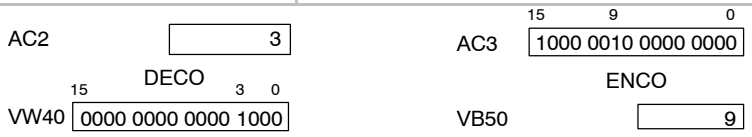
Tabelle 6-20 Gültige Operanden für die Operationen Hexadezimalzahl in Bit wandeln und Bit in Hexadezimalzahl wandeln

Eingänge/ Ausgänge	Datentypen	Operanden
IN	BYTE	EB, AB, VB, MB, SMB, SB, LB, AC, *VD, *LD, *AC, Konstante
	WORD	EW, AW, VW, MW, SMW, SW, T, Z, LW, AC, AEW, *VD, *LD, *AC, Konstante
OUT	BYTE	EB, AB, VB, MB, SMB, SB, LB, AC, *VD, *LD, *AC
	WORD	EW, AW, VW, MW, SMW, SW, T, Z, LW, AC, AAW, *VD, *LD, *AC

Beispiel: Operationen Bit in Hexadezimalzahl wandeln und Hexadezimalzahl in Bit wandeln

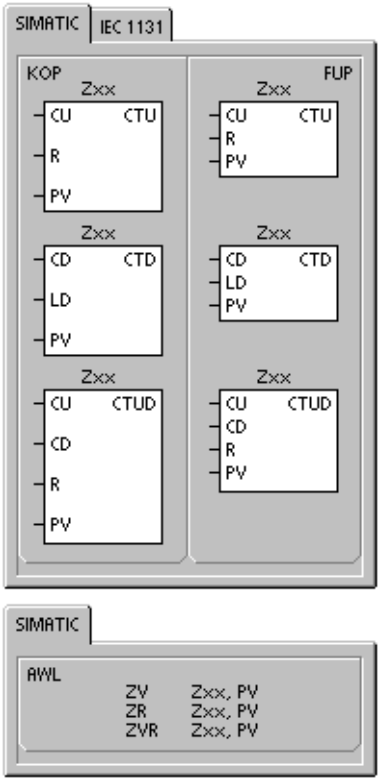



```
Netzwerk 1    //Akkumulator 2 enthält Fehlerbits.
              //1. Die Operation DECO setzt das Bit in VW40,
              //    das diesem Fehlercode entspricht.
              //2. Die Operation ENCO wandelt das
              //    niederwertigste gesetzte Bit in einen Fehlercode
              //um,
              //    der in VB50 gespeichert wird.
LD      E3.1
DECO    AC2, VW40
ENCO    AC3, VB50
```



Zähloperationen

SIMATIC: Zähler



Vorwärtszählen

Die Operation Vorwärtszählen (CTU/ZV) zählt bei steigender Flanke am Vorwärtszähleingang (CU) vom aktuellen Wert des Zählers an vorwärts. Ist der aktuelle Wert Zxx größer als oder gleich dem voreingestellten Wert PV, dann wird das Zählerbit Zxx aktiviert. Der Zähler wird zurückgesetzt, wenn der Rücksetzeingang (R) aktiviert wird oder die Operation Rücksetzen ausgeführt wird. Der Zähler hört auf zu zählen, wenn der Höchstwert (32.767) erreicht ist.

Funktionsweise in AWL:

- Rücksetzeingang: Spitze des Stack
- Vorwärtszähleingang: An die zweite Stelle im Stack geladener Wert

Rückwärtszählen

Die Operation Rückwärtszählen (CTD/ZR) zählt bei steigender Flanke am Rückwärtszähleingang (CD) vom aktuellen Wert des Zählers an rückwärts. Ist der aktuelle Wert Zxx gleich 0, dann wird das Zählerbit Zxx eingeschaltet. Der Zähler setzt das Zählerbit Zxx zurück und lädt den aktuellen Wert in den voreingestellten Wert PV, wenn der Ladeeingang LD eingeschaltet wird. Der Zähler stoppt, wenn er Null erreicht, und das Zählerbit Zxx wird eingeschaltet.

Funktionsweise in AWL:

- Ladeeingang: Spitze des Stack
- Rückwärtszähleingang: An die zweite Stelle im Stack geladener Wert

Vorwärts-/Rückwärtszählen

Die Operation Vorwärts-/Rückwärtszählen (CTUD/ZVR) zählt bei steigender Flanke am Vorwärtszähleingang (CU) vorwärts und bei steigender Flanke am Rückwärtszähleingang (CD) rückwärts. Der aktuelle Wert Zxx des Zählers enthält den aktuellen Zählwert. Der voreingestelle Wert PV wird bei jeder Ausführung der Operation mit dem aktuellen Wert verglichen 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.

Ist der aktuelle Wert Zxx größer als oder gleich dem voreingestellten Wert PV, dann wird das Zählerbit Zxx aktiviert. In allen anderen Fällen wird das Zählerbit ausgeschaltet. Der Zähler wird zurückgesetzt, wenn der Rücksetzeingang (R) aktiviert wird oder die Operation Rücksetzen ausgeführt wird. Der Rückwärtszähler hört auf zu zählen, wenn er PV erreicht.

Funktionsweise in AWL:

- Rücksetzeingang: Spitze des Stack
- Rückwärtszähleingang: An die zweite Stelle im Stack geladener Wert
- Vorwärtszähleingang: An die dritte Stelle im Stack geladener Wert

Tabelle 6-21 Gültige Operanden für die SIMATIC Zähler

Eingänge/Ausgänge	Datentypen	Operanden
Zxx	WORD	Konstante (Z0 bis Z255)
CU, CD, LD, R	BOOL	E, A, V, M, SM, S, T, Z, L, Signalfluss
PV	INT	EW, AW, VW, MW, SMW, SW, LW, T, Z, AC, AEW, *VD, *LD, *AC, Konstante

Tipp

Da jeder Zähler einen eigenen aktuellen Wert besitzt, dürfen Sie nicht mehreren Zählern die gleiche Nummer zuordnen (Vorwärtszähler, Vorwärts-/Rückwärtszähler und Rückwärtszähler mit gleicher Nummer greifen auf den gleichen aktuellen Wert zu).

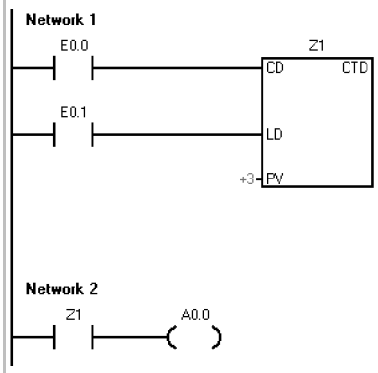
Wenn Sie einen Zähler mit der Operation Rücksetzen zurücksetzen, werden das Zählerbit und der aktuelle Wert des Zählers zurückgesetzt. Mit der Nummer des Zählers sprechen Sie den aktuellen Wert und auch das Zählerbit des Zählers an.

Tabelle 6-22 Funktionsweise der Zähler

Typ	Betrieb	Zählerbit	Einschalten/Erster Zyklus
ZV	CU inkrementiert den aktuellen Wert. Der aktuelle Wert wird weiter inkrementiert, bis er 32.767 erreicht.	Das Zählerbit wird eingeschaltet, wenn: Aktueller Wert >= Voreinstellung	Zählerbit ist aus. Aktueller Wert kann gespeichert werden.[1]
ZVR	CU inkrementiert den aktuellen Wert. CD dekrementiert den aktuellen Wert. Der aktuelle Wert wird weiter inkrementiert oder dekrementiert, bis der Zähler zurückgesetzt wird.	Das Zählerbit wird eingeschaltet, wenn: Aktueller Wert >= Voreinstellung	Zählerbit ist aus. Aktueller Wert kann gespeichert werden.[1]
ZR	CD dekrementiert den aktuellen Wert, bis der aktuelle Wert 0 erreicht.	Das Zählerbit wird eingeschaltet, wenn: Aktueller Wert = 0	Zählerbit ist aus. Aktueller Wert kann gespeichert werden.[1]

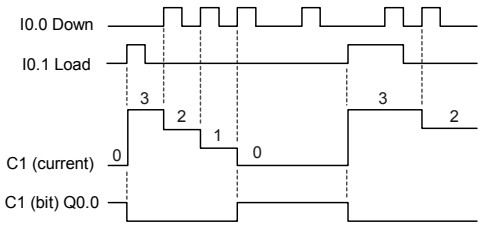
[1] Sie können den aktuellen Wert des Zählers als remanent definieren. Informationen zu remanentem Speicher der S7-200 CPU finden Sie in Kapitel 4.

Beispiel: SIMATIC-Operation Rückwärtszählen

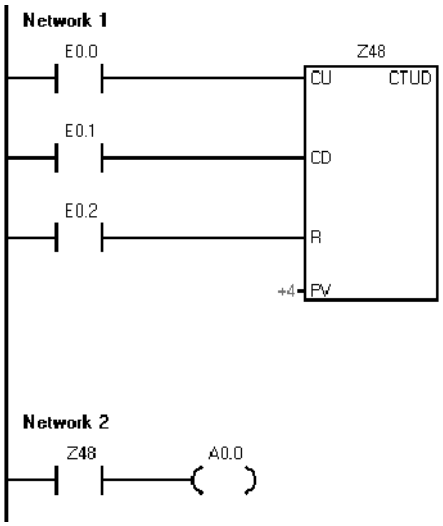



```
Netzwerk 1    //Der aktuelle Wert des Rückwärtszählers Z1 zählt von 3 bis 0,
              //wenn E0.1 = AUS,
              //E0.0 AUS-EIN dekrementiert den aktuellen Wert von Z1
              //E0.1 = EIN lädt den voreingestellten Wert 3 zum
              //Rückwärtszählen.
LD      E0.0
LD      E0.1
ZR      Z1, +3

Netzwerk 2    //Das Bit von Z1 ist EIN, wenn der aktuelle Wert des Zählers Z1
= 0 ist.
LD      Z1
=       A0.0
```

Impulsdiagramm

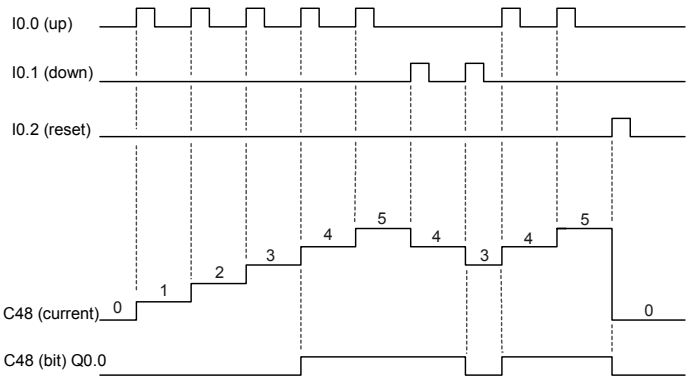


Beispiel: SIMATIC-Operation Vorwärts-/Rückwärtszählen




```
Netzwerk 1      //E0.0 zählt vorwärts.
                //E0.1 zählt rückwärts.
                //E0.2 setzt den aktuellen Wert auf 0 zurück.
LD      E0.0
LD       E0.1
LD      E0.2
ZVR     Z48, +4

Netzwerk 2      //Der Vorwärts-/Rückwärtszähler Z48 schaltet das Bit
                //Z48 ein,
                //wenn der aktuelle Wert >= 4 ist.
LD      Z48
=       A0.0
```

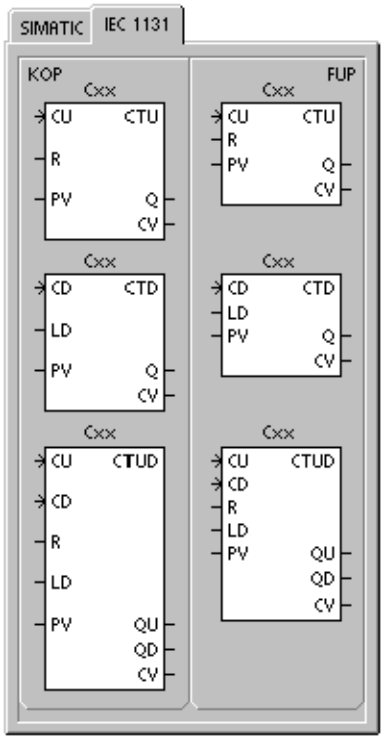
Impulsdiagramm



IEC: Zähler

Vorwärtszähler

Der Vorwärtszähler (CTU) zählt bei steigender Flanke am Vorwärtszähleingang (CU) vom aktuellen Wert bis zum voreingestellten Wert (PV) vorwärts. Ist der aktuelle Wert (CV) größer als oder gleich dem voreingestellten Wert, dann wird das Ausgangsbit des Zählers (Q) aktiviert. Der Zähler wird zurückgesetzt, wenn der Rücksetzeingang (R) aktiviert wird. Der Vorwärtszähler hört auf zu zählen, wenn der voreingestellte Wert erreicht ist.



Rückwärtszähler

Der Rückwärtszähler (CTD) zählt bei steigender Flanke am Rückwärtszähleingang (CD) vom voreingestellten Wert (PV) rückwärts. Ist der aktuelle Wert (CV) gleich Null, dann wird das Ausgangsbit des Zählers (Q) eingeschaltet. Der Zähler wird zurückgesetzt und lädt den aktuellen Wert (CV) in den voreingestellten Wert, wenn der Ladeeingang (LD) eingeschaltet wird. Der Rückwärtszähler hört auf zu zählen, wenn er Null erreicht.

Vorwärts-/Rückwärtszähler

Der Vorwärts-/Rückwärtszähler (CTUD) zählt bei steigender Flanke am Vorwärtszähleingang (CU) vom aktuellen Wert (CV) vorwärts und bei steigender Flanke am Rückwärtszähleingang (CD) vom voreingestellten Wert rückwärts. Ist der aktuelle Wert gleich der Voreinstellung, dann wird der Ausgang (QU) für Vorwärtszählen eingeschaltet. Ist der aktuelle Wert gleich Null, dann wird der Ausgang (QD) für Rückwärtszählen eingeschaltet. Der Zähler lädt den voreingestellten Wert (PV) in den aktuellen Wert, wenn der Ladeeingang (LD) eingeschaltet wird. Der Zähler wird auch zurückgesetzt und lädt den Wert 0 in den aktuellen Wert, wenn der Rücksetzeingang (R) eingeschaltet wird. Der Rückwärtszähler hört auf zu zählen, wenn er die Voreinstellung oder 0 erreicht.

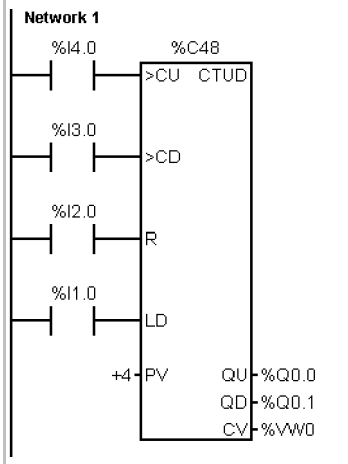
Tabelle 6-23 Gültige Operanden für die IEC-Zähler

Eingänge/ Ausgänge	Datentypen	Operanden
Zxx	CTU, CTD, CTUD	Konstante (Z0 bis Z255)
CU, CD, LD, R	BOOL	E, A, V, M, SM, S, T, Z, L, Signalfluss
PV	INT	EW, AW, VW, MW, SMW, SW, LW, AC, AEW, *VD, *LD, *AC, Konstante
Q, QU, QD	BOOL	E, A, V, M, SM, S, L
CV	INT	EW, AW, VW, MW, SW, LW, AC, *VD, *LD, *AC

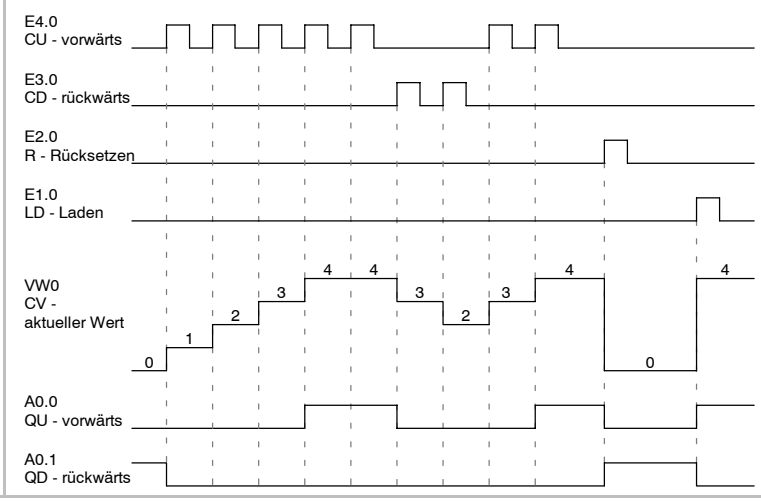
Tipp

Da jeder Zähler einen eigenen aktuellen Wert besitzt, dürfen Sie nicht mehreren Zählern die gleiche Nummer zuordnen (Vorwärtszähler, Rückwärtszähler und Vorwärts-/Rückwärtszähler greifen auf den gleichen aktuellen Wert des Zählers zu.)

Beispiel: IEC: Zähler



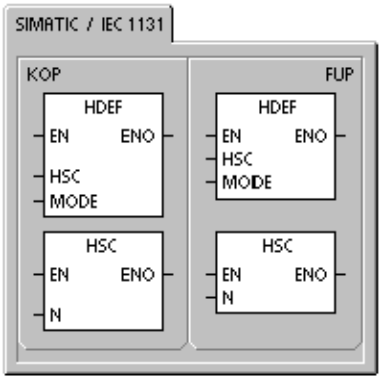
Impulsdiagramm



6

Schnelle Zähler

Modus für schnellen Zähler definieren



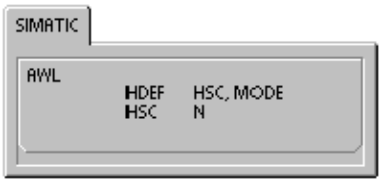
Die Operation Modus für schnellen Zähler definieren (HDEF) stellt für einen bestimmten schnellen Zähler (HSCx) eine Betriebsart ein. Die Betriebsart definiert Taktgeber, Richtung, Start- und Rücksetzfunktionen des schnellen Zählers.

Für jeden schnellen Zähler führen Sie eine Operation Modus für schnellen Zähler definieren aus.

Fehlerbedingungen, die ENO = 0 setzen

- 0003 (Eingangskonflikt)
- 0004 (unzulässige Operation im Interrupt)
- 000A (HSC-Neudefinition)

Schneller Zähler



Die Operation Schnellen Zähler aktivieren (HSC) 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.

Für die schnellen Zähler können zwölf verschiedene Betriebsarten eingestellt werden (siehe Tabelle 6-25).

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 (1x) oder vierfache (4x) Zählgeschwindigkeiten auswählen. Alle Zähler laufen mit maximaler Geschwindigkeit, ohne sich gegenseitig zu beeinträchtigen.

Fehlerbedingungen, die ENO = 0 setzen

- 0001 (HSC vor HDEF)
- 0005 (HSC/PLS gleichzeitig)

Tabelle 6-24 Gültige Operanden für die Operationen für schnelle Zähler

Eingänge/Ausgänge	Datentypen	Operanden
HSC, MODE	BYTE	Konstante
N	WORD	Konstante

Programmierung Tips

Die Tipps für die Programmierung auf der Dokumentations-CD bieten Programme mit schnellen Zählern (siehe Tip 4 und Tip 29).

Schnelle Zähler zählen schnell auftretende Ereignisse, die mit der Zyklusrate der S7-200 nicht gesteuert werden können. Die maximale Zählfrequenz eines schnellen Zählers hängt von der Variante Ihrer S7-200 CPU ab. Weitere Informationen finden Sie im Anhang A.

Tipp

Die CPU 221 und die CPU 222 unterstützen vier schnelle Zähler: HSC0, HSC3, HSC4 und HSC5. Diese CPUs unterstützen HSC1 und HSC2 nicht.

Die CPU 224, CPU 226 und die CPU 226XM unterstützen sechs schnelle Zähler: HSC0 bis HSC5.

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. (Sie können alternativ auch alle Interruptereignisse in einem einzigen Interruptprogramm bearbeiten.)

Beschreibung der unterschiedlichen schnellen Zähler

Alle Zähler arbeiten in der gleichen Zählerart auf die gleiche Weise. Es gibt vier grundlegende Zählerarten: Einphasenzähler mit interner Richtungssteuerung, Einphasenzähler mit externer Richtungssteuerung, Zweiphasenzähler mit 2 Takteingänge und A/B-Zähler. Beachten Sie, dass nicht jeder Zähler alle Zählerarten unterstützt. 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. Wird der Starteingang eingeschaltet, während der Rücksetzeingang aktiv ist, dann wird der aktuelle Wert gelöscht.

Bevor Sie einen schnellen Zähler verwenden, müssen Sie mit einer Operation HDEF (Modus für schnellen Zähler definieren) eine Zählerart einstellen. Mit dem Merker des ersten Zyklus, SM0.1 (dieses Bit wird für den ersten Zyklus eingeschaltet und danach ausgeschaltet), rufen Sie ein Unterprogramm auf, das die Operation HDEF enthält.

Programmieren eines schnellen Zählers

Operations-Assistent

Mit dem Operations-Assistent für schnelle Zähler richten Sie den Zähler ein. Der Assistent benötigt die folgenden Angaben: Typ und Betriebsart des Zählers, Voreinstellung des Zählers, aktueller Wert des Zählers und anfängliche Zählrichtung. Sie rufen den Operations-Assistenten für schnelle Zähler mit dem Menübefehl **Extras > Operations-Assistent** auf. Dann wählen Sie im Fenster des Operations-Assistenten die Option HSC.

Zum Programmieren eines schnellen Zählers müssen Sie die folgenden Aufgaben ausführen:

- Zähler und Betriebsart definieren.
- Steuerbyte einrichten.
- Aktuellen Wert einstellen (Anfangswert).
- Voreingestellen Wert (Zielwert) einstellen.
- Interruptprogramm zuweisen und freigeben.
- Schnellen Zähler aktivieren.

Definieren der Zählerarten und der Zählereingänge

Mit der Operation Modus für schnellen Zähler definieren weisen Sie die Zählerarten und die Zählereingänge zu.

Tabelle 6-25 zeigt die Eingänge von schnellen Zählern, die für Funktionen wie Taktgeber, Richtungssteuerung, Rücksetzen und Starten verwendet werden. Ein Eingang kann nicht für zwei verschiedene Funktionen verwendet werden. Wird ein Eingang jedoch nicht von der aktuellen Zählerart des definierten schnellen Zählers benötigt, kann er für andere Zwecke genutzt werden. Setzen Sie beispielsweise HSC0 in Zählerart 1 ein, in der die Eingänge E0.0 und E0.2 benötigt werden, können Sie E0.1 für Flankeninterrupts oder für HSC3 verwenden.

Tipp

Beachten Sie, dass alle Zählerarten von HSC0 immer den Eingang E0.0 verwenden und alle Zählerarten von HSC4 immer den Eingang E0.3 benötigen, so dass diese Eingänge nie zur Verfügung stehen, wenn diese Zähler in Betrieb sind.

Tabelle 6-25 Eingänge der schnellen Zähler

Zähler-art	Beschreibung	Eingänge			
	HSC0	E0.0	E0.1	E0.2	
	HSC1	E0.6	E0.7	E1.0	E1.1
	HSC2	E1.2	E1.3	E1.4	E1.5
	HSC3	E0.1			
	HSC4	E0.3	E0.4	E0.5	
	HSC5	E0.4			
0	Einphasenzähler mit interner Richtungssteuerung	Taktgeber			
1		Taktgeber		Rücksetzen	
2		Taktgeber		Rücksetzen	Starten
3	Einphasenzähler mit externer Richtungssteuerung	Taktgeber	Richtung		
4		Taktgeber	Richtung	Rücksetzen	
5		Taktgeber	Richtung	Rücksetzen	Starten
6	Zweiphasenzähler mit 2 Takteingängen	Taktgeber vorwärts	Taktgeber rückwärts		
7		Taktgeber vorwärts	Taktgeber rückwärts	Rücksetzen	
8		Taktgeber vorwärts	Taktgeber rückwärts	Rücksetzen	Starten
9	A/B-Zähler	Taktgeber A	Taktgeber B		
10		Taktgeber A	Taktgeber B	Rücksetzen	
11		Taktgeber A	Taktgeber B	Rücksetzen	Starten

Beispiele für die Zählerarten von schnellen Zählern

Die folgenden Impulsdiagramme (Bilder 6-22 bis 6-26) zeigen, wie jeder Zähler entsprechend seiner Betriebsart arbeitet.

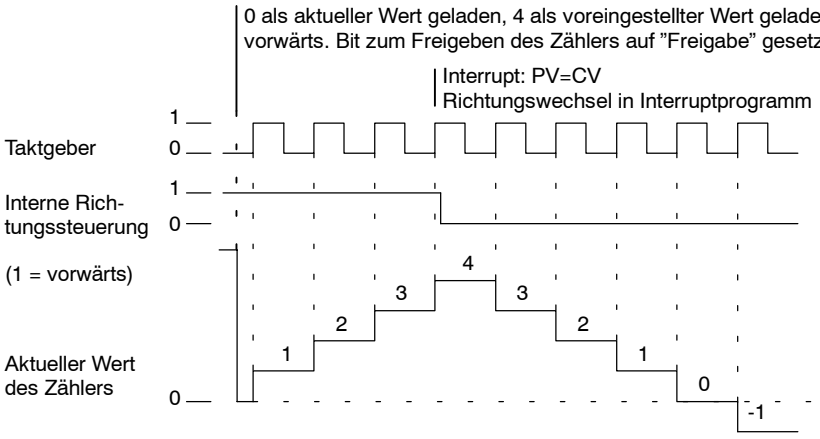


Bild 6-22 Beispiel für den Betrieb in einer der Zählerarten 0, 1 oder 2

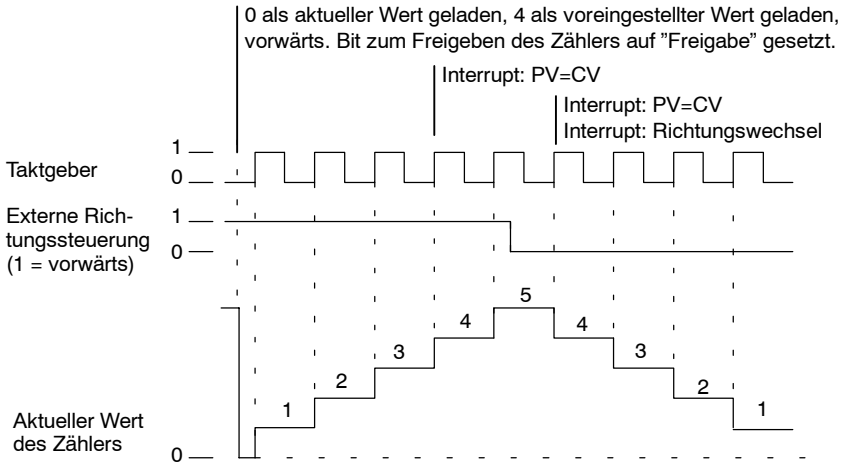


Bild 6-23 Beispiel für den Betrieb in einer der Zählerarten 3, 4 oder 5

Wenn Sie 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

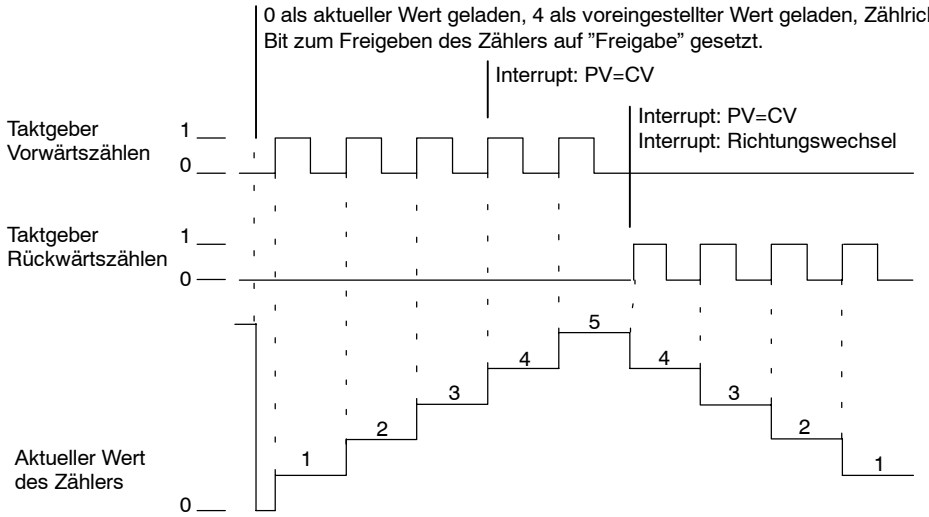


Bild 6-24 Beispiel für den Betrieb in einer der Zählerarten 6, 7 oder 8

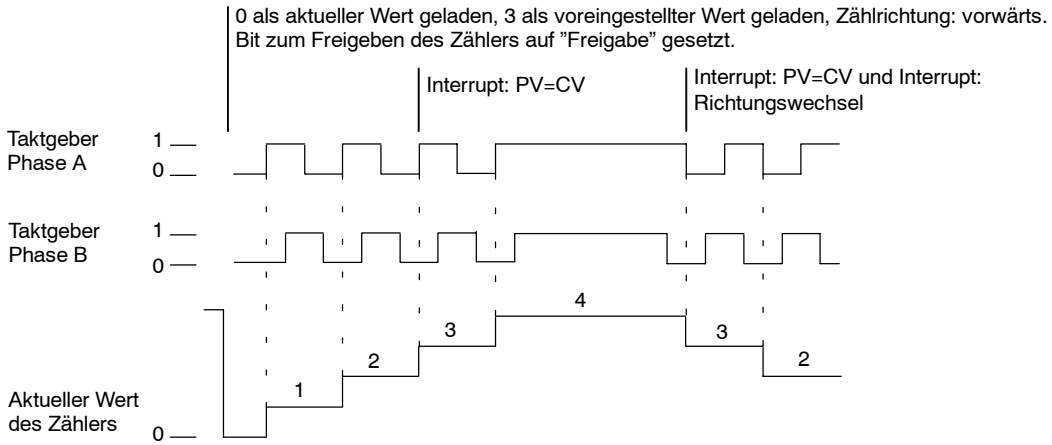


Bild 6-25 Beispiel für den Betrieb in einer der Zählerarten 9, 10 oder 11 (A/B-Zähler, einfache Geschwindigkeit)

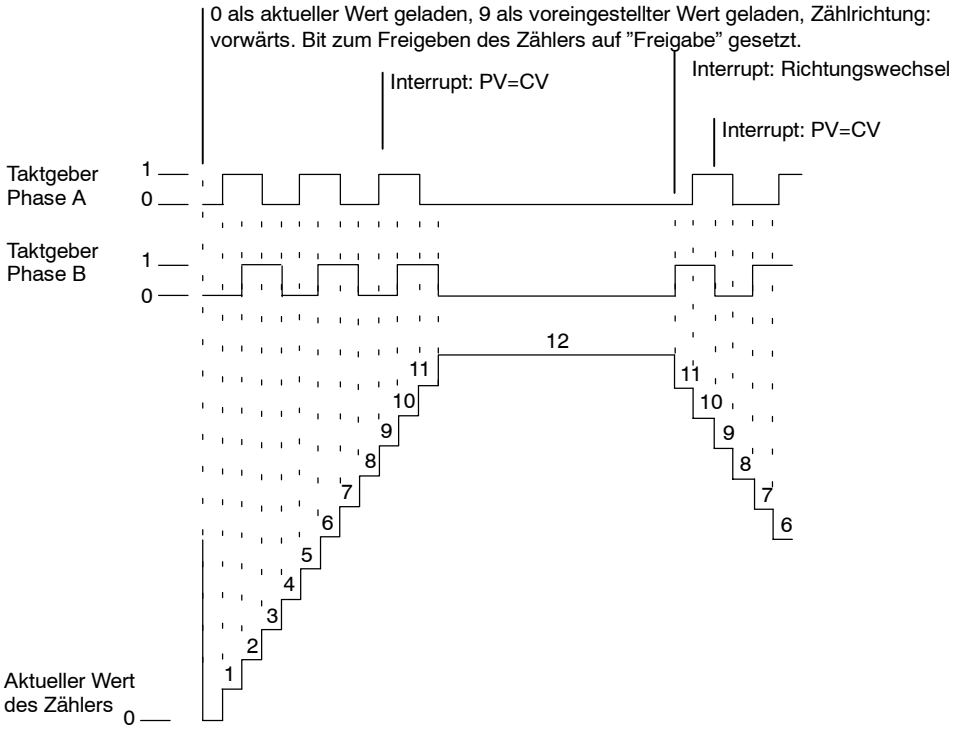


Bild 6-26 Beispiel für den Betrieb in einer der Zählerarten 9, 10 oder 11 (A/B-Zähler, vierfache Geschwindigkeit)

Funktionsweise der Rücksetz- und Starteingänge

Der Betrieb der Rücksetz- und Starteingänge ist in Bild 6-27 dargestellt und gilt für alle Zähler, die diese Eingänge verwenden. In den Diagrammen für die Rücksetz- und Starteingänge ist der Aktivitätszustand beider Eingänge mit Pegel High programmiert.

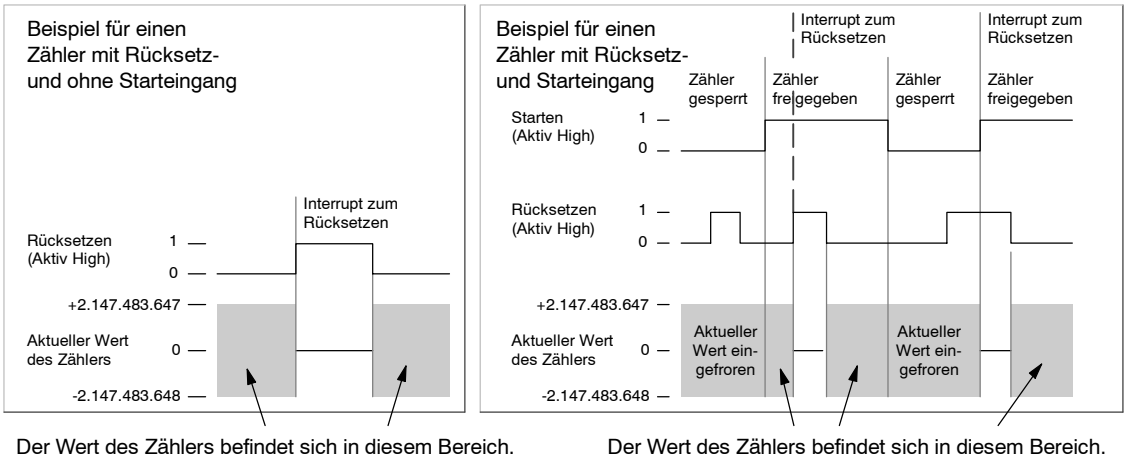


Bild 6-27 Beispiel für den Betrieb eines Zählers mit Rücksetz- und mit und ohne Starteingang

Vier Zähler 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 ausgführt wird. Die Bits werden in Tabelle 6-26 beschrieben.

Tipp

Bevor die Operation HDEF ausgeführt werden kann, müssen Sie diese drei Steuerbits auf den gewünschten Zustand setzen. Andernfalls übernimmt der Zähler die voreingestellte Konfiguration für die gewählte Zählerart.

Wird die Operation HDEF ausgeführt, können Sie die Zählereinstellung nicht mehr ändern, es sei denn, Sie versetzen die S7-200 in den Betriebszustand STOP.

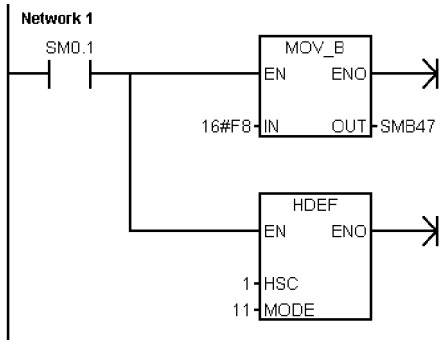
Tabelle 6-26 Steuerbits für Pegel für Rücksetzen und Starten aktiv sowie für einfache bzw. vierfache Geschwindigkeit

HSC0	HSC1	HSC2	HSC4	Beschreibung (nur wenn HDEF ausgeführt wird)
SM37.0	SM47.0	SM57.0	SM147.0	Steuerbit für Pegel bei Rücksetzen aktiv[1]: 0 = Rücksetzen: Aktiv High 1 = Rücksetzen: Aktiv Low
---	SM47.1	SM57.1	---	Steuerbit für Pegel bei Starten aktiv[1]: 0 = Starten: Aktiv High 1 = Starten: Aktiv Low
SM37.2	SM47.2	SM57.2	SM147.2	Zählgeschwindigkeit A/B-Zähler: 0 = vierfache Zählgeschwindigkeit 1 = einfache Zählgeschwindigkeit

1 Die Voreinstellungen für Rücksetz- und Starteingang sind Aktiv High. Bei A/B-Zählern ist die vierfache Zählgeschwindigkeit (viermal die Frequenz des Taktgebers) voreingestellt.

Beispiel: Operation Modus für schnellen Zähler definieren

MAIN




```
Netzwerk 1   //Im ersten Zyklus:
             //1.  Für Start- und Rücksetzeingang
             //    Aktiv High und vierfache Zählgeschwindigkeit
             //    wählen.
             //2.  HSC1 als A/B-Zähler einrichten
             //    mit Rücksetz- und Starteingang.
LD    SM0.1
MOVB  16#F8, SMB47
HDEF  1, 11
```

Einrichten des Steuerbyte

Wenn Sie Zähler und Zählerart definiert haben, können Sie die dynamischen Parameter des Zählers programmieren. Jeder schnelle Zähler verfügt über ein Steuerbyte für die folgenden Aktivitäten:

- ❏ Zähler aktivieren oder deaktivieren
- ❏ Richtungssteuerung (nur Betriebsarten 0, 1 und 2) oder anfängliche Zählrichtung in allen anderen Betriebsarten
- ❏ Aktuellen Wert laden
- ❏ Voreingestellten Wert laden

Das Steuerbyte und die zugeordneten aktuellen und voreingestellten Werte werden bei Ausführung der Operation HSC überprüft. Tabelle 6-27 beschreibt die Steuerbits.

Tabelle 6-27 Steuerbits für HSC0, HSC1, HSC2, HSC3, HSC4 und HSC5

HSC0	HSC1	HSC2	HSC3	HSC4	HSC5	Beschreibung
SM37.3	SM47.3	SM57.3	SM137.3	SM147.3	SM157.3	Steuerbit für Zählrichtung: 0 = Rückwärtszählen 1 = Vorwärtszählen
SM37.4	SM47.4	SM57.4	SM137.4	SM147.4	SM157.4	Zählrichtung in HSC schreiben: 0 = Keine Aktualisierung 1 = Richtung aktualisieren
SM37.5	SM47.5	SM57.5	SM137.5	SM147.5	SM157.5	Neuen voreingestellten Wert in HSC schreiben: 0 = Keine Aktualisierung 1 = Voreingestellten Wert aktualisieren
SM37.6	SM47.6	SM57.6	SM137.6	SM147.6	SM157.6	Neuen aktuellen Wert in HSC schreiben: 0 = Keine Aktualisierung 1 = Aktuellen Wert aktualisieren
SM37.7	SM47.7	SM57.7	SM137.7	SM147.7	SM157.7	Freigabe HSC: 0 = HSC sperren 1 = HSC freigeben

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. Zum Laden eines neuen aktuellen Werts oder eines neuen voreingestellten Werts in den schnellen Zähler müssen Sie das Steuerbyte und die Sondermerkerbytes einrichten, die den aktuellen Wert und/oder den voreingestellten Wert enthalten. Sie müssen auch die Operation HSC ausführen, damit die neuen Werte in den schnellen Zähler übertragen werden. Tabelle 6-28 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 gelesen werden, indem der Speicherbereich HC (aktueller Wert des schnellen Zählers) und die Nummer des Zählers (0, 1, 2, 3, 4 oder 5) angegeben werden (siehe Tabelle 6-28). Sie können den aktuellen Wert direkt lesen. Zum Schreiben müssen Sie jedoch die Operation HSC verwenden.

Tabelle 6-28 Neue aktuelle und neue voreingestellte Wertevon HSC0, HSC1, HSC2, HSC3, HSC4 und HSC5

Zu ladender Wert	HSC0	HSC1	HSC2	HSC3	HSC4	HSC5
Neuer aktueller Wert	SMD38	SMD48	SMD58	SMD138	SMD148	SMD158
Neuer voreingestellter Wert	SMD42	SMD52	SMD62	SMD142	SMD152	SMD162

Tabelle 6-29 Aktuelle Werte von HSC0, HSC1, HSC2, HSC3, HSC4 und HSC5

Wert	HSC0	HSC1	HSC2	HSC3	HSC4	HSC5
Strom	HC0	HC1	HC2	HC3	HC4	HC5

Adressierung von schnellen Zählern (HC)

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 6-28).

Bild 6-28 Zugriff auf den aktuellen Wert eines schnellen Zählers

Zuweisen von Interrupts

Alle Zählerarten unterstützen einen Interrupt für die Situation, dass der aktuelle Wert gleich dem voreingestellten Wert ist. Zählerarten, die einen externen Rücksetzeingang verwenden, unterstützen einen Interrupt für den Fall, dass das externe Rücksetzen aktiviert wird. Alle Zählerarten mit Ausnahme der Zählerarten 0, 1 und 2 unterstützen einen Interrupt für Zählrichtungswechsel. Jede dieser Interruptbedingungen kann einzeln freigegeben oder gesperrt werden. Eine ausführliche Beschreibung zur Verwendung von Interrupts finden Sie im Abschnitt zu den Kommunikations- und Interruptoperationen.

Hinweis

Es kann ein schwerer Fehler auftreten, wenn Sie im Interruptprogramm für externes Rücksetzen einen neuen aktuellen Wert laden oder den schnellen Zähler deaktivieren und anschließend wieder freigeben.

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 6-30 beschreibt die Statusbits für die schnellen Zähler.

Tipp

Die Statusbits 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.

Tabelle 6-30 Statusbits für HSC0, HSC1, HSC2, HSC3, HSC4 und HSC5

HSC0	HSC1	HSC2	HSC3	HSC4	HSC5	Beschreibung
SM36.0	SM46.0	SM56.0	SM136.0	SM146.0	SM156.0	Nicht verwendet.
SM36.1	SM46.1	SM56.1	SM136.1	SM146.1	SM156.1	Nicht verwendet.
SM36.2	SM46.2	SM56.2	SM136.2	SM146.2	SM156.2	Nicht verwendet.
SM36.3	SM46.3	SM56.3	SM136.3	SM146.3	SM156.3	Nicht verwendet.
SM36.4	SM46.4	SM56.4	SM136.4	SM146.4	SM156.4	Nicht verwendet.
SM36.5	SM46.5	SM56.5	SM136.5	SM146.5	SM156.5	Statusbit aktuelle Zählrichtung: 0 = Rückwärtszählen 1 = Vorwärtszählen
SM36.6	SM46.6	SM56.6	SM136.6	SM146.6	SM156.6	Statusbit aktueller Wert gleich voreingestellter Wert: 0 = Ungleich 1 = Gleich
SM36.7	SM46.7	SM56.7	SM136.7	SM146.7	SM156.7	Statusbit aktueller Wert ist größer als voreingestellter Wert: 0 = Kleiner als oder gleich 1 = Größer als

Beispiele für Initialisierungssequenzen für schnelle Zähler

HSC1 wird in den folgenden Beschreibungen zu Initialisierung und Bearbeitungsreihenfolge als Beispiel herangezogen. 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, 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.

Tipp

Die im folgenden 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 SMB47 entsprechend einstellen und anschließend die Operation HSC ausführen.

Initialisieren der Zählerarten 0, 1 und 2

Gehen Sie folgendermaßen vor, um HSC1 als Einphasen-Vorwärts-/Rückwärtszähler mit interner Richtungssteuerung (Zählerart 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 SMB47 mit den gewünschten Einstellungen. Beispiel:

SMB47 = 16#F8	*Bewirkt folgende Ergebnisse:* Aktiviert den Zähler. Schreibt einen neuen aktuellen Wert. Schreibt einen neuen voreingestellten Wert. Stellt Vorwärtszählen ein. Setzt die Start- und Rücksetzeingänge auf Aktiv High.

3. Führen Sie die Operation HDEF aus, wobei der Eingang HSC auf 1 gesetzt ist und der Eingang MODE auf einen der folgenden Werte gesetzt ist: 0 für kein externes Rücksetzen und Starten, 1 für externes Rücksetzen und kein Starten oder 2 für externes Rücksetzen und Starten.
4. Laden Sie den gewünschten aktuellen Wert in SMD48 (Doppelwortwert). (Wenn Sie den Wert 0 laden, setzen Sie den Merker zurück).
5. Laden Sie den gewünschten voreingestellten Wert in SMD52 (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. Eine ausführlichen Beschreibung der Interruptverarbeitung finden Sie in dem Abschnitt zu Interruptoperationen.
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 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 Einphasen-Vorwärts-/Rückwärtszähler mit externer Richtungssteuerung (Zählerart 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 SMB47 mit den gewünschten Einstellungen. Beispiel:

SMB47 = 16#F8	*Bewirkt folgende Ergebnisse:* Aktiviert den Zähler. Schreibt einen neuen aktuellen Wert. Schreibt einen neuen voreingestellten Wert. Stellt Vorwärtszählen ein. Setzt die Start- und Rücksetzeingänge auf Aktiv High.

3. Führen Sie die Operation HDEF aus, wobei der Eingang HSC auf 1 gesetzt ist und der Eingang MODE auf einen der folgenden Werte gesetzt ist: 3 für kein externes Rücksetzen und Starten, 4 für externes Rücksetzen und kein Starten oder 5 für externes Rücksetzen und Starten.
4. Laden Sie den gewünschten aktuellen Wert in SMD48 (Doppelwortwert). (Wenn Sie den Wert 0 laden, setzen Sie den Merker zurück).
5. Laden Sie den gewünschten voreingestellten Wert in SMD52 (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. Eine ausführlichen Beschreibung der Interruptverarbeitung finden Sie in dem Abschnitt zu Interruptoperationen.
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 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 Zweiphasen-Vorwärts-/Rückwärtszähler mit Vorwärts-/Rückwärts-Taktgebern (Zählerart 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 SMB47 mit den gewünschten Einstellungen. Beispiel:

SMB47 = 16#F8	*Bewirkt folgende Ergebnisse:* Aktiviert den Zähler. Schreibt einen neuen aktuellen Wert. Schreibt einen neuen voreingestellten Wert. Stellt Vorwärtszählen ein. Setzt die Start- und Rücksetzeingänge auf Aktiv High.

3. Führen Sie die Operation HDEF aus, wobei der Eingang HSC auf 1 gesetzt ist und der Eingang MODE auf einen der folgenden Werte gesetzt ist: 6 für kein externes Rücksetzen und Starten, 7 für externes Rücksetzen und kein Starten oder 8 für externes Rücksetzen und Starten.
4. Laden Sie den gewünschten aktuellen Wert in SMD48 (Doppelwortwert). (Wenn Sie den Wert 0 laden, setzen Sie den Merker zurück).
5. Laden Sie den gewünschten voreingestellten Wert in SMD52 (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. (siehe Abschnitt zu Interrupts).

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 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 (für Zählerart 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 SMB47 mit den gewünschten Einstellungen.

   Beispiel (einfache Zählgeschwindigkeit):
   SMB47 = 16#FC *Bewirkt folgende Ergebnisse:*
   Aktiviert den Zähler.
   Schreibt einen neuen aktuellen Wert.
   Schreibt einen neuen voreingestellten Wert.
   Stellt Vorwärtszählen ein.
   Setzt die Start- und Rücksetzeingänge auf Aktiv High.

   Beispiel (vierfache Zählgeschwindigkeit):
   SMB47 = 16#F8 *Bewirkt folgende Ergebnisse:*
   Aktiviert den Zähler.
   Schreibt einen neuen aktuellen Wert.
   Schreibt einen neuen voreingestellten Wert.
   Stellt Vorwärtszählen ein.
   Setzt die Start- und Rücksetzeingänge auf Aktiv High.
3. Führen Sie die Operation HDEF aus, wobei der Eingang HSC auf 1 gesetzt ist und der Eingang MODE auf einen der folgenden Werte gesetzt ist: 9 für kein externes Rücksetzen und Starten, 10 für externes Rücksetzen und kein Starten oder 11 für externes Rücksetzen und Starten.
4. Laden Sie den gewünschten aktuellen Wert in SMD48 (Doppelwortwert). (Wenn Sie den Wert 0 laden, setzen Sie den Merker zurück).
5. Laden Sie den gewünschten voreingestellten Wert in SMD52 (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. Eine ausführlichen Beschreibung der Interruptfreigabe (ENI) finden Sie in dem Abschnitt zu Interruptoperationen.
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 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 Einphasen-Zähler mit interner Richtungssteuerung (Zählerart 0, 1 oder 2) einen Richtungswechsel zu konfigurieren:

1. Laden Sie SM47, um die gewünschte Richtung einzustellen.

SMB47 = 16#90	Aktiviert den Zähler. Setzt die Zählrichtung des HSC auf Rückwärtszählen.
SMB47 = 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)

Wenn Sie den aktuellen Wert ändern, wird der Zähler automatisch gesperrt. Solange er gesperrt ist, wird weder gezählt, noch können Interrupts auftreten.

Gehen Sie folgendermaßen vor, um den aktuellen Wert von HSC1 (beliebige Zählerart) zu ändern:

1. Laden Sie SMB47, um den gewünschten aktuellen Wert einzugeben.

SMB47 = 16#C0	Aktiviert den Zähler. Schreibt einen neuen aktuellen Wert.

2. Laden Sie den gewünschten aktuellen Wert in SMD48 (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 SMB47, um den gewünschten voreingestellten Wert einzugeben.

SMB47 = 16#A0	Aktiviert den Zähler. Schreibt einen neuen voreingestellten Wert.

2. Laden Sie den gewünschten voreingestellten Wert in SMD52 (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:

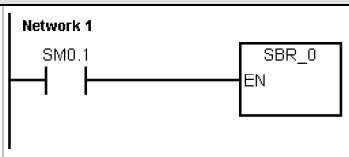
1. Laden Sie SMB47, um den schnellen Zähler zu sperren.

SMB47 = 16#00	Sperrt den Zähler.

2. Führen Sie die Operation HSC aus, damit die S7-200 den Zähler HSC1 sperrt.

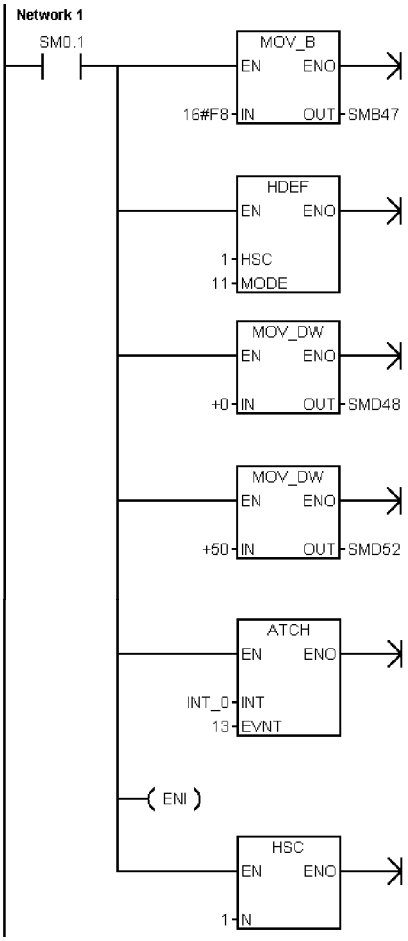
Beispiel: Operation Schnellen Zähler aktivieren

MAIN



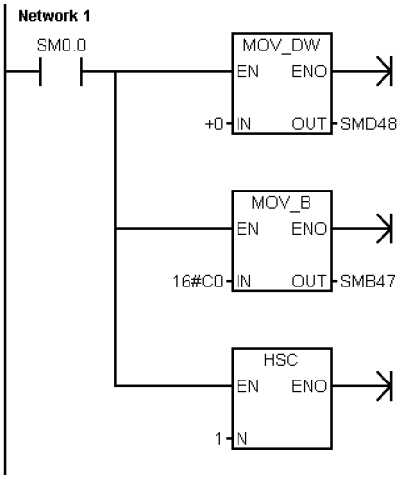

```
Netzwerk 1      //Im ersten Zyklus SBR_0 aufrufen.
LD      SM0.1
CALL    SBR0
```

SBR 0




```
Netzwerk 1      //Im ersten Zyklus, HSC1 einrichten:
                //1.  Aktiviert den Zähler.
                //    - Neuen aktuellen Wert schreiben.
                //    - Neuen voreingestellten Wert schreiben.
                //    - Anfangszählrichtung auf Vorwärtszählen setzen.
                //    - Für Start- und Rücksetzeingänge Aktiv High einstellen.
                //    - Vierfache Zählgeschwindigkeit wählen.
                //2. HSC1 als A/B-Zähler einrichten
                //   mit Rücksetz- und Starteingang.
                //3. Aktuellen Wert von HSC1 zurücksetzen.
                //4. Voreingestellten Wert von HSC1 auf 50 setzen.
                //5. Wenn HSC1: aktueller Wert = voreingestellter Wert,
                //   Ereignis 13 Interruptprogramm INT_0 zuordnen.
                //6. Alle Interruptereignisse freigeben.
                //7. HSC1 programmieren.
LD      SM0.1
MOVB    16#F8, SMB47
HDEF    1, 11
MOVD    +0, SMD48
MOVD    +50, SMD52
ATCH    INT_0, 13
ENI
HSC     1
```

INT 0




```
Netzwerk 1      //HSC1 programmieren:
                //1. Aktuellen Wert von HSC1 zurücksetzen.
                //2. Nur neuen aktuellen Wert schreiben
                //   und HSC1 aktiviert lassen.
LD      SM0.0
MOVD    +0, SMD48
MOVB    16#C0, SMB47
HSC     1
```

Operation Impulsausgabe

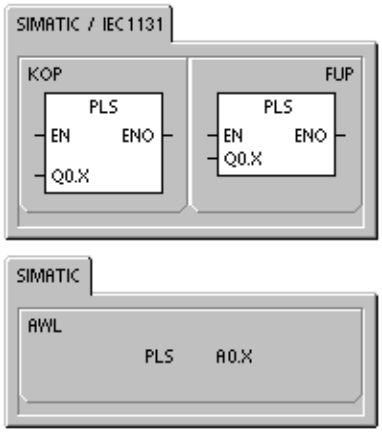
Positionier-steuerung

Mit der Operation Impulsausgabe (PLS) steuern Sie die Funktionen Impulsfolge (PTO) und Impulsdauermodulation (PWM), die für die schnellen Ausgänge (A0.0 und A0.1) zur Verfügung stehen. Mit dem Positionier-Assistent können Sie die Impulsausgabe einrichten.

Die Funktion Impulsfolge (PTO) stellt einen Rechteckausgang (50 % relative Einschaltdauer) zur Verfügung, wobei der Anwender die Zykluszeit und die Anzahl der Impulse angeben kann.

Die Funktion Impulsdauermodulation (PWM) stellt einen fortlaufenden Ausgang mit variabler relativer Einschaltdauer zur Verfügung, wobei der Anwender die Zykluszeit und die Impulsdauer angeben kann.

Die S7-200 verfügt über zwei PTO/PWM-Generatoren, die entweder eine schnelle Impulsfolge oder eine impulsdauermodulierte Wellenform liefern. Ein Generator ist dem Digitalausgang A0.0 zugewiesen, der andere Generator ist dem Digitalausgang A0.1 zugewiesen. Ein bestimmter Sondermerker (SM) speichert für jeden Generator die folgenden Daten: ein Steuerbyte (8-Bit-Wert), einen Impulszählwert (ein vorzeichenloser 32-Bit-Wert) und einen Zykluszeit- und Impulsdauerwert (ein vorzeichenloser 16-Bit-Wert).



Die PTO/PWM-Generatoren und das Prozessabbild verwenden die Ausgänge A0.0 und A0.1 gemeinsam. Ist eine Funktion PTO oder PWM an A0.0 oder A0.1 aktiv, steuert der PTO/PWM-Generator den Ausgang, wodurch dieser nicht anderweitig genutzt werden kann. Die Ausgangswellenform wird nicht durch den Zustand des Prozessabbilds, durch den geforcten Wert des Ausgangs oder durch eine direkte Ausgangsoperation beeinflusst. Ist der PTO/PWM-Generator nicht aktiv, steuert das Prozessabbild den Ausgang. Das Prozessabbild legt den anfänglichen und den letzten Zustand der Ausgangswellenform fest, so dass die Wellenform bei Pegel High oder Low beginnt und endet.

Tabelle 6-31 Gültige Operanden für die Operation Impulsausgabe

Eingänge/Ausgänge	Datentypen	Operanden
Q0.x	WORD	Konstante: 0 (= A0.0) *oder* 1 (= A0.1)

Tipp

Vor Aktivieren einer der Funktionen PTO oder PWM müssen Sie den Wert im Prozessabbild für A0.0 und A0.1 auf 0 setzen.

Die Voreinstellung für die Steuerbits, die Zykluszeit, die Impulsdauer und den Impulszählwert ist 0.

Die Ausgänge PTO/PWM benötigen eine Mindestlast von 10 % der Nennlast, um saubere Übergänge (ein/aus und aus/ein) erzeugen zu können.

Programmierung Tips

Die Tipps für die Programmierung auf der Dokumentations-CD enthalten Programme, in denen die Operation PLS für die Funktionen PTO/PWM verwendet wird (siehe Tip 7, Tip 22, Tip 23, Tip 30 und Tip 50).

Impulsfolge (PTO)

PTO stellt einen Rechteckausgang (50 % relative Einschaltdauer) für eine bestimmte Anzahl von Impulsen und eine festgelegte Zykluszeit zur Verfügung (siehe Bild 6-29). Die Funktion PTO kann entweder eine einzelne Impulsfolge oder mehrere Impulsfolgen (über ein Impulsprofil) erzeugen. Sie geben die Anzahl der Impulse und die Zykluszeit (in Mikro- oder Millisekunden) an:

- Anzahl Impulse: 1 bis 4.294.967.295
- Zykluszeit: 50 µs bis 65.535 µs oder 2 ms bis 65.535 ms.

Wenn Sie für die Zykluszeit eine ungerade Anzahl Mikro- oder Millisekunden angeben (z.B. 75 ms), führt dies zu einer gewissen Verzerrung des Betriebsspiels.

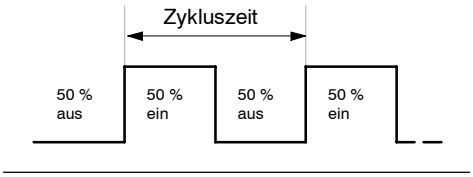


Bild 6-29 Impulsfolge (PTO)

Die Tabelle 6-32 führt Einschränkungen für den Impulszählwert und die Zykluszeit auf.

Tabelle 6-32 Impulszählwert und Zykluszeit bei der Funktion PTO

Impulszählwert/Zykluszeit	Reaktion
Zykluszeit < 2 Zeiteinheiten	Die Voreinstellung der Zykluszeit ist 2 Zeiteinheiten.
Impulszählwert = 0	Die Voreinstellung des Impulszählwerts ist 1 Impuls.

Die Funktion PTO lässt die Aneinanderreihung von Impulsfolgen (Pipeline) zu. Ist die aktive Impulsfolge beendet, beginnt sofort die Ausgabe einer neuen Impulsfolge. Dadurch ist die Kontinuität der Impulsfolgeausgabe gewährleistet.

Einzel-Segment-Pipeline an PTO-Impulsen

Bei einer Einzel-Segment-Pipeline müssen Sie die Sondermerker für die nächste Impulsfolge aktualisieren. Ist das erste PTO-Segment gestartet, müssen Sie sofort die Sondermerker für die zweite Wellenform ändern und die Operation PLS erneut ausführen. Die Attribute der zweiten Impulsfolge werden in einer Pipeline angeordnet, bis die erste Impulsfolge beendet ist. Es kann immer nur ein Eintrag in der Pipeline gespeichert werden. Ist die erste Impulsfolge beendet, beginnt die Ausgabe der zweiten Wellenform und die Pipeline ist wieder verfügbar für die Angabe einer weiteren Impulsfolge. Sie können dann auf die gleiche Art die Attribute für die nächste Impulsfolge definieren.

Die Übergänge zwischen den Impulsfolgen sind glatt, es sei denn, die Zeitbasis wird verändert oder die aktive Impulsfolge wird beendet, bevor die Einrichtung einer neuen Impulsfolge von der Operation PLS erfasst wird.

Multi-Segment-Pipeline an PTO-Impulsen

Bei einer Multi-Segment-Pipeline liest die S7-200 automatisch die Eigenschaften der einzelnen Impulsfolge-Segmente aus einer Profiltabelle im Variablenspeicher. Die in dieser Betriebsart verwendeten Sondermerker sind das Steuerbyte, das Statusbyte und der Anfangsversatz im Variablenspeicher der Profiltabelle (SMW168 oder SMW178). Die Zeitbasis kann in Mikrosekunden oder in Millisekunden angegeben werden, doch die Angabe bezieht sich auf alle Zykluszeitwerte in der Profiltabelle und kann nicht geändert werden, wenn das Profil aktiv ist. Die Ausführung der Operation PLS startet den Multi-Segment-Betrieb.

Jeder Segmenteintrag umfasst 8 Bytes und besteht aus einem Zykluszeitwert (16 Bit), einem Zykluszeitdelta (16 Bit) und einem Impulszählwert (32 Bit). Tabelle 6-33 zeigt das Format der Profiltabelle. Sie können die Zykluszeit automatisch erhöhen oder verringern, indem Sie für jeden Impuls einen bestimmten Wert programmieren. Ein positiver Wert im Feld für das Zykluszeitdelta erhöht die Zykluszeit, ein negativer Wert im Feld für das Zykluszeitdelta verringert die Zykluszeit und der Wert 0 führt zu einer unveränderten Zykluszeit.

Während der Bearbeitung des PTO-Profils ist die Nummer des aktuell aktiven Segments in SMB166 (bzw. in SMB176) verfügbar.

Tabelle 6-33 Format der Profiltabelle für Multi-Segment-Impulsfolgen

Byte-Versatz	Bitmuster für Sieben-Segment-Anzeige erzeugen	Beschreibung der Tabelleneinträge
0		Anzahl der Segmente: 1 bis 255[1]
1	Nr. 1	Anfängliche Zykluszeit (2 bis 65.535 Einheiten der Zeitbasis)
3		Zykluszeitdelta pro Impuls (Wert mit Vorzeichen) (-32.768 bis 32.767 Einheiten der Zeitbasis)
5		Impulszählwert (1 bis 4.294.967.295)
9	Nr. 2	Anfängliche Zykluszeit (2 bis 65.535 Einheiten der Zeitbasis)
11		Zykluszeitdelta pro Impuls (Wert mit Vorzeichen) (-32.768 bis 32.767 Einheiten der Zeitbasis)
13		Impulszählwert (1 bis 4.294.967.295)
(Fortsetzung)	Nr. 3	(Fortsetzung)

1 Wenn Sie für die Anzahl der Segmente den Wert 0 eingeben, tritt ein schwerer Fehler auf. Es wird keine PTO-Ausgabe erzeugt.

Impulsdauermodulation (PWM)

Die Funktion Impulsdauermodulation bietet Ihnen eine feste Zykluszeit mit variabler relativer Einschaltdauer (siehe Bild 6-30). Sie können die Zykluszeit und die Impulsdauer in Mikro- oder Millisekunden angeben.

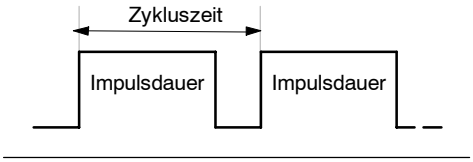


Bild 6-30 Impulsdauermodulation (PWM)

- ❏ Zykluszeit: 50 μs bis 65.535 μs oder 2 ms bis 65.535 ms
- ❏ Impulsdauer: 0 μs bis 65.535 μs oder 0 ms bis 65.535 ms.

Wie Tabelle 6-34 zeigt, ist der Ausgang dauernd eingeschaltet, wenn Sie die Impulsdauer gleich der Zykluszeit einstellen (wodurch sich ein Betriebsspiel von 100 Prozent ergibt). Wenn Sie für die Impulsdauer den Wert 0 einstellen (wodurch sich ein Betriebsspiel von 0 Prozent ergibt), wird der Ausgang ausgeschaltet.

Tabelle 6-34 Impulsdauer und Zykluszeit und Reaktionen der Funktion PWM

Impulsdauer/Zykluszeit	Reaktion
Impulsdauer >= Zykluszeitwert	Das Betriebsspiel beträgt 100 %: der Ausgang ist ständig eingeschaltet.
Impulsdauer = 0	Das Betriebsspiel beträgt 0 %: der Ausgang ist ausgeschaltet.
Zykluszeit < 2 Zeiteinheiten	Die Voreinstellung der Zykluszeit ist zwei Zeiteinheiten.

Sie können die Eigenschaften einer PWM-Wellenform auf zwei Arten ändern:

- ❏ Synchrones Aktualisieren: Ist keine Änderung der Zeitbasis erforderlich, kann synchron aktualisiert werden. Beim synchronen Aktualisieren wird die Wellenform an der Zyklusgrenze geändert, so dass ein glatter Übergang stattfindet.
- ❏ Asynchrones Aktualisieren: Bei der Operation PWM wird die Impulsdauer typischerweise verändert, während die Zykluszeit konstant bleibt, so dass keine Änderungen der Zeitbasis erforderlich sind. Ist es jedoch aus anderen Gründen erforderlich, die Zeitbasis des PTO/PWM-Generators zu ändern, dann wird asynchron aktualisiert. Beim asynchronen Aktualisieren wird der PTO/PWM-Generator kurzzeitig deaktiviert, und zwar asynchron zur PWM-Wellenform. Dies kann unerwünschtes Flattern im gesteuerten Gerät hervorrufen. Deshalb sind synchrone PWM-Aktualisierungen empfehlenswert. Wählen Sie eine Zeitbasis, die sich für alle Werte der Zykluszeit eignet.

Tipp

Das PWM-Bit für die Aktualisierungsmethode (SM67.4 oder SM77.4) im Steuerbyte gibt die Art der Aktualisie-
rung an, wenn die Operation PLS ausgeführt wird, um die Änderungen vorzunehmen.

Hat sich die Zeitbasis geändert, wird in jedem Fall asynchron aktualisiert, unabhängig vom Zustand des Bits für die PWM-Aktualisierungsmethode.

Konfigurieren und Steuern der Funktionen PTO/PWM mit Sondermerkern

Die Operation PLS liest die in den Sondermerkern gespeicherten Daten und programmiert den PTO/PWM-Generator entsprechend. SMB67 steuert PTO 0 oder PWM 0 und SMB77 steuert PTO 1 oder PWM 1. Tabelle 6-35 beschreibt die Register, mit denen die Funktionen PTO und PWM gesteuert werden. Mit Hilfe der Tabelle 6-36 können Sie schnell den Wert festlegen, den Sie in dem PTO/PWM-Steuerungsregister ablegen müssen, um die gewünschte Operation aufzurufen.

Sie können die Eigenschaften der PTO- oder PWM-Wellenform ändern, indem Sie die Sondermerker (einschließlich des Steuerbyte) ändern und anschließend die Operation PLS ausführen. Sie können das Erzeugen von PTO- oder PWM-Wellenformen jederzeit deaktivieren, indem Sie den Wert 0 in das PTO/PWM-Freigabebit im Steuerbyte (SM67.7 oder SM77.7) schreiben und anschließend die Operation PLS ausführen.

Das PTO-Leerlaufbit im Statusbyte (SM66.7 bzw. SM76.7) dient dazu, den Abschluss der programmierten Impulsfolge anzuzeigen. Außerdem kann nach Abschluss einer Impulsfolge ein Interruptprogramm aufgerufen werden. (Lesen Sie hierzu die Beschreibungen der Interruptoperationen und der Kommunikationsoperationen.) Wenn Sie mit der Multi-Segment-Funktion arbeiten, wird das Interruptprogramm nach Beendigung der Profiltabelle aufgerufen.

Die folgenden Bedingungen setzen SM66.4 (oder SM76.4) und SM66.5 (oder SM76.5):

- Wenn Sie ein Zykluszeitdelta angeben, das nach einer Anzahl von Impulsen zu einer unzulässigen Zykluszeit führt, wird eine arithmetische Überlaufbedingung erzeugt, die die PTO-Funktion beendet und das Bit für Deltaberechnungsfehler (SM66.4 oder SM76.4) auf 1 setzt. Der Ausgang wird wieder über das Prozessabbild gesteuert.
- Wenn Sie ein PTO-Profil, das in Bearbeitung ist, manuell beenden (deaktivieren), wird das Bit für Anwenderabbruch (SM66.5 oder SM76.5) auf 1 gesetzt.
- Wenn Sie versuchen, in eine volle Pipeline zu laden, wird das Bit für PTO-Überlauf (SM66.6 oder SM76.6) auf 1 gesetzt. Wenn Sie einen späteren Überlauf erkennen möchten, müssen Sie dieses Bit manuell löschen, nachdem der Überlauf erkannt wurde. Durch den Wechsel in den Betriebszustand RUN wird dieses Bit auf 0 gesetzt.

Tipp

Wenn Sie einen neuen Impulszählwert (SMD72 bzw. SMD82), eine Impulsdauer (SMW70 bzw. SMW80) oder eine Zykluszeit (SMW68 bzw. SMW78) laden, müssen Sie auch die entsprechenden Aktualisierungs-bits im Steuerungsregister vor Ausführung der Operation PLS setzen. Bei Multi-Segment-Impulsfolgen müssen Sie auch den Anfangsversatz (SMW168 bzw. SMW178) der Profiltabelle sowie die Werte der Profiltabelle laden, bevor Sie die Operation PLS ausführen.

Tabelle 6-35 Sondermerker für die PTO/PWM-Steuerregister

A0.0	A0.1	Statusbits		
SM66.4	SM76.4	PTO-Profil abgebr.n (Delta-Berechnungsfehler):	0 = kein Fehler	1 = abgebrochen
SM66.5	SM76.5	PTO-Profil auf Anwenderbefehl abgebrochen:	0 = nicht abgebr.	1 = abgebrochen
SM66.6	SM76.6	PTO-Pipeline Überlauf/Unterlauf:	0 = kein Überlauf	1 = Überlauf/Unterlauf
SM66.7	SM76.7	PTO-Leerlauf:	0 = Bearbeitung	1 = PTO-Leerlauf
A0.0	**A0.1**	**Steuerbits**		
SM67.0	SM77.0	PTO/PWM: Zykluszeit aktualisieren:	0 = keine Aktualis.	1 = Zykluszeit aktualisieren
SM67.1	SM77.1	PWM: Zeitwert der Impulsdauer aktualisieren: sieren	0 = keine Aktualis.	1 = Impulsdauer aktuali
SM67.2	SM77.2	PTO: Impulszählwert aktualisieren: alisieren	0 = keine Aktualis.	1 = Impulszählwert aktu-
SM67.3	SM77.3	PTO/PWM: Zeitbasis wählen:	0 = 1 μs/Takt	1 = 1 ms/Takt
SM67.4	SM77.4	PWM: Aktualisierungsmethode:	0 = asynchron	1 = synchron
SM67.5	SM77.5	PTO: Einzel-/Multi-Segment-Funktion:	0 = Einzel	1 = Multi
SM67.6	SM77.6	PTO/PWM: Funktion wählen:	0 = PTO	1 = PWM
SM67.7	SM77.7	PTO/PWM: Freigabe:	0 = sperren	1 = freigeben
A0.0	**A0.1**	**Sonstige PTO/PWM-Register**		
SMW68	SMW778	PTO/PWM: Zykluszeit	Bereich: 2 bis 65.535	
SMW70	SMW80	PWM: Impulsdauer	Bereich: 0 bis 65.535	
SMD72	SMD82	PTO: Impulszählwert	Bereich: 1 bis 4.294.967.295	
SMB166	SMB176	Anzahl der Segment in Bearbeitung	Nur PTO-Multi-Segment-Funktion	
SMW168	SMW178	Anfangsadresse der Profiltabelle (Byteversatz von V0)	Nur PTO-Multi-Segment-Funktion	

Tabelle 6-36 Referenz PTO/PWM-Steuerbyte

Steuerungs-register (Hexadezi-malwert)		Ergebnis der Operation PLS						
	Frei-gabe	Funk-tion	PTO-Segment	PWM-Aktualisierung	Zeitbasis	Impuls-zählwert	Impuls-dauer	Zyklus-zeit
16#81	Ja	PTO	Einzel		1 μs/Zyklus			Laden
16#84	Ja	PTO	Einzel		1 μs/Zyklus	Laden		
16#85	Ja	PTO	Einzel		1 μs/Zyklus	Laden		Laden
16#89	Ja	PTO	Einzel		1 ms/Zyklus			Laden
16#8C	Ja	PTO	Einzel		1 ms/Zyklus	Laden		
16#8D	Ja	PTO	Einzel		1 ms/Zyklus	Laden		Laden
16#A0	Ja	PTO	Multi		1 μs/Zyklus			
16#A8	Ja	PTO	Multi		1 ms/Zyklus			
16#D1	Ja	PWM		Synchron	1 μs/Zyklus			Laden
16#D2	Ja	PWM		Synchron	1 μs/Zyklus		Laden	
16#D3	Ja	PWM		Synchron	1 μs/Zyklus		Laden	Laden
16#D9	Ja	PWM		Synchron	1 ms/Zyklus			Laden
16#DA	Ja	PWM		Synchron	1 ms/Zyklus		Laden	
16#DB	Ja	PWM		Synchron	1 ms/Zyklus		Laden	Laden

Berechnen der Werte für die Profiltabelle

Die Funktion Multi-Segment-Pipeline des PTO/PWM-Generators kann in vielen Anwendungen nützlich sein, ganz besonders bei der Steuerung von Schrittmotoren.

Sie können z.B. die Operation PTO mit einem Impulsprofil einsetzen, um einen Schrittmotor mit den einfachen Sequenzen Hochfahren, Betrieb und Herunterfahren oder mit sehr viel komplexeren Sequenzen zu steuern. Hierzu definieren Sie ein Impulsprofil, das aus maximal 255 Segmenten besteht, wobei jedes Segment einer Funktion entspricht, also z.B. Hochfahren, Betrieb und Herunterfahren.

Bild 6-31 zeigt ein Beispiel für eine Profiltabelle, die erforderlich ist, um eine Ausgangswellenform zu erzeugen, die einen Schrittmotor (Segment 1) beschleunigt, den Motor mit konstanter Drehzahl betreibt (Segment 2) und den Motor bremst (Segment 3).

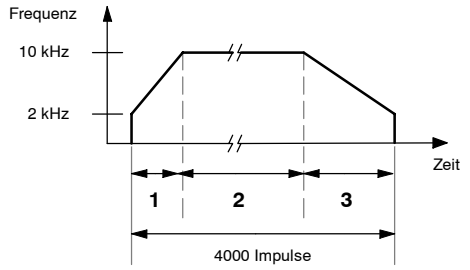


Bild 6-31 Frequenz-/Zeitdiagramm

Zu diesem Beispiel: Die Impulsfrequenz am Anfang und am Ende beträgt 2 kHz, die maximale Impulsfrequenz beträgt 10 kHz und es sind 4000 Impulse erforderlich, um die gewünschte Anzahl Motorumdrehungen zu erreichen. Da die Werte der Profiltabelle nach dem Intervall (Zykluszeit) anstelle der Frequenz angegeben werden, müssen Sie die angegebene Frequenz in Zykluszeitwerte umwandeln. So beträgt die Anfangs- und Endzykluszeit 500 µs und die Zykluszeit, die der maximalen Frequenz entspricht, beträgt 100 µs. Im Beschleunigungsabschnitt des Ausgangsprofils muss die maximale Impulsfrequenz nach ungefähr 200 Impulsen erreicht sein. Außerdem soll der Abschnitt des Abbremsens des Profils nach ungefähr 400 Impulsen beendet sein.

Der Wert des Zykluszeitdeltas für ein bestimmtes Segment, mit dem der PTO/PWM-Generator die Zykluszeit der einzelnen Impulse einstellt, kann anhand einer einfachen Formel berechnet werden:

Zykluszeitdelta für ein Segment = | ZZ_Ende_{Seg} - ZZ_Anfang_{Seg} | / $Anzahl_{Seg}$

Erklärung:
ZZ_Ende_{Seg} = Endzykluszeit des Segments
ZZ_Anfang_{Seg} = Anfangszykluszeit des Segments
$Anzahl_{Seg}$ = Anzahl Impulse in diesem Segment

Mit dieser Formel können Sie die Zykluszeitdeltawerte für die Beispielanwendung berechnen:

Segment 1 (Beschleunigen):
Zykluszeitdelta = -2

Segment 2 (konstante Drehzahl):
Zykluszeitdelta = 0

Segment 3 (Abbremsen):
Zykluszeitdelta = 1

Tabelle 6-37 führt die Werte zum Erzeugen der Beispiel-Wellenform auf (vorausgesetzt, die Profiltabelle befindet sich im Variablenspeicher mit Beginn an V500). Sie können Operationen in Ihr Programm aufnehmen, um diese Werte in den Variablenspeicher zu laden, oder Sie können die Werte des Profils im Datenbaustein definieren.

Tabelle 6-37 Werte der Profiltabelle

Adresse	Wert	Beschreibung	
VB500	3	Anzahl Segmente gesamt	
VW501	500	Anfangszykluszeit	Segment 1
VW503	-2	Anfangszykluszeitdelta	
VD505	200	Anzahl Impulse	
VW509	100	Anfangszykluszeit	Segment 2
VW511	0	Zykluszeitdelta	
VD513	3400	Anzahl Impulse	
VW517	100	Anfangszykluszeit	Segment 3
VW519	1	Zykluszeitdelta	
VD521	400	Anzahl Impulse	

Sie müssen die Zykluszeit des letzten Impulses eines Segments kennen, um festzustellen, ob die Übergänge zwischen den Wellenformen sauber sind. Sofern das Zykluszeitdelta nicht 0 ist, müssen Sie die Zykluszeit des letzten Impulses eines Segments berechnen, weil dieser Wert nicht im Profil angegeben wird. Berechnen Sie die Zykluszeit des letzten Impulses eines Segments mit folgender Formel:

Zykluszeit des letzten Impulses eines Segments = ZZ_Anfang_{Seg} + ($Delta_{Seg}$ * ($Anzahl_{Seg}$ - 1))

Erklärung:
ZZ_Anfang_{Seg} = Anfangszykluszeit des Segments

$Delta_{seg}$ = Zykluszeitdelta für dieses Segment

$Anzahl_{Seg}$ = Anzahl Impulse in diesem Segment

Dieses stark vereinfachte Beispiel dient zur Erklärung des Prinzips, doch tatsächliche Anwendungen erfordern häufig komplexere Profile. Denken Sie daran, dass das Zykluszeitdelta nur als ganze Zahl in Mikrosekunden oder in Millisekunden angegeben werden kann und dass die Zykluszeit bei jedem Impuls verändert wird.

Daraus resultiert, dass das Zykluszeitdelta für ein bestimmtes Segment gegebenenfalls iterativ berechnet werden sollte. Es kann Flexibilität für den Wert der Endzykluszeit oder die Anzahl der Impulse für ein bestimmtes Segment erforderlich sein.

Die Dauer eines bestimmten Profilsegments kann beim Festlegen der richtigen Werte für die Profiltabelle nützlich sein. Berechnen Sie die Zeit, die benötigt wird, um ein bestimmtes Segment eines Profils zu bearbeiten, anhand folgender Formel:

Dauer des Segments = $Anzahl_{Seg}$ * (ZZ_Anfang + (($Delta_{Seg}$/2) * ($Anzahl_{Seg}$ - 1)))

Erklärung:
$Anzahl_{Seg}$ = Anzahl Impulse in diesem Segment

ZZ_Anfang_{Seg} = Anfangszykluszeit des Segments

$Delta_{seg}$ = Zykluszeitdelta für dieses Segment

Beispiel für eine PWM-Ausgabe

Tipp

Bei der folgenden Beschreibung der PWM-Initialisierung und der Funktionsfolgen ist es empfehlenswert, die Impulsausgabe mit dem Merker des ersten Zyklus (SM0.1) zu starten. Wenn Sie ein Unterprogramm für die Initialisierung mit dem Merker des ersten Zyklus aufrufen, wird die Zykluszeit verringert, weil nachfolgende Zyklen dieses Unterprogramm nicht mehr aufrufen. (Der Merker des ersten Zyklus wird nur im ersten Zyklus bei einem Wechsel in den Betriebszustand RUN gesetzt.) Es kann in Ihrer Anwendung jedoch andere Randbedingungen geben, die es erforderlich machen, die Impulsausgabe zu starten (bzw. erneut zu starten). In dem Fall können Sie das Unterprogramm für die Initialisierung auf andere Weise aufrufen.

Initialisieren der PWM-Ausgabe

Üblicherweise initialisieren Sie die PWM-Funktion für die Impulsausgabe mit einem Unterprogramm. Sie rufen das Unterprogramm für die Initialisierung im Hauptprogramm auf. Setzen Sie mit dem Merker des ersten Zyklus (SM0.1) den Ausgang für die Funktion PWM auf 0 und rufen Sie 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.

Nachdem Sie den Aufruf des Unterprogramms für die Initialisierung im Hauptprogramm erstellt haben, gehen Sie folgendermaßen vor, um die Steuerungslogik zum Einrichten der Impulsausgabe an A0.0 im Unterprogramm für die Initialisierung zu erstellen:

1. Richten Sie das Steuerbyte ein, indem Sie einen der folgenden Werte in SMB67 laden: 16#D3 (Mikrosekunden) oder 16#DB (Millisekunden).

   Beide Werte geben die PTO/PWM-Funktion frei, wählen die Funktion PWM, stellen ein, dass die Impulsdauer und die Zykluszeit aktualisiert werden sollen, und wählen die Zeitbasis aus (Mikrosekunden bzw. Millisekunden).
2. Laden Sie einen Wortwert für die Zykluszeit in SMW68.
3. Laden Sie einen Wortwert für die Impulsdauer in SMW70.
4. Führen Sie die Operation PLS aus (damit die S7-200 den PTO/PWM-Generator programmiert).
5. Wenn Sie einen neuen Wert für das Steuerbyte für nachfolgende Änderungen der Impulsdauer laden möchten (optional), laden Sie einen der folgenden Werte in SMB67: 16#D2 (Mikrosekunden) oder 16#DA (Millisekunden).
6. Beenden Sie das Unterprogramm.

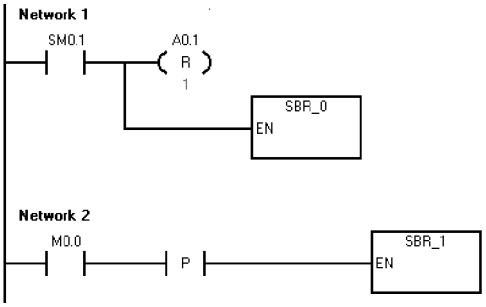
Ändern der Impulsdauer für die PWM-Ausgabe

Wenn Sie bereits einen der Werte 16#D2 oder 16#DA in SMB67 geladen haben (siehe Schritt 5. oben), können Sie die Impulsdauer für die Impulsausgabe (A0.0) in einem Unterprogramm ändern. Nachdem Sie einen Aufruf für dieses Unterprogramm erstellt haben, gehen Sie folgendermaßen vor, um die Steuerungslogik zum Ändern der Impulsdauer zu erstellen:

1. Laden Sie einen Wortwert für die neue Impulsdauer in SMW70.
2. Führen Sie die Operation PLS aus (damit die S7-200 den PTO/PWM-Generator programmiert).
3. Beenden Sie das Unterprogramm.

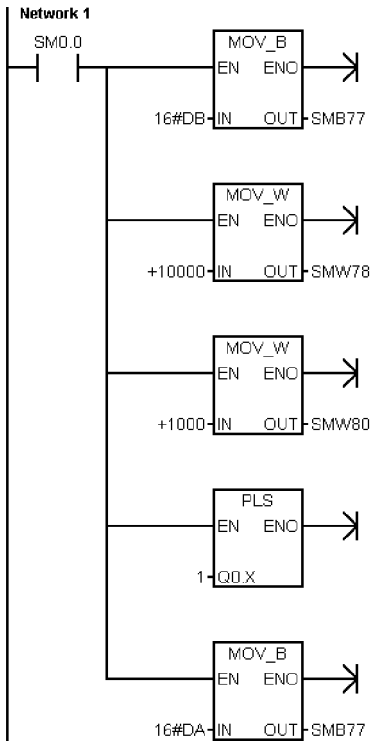
Beispiel: Impulsdauermodulation (PWM)

MAIN



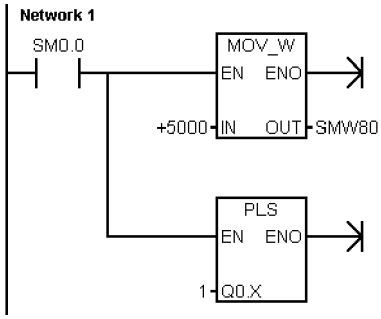

```
Netzwerk 1    //Im ersten Zykuls,
              //Bit im Prozessabbild auf Low setzen und SBR_0 aufru
              //fen.
LD      SM0.1
R       A0.1, 1
CALL    SBR0
Netzwerk 2    //M0.0 an anderer Stelle im Programm setzen,
              //um die Impulsdauer auf ein Betriebsspiel von 50 % zu
              //setzen.
LD      M0.0
EU
CALL    SBR_1
```

SBR0



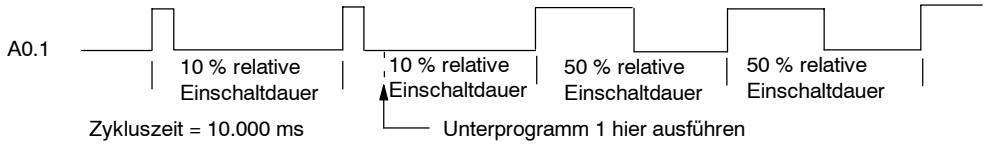

```
Netzwerk 1    //Unterprogramm 0 beginnen:
              //1. Steuerbyte einrichten.
              //  - Funktion PWM wählen.
              //  - Inkremente in ms und
              //    synchrones Aktualisieren wählen.
              //  - Laden der Impulsdauer und der
              //    Zykluszeitwerte freigeben.
              //  - Funktion PWM freigeben.
              //2. Zykluszeit auf 10.000 ms setzen.
              //3. Impulsdauer auf 1000 ms setzen.
              //4. Funktion PWM aufrufen: PLS1=>A0.1.
              //5. Steuerbyte für nachfolgende
              //   Änderungen der Zykluszeit laden.
LD      SM0.0
MOVB    16#DB, SMB77
MOVW    +10000, SMW78
MOVW    +1000, SMW80
PLS     1
MOVB    16#DA, SMB77
```

SBR1




```
Netzwerk 1    //Unterprogramm 1 beginnen:
              //Impulsdauer auf 5000 ms setzen.
              //Änderung Impulsdauer bestätigen.
LD      SM0.0
MOVW    +5000, SMW80
PLS     1
```

Impulsdiagramm



Beispiel für eine PTO-Ausgabe

Tipp

Bei der folgenden Beschreibung der PTO-Initialisierung und der Funktionsfolgen ist es empfehlenswert, die Impulsausgabe mit dem Merker des ersten Zyklus (SM0.1) zu starten. Wenn Sie ein Unterprogramm für die Initialisierung mit dem Merker des ersten Zyklus aufrufen, wird die Zykluszeit verringert, weil nachfolgende Zyklen dieses Unterprogramm nicht mehr aufrufen. (Der Merker des ersten Zyklus wird nur im ersten Zyklus bei einem Wechsel in den Betriebszustand RUN gesetzt.) Es kann in Ihrer Anwendung jedoch andere Randbedingungen geben, die es erforderlich machen, die Impulsausgabe zu starten (bzw. erneut zu starten). In dem Fall können Sie das Unterprogramm für die Initialisierung auf andere Weise aufrufen.

Initialisierung der PTO-Ausgabe für Einzel-Segment-Betrieb

Üblicherweise konfigurieren und initialisieren Sie die PTO-Funktion für die Impulsausgabe mit einem Unterprogramm. Sie rufen das Unterprogramm für die Initialisierung im Hauptprogramm auf. Setzen Sie mit dem Merker des ersten Zyklus (SM0.1) den Ausgang für die Funktion PTO auf 0 und rufen Sie 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.

Nachdem Sie den Aufruf des Unterprogramms für die Initialisierung im Hauptprogramm erstellt haben, gehen Sie folgendermaßen vor, um die Steuerungslogik zum Einrichten der Impulsausgabe an A0.0 im Unterprogramm für die Initialisierung zu erstellen:

1. Richten Sie das Steuerbyte ein, indem Sie einen der folgenden Werte in SMB67 laden: 16#85 (Mikrosekunden) oder 16#8D (Millisekunden).

   Beide Werte geben die PTO/PWM-Funktion frei, wählen die Funktion PTO, stellen ein, dass der Impulszählwert und die Zykluszeit aktualisiert werden sollen, und wählen die Zeitbasis aus (Mikrosekunden bzw. Millisekunden).

2. Laden Sie einen Wortwert für die Zykluszeit in SMW68.
3. Laden Sie einen Doppelwortwert für den Impulszählwert in SMD72.
4. (Optional) Möchten Sie nach Beendigung der Impulsfolge eine zugeordnete Operation ausführen, können Sie einen Interrupt programmieren, indem Sie das Ereignis Impulsfolge beendet (Interruptereignis 19) einem Interruptprogramm zuordnen. Verwenden Sie hierzu die Operation ATCH und führen Sie anschließend die Operation Alle Interruptereignisse freigeben (ENI) aus.
5. Führen Sie die Operation PLS aus (damit die S7-200 den PTO/PWM-Generator programmiert).
6. Beenden Sie das Unterprogramm.

Ändern der PTO-Zykluszeit (Einzel-Segment-Betrieb)

Beim Einzel-Segment-Betrieb der Funktion PTO können Sie die Zykluszeit in einem Interrupt- oder Unterprogramm ändern: Zum Ändern der PTO-Zykluszeit in einem Interrupt- oder Unterprogramm gehen Sie folgendermaßen vor, wenn Sie im Einzel-Segment-Betrieb arbeiten:

1. Setzen Sie das Steuerbyte (PTO/PWM-Funktion freigeben, PTO-Funktion auswählen, Zeitbasis einstellen und Zykluszeitwert aktualisieren), indem Sie einen der folgenden Werte in SMB67 laden: 16#81 (Mikrosekunden) oder 16#89 (Millisekunden).
2. Laden Sie einen Wortwert für die neue Zykluszeit in SMW68.
3. Führen Sie die Operation PLS aus (damit die S7-200 den PTO/PWM-Generator programmiert). Die S7-200 beendet eine Funktion PTO, die gerade bearbeitet wird, bevor die PTO-Wellenform mit der aktualisierten Zykluszeit erzeugt wird.
4. Beenden Sie das Interrupt- bzw. das Unterprogramm.

Ändern des PTO-Impulszählwerts (Einzel-Segment-Betrieb)

Beim Einzel-Segment-Betrieb der Funktion PTO können Sie den Impulszählwert in einem Interrupt- oder Unterprogramm ändern: Zum Ändern des PTO-Impulszählwerts in einem Interrupt- oder Unterprogramm gehen Sie folgendermaßen vor, wenn Sie im Einzel-Segment-Betrieb arbeiten:

1. Setzen Sie das Steuerbyte (PTO/PWM-Funktion freigeben, PTO-Funktion auswählen, Zeitbasis einstellen und Impulszählwert aktualisieren), indem Sie einen der folgenden Werte in SMB67 laden: 16#84 (Mikrosekunden) oder 16#8C (Millisekunden).
2. Laden Sie einen Doppelwortwert für den neuen Impulszählwert in SMD72.
3. Führen Sie die Operation PLS aus (damit die S7-200 den PTO/PWM-Generator programmiert). Die S7-200 beendet eine Funktion PTO, die gerade bearbeitet wird, bevor die Wellenform mit dem aktualisierten Impulszählwert erzeugt wird.
4. Beenden Sie das Interrupt- bzw. das Unterprogramm.

Ändern der PTO-Zykluszeit und des Impulszählwerts (Einzel-Segment-Betrieb)

Im Einzel-Segment-Betrieb der Funktion PTO können Sie die Zykluszeit und den Impulszählwert in einem Interrupt- oder Unterprogramm ändern. Zum Ändern der PTO-Zykluszeit und des Impulszählwerts in einem Interrupt- oder Unterprogramm gehen Sie folgendermaßen vor, wenn Sie im Einzel-Segment-Betrieb arbeiten:

1. Setzen Sie das Steuerbyte (PTO/PWM-Funktion freigeben, PTO-Funktion auswählen, Zeitbasis einstellen und Zykluszeit und Impulszählwert aktualisieren), indem Sie einen der folgenden Werte in SMB67 laden: 16#85 (Mikrosekunden) oder 16#8D (Millisekunden).
2. Laden Sie einen Wortwert für die neue Zykluszeit in SMW68.
3. Laden Sie einen Doppelwortwert für den neuen Impulszählwert in SMC72.
4. Führen Sie die Operation PLS aus (damit die S7-200 den PTO/PWM-Generator programmiert). Die S7-200 beendet eine Funktion PTO, die gerade bearbeitet wird, bevor die PTO-Wellenform mit der aktualisierten Zykluszeit und dem aktualisierten Impulszählwert erzeugt wird.
5. Beenden Sie das Interrupt- bzw. das Unterprogramm.

Initialisierung der PTO-Ausgabe für Multi-Segment-Betrieb

Üblicherweise konfigurieren und initialisieren Sie die Funktion PTO für die Impulsausgabe im Multi-Segment-Betrieb in einem Unterprogramm. Sie rufen das Unterprogramm für die Initialisierung im Hauptprogramm auf. Setzen Sie mit dem Merker des ersten Zyklus (SM0.1) den Ausgang für die Funktion PTO auf 0 und rufen Sie ein Unterprogramm auf, in dem die Initialisierung durchgeführt wird. Wenn Sie das Unterprogramm für die Initialisierung mit dem Merker des ersten Zyklus aufrufen, rufen nachfolgende Zyklen das Unterprogramm nicht mehr auf, wodurch sich die Zykluszeit verkürzt.

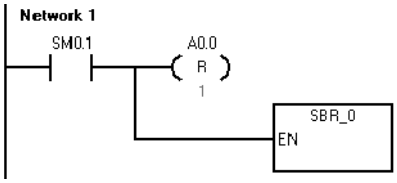
Nachdem Sie den Aufruf des Unterprogramms für die Initialisierung im Hauptprogramm erstellt haben, gehen Sie folgendermaßen vor, um die Steuerungslogik zum Einrichten der Impulsausgabe an A0.0 im Unterprogramm für die Initialisierung zu erstellen:

1. Richten Sie das Steuerbyte ein, indem Sie einen der folgenden Werte in SMB67 laden: 16#A0 (Mikrosekunden) oder 16#A8 (Millisekunden).

   Beide Werte geben die Operation PTO/PWM frei, wählen die Funktion PTO, stellen Multi-Segment-Betrieb und die Zeitbasis ein (Mikrosekunden oder Millisekunden).
2. Laden Sie einen Wortwert für den Anfangsversatz im Variablenspeicher für die Profiltabelle in SMW168.
3. Richten Sie die Segmentwerte in der Profiltabelle im Variablenspeicher ein. Achten Sie darauf, dass die Anzahl Segmente (das erste Byte der Tabelle) korrekt angegeben ist.
4. (Optional) Möchten Sie nach Beendigung des PTO-Profils eine zugeordnete Operation ausführen, können Sie einen Interrupt programmieren, indem Sie das Ereignis Impulsfolge beendet (Interruptereignis 19) einem Interruptprogramm zuordnen und die Operation Alle Interruptereignisse freigeben ausführen. Verwenden Sie hierzu die Operation ATCH und führen Sie anschließend die Operation Alle Interruptereignisse freigeben (ENI) aus.
5. Führen Sie die Operation PLS aus (damit die S7-200 den PTO/PWM-Generator programmiert).
6. Beenden Sie das Unterprogramm.

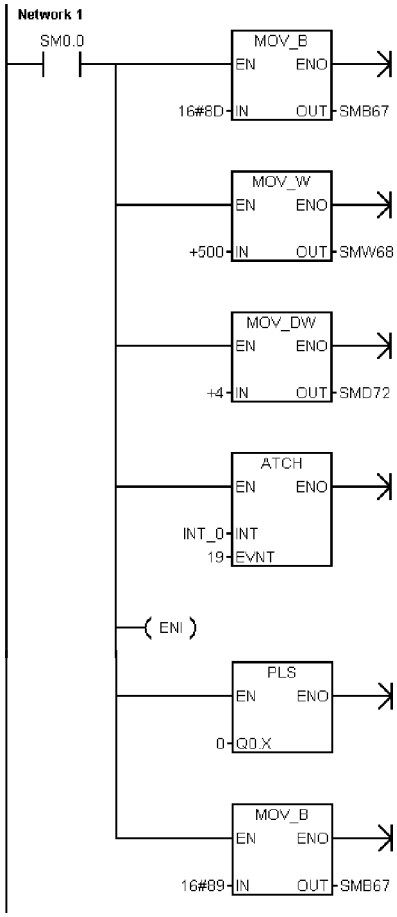
Beispiel: Einzel-Segment-Impulsfolge (PTO)

MAIN




```
Netzwerk 1      //Im ersten Zykuls,
                //Bit im Prozessabbild auf Low setzen und Unterprogramm 0
aufrufen.
LD      SM0.1
R       A0.0, 1
CALL    SBR0
```

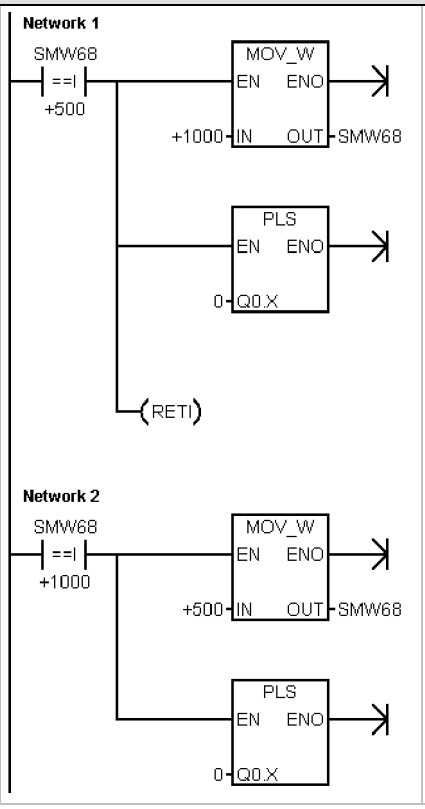
SBR0




```
Netzwerk 1      //Unterprogramm 0 beginnen: PTO einrichten.
                //1. Steuerbyte einrichten:
                //  - Funktion PTO wählen.
                //  - Einzel-Segment-Betrieb einstellen.
                //  - In ms inkrementieren.
                //  - Laden des Impulszählwerts und des
                //    Zykluszeitwerts freigeben.
                //  - Funktion PTO freigeben.
                //2. Zykluszeit auf 500 ms setzen.
                //3. Impulszählwert auf 4 Impulse setzen.
                //4. Interruptprogramm 0 als Interrupt
                //  für die Bearbeitung der Interrupts PTO beendet freigeben.
                //5. Alle Interruptereignisse freigeben.
                //6. PTO-Funktion aufrufen, PLS0 => A0.0.
                //7. Steuerbyte für nachfolgende Änderungen der Zykluszeit
                //  vorbereiten.
LD      SM0.0
MOVB    16#8D, SMB67
MOVW    +500, SMW68
MOVD    +4, SMD72
ATCH    INT_0, 19
ENI
PLS     0
MOVB    16#89, SMB67
```

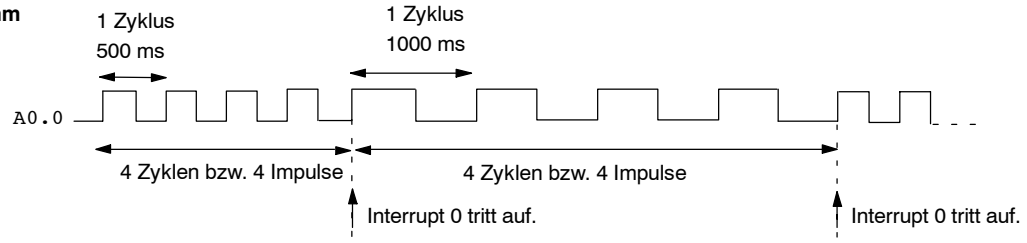
Beispiel: Einzel-Segment-Impulsfolge (PTO), Fortsetzung

I N T 0



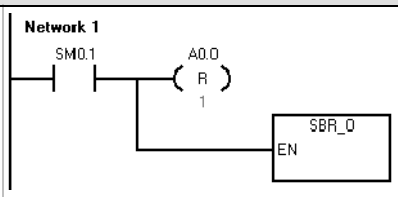

```
Netzwerk 1    //Wenn die aktuelle Zykluszeit 500 ms beträgt:
              //Zykluszeit auf 1000 ms setzen und 4 Impulse erzeugen.
LDW=   SMW68, +500
MOVW   +1000, SMW68
PLS    0
CRETI

Netzwerk 2    //Wenn die aktuelle Zykluszeit 1000 ms beträgt:
              //Zykluszeit auf 500 ms setzen und 4 Impulse erzeugen.
LDW=   SMW68, +1000
MOVW   +500, SMW68
PLS    0
```

Impulsdiagramm



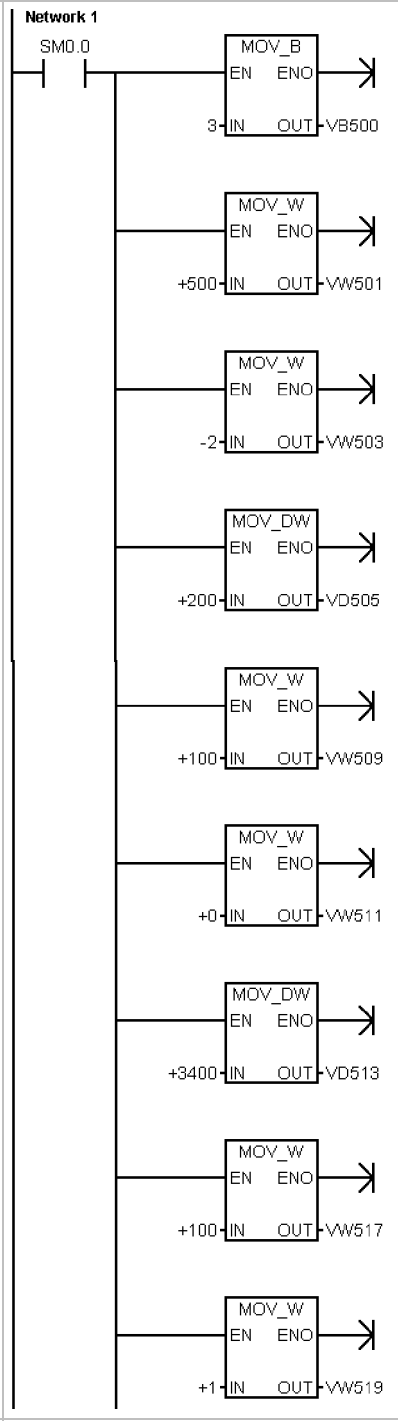
Beispiel: Multi-Segment-Impulsfolge (PTO)

MAIN




```
Netzwerk 1      //Im ersten Zykuls,
                //Bit im Prozessabbild auf Low setzen und Unterprogramm 0
                //aufrufen.
LD      SM0.1
R       A0.0, 1
CALL    SBR0
```

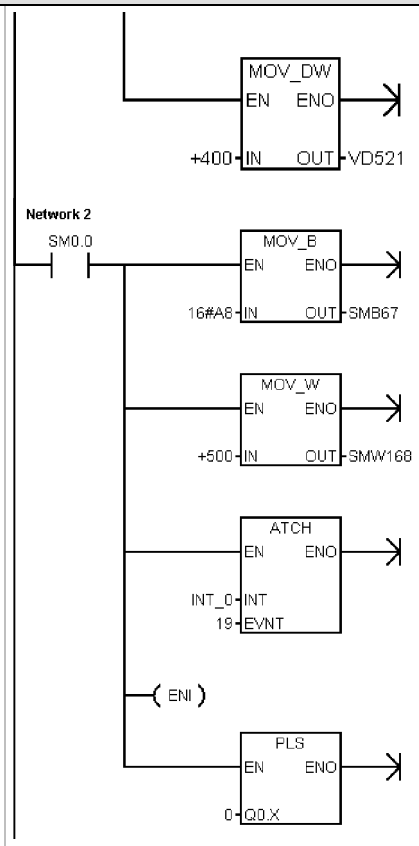
SBR0




```
Netzwerk 1      //PTO-Profiltabelle im voraus laden:
                //Anzahl Segmente in Profiltabelle auf 3 setzen.
                //Alle 3 Segmente konfigurieren.
                //
                //1. Segment 1 einrichten:
                //    - Anfangszykluszeit = 500 ms einstellen.
                //    - Zykluszeitdelta auf -2 ms setzen.
                //    - Anzahl Impulse auf 200 setzen.
                //2. Segment 2 einrichten:
                //    - Anfangszykluszeit = 100 ms einstellen.
                //    - Zykluszeitdelta auf 0 ms setzen.
                //    - Anzahl Impulse auf 3400 setzen.
                //3. Segment 3 einrichten:
                //    - Anfangszykluszeit = 100 ms einstellen.
                //    - Zykluszeitdelta auf 1 ms setzen.
                //    - Anzahl Impulse auf 400 setzen.

LD      SM0.0
MOVB    3, VB500
MOVW    +500, VW501   //Segment 1
MOVW    -2, VW503
MOVD    +200, VD505
MOVW    +100, VW509   //Segment 2
MOVW    +0, VW511
MOVD    +3400, VD513
MOVW    +100, VW517   //Segment 3
MOVW    +1, VW519
MOVD    +400, VD521
```

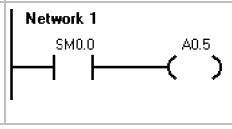
Beispiel: Multi-Segment-Impulsfolge (PTO), Fortsetzung

SBR0 Fortsetzung




```
Netzwerk 2      //1. Steuerbyte einrichten:
                //    - Funktion PTO wählen.
-               //    - Multi-Segment-Betrieb einstellen.
                //    - In ms inkrementieren.
                //    - Funktion PTO freigeben.
                //2. Anfangsadresse für Profiltabelle auf V500 setzen.
                //3. Interruptprogramm 0 als Interrupt
                //    für die Bearbeitung der Interrupts PTO beendet freigeben.
                //4. Alle Interrupts freigeben.
                //5. PTO-Funktion aufrufen, PLS0 => A0.0.
LD      SM0.0
MOVB    16#A8, SMB67
MOVW    +500, SMW168
ATCH    INT_0, 19
ENI
PLS     0
```

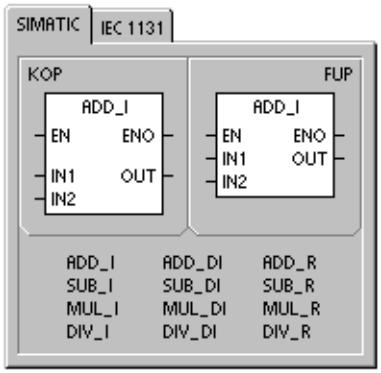
INT0

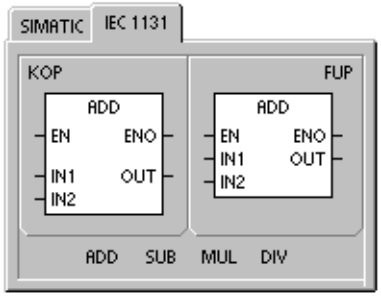


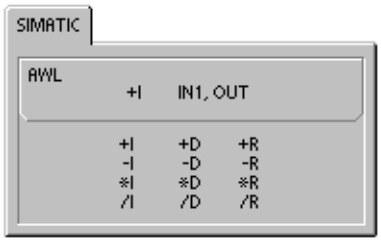

```
Netzwerk 1      //Wenn das PTO-Profil vollständig ist,
                //Ausgang A0.5 einschalten.
LD      SM0.0
=       A0.5
```

Arithmetische Operationen

Operationen Addieren, Subtrahieren, Multiplizieren und Dividieren







Addieren

IN1 + IN2 = OUT
IN1 + OUT = OUT

Subtrahieren

IN1 - IN2 = OUT
OUT - IN1 = OUT

KOP und FUP
AWL

Die Operationen Ganze Zahlen (16 Bit) addieren (+I) und Ganze Zahlen (16 Bit) subtrahieren (-I) addieren bzw. subtrahieren zwei ganze Zahlen (16 Bit) und liefern ein Ergebnis (16 Bit). Die Operationen Ganze Zahlen (32 Bit) addieren (+D) und Ganze Zahlen (32 Bit) subtrahieren (-D) addieren bzw. subtrahieren zwei ganze Zahlen (32 Bit) und liefern ein Ergebnis (32 Bit). Die Operationen Realzahlen addieren (+R) und Realzahlen subtrahieren (-R) addieren bzw. subtrahieren zwei Realzahlen (32 Bit) und liefern eine Realzahl (32 Bit) als Ergebnis.

Multiplizieren

IN1 * IN2 = OUT
IN1 * OUT = OUT

Dividieren

IN1/ IN2 = OUT
OUT / IN1 = OUT

KOP und FUP
AWL

Die Operationen Ganze Zahlen (16 Bit) multiplizieren (*I) und Ganze Zahlen (16 Bit) dividieren (/I) multiplizieren bzw. dividieren zwei ganze Zahlen (16 Bit) und liefern ein Ergebnis (16 Bit). (Bei der Division wird kein Divisionsrest erfasst.) Die Operationen Ganze Zahlen (32 Bit) multiplizieren (*D) und Ganze Zahlen (32 Bit) dividieren (/D) multiplizieren bzw. dividieren zwei ganze Zahlen (32 Bit) und liefern ein Ergebnis (32 Bit). (Bei der Division wird kein Divisionsrest erfasst.) Die Operationen Realzahlen multiplizieren (*R) und Realzahlen dividieren (/R) multiplizieren bzw. dividieren zwei Realzahlen (32 Bit) und liefern eine Realzahl (32 Bit) als Ergebnis.

Sondermerker und ENO

SM1.1 zeigt einen Überlauffehler oder unzulässige Werte an. Wird SM1.1 gesetzt, dann ist der Status von SM1.0 und SM1.2 ungültig und die ursprünglichen Eingangsoperanden werden nicht verändert. Werden SM1.1 und SM1.3 nicht gesetzt, dann hat die arithmetische Operation ein gültiges Ergebnis hervorgebracht und SM1.0 und SM1.2 enthalten den gültigen Status. Wird SM1.3 während einer Division gesetzt, dann werden die anderen Statusbits für arithmetische Operationen nicht verändert.

Fehlerbedingungen, die ENO = 0 setzen

- SM1.1 (Überlauf)
- SM1.3 (Division durch Null)
- 0006 (Indirekte Adresse)

Betroffene Sondermerker

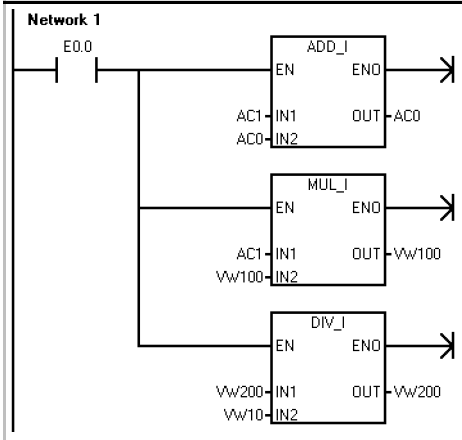
- SM1.0 (Null)
- SM1.1 (Überlauf, unzulässiger Wert während der Bearbeitung erzeugt oder unzulässiger Eingangsparameter erkannt)
- SM1.2 (Negativ)
- SM1.3 (Division durch Null)

Tabelle 6-38 Gültige Operanden für die Operationen Addieren, Subtrahieren, Multiplizieren und Dividieren

Eingänge/ Ausgänge	Datentypen	Operanden
IN1, IN2	INT	EW, AW, VW, MW, SMW, SW, T, Z, LW, AC, AEW, *VD, *AC, *LD, Konstante
	DINT	ED, AD, VD, MD, SMD, SD, LD, AC, HC, *VD, *LD, *AC, Konstante
	REAL	ED, AD, VD, MD, SMD, SD, LD, AC, *VD, *LD, *AC, Konstante
OUT	INT	EW, AW, VW, MW, SMW, SW, LW, T, Z, AC, *VD, *AC, *LD
	DINT, REAL	ED, AD, VD, MD, SMD, SD, LD, AC, *VD, *LD, *AC

Realzahlen bzw. Gleitpunktzahlen werden in dem Format dargestellt, das in der Richtlinie ANSI/IEEE 754-1985 (einfachgenau) beschrieben ist. Ausführliche Informationen zu diesen Zahlen finden Sie in den Richtlinien.

Beispiel: Festpunktarithmetik

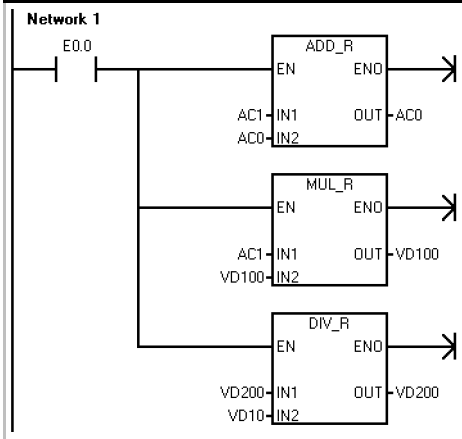


Netzwerk 1

```
LD    E0.0
+I    AC1, AC0
*I    AC1, VW100
/I    VW10, VW200
```

Addieren				Multiplizieren				Dividieren			
40	+ 60	= 100		40	* 20	= 800		4000	/ 40	= 100	
AC1	AC0	AC0		AC1	VW100	VW100		VW200	VW10	VW200	

Beispiel: Gleitpunktarithmetik



Netzwerk 1

```
LD    E0.0
+R    AC1, AC0
*R    AC1, VD100
/R    VD10, VD200
```

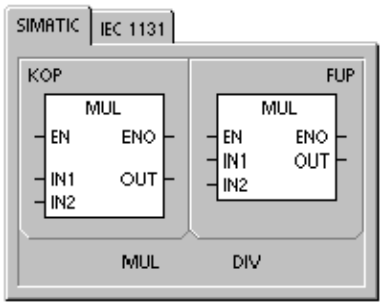
Addieren				Multiplizieren				Dividieren			
4000,0	+ 6000,0	= 10000,0		400,0	* 200,0	= 80000.0		4000,0	/ 41,0	= 97,5609	
AC1	AC0	AC0		AC1	VD100	VD100		VD200	VD10	VD200	

Ganze Zahlen (16 Bit) in ganze Zahl (32 Bit) multiplizieren und Ganze Zahlen (16 Bit) dividieren mit Divisionsrest

Ganze Zahlen (16 Bit) in ganze Zahl (32 Bit) multiplizieren

IN1 * IN2 = OUT KOP und FUP
IN1 * OUT = OUT AWL

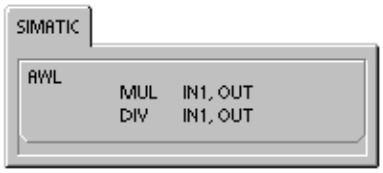
Die Operation Ganze Zahlen (16 Bit) in ganze Zahl (32 Bit) multiplizieren (MUL) multipliziert zwei ganze Zahlen (16 Bit) und liefert ein 32-Bit-Ergebnis. Beim Multiplizieren in AWL wird das niederwertigste Wort (16 Bit) von OUT (32 Bit) als einer der Faktoren verwendet.



Ganze Zahlen (16 Bit) dividieren mit Divisionsrest

IN1/ IN2 = OUT KOP und FUP
OUT / IN1 = OUT AWL

Die Operation Ganze Zahlen (16 Bit) in ganze Zahl (32 Bit) dividieren (DIV) dividiert zwei ganze Zahlen (16 Bit) und liefert ein 32-Bit-Ergebnis, das sich aus einem 16-Bit-Divisionsrest (dem höchstwertigen Wort) und einem 16-Bit-Quotienten (dem niederwertigsten Wort) zusammensetzt.



Beim Dividieren in AWL wird das niederwertigste Wort (16 Bit) von OUT (32 Bit) als Dividend verwendet.

Sondermerker und ENO

Bei beiden Operationen auf dieser Seite zeigen die Sondermerkerbits (SM) Fehler und unzulässige Werte an. Wird SM1.3 (Division durch Null) während einer Division gesetzt, dann werden die anderen Statusbits für arithmetische Operationen nicht verändert. Andernfalls enthalten alle unterstützten arithmetischen Statusbits nach Abschluss der Operation den gültigen Status.

Fehlerbedingungen, die ENO = 0 setzen

- SM1.1 (Überlauf)
- SM1.3 (Division durch Null)
- 0006 (Indirekte Adresse)

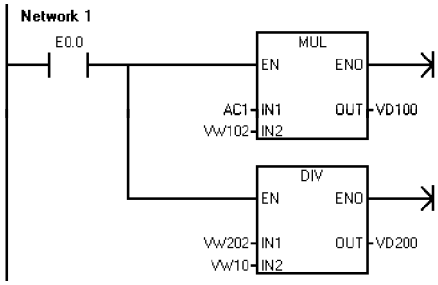
Betroffene Sondermerker

- SM1.0 (Null)
- SM1.1 (Überlauf)
- SM1.2 (Negativ)
- SM1.3 (Division durch Null)

Tabelle 6-39 Gültige Operanden für die Operationen Ganze Zahlen (16 Bit) in ganze Zahl (32 Bit) multiplizieren und Ganze Zahlen (16 Bit) dividieren mit Divisionsrest

Eingänge/ Ausgänge	Datentypen	Operanden
IN1, IN2	INT	EW, AW, VW, MW, SMW, SW, T, Z, LW, AC, AEW, *VD, *LD, *AC, Konstante
OUT	DINT	ED, AD, VD, MD, SMD, SD, LD, AC, *VD, *LD, *AC

Beispiel: Operationen Ganze Zahlen (16 Bit) in ganze Zahl (32 Bit) multiplizieren und Ganze Zahlen (16 Bit) dividieren mit Divisionsrest



Netzwerk 1
LD E0.0
MUL AC1, VD100
DIV VW10, VD200

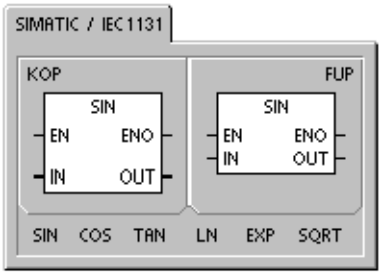
Ganze Zahlen (16 Bit) in ganze Zahl (32 Bit) multiplizieren: 400 (AC1) * 200 (VW102) = 80000 (VD100)

Ganze Zahlen (16 Bit) dividieren mit Divisionsrest: 4000 (VW202) / 41 (VW10) = Divsionsrest 23 (VW200) Quotient 97 (VW202), VD200

Hinweis: VD100 enthält: VW100 und VW102, und VD200 enthält: VW200 und VW202.

Numerische Funktionen

Sinus, Cosinus und Tangens

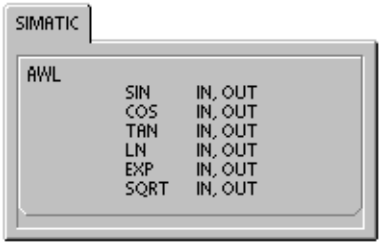


Die Operationen Sinus (SIN), Cosinus (COS) und Tangens (TAN) werten die trigonometrische Funktion des Winkelwerts IN aus und legen das Ergebnis in OUT ab. Der Eingabewinkel ist in Bogenmaß angegeben.

SIN (IN) = OUT COS (IN) = OUT TAN (IN) = OUT

So wandeln Sie Winkel von Grad in Bogenmaß um: Multiplizieren Sie den Winkel in Grad mit der Operation MUL_R (*R) mit 1,745329E-2 (ungefähr mit π/180).

Natürlicher Logarithmus und Natürlicher Exponent



Die Operation Natürlicher Logarithmus (LN) bildet den natürlichen Logarithmus vom Wert in IN und legt das Ergebnis in OUT ab.

Die Operation Natürlicher Exponent (EXP) führt die Exponentialfunktion e potenziert mit dem Wert von IN aus und legt das Ergebnis in OUT ab.

LN (IN) = OUT EXP (IN) = OUT

So berechnen Sie den Logarithmus zur Basis 10 des natürlichen Logarithmus: Dividieren Sie den natürlichen Logarithmus durch 2,302585 (ungefähr der natürliche Logarithmus von 10).

So potenzieren Sie ein Realzahl mit einer anderen Realzahl, einschließlich Bruchexponenten: Verbinden Sie die Operation Natürlicher Exponent mit der Operation Natürlicher Logarithmus. Für X hoch Y beispielsweise geben Sie die folgende Operation ein: EXP (Y * LN (X)).

Quadratwurzel einer Realzahl ziehen

Die Operation Quadratwurzel einer Realzahl ziehen (SQRT) zieht die Quadratwurzel einer Realzahl (IN) und liefert als Ergebnis eine Realzahl (OUT).

SQRT (IN) = OUT

So berechnen Sie eine andere Wurzel:

5 hoch 3 = 5^3 = EXP(3*LN(5)) = 125
Kubikwurzel von 125 = 125^(1/3) = EXP((1/3)*LN(125)) = 5
Quadratwurzel von 5 hoch 3 = 5^(3/2) = EXP(3/2*LN(5)) = 11,18034

Sondermerker und ENO für die numerischen Funktionen

Bei allen auf dieser Seite beschriebenen Operationen dient SM1.1 als Anzeige für Überlauffehler und unzulässige Werte. Wird SM1.1 gesetzt, dann ist der Status von SM1.0 und SM1.2 ungültig und die ursprünglichen Eingangsoperanden werden nicht verändert. Wird SM1.1 nicht gesetzt, dann hat die arithmetische Operation ein gültiges Ergebnis hervorgebracht und SM1.0 und SM1.2 enthalten den gültigen Status.

Fehlerbedingungen, die ENO = 0 setzen

- SM1.1 (Überlauf)
- 0006 (Indirekte Adresse)

Betroffene Sondermerker

- SM1.0 (Null)
- SM1.1 (Überlauf)
- SM1.2 (Negativ)

Tabelle 6-40 Gültige Operanden für numerische Funktionen

Eingänge/Ausgänge	Datentypen	Operanden
IN	REAL	ED, AD, VD, MD, SMD, SD, LD, AC, *VD, *LD, *AC, Konstante
OUT	REAL	ED, AD, VD, MD, SMD, SD, LD, AC, *VD, *LD, *AC

Realzahlen bzw. Gleitpunktzahlen werden in dem Format dargestellt, das in der Richtlinie ANSI/IEEE 754-1985 (einfachgenau) beschrieben ist. Ausführliche Informationen zu diesen Zahlen finden Sie in den Richtlinien.

Operationen Inkrementieren und Dekrementieren

Um 1 erhöhen

IN + 1 = OUT *KOP und FUP*
OUT + 1 = OUT *AWL*

Um 1 vermindern

IN - 1 = OUT *KOP und FUP*
OUT - 1 = OUT *AWL*

Die Operationen Inkrementieren und Dekrementieren addieren bzw. subtrahieren den Wert 1 zu oder vom Eingang IN und legen das Ergebnis in der Variable OUT ab.

Die Operationen Byte um 1 erhöhen (INCB) und Byte um 1 vermindern (DECB) sind vorzeichenlos.

Die Operationen Doppelwort um 1 erhöhen (INCW) und Doppelwort um vermindern (DECW) haben ein Vorzeichen.

Die Operationen Doppelwort um 1 erhöhen (INCD) und Doppelwort um 1 vermindern (DECD) haben ein Vorzeichen.

Fehlerbedingungen, die ENO = 0 setzen:

- SM1.1 (Überlauf)
- 0006 (Indirekte Adresse)

Betroffene Sondermerker:

- SM1.0 (Null)
- SM1.1 (Überlauf)
- SM1.2 (Negativ) bei Wort- und Doppelwortoperationen

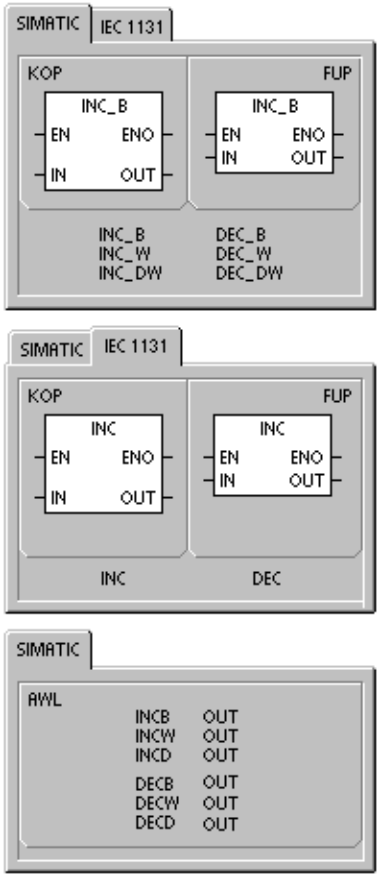
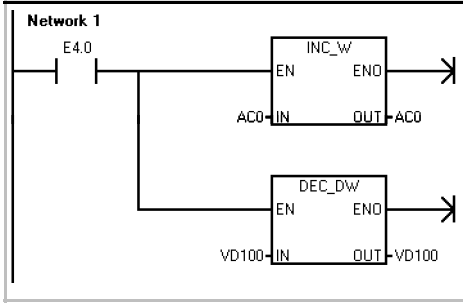


Tabelle 6-41 Gültige Operanden für die Operationen Inkrementieren und Dekrementieren

Eingänge/ Ausgänge	Datentypen	Operanden
IN	BYTE	EB, AB, VB, MB, SMB, SB, LB, AC, *VD, *LD, *AC, Konstante
	INT	EW, AW, VW, MW, SMW, SW, T, Z, LW, AC, AEW, *VD, *LD, *AC, Konstante
	DINT	ED, AD, VD, MD, SMD, SD, LD, AC, HC, *VD, *LD, *AC, Konstante
OUT	BYTE	EB, AB, VB, MB, SMB, SB, LB, AC, *VD, *AC, *LD
	INT	EW, AW, VW, MW, SMW, SW, T, Z, LW, AC,*VD, *LD, *AC
	DINT	ED, AD, VD, MD, SMD, SD, LD, AC, *VD, *LD, *AC

Beispiel: Operationen Inkrementieren und Dekrementieren

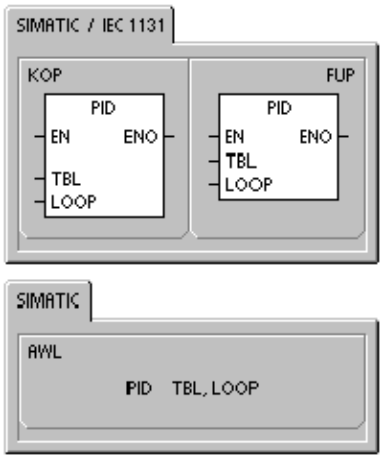


Netzwerk 1

```
LD    E4.0
INCW  AC0
DECD  VD100
```

Wort um 1 erhöhen	125	+ 1 =	126	
	AC0		AC0	
Doppelwort um 1 vermindern	128000	- 1 =	127999	
	VD100		VD100	

Operation PID-Regler (Proportional/Integral/Differential)



Die Operation PID-Regler (PID) berechnet die PID-Regelung für den angegebenen Regelkreis LOOP mit Hilfe der Informationen zu Eingabewerten und Konfiguration in der Definitionstabelle (TBL).

Fehlerbedingungen, die ENO = 0 setzen:

- SM1.1 (Überlauf)
- 0006 (Indirekte Adresse)

Betroffene 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: TBL 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.

Sie dürfen acht Operationen PID in einem Programm 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.

Tabelle 6-42 Gültige Operanden für die Operation PID-Regler

Eingänge/Ausgänge	Datentypen	Operanden
TBL	BYTE	VB
LOOP	BYTE	Konstante (0 bis 7)

Operations-Assistent

STEP 7-Micro/WIN bietet Ihnen einen PID-Assistenten, der Sie durch die Vorgehensweise zum Definieren eines PID-Reglers führt. Wählen Sie den Menübefehl **Extras > Operations-Assistent** und wählen im Assistenten die Option **PID**.

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:

Zuweisen	=	Proportionalanteil	+	Integralanteil	+	Differentialanteil
M(t)	**=**	$K_C * e$	**+**	$K_C \int_0^t e\,dt + M_{initial}$	**+**	$K_C * de/dt$

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 Abtastens und des vorherigen Abtastens, während der Proportionalanteil nur das Ergebnis des aktuellen Abtastens 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 der Regeldifferenz (bei Abtastzeit n - 1)
- K_D proportionale Konstante des Differentialanteils

Die S7-200 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:
- Mn — 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
- MDn — Wert des Differentialanteils der Stellgröße bei Abtastzeit n

Proportionalanteil in der PID-Gleichung

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 der Prozessvariablen (PV) bei einer angegebenen Abtastzeit. Die von der S7-200 verwendete Gleichung für den Proportionalanteil lautet wie folgt:

MP_n	=	K_C	*	$(SW_n - IW_n)$

Erklärung:
- MPn — 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

Integralanteil in der PID-Gleichung

Der Integralanteil MI ist proportional zu der Summe der Regeldifferenz über der Zeit. Die von der S7-200 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 — Verstärung
- 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 finden Sie in 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.

Differentialanteil in der PID-Gleichung

Der Differentialanteil MD ist proportional zu der Änderung der Regeldifferenz. Die von der S7-200 verwendete 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 Schrittänderungen 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	Verstärung
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. Bei der ersten Berechnung wird der Wert von SW_{n-1} mit dem Wert von SW_n initialisiert.

Auswählen des 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 ”INF” 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 Eingangswerte

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 Fahrgeschwindigkeits-
regler (Tempomat) im Pkw. Die Prozessvariable ist ein Wert, der sich 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:


```
ITD     AEW0, AC0       //Eingangswert in Doppelwortwert wandeln
DTR     AC0,  AC0       //Ganze Zahl (32 Bit) in Realzahl wandeln
```

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_{Norm} = $((R_{ur}$ / Spanne) + 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 beim Tempomat 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_{Skal} = (M_n - Versatz) * 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      //Überträgt die Stellgröße in den Akkumulator
-R     0,5,  AC0        //Diese Anweisung nur aufnehmen, wenn der Wert zweipolig ist
*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:

```
ROUND  AC0, AC0         //Realzahl in ganze Zahl (32 Bit) wandeln
DTI    AC0,  LW0        //Wert in ganze Zahl (16 Bit) wandeln
MOVW   LW0, AAW0        //Wert in Analogausgang schreiben
```

Vorwärts- und Rückwärtsverhalten in Regelkreisen

Der Regelkreis zeigt Vorwärts-Verhalten, wenn die Verstärkung positiv ist. Eine negative Verstärkung bewirkt ein Rückwärts-Verhalten. (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 (siehe Abschnitt zu den Betriebsarten weiter unten).

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:

MX = 1.0 - (MP_n + MD_n) *wenn die errechnete Stellgröße M_n > 1,0*

oder

MX = - (MP_n + MD_n) *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 Prozessvariable 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 Handbetrieb werden keine PID-Berechnungen ausgeführt.

Die Operation PID hat ein Verlaufsbit für den Signalzustand, ähnlich wie bei Zähloperationen. Mit diesem Verlaufsbit erkennt die Operation einen Wechsel von 0 nach 1 im Signalfluss. Wird der Wechsel im Signalfluss erkannt, führt die Operation eine Reihe von Aktionen aus, damit ein glatter Übergang vom Handbetrieb zum Automatikbetrieb gewährleistet ist. Damit der Übergang in den Automatikbetrieb nicht sprunghaft verläuft, muss der Wert der Stellgröße für Handbetrieb 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 bei einer steigenden Flanke ein glatter Übergang vom Handbetrieb zum Automatikbetrieb 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 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 glatt verläuft.

Alarme 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 S7-200 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 6-43).

Tabelle 6-43 Tabelle für den Regelkreis

Versatz	Feld	Format	Typ	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.

PID-Programmbeispiel

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 % (leer) und 100 % (voll) liegen. Die Stellgröße ist ein Wert zwischen 0 % und 100 % für die Antriebsdrehzahl der Pumpe.

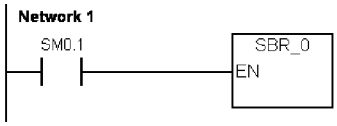
Der Sollwert wird vorab festgelegt und direkt in die Tabelle für den Regelkreis eingetragen. Die Prozessvariable (der Istwert) wird als einpoliger Analogwert vom Schwimmer geliefert. Die Stellgröße wird als einpoliger Analogwert geschrieben und dient zum Regeln der Pumpendrehzahl. Die Spanne für den 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 = 0,25, T_S = 0,1 s und T_I = 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 Handbetrieb auf Automatikbetrieb umgeschaltet. Ein Digitaleingang wird für den Übergang von Handbetrieb zu Automatikbetrieb verwendet. Dieser Eingang (E0.0) dient der Steuerung Hand-/Automatikbetrieb: 0 = Handbetrieb und 1 = Automatikbetrieb. Im Handbetrieb schreibt der Operator die Pumpendrehzahl als Realzahl zwischen 0,0 und 1,0 in VD108.

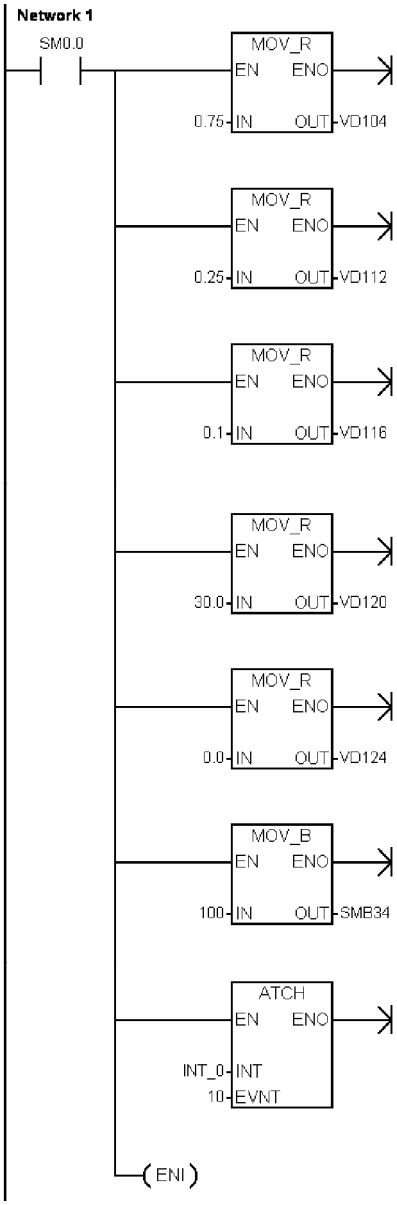
Beispiel: PID-Regler

MAIN




```
Netzwerk 1    //Im ersten Zyklus,
              //Unterprogramm für die Initialisierung aufrufen
LD      SM0.1
CALL    SBR0
```

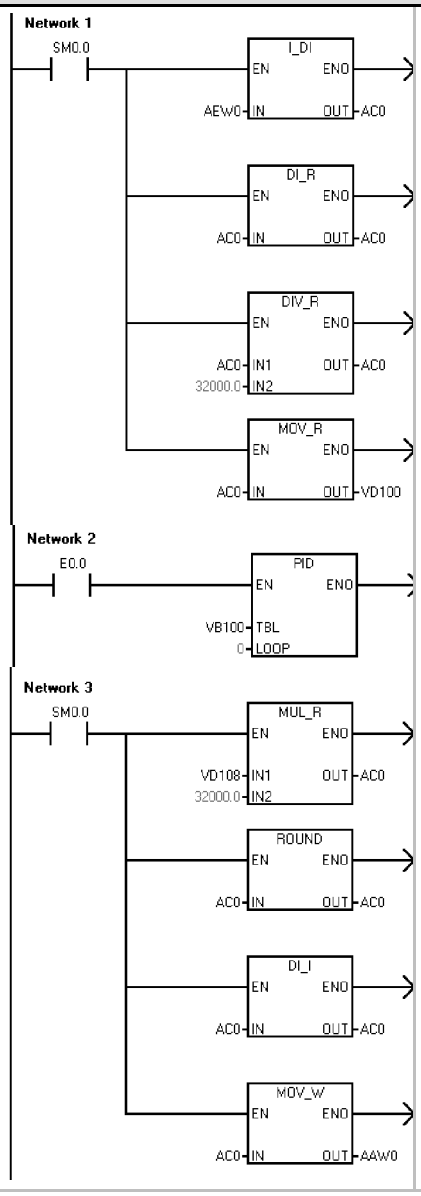
SBR0




```
Netzwerk 1    //PID-Parameter laden und
              //PID-Interruptprogramm zuweisen:
              //1. Sollwert laden = 75 % voll.
              //2. Verstärkung = 0,25 laden.
              //3. Abtastzeit = 0,1 s laden.
              //4. Integralzeit = 30 Min. laden.
              //5. Kein Differentialverhalten setzen.
              //6. Intervall (100 ms) für
              //   zeitgesteuerten Interrupt INT_0 einstellen.
              //7. Mit zeitgesteuertem Interrupt
              //   Operation PID aufrufen.
              //8. Interrupts freigeben
LD      SM0.0
MOVR    0,75, VD104
MOVR    0,25, VD112
MOVR    0,1, VD116
MOVR    30,0, VD120
MOVR    0,0, VD124
MOVB    100, SMB34
ATCH    INT_0, 10
ENI
```

Beispiel: PID-Regler, Fortsetzung

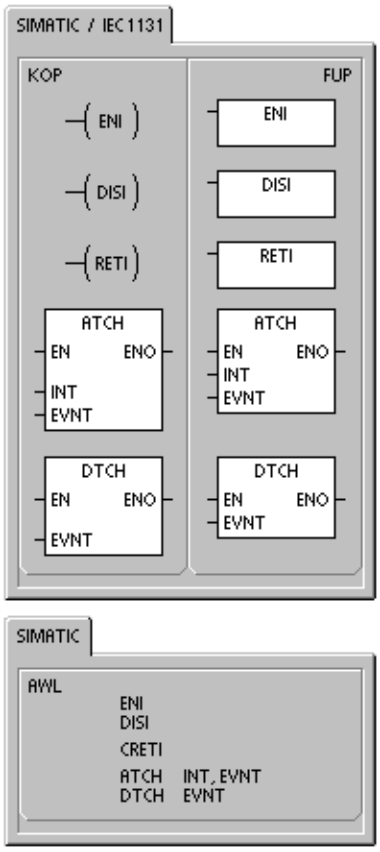
INT 0




```
Netzwerk 1      //Prozessvariable in normalisierte Realzahl skalieren.
                //1. Ganze Zahl (16 Bit) in ganze Zahl (32 Bit) wandeln.

                //2. Ganze Zahl (32 Bit) in Realzahl wandeln.
                //3. Wert normalisieren.
                //4. Normalisierte Prozessvariable in TABLE speichern
LD      SM0.0
ITD     AEW0, AC0
DTR     AC0, AC0
/R      32000.0, AC0
MOVR    AC0, VD100

Netzwerk 2      //Regler bei Einnehmen des Automatikbetriebs
                //ausführen.
LD      E0.0
PID     VB100, 0

Netzwerk 3      //Stellgröße Mn in ganze Zahl skalieren.
                //Mn ist ein einpoliger Wert und kann nicht negativ sein.

                //1. Stellgröße in Akkumulator übertragen.
                //2. Wert im Akkumulator skalieren.
                //3. Realzahl in ganze Zahl (32 Bit) wandeln.
                //4. Ganze Zahl (32 Bit) in ganze Zahl (16 Bit) wandeln.

                //5. Wert in Analogausgang schreiben.
LD      SM0.0
MOVR    VD108, AC0
*R      32000,0, AC0
ROUND   AC0, AC0
DTI     AC0, AC0
MOVW    AC0, AAW0
```

Interruptoperationen

Alle Interruptereignisse freigeben und Alle Interruptereignisse sperren



Die Operation Alle Interruptereignisse freigeben (ENI) gibt die Bearbeitung aller zugeordneten Interruptereignisse frei. Die Operation Alle Interruptereignisse sperren (DISI) sperrt die Bearbeitung aller Interruptereignisse.

Wenn Sie in den Betriebszustand RUN wechseln, sind die Interrupts zunächst gesperrt. Befindet sich die CPU im Betriebszustand RUN, können Sie mit der Operation ENI alle Interruptereignisse freigeben. Wenn Sie die Operation Alle Interruptereignisse sperren ausführen, können keine Interrupts mehr bearbeitet werden. Aktive Interruptereignisse werden jedoch weiterhin in die Warteschlange aufgenommen.

Fehlerbedingungen, die ENO = 0 setzen:

- 0004 (Versuch, eine der Operationen ENI, DISI oder HDEF im Interruptprogramm auszuführen)

Interruptprogramm bedingt beenden

Die Operation Interruptprogramm bedingt beenden (CRETI) beendet einen Interrupt in Abhängigkeit von dem Zustand der vorherigen Verknüpfung.

Interrupt zuordnen

Die Operation Interrupt zuordnen (ATCH) ordnet einem Interruptereignis (EVNT) die Nummer eines Interruptprogramms INT zu und gibt dann das Ereignis frei.

Fehlerbedingungen, die ENO = 0 setzen:

- 0002 (Zuweisung der Eingänge eines HSC in Konflikt)

Interrupt trennen

Die Operation Interrupt trennen (DTCH) trennt ein Interruptereignis (EVNT) von allen Interruptprogrammen und sperrt dann das Ereignis.

Tabelle 6-44 Gültige Operanden für die Operationen Interrupt zuordnen und Interrupt trennen

Eingänge/ Ausgänge	Datentypen	Operanden		
INT	BYTE	Konstante (0 bis 127)		
EVNT	BYTE	Konstante	*CPU 221 und CPU 222:*	0 bis 12, 19 bis 23 und 27 bis 33
			CPU 224:	0 bis 23 und 27 bis 33
			CPU 226 und CPU 226XM:	0 bis 33

Funktionsweise 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 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.

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 oder bis die Warteschlange überläuft.

Einzelne Interruptereignisse können Sie sperren, indem Sie die Zuordnung von Ereignis und Programm mit der Operation Interrupt trennen aufheben. Die Operation Interrupt trennen setzt den Interrupt in einen inaktiven Zustand, in dem er übergangen und deshalb nicht bearbeitet wird. Tabelle 6-45 führt die verschiedenen Interruptereignisse auf.

Tabelle 6-45 Interruptereignisse

EVENT	Beschreibung		CPU 221 CPU 222	CPU 224	CPU 226 CPU 226XM
0	E0.0	Steigende Flanke	ja	ja	ja
1	E0.0	Fallende Flanke	ja	ja	ja
2	E0.1	Steigende Flanke	ja	ja	ja
3	E0.1	Fallende Flanke	ja	ja	ja
4	E0.2	Steigende Flanke	ja	ja	ja
5	E0.2	Fallende Flanke	ja	ja	ja
6	E0.3	Steigende Flanke	ja	ja	ja
7	E0.3	Fallende Flanke	ja	ja	ja
8	Schnittstelle 0	Zeichen empfangen	ja	ja	ja
9	Schnittstelle 0	Übertragung beendet	ja	ja	ja
10	Zeitgesteuerter Interrupt 0	SMB34	ja	ja	ja
11	Zeitgesteuerter Interrupt 1	SMB35	ja	ja	ja
12	HSC0 Wert)	CV=PV (aktueller Wert = voreingestellter	ja	ja	ja
13	HSC1 Wert)	CV=PV (aktueller Wert = voreingestellter		ja	ja
14	HSC1	Richtungswechsel		ja	ja
15	HSC1	Externes Rücksetzen		ja	ja
16	HSC2 Wert)	CV=PV (aktueller Wert = voreingestellter		ja	ja
17	HSC2	Richtungswechsel		ja	ja
18	HSC2	Externes Rücksetzen		ja	ja
19	PLS0	PTO-Impulszählwert erreicht	ja	ja	ja
20	PLS1	PTO-Impulszählwert erreicht	ja	ja	ja
21	Zeit T32	Interrupt CT=PT	ja	ja	ja
22	Zeit T96	Interrupt CT=PT	ja	ja	ja
23	Schnittstelle 0	Meldungsempfang beendet	ja	ja	ja
24	Schnittstelle 1	Meldungsempfang beendet			ja
25	Schnittstelle 1	Zeichen empfangen			ja
26	Schnittstelle 1	Übertragung beendet			ja
27	HSC0	Richtungswechsel	ja	ja	ja
28	HSC0	Externes Rücksetzen	ja	ja	ja
29	HSC4 Wert)	CV=PV (aktueller Wert = voreingestellter	ja	ja	ja
30	HSC4	Richtungswechsel	ja	ja	ja
31	HSC4	Externes Rücksetzen	ja	ja	ja
32	HSC3 Wert)	CV=PV (aktueller Wert = voreingestellter	ja	ja	ja
33	HSC5 Wert)	CV=PV (aktueller Wert = voreingestellter	ja	ja	ja

Verarbeitung von Interruptprogrammen durch die S7-200

Das Interruptprogramm wird in Reaktion auf ein zugeordnetes internes oder externes Ereignis ausgeführt. Nachdem die letzte Operation eines Interruptprogramms ausgeführt wurde, wird die Steuerung wieder an das Hauptprogramm übergeben.Sie können das Interruptprogramm beenden, indem Sie die Operation Interruptprogramm bedingt beenden (CRETI) ausführen. Tabelle 6-46 führt einige Richtlinien und Einschränkungen für die Verwendung von Interruptprogrammen in Ihrem Programm auf.

Tabelle 6-46 Richtlinien und Einschränkungen für die Verwendung von Interruptprogrammen

Richtlinien
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
Sie dürfen die Operationen Alle Interruptereignisse sperren (DISI), Alle Interruptereignisse freigeben (ENI), Modus für schnellen Zähler definieren (HDEF) und Bearbeitung beenden (END) nicht in einem Interruptprogramm verwenden.

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.

Sie können Daten in einem Hauptprogramm und in Interruptprogrammen gemeinsam nutzen.

Sie können Daten in einem Hauptprogramm und in einem oder in mehreren Interruptprogrammen gemeinsam nutzen. Weil es nicht vorhersehbar ist, wann die S7-200 einen Interrupt erzeugt, ist es empfehlenswert, die Anzahl von Variablen zu begrenzen, die sowohl im Interruptprogramm als auch an anderen Stellen im Programm verwendet werden. 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. Wenn Sie die lokale Variablentabelle des Interruptprogramms verwenden, stellen Sie sicher, dass Ihr Interruptprogramm nur den temporären Speicher verwendet und keine Daten an anderen Stellen im Programm überschreibt.

Es gibt eine Reihe von Programmiertechniken, 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 (MOVB, MOVW, MOVD, MOVR) 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 Programmiertechnik kann allerdings verzögerte Reaktionen auf Interruptereignisse verursachen.

Aufrufen von Unterprogrammen in Interruptprogrammen

Sie dürfen Unterprogramme und Interruptprogramme in einer Ebene verschachteln, d.h. Sie dürfen ein Unterprogramm aus einem Interruptprogramm aufrufen. Wird ein Unterprogramm aus einem Interruptprogramm aufgerufen, nutzen beide Programme die Akkumulatoren und den logischen Stack gemeinsam.

Von der S7-200 unterstützte Arten von Interrupts

Die S7-200 unterstützt die folgenden Arten von Interruptprogrammen:

- Kommunikationsschnittstellen-Interrupts: Die S7-200 erzeugt Ereignisse, mit denen Ihr Programm die Kommunikationsschnittstelle steuern kann.
- E/A-Interrupts: Die S7-200 erzeugt Ereignisse für verschiedene Zustandsänderungen unterschiedlicher E/A. Durch diese Ereignisse kann Ihr Programm auf die schnellen Zähler, Impulsausgaben und auf steigende oder fallende Flanken an den Eingängen reagieren.
- Zeitgesteuerte Interrupts: Die S7-200 erzeugt Ereignisse, mit denen Ihr Programm in bestimmten Zeitabständen reagieren kann.

Kommunikationsschnittstellen-Interrupts:

Die serialle Kommunikationsschnittstelle der S7-200 kann von Ihrem Programm 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 finden Sie in 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 S7-200 kann bei einer steigenden und/oder fallenden Flanke an einem Eingang (entweder E0.0, E0.1, E0.2 oder E0.3) einen Interrupt erzeugen. 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.

Mit den Interrupts der schnellen Zähler können Sie auf folgende Ereignisse reagieren: aktueller Wert gleich voreingestellter Wert, Zählrichtung wechselt und löst dadurch beispielsweise einen Drehrichtungswechsel eines Motors aus, 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. Sie können Aktionen, die zyklisch ausgeführt werden sollen, mit einem zeitgesteuerten Interrupt definieren. Die Zykluszeit wird in Inkrementen von 1 ms zwischen 1 ms und 255 ms gesetzt. Die Zykluszeit für den zeitgesteuerten Interrupt 0 schreiben Sie in SMB34, die Zykluszeit für den zeitgesteuerten Interrupt 1 schreiben Sie in SMB35.

Das zeitgesteuerte Interruptereignis ruft jedesmal, wenn die Zeit abläuft, das entsprechende Interruptprogramm auf. Sie setzen zeitgesteuerte Interrupts dazu ein, in regelmäßigen Abständen Analogeingänge abzufragen oder einen PID-Regler auszuführen.

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 an SMB34 und SMB35 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 Intervall 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).

Die zeitgesteuerten Interrupts T32/T96 dienen zum zeitgesteuerten Reagieren nach Ablauf eines angegebenen Intervalls. Diese Interrupts werden nur von der Einschaltverzögerungen (TON) und der Ausschaltverzögerung (TOF) mit einer Auflösung von 1 ms, T32 und T96, unterstützt. Ansonsten verfügen die Zeiten T32 und T96 über die übliche Funktionalität. Wird der Interrupt aktiviert, wird das zugeordnete Interruptprogramm ausgeführt, sobald der aktuelle Wert der aktiven Zeit dem voreingestellten Zeitwert während der normalen Aktualisierungen der 1-ms-Zeit, die von der S7-200 ausgeführt werden, entspricht. Sie geben diese Interrupts frei, indem Sie dem Interruptereignis T32/T96 ein Interruptprogramm zuordnen.

Prioritäten der Interrupts und Warteschlangen

Interrupts werden von der S7-200 in der Reihenfolge ihres Auftretens und unter Beachtung der jeweiligen Prioritätsgruppe abgearbeitet. Es ist immer nur ein Programm zur Interruptbearbeitung aktiv. Wird gerade ein Interruptprogramm bearbeitet, dann wird dieses Progamm 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.

Die drei Warteschlangen für Interrupts und die maximale Anzahl Interrupts, die in die Warteschlangen aufgenommen werden kann, zeigt Tabelle 6-47.

Tabelle 6-47 Maximale Anzahl Einträge je Interrupt-Warteschlange

Warteschlange	CPU 221, CPU 222, CPU 224	CPU 226 und CPU 226XM
Kommunikationsinterrupts	4	8
E/A-Interrupts	16	16
Zeitgesteuerte Interrupts	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 6-48 zeigt die Überlaufbits der Warteschlangen für Interrupts. 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 6-49 zeigt alle Interruptereignisse mit Priorität und zugeordneter Ereignisnummer.

Tabelle 6-48 Überlauf der Warteschlange für Interrupts

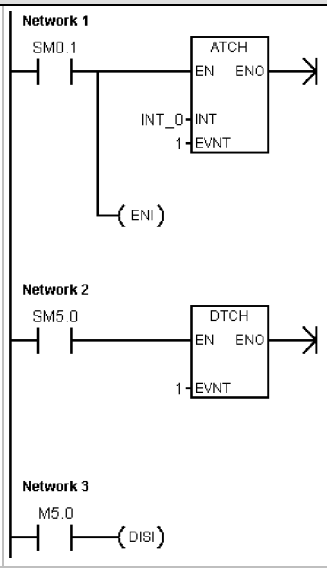
Beschreibung (0 = kein Überlauf, 1 = Überlauf)	Sondermerker
Kommunikationsinterrupts	SM4.0
E/A-Interrupts	SM4.1
Zeitgesteuerte Interrupts	SM4.2

Tabelle 6-49 Reihenfolge der Prioritäten für Interruptereignisse

EVENT	Beschreibung		Prioritätsklasse	Priorität in Gruppen
8	Schnittstelle 0	Zeichen empfangen	Kommunikation *höchste Priorität*	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
19	PLS0	PTO-Impulszählwert erreicht	Digitale Interrupts *mittlere Priorität*	0
20	PLS1	PTO-Impulszählwert erreicht		1
0	E0.0	Steigende Flanke		2
2	E0.1	Steigende Flanke		3
4	E0.2	Steigende Flanke		4
6	E0.3	Steigende Flanke		5
1	E0.0	Fallende Flanke		6
3	E0.1	Fallende Flanke		7
5	E0.2	Fallende Flanke		8
7	E0.3	Fallende Flanke		9
12	HSC0 Wert)	CV=PV (aktueller Wert = voreingestellter		10
27	HSC0	Richtungswechsel		11
28	HSC0	Externes Rücksetzen		12
13	HSC1 Wert)	CV=PV (aktueller Wert = voreingestellter		13
14	HSC1	Richtungswechsel		14
15	HSC1	Externes Rücksetzen		15
16	HSC2 Wert)	CV=PV (aktueller Wert = voreingestellter		16
17	HSC2	Richtungswechsel		17
18	HSC2	Externes Rücksetzen		18
32	HSC3 Wert)	CV=PV (aktueller Wert = voreingestellter		19
29	HSC4 Wert)	CV=PV (aktueller Wert = voreingestellter		20
30	HSC4	Richtungswechsel		21
31	HSC4	Externes Rücksetzen		22
33	HSC5 Wert)	CV=PV (aktueller Wert = voreingestellter		23
10	Zeitgesteuerter Interrupt 0	SMB34	Zeitgesteuerte Interrupts *niedrigste Priorität*	0
11	Zeitgesteuerter Interrupt 1	SMB35		1
21	Zeit T32	Interrupt CT=PT		2
22	Zeit T96	Interrupt CT=PT		3

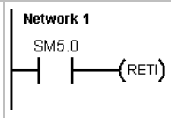
Beispiel: Interruptoperationen

MAIN




```
Netzwerk 1    //Im ersten Zyklus:
              //1. Interruptprogramm INT_0 als Interrupt für eine fallende
              //Flanke
              //  an E0.0 definieren.
              //2. Alle Interruptereignisse freigeben.
LD      SM0.1
ATCH    INT_0, 1
ENI
Netzwerk 2    //Wird ein E/A-Fehler erkannt,
              //Interrupt Erkennung fallende Flanke für E0.0 deaktivieren.
              //Dieses Netzwerk ist optional.
LD      SM5.0
DTCH    1
Netzwerk 3    //Wenn M5.0 eingeschaltet ist,
              //alle Interrupts sperren.
LD      M5.0
DISI
```

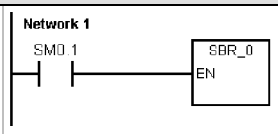
INT 0




```
Netzwerk 1    //Interruptprogramm für fallende Flanke E0.0:
              //Aufgrund eines E/A-Fehlers Interruptprogramm bedingt been
              //den.
LD      SM5.0
CRETI
```

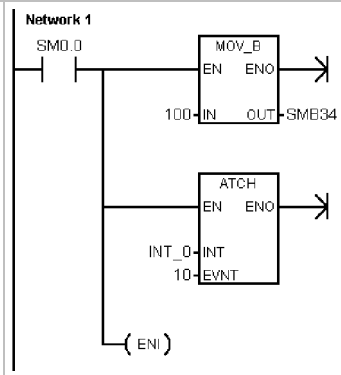
Beispiel: Zeitgesteuerter Interrupt zum Lesen des Werts eines Analogeingangs

MAIN



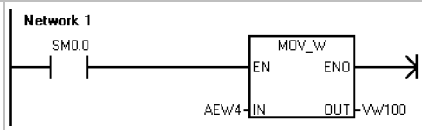

```
Netzwerk 1    //Im ersten Zyklus, Unterprogramm 0 aufrufen.
LD      SM0.1
CALL    SBR0
```

SBR 0




```
Netzwerk 1    //1. Intervall für zeitgesteuerten Interrupt 0 auf 100 ms setzen.
              //2. Zeitgesteuerten Interrupt 0 (Ereignis 10) INT_0 zuordnen.
              //3. Alle Interruptereignisse freigeben.
LD      SM0.0
MOVB    100, SMB34
ATCH    INT_0, 10
ENI
```

INT 0



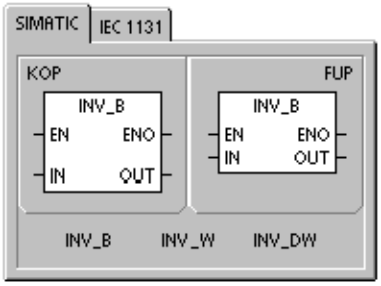

```
Netzwerk 1    //Wert von AEW4 alle 100 ms lesen.
LD      SM0.0
MOVW    AEW4, VW100
```

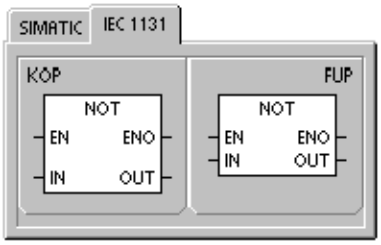
Verknüpfungsoperationen

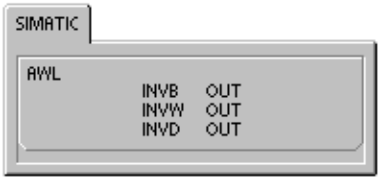
Invertieroperationen

Einerkomplement von Byte, Wort oder Doppelwort erzeugen

Die Operationen Einerkomplement von Byte erzeugen (INVB), Einerkomplement von ganzer Zahl (16 Bit) erzeugen (INVW) und Einerkomplement von ganzer Zahl (32 Bit) erzeugen (INVD) bilden das Einerkomplement des Eingangs IN und laden das Ergebnis in die Adresse im Speicher OUT.







Fehlerbedingungen, die ENO = 0 setzen

- 0006 (Indirekte Adresse)

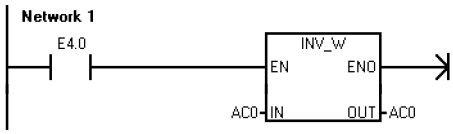
Betroffene Sondermerker:

- SM1.0 (Null)

Tabelle 6-50 Gültige Operanden für die Invertieroperationen

Eingänge/ Ausgänge	Datentypen	Operanden
IN	BYTE	EB, AB, VB, MB, SMB, SB, LB, AC, *VD, *LD, *AC, Konstante
	WORD	EW, AW, VW, MW, SMW, SW, T, Z, LW, AC, AEW, *VD, *LD, *AC, Konstante
	DWORD	ED, AD, VD, MD, SMD, SD, LD, AC, HC, *VD, *LD, *AC, Konstante
OUT	BYTE	EB, AB, VB, MB, SMB, SB, LB, AC,*VD, *LD, *AC
	WORD	EW, AW, VW, MW, SMW, SW, T, Z, LW, AC, *VD, *LD, *AC
	DWORD	ED, AD, VD, MD, SMD, SD, LD, AC, *VD, *LD, *AC

Beispiel: Invertieroperation



Netzwerk 1

```
LD    E4.0
INVW  AC0
```

Wort invertieren

AC0 1101 0111 1001 0101

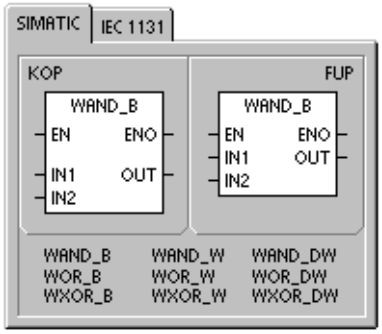
Einerkomplement erzeugen

AC0 0010 1000 0110 1010

Operationen Bitwert durch UND, ODER oder EXKLUSIV ODER verknüpfen

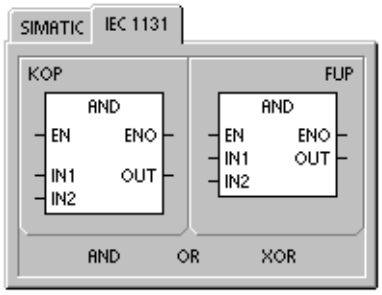
Bytes durch UND verknüpfen, Wörter durch UND verknüpfen und Doppelwörter durch UND verknüpfen

Die Operationen Bytes durch UND verknüpfen (UNDB), Wörter durch UND verknüpfen (UNDW) und Doppelwörter durch UND verknüpfen (UNDD) verknüpfen die entsprechenden Bits von zwei Eingangswerten IN1 und IN2 durch UND und laden das Ergebnis in eine Adresse im Speicher OUT.



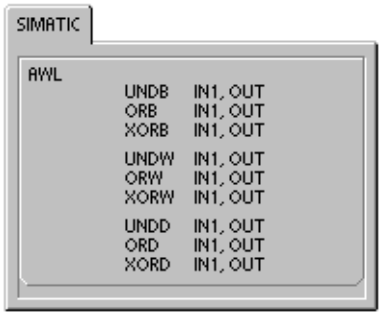
Bytes durch ODER verknüpfen, Wörter durch ODER verknüpfen und Doppelwörter durch ODER verknüpfen

Die Operationen Bytes durch ODER verknüpfen (ORB), Wörter durch ODER verknüpfen (ORW) und Doppelwörter durch ODER verknüpfen (ORD) verknüpfen die entsprechenden Bits von zwei Eingangswerten IN1 und IN2 durch ODER und laden das Ergebnis in eine Adresse im Speicher OUT.



Bytes durch EXKLUSIV ODER verknüpfen, Wörter durch EXKLUSIV ODER verknüpfen und Doppelwörter durch EXKLUSIV ODER verknüpfen

Die Operationen Bytes durch EXKLUSIV ODER verknüpfen (XROB), Wörter durch EXKLUSIV ODER verknüpfen (XORW) und Doppelwörter durch EXKLUSIV ODER verknüpfen (XORD) vverknüpfen die entsprechenden Bits von zwei Eingangswerten IN1 und IN2 durch EXKLUSIV ODER und laden das Ergebnis in eine Adresse im Speicher OUT.



Sondermerker und ENO

Bei allen auf dieser Seite beschriebenen Operationen wirken sich die folgenden Bedingungen auf die Sondermerker und ENO aus.

Fehlerbedingungen, die ENO = 0 setzen

- 0006 (Indirekte Adresse)

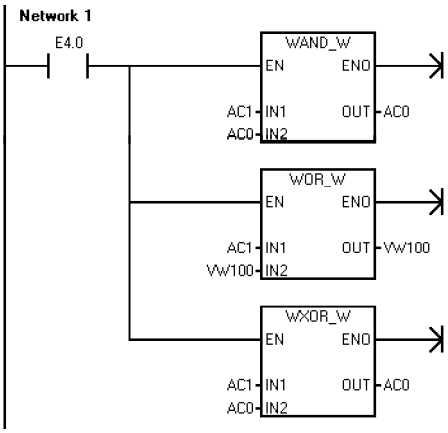
Betroffene Sondermerker:

- SM1.0 (Null)

Tabelle 6-51 Gültige Operanden für die Operationen UND, ODER und EXKLUSIV ODER

Eingänge/ Ausgänge	Datentypen	Operanden
IN1, IN2	BYTE	EB, AB, VB, MB, SMB, SB, LB, AC, *VD, *LD, *AC, Konstante
	WORD	EW, AW, VW, MW, SMW, SW, T, Z, LW, AC, AEW, *VD, *LD, *AC, Konstante
	DWORD	ED, AD, VD, MD, SMD, SD, LD, AC, HC, *VD, *LD, *AC, Konstante
OUT	BYTE	EB, AB, VB, MB, SMB, SB, LB, AC, *VD, *AC, *LD
	WORD	EW, AW, VW, MW, SMW, SW, T, Z, LW, AC, *VD, *AC, *LD
	DWORD	ED, AD, VD, MD, SMD, SD, LD, AC, *VD, *AC, *LD

Beispiel: Operationen Bitwert durch UND, ODER oder EXKLUSIV ODER verknüpfen



Netzwerk 1

```
LD      E4.0
UNDW    AC1, AC0
ORW     AC1, VW100
XORW    AC1, AC0
```

UND

AC1 0001 1111 0110 1101
UND
AC0 1101 0011 1110 0110
gleich
AC0 0001 0011 0110 0100

ODER

AC1 0001 1111 0110 1101
ODER
VW100 1101 0011 1010 0000
gleich
VW100 1101 1111 1110 1101

EXKLUSIV ODER

AC1 0001 1111 0110 1101
EXKLUSIV ODER
AC0 0001 0011 0110 0100
gleich
AC0 0000 1100 0000 1001

Übertragungsoperationen

Byte, Wort, Doppelwort oder Realzahl übertragen

Die Operationen Byte übertragen (MOVB), Wort übertragen (MOVW), Doppelwort übertragen (MOVD) und Realzahl übertragen (MOVR) übertragen einen Wert aus einer Adresse im Speicher IN in eine neue Adresse im Speicher OUT, ohne den ursprünglichen Wert zu verändern.

Mit der Operation Doppelwort übertragen erstellen Sie einen Pointer. Ausführliche Informationen finden Sie im Abschnitt zu Pointern und indirekter Adressierung in Kapitel 4.

Bei der IEC-Operation Werte übertragen können für den Eingang und den Ausgang verschiedene Datentypen verwendet werden, sie müssen jedoch die gleiche Größe haben.

Fehlerbedingungen, die ENO = 0 setzen

- 0006 (Indirekte Adresse)

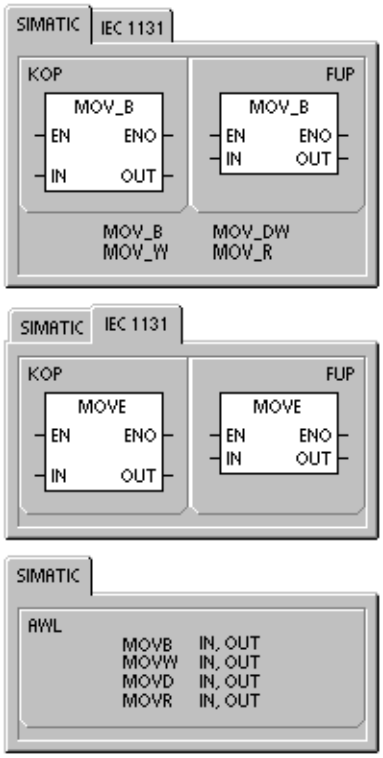
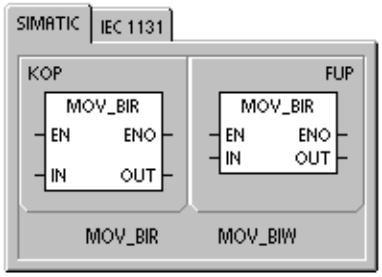
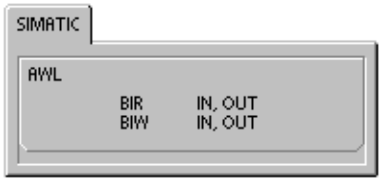


Tabelle 6-52 Gültige Operanden für die Übertragungsoperationen

Eingänge/ Ausgänge	Datentypen	Operanden
IN	BYTE	EB, AB, VB, MB, SMB, SB, LB, AC, *VD, *LD, *AC, Konstante
	WORD, INT	EW, AW, VW, MW, SMW, SW, T, Z, LW, AC, AEW, *VD, *AC, *LD, Konstante
	DWORD, DINT	ED, AD, VD, MD, SMD, SD, LD, AC, HC, &EB, &AB, &VB, &MB, &SB, &T, &Z, *VD, *LD, *AC, Konstante
	REAL	ED, AD, VD, MD, SMD, SD, LD, AC, *VD, *LD, *AC, Konstante
OUT	BYTE	EB, AB, VB, MB, SMB, SB, LB, AC, *VD, *LD, *AC
	WORD, INT	EW, AW, VW, MW, SMW, SW, T, Z, LW, AC, AAW, *VD, *LD, *AC
	DWORD, DINT, REAL	ED, AD, VD, MD, SMD, SD, LD, AC, *VD, *LD, *AC

Byte direkt lesen bzw. schreiben und übertragen





Mit der Operation Byte übertragen können Sie ein Byte direkt vom physikalischen Eingang bzw. Ausgang an eine Adresse im Speicher übertragen.

Die Operation Byte direkt lesen und übertragen (BIR) liest den physikalischen Eingang (IN) und schreibt das Ergebnis in die Adresse im Speicher (OUT), das Prozessabbild wird dabei jedoch nicht aktualisiert.

Die Operation Byte direkt schreiben und übertragen (BIW) liest die Daten der Adresse im Speicher (IN) und schreibt sie in den physikalischen Ausgang (OUT) und in das entsprechende Prozessabbild.

Fehlerbedingungen, die ENO = 0 setzen

- 0006 (Indirekte Adresse)
- Kein Zugriff auf Erweiterungsmodul

Tabelle 6-53 Gültige Operanden für die Operation Byte direkt lesen und übertragen

Eingänge/ Ausgänge	Datentypen	Operanden
IN	BYTE	EB, *VD, *LD, *AC
OUT	BYTE	EB, AB, VB, MB, SMB, SB, LB, AC, *VD, *LD, *AC

Tabelle 6-54 Gültige Operanden für die Operation Byte direkt schreiben und übertragen

Eingänge/ Ausgänge	Datentypen	Operanden
IN	BYTE	EB, AB, VB, MB, SMB, SB, LB, AC, *VD, *LD, *AC, Konstante
OUT	BYTE	AB, *VD, *LD, *AC

Operationen Wertebereich übertragen

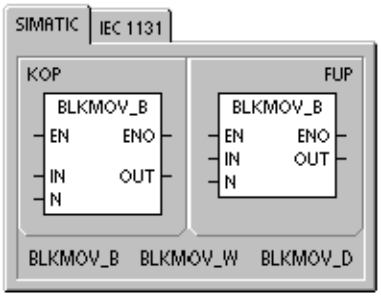
Anzahl an Bytes, Wörtern oder Doppelwörtern übertragen

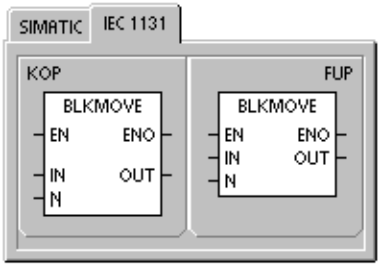
Die Operationen Anzahl an Bytes übertragen (BMB), Anzahl an Wörtern übertragen (BMW) und Anzahl an Doppelwörtern übertragen (BMD) übertragen eine bestimmte Datenmenge an eine neue Adresse im Speicher, indem die Anzahl Bytes, Wörter oder Doppelwörter N, die an der Eingangs-
adresse IN beginnen, in einen neuen Bereich übertragen werden, der an Ausgangsadresse OUT beginnt.

N kann zwischen 1 und 255 liegen.

Fehlerbedingungen, die ENO = 0 setzen

- 0006 (Indirekte Adresse)
- 0091 (Operand außerhalb des Bereichs)





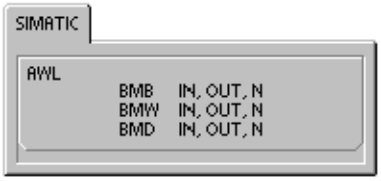
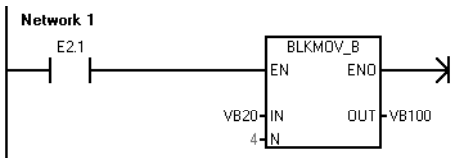


Tabelle 6-55 Gültige Operanden für die Operation Wertebereich übertragen

Eingänge/ Ausgänge	Datentypen	Operanden
IN	BYTE	EB, AB, VB, MB, SMB, SB, LB, *VD, *LD, *AC
	WORD, INT	EW, AW, VW, MW, SMW, SW, T, Z, LW, AEW, *VD, *LD, *AC
	DWORD, DINT	ED, AD, VD, MD, SMD, SD, LD, *VD, *LD, *AC
OUT	BYTE	EB, AB, VB, MB, SMB, SB, LB, *VD, *LD, *AC
	WORD, INT	EW, AW, VW, MW, SMW, SW, T, Z, LW, AAW, *VD, *LD, *AC
	DWORD, DINT	ED, AD, VD, MD, SMD, SD, LD, *VD, *LD, *AC
N	BYTE	EB, AB, VB, MB, SMB, SB, LB, AC, Konstante, *VD, *LD, *AC

Beispiel: Operation Wertebereich übertragen



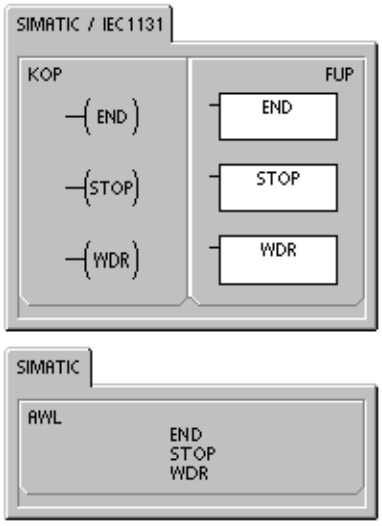

```
Netzwerk 1      //Feld 1 (VB20 bis VB23)
                //in Feld 2 (VB100 bis VB103) übertragen
LD      E2.1
BMB     VB20, VB100, 4
```

	VB20	VB21	VB22	VB23
Feld 1	30	31	32	33
	VB100	VB101	VB102	VB103
Feld 2	30	31	32	33

Operationen für die Programmsteuerung

Bearbeitung bedingt beenden

Die Operation Bearbeitung bedingt beenden (END) beendet den aktuellen Zyklus in Abhängigkeit von dem Zustand der vorherigen Verknüpfung. Sie können die Operation Bearbeitung bedingt beenden im Hauptprogramm verwenden, Sie dürfen sie jedoch nicht in Unterprogrammen und Interruptprogrammen einsetzen.



STOP

Die Operation In STOP gehen (STOP) beendet die Bearbeitung des Anwenderprogramms sofort, indem die S7-200 CPU vom Betriebszustand RUN in den Betriebszustand STOP versetzt wird.

Wird die Operation STOP in einem Interruptprogramm ausgeführt, dann wird dieses sofort beendet und alle anstehenden Interrupts werden ignoriert. Es werden die übrigen Aufgaben des aktuellen Zyklus ausgeführt, einschließlich der Bearbeitung des Hauptprogramms, und der Übergang von RUN in STOP geschieht am Ende des aktuellen Zyklus.

Überwachungszeit rücksetzen

Die Operation Überwachungszeit rücksetzen (WDR) startet die Zeitüberwachung des Systems der S7-200 CPU neu, um die Zeit zu verlängern, die der Zyklus dauern darf, ohne dass ein Zeitüberwachungsfehler auftritt.

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 geforcten 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
- Erweiterungsmodule mit Digitalausgängen enthalten auch eine Zeitüberwachung, die die Ausgänge ausschaltet, wenn die S7-200 nicht in das Modul schreibt. Mit der Operation Direkt schreiben für jedes Erweiterungsmodul mit Digitaleingängen erreichen Sie, dass die Ausgänge während verlängerter Zyklen eingeschaltet bleiben. Siehe auch das auf diese Beschreibung folgende Beispiel.

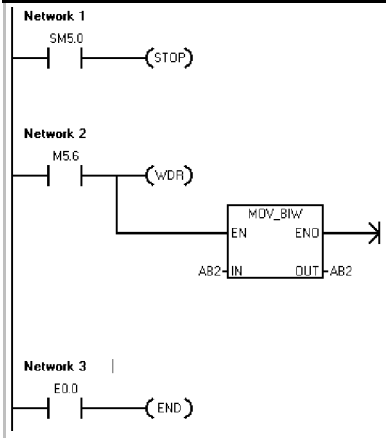
Tipp

Wenn Sie davon ausgehen, dass die Zykluszeit wahrscheinlich 500 ms überschreiten oder die Interrupttaktivität stark ansteigen wird, so dass der Hauptzyklus länger als 500 ms unterbrochen wird, sollten Sie die Überwachungszeit mit der Operation Überwachungszeit rücksetzen neu starten.

Jedesmal, wenn Sie die Operation Überwachungszeit rücksetzen verwenden, müssen Sie auch die Operation Direkt schreiben für ein Ausgangsbyte (AB) pro digitalem Ausgabemodul verwenden, um die Zeitüberwachung des Erweiterungsmoduls zurückzusetzen.

Wenn Sie die Operation Überwachungszeit rücksetzen verwenden, damit ein Programm mit langer Zykluszeit ausgeführt werden kann, und sie den Betriebsartenschalter in die Stellung STOP bringen, geht die S7-200 innerhalb von 1,4 Sekunden in den Betriebszustand STOP.

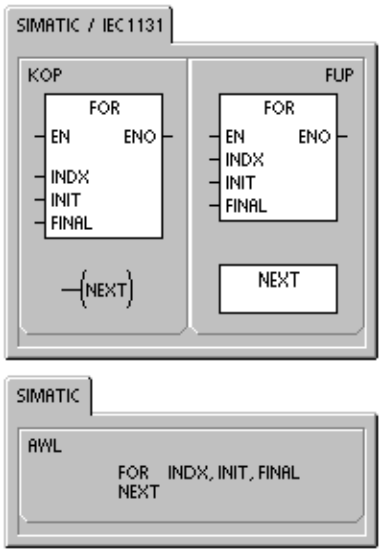
Beispiel: Operationen In STOP gehen, Bearbeitung beenden und Überwachungszeit rücksetzen




```
Netzwerk 1      //Wird ein E/A-Fehler erkannt:
                //Übergang in STOP erzwingen.
LD      SM5.0
STOP

Netzwerk 2      //Wenn M5.6 eingeschaltet ist, Verlängerung des Zyklus ermögli
                //chen:
                //1. Überwachungszeit für die S7-200 rücksetzen.
                //2. Zeitüberwachung des ersten Ausgabemoduls rücksetzen.
LD      M5.6
WDR
BIW     AB2, AB2

Netzwerk 3      //Ist E0.0 eingeschaltet, aktuellen Zyklus beenden.
LD      E0.0
END
```

6

Operationen Programmschleife mit FOR und Ende Programmschleife mit NEXT



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).

Die Operation Programmschleife mit FOR führt den Code zwischen FOR und NEXT aus. Geben Sie den Indexwert bzw. aktuellen Zählwert der Programmschleife INDX, den Anfangswert INIT und den Endwert FINAL an.

Die Operation Ende Programmschleife mit NEXT kennzeichnet das Ende der Programmschleife mit FOR.

Fehlerbedingungen, die ENO = 0 setzen

- 0006 (Indirekte Adresse)

Wenn Sie eine Programmschleife mit FOR/NEXT aktivieren, wird die Schleife solange bearbeitet, bis die Wiederholungen abgearbeitet sind, es sei denn, Sie ändern den Endwert innerhalb der Schleife selbst. Sie können die Werte ändern, während die Programmschleife FOR/NEXT in Bearbeitung ist. Wird die Schleife erneut aktiviert, wird der Anfangswert in den Indexwert (den aktuellen Schleifenzählwert) kopiert.

Die Programmschleife FOR/NEXT setzt sich beim nächsten Aktivieren der Operation selbst zurück.

Beträgt beispielsweise der Wert von INIT 1 und der Wert von FINAL 10, dann werden die Operationen, die sich zwischen den Operationen FOR und NEXT befinden, zehnmal ausgeführt, wobei sich der Zählwert INDX jeweils um 1 erhöht:
1, 2, 3, ...10.

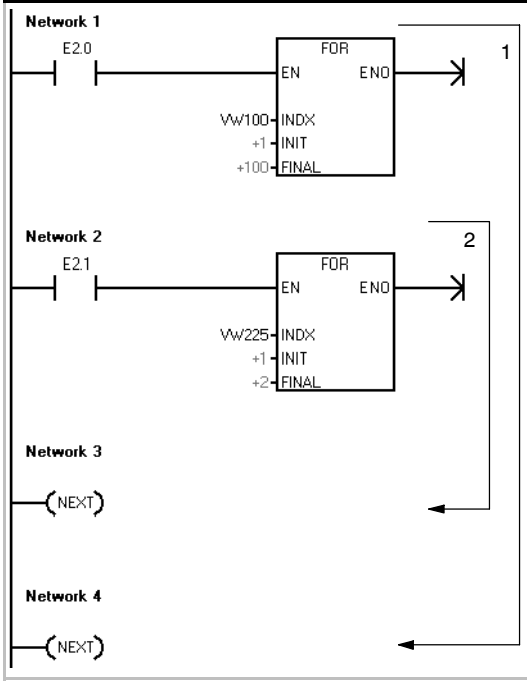
Ist der Anfangswert größer als der Endwert, wird die Schleife nicht ausgeführt. Nach jeder Bearbeitung der Operationen, die zwischen den Operationen FOR und NEXT liegen, wird der Wert von INDX um 1 erhöht und das Ergebnis mit dem Endwert verglichen. Ist INDX größer als der Endwert, wird die Schleife beendet.

Ist die Spitze des Stack 1, wenn Ihr Programm in die FOR/NEXT-Schleife geht, dann ist die Spitze des Stack 1, wenn Ihr Programm die FOR/NEXT-Schleife wieder verlässt.

Tabelle 6-56 Gültige Operanden für die für die Operationen Programmschleife mit FOR und Ende Programmschleife mit NEXT

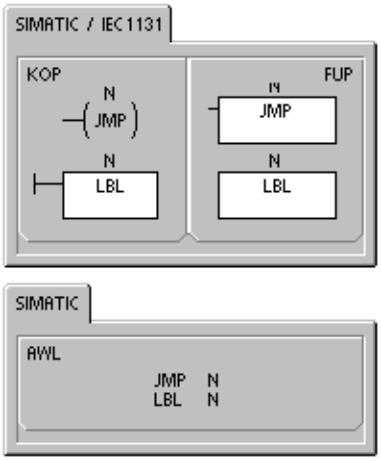
Eingänge/Ausgänge	Datentypen	Operanden
INDX	INT	EW, AW, VW, MW, SMW, SW, T, Z, LW, AC, *VD, *LD, *AC
INIT, FINAL	INT	VW, EW, AW, MW, SMW, SW, T, Z, LW, AC, AEW, *VD, *LD, *AC, Konstante

Beispiel: Operationen Programmschleife mit FOR und Ende Programmschleife mit NEXT




```
Netzwerk 1      //Wird E2.0 eingeschaltet, wird die äußere
                //Schleife
                //(Pfeil 1) 100 mal ausgeführt.
LD      E2.0
FOR     VW100, +1, +100

Netzwerk 2      //Die innere Schleife (Pfeil 2) wird für jede Bear
                //beitung
                //der äußeren Schleife 2mal ausgeführt,
                //wenn E2.1 eingeschaltet ist.
LD      E2.1
FOR     VW225, +1, +2

Netzwerk 3      //Ende der Schleife 2.
NEXT

Netzwerk 4      //Ende der Schleife 1.
NEXT
```

Sprungoperationen



Die Operation Zu Sprungmarke springen (JMP) verzweigt innerhalb des Programms zu der angegebenen Sprungmarke N.

Die Operation Sprungmarke definieren (LBL) gibt das Ziel N an, zu dem gesprungen werden soll.

Die Operation Zu Sprungmarke springen können Sie im Hauptprogramm, in Unterprogrammen und in Interruptprogrammen verwenden. Die Operation Zu Sprungmarke springen und die entsprechende Operation Sprungmarke definieren müssen sich immer im gleichen Segment des Code befinden (entweder im Hauptprogramm, in einem Unterprogramm oder in einem Interruptprogramm).

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.

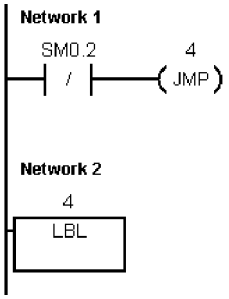
Sie können die Operation Zu Sprungmarke springen in einem SCR-Segment anorden, dann muss sich aber auch die entsprechende

Operation Sprungmarke definieren im gleichen SCR-Segment befinden.

Tabelle 6-57 Gültige Operanden für die Sprungoperationen

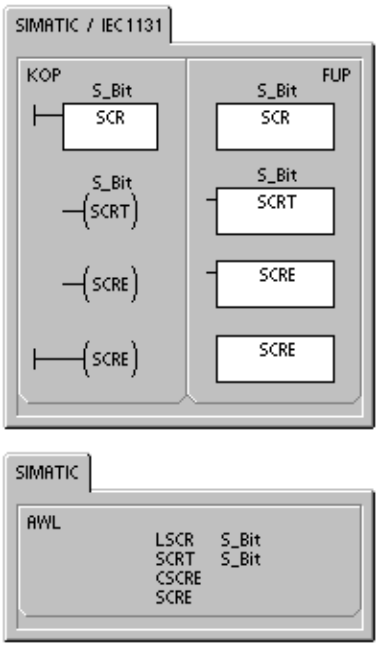
Eingänge/ Ausgänge	Datentypen	Operanden
N	WORD	Konstante (0 bis 255)

Beispiel: Operation Zu Sprungmarke springen




```
Netzwerk 1      //Sind keine remanenten Daten verloren gegangen, zu LBL4 sprin
                //gen.
LDN     SM0.2
JMP     4
Netzwerk 2
LBL     4
```

Operationen für das Ablaufsteuerungsrelais (SCR)



Die Operationen für das Ablaufsteuerungsrelais bieten Ihnen eine einfache und dennoch leistungsstarke Programmiertechnik für die Schrittsteuerung, die sich ganz natürlich in ein KOP, FUP oder AWL-Programm integrieren lässt.

Wenn Ihre Anwendung aus einer Folge von Funktionen besteht, die wiederholt ausgeführt werden müssen, können Sie Ihr Programm mit Ablaufsteue-
rungsrelais so strukturieren, dass es Ihrer Anwendung direkt entspricht. Auf diese Weise können Sie Ihre Anwendung schneller und einfacher programmieren und testen.

Die Operation Ablaufsteuerungsrelais laden (LSCR) lädt den Wert des S-Bit, welches von der Operation N angegeben wird, in das Ablaufsteuerungsrelais (SCR) und den logischen Stack.

Das SCR-Segment wird durch das Ergebnis des SCR-Stack aktiviert bzw. deaktiviert. Der Wert des SCR-Stack wird in die Spitze des logischen Stack kopiert, so dass die Boxen und Spulen ohne zwischengeschalteten Kontakt direkt an die linke Stromschiene angeschlossen werden können.

Einschränkungen

Wenn Sie mit Ablaufsteuerungsrelais arbeiten, müssen Sie folgende Einschränkungen beachten:

- Sie können das gleiche S-Bit nicht in mehr als einer Programmeinheit verwenden. Beispiel: Wenn Sie S0.1 im Hauptprogramm verwenden, können Sie es nicht im Unterprogramm einsetzen.
- Sie können nicht in ein oder aus einem SCR-Segment springen. Sie können die Operationen Zu Sprungmarke springen und Sprungmarke definieren jedoch einsetzen, um SCR-Segmente zu überspringen oder um innerhalb eines SCR-Segments zu springen.
- Sie dürfen die Operation END nicht in einem SCR-Segment einsetzen.

Tabelle 6-58 Gültige Operanden für die Operationen für das Ablaufsteuerungsrelais

Eingänge/Ausgänge	Datentypen	Operanden
S_Bit	BOOL	S

Bild 6-32 zeigt den S-Stack und den logischen Stack sowie die Auswirkungen der Operation Ablaufsteuerungsrelais laden. Folgendes gilt für Operationen mit Ablaufsteuerungsrelais:

- Die Operation Ablaufsteuerungsrelais laden (LSCR) kennzeichnet den Beginn eines SCR-Segments. Die Operation Ende Ablaufsteuerungsrelais (SCRE) kennzeichnet das Ende eines SCR-Segments. Alle Operationen zwischen der Operation Ablaufsteuerungsrelais laden und der Operation Ende Ablaufsteuerungsrelais sind hinsichtlich der Ausführung von dem Wert des S-Stack abhängig. Alle Operationen zwischen der Operation Ende Ablaufsteuerungsrelais und der nächsten Operation Ablaufsteuerungsrelais laden sind nicht von dem Wert des S-Stack abhängig.
- Die Operation Transition Ablaufsteuerungsrelais (SCRT) bietet die Möglichkeit, die Steuerung vom aktiven SCR-Segment zu einem anderen SCR-Segment zu übergeben.

  Die Operation Transition Ablaufsteuerungsrelais wird bei Signalfluss ausgeführt. Sie setzt das S-Bit des gerade aktiven Segments zurück und setzt das S-Bit des angegebenen Segments. Durch das Rücksetzen des S-Bits des aktiven Segments wird der S-Stack zur Zeit der Ausführung der Operation Transition Ablaufsteuerungsrelais nicht beeinflusst. Deshalb bleibt das SCR-Segment aktiviert, bis es verlassen wird.

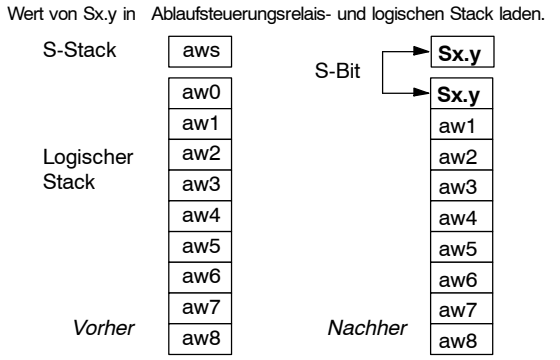
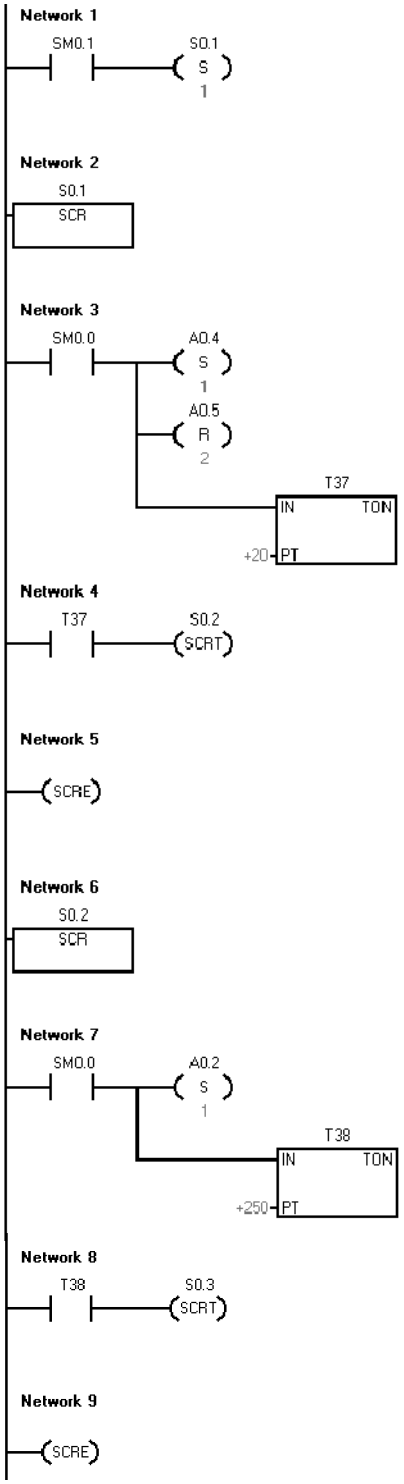


Bild 6-32 Auswirkungen der Operation LSCR auf den logischen Stack

- Die Operation Ablaufsteuerungsrelais bedingt beenden (CSCRE) bietet die Möglichkeit, ein aktives SCR-Segment zu verlassen, ohne die Operationen zwischen den Operationen Ablaufsteuerungsrelais bedingt beenden und Ende Ablaufsteuerungsrelais ausführen zu müssen. Die Operation Ablaufsteuerungsrelais bedingt beenden hat keinen Einfluss auf das S-Bit oder den S-Stack.

In dem folgenden Beispiel setzt der Sondermerker SM0.1 (Merker des ersten Zyklus) S0.1. S0.1 ist im ersten Zyklus der aktive 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).

Beispiel: Operationen für Ablaufsteuerungsrelais




```
Netzwerk 1      //Im ersten Zyklus Schritt 1 aktivieren.
LD      SM0.1
S       S0.1, 1

Netzwerk 2      //Beginn des Steuerungsbereichs für Schritt 1.
LSCR    S0.1

Netzwerk 3      //Signale für Straße 1 steuern:
                //1. Setzen: Rotes Licht einschalten.
                 //2. Rücksetzen: Gelbes Licht ausschalten und grünes Licht

                //einschalten.
                //3. Verzögerungszeit von 2 Sekunden starten.
LD      SM0.0
S       A0.4, 1
R       A0.5, 2
TON     T37, +20

Netzwerk 4      //Nach einer Verzögerung von 2 Sekunden zu Schritt 2
                //weiterschalten.
LD       T37
SCRT    S0.2

Netzwerk 5      //Ende des SCR-Bereichs für Schritt 1.
SCRE

Netzwerk 6      //Beginn des Steuerungsbereichs für Schritt 2.
LSCR    S0.2

Netzwerk 7      //Signale für Straße 2 steuern:
                //1. Setzen: Grünes Licht einschalten.
                //2. Verzögerungszeit von 25 Sekunden starten.
LD      SM0.0
S       A0.2, 1
TON     T38, +250

Netzwerk 8      //Nach einer Verzögerung von 25 Sekunden zu Schritt 3
                //weiterschalten.
LD       T38
SCRT    S0.3

Netzwerk 9      //Ende des SCR-Bereichs für Schritt 2.
SCRE
```

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. Dies zeigt Bild 6-33.

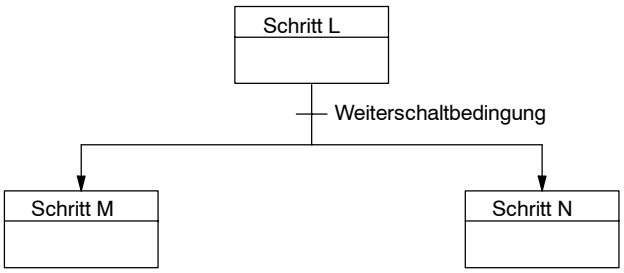
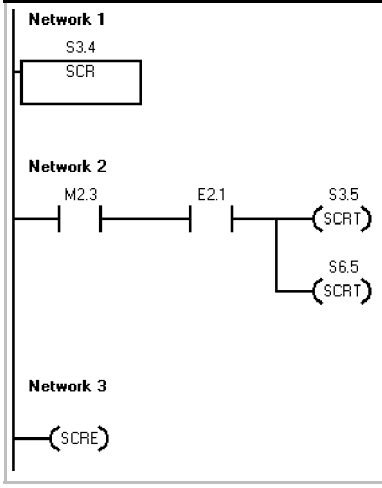


Bild 6-33 Teilung von Ablaufketten

Die Teilung von Ablaufketten kann in ein Ablaufsteuerungsprogramm implementiert werden, indem mehrere Operationen SCRT von der gleichen Weiterschaltbedingung aktiviert werden (siehe folgendes Beispiel).

Beispiel: Teilung von Ablaufketten




```
Netzwerk 1      //Beginn des Steuerungsbereichs für Schritt L
LSCR    S3.4

Netzwerk 2
LD      M2.3
U       E2.1
SCRT    S3.5            //Weiterschalten zu Schritt M
SCRT    S6.5            //Weiterschalten zu Schritt N

Netzwerk 3      //Ende des SCR-Bereichs für Schritt L
SCRE
```

Zusammenführung von Ablaufketten

Eine ähnliche Situation wie bei der Teilung von Ablaufketten 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 6-34 zeigt die Zusammenführung von zwei Ablaufketten.

Die Zusammenführung von Ablaufketten kann in eine Ablaufsteuerung implementiert werden, indem von Schritt L zu Schritt L' und von Schritt M zu M' weitergeschaltet wird. Sind beide SCR-Bits, die L' und M' darstellen, wahr, dann kann der Schritt N aktiviert werden (siehe folgendes Beispiel).

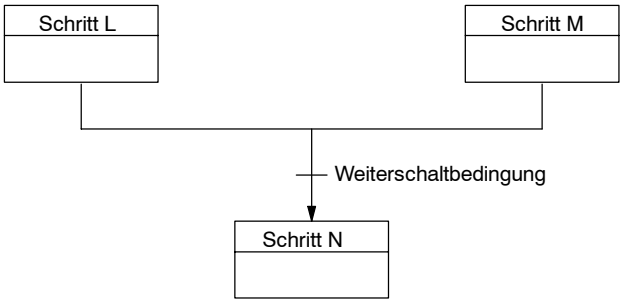
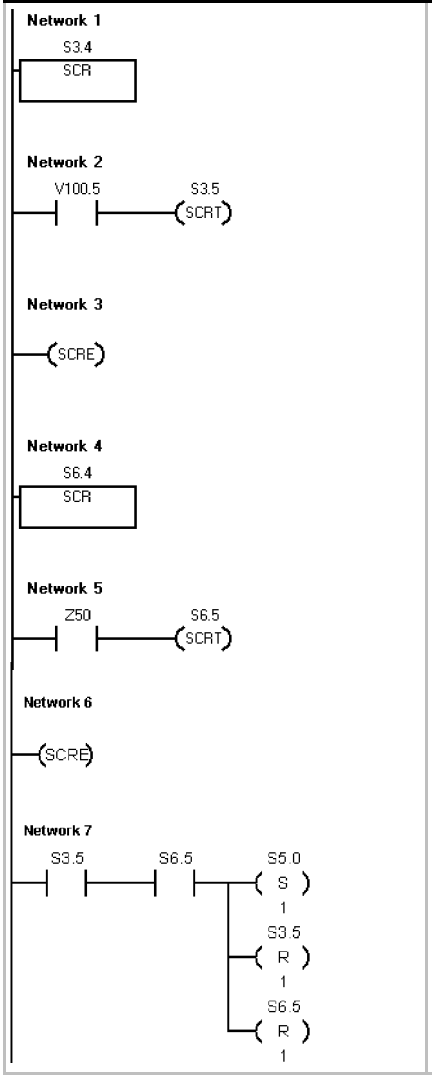


Bild 6-34 Zusammenführung von Ablaufketten

Beispiel: Zusammenführung von Ablaufketten




```
Netzwerk 1      //Beginn des Steuerungsbereichs für Schritt L
LSCR    S3.4

Netzwerk 2      //Weiterschalten zu Schritt L'
LD      V100.5
SCRT    S3.5

Netzwerk 3      //Ende des SCR-Bereichs für Schritt L
SCRE

Netzwerk 4      //Beginn des Steuerungsbereichs für Schritt M
LSCR    S6.4

Netzwerk 5      //Weiterschalten zu Schritt M'
LD      Z50
SCRT    S6.5

Netzwerk 6      //Ende des SCR-Bereichs für Schritt M
SCRE

Netzwerk 7      //Wenn Schritt L' und Schritt M' aktiviert sind:
                //1. Schritt N aktivieren (S5.0)
                //2. Schritt L' rücksetzen (S3.5)
                //3. Schritt M' rücksetzen (S6.5)
LD      S3.5
U       S6.5
S       S5.0, 1
R       S3.5, 1
R       S6.5, 1
```

In anderen Situationen kann eine Ablaufkette zu einer von mehreren möglichen Ablaufketten umgeleitet werden. Dies richtet sich danach, welche Weiterschaltbedingung als erste wahr wird. Eine solche Situation und das entsprechende SCR-Programm zeigt Bild 6-35.

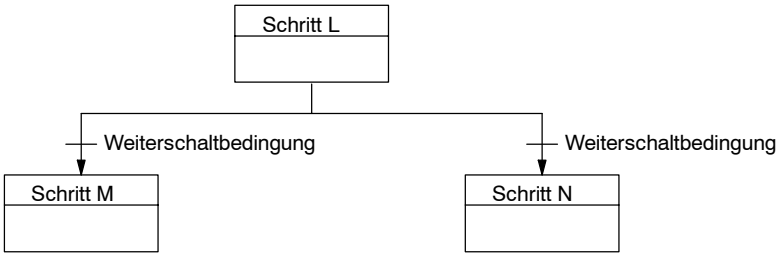
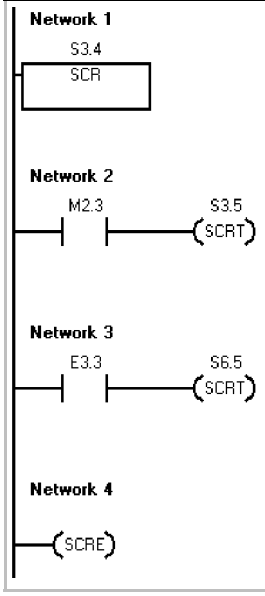


Bild 6-35 Teilung einer Ablaufkette je nach Weiterschaltbedingung

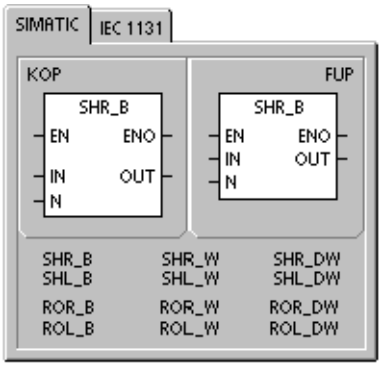
Beispiel: Weiterschaltbedingungen




```
Netzwerk 1      //Beginn des Steuerungsbereichs für Schritt L
LSCR    S3.4

Netzwerk 2      //Weiterschalten zu Schritt M
LD      M2.3
SCRT    S3.5

Netzwerk 3      //Weiterschalten zu Schritt N
LD      E3.3
SCRT    S6.5

Netzwerk 4      //Ende des SCR-Bereichs für Schritt L
SCRE
```

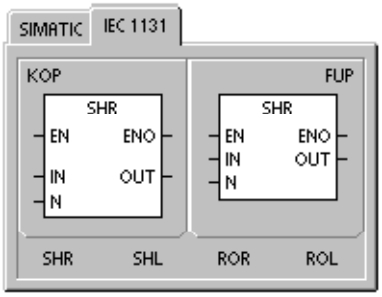
Schiebe- und Rotieroperationen

Operationen Rechts schieben und Links schieben



Die Schiebeoperationen schieben den Eingangswert IN um die Schiebezahl N nach rechts bzw. links und laden das Ergebnis in den Ausgang OUT.

Die Schiebeoperationen belegen die Plätze der hinausgeschobenen Bits mit Nullen. Ist die Schiebezahl (N) größer als oder gleich der maximalen Schiebezahl (8 bei Byteoperationen, 16 bei Wortoperationen und 32 bei Doppelwortoperationen), dann wird der Wert um die bei der Operation maximal mögliche Schiebezahl geschoben. Ist der Schiebewert größer als 0, dann nimmt der Überlaufmerker (SM1.1) den Wert des zuletzt herausgeschobenen Bit an. Der Nullmerker (SM1.0) wird gesetzt, wenn das Ergebnis der Schiebeoperation Null ist.



Byteoperationen haben kein Vorzeichen. Bei Wort- und Doppelwortoperationen wird das Vorzeichenbit geschoben, wenn Sie mit Datentypen mit Vorzeichen arbeiten.

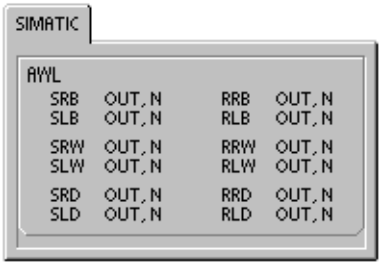
Fehlerbedingungen, die ENO = 0 setzen

- 0006 (Indirekte Adresse)

Betroffene Sondermerker:

- SM1.0 (Null)
- SM1.1 (Überlauf)

Operationen Rechts rotieren und Links rotieren



Die Rotieroperationen rotieren den Eingangswert (IN) um die Schiebezahl (N) nach rechts bzw. links und laden das Ergebnis in die Adresse im Speicher (OUT). Die Rotierfunktion ist kreisförmig.

Ist die Schiebezahl größer als oder gleich der maximalen Schiebezahl für die Operation (8 bei Byteoperationen, 16 bei Wortoperationen und 32 bei Doppelwortoperationen), dann führt die S7-200 eine Moduloperation mit der Schiebezahl aus, um eine gültige Schiebezahl zu erhalten, bevor rotiert wird. Das Ergebnis ist eine Schiebezahl von 0 bis 7 bei Byteoperationen, von 0 bis 15 bei Wortoperationen und von 0 bis 31 bei Doppelwortoperationen.

Ist die Schiebezahl gleich Null, dann wird nicht rotiert. Wird die Rotieroperation ausgeführt, dann wird der Wert des zuletzt rotierten Bit in das Überlaufbit (SM1.1) kopiert.

Ist die Schiebezahl kein ganzzahliges Vielfaches von 8 (bei Byteoperationen), 16 (bei Wortoperationen) oder 32 (bei Doppelwortoperationen), dann wird das zuletzt rotierte Bit in den Überlaufmerker (SM1.1) kopiert. Der Nullmerker (SM1.0) wird gesetzt, wenn der zu rotierende Wert Null ist.

Byteoperationen haben kein Vorzeichen. Bei Wort- und Doppelwortoperationen wird das Vorzeichenbit geschoben, wenn Sie mit Datentypen mit Vorzeichen arbeiten.

Fehlerbedingungen, die ENO = 0 setzen

- 0006 (Indirekte Adresse)

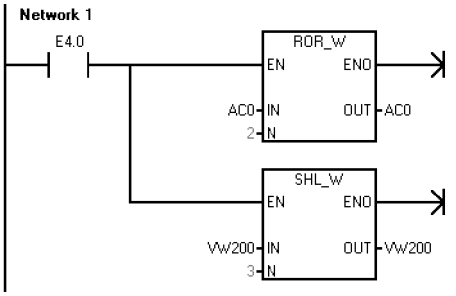
Betroffene Sondermerker:

- SM1.0 (Null)
- SM1.1 (Überlauf)

Tabelle 6-59 Gültige Operanden für die Schiebe- und Rotieroperationen

Eingänge/ Ausgänge	Datentypen	Operanden
IN	BYTE	EB, AB, VB, MB, SMB, SB, LB, AC, *VD, *LD, *AC, Konstante
	WORD	EW, AW, VW, MW, SMW, SW, T, Z, LW, AC, AEW, *VD, *LD, *AC, Konstante
	DWORD	ED, AD, VD, MD, SMD, SD, LD, AC, HC, *VD, *LD, *AC, Konstante
OUT	BYTE	EB, AB, VB, MB, SMB, SB, LB, AC, *VD, *LD, *AC
	WORD	EW, AW, VW, MW, SMW, SW, T, Z, LW, AC, *VD, *LD, *AC
	DWORD	ED, AD, VD, MD, SMD, SD, LD, AC, *VD, *LD, *AC
N	BYTE	EB, AB, VB, MB, SMB, SB, LB, AC, *VD, *LD, *AC, Konstante

Beispiel: Schiebe- und Rotieroperationen



Netzwerk 1

```
LD      E4.0
RRW     AC0, 2
SLW     VW200, 3
```

Rotieren

	Vor dem Rotieren	Überlauf
AC0	0100 0000 0000 0001	x

	Nach dem ersten Rotieren	Überlauf
AC0	1010 0000 0000 0000	1

	Nach dem zweiten Rotieren	Überlauf
AC0	0101 0000 0000 0000	0

Nullmerker (SM1.0) = 0
Überlaufmerker (SM1.1) = 0

Schieben

	Vor dem Schieben	Überlauf
VW200	1110 0010 1010 1101	x

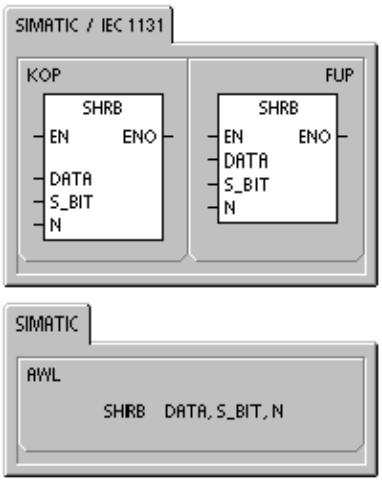
	Nach dem ersten Schieben	Überlauf
VW200	1100 0101 0101 1010	1

	Nach dem zweiten Schieben	Überlauf
VW200	1000 1010 1011 0100	1

	Nach dem dritten Schieben	Überlauf
VW200	0001 0101 0110 1000	1

Nullmerker (SM1.0) = 0
Überlaufmerker (SM1.1) = 1

Operation Wert in Schieberegister schieben



Die Operation Wert in Schieberegister schieben schiebt einen Wert in das Schieberegister. Diese Operation dient dazu, einen Produktfluss oder Daten auf einfache Weise in Reihenfolge zu bringen und zu steuern. Mit dieser Operation können Sie einmal pro Zyklus das gesamte Register um ein Bit verschieben.

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).

Jedes von der Operation SHRB hinausgeschobene Bit wird im Überlaufmerker (SM1.1) abgelegt.

Diese Operation wird definiert durch das niederwertigste Bit (S_BIT) und die Anzahl der Bits, die durch die Länge (N) angegeben wird.

Fehlerbedingungen, die ENO = 0 setzen

- 0006 (Indirekte Adresse)
- 0091 (Operand außerhalb des Bereichs)
- 0092 (Fehler in Zählfeld)

Betroffene Sondermerker:

- SM1.1 (Überlauf)

Tabelle 6-60 Gültige Operanden für die Operation Wert in Schieberegister schieben

Eingänge/ Ausgänge	Datentypen	Operanden
DATA, S_BIT	BOOL	E, A, V, M, SM, S, T, Z, L
N	BYTE	EB, AB, VB, MB, SMB, SB, LB, AC, *VD, *LD, *AC, Konstante

Berechnen Sie die Adresse des höchstwertigen Bit im Schieberegister (MSB.b) mit Hilfe folgender Gleichung:

MSB.b = *[(Byte von S_BIT) + ([N] - 1 + (Bit von S_BIT)) / 8].[Divisionsrest der Division durch 8]*

Beispiel: Ist S_BIT gleich V33.4 und N ist 14, dann zeigt die folgende Berechnung, dass MSB.b gleich V35.1 ist.

MSB.b = V33 + ([14] - 1 +4)/8
= V33 + 17/8
= V33 + 2 mit einem Divisionsrest von 1
= V35.1

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. Die hinausgeschobenen Daten werden im Überlaufmerker (SM1.1) abgelegt.

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 6-36 zeigt das Schieben von negativen und positiven Werten von N.

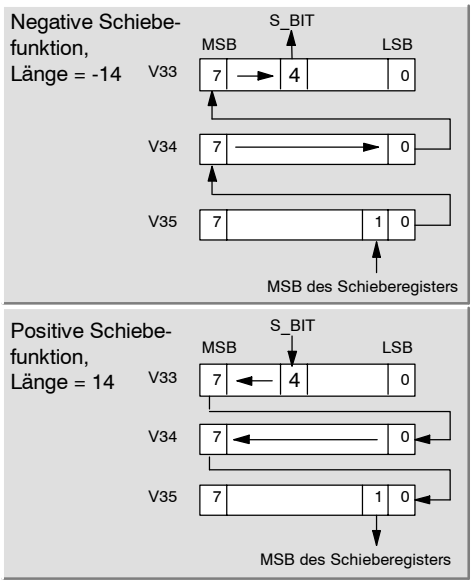
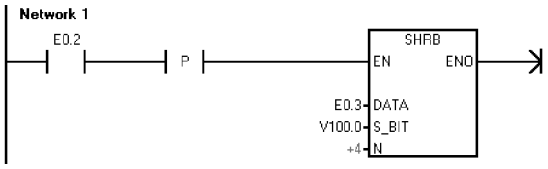


Bild 6-36 Zugänge und Abgänge im Schieberegister

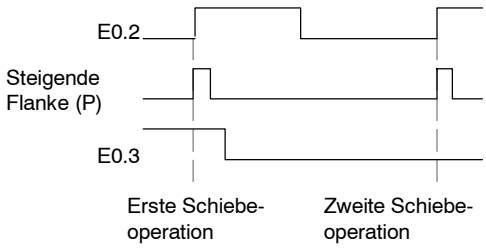
Beispiel: Operation Wert in Schieberegister schieben

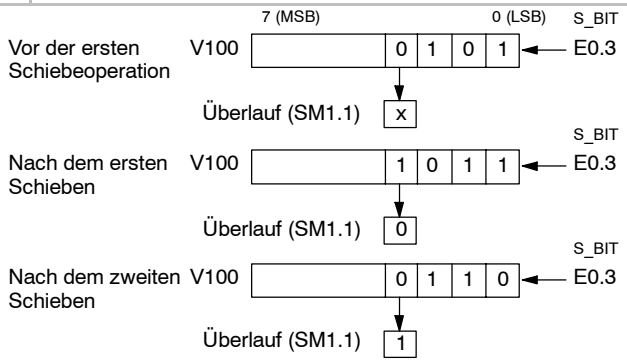


Netzwerk 1

```
LD      E0.2
EU
SHRB    E0.3, V100.0, +4
```

Impulsdiagramm





Operation Bytes im Wort tauschen

Die Operation Bytes im Wort tauschen tauscht das höchstwertige Byte mit dem niederwertigsten Byte des Worts IN.

Fehlerbedingungen, die ENO = 0 setzen

- 0006 (Indirekte Adresse)

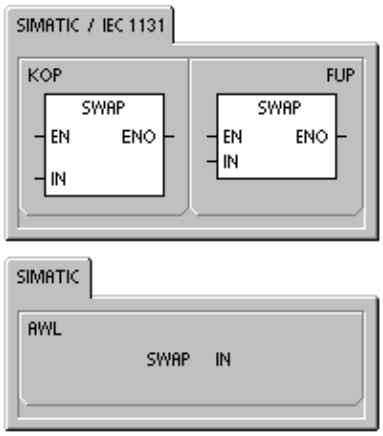
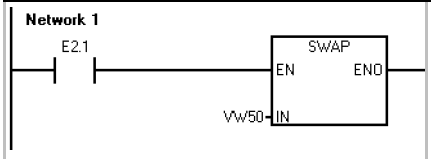


Tabelle 6-61 Gültige Operanden für die Operation Bytes im Wort tauschen

Eingänge/ Ausgänge	Datentypen	Operanden
IN	WORD	EW, AW, VW, MW, SMW, SW, T, Z, LW, AC,*VD, *LD, *AC

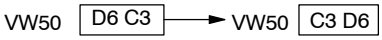
Beispiel: Operation Bytes im Wort tauschen



Netzwerk 1

```
LD      E2.1
SWAP    VW50
```

Bytes im Wort tauschen



Zeichenkettenoperationen

Zeichenkettenlänge

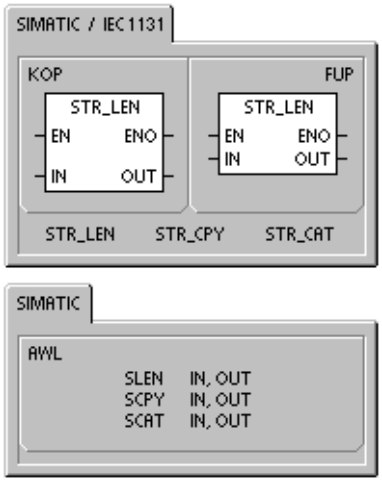
Die Operation Zeichenkettenlänge (SLEN) gibt die Länge der von IN angegebenen Zeichenkette an.

Zeichenkette kopieren

Die Operation Zeichenkette kopieren (SCPY) kopiert die von IN angegebene Zeichenkette in die von OUT angegebene Zeichenkette.

Zeichenkette verketten

Die Operation Zeichenkette verketten (SCAT) hängt die von IN angegebene Zeichenkette an das Ende der von OUT angegebenen Zeichenkette an.



Sondermerker und ENO

Bei den Operationen Zeichenkettenlänge, Zeichenkette kopieren und Zeichenkette verketten wirken sich die folgenden Bedingungen auf ENO aus.

Fehlerbedingungen, die ENO = 0 setzen

- 0006 (Indirekte Adresse)
- 0091 (Bereichsfehler)

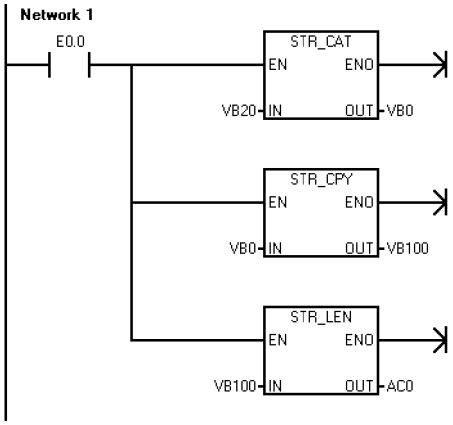
Tabelle 6-62 Gültige Operanden für die Operation Zeichenkettenlänge

Eingänge/ Ausgänge	Datentypen	Operanden
IN	BYTE (Zeichenkette)	VB, LB, *VD, *LD, *AC
OUT	BYTE	EB, AB, VB, MB, SMB, SB, LB, AC, *VD, *LD, *AC

Tabelle 6-63 Gültige Operanden für die Operationen Zeichenkette kopieren und Zeichenkette verketten

Eingänge/ Ausgänge	Datentypen	Operanden
IN, OUT	BYTE (Zeichenkette)	VB, LB, *VD, *LD, *AC

Beispiel: Operationen Zeichenkette verketten, Zeichenkette kopieren und Zeichenkettenlänge




```
Netzwerk 1      //1. Zeichenkette in VB20 anhängen
                //   an Zeichenkette in VB0.
                //2. Zeichenkette an VB0 kopieren
                //   in neue Zeichenkette an VB100.
                //3. Länge der Zeichenkette holen,
                //   die an VB100 beginnt.
LD      E0.0
SCAT    VB20, VB0
STRCPY  VB0, VB100
STRLEN  VB100, AC0
```

Vor Ausführung des Programms

VB0						VB6
6	'H'	'E'	'L'	'L'	'O'	' '

VB20					VB25
5	'W'	'O'	'R'	'L'	'D'

Nach Ausführung des Programms

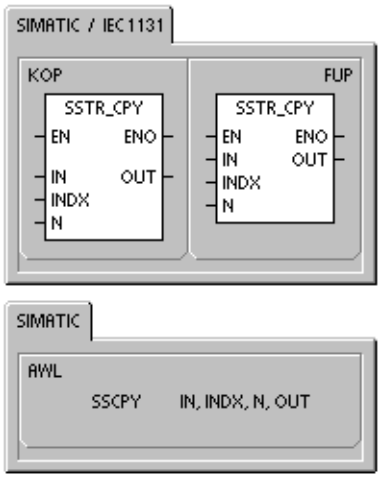
VB0											VB11
11	'H'	'E'	'L'	'L'	'O'	' '	'W'	'O'	'R'	'L'	'D'

VB100											VB111
11	'H'	'E'	'L'	'L'	'O'	' '	'W'	'O'	'R'	'L'	'D'

AC0
11

Teilzeichenkette aus Zeichenkette kopieren

Die Operation Teilzeichenkette aus Zeichenkette kopieren (SSCPY) kopiert die angegebene Anzahl Zeichen N aus der von IN angegebenen Zeichenkette, mit Beginn an Index INDX, in eine neue, von OUT angegebene Zeichenkette.



Fehlerbedingungen, die ENO = 0 setzen

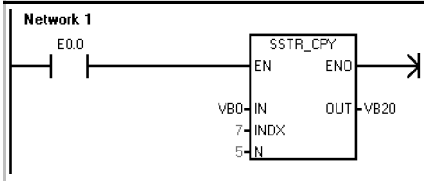
- 0006 (Indirekte Adresse)
- 0091 (Bereichsfehler)
- 009B (Index=0)

6

Tabelle 6-64 Gültige Operanden für die Operation Teilzeichenkette aus Zeichenkette kopieren

Eingänge/ Ausgänge	Datentypen	Operanden
IN, OUT	BYTE (Zeichenkette)	VB, LB, *VD, *LD, *AC
INDX, N	BYTE	EB, AB, VB, MB, SMB, SB, LB, AC, *VD, *LD, *AC, Konstante

Beispiel: Operation Teilzeichenkette aus Zeichenkette kopieren




```
Netzwerk 1      //Mit Beginn beim siebten Zeichen in der Zeichenkette an
                //VB0
                //5 Zeichen in neue Zeichenkette an VB20 kopieren.
LD      E0.0
SSCPY   VB0, 7, 5, VB20
```

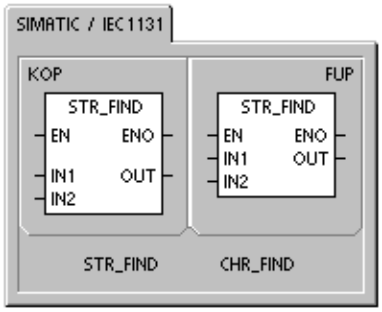
Vor Ausführung des Programms

VB0											VB11
11	'H'	'E'	'L'	'L'	'O'	' '	'W'	'O'	'R'	'L'	'D'

Nach Ausführung des Programms

VB20					VB25
5	'W'	'O'	'R'	'L'	'D'

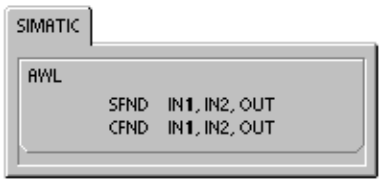
Zeichenkette in Zeichenkette suchen



Die Operation Zeichenkette in Zeichenkette suchen (SFND) sucht nach dem ersten Vorkommen der Zeichenkette IN2 in der Zeichenkette IN1. Die Suche beginnt an der von OUT angegebenen Anfangsposition. Wird eine Zeichenfolge gefunden, die genau der Zeichenkette IN2 entspricht, wird die Position des ersten Zeichens in der Zeichenfolge in OUT geschrieben. Wird die Zeichenkette IN2 nicht in der Zeichenkette IN1 gefunden, wird OUT = 0 gesetzt.

Fehlerbedingungen, die ENO = 0 setzen

- 0006 (Indirekte Adresse)
- 0091 (Bereichsfehler)
- 009B (Index=0)



Erstes Zeichen in Zeichenkette suchen

Die Operation Erstes Zeichen in Zeichenkette suchen (CFND) sucht in der Zeichenkette IN1 nach dem ersten Vorkommen eines beliebigen Zeichens aus dem in Zeichenkette IN2 beschriebenen Zeichensatz. Die Suche beginnt an der Anfangsposition OUT. Wird ein passendes Zeichen gefunden, wird die Position des Zeichens in OUT geschrieben. Wird kein passendes Zeichen gefunden, wird OUT = 0 gesetzt.

Fehlerbedingungen, die ENO = 0 setzen

- 0006 (Indirekte Adresse)
- 0091 (Bereichsfehler)
- 009B (Index=0)

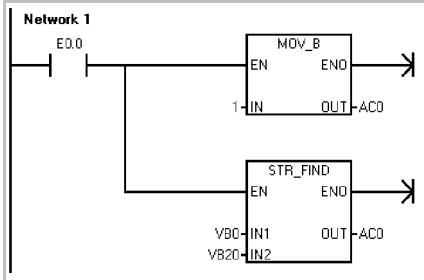
6

Tabelle 6-65 Gültige Operanden für die Operationen Zeichenkette in Zeichenkette suchen und Erstes Zeichen in Zeichenkette suchen

Eingänge/ Ausgänge	Datentypen	Operanden
IN1, IN2	BYTE (Zeichenkette)	VB, LB, *VD, *LD, *AC
OUT	BYTE	EB, AB, VB, MB, SMB, SB, LB, AC, *VD, *LD, *AC

Beispiel: Operation Zeichenkette in Zeichenkette suchen

Im folgenden Beispiel wird eine Zeichenkette, die an VB0 abgelegt ist, als Befehl zum Ein- und Ausschalten einer Pumpe eingesetzt. Die Zeichenkette 'On' ist an VB20 gespeichert und die Zeichenkette 'Off' ist an VB30 gespeichert. Das Ergebnis der
Operation Zeichenkette in Zeichenkette suchen wird in AC0 abgelegt (Parameter OUT). Ist das Ergebnis nicht 0, dann wurde die Zeichenkette 'On' in der Zeichenkette des Befehls (VB12) gefunden.




```
Netzwerk 1    //1. AC0 auf 1 setzen.
              //   (AC0 wird als Parameter OUT verwendet.)
              //2. Zeichenkette an VB0 nach Zeichenkette
              //   an VB20 ('On') durchsuchen, mit erster Position
              //   beginnen (AC0=1).
LD      E0.0
MOVB    1, AC0
SFND    VB0, VB20, AC0
```

VB0												
12	'T'	'u'	'R'	'n'	' '	'P'	'u'	'm'	'P'	' '	'O'	'n'

VB12

VB20			VB22
2	'O'	'n'	

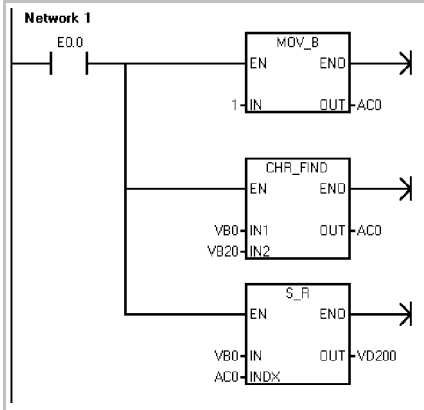
VB40			VB33
3	'O'	'f'	'f'

Wird die Zeichenkette in VB20 gefunden: AC0 = 11

Wird die Zeichenkette in VB20 nicht gefunden: AC0 = 0

Beispiel: Operation Zeichen in Zeichenkette suchen

Im folgenden Beispiel enthält die Zeichenkette an VB0 die Temperatur. Die Zeichenkette an VB20 speichert alle numerischen Zeichen (und + und -), die auf eine Temperatur in einer Zeichenkette verweisen können. Das Beispielprogramm sucht die Anfangsposition einer Nummer in der Zeichenkette und wandelt dann das numerische Zeichen in eine Realzahl um. VD200 speichert den Realzahlwert der Temperatur.




```
Netzwerk 1    //1. AC0 auf 1 setzen.
              //   (AC0 wird als Parameter OUT verwendet
              //   und zeigt auf die erste Position in der Zeichenkette.)
              //2. Numerisches Zeichen
              //   in der Zeichenkette an VB0 suchen.
              //3. Zeichenkette in Realzahl wandeln.
LD      E0.0
MOVB    1, AC0
CFND    VB0, VB20, AC0
STR     VB0, AC0, VD200
```

VB0												VB11
11	'T'	'e'	'm'	'P'	' '	' '	' '	'9'	'8'	'.'	'6'	'f'

VB20												VB32
12	'1'	'2'	'3'	'4'	'5'	'6'	'7'	'8'	'9'	'0'	'+'	'-'

Anfangsposition der in VB0 gespeicherten Temperatur: AC0 = 7

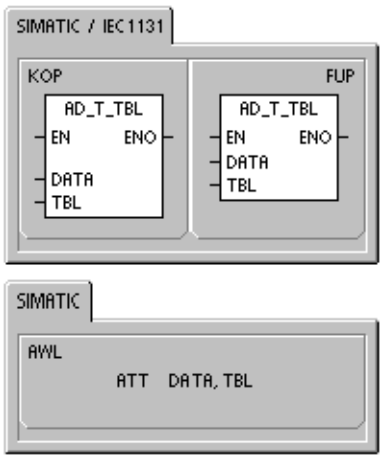
Realzahlwert der Temperatur: VD200 = 98.6

Tabellenoperationen

Wert in Tabelle eintragen

Die Operation Wert in Tabelle eintragen trägt Wortwerte (DATA) in eine Tabelle (TABLE) ein. Der erste Wert in der Tabelle gibt die maximale Länge der Tabelle (TL) an. Der zweite Wert (EC) gibt die Anzahl der Tabelleneinträge an. 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 enthalten.



Fehlerbedingungen, die ENO = 0 setzen

- SM1.4 (Tabellenüberlauf)
- 0006 (Indirekte Adresse)
- 0091 (Operand außerhalb des Bereichs)

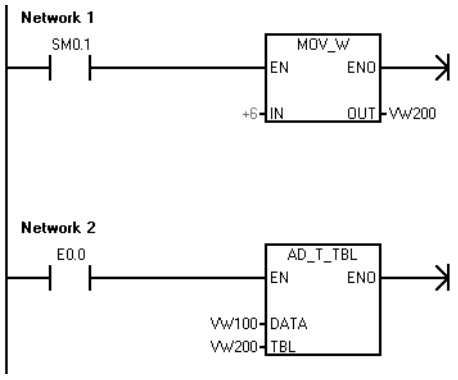
Betroffene Sondermerker:

- SM1.4 wird auf 1 gesetzt, wenn Sie versuchen, zu viele Werte in die Tabelle einzutragen.

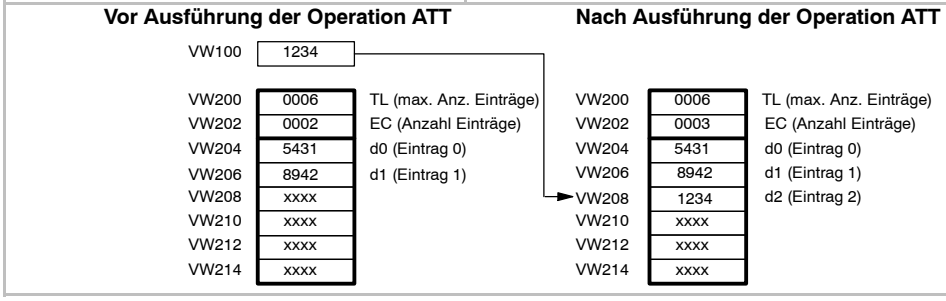
Tabelle 6-66 Gültige Operanden für die Tabellenoperationen

Eingänge/ Ausgänge	Datentypen	Operanden
DATEN	INT	EW, AW, VW, MW, SMW, SW, T, Z, LW, AC, AEW, *VD, *LD, *AC, Konstante
TBL	WORD	EW, AW, VW, MW, SMW, SW, T, Z, LW, *VD, *LD, *AC

Beispiel: Operation Wert in Tabelle eintragen

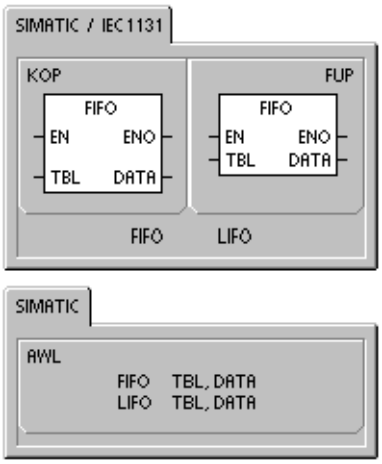



```
Netzwerk 1      //Maximale Tabellenlänge laden
LD      SM0.1
MOVW    +6, VW200

Netzwerk 2
LD      E0.0
ATT     VW100, VW200
```



Ersten Wert aus Tabelle löschen und Letzten Wert aus Tabelle löschen

Eine Tabelle kann maximal 100 Einträge enthalten.



Ersten Wert aus Tabelle löschen

Die Operation Ersten Wert aus Tabelle löschen (FIFO) überträgt den ältesten (den ersten) Eintrag in einer Tabelle in die Ausgangsadresse, indem sie den ersten Eintrag in der Tabelle (TBL) löscht und den Wert in die von DATA angegebene Adresse überträgt. 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.

Letzten Wert aus Tabelle löschen

Die Operation Letzten Wert aus Tabelle löschen (FIFO) überträgt den neuesten (den letzten) Eintrag in einer Tabelle in die Ausgangsadresse, indem sie den letzten Eintrag in der Tabelle (TBL) löscht und den Wert in die von DATA angegebene Adresse überträgt. Die Anzahl der Einträge (EC) verringert sich jedesmal, wenn diese Operation ausgeführt wird, um 1.

Fehlerbedingungen, die ENO = 0 setzen

- SM1.5 (Leere Tabelle)
- 0006 (Indirekte Adresse)
- 0091 (Operand außerhalb des Bereichs)

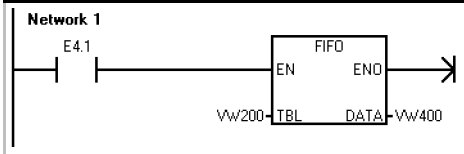
Betroffene Sondermerker:

- SM1.5 wird auf 1 gesetzt, wenn Sie versuchen, einen Eintrag in einer leeren Tabelle zu löschen.

Tabelle 6-67 Gültige Operanden für die Operationen Ersten Wert aus Tabelle löschen und Letzten Wert aus Tabelle löschen

Eingänge/ Ausgänge	Datentypen	Operanden
TBL	WORD	EW, AW, VW, MW, SMW, SW, T, Z, LW, *VD, *LD, *AC
DATEN	INT	EW, AW, VW, MW, SMW, SW, T, Z, LW, AC, AAW, *VD, *LD, *AC

Beispiel: Operation Ersten Wert aus Tabelle löschen



Netzwerk 1

```
LD      E4.1
FIFO    VW200, VW400
```

Vor Ausführung der Operation FIFO

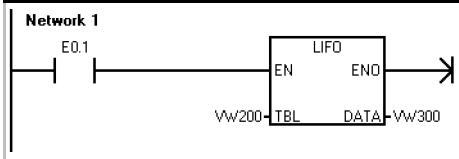
VW200	0006	TL (max. Anzahl Einträge)
VW202	0003	EC (Anzahl der Einträge)
VW204	5431	d0 (Eintrag 0)
VW206	8942	d1 (Eintrag 1)
VW208	1234	d2 (Eintrag 2)
VW210	xxxx	
VW212	xxxx	
VW214	xxxx	

VW400 5431

Nach Ausführung der Operation F

VW200	0006	TL (max. Anzahl Einträge)
VW202	0002	EC (Anzahl der Einträge)
VW204	8942	d0 (Eintrag 0)
VW206	1234	d1 (Eintrag 1)
VW208	xxxx	
VW210	xxxx	
VW212	xxxx	
VW214	xxxx	

Beispiel: Operation Letzten Wert aus Tabelle löschen



Netzwerk 1

```
LD      E0.1
LIFO    VW200, VW300
```

Vor Ausführung der Operation LIFO

VW200	0006	TL (max. Anzahl Einträge)
VW202	0003	EC (Anzahl der Einträge)
VW204	5431	d0 (Eintrag 0)
VW206	8942	d1 (Eintrag 1)
VW208	1234	d2 (Eintrag 2)
VW210	xxxx	
VW212	xxxx	
VW214	xxxx	

VW300 1234

Nach Ausführung der Operation LI

VW200	0006	TL (max. Anzahl Einträge)
VW202	0002	EC (Anzahl der Einträge)
VW204	5431	d0 (Eintrag 0)
VW206	8942	d1 (Eintrag 1)
VW208	xxxx	
VW210	xxxx	
VW212	xxxx	
VW214	xxxx	

Speicher mit Bitmuster belegen

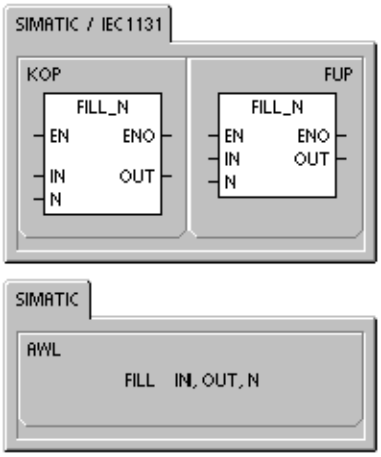
Die Operation Speicher mit Bitmuster belegen (FILL) schreibt den in Adresse IN enthaltenen Wortwert in N aufeinander folgende Wörter mit

Beginn an Adresse OUT.

N kann zwischen 1 und 255 liegen.

Fehlerbedingungen, die ENO = 0 setzen

- 0006 (Indirekte Adresse)
- 0091 (Operand außerhalb des Bereichs)

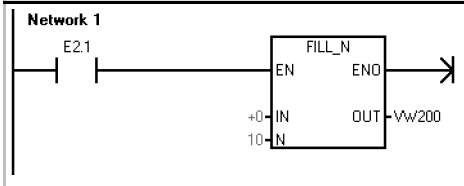


6

Tabelle 6-68 Gültige Operanden für die Operation Speicher mit Bitmuster belegen

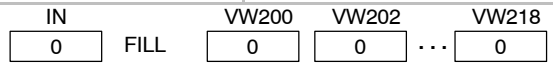
Eingänge/ Ausgänge	Datentypen	Operanden
IN	INT	EW, AW, VW, MW, SMW, SW, T, Z, LW, AC, AEW, *VD, *LD, *AC, Konstante
N	BYTE	EB, AB, VB, MB, SMB, SB, LB, AC, *VD, *LD, *AC, Konstante
OUT	INT	EW, AW, VW, MW, SMW, SW, T, Z, LW, AAW, *VD, *LD, *AC

Beispiel: Operation Speicher mit Bitmuster belegen



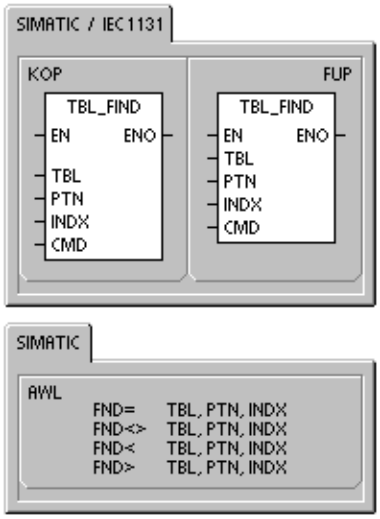
Netzwerk 1

```
LD      E2.1
FILL    +0, VW200, 10
```



Wert in Tabelle suchen

Die Operation Wert in Tabelle suchen (FND) sucht in einer Tabelle nach Daten, die bestimmten Kriterien entsprechen. Die Operation Wert in Tabelle suchen durchsucht die Tabelle (TBL) beginnend bei dem Tabelleneintrag INDX nach dem Datenwert oder Datenmuster PTN, der bzw. das den von CMD angegebenen Suchkriterien entspricht. Der Parameter CMD enthält einen numerischen Wert von 1 bis 4, der einer der Relationen =, <>, < oder > entspricht.



Wird ein Tabelleneintrag gefunden, der die Suchkriterien erfüllt, dann zeigt INDX auf den jeweiligen Eintrag in der Tabelle. Um den nächsten Tabellen-
eintrag zu suchen, der die Kriterien erfüllt, muss INDX um 1 erhöht werden, bevor die Suchoperation erneut ausgeführt werden kann. Wird kein passender Eintrag gefunden, dann ist der Wert von INDX gleich der Eintrags-
anzahl.

Eine Tabelle kann maximal 100 Einträge enthalten. Die Einträge in der Tabelle (der Bereich, der durchsucht werden soll) sind von 0 bis zum maximalen Wert 99 durchnummeriert.

Fehlerbedingungen, die ENO = 0 setzen

- 0006 (Indirekte Adresse)
- 0091 (Operand außerhalb des Bereichs)

Tabelle 6-69 Gültige Operanden für die Operation Wert in Tabelle suchen

Eingänge/ Ausgänge	Datentypen	Operanden
TBL	WORD	EW, AW, VW, MW, SMW, T, Z, LW, *VD, *LD, *AC
PTN	INT	EW, AW, VW, MW, SMW, SW, T, Z, LW, AC, AEW, *VD, *LD, *AC, Konstante
INDX	WORD	EW, AW, VW, MW, SMW, SW, T, Z, LW, AC, *VD, *LD, *AC
CMD	BYTE	(Konstante) 1: Gleich (=), 2: Ungleich (<>), 3: Kleiner als (<), 4: Größer als (>)

Tipp

Wenn Sie die Operation Wert in Tabelle suchen in Tabellen verwenden, die mit einer der Operationen Wert in Tabelle eintragen, Letzten Wert aus Tabelle löschen oder Ersten Wert aus Tabelle löschen erzeugt wurden, entsprechen sich die Eintragszahl und die Dateneinträge direkt. Das Wort für die maximale Anzahl Einträge, das für die Operationen Wert in Tabelle eintragen, Letzten Wert aus Tabelle löschen und Ersten Wert aus Tabelle löschen erforderlich ist, wird für die Operation Wert in Tabelle suchen nicht benötigt (siehe Bild 6-37).

Deshalb müssen Sie den Operand TBL einer Operation Wert in Tabelle suchen eine Wortadresse (zwei Bytes) höher setzen als den Operanden TBL einer entsprechenden Operation Wert in Tabelle eintragen, Letzten Wert aus Tabelle löschen oder Ersten Wert aus Tabelle löschen.

Tabellenformat für ATT, LIFO und FIFO

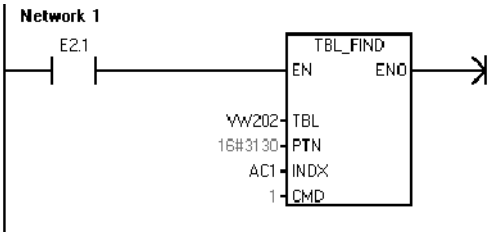
VW200	0006	TL (max. Anzahl Einträge)
VW202	0006	EC (Anzahl der Einträge)
VW204	xxxx	d0 (Eintrag 0)
VW206	xxxx	d1 (Eintrag 1)
VW208	xxxx	d2 (Eintrag 2)
VW210	xxxx	d3 (Eintrag 3)
VW212	xxxx	d4 (Eintrag 4)
VW214	xxxx	d5 (Eintrag 5)

Tabellenformat für TBL_FIND

VW202	0006	EC (Anzahl der Einträg
VW204	xxxx	d0 (Eintrag 0)
VW206	xxxx	d1 (Eintrag 1)
VW208	xxxx	d2 (Eintrag 2)
VW210	xxxx	d3 (Eintrag 3)
VW212	xxxx	d4 (Eintrag 4)
VW214	xxxx	d5 (Eintrag 5)

Bild 6-37 Unterschiedliche Tabellenformate bei Operation Wert in Tabelle suchen und den Operationen ATT, LIFO und FIFO

Beispiel: Operation Wert in Tabelle suchen



Netzwerk 1

```
LD      E2.1
FND=    VW202, 16#3130, AC1
```

Ist E2.1 eingeschaltet, dann wird die Tabelle nach einem Wert, der der Angabe 3130 in Hexadezimalziffern entspricht, durchsucht.

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)

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.

AC1 0 — AC1 muss auf 0 gesetzt sein, damit ab dem obersten Tabelleneintrag gesucht wird.

Tabelle durchsuchen

AC1 2 — AC1 enthält die Nummer des ersten Eintrags, der den Suchkriterien entspricht.

AC1 3 — Erhöhen Sie INDX um 1, bevor Sie die übrigen Einträge der Tabelle durchsuchen.

Tabelle durchsuchen

AC1 4 — AC1 enthält die Nummer des zweiten Eintrags, der den Suchkriterien entspricht.

AC1 5 — Erhöhen Sie INDX um 1, bevor Sie die übrigen Einträge der Tabelle durchsuchen.

Tabelle durchsuchen

AC1 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.

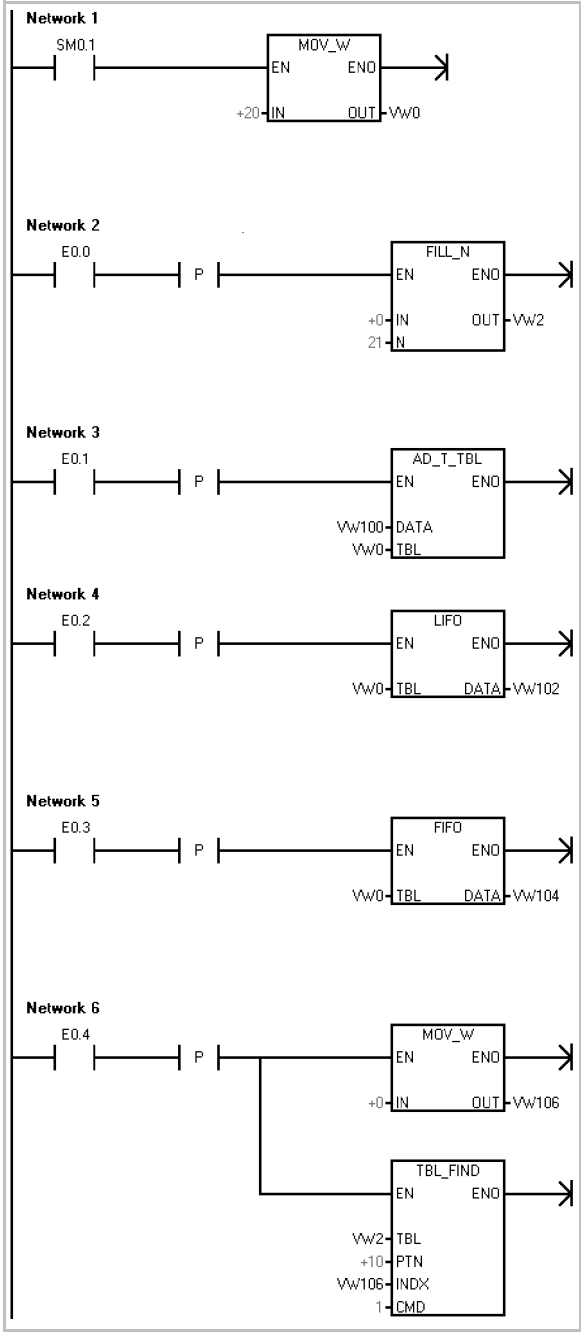
AC1 0 — Bevor Sie die Tabelle erneut durchsuchen können, müssen Sie den Wert von INDX auf 0 zurücksetzen.

Beispiel: Erstellen einer Tabelle

Das folgende Programm erstellt eine Tabelle mit 20 Einträgen. Die erste Adresse in der Tabelle gibt die Länge der Tabelle an (in diesem Fall 20 Einträge). Die zweite Adresse zeigt die aktuelle Anzahl der Einträge in der Tabelle an. Die anderen Adressen enthalten die Einträge. Eine Tabelle kann maximal 100 Einträge enthalten. Darin eingeschlossen sind nicht die Parameter für die maximale Länge der Tabelle und die tatsächliche Anzahl der Einträge (hier VW0 und VW2). Die tatsächliche Anzahl der Einträge in der Tabelle (hier VW2) wird von der CPU automatisch bei jedem Befehl erhöht bzw. verringert.

Bevor Sie mit einer Tabelle arbeiten, müssen Sie die maximale Anzahl Einträge in der Tabelle zuordnen. Ansonsten können Sie keine Einträge in die Tabelle eingeben. Stellen Sie außerdem sicher, dass alle Lese- und Schreibbefehle über Flanken aktiviert werden.

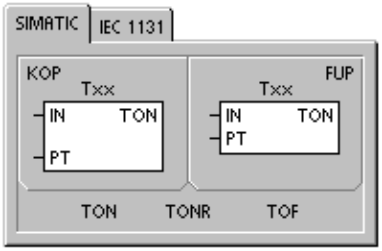
Zum Durchsuchen der Tabelle muss der Index (VW106) vor Ausführen der Suche auf 0 gesetzt werden. Wird eine Entsprechung gefunden, gibt der Index die Nummer des Tabelleneintrags an. Wird jedoch keine Entsprechung gefunden, gibt der Index die aktuelle Eintragszahl der Tabelle (VW2) an.

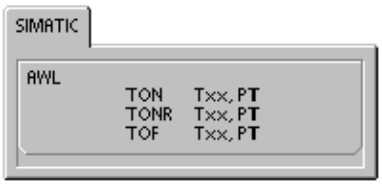



```
Netzwerk 1      //Tabelle mit 20 Einträgen mit Beginn an
                // Adresse 4 erstellen.
                //1.  Im ersten Zyklus maximale Länge der
                //Tabelle definieren.
LD              SM0.1
MOVW            +20, VW0
Netzwerk 2      //Tabelle mit Eingang E0.0 zurücksetzen.
                //Bei steigender Flanke an E0.0,
                //Adressen ab VW2 mit "+0" belegen.
LD      E0.0
EU
FILL    +0, VW2, 21
Netzwerk 3      //Wert mit Eingang E0.1 in Tabelle schrei
                //ben. Bei steigender Flanke an E0.1,
                //Wert aus Adresse VW100 in Tabelle ko
                //pieren.
LD      E0.1
EU
ATT     VW100, VW0
Netzwerk 4      //Ersten Tabellenwert mit E0.2 lesen.
                //Letzten Tabellenwert in VW102 übertra
                //gen.  Dadurch wird die Anzahl der
                // Einträge  verringert.
                //Bei steigender Flanke an E0.2,
                //letzten Tabellenwert in VW102 übertragen.
LD      E0.2
EU
LIFO    VW0, VW102
Netzwerk 5      //Letzten Tabellenwert mit E0.3 lesen.
                //Ersten Tabellenwert in VW102 übertragen.
                //Dadurch wird die Anzahl der Einträge
                //verringert.
                //Bei steigender Flanke an E0.0,
                //ersten Tabellenwert in VW104 übertragen.
LD      E0.3
EU
FIFO    VW0, VW104
Netzwerk 6      //Tabelle nach erster Adresse durchsuchen,
                //die den Wert 10 enthält.
                //1.  Bei steigender Flanke an E0.4,
                //    Pointer auf Index zurücksetzen.
                //2.  Tabelleneintrag suchen, der gleich 10
                //ist.
LD      E0.4
EU
MOVW    +0, VW106
FND=    VW2, +10, VW106
```

Zeitoperationen

SIMATIC: Zeitoperationen





Zeit als Einschaltverzögerung starten
Zeit als speichernde Einschaltverzögerung starten

Die Operationen Zeit als Einschaltverzögerung starten (TON) und Zeit als speichernde Einschaltverzögerung starten (TONR) zählen den Zeitwert, wenn der Freigabeeingang eingeschaltet ist. Die Nummer der Zeit (Txx) legt die Auflösung der Zeit fest.

Ausschaltverzögerung

Die Operation Zeit als Ausschaltverzögerung starten (TOF) dient dazu, das Ausschalten eines Ausgangs für einen bestimmten Zeitraum zu verzögern, nachdem der Eingang ausgeschaltet wurde. Die Nummer der Zeit (Txx) legt die Auflösung der Zeit fest.

Tabelle 6-70 Gültige Operanden für die SIMATIC Zeiten

Eingänge/ Ausgänge	Datentypen	Operanden
Txx	WORD	Konstante (T0 bis T255)
IN	BOOL	E, A, V, M, SM, S, T, Z, L, Signalfluss
PT	INT	EW, AW, VW, MW, SMW, SW, T, Z, LW, AC, AEW, *VD, *LD, *AC, Konstante

Tipp
Sie dürfen einer Operation Zeit als Ausschaltverzögerung starten (TOF) und einer Operation Zeit als Einschaltverzögerung starten (TON) nicht die gleiche Zeitnummer (Txx) geben. Sie dürfen beispielsweise nicht eine Zeit TON T32 und eine Zeit TOF T32 gleichzeitig vergeben.

Wie Sie in Tabelle 6-71 sehen, führen die drei Arten von Zeiten verschiedene Zeitmessungen aus:

- Sie können die Zeit als Einschaltverzögerung (TON) für ein einzelnes Intervall verwenden.
- Sie können die Zeit als speichernde Einschaltverzögerung (TONR) zum Summieren von mehreren Intervallen verwenden.
- Sie können die Zeit als Ausschaltverzögerung (TOF) zum Verlängern der Zeit über eine Ausschaltbedingung hinaus verwenden (z.B. zum Kühlen eines Motors nach dem Ausschalten).

Tabelle 6-71 Funktionsweise der Zeiten

Typ	Aktueller Wert > = Voreinstellung	Zustand des Freigabeeingangs (IN)	Einschalten / Erster Zyklus
TON	Zeitbit ein Aktueller Wert zählt bis 32767	EIN: Aktueller Wert zählt die Zeit AUS: Zeitbit aus, aktueller Wert = 0	Zeitbit aus Aktueller Wert = 0
TONR	Zeitbit ein Aktueller Wert zählt bis 32767	EIN: Aktueller Wert zählt die Zeit AUS: Zeitbit und aktueller Wert bleiben im letzten Zustand	Zeitbit aus Der aktuelle Wert kann gespeichert werden.[1]
TOF	Zeitbit aus Aktueller Wert = Voreinstellung, Zählstopp	EIN: Zeitbit ein, aktueller Wert = 0 AUS: Zeit zählt nach fallender Flanke (ein-aus)	Zeitbit aus Aktueller Wert = 0

[1] Der aktuelle Wert der speichernden Einschaltverzögerung kann so konfiguriert werden, dass er auch nach dem Ausschalten remanent bleibt. Informationen zu remanentem Speicher der S7-200 CPU finden Sie in Kapitel 4.

Programmierung Tips

Die Tipps für die Programmierung auf der Dokumentations-CD enthalten ein Beispielprogramm mit einer Operation Zeit als Einschaltverzögerung starten (TON) (siehe Tip 31).

Die Operationen TON und TONR zählen die Zeit, wenn der Freigabeeingang eingeschaltet ist. Ist der aktuelle Wert gleich oder größer als der voreingestellten Zeitwert, dann wird das Zählerbit eingeschaltet.

- ❏ Der aktuelle Wert einer Einschaltverzögerung (TON) wird gelöscht, wenn der Freigabeeingang ausgeschaltet wird, während der aktuelle Wert einer speichernden Einschaltverzögerung (TONR) gespeichert wird, wenn der Eingang ausgeschaltet ist.
- ❏ Mit der Zeit TONR können Sie die Zeit messen, wenn der Eingang ein- und ausgeschaltet wird. Mit der Operation Rücksetzen (R) löschen Sie den aktuellen Wert der Zeit TONR.
- ❏ Sowohl die Einschaltverzögerung (TON) als auch die speichernde Einschaltverzögerung (TONR) zählen weiter, nachdem der voreingestellte Zeitwert erreicht wurde, sowie beide auch aufhören zu zählen, wenn der Höchstwert 32.767 erreicht wird.

Die Operation TOF dient dazu, das Ausschalten eines Ausgangs für einen bestimmten Zeitraum zu verzögern, nachdem der Eingang ausgeschaltet wurde. Wird der Freigabeeingang eingeschaltet, wird sofort das Zeitbit eingeschaltet und der aktuelle Wert auf 0 gesetzt. Wird der Eingang ausgeschaltet, zählt die Zeit, bis die abgelaufene Zeit den voreingestellten Zeitwert erreicht.

- ❏ Wird der voreingestellte Wert erreicht, wird das Zeitbit ausgeschaltet und der aktuelle Wert wird nicht weiter erhöht. Wird der Eingang jedoch wieder eingeschaltet, bevor die Zeit TOF den voreingestellten Wert erreicht, bleibt das Zeitbit eingeschaltet.
- ❏ Am Freigabeeingang muss eine fallende Flanke anliegen, damit die Zeit TOF Zeitintervalle zu zählen beginnt.
- ❏ Befindet sich die Zeit TOF in einem SCR-Bereich, der nicht aktiv ist, wird der aktuelle Wert auf 0 gesetzt, das Zeitbit ausgeschaltet und der aktuelle Wert zählt nicht.

Tipp

Sie können eine Zeit TONR nur mit der Operation Rücksetzen (R) zurücksetzen. Sie können mit der Operation Rücksetzen auch die Zeiten TON und TOF zurücksetzen. Die Operation Rücksetzen führt folgendes aus:

- Zeitbit = aus
- Aktueller Wert der Zeit = 0

Nach dem Rücksetzen einer Zeit TOF, muss der Freigabeeingang von EIN nach AUS wechseln, damit die Zeit neu gestartet werden kann.

Festlegen der Auflösung einer Zeit

Zeiten zählen Zeitintervalle. Die Auflösung (bzw. die Zeitbasis) der Zeit legt die Zeitspanne pro Intervall fest. Bei einer Zeit TON mit einer Auflösung von 10 ms beispielsweise zählt die Zeit die Anzahl der Intervalle von 10 ms, die ablaufen, nachdem die Zeit TON aktiviert wurde: bei einem Zählwert von 50 und der Auflösung von 10 ms sind dies 500 ms. Die SIMATIC Zeiten stehen mit drei verschiedenen Auflösungen zur Verfügung: 1 ms, 10 ms und 100 ms. Die Nummer der Zeit legt die Auflösung fest (siehe Tabelle 6-72).

Tipp

Um ein Mindestzeitintervall zu gewährleisten, erhöhen Sie den voreingestellten Wert (PV) um 1. Beispiel: Um ein Mindestzeitintervall von 2100 ms für eine Zeit mit einer Auflösung von 100 ms zu gewährleisten, setzen Sie den voreingestellten Wert (PV) auf 22.

Tabelle 6-72 Nummern der Zeit und Auflösungen

Zeit	Auflösung	Höchstwert		Nummer der Zeit
TONR (remanent)	1 ms	32,767 s	(0,546 Min.)	T0, T64
	10 ms	327,67 s	(5,46 Min.)	T1 bis T4, T65 bis T68
	100 ms	3276,7 s	(54,6 Min.)	T5 bis T31, T69 bis T95
TON, TOF (nicht remanent)	1 ms	32,767 s	(0,546 Min.)	T32, T96
	10 ms	327,67 s	(5,46 Min.)	T33 bis T36, T97 bis T100
	100 ms	3276,7 s	(54,6 Min.)	T37 bis T63, T101 bis T255

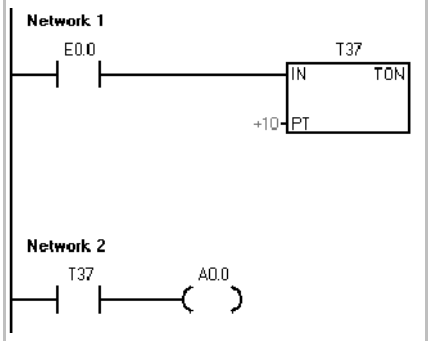
Auswirkung der Auflösung auf den Betrieb der Zeit

Bei einer Zeit mit einer Auflösung von 1 ms werden das Zeitbit und der aktuelle Wert asynchron zum Zyklus aktualisiert. In Zyklen von über 1 ms werden das Zeitbit und der aktuelle Wert mehrfach während des Zyklus aktualisiert.

Bei einer Zeit mit einer Auflösung von 10 ms werden das Zeitbit und der aktuelle Wert zu Beginn eines jeden Zyklus aktualisiert. Das Zeitbit und der aktuelle Wert bleiben während des Zyklus konstant und die Zeitintervalle, die während des Zyklus gezählt werden, werden zu Beginn eines jeden Zyklus zum aktuellen Wert addiert.

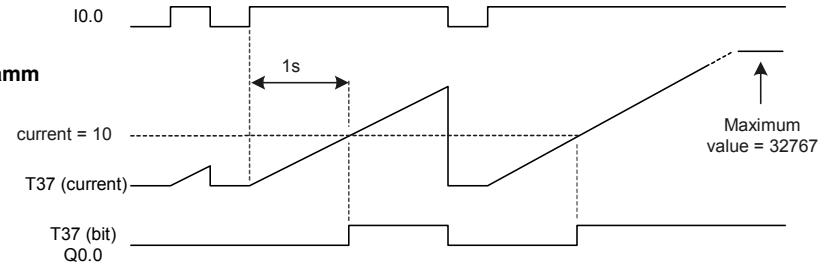
Bei einer Zeit mit einer Auflösung von 100 ms werden das Zeitbit und der aktuelle Wert aktualisiert, wenn die Operation ausgeführt wird. Deshalb müssen Sie darauf achten, dass Ihr Programm die Operation für eine 100-ms-Zeit nur einmal pro Zyklus ausführt, damit die Zeit den richtigen Zeitwert speichert.

Beispiel: SIMATIC - Zeit als Einschaltverzögerung starten




```
Netzwerk 1      //100-ms-Zeit T37 läuft nach (10 x 100 ms = 1s) ab
                //E0.0 EIN = T37 aktivieren, E0.0 AUS = T37 deaktivieren
                //und rücksetzen
LD      E0.0
TON     T37, +10

Netzwerk 2      //Das Bit T37 wird von der Zeit T37 gesteuert.
LD      T37
=       A0.0
```

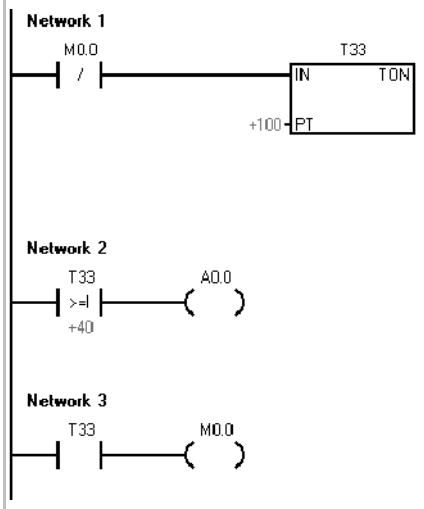
Impulsdiagramm



Tipp

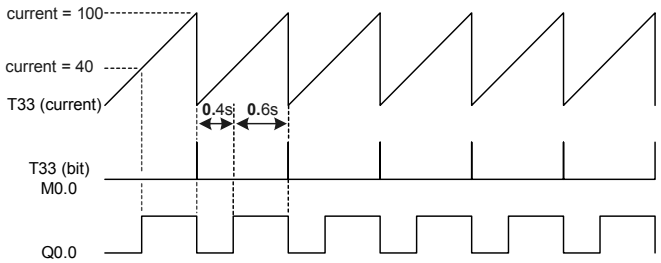
Um sicherzustellen, dass der Ausgang einer Zeit, die sich selbst zurücksetzt, jedesmal einen Zyklus lang eingeschaltet wird, wenn die Zeit den voreingestellten Wert erreicht, müssen Sie für den Freigabeeingang der Zeit statt eines Zeitbit einen Öffnerkontakt verwenden.

Beispiel: SIMATIC - Zeit als Einschaltverzögerung, die sich selbst zurücksetzt

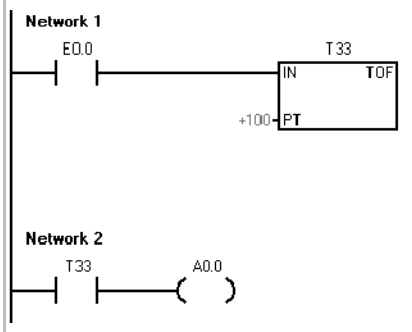



```
Netzwerk 1    //10-ms-Zeit T33 läuft nach (100 x 10 ms = 1 s) ab
              //Impuls M0.0 ist zu schnell für die Beobachtung in der
              //Statusansicht.
LDN     M0.0
TON     T33, +100

Netzwerk 2    //Der Vergleich wird mit einer Geschwindigkeit wahr, die
              //in der Statusansicht sichtbar ist. A0.0 nach (40 x 10 ms)
              //einschalten, um eine Wellenform von 40 % aus/60 % ein
              //zu erhalten.
LDW>=   T33, +40
=       A0.0

Netzwerk 3    //Impuls T33 (Bit) ist zu schnell für die Beobachtung in der
              //Statusansicht.
              //Zeit mit M0.0 nach Zeitraum von (100 x 10 ms = s)
              //zurücksetzen.
LD      T33
=       M0.0
```

Impulsdiagramm

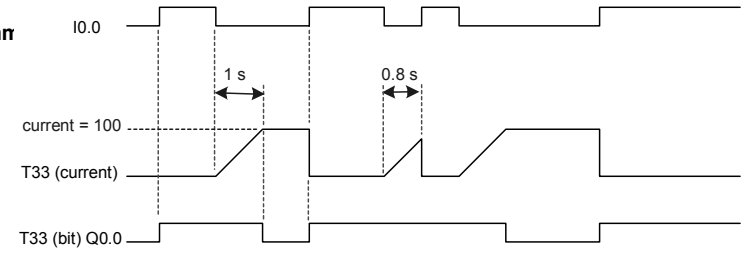


Beispiel: SIMATIC - Zeit als Ausschaltverzögerung starten

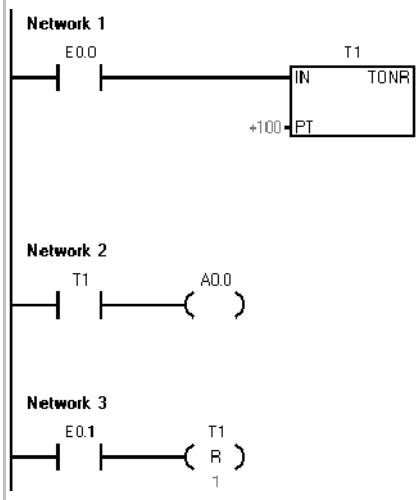



```
Netzwerk 1    //10-ms-Zeit T33 läuft nach (100 x 10 ms = 1s) ab
              //E0.0 EIN-AUS = T33 aktivieren
              //E0.0 AUS-EIN = T33 deaktivieren und rücksetzen
LD      E0.0
TOF     T33, +100

Netzwerk 2    //Die Zeit T33 steuert A0.0 über den Zeitkontakt T33.
LD      T33
=       A0.0
```

Impulsdiagramm

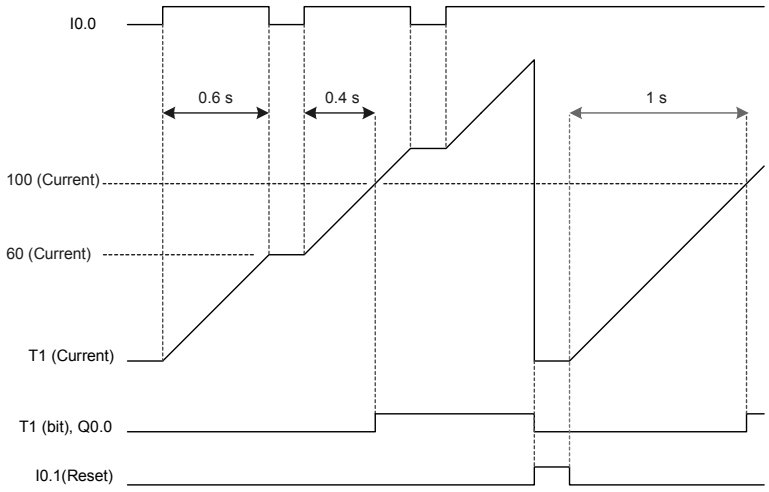


Beispiel: SIMATIC - Zeit als speichernde Einschaltverzögerung starten




```
Netzwerk 1      //Die 10-ms-Zeit TONR läuft ab bei PT = (100 x 10 ms = 1
                //s).
LD      E0.0
TONR    T1, +100

Netzwerk 2      //Das Bit T1 wird von der Zeit T1 gesteuert.
                //A0.0 einschalten, wenn die Zeit insgesamt
                //1 s gelaufen ist.
LD      T1
=       A0.0

Netzwerk 3      //Die Zeiten TONR müssen durch eine Operation Rückset
                //zen
                //mit einer Adresse T zurückgesetzt werden.
                //Zeit T1 (aktueller Wert und Bit) zurücksetzen, wenn E0.1
                //eingeschaltet ist.
LD      E0.1
R       T1, 1
```

Impulsdiagramm



IEC: Zeiten

Einschaltverzögerung

Die Operation Zeit als Einschaltverzögerung starten (TON) zählt die Zeit, wenn der Freigabeeingang eingeschaltet ist.

Ausschaltverzögerung

Die Operation Zeit als Ausschaltverzögerung starten (TOF) verzögert das Ausschalten eines Ausgangs für einen bestimmten Zeitraum, nachdem der Eingang ausgeschaltet wurde.

Impuls

Die Zeit Impuls (TP) erzeugt während eines bestimmten Zeitraums Impulse.

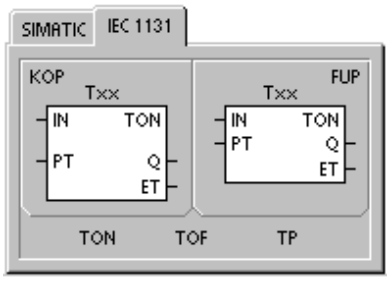


Tabelle 6-73 Gültige Operanden für die IEC Zeiten

Eingänge/ Ausgänge	Datentypen	Operanden
Txx	TON, TOF, TP	Konstante (T32 bis T63, T96 bis T255)
IN	BOOL	E, A, V, M, SM, S, T, Z, L, Signalfluss
PT	INT	EW, AW, VW, MW, SMW, SW, LW, AC, AEW, *VD, *LD, *AC, Konstante
A	BOOL	E, A, V, M, SM, S, L
ET	INT	EW, AW, VW, MW, SMW, SW, LW, AC, AAW, *VD, *LD, *AC

Tipp
Sie dürfen den Zeiten TOF, TON und TP nicht die gleiche Nummer zuweisen. Sie dürfen beispielsweise nicht eine Zeit TON T32 und eine Zeit TOF T32 gleichzeitig vergeben.

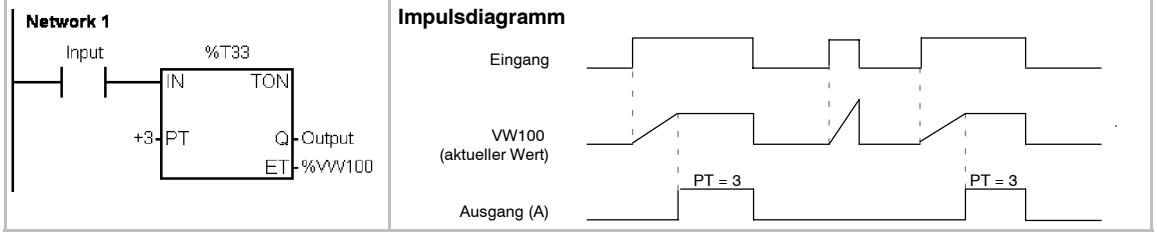
- Die Operation TON zählt die Zeitintervalle bis zum voreingestellten Wert, wenn der Freigabeeingang (IN) eingeschaltet wird. Ist die abgelaufene Zeit (ET) gleich der voreingestellten Zeit (PT), dann wird das Ausgangsbit (Q) der Zeit eingeschaltet. Das Ausgangsbit wird zurückgesetzt, wenn der Freigabeeingang ausgeschaltet wird. Ist die voreingestellte Zeit erreicht, stoppt die Zeit und wird deaktiviert.
- Die Operation TOF verzögert das Ausschalten eines Ausgangs für einen bestimmten Zeitraum, nachdem der Eingang ausgeschaltet wurde. Die Zeit zählt bis zum voreingestellten Wert, wenn der Freigabeeingang (IN) ausgeschaltet wird. Ist die abgelaufene Zeit (ET) gleich der voreingestellten Zeit (PT), dann wird das Ausgangsbit (Q) der Zeit ausgeschaltet. Ist die Voreinstellung erreicht, wird das Ausgangsbit der Zeit ausgeschaltet und die abgelaufene Zeit solange gespeichert, bis der Freigabeeingang (IN) wieder eingeschaltet wird. Wird der Freigabeeingang (IN) für einen Zeitraum ausgeschaltet, der kürzer ist als die voreingestellte Zeit (PT), bleibt das Ausgangsbit eingeschaltet.
- Die Operation TP erzeugt während eines bestimmten Zeitraums Impulse. Wird der Freigabeeingang (IN) eingeschaltet, wird das Ausgangsbit (Q) eingeschaltet. Das Ausgangsbit bleibt für den in der voreingestellten Zeit (PT) angegebenen Impuls eingeschaltet. Erreicht die abgelaufene Zeit (ET) die Voreinstellung (PT), wird das Ausgangsbit (Q) ausgeschaltet. Die abgelaufene Zeit bleibt solange eingeschaltet, bis der Freigabeeingang ausgeschaltet wird. Wird das Ausgangsbit eingeschaltet, bleibt es solange eingeschaltet, bis der Impuls abgelaufen ist.

Jede Erhöhung um 1 des aktuellen Werts 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. Die IEC Zeiten (TON, TOF und TP) stehen mit drei verschiedenen Auflösungen zur Verfügung. Die Auflösung richtet sich nach der Nummer der Zeit (siehe Tabelle 6-74).

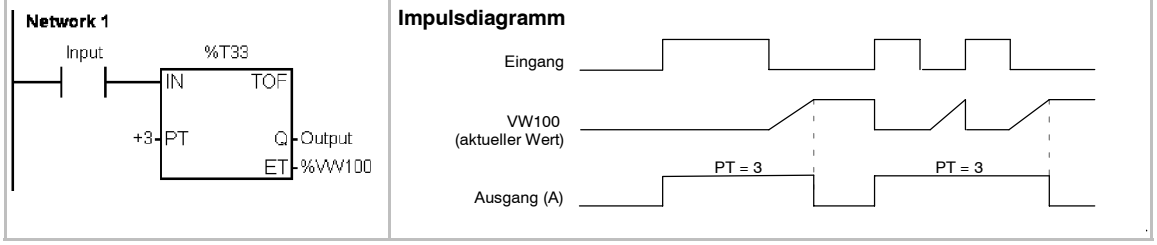
Tabelle 6-74 Auflösung der IEC-Zeiten

Auflösung	Höchstwert		Nummer der Zeit
1 ms	32,767 s	(0,546 Min.)	T32, T96
10 ms	327,67 s	(5,46 Min.)	T33 bis T36, T97 bis T100
100 ms	3276,7 s	(54,6 Min.)	T37 bis T63, T101 bis T255

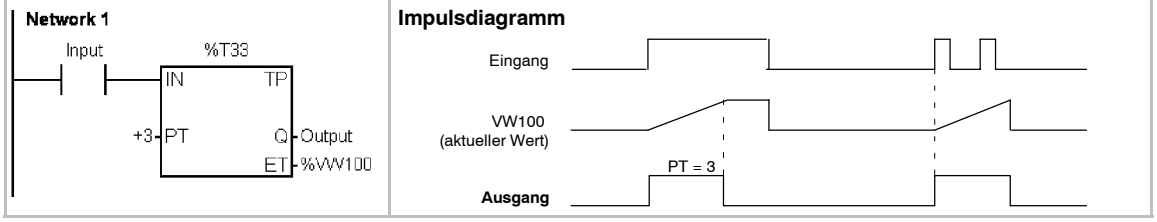
Beispiel: IEC - Zeit als Einschaltverzögerung starten



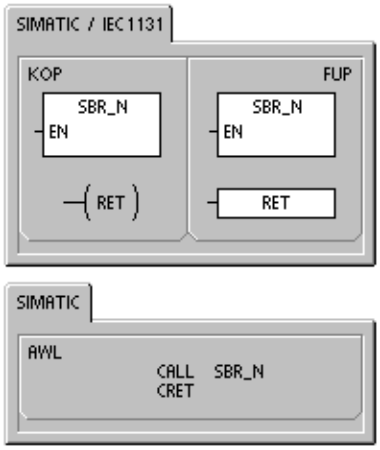
Beispiel: IEC - Zeit als Ausschaltverzögerung starten



Beispiel: IEC - Impuls



Unterprogrammoperationen



Die Operation Unterprogramm aufrufen (CALL) ruft ein Unterprogramm SBR_N auf. Sie können die Operation CALL mit oder ohne Parameter verwenden. Nachdem die Bearbeitung eines Unterprogramms beendet ist, wird das Hauptprogramm an der Operation weiterbearbeitet, die auf die Operation CALL folgt.

Die Operation Interruptprogramm bedingt beenden (CRET) beendet ein Unterprogramm in Abhängigkeit von dem Zustand der vorherigen Verknüpfung.

Zum Einfügen eines Unterprogramms wählen Sie den Menübefehl **Bearbeiten > Einfügen > Unterprogramm**.

Fehlerbedingungen, die ENO = 0 setzen

- 0008 (Maximale Schachtelungstiefe für Unterprogramme überschritten)
- 0006 (Indirekte Adresse)

Im Hauptprogramm können Sie maximal acht Unterprogramme ineinander verschachteln (den Aufruf eines Unterprogramms in einem Unterprogramm anordnen). In einem Interruptprogramm dürfen Sie keine Unterprogramme verschachteln.

Sie dürfen in einem Unterprogramm, das von einem Interruptprogramm aufgerufen wird, keinen Aufruf eines Unterprogramms anordnen. Rekursion (ein Unterprogramm ruft sich selbst auf) ist zulässig, doch sollten Sie Rekursion in Unterprogrammen nur mit Vorsicht einsetzen.

Tabelle 6-75 Gültige Operanden für die Unterprogrammoperation

Eingänge/ Ausgänge	Datentypen	Operanden
SBR_n	WORD	Konstante *CPU 221, CPU 222, CPU 224 und CPU 226*: 0 bis 63 *CPU 226XM:* 0 bis 127
IN	BOOL	V, E, A, M, SM, S, T, Z, L, Signalfluss
	BYTE	VB, EB, AB, MB, SMB, SB, LB, AC, *VD, *LD, *AC[1], Konstante
	WORD, INT	VW, T, Z, EW, AW, MW, SMW, SW, LW, AC, AEW, *VD, *LD, *AC[1], Konstante
	DWORD, DINT	VD, ED, AD, MD, SMD, SD, LD, AC, HC, *VD, *LD, *AC[1], &VB, &EB, &AB, &MB, &T, &Z, &SB, Konstante
In/Out	BOOL	V, E, A, M, SM[2], S, T, Z, L
	BYTE	VB, EB, AB, MB, SMB[2], SB, LB, AC, *VD, *LD, *AC[1]
	WORD, INT	VW, T, Z, EW, AW, MW, SMW[2], SW, LW, AC, *VD, *LD, *AC[1]
	DWORD, DINT	VD, ED, AD, MD, SMD[2], SD, LD, AC, *VD, *LD, *AC[1]
OUT	BOOL	V, E, A, M, SM[2], S, T, Z, L
	BYTE	VB, EB, AB, MB, SMB[2], SB, LB, AC, *VD, *LD, *AC[1]
	WORD, INT	VW, T, Z, EW, AW, MW, SMW[2], SW, LW, AC, AAW, *VD, *LD, *AC[1]
	DWORD, DINT	VD, ED, AD, MD, SMD[2], SD, LD, AC, *VD, *LD, *AC[1]

1 Muss mindestens Versatz 1 sein.
2 Muss mindestens Versatz 30 sein.

Tipp
STEP 7-Micro/WIN fügt am Ende eines Unterprogramms automatisch die Operation Unterprogramm beenden ein.

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.

Akkumulatoren werden von Unterprogrammen und dem aufrufenden Programm gemeinsam genutzt. Der Aufruf eines Unterprogramms bewirkt nicht, dass die Akkumulatoren gespeichert oder wiederhergestellt werden.

Aufrufen eines Unterprogramms mit Parameterübergabe

Unterprogramme können übergebene Parameter enthalten. Die Parameter werden in der lokalen Variablentabelle des Unterprogramms definiert. Parameter benötigen einen symbolischen Namen (maximal 23 Zeichen), einen Variablentyp und einen Datentyp. Sechzehn Parameter können an ein oder von einem Unterprogramm übergeben werden.

Die Spalte ”Variablentyp” in der lokalen Variablentabelle definiert, ob die Variable an das Unterprogramm übergeben wird (IN), ob es sich um eine Durchgangsvariable (IN_OUT) handelt oder ob die Variable vom Unterprogramm übergeben wird (OUT). Tabelle 6-76 führt die Parametertypen für Unterprogramme auf. Zum Ergänzen eines Parametereintrags positionieren Sie den Cursor auf dem Feld des Variablentyps (IN, IN_OUT oder OUT), den Sie definieren möchten. Rufen Sie mit der rechten Maustaste das Popup-Menü auf. Wählen Sie den Befehl ”Einfügen” und anschließend ”Reihe unterhalb”. Daraufhin wird ein neuer Parametereintrag des gewählten Variablentyps unterhalb des aktuellen Eintrags angezeigt.

Tabelle 6-76 Parametertypen für Unterprogramme

Parameter	Beschreibung
IN	Parameter werden an das Unterprogramm übergeben. Handelt es sich bei dem Parameter um eine direkte Adresse (z.B. VB10), wird der Wert der angegebenen Adresse an das Unterprogramm übergeben. Handelt es sich bei dem Parameter um eine indirekte Adresse (z.B. *AC1), wird der Wert, auf den die Adresse zeigt, an das Unterprogramm übergeben. Handelt es sich bei dem Parameter um eine Datenkonstante (16#1234) oder eine Adresse (&VB100), wird der Wert der Konstanten bzw. der Adresse an das Unterprogramm übergeben.
IN_OUT	Der Wert der angegebenen Parameteradresse wird an das Unterprogramm übergeben und der resultierende Wert des Unterprogramms wird an die gleiche Adresse zurückgegeben. Konstanten (z.B. 16#1234) und Adressen (z.B. &VB100) sind für Durchgangsparameter nicht zulässig.
OUT	Der resultierende Wert des Unterprogramms wird an die angegebene Parameteradresse ausgegeben. Konstanten (z.B. 16#1234) und Adressen (z.B. &VB100) sind als Ausgangsparameter nicht zulässig.
TEMP	Lokaler Speicher, der nicht für Übergabeparameter benötigt wird, kann für temporären Speicher innerhalb des Unterprogramms eingesetzt werden.

Das Feld Datentyp in der lokalen Variablentabelle definiert die Größe und das Format der Parameter (siehe Bild 6-38). Die Parametertypen sind im folgenden aufgeführt.

- ❏ BOOL: Dieser Datentyp (boolesch) wird für Ein- und Ausgänge mit einem Bit verwendet. Im folgenden Beispiel ist IN3 ein boolescher Eingang.
- ❏ BYTE, WORD, DWORD: Diese Datentypen kennzeichnen einen vorzeichenlosen Eingangs- oder Ausgangsparameter mit 1, 2 oder 4 Bytes.
- ❏ INT, DINT: Diese Datentypen kennzeichnen Eingangs- oder Ausgangsparameter mit Vorzeichen mit 1, 2 oder 4 Bytes.

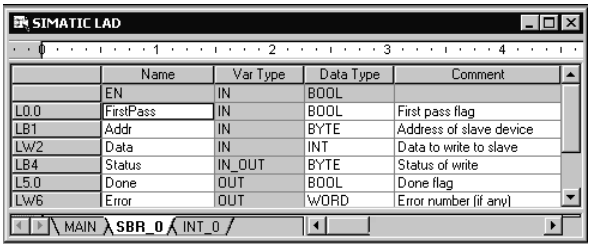


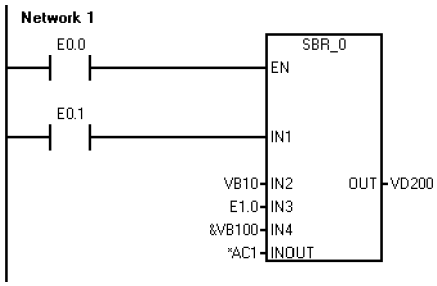
Bild 6-38 Lokale Variablentabelle

- ❏ REAL: Dieser Datentyp kennzeichnet einen einfachgenauen (4 Byte) IEEE-Gleitpunktwert.
- ❏ Signalfluss: Boolescher Signalfluss ist nur für boolesche Eingänge (Bitwerte) zulässig. Diese Deklaration teilt STEP 7-Micro/WIN mit, dass dieser Eingangsparameter das Ergebnis des Signalflusses von Verknüpfungsoperationen ist. Boolesche Signalflusseingänge müssen vor allen anderen Einträgen in der lokalen Variablentabelle angeordnet werden. Nur Eingangsparameter dürfen auf diese Weise verwendet werden. Der Freigabeeingang (EN) und der Eingang IN1 im folgenden Beispiel arbeiten mit boolescher Logik.

Beispiel: Operation Unterprogramm aufrufen

Es gibt zwei Beispiele in AWL. Der erste Satz AWL-Operationen kann nur im AWL-Editor angezeigt werden, weil die BOOL-Parameter für die Signalflusseingänge nicht im Lokaldatenspeicher abgelegt sind.

Der zweite Satz AWL-Operationen kann auch im KOP- und FUP-Editor angezeigt werden, weil im Lokaldatenspeicher der
Zustand der BOOL-Eingangsparameter abgelegt wird, die in KOP und FUP als Signalflusseingänge dargestellt werden.



Nur AWL:

Netzwerk 1

```
LD      E0.0
CALL    SBR_0, E0.1, VB10, E1.0, &VB100, *AC1, VD200
```

Für die richtige Darstellung in KOP und FUP:

Netzwerk 1

```
LD      E0.0
=       L60.0
LD      E0.1
=       L63.7
LD      L60.0
CALL    SBR_0, L63.7, VB10, E1.0, &VB100, *AC1, VD200
```

6

Adressparameter wie IN4 (&VB100) werden als DWORD (vorzeichenloser Doppelwortwert) an ein Unterprogramm übergeben. Der Type einer Konstante muss für den Parameter in dem aufrufenden Programm durch einen Konstanten-Bezeichner vor dem Wert der Konstanten angegeben werden. Wenn beispielsweise eine vorzeichenlose Doppelwortkonstante mit einem Wert von 12.345 als Parameter übergeben werden soll, muss der Parameter folgendermaßen angegeben werden: DW#12345. Ohne den Konstanten-Bezeichner kann die Konstante auch für einen anderen Datentyp gehalten werden.

Der Datentyp der Eingangs- und Ausgangsparameter wird nicht automatisch umgewandelt. Gibt die lokale Variablentabelle beispielsweise an, dass ein Parameter vom Datentyp REAL ist, während in dem aufrufenden Programm für diesen Parameter der Datentyp DWORD (Doppelwort) angegeben wird, ist der Wert im Unterprogramm ein Doppelwort.

Werden Werte an ein Unterprogramm übergeben, werden sie im lokalen Speicher des Unterprogramms abgelegt. Die Spalte ganz links in der Variablentabelle zeigt für jeden übergebenen Parameter die lokale Adresse im Speicher an. Die Werte der Eingangsparameter werden in den lokalen Speicher des Unterprogramms kopiert, wenn das Unterprogramm aufgerufen wird. Die Werte der Ausgangsparameter werden aus dem Lokaldatenspeicher des Unterprogramms in die angegebenen Adressen der Ausgangsparameter kopiert, wenn die Ausführung des Unterprogramms beendet ist.

Größe und Datentyp der Datenelemente werden in der Codierung der Parameter wiedergegeben. Die Parameterwerte werden dem lokalen Speicher der Unterprogramme folgendermaßen zugewiesen:

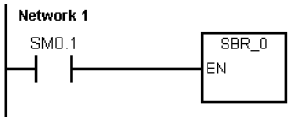
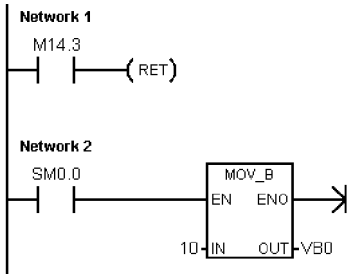
- Die Parameterwerte werden dem lokalen Speicher in der Reihenfolge zugewiesen, die von der Operation Unterprogramm mit Parametern aufrufen angegeben wird, und zwar mit Beginn bei L.0.
- Zwischen einem und acht aufeinanderfolgende Bit-Parameterwerte werden einem Byte mit Beginn bei Lx.0 bis Lx.7 zugordnet.
- Byte-, Wort- und Doppelwortwerte werden dem Lokaldatenbereich auf Bytegrenzen (LBx, LWx oder LDx) zugewiesen.

In der Operation Unterprogramm mit Parameter aufrufen müssen die Parameter in folgender Reihenfolge angeordnet werden: zuerst Eingangsparameter, dann Durchgangsparameter, zum Schluss Ausgangsparameter.

Wenn Sie in AWL programmieren, hat die Operation CALL folgendes Format:

CALL Nummer des Unterprogramms, Parameter 1, Parameter 2, ... , Parameter.

Beispiel: Operationen Unterprogramm aufrufen und Unterprogramm beenden

MAIN		Netzwerk 1 //Im ersten Zyklus, Unterprogramm 0 für die Initialisierung //aufrufen. LD SM0.1 CALL SBR0
SBR0		Netzwerk 1 //Mit einem bedingten Ende können Sie //das Unterprogramm vor dem letzten Netzwerk verlassen. LD M14.3 CRET Netzwerk 2 //Dieses Netzwerk wird übersprungen, wenn M14.3 //eingeschaltet ist. LD SM0.0 MOVB 10, VB0

6

Kommunikation im Netz

7

Die S7-200 löst Ihre Kommunikations- und Vernetzungsanforderungen durch Unterstützung einfacher und komplexer Netze. Die S7-200 bietet außerdem Werkzeuge für die Kommunikation mit anderen Geräten, z.B. mit Druckern und Waagen, die über eigene Kommunikationsprotokolle verfügen.

Mit STEP 7-Micro/WIN ist das Einrichten und Konfigurieren Ihres Netzes ein Kinderspiel.

In diesem Kapitel

Grundlagen der S7-200 Kommunikation im Netz 210
Einstellen des Kommunikationsprotokolls für Ihr Netz 214
Installieren und Deinstallieren von Kommunikationsschnittstellen 220
Aufbauen des Netzes 222
Erstellen von anwenderdefinierten Protokollen in der frei programmierbaren Kommunikation 227
Modems und STEP 7-Micro/WIN im Netz 229
Für erfahrene Anwender 235
Konfigurieren des RS-232/PPI-Multi-Master-Kabels für den entfernten Betrieb 241

Grundlagen der S7-200 Kommunikation im Netz

Einstellen der Kommunikationsschnittstelle für Ihr Netz

Die S7-200 unterstützt viele verschiedene Arten von Kommunikationsnetzen. Das Netz stellen Sie im Dialogfeld ”PG/PC-Schnittstelle einstellen” ein. Ein eingestelltes Netz wird als Schnittstelle bezeichnet. Es gibt folgende Arten von Schnittstellen für den Zugriff auf diese Kommunikationsnetze:

- ❑ PPI-Multi-Master-Kabel
- ❑ CP-Kommunikationskarten
- ❑ Ethernet-Kommunikationskarten

Zum Einstellen der Kommunikationsschnittstelle für STEP 7-Micro/WIN gehen Sie folgendermaßen vor (siehe Bild 7-1).

1. Doppelklicken Sie auf das Symbol im Dialogfeld ”Kommunikation”.
2. Wählen Sie den Schnittstellenparameter für STEP 7-Micro/WIN.

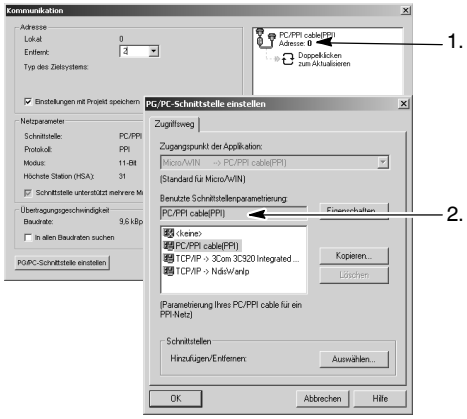


Bild 7-1 STEP 7-Micro/WIN Kommunikationsschnittstelle

PPI-Multi-Master-Kabel

Die S7-200 unterstützt die Kommunikation über zwei verschiedene Arten von PPI-Multi-Master-Kabeln. Diese Kabel ermöglichen die Kommunikation über eine RS-232-Schnittstelle oder eine USB-Schnittstelle.

Wie Sie in Bild 7-2 sehen, ist es ganz einfach, die Art des PPI-Multi-Master-Kabels einzustellen. Sie gehen folgendermaßen vor:

1. Wählen Sie im Dialogfeld ”PG/PC-Schnittstelle einstellen” die Schaltfläche “Eigenschaften”.
2. Öffnen Sie im Dialogfeld ”Eigenschaften” das Register ”Lokale Verbindung”.
3. Wählen Sie den USB-Port oder den gewünschten COM-Port .

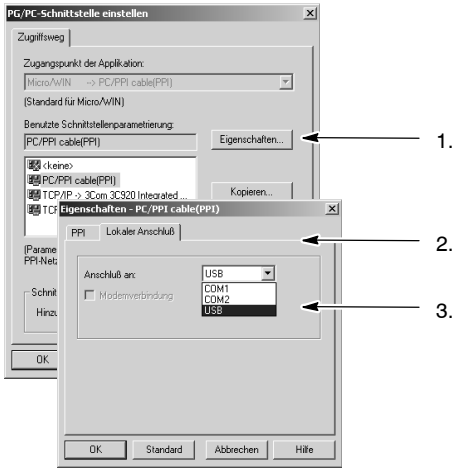


Bild 7-2 Auswahl des PPI-Multi-Master-Kabels

Tipp

Beachten Sie, dass nur ein USB-Kabel zur Zeit verwendet werden kann.

Tipp

In den Beispielen in diesem Handbuch wird das RS-232/PPI-Multi-Master-Kabel verwendet. Das RS-232/PPI-Multi-Master-Kabel ersetzt das frühere PC/PPI-Kabel. Außerdem ist ein USB/PPI-Multi-Master-Kabel erhältlich. Die Bestellnummern finden Sie im Anhang E.

Master- und Slave-Geräte in einem PROFIBUS-Netz

Die S7-200 unterstützt ein Master/Slave-Netz und kann sowohl als Master als auch als Slave im PROFIBUS-Netz eingesetzt werden, während STEP 7-Micro/WIN immer Master ist.

Master

Ein Gerät, bei dem es sich um einen Master im Netz handelt, kann eine Anforderung an ein anderes Gerät im Netz schicken. Ein Master kann auch auf Anforderungen anderer Master im Netz reagieren. Typische Master sind z.B. STEP 7-Micro/WIN, HMI-Geräte wie das TD 200 und die S7-300 oder S7-400 Automatisierungssysteme. Die S7-200 funktioniert als Master, wenn sie von anderen S7-200 Geräten Informationen anfordert (Punkt-zu-Punkt-Kommunikation).

Tipp
Ein TP070 funktioniert nicht in einem Netz mit anderen Master-Geräten.

Slaves

Ein Gerät, das als Slave eingerichtet ist, kann nur auf Anforderungen eines Master-Geräts reagieren. Ein Slave kann keine Anforderungen senden. In den meisten Netzen wird die S7-200 als Slave eingesetzt. Als Slave reagiert die S7-200 auf Anforderungen von Master-Geräten im Netz, z.B. Operator Panels oder STEP 7-Micro/WIN.

Einstellen von Baudrate und Netzadresse

Die Geschwindigkeit, mit der Daten im Netz übertragen werden, ist die Baudrate. Die Baudrate wird üblicherweise in Kilobaud (kBaud) oder in Megabaud (MBaud) angegeben. Die Baudrate misst, wie viele Daten in einem bestimmten Zeitraum übertragen werden können. Eine Baudrate von 19,2 kBaud beispielsweise gibt eine Übertragungsgeschwindigkeit von 19.200 Bits pro Sekunde an.

Für alle Geräte, die in einem bestimmten Netz kommunizieren, muss die gleiche Baudrate für die Datenübertragung eingestellt werden. Deshalb wird die höchste Baudrate im Netz durch das langsamste, ans Netz angeschlossene Gerät festgelegt.

Tabelle 7-1 führt die von der S7-200 unterstützten Baudraten auf.

Tabelle 7-1 Von der S7-200 unterstützte Baudraten

Netz	Baudrate
Standardnetz	9,6 kBaud bis 187,5 kBaud
Bei Einsatz eines EM 277	9,6 kBaud bis 12 MBaud
Frei programmierbare Kommunikation	1200 Baud bis 115,2 kBaud

Die Netzadresse ist eine eindeutige Nummer, die Sie einem Gerät im Netz zuweisen. Die eindeutige Netzadresse stellt sicher, dass die Daten ans richtige Gerät übertragen oder von dort abgefragt werden. Die S7-200 unterstützt Netzadressen von 0 bis 126. Bei einer S7-200 mit zwei Schnittstellen hat jede Schnittstelle eine Netzadresse. Tabelle 7-2 führt die werkseitigen Voreinstellungen für die S7-200 Geräte auf.

Tabelle 7-2 Voreingestellte Adressen für die S7-200 Geräte

S7-200 Gerät	Voreingestellte Adresse
STEP 7-Micro/WIN	0
HMI (TD 200, TP oder OP)	1
S7-200 CPU	2

Einstellen von Baudrate und Netzadresse für STEP 7-Micro/WIN

Sie müssen die Baudrate und die Netzadresse für STEP 7-Micro/WIN einstellen. Die Baudrate muss die gleiche sein, wie die der anderen Geräte im Netz, und die Netzadresse muss eindeutig sein.

Üblicherweise wird die Netzadresse (0) für STEP 7-Micro/WIN nicht geändert. Wird in Ihrem Netz ein anderes Programmierpaket eingesetzt, dann müssen Sie evtl. die Netzadresse für STEP 7-Micro/WIN ändern.

Wie Sie in Bild 7-3 sehen, ist es ganz einfach, die Baudrate und die Netzadresse für STEP 7-Micro/WIN einzustellen. Nachdem Sie in der Navigationsleiste auf das Symbol "Kommunikation" geklickt haben, gehen Sie folgendermaßen vor:

1. Doppelklicken Sie auf das Symbol im Dialogfeld "Kommunikation".
2. Wählen Sie im Dialogfeld "PG/PC-Schnittstelle einstellen" die Schaltfläche "Eigenschaften".
3. Wählen Sie die Netzadresse für STEP 7-Micro/WIN.
4. Wählen Sie die Baudrate für STEP 7-Micro/WIN.

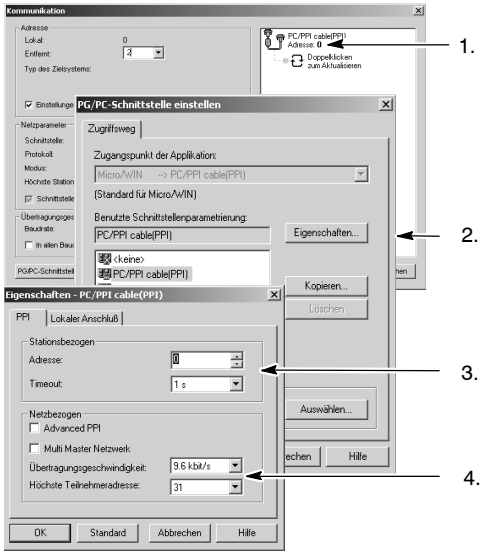


Bild 7-3 Einrichten von STEP 7-Micro/WIN

Einstellen von Baudrate und Netzadresse für die S7-200

Sie müssen auch die Baudrate und die Netzadresse für die S7-200 einstellen. Der Systemdatenbaustein der S7-200 speichert die Baudrate und die Netzadresse. Nachdem Sie die Parameter für die S7-200 ausgewählt haben, müssen Sie den Systemdatenbaustein in die S7-200 laden.

Die voreingestellte Baudrate für die Schnittstelle der S7-200 beträgt 9,6 kBaud, und die voreingestellte Netzadresse ist 2.

Wie Sie in Bild 7-4 sehen, stellen Sie die Baudrate und die Netzadresse für die S7-200 in STEP 7-Micro/WIN ein. Nachdem Sie in der Navigationsleiste auf das Symbol des Systemdatenbausteins geklickt oder den Menübefehl **Ansicht > Komponente > Systemdatenbaustein** gewählt haben, gehen Sie folgendermaßen vor:

1. Wählen Sie die Netzadresse für die S7-200.
2. Wählen Sie die Baudrate für die S7-200.
3. Laden Sie den Systemdatenbaustein in die S7-200.

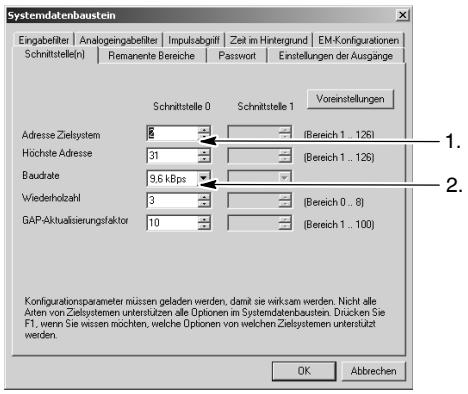


Bild 7-4 Konfigurieren der S7-200 CPU

Tipp

Sie können alle Optionen für die Baudrate wählen. STEP 7-Micro/WIN prüft diese Einstellung beim Laden des Systemdatenbausteins. Baudraten, bei denen STEP 7-Micro/WIN nicht mit der S7-200 kommunizieren kann, werden nicht geladen.

Einstellen der entfernten Adresse

Bevor Sie die aktualisierten Einstellungen in die S7-200 laden können, müssen Sie die Kommunikationsschnittstelle (COM) von STEP 7-Micro/WIN (lokal) und die Adresse der S7-200 (entfernt) so einstellen, dass beide den aktuellen Einstellungen der entfernten S7-200 entsprechen (siehe Bild 7-5).

Nachdem Sie die aktualisierten Einstellungen geladen haben, müssen Sie evtl. die Einstellung der Baudrate im Dialogfeld ”PG/PC-Schnittstelle einstellen” neu konfigurieren (sofern sich diese Einstellung von der Einstellung unterscheidet, die in die entfernte S7-200 geladen wurde). Wie Sie die Baudrate konfigurieren, sehen Sie in Bild 7-3.

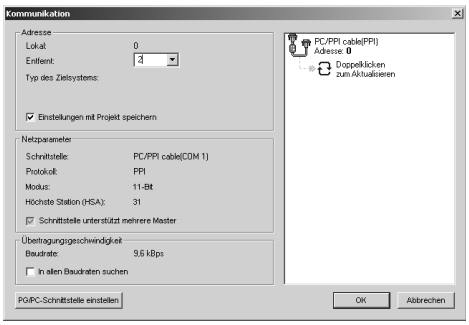


Bild 7-5 Einrichten von STEP 7-Micro/WIN

Suchen nach S7-200 CPUs im Netz

Sie können nach den S7-200 CPUs, die an Ihr Netz angeschlossen sind, suchen und sie identifizieren. Sie können im Netz auch bei einer bestimmten Baudrate oder bei allen Baudraten suchen, wenn Sie die S7-200 CPUs identifizieren möchten.

Nur mit PPI-Multi-Master-Kabeln können Sie bei allen Baudraten suchen. Diese Funktion steht bei der Kommunikation über eine CP-Karte nicht zur Verfügung. Die Suche startet bei der gerade eingestellten Baudrate.

1. Öffnen Sie das Dialogfeld ”Kommunikation” und doppelklicken Sie auf das Symbol ”Aktualisieren”, um die Suche zu starten.
2. Wenn Sie bei allen Baudraten suchen möchten, aktivieren Sie das Kontrollkästchen ”In allen Baudraten suchen”.

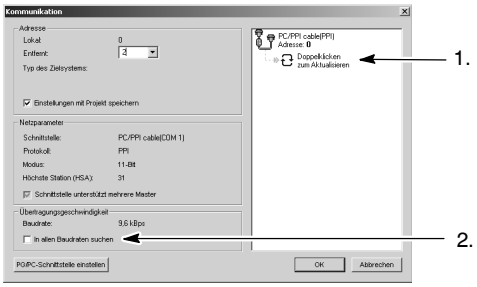


Bild 7-6 Suchen nach CPUs im Netz

Einstellen des Kommunikationsprotokolls für Ihr Netz

Im folgenden werden die Protokolle aufgeführt, die von den S7-200 CPUs unterstützt werden.

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

Die Protokolle basieren auf der Kommunikationsarchitektur des Sieben-Lagen-Modells für die Kommunikation offener Systeme (OSI). Die Protokolle werden in einem Token-Ring-Netz implementiert, das dem PROFIBUS-Standard 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 Stoppbit. 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 Protokolle können in einem Netz gleichzeitig aktiv sein, ohne dass sie sich gegenseitig beeinträchtigen, solange die Baudrate für alle Protokolle die gleiche ist.

Für die S7-200 CPU steht mit den Erweiterungsmodulen CP243-1 und CP243-1 IT Ethernet zur Verfügung.

PPI-Protokoll

Das PPI-Protokoll ist ein Master-/Slave-Protokoll: der Master sendet Anforderungen an den Slave, und der Slave reagiert (siehe Bild 7-7). Slaves initiieren keine Meldungen, sondern warten, bis ein Master eine Anforderung sendet oder die Slaves nach einer Reaktion abfragt.

Master kommunizieren mit Slaves über eine gemeinsam genutzte Verbindung, die vom PPI-Protokoll verwaltet wird. Das PPI-Protokoll begrenzt die Anzahl der Master, die mit den Slaves kommunizieren können, nicht. Sie können jedoch maximal 32 Master ins Netz aufnehmen.

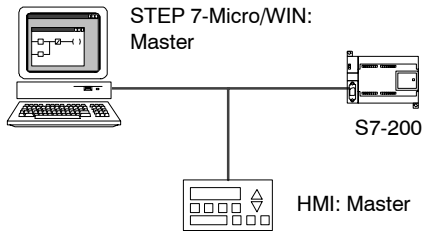


Bild 7-7 PPI-Netz

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 und In Netz schreiben Meldungen aus anderen S7-200 lesen bzw. in andere S7-200 schreiben. Auch wenn die S7-200 als PPI-Master eingesetzt wird, reagiert sie immer noch als Slave auf Anforderungen von anderen Mastern.

Durch PPI Advanced können die Geräte im Netz eine logische Verbindung zwischen den Geräten aufbauen. Bei PPI Advanced werden von jedem Gerät eine begrenzte Anzahl Verbindungen bereitgestellt. Tabelle 7-3 führt die von der S7-200 unterstützte Anzahl Verbindungen auf.

Alle S7-200 CPUs unterstützen die Protokolle PPI und PPI Advanced, während das Modul EM 277 nur das Protokoll PPI Advanced unterstützt.

Tabelle 7-3 Anzahl der Verbindungen für die S7-200 CPU und das EM 277

Modul		Baudrate	Verbindungen
S7-200 CPU	Schnittstelle 0	9,6 kBaud, 19,2 kBaud oder 187,5 kBaud	4
	Schnittstelle 1	9,6 kBaud, 19,2 kBaud oder 187,5 kBaud	4
EM 277		9,6 kBaud bis 12 MBaud	6 je Modul

MPI-Protokoll

Beim MPI-Protokoll ist sowohl die Kommunikation Master-Master als auch die Kommunikation Master-Slave möglich (siehe Bild 7-8). Für die Kommunikation mit einer S7-200 CPU baut STEP 7-Micro/WIN eine Master/Slave-Verbindung auf. Das MPI-Protokoll kommuniziert nicht mit einer S7-200 CPU, die als Master eingesetzt wird.

Geräte im Netz kommunizieren über getrennte Verbindungen (die vom MPI-Protokoll verwaltet werden) zwischen zwei beliebigen Geräten. Die Kommunikation zwischen den Geräten ist auf die Anzahl der Verbindungen begrenzt, die von der S7-200 CPU bzw. vom EM 277 unterstützt wird. Tabelle 7-3 führt die von der S7-200 unterstützte Anzahl Verbindungen auf.

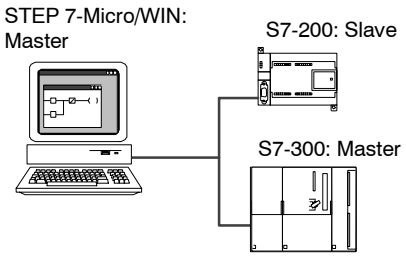


Bild 7-8 MPI-Netz

Beim MPI-Protokoll lesen und schreiben die Automatisierungssysteme S7-300 und S7-400 Daten mit den Operationen XGET und XPUT aus der und in die S7-200 CPU. Informationen zu diesen Operationen finden Sie im Programmierhandbuch der S7-300 bzw. der S7-400.

PROFIBUS-Protokoll

Das PROFIBUS-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-Netze haben üblicherweise einen Master und mehrere Slave-Ein-/Ausgabegeräte (siehe Bild 7-9). 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 fortlaufend Ausgabedaten in die Slaves und liest die Eingabedaten.

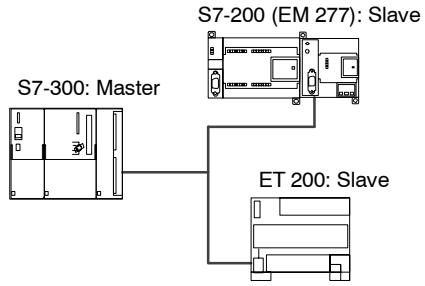


Bild 7-9 PROFIBUS-Netz

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.

TCP/IP-Protokoll

Die S7-200 kann die TCP/IP-Ethernet-Kommunikation durch den Einsatz eines der Erweiterungsmodule Ethernet (CP 243-1) oder Internet (CP 243-1 IT) unterstützen. Tabelle 7-4 zeigt die Baudrate und die Anzahl der Verbindungen, die von diesen Modulen unterstützt werden.

Tabelle 7-4 Anzahl der Verbindungen für die Module Ethernet (CP 243-1) und Internet (CP 243-1 IT)

Modul	Baudrate	Verbindungen
Ethernet-Modul (CP 243-1)	10 bis 100 MBaud	8 Allzweckverbindungen
Internet-Modul (CP 243-1 IT)		1 STEP 7-Micro/WIN-Verbindung

Weitere Informationen finden Sie in den Handbüchern *CP 243-1 Kommunikationsprozessor für Industrial Ethernet* und *CP 243-1 IT Kommunikationsprozessor für Industrial Ethernet und Informationstechnologie*.

Beispiele für Netzkonfigurationen nur mit S7-200 Geräten

PPI-Netze mit einem Master

Um ein einfaches Netz mit einem Master aufzubauen, werden das Programmiergerät und die S7-200 CPU entweder über ein PPI-Multi-Master-Kabel oder über eine CP-Karte (Kommunikationsprozessor), die im Programmiergerät gesteckt ist, miteinander verbunden.

Im Beispielnetz oben in Bild 7-10 ist das Programmiergerät (STEP 7-Micro/WIN) der Master im Netz. Im Beispielnetz unten in Bild 7-10 ist ein HMI-Gerät (z.B. ein TD 200, TP oder OP) der Master im Netz.

In beiden Beispielen ist die S7-200 CPU ein Slave, der auf Anforderungen vom Master reagiert.

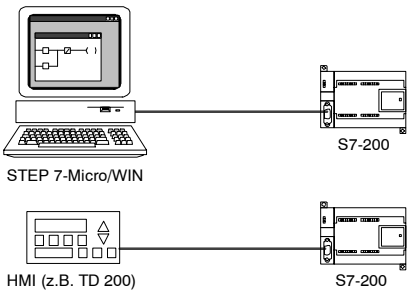


Bild 7-10 PPI-Netz mit einem Master

In einem PPI-Netz mit einem Master richten Sie STEP 7-Micro/WIN für das PPI-Protokoll ein: Deaktivieren Sie die Kontrollkästchen ”Multi-Master-Netz” und ”PPI Advanced”.

PPI-Netz mit mehreren Mastern

Bild 7-11 zeigt ein Beispiel für ein Netz mit mehreren Mastern und einem Slave. Das Programmiergerät (STEP 7-Micro/WIN) verwendet entweder eine CP-Karte oder ein PPI-Multi-Master-Kabel. STEP 7-Micro/WIN und das HMI-Gerät nutzen das Netz gemeinsam.

STEP 7-Micro/WIN und das HMI-Gerät sind beide Master und müssen getrennte Netzadressen besitzen. Wird das PPI-Multi-Master-Kabel verwendet, ist das Kabel ein Master und
es nutzt die von STEP 7-Micro/WIN zur Verfügung gestellte Netzadresse. Die S7-200 CPU ist ein Slave.

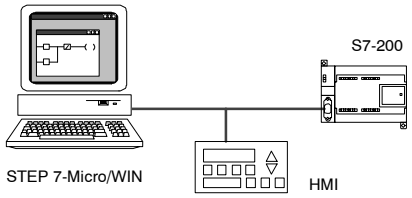


Bild 7-11 Mehrere Master mit einem Slave

Bild 7-12 zeigt ein PPI-Netz mit mehreren Mastern, die mit mehreren Slaves kommunizieren. In diesem Beispiel können STEP 7-Micro/WIN und das HMI-Gerät beide Daten von einem beliebigen S7-200 CPU-Slave anfordern. STEP 7-Micro/WIN und das HMI-Gerät nutzen das Netz gemeinsam.

Alle Geräte (Master und Slaves) haben verschiedene Netzadressen. Wird das PPI-Multi-Master-Kabel verwendet, ist das Kabel ein Master und es nutzt die von STEP 7-Micro/WIN zur Verfügung gestellte Netzadresse. Die S7-200 CPUs sind Slaves.

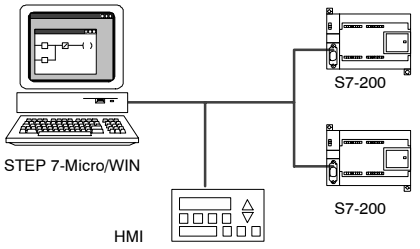


Bild 7-12 Mehrere Master und mehrere Slaves

In einem Netz mit mehreren Mastern und einem oder mehreren Slaves richten Sie STEP 7-Micro/WIN für die Verwendung des PPI-Protokolls ein und aktivieren die Kontrollkästchen ”Multi-Master-Netz” und ”PPI Advanced”. Bei Verwendung eines PPI-Multi-Master-Kabels werden die Kontrollkästchen ”Multi-Master-Netz” und ”PPI Advanced” ignoriert.

Komplexe PPI-Netze

Bild 7-13 zeigt ein Beispiel für ein Netz, das mehrere Master in der Punkt-zu-Punkt-Kommunikation nutzt.

STEP 7-Micro/WIN und das HMI-Gerät lesen und schreiben über das Netz aus den und die S7-200 CPUs und die S7-200 CPUs lesen und schreiben untereinander mit Hilfe der Operationen Aus Netz lesen und In Netz schreiben (Punkt-zu-Punkt-Kommunikation).

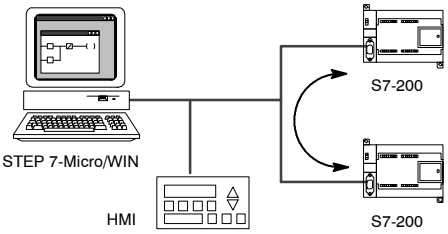


Bild 7-13 Punkt-zu-Punkt-Kommunikation

Bild 7-14 zeigt ein weiteres Beispiel für ein komplexes PPI-Netz mit mehreren Mastern und Punkt-zu-Punkt-Kommunikation. In diesem Beispiel beobachtet jedes HMI-Gerät eine S7-200 CPU.

Die S7-200 CPUs lesen und schreiben untereinander mit den Operationen NETR und NETW (Punkt-zu-Punkt-Kommunikation).

In komplexen PPI-Netzen richten Sie STEP 7-Micro/WIN für die Verwendung des PPI-Protokolls ein und aktivieren die Kontrollkästchen ”Multi-Master-Netz” und ”PPI Advanced”. Bei Verwendung eines PPI-Multi-Master-Kabels werden die Kontrollkästchen ”Multi-Master-Netz” und ”PPI Advanced” ignoriert.

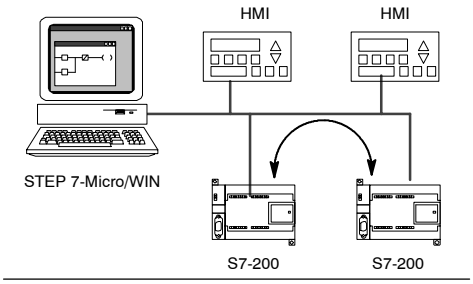


Bild 7-14 HMI-Geräte und Punkt-zu-Punkt-Kommunikation

Beispiele für Netzkonfigurationen mit S7-200, S7-300 und S7-400 Geräten

Netze mit Baudraten bis zu 187,5 kBaud

In dem Beispielnetz in Bild 7-15 nutzt die S7-300 die Operationen XPUT und XGET für die Kommunikation mit einer S7-200 CPU. Die S7-300 kann nicht mit einer S7-200 CPU im Mastermodus kommunizieren.

Für die Kommunikation mit S7-CPUs richten Sie STEP 7-Micro/WIN für die Verwendung des PPI-Protokolls ein und aktivieren die Kontrollkästchen ”Multi-Master-Netz” und ”PPI Advanced”. Bei Verwendung eines PPI-Multi-Master-Kabels werden die Kontrollkästchen ”Multi-Master-Netz” und ”PPI Advanced” ignoriert.

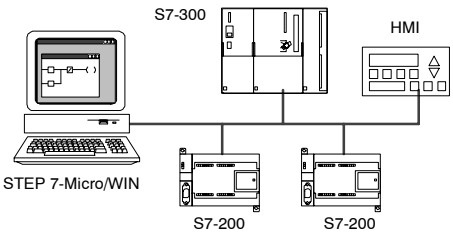


Bild 7-15 Baudraten bis zu 187,5 kBaud

Netze mit Baudraten über 187,5 kBaud

Bei Baudraten über 187,5 kBaud muss die S7-200 CPU über ein EM 277 ans Netz angeschlossen sein (siehe Bild 7-16). STEP 7-Micro/WIN muss über eine Kommunikationsprozessorkarte (CP) angeschlossen sein.

In dieser Konfiguration kann die S7-300 mit den S7-200 Geräten über die Operationen XPUT und XGET kommunizieren und das HMI-Gerät kann entweder die S7-200 Geräte oder die S7-300 beobachten.

Das EM 277 ist immer ein Slave-Gerät.

STEP 7-Micro/WIN kann alle S7-200 CPUs über das angeschlossene EM 277 programmieren und beobachten. Für die Kommunikation mit einem EM 277 über 187,5 kBaud richten Sie STEP 7-Micro/WIN für die Verwendung des MPI-Protokoll mit einer CP-Karte ein. Die maximale Baudrate für das PPI-Multi-Master-Kabel beträgt 187,5 kBaud.

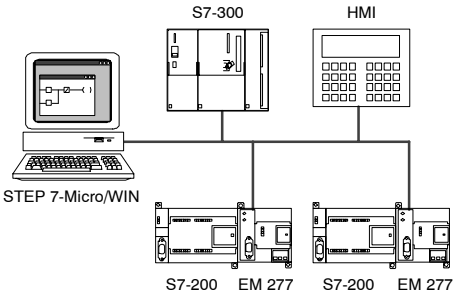


Bild 7-16 Baudraten über 187,5 kBaud

Beispiel für eine PROFIBUS-DP-Netzkonfiguration

Netze mit S7-315-2 DP als PROFIBUS-Master und EM 277 als PROFIBUS-Slave

Bild 7-17 zeigt ein Beispiel für ein PROFIBUS-Netz, in dem eine S7-315-2 DP als PROFIBUS-Master eingesetzt wird. Ein Erweiterungsmodul EM 277 ist ein PROFIBUS-Slave.

Die S7-315-2 DP kann Daten aus dem EM 277 lesen und Daten in das EM 277 schreiben, und zwar von 1 Byte bis zu 128 Bytes. Die S7-315-2 DP liest oder schreibt Adressen des Variablenspeichers in die S7-200.

Dieses Netz unterstützt Baudraten von 9600 Baud bis 12 MBaud.

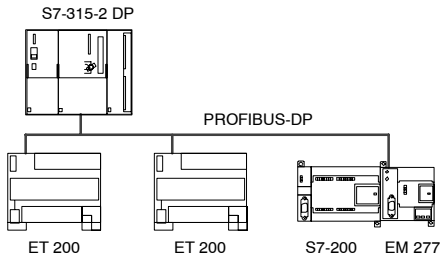


Bild 7-17 Netz mit einer S7-315-2 DP

Netze mit STEP 7-Micro/WIN und HMI-Gerät

Bild 7-18 zeigt ein Beispiel für ein Netz mit einer S7-315-2 DP als PROFIBUS-Master und mit einem EM 277 als PROFIBUS-Slave. In dieser Konfiguration beobachtet das HMI-Gerät die S7-200 über das EM 277. STEP 7-Micro/WIN programmiert die S7-200 über das EM 277.

Dieses Netz unterstützt Baudraten von 9600 Baud bis 12 MBaud. STEP 7-Micro/WIN benötigt für Baudraten über 187,5 kBaud eine CP-Karte.

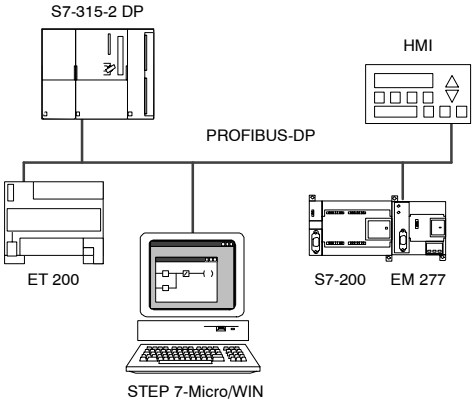


Bild 7-18 PROFIBUS-Netz

Richten Sie STEP 7-Micro/WIN für das PROFIBUS-Protokoll für eine CP-Karte ein. Sind im Netz nur DP-Geräte vorhanden, wählen Sie das DP- oder Standardprofil. Sind im Netz andere Geräte vorhanden, nicht nur DP-Geräte, z.B. ein TD 200, dann wählen Sie das Universalprofil (DP/FMS) für alle PROFIBUS-Master. Alle Master im Netz müssen das gleiche PROFIBUS-Profil (DP, Standard oder Universal) verwenden, damit das Netz funktionsfähig ist.

Die PPI-Multi-Master-Kabel funktionieren in Netzen mit maximal 187.5 kBaud nur, wenn alle Master-Geräte das Universalprofil (DP/FMS) nutzen.

Beispiele für Netzkonfigurationen mit Ethernet- und/oder Internet-Geräten

In der Konfiguration in Bild 7-19 ermöglicht eine Ethernet-Verbindung, dass STEP 7-Micro/WIN mit den S7-200 CPUs kommunizieren kann, die ein Ethernet-Modul (CP 243-1) oder ein Internet-Modul (CP 243-1 IT) nutzen. Die S7-200 CPUs können über die Ethernet-Verbindung Daten austauschen. Über einen Standard-Browser auf dem PC, auf dem STEP 7-Micro/WIN installiert ist, können Sie auf die Homepage des Internet-Moduls (CP 243-1 IT) zugreifen.

Bei Ethernet-Netzen richten Sie STEP 7-Micro/WIN für die Verwendung des TCP/IP-Protokolls ein.

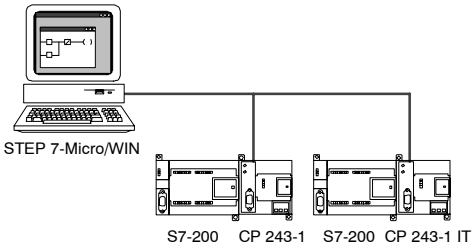


Bild 7-19 Ethernet-Netz mit 10/100 MBaud

Tipp

Im Dialogfeld ”PG/PC-Schnittstelle einstellen” gibt es zwei TCP/IP-Optionen. Die Option ”TCP/IP -> NdisWanIp” wird von der S7-200 nicht unterstützt.

- Im Dialogfeld ”PG/PC-Schnittstelle einstellen” richten sich die Optionen nach der Art der von Ihrem PC bereitgestellten Ethernet-Schnittstelle. Wählen Sie die Option, die Ihren Computer mit dem Ethernet-Netz verbindet, an das das Modul CP 243-1 bzw. CP 243-1 IT angeschlossen ist.
- Im Dialogfeld ”Kommunikation” müssen Sie die entfernte IP-Adresse der Ethernet-/Internet-Module eingeben, mit denen Sie kommunizieren möchten.

Installieren und Deinstallieren von Kommunikationsschnittstellen

Im Dialogfeld ”PG/PC-Schnittstelle einstellen” rufen Sie das Dialogfeld ”Schnittstellen installieren/deinstallieren” auf, um auf Ihrem PC Kommunikationsschnittstellen zu installieren oder zu deinstallieren.

1. Klicken Sie im Dialogfeld ”PG/PC-Schnittstelle einstellen” auf die Schaltfläche ”Auswählen”, um das Dialogfeld ”Schnittstellen installieren/deinstallieren” aufzurufen.

   Im Auswahlfeld werden die verfügbaren Schnittstellen aufgeführt. Im Feld ”Installiert” werden die Schnittstellen angezeigt, die bereits auf Ihrem Computer installiert sind.

2. *So ergänzen Sie eine Kommunikationsschnittstelle:* Wählen Sie die auf Ihrem PC installierte Kommunikationshardware und klicken Sie auf ”Installieren”. Wenn Sie das Dialogfeld ”Schnittstellen installieren/deinstallieren” schließen, zeigt das Dialogfeld ”PG/PC-Schnittstelle einstellen” die Schnittstelle im Feld ”Benutzte Schnittstellenparametrierung” an.

3. *So deinstallieren Sie eine Kommunikationsschnittstelle:* Wählen Sie die Schnittstelle, die Sie deinstallieren möchten, und klicken Sie auf ”Deinstallieren”. Wenn Sie das Dialogfeld ”Schnittstellen installieren/deinstallieren” schließen, zeigt das Dialogfeld ”PG/PC-Schnittstelle einstellen” die Schnittstelle nicht mehr im Feld ”Benutzte Schnittstellenparametrierung” an.

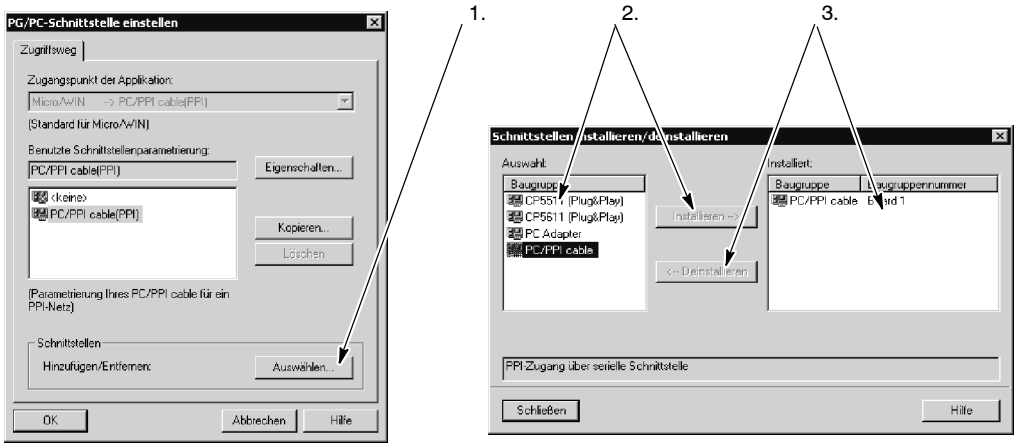


Bild 7-20 Dialogfelder ”PG/PC-Schnittstelle einstellen” und ”Schnittstellen installieren/deinstallieren”

Tipp

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 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. 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.

Ändern der Schnittstelleneinstellungen Ihres PC für den PPI-Multi-Master-Betrieb

Wenn Sie das USB/PPI-Multi-Master-Kabel oder das RS-232/PPI-Multi-Master-Kabel im PPI-Modus einsetzen, müssen Sie die Einstellungen für die Schnittstelle an Ihrem Computer nicht ändern, und der Betrieb in Multi-Master-Netzen ist unter dem Betriebssystem Windows NT möglich.

Wenn Sie das RS-232/PPI-Multi-Master-Kabel im Modus PPI/frei programmierbare Kommunikation für die Kommunikation zwischen einer S7-200 CPU und STEP 7-Micro/WIN in einem Betriebssystem einsetzen, das die PPI-Multi-Master-Konfiguration unterstützt (Windows NT unterstützt PPI-Multi-Master nicht), müssen Sie evtl. die Schnittstelleneinstellungen an Ihrem Computer ändern.

1. Klicken Sie auf dem Windows-Desktop mit der rechten Maustaste auf das Symbol ”Arbeitsplatz” und wählen Sie den Menübefehl ”Eigenschaften”.
2. Öffnen Sie das Register ”Geräte-Manager”. Wählen Sie bei Windows 2000 zunächst das Register ”Hardware” und dann die Schaltfläche ”Geräte-Manager”.
3. Doppelklicken Sie auf ”Anschlüsse (COM und LPT)”.
4. Doppelklicken Sie auf die Kommunikationsschnittstelle, mit der Sie derzeit arbeiten (z.B. COM1).
5. Klicken Sie im Register ”Anschlusseinstellungen” auf die Schaltfläche ”Erweitert”.
6. Stellen Sie für den Empfangspuffer und für den Sendepuffer den niedrigsten Wert (1) ein.
7. Übernehmen Sie die Änderung mit ”OK”, schließen Sie alle Fenster und starten Sie den PC neu, damit die Änderungen wirksam werden.

Aufbauen des Netzes

Allgemeine Richtlinien

Versehen Sie blitzschlaggefährdete Leitungen immer mit einem geeigneten Überspannungsschutz.

Vermeiden Sie es, Niederspannungssignalleitungen und Kommunikationskabel in der gleichen Kabelbahn wie AC-Versorgungsleitungen und schnellschaltende Hochgeschwindigkeits-DC-Leitungen zu verlegen. Leitungen sollten Sie paarweise verlegen: den Neutral- oder Nullleiter zusammen mit dem Phasenleiter oder der Signalleitung.

Die Kommunikationsschnittstelle der S7-200 CPU ist nicht potentialgetrennt. Verwenden Sie evtl. einen RS-485-Busverstärker oder ein Erweiterungsmodul EM 277 für die Potentialtrennung in Ihrem Netz.

Vorsicht

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

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

Stellen Sie sicher, dass alle Geräte, die Sie über ein Kommunikationskabel miteinander verbinden, entweder den gleichen Bezugsleiter im Stromkreis haben oder elektrisch getrennt sind, damit keine unerwünschten Ströme auftreten. Weitere Hinweise zu Erdung und Bezugspunkten in galvanisch getrennten Stromkreisen finden Sie in Abschnitt 3.

Ermitteln von Abständen, Übertragungsgeschwindigkeiten und Kabeln für Ihr Netz

Wie Sie in Tabelle 7-5 sehen, richtet sich die maximale Länge eines Netzsegments nach zwei Faktoren: Potentialtrennung (RS-485-Busverstärker) und Baudrate.

Potentialtrennung ist erforderlich, wenn Sie Geräte mit unterschiedlichen Erdungspotentialen miteinander verbinden. Unterschiedliche Erdungspotentiale können auftreten, wenn die Erdungspunkte sehr weit voneinander entfernt sind. Auch bei kurzen Entfernungen können Ladeströme von schweren Maschinen unterschiedliche Erdungspotentiale bewirken.

Tabelle 7-5 Maximale Länge des Netzwerkkabels

Baudrate	Nicht potentialgetrennte CPU-Schnittstelle[1]	CPU-Schnittstelle mit Busverstärker oder EM 277
9,6 kBaud bis 187,5 kBaud	50 m	1000 m
500 kBaud	Nicht unterstützt	400 m
1 MBaud bis 1,5 MBaud	Nicht unterstützt	200 m
3 MBaud bis 12 MBaud	Nicht unterstützt	100 m

[1] Die maximal zulässige Entfernung mit Potentialtrennung bzw. Busverstärker beträgt 50 m. Sie messen diese Entfernung zwischen dem ersten Teilnehmer und dem letzten Teilnehmer im Segment.

Busverstärker im Netz

Ein RS-485-Busverstärker schließt das Netzsegment mit einem Abschlusswiderstand ab. Busverstärker können Sie zu folgenden Zwecken einsetzen:

- *So erhöhen Sie die Ausdehnung eines Netzes:* Wenn Sie einen Busverstärker in Ihr Netz aufnehmen, können Sie das Netz um 50 m erweitern. Wenn Sie zwei Busverstärker anschließen, ohne dass sich andere Teilnehmer zwischen den Busverstärkern befinden (wie in Bild 7-21), können Sie das Netz auf die maximale Kabellänge für die Baudrate erweitern. Sie können maximal 9 Busverstärker in Reihe in einem Netz einsetzen, doch die Gesamtlänge des Netzes darf 9600 m nicht überschreiten.
- *So nehmen Sie Geräte in ein Netz auf:* Jedes Segment kann maximal 32 Geräte enthalten, die in einem Abstand von 50 m und mit einer Baudrate von 9600 Baud miteinander verbunden sind. Wenn Sie einen Busverstärker einsetzen, können Sie ein weiteres Segment (32 Geräte) ins Netz aufnehmen.
- *So trennen Sie verschiedene Netzsegmente galvanisch voneinander:* Durch die Potentialtrennung im Netz wird die Qualität der Übertragung erhöht, indem die Netzsegmente getrennt werden, die über verschiedene Erdungspotentiale verfügen.

Ein Busverstärker im Netz zählt als Teilnehmer im Segment, auch wenn der Busverstärker über keine Netzadresse verfügt.

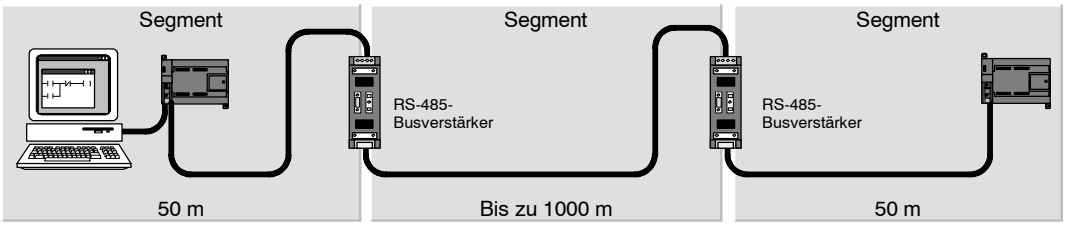


Bild 7-21 Beispiel für ein Netz mit Busverstärkern

Auswählen des Netzwerkkabels

S7-200 Netze setzen verdrillte Doppelleitungen gemäß RS-485 ein. Tabelle 7-6 führt die technischen Daten für das Netzwerkkabel auf. Sie können maximal 32 Geräte in einem Netzsegment anschließen.

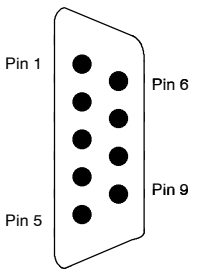
Tabelle 7-6 Allgemeine technische Daten eines Netzwerkkabels

Technische Daten	Beschreibung
Kabelart	Geschirmtes, verdrilltes Leiterpaar
Schleifenwiderstand	$\leq$115 Ω/km
Effektive Kapazität	30 pF/m
Nennimpedanz	Ca. 135 Ω bis 160 Ω *(Frequenz = 3 MHz bis 20 MHz)*
Dämpfung	0,9 dB/100 m *(Frequenz = 200 kHz)*
Aderquerschnitt	0,3 mm^2 bis 0,5 mm^2
Kabeldurchmesser	8 mm ±0,5 mm

Anschlussbelegungen

Die Kommunikationsschnittstellen der S7-200 CPU arbeiten mit RS-485-Signalpegeln und verfügen über 9polige Sub-D-Stecker gemäß dem PROFIBUS-Standard nach der EG-Richtlinie EN 50170. Tabelle7-7 zeigt den Stecker, der die physikalische Verbindung zur Kommunikationsschnittstelle herstellt, und beschreibt die Anschlussbelegung der Kommunikationsschnittstelle.

Tabelle 7-7 Anschlussbelegung der S7-200 Kommunikationsschnittstelle

Klemmenblock	Pin	PROFIBUS-Signal	Schnittstelle 0/Schnittstelle 1
	1	Schirmung	Erdungsanschluss
	2	24-V-Rückleiter	Logischer Leiter
	3	RS-485 Signal B	RS-485 Signal B
	4	Anforderung zum Senden	RTS (TTL)
	5	5-V-Rückleiter	Logischer Leiter
	6	+5 V	+5 V, 100-Ω-Reihenwiderstand
	7	+24 V	+24 V
	8	RS-485 Signal A	RS-485 Signal A
	9	-/-	10-Bit-Protokoll (Eingang)
	Gehäuse des Steckverbinders	Schirmung	Erdungsanschluss

Abschließen des Netzwerkkabels

Siemens bietet zwei Arten von Busanschlusssteckern, mit denen Sie mehrere Geräte schnell und einfach an ein Netz anschließen können: einen Standard-Busanschlussstecker (Anschlussbelegung in Tabelle 7-7) und einen Busanschlussstecker, der eine Programmierschnittstelle umfasst, mit der Sie ein Programmiergerät oder ein HMI-Gerät ans Netz anschließen können, ohne bestehende Netzverbindungen stören zu müssen. Dier Programmierschnittstellenstecker überträgt alle Signale (auch die der Spannungspins) von der S7-200 an die Programmierschnittstelle. Dies ist insbesondere zum Anschließen von Geräten nützlich, die von der S7-200 mit Spannung versorgt werden (z.B. das TD 200).

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. Bild 7-22 zeigt den Abschluss der Kabelstecker.

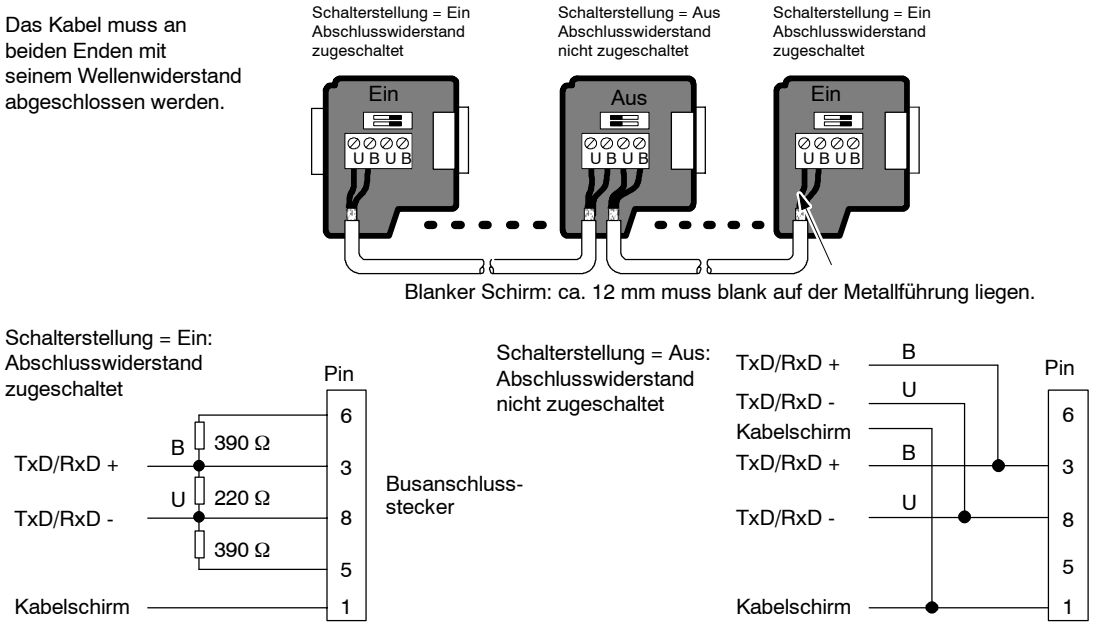


Bild 7-22 Abschließen des Netzwerkkabels

Wählen eines PPI-Multi-Master-Kabels oder einer CP-Karte für Ihr Netz

Wie Sie in Tabelle 7-8 sehen, unterstützt STEP 7-Micro/WIN das RS-232/PPI-Multi-Master-Kabel und das USB/PPI-Multi-Master-Kabel sowie verschiedene CP-Karten, die es ermöglichen, dass das Programmiergerät (Ihr Computer bzw. ein SIMATIC Programmiergerät) als Master im Netz fungiert.

Bei Baudraten bis zu 187,5 kBaud bietet das PPI-Multi-Master-Kabel die einfachste und kostengünstigste Verbindung zwischen STEP 7-Micro/WIN und einer S7-200 CPU bzw. einem S7-200 Netz. Es sind zwei Arten von PPI-Multi-Master-Kabeln erhältlich. Beide Kabel können für die lokale Verbindung zwischen STEP 7-Micro/WIN und einem S7-200 Netz eingesetzt werden.

Das USB/PPI-Multi-Master-Kabel ist ein Plug-&-Play-Gerät für PCs, die die USB-Version 1.1 unterstützen. Das Kabel bietet Potentialtrennung zwischen Ihrem PC und dem S7-200 Netz und unterstützt die PPI-Kommunikation bei Baudraten von maximal 187,5 kBaud. Es müssen keine Schalter eingestellt werden. Sie müssen das Kabel nur anschließen, das PC/PPI-Kabel als Schnittstelle wählen, das PPI-Protokoll einstellen und für die Schnittstelle im Register ”PC-Verbindung” die Option ”USB” wählen. Für die Verwendung durch STEP 7-Micro/WIN darf nur ein USB/PPI-Multi-Master-Kabel an den PC angeschlossen sein.

Das RS-232/PPI-Multi-Master-Kabel verfügt über acht DIP-Schalter: zwei dieser Schalter dienen dazu, das Kabel für den Betrieb mit STEP 7-Micro/WIN zu konfigurieren.

- ❏ Wenn Sie das Kabel an den PC anschließen, wählen Sie PPI-Modus (Schalter 5 = 1) und lokalen Betrieb (Schalter 6 = 0).
- ❏ Wenn Sie das Kabel an ein Modem anschließen, wählen Sie PPI-Modus (Schalter 5 = 1) und entfernten Betrieb (Schalter 6 = 1).

Das Kabel bietet Potentialtrennung zwischen Ihrem PC und dem S7-200 Netz. Wählen Sie das PC/PPI-Kabel als Schnittstelle und wählen Sie im Register ”PC-Verbindung” die RS-232-Schnittstelle, die Sie verwenden möchten. Wählen Sie im Register ”PPI” die Stationsadresse und die Baudrate für das Netz. Sie brauchen keine anderen Einstellungen vorzunehmen, weil das RS-232/PPI-Multi-Master-Kabel das Protokoll automatisch wählt.

Sowohl das USB/PPI- als auch das RS-232/PPI-Multi-Master-Kabel verfügen über LEDs, die die Kommunikationsaktivität am PC und die Kommunikationsaktivität im Netz anzeigen.

- ❏ Die LED Tx zeigt an, dass das Kabel Daten an den PC sendet.
- ❏ Die LED Rx zeigt an, dass das Kabel Daten vom PC empfängt.
- ❏ Die LED PPI zeigt an, dass das Kabel Daten im Netz überträgt. Weil die Multi-Master-Kabel im Besitz des Token sind, leuchtet die LED PPI ständig auf, nachdem die Kommunikation von STEP 7-Micro/WIN initialisiert wurde. Die LED PPI wird ausgeschaltet, wenn die Verbindung zu STEP 7-Micro/WIN geschlossen wird. Die LED PPI blinkt mit einer Frequenz von 1 Hz, wenn auf den Eintritt ins Netz gewartet wird.

Die CP-Karten enthalten spezifische Hardware, die das Programmiergerät darin unterstützen, ein Netz mit mehreren Mastern zu verwalten. Außerdem können so unterschiedliche Protokolle und mehrere Baudraten unterstützt werden.

Alle CP-Karten verfügen über eine RS-485-Schnittstelle zum Anschließen an das Netz. Die Karte CP 5511 PCMCIA besitzt einen Adapter, der die 9polige D-Schnittstelle zur Verfügung stellt. Sie schließen ein Ende des Kabels an die RS-485-Schnittstelle der Karte und das andere Ende an eine Programmierschnittstelle in Ihrem Netz an.

Wenn Sie für die PPI-Kommunikation eine CP-Karte verwenden, unterstützt es STEP 7-Micro/WIN nicht, dass zwei verschiedene Anwendungen auf derselben CP-Karte gleichzeitig laufen. Sie müssen die andere Anwendung schließen, bevor Sie STEP 7-Micro/WIN über die CP-Karte mit dem Netz verbinden. Wenn Sie mit der MPI- oder PROFIBUS-Kommunikation arbeiten, können mehrere STEP 7-Micro/WIN Anwendungen gleichzeitig über das Netz kommunizieren.

Vorsicht

Wenn Sie einen nicht potentialgetrennten RS-485/RS-232-Wandler einsetzen, kann dadurch die RS-232-Schnittstelle Ihres PC beschädigt werden.

Die RS-232/PPI- und USB/PPI-Multi-Master-Kabel von Siemens (Bestellnummern 6ES7 901-3CB30-0XA0 und 6ES7 901-3DB30-0XA0) bieten elektrische Trennung zwischen der RS-485-Schnittstelle an der S7-200 CPU und der RS-232- bzw. USB-Schnittstelle die an Ihren Computer angeschlossen wird. Wenn Sie ein Multi-Master-Kabel nicht von Siemens verwenden, müssen Sie für die Potentialtrennung an der RS-232-Schnittstelle Ihres PC sorgen.

Tabelle 7-8 Von STEP 7-Micro/WIN unterstützte CP-Karten und Protokolle

Konfiguration	Baudrate	Protokoll
RS-232/PPI-Multi-Master-Kabel bzw. USB/PPI-Multi-Master-Kabel[1] angeschlossen an eine Schnittstelle am Programmiergerät	9,6 kBaud bis 187,5 kBaud	PPI
CP 5511 Typ II, PCMCIA-Karte (für ein Notebook)	9,6 kBaud bis 12 MBaud	PPI, MPI und PROFIBUS
CP 5512 Typ II, PCMCIA-Karte (für ein Notebook)	9,6 kBaud bis 12 MBaud	PPI, MPI und PROFIBUS
CP 5611 (ab Version 3) PCI-Karte	9,6 kBaud bis 12 MBaud	PPI, MPI und PROFIBUS
CP 1613, S7-1613 PCI-Karte	10 MBaud oder 100 MBaud	TCP/IP
CP 1612, SoftNet-S7 PCI-Karte	10 MBaud oder 100 MBaud	TCP/IP
CP 1512, SoftNet-S7 PCMCIA-Karte (für einen Notebook-Computer)	10 MBaud oder 100 MBaud	TCP/IP

[1] Die Multi-Master-Kabel bieten Potentialtrennung zwischen der RS-485-Schnittstelle (an der S7-200 CPU) und der Schnittstelle für den Anschluss an Ihren PC. Wenn Sie einen nicht potentialgetrennten RS-485/RS-232-Wandler einsetzen, kann dadurch die RS-232-Schnittstelle Ihres PC beschädigt werden.

HMI-Geräte im Netz

Die S7-200 CPU unterstützt viele Arten von HMI-Geräten von Siemens sowie von anderen Herstellern. Während Sie bei einigen dieser HMI-Geräte (z.B. beim TD 200 oder beim TP070) nicht das Kommunikationsprotokoll für das Gerät einstellen können, können Sie bei anderen Geräten (z.B. beim OP7 und beim TP170) das Kommunikationsprotokoll für das Gerät einstellen.

Wenn Sie bei Ihrem HMI-Gerät das Kommunikationsprotokoll einstellen können, sollten Sie folgendes beachten:

- Ist das HMI-Gerät an die Kommunikationsschnittstelle der S7-200 CPU angeschlossen und es befinden sich keine anderen Geräte im Netz, dann stellen Sie für das HMI-Gerät das PPI- oder das MPI-Protokoll ein.
- Ist das HMI-Gerät an ein EM 277 PROFIBUS-Modul angeschlossen, dann stellen Sie das MPI- oder das PROFIBUS-Protokoll ein.
  - Enthält das Netz, in dem sich das HMI-Gerät befindet, S7-300 oder S7-400 Automatisierungssysteme, dann stellen Sie für das HMI-Gerät das MPI-Protokoll ein.
  - Handelt es sich bei dem Netz, in dem sich das HMI-Gerät befindet, um ein PROFIBUS-Netz, dann stellen Sie für das HMI-Gerät das PROFIBUS-Protokoll ein und wählen ein Profil, das dem der anderen Master im PROFIBUS-Netz entspricht.
- Ist das HMI-Gerät an die Kommunikationsschnittstelle der S7-200 CPU angeschlossen, die als Master eingerichtet ist, dann stellen Sie für das HMI-Gerät das PPI-Protokoll ein. PPI Advanced ist optimal. Das MPI- und das PROFIBUS-Protokoll unterstützen nicht die S7-200 CPU als Master.

Erstellen von anwenderdefinierten Protokollen in der frei programmierbaren Kommunikation

In der frei programmierbaren Kommunikation kann Ihr Programm die Kommunikationsschnittstelle der S7-200 CPU steuern. Sie können in der frei programmierbaren Kommunikation anwenderdefinierte Kommunikationsprotokolle implementieren und so mit vielen Arten von intelligenten Geräten kommunizieren. In der frei programmierbaren Kommunikation werden ASCII- und Binärprotokolle unterstützt.

Die frei programmierbare Kommunikation aktivieren Sie mit den Sondermerkerbytes SMB30 (für Schnittstelle 0) und SMB130 (für Schnittstelle 1). Ihr Programm steuert den Betrieb der Kommunikationsschnittstelle wie folgt:

- Operation Meldung aus Zwischenspeicher übertragen (XMT) und Sende-Interrupt: Mit der Operation Meldung aus Zwischenspeicher übertragen kann die S7-200 bis zu 255 Zeichen von der Kommunikationsschnittstelle senden. Der Sende-Interrupt benachrichtigt das Programm in der S7-200, wenn die Übertragung beendet ist.
- Empfangs-Interrupt: Der Empfangs-Interrupt benachrichtigt das Anwenderprogramm, das ein Zeichen an der Kommunikationsschnittstelle empfangen wurde. Das Programm kann dann je nach dem implementierten Protokoll auf das Zeichen reagieren.
- Operation Meldung in Zwischenspeicher empfangen (RCV): Die Operation Meldung in Zwischenspeicher empfangen empfängt die gesamte Meldung von der Kommunikationsschnittstelle und erzeugt einen Interrupt für das Programm, wenn die Meldung vollständig empfangen wurde. Mit den Sondermerkern der S7-200 richten Sie die Operation Meldung in Zwischenspeicher empfangen anhand von definierten Bedingungen zum Starten und Stoppen des Meldungsempfangs ein. Mit Hilfe der Operation Meldung in Zwischenspeicher empfangen kann das Programm eine Meldung nach den spezifischen Zeichen bzw. nach bestimmten Zeitabständen starten oder stoppen. Die meisten Protokolle können mit der Operation Meldung in Zwischenspeicher empfangen implementiert werden.

Die frei programmierbare Kommunikation kann nur im Betriebszustand RUN der S7-200 aktiviert werden. Wenn die S7-200 in den Betriebszustand STOP versetzt wird, wird die frei programmierbare Kommunikation angehalten und die Kommunikationsschnittstelle kehrt mit den Einstellungen, die im Systemdatenbaustein der S7-200 eingerichtet sind, zum PPI-Protokoll zurück.

Tabelle 7-9 Frei programmierbare Kommunikation

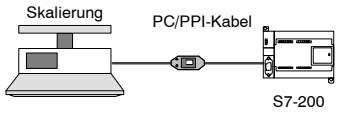
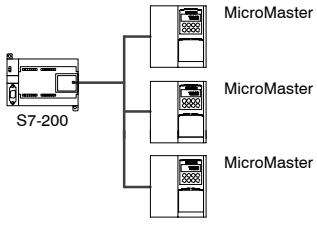
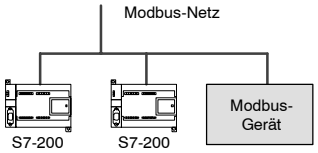
Netzkonfiguration		Beschreibung
Frei programmierbare Kommunikation über eine RS-232-Verbindung		Beispiel: Einsatz einer S7-200 mit einer elektronischen Waage, die über eine RS-232-Schnittstelle verfügt. ■ Das RS-232/PPI-Multi-Master-Kabel verbindet die RS-232-Schnittstelle der Waage mit der RS-485-Schnittstelle der S7-200 CPU. '(Stellen Sie für das Kabel den Modus PPI/frei programmierbare Kommunikation ein, Schalter 5=0.) ■ Die S7-200 CPU nutzt die frei programmierbare Kommunikation für die Kommunikation mit der Waage. ■ Die Baudrate kann zwischen 1200 Baud und 115,2 kBaud liegen. ■ Das Anwenderprogramm definiert das Protokoll.
Einsatz des USS-Protokolls		Beispiel: Einsatz einer S7-200 mit SIMODRIVE MicroMaster-Antrieben. ■ STEP 7-Micro/WIN verfügt über eine USS-Bibliothek. ■ Die S7-200 CPU ist ein Master und die Antriebe sind Slaves. Tipps für die Programmierung *Bei den Tipps für die Programmierung auf der Dokumentations-CD finden Sie ein USS-Beispielprogramm. Siehe Tip 28.*

Tabelle 7-9 Frei programmierbare Kommunikation, continued

Netzkonfiguration		Beschreibung
Erstellen eines Anwenderprogramms, das ein Slave-Gerät in einem anderen Netz emuliert		Beispiel: Anschließen von S7-200 CPUs an ein Modbus-Netz. ■ Das Anwenderprogramm der S7-200 emuliert einen Modbus-Slave. ■ STEP 7-Micro/WIN verfügt über eine Modbus-Bibliothek. Tipps für die Programmierung *Bei den Tipps für die Programmierung auf der Dokumentations-CD finden Sie ein Modbus-Beispielprogramm. Siehe Tip 41.*

RS-232/PPI-Multi-Master-Kabel und frei programmierbare Kommunikation mit RS-232-Geräten

Mit Hilfe des RS-232/PPI-Multi-Master-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 Kabel muss für den Modus PPI/frei programmierbare Kommunikation eingestellt werden (Schalter 5 = 0). Schalter 6 stellt entweder lokalen Modus (DCE) (Schalter 6 = 0) oder entfernten Modus (DTE) (Schalter 6 = 1) ein.

Das RS-232/PPI-Multi-Master-Kabel ist im Sendemodus, wenn Daten von der RS-232-Schnittstelle an die RS-485-Schnittstelle gesendet werden. Das 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 RS-232/PPI-Multi-Master-Kabel unterstützt Baudraten zwischen 1200 Baud und 115,2 kBaud. Mit Hilfe der DIP-Schalter am PC/PPI-Kabel stellen Sie das RS-232/PPI-Multi-Master-Kabel auf die gewünschte Baudrate ein. Tabelle 7-10 zeigt die Schalterstellungen für die entsprechenden Baudraten.

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. Die eingestellte Baudrate beeinflusst die Umschaltezeit (siehe Tabelle 7-10).

Arbeiten Sie mit dem RS-232/PPI-Multi-Master-Kabel in einem System, in dem auch die frei programmierbare Kommunikation eingesetzt wird, dann muss das Programm in der S7-200 die Umschaltezeit aus folgenden Gründen beachten:

Tabelle 7-10 Umschaltezeit und Einstellungen

Baudrate	Umschaltezeit	Einstellungen (1 = oben)
115200	0,15 ms	110
57600	0,3 ms	111
38400	0,5 ms	000
19,200	1,0 ms	001
9600	2,0 ms	010
4800	4,0 ms	011
2400	7,0 ms	100
1200	14,0 ms	101

❑ Die S7-200 reagiert auf Meldungen, die von dem RS-232-Gerät gesendet werden.

Nachdem die S7-200 vom RS-232-Gerät eine Anforderung empfangen hat, muss die S7-200 das Senden der Reaktionsmeldung für einen Zeitraum, der größer oder gleich der Umschaltezeit des Kabels ist, verzögern.

❑ Das RS-232-Gerät reagiert auf Meldungen, die von der S7-200 gesendet werden.

Nachdem die S7-200 vom RS-232-Gerät eine Reaktion empfangen hat, muss die S7-200 das Senden der nächsten Anforderung für einen Zeitraum, der größer oder gleich der Umschaltezeit des Kabels ist, verzögern.

In beiden Situationen erhält das RS-232/PPI-Multi-Master-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.

Modems und STEP 7-Micro/WIN im Netz

STEP 7-Micro/WIN Version 3.2 nutzt die in Windows üblichen Telefon- und Modemeinstellungen zum Auswählen und Einrichten von Telefonmodems. Die Optionen für Telefone und Modems befinden sich in der Windows Systemsteuerung. Mit diesen Einstellungen können Sie:

- ❑ Die meisten internen und externen von Windows unterstützten Modems verwenden.
- ❑ Die Standardkonfigurationen für die meisten von Windows unterstützten Modems verwenden.
- ❑ Die in Windows üblichen Wählparameter bei der Auswahl von Standorten, Landes- und Ortskennzahlen, Impuls- oder Tonwählverfahren und bei der Unterstützung von Calling Cards verwenden.
- ❑ Höhere Baudraten verwenden, wenn Sie mit dem Modemmodul EM 241 kommunizieren.

Sie rufen das Dialogfeld ”Eigenschaften von Modems” über die Windows Systemsteuerung auf. In diesem Dialogfeld können Sie das lokale Modem einrichten. Sie wählen Ihr Modem in der Liste der von Windows unterstützten Modems aus. Wird Ihr Modemtyp nicht im Windows Dialogfeld ”Eigenschaften von Modems” aufgeführt, wählen Sie ein ähnliches Modem oder wenden Sie sich an den Modemhersteller, um die Modemkonfigurationsdateien für Windows zu erhalten.

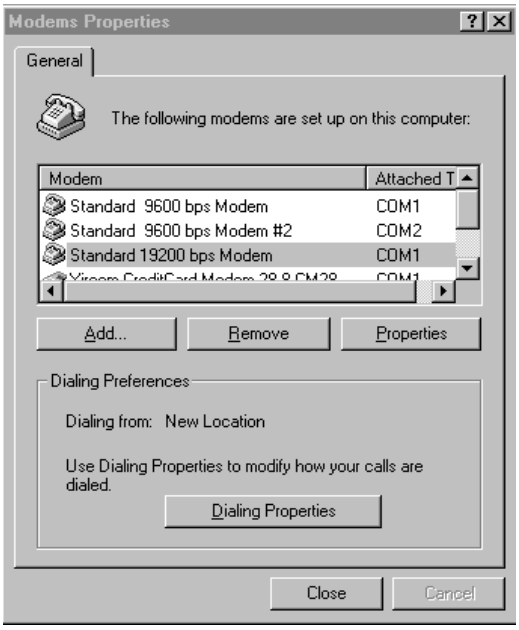


Bild 7-23 Einrichten des lokalen Modems

In STEP 7-Micro/WIN können Sie auch Funk- und Mobilmodems verwenden. Diese Arten von Modems werden im Dialogfeld ”Eigenschaften von Modems” in Windows nicht aufgeführt, doch sie stehen zur Verfügung, wenn Sie eine Verbindung für STEP 7-Micro/WIN einrichten.

Einrichten einer Modemverbindung

Eine Verbindung ordnet den physikalischen Eigenschaften einer Verbindung einen Namen für die Identifizierung zu. Bei Telefonmodems umfassen diese Eigenschaften die Art des Modems, 10- oder 11-Bit-Protokoll und Timeout-Werte. Bei Mobilmodems können Sie für die Verbindung eine PIN-Nummer und andere Parameter einstellen. Bei Funkmodems geben die Eigenschaften auch die Baudrate, Parität, Flusskontrolle und andere Parameter an.

Verbindungs-Assistent

Ergänzen einer Verbindung

Im Verbindungs-Assistent können Sie neue Verbindungen ergänzen, Verbindungen löschen oder bearbeiten (siehe Bild 7-24).

1. Doppelklicken Sie auf das Symbol im Dialogfeld ”Kommunikation”.
2. Doppelklicken Sie auf das PC/PPI-Kabel, um die PG/PC-Schnittstelle aufzurufen. Wählen Sie das PPI-Kabel und klicken Sie auf ”Eigenschaften”. Aktivieren Sie im Register ”Lokale Verbindung” das Kontrollkästchen ”Modemverbindung”.
3. Doppelklicken Sie im Dialogfeld ”Kommunikation” auf die Schaltfläche ”Modem verbinden”.
4. Klicken Sie auf die Schaltfläche ”Einstellungen”, um das Dialogfeld ”Einstellungen der Modemverbindungen” aufzurufen.
5. Klicken Sie auf die Schaltfläche ”Ergänzen”, um den Assistent zum Ergänzen einer Modemverbindung aufzurufen.
6. Der Assistent führt Sie durch die Einrichtung einer Verbindung.

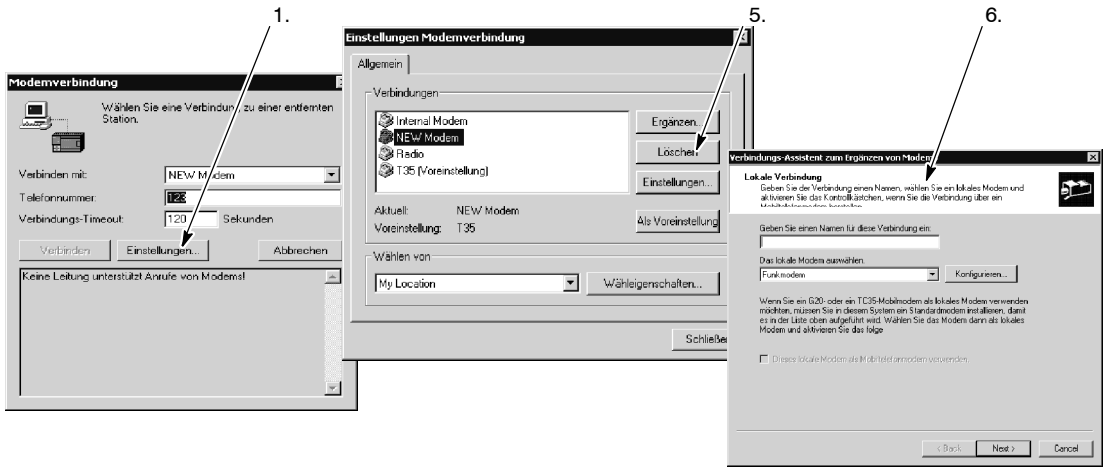


Bild 7-24 Ergänzen einer Modemverbindung

7

Anschließen eines Modems an die S7-200

Nachdem Sie eine Modemverbindung ergänzt haben, können Sie die Verbindung zu einer S7-200 CPU aufbauen.

1. Öffnen Sie das Dialogfeld ”Kommunikation” und doppelklicken Sie auf das Symbol ”Verbinden”, um das Dialogfeld ”Modemverbindung” aufzurufen.
2. Klicken Sie im Dialogfeld ”Modemverbindung” auf ”Verbinden”, um das Modem anzuwählen.

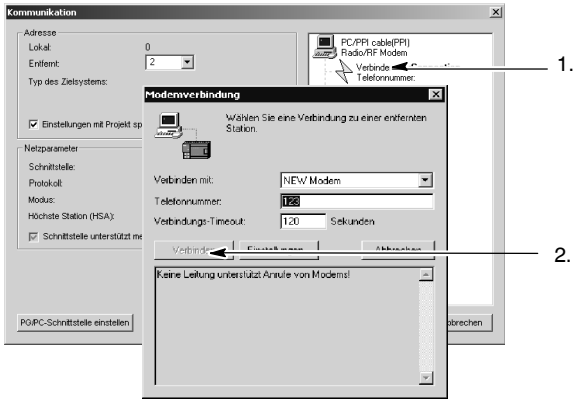


Bild 7-25 Anschließen an die S7-200

Einrichten eines entfernten Modems

Modem-Assistent

Das entfernte Modem ist das Modem, das an die S7-200 angeschlossen ist. Handelt es sich bei dem entfernten Modem um ein Modemmodul EM 241, ist keine Konfiguration erforderlich. Wenn Sie die Verbindung zu einem Standalone-Modem oder einem Mobilmodem herstellen, müssen Sie die Verbindung einrichten.

Im erweiterten Modemassistenten richten Sie das entfernte Modem ein, das an die S7-200 CPU angeschlossen ist. Für die einwandfreie Kommunikation mit der RS-485-Halbduplex-Schnittstelle der S7-200 CPU sind besondere Modemkonfigurationen erforderlich. Wählen Sie einfach den Modemtyp und geben Sie die vom Assistenten verlangten Informationen ein. Ausführliche Informationen hierzu finden Sie in der Online-Hilfe.

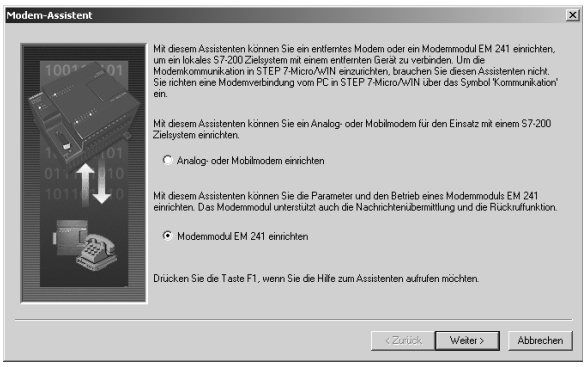


Bild 7-26 Modem-Assistent

Konfigurieren eines PPI-Multi-Master-Kabels für den Betrieb mit einem entfernten Modem

Das RS-232/PPI-Multi-Master-Kabel kann AT-Befehle für Modems bei Einschalten des Kabels senden. Beachten Sie, dass diese Konfiguration nur erforderlich ist, wenn die voreingestellten Modemeinstellungen geändert werden müssen (siehe Bild 7-27).

Modembefehle können in den allgemeinen Befehlen angegeben werden. Der Befehl zum automatischen Antworten ist die einzige Voreinstellung.

Autorisierungsbefehle und PIN-Nummern für Mobiltelefone können im Feld ”Autorisierung Mobiltelefon” eingegeben werden, z.B. +CPIN=1234.

Jeder Befehl wird einzeln an das Modem gesendet. Vor jedem Befehl wird ein AT-Modembefehl gesendet.

Diese Befehle werden über die Schaltfläche ”Programmieren/Test” im Kabel initialisiert.

Das Bitmap zeigt die empfohlenen Schalterstellungen entsprechend den gewählten Parametern an.

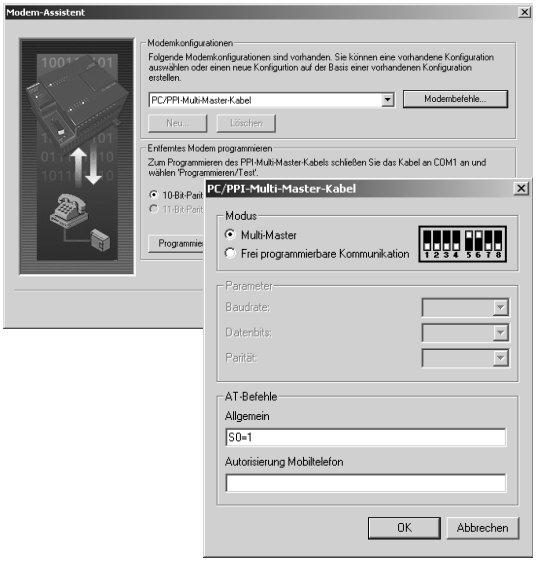


Bild 7-27 Modem-Assistent - Senden von Modembefehlen

Wenn Sie das RS-232/PPI-Multi-Master-Kabel mit STEP 7-Micro/WIN konfigurieren, müssen Sie den RS-485-Steckverbinder an die S7-200 CPU anschließen. Dies ist die Quelle für die 24-V-Spannung für den Betrieb des Kabels. Beachten Sie, dass Sie die S7-200 CPU mit Spannung versorgen.

Nachdem Sie die Konfiguration des RS-232/PPI-Multi-Master-Kabels in STEP 7-Micro/WIN beendet haben, trennen Sie das Kabel vom PC und schließen es an das Modem an. Schalten Sie das Modem und das Kabel aus und wieder ein. Jetzt können Sie das Kabel für den entfernten Betrieb in einem PPI-Multi-Master-Netz einsetzen.

Tipp

Für Ihr Modem müssen die werkseitigen Voreinstellungen eingestellt sein, damit Sie ein PPI-Multi-Master-Kabel verwenden können.

Konfigurieren eines PPI-Multi-Master-Kabels für den Betrieb in der frei programmierbaren Kommunikation

Das RS-232/PPI-Multi-Master-Kabel kann auch AT-Befehle für Modems senden, wenn das Kabel für die frei programmierbare Kommunikation konfiguriert ist. Beachten Sie, dass diese Konfiguration nur erforderlich ist, wenn die voreingestellten Modemeinstellungen geändert werden müssen.

Das Kabel muss jedoch auch so konfiguriert werden, dass es den Einstellungen der S7-200 Schnittstelle hinsichtlich Baudrate, Parität und Anzahl der Datenbits entspricht. Dies ist erforderlich, weil das S7-200 Anwendungsprogramm die Konfiguration dieser Parameter steuert.

Es können Baudraten zwischen 1,2 kBaud und 115,2 kBaud eingestellt werden.

Es können sieben oder acht Datenbits gewählt werden.

Es kann gerade, ungerade oder keine Parität gewählt werden.

Das Bitmap zeigt die empfohlenen Schalterstellungen entsprechend den gewählten Parametern an.

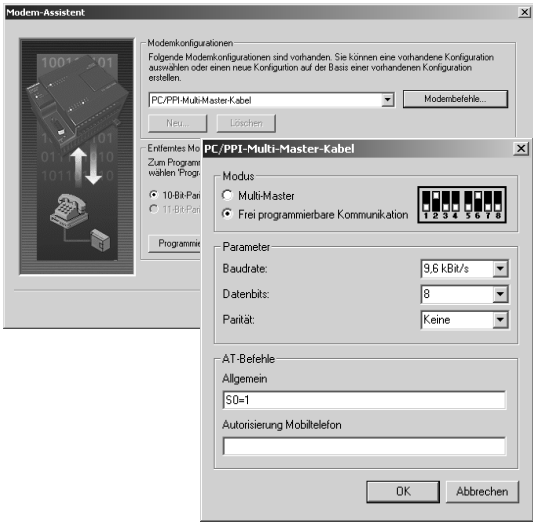


Bild 7-28 Modem-Assistent - Senden von Modembefehlen in der frei programmierbaren Kommunikation

Wenn Sie das RS-232/PPI-Multi-Master-Kabel mit STEP 7-Micro/WIN konfigurieren, müssen Sie den RS-485-Steckverbinder an die S7-200 CPU anschließen. Dies ist die Quelle für die 24-V-Spannung für den Betrieb des Kabels. Beachten Sie, dass Sie die S7-200 CPU mit Spannung versorgen.

Nachdem Sie die Konfiguration des RS-232/PPI-Multi-Master-Kabels in STEP 7-Micro/WIN beendet haben, trennen Sie das Kabel vom PC und schließen es an das Modem an. Schalten Sie das Modem und das Kabel aus und wieder ein. Jetzt können Sie das Kabel für den entfernten Betrieb in einem PPI-Multi-Master-Netz einsetzen.

Tipp

Für Ihr Modem müssen die werkseitigen Voreinstellungen eingestellt sein, damit Sie ein PPI-Multi-Master-Kabel verwenden können.

Verwenden eines Telefonmodems mit dem RS-232/PPI-Multi-Master-Kabel

Sie können mit einem RS-232/PPI-Multi-Master-Kabel die RS-232-Kommunikationsschnittstelle eines Modems an eine S7-200 CPU anschließen (siehe Bild7-29).

- ❏ Schalter 1, 2 und 3 stellen die Baudrate ein.
- ❏ Schalter 5 stellt PPI oder PPI/frei programmierbare Kommunikation ein.
- ❏ Schalter 6 stellt den lokalen Modus (wie Datenübertragungseinrichtung - DCE) oder den entfernten Modus (wie Datenendgerät - DTE) ein.
- ❏ Schalter 7 stellt das 10-Bit- oder das 11-Bit-PPI-Protokoll ein.

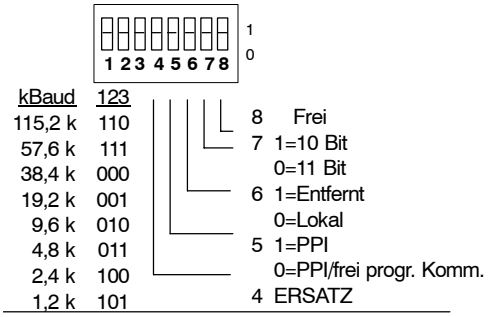


Bild 7-29 Einstellungen für das RS-232/PPI-Multi-Master-Kabel

Schalter 5 stellt den Betrieb im PPI-Modus oder im Modus PPI/frei programmierbare Kommunikation ein. Wenn Sie über STEP 7-Micro/WIN mit der S7-200 über Modems kommunizieren, wählen Sie den PPI-Modus (Schalter 5 = 1). Ansonsten stellen Sie den Modus PPI/frei programmierbare Kommunikation (Schalter 5 = 0) ein.

Schalter 7 des RS-232/PPI-Multi-Master-Kabels stellt den 10-Bit- oder den 11-Bit-Modus für PPI/frei programmierbare Kommunikation ein. Verwenden Sie Schalter 7 nur, wenn die S7-200 über ein Modem im Modus PPI/frei programmierbare Kommunikation an STEP 7-Micro/WIN angeschlossen ist. Andernfalls stellen Sie mit Schalter 7 den 11-Bit-Modus ein, um den einwandfreien Betrieb mit anderen Geräten zu gewährleisten.

Mit Schalter 6 des RS-232/PPI-Multi-Master-Kabels können Sie die RS-232-Schnittstelle des Kabels für den lokalen Modus (DCE) oder den entfernten Modus (DTE) einstellen.

- ❏ Wenn Sie das RS-232/PPI-Multi-Master-Kabel mit STEP 7-Micro/WIN verwenden oder wenn das RS-232/PPI-Multi-Master-Kabel an einen Computer angeschlossen ist, stellen Sie für das RS-232/PPI-Multi-Master-Kabel den lokalen Modus (DCE) ein.
- ❏ Wenn Sie das RS-232/PPI-Multi-Master-Kabel mit einem Modem (d.h. mit einem DCE-Gerät) verwenden, stellen Sie für das RS-232/PPI-Multi-Master-Kabel den entfernten Modus (DTE) ein.

Dadurch benötigen Sie keinen Null-Modem-Adapter zwischen dem RS-232/PPI-Multi-Master-Kabel und dem Modem. Je nach Modemanschluss kann es sein, dass Sie trotzdem noch einen Adapter 9polig/25polig benötigen.

Bild 7-30 zeigt die Anschlussbelegung eines üblichen Modemadapters.

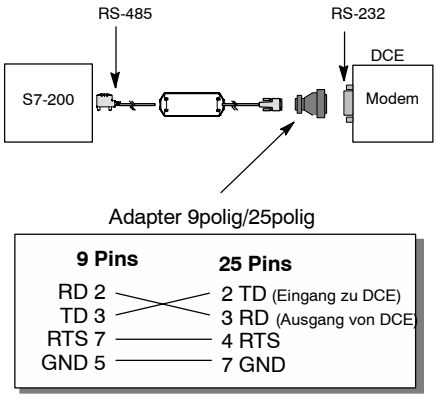


Bild 7-30 Anschlussbelegung der Adapter

Im Anhang A finden Sie weitere Informationen zum RS-232/PPI-Multi-Master-Kabel. Tabelle A-64 zeigt die Anschlussbelegung und die Funktionen der RS-485- und RS-232-Schnittstellen des RS-232/PPI-Multi-Master-Kabels im lokalen Modus (DCE). Tabelle A-65 zeigt die Anschlussbelegung und die Funktionen der RS-485- und RS-232-Schnittstellen des RS-232/PPI-Multi-Master-Kabels im entfernten Modus (DTE). Das RS-232/PPI-Multi-Master-Kabel liefert RTS nur im entfernten Modus (DTE).

Verwenden eines Funkmodems mit dem RS-232/PPI-Multi-Master-Kabel

Sie können mit einem RS-232/PPI-Multi-Master-Kabel die RS-232-Kommunikationsschnittstelle eines Funkmodems an eine S7-200 CPU anschließen. Der Betrieb mit Funkmodems unterscheidet sich jedoch vom Betrieb mit Telefonmodems.

PPI-Modus

Wenn das RS-232/PPI-Multi-Master-Kabel für den PPI-Modus (Schalter 5 = 1) eingestellt ist, wählen Sie üblicherweise den entfernten Modus (Schalter 6 = 1) für den Betrieb mit einem Modem. Wenn Sie jedoch den entfernten Modus eingestellt haben, sendet das Kabel bei jedem Einschalten die Zeichenkette 'AT' und wartet, dass das Modem mit 'OK' antwortet. Bei Telefonmodems wird über diese Sequenz die Baudrate eingestellt, doch Funkmodems akzeptieren im Allgemeinen keine AT-Befehle.

Deshalb müssen Sie für den Betrieb mit einem Funkmodem den lokalen Modus (Schalter 6 = 0) einstellen und einen Null-Modem-Adapter zwischen dem RS-232-Steckverbinder des Kabels und der RS-232-Schnittstelle an Ihrem Funkmodem verwenden. Null-Modem-Adapter gibt es mit den Konfigurationen 9polig/9polig oder 9polig/25polig.

Richten Sie das Funkmodem für den Betrieb mit den Baudraten 9,6, 19,2, 38,4, 57,6 und 115,2 kBaud ein. Das RS-232/PPI-Multi-Master-Kabel passt sich automatisch an eine dieser Baudraten an, wenn das erste Zeichen vom Funkmodem gesendet wird.

PPI/frei programmierbare Kommunikation

Wenn das RS-232/PPI-Multi-Master-Kabel für den Modus PPI/frei programmierbare Kommunikation (Schalter 5 = 0) eingestellt ist, wählen Sie den entfernten Modus (Schalter 6 = 1) für den Betrieb mit einem Funkmodem. Konfigurieren Sie das Kabel so, dass es keine AT-Befehle sendet, um das Modem einzurichten.

Die Schalter 1, 2 und 3 am RS-232/PPI-Multi-Master-Kabel stellen die Baudrate ein (siehe Bild 7-29). Wählen Sie eine Einstellung für die Baudrate, die der Baudrate des Zielsystems und des Funkmodems entspricht.

Für erfahrene Anwender

Optimieren der Leistungsfähigkeit des Netzes

Die folgenden Faktoren beeinflussen die Leistungsfähigkeit eines Netzes (wobei die Baudrate und die Anzahl der Master die stärkste Auswirkung haben).

- ❑ Baudrate: Wenn Sie das Netz mit der höchsten von allen Geräten unterstützten Baudrate betreiben, hat dies die größten Auswirkungen auf die Leistungsfähigkeit des Netzes.
- ❑ Anzahl der Master im Netz: 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.
- ❑ Auswahl der Adressen für Master und Slaves: 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. Sie können die Adressen der Slaves auf beliebige Werte einstellen, 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.
- ❑ GAP-Aktualisierungsfaktor: Der GAP-Aktualisierungsfaktor wird nur verwendet, wenn die S7-200 CPU als PPI-Master genutzt wird. Dann gibt der GAP-Aktualisierungsfaktor an, wie häufig Lücken in den Adressen auf andere Master geprüft werden sollen. Den GAP-Aktualisierungsfaktor stellen Sie in STEP 7-Micro/WIN in der CPU-Konfiguration für eine CPU-Schnittstelle ein. So konfigurieren Sie die S7-200 so, dass die Lücken zwischen den Adressen in regelmäßigen Abständen geprüft werden. Bei einem GAP-Aktualisierungsfaktor von 1 prüft die S7-200 die Lücken in den Adressen jedesmal, wenn sie im Besitz des Token ist. Bei einem GAP-Aktualisierungsfaktor von 2 prüft die S7-200 die Lücken in den Adressen jedes zweite Mal, wenn sie im Besitz des Token ist. 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. Die Voreinstellung für den GAP-Aktualisierungsfaktor ist 10.
- ❑ Höchste Teilnehmeradresse (HSA): Die höchste Teilnehmeradresse wird nur verwendet, wenn die S7-200 CPU als PPI-Master genutzt wird. Dann gibt die höchste HSA die höchste Adresse an, an der ein Master nach neuen Mastern suchen soll. Die HSA stellen Sie in STEP 7-Micro/WIN in der CPU-Konfiguration für eine CPU-Schnittstelle ein. 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. Im allgemeinen müssen Sie die höchste Teilnehmeradresse für alle Master auf den gleichen Wert setzen. Die höchste Teilnehmeradresse sollte der Adresse des höchsten Master entsprechen oder größer sein. Die Voreinstellung für die höchste Teilnehmeradresse ist 31.

Berechnen der Token-Umlaufzeit für ein Netz

In einem Netz mit Token-Passing verfügt der Teilnehmer mit dem Token als einziger Teilnehmer über die Sendeberechtigung. Die Token-Umlaufzeit (die Zeit, die erforderlich ist, damit der Token nacheinander an jeden der Master im logischen Ring weitergegeben werden kann) misst die Leistungsfähigkeit Ihres Netzes.

Bild 7-31 zeigt ein Netz als Beispiel zum Berechnen der Token-Umlaufzeit für ein Multi-Master-Netz. In diesem Beispiel kommuniziert das TD 200 (Teilnehmer 3) mit der CPU 222 (Teilnehmer 2), das TD 200 (Teilnehmer 5) kommuniziert mit der CPU 222 (Teilnehmer 4) usw. Die beiden CPUs 224 erfassen mit den Operationen Aus Netz lesen und In Netz schreiben Daten der anderen S7-200 Geräte: Die CPU 224 (Teilnehmer 6) sendet Meldungen an die Teilnehmer 2, 4 und 8, die CPU 224 (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 224) und aus zwei Slave-Geräten (den beiden CPUs 222).

Programmierung Tips

Bei den Tipps für die Programmierung auf der Dokumentations-CD finden Sie eine Beschreibung von Netzen mit Token-Passing. Siehe Tip 42.

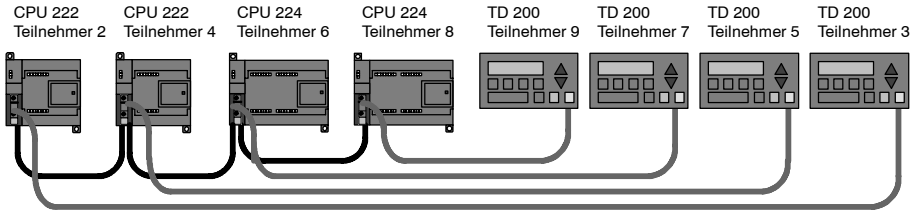


Bild 7-31 Beispiel für ein Netz mit Token-Passing

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.

Die Token-Umlaufzeit richtet sich danach, wie lange der Token bei den einzelnen Teilnehmern verweilt. Sie errechnen die Token-Umlaufzeit für ein Netz mit mehreren Mastern, indem Sie die Token-Verweilzeiten der einzelnen Master addieren. Ist der PPI-Master-Modus aktiviert (im PPI-Protokoll Ihres Netzes), können Sie anderen S7-200 mit Hilfe der Operationen Aus Netz lesen und In Netz schreiben Meldungen senden. Senden Sie mit Hilfe dieser Operationen Meldungen, können Sie mit der im folgenden Bild gezeigten Formel die ungefähre Token-Verweilzeit anhand der folgenden Voraussetzungen berechnen: jeder Teilnehmer sendet eine Anforderung, wenn er im Besitz des Token ist. Bei der Anforderung kann es sich um einen Lese- oder Schreibauftrag für zusammenhängende Adressen handeln. Beim Kommunikationspuffer der S7-200 gibt es keine Nutzungskonflikte, und es gibt keine S7-200 mit einer Zykluszeit, die länger als 10 ms ist.

Token-Verweilzeit (T_{vwz}) = (Zeitbedarf 128 + *n* Datenzeichen) x 11 Bits/Zeichen x 1/Baudrate	
Token-Umlaufzeit (T_{ulz}) = T_{vwz} von Master 1 + T_{ulz} von Master 2 + ... + T_{ulz} von Master *m*	
Erläuterung	*n* ist die Anzahl der Zeichen an Daten (Bytes) *m* ist die Anzahl der Master

Die folgende Gleichung berechnet die Umlaufzeiten (eine “Bitzeit” entspricht der Länge einer Signalperiode) für die in Bild 7-31 gezeigten Beispiele:

T (Token-Verweilzeit) = (128 + 4 Zeichen) x 11 Bits/Zeichen x 1/9600 Bitzeit/s
= 151,25 ms je Master
T (Token-Umlaufzeit) = 151,25 ms/Master $\leqq \times$6 Master
= 907,5 ms

Tipp
Die Software SIMATIC NET COM PROFIBUS bietet eine Analyse zum Ermitteln der Leistungsfähigkeit eines Netzes.

Vergleichen von Token-Umlaufzeiten

Tabelle 7-11 zeigt Vergleiche der Token-Umlaufzeiten bei unterschiedlicher Anzahl von Teilnehmern, unterschiedlichen Datenvolumina und unterschiedlicher Baudrate. Die angegebenen Zeiten beziehen sich auf den Fall, dass Sie die Operationen Aus Netz lesen und In Netz schreiben in der S7-200 CPU oder anderen Mastern einsetzen.

Tabelle 7-11 Token-Umlaufzeit (in Sekunden)

Baudrate	Übertragene Bytes	Anzahl der Master								
		2	3	4	5	6	7	8	9	10
9,6 kBaud	1	0,30	0,44	0,59	0,74	0,89	1,03	1,18	1,33	1,48
	16	0,33	0,50	0,66	0,83	0,99	1,16	1,32	1,49	1,65
19,2 kBaud	1	0,15	0,22	0,30	0,37	0,44	0,52	0,59	0,67	0,74
	16	0,17	0,25	0,33	0,41	0,50	0,58	0,66	0,74	0,83
187,5 kBaud	1	0,009	0,013	0,017	0,022	0,026	0,030	0,035	0,039	0,043
	16	0,011	0,016	0,021	0,026	0,031	0,037	0,042	0,047	0,052

Übersicht über die Verbindungen zwischen Netzgeräten

Netzgeräte kommunizieren über einzelne Verbindungen, bei denen es sich um ”private” Verbindungen zwischen dem Master und dem Slave handelt. Wie Sie in Bild 7-32 sehen, richtet sich das Kommunikationsprotokoll danach, wie die Verbindungen bearbeitet werden:

- Das PPI-Protokoll nutzt eine gemeinsame Verbindung für alle Geräte im Netz.
- Die Protokolle PPI Advanced, MPI und PROFIBUS nutzen getrennte Verbindungen für die Kommunikation zwischen zwei Geräten.

Wenn Sie eines der Protokolle PPI Advanced, MPI oder PROFIBUS verwenden, kann eine Verbindung, die zwischen einem Master und einem Slave aufgebaut wurde, nicht durch einen zweiten Master gestört werden. Die S7-200 CPUs und das EM 277 reservieren immer eine Verbindung für STEP 7-Micro/WIN und eine Verbindung für HMI-Geräte. Andere Master-Geräte können diese reservierten Verbindungen nicht nutzen. Dadurch wird sichergestellt, dass Sie zumindest immer ein Programmiergerät und mindestens ein HMI-Gerät an die S7-200 CPU oder an das EM 277 anschließen können, wenn der Master ein Protokoll nutzt, das solche Verbindungen unterstützt, also z.B. PPI Advanced.

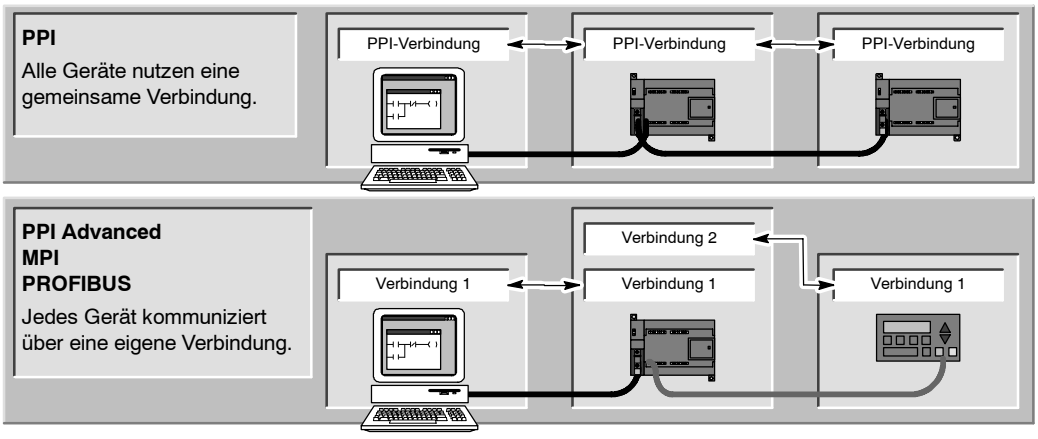


Bild 7-32 Verwaltung der Kommunikationsverbindungen

Wie Sie in Tabelle 7-12 sehen bieten die S7-200 CPU und das EM 277 eine bestimmte Anzahl an Verbindungen. Beide Schnittstellen (Schnittstelle 0 und Schnittstelle 1) einer S7-200 CPU unterstützen bis zu vier separate Verbindungen. (Es kann also maximal acht Verbindungen für die S7-200 CPU geben.) Hinzu kommt die gemeinsam genutzte PPI-Verbindung. Ein EM 277 unterstützt sechs Verbindungen.

Tabelle 7-12 Fähigkeiten der S7-200 CPU und des EM 277

Verbindungsstelle	Baudrate	Verbindungen	Einstellungen in STEP 7-Micro/WIN für das Protokoll
S7-200 CPUSchnittstelle 0	9,6 kBaud, 19,2 kBaud oder 187,5 kBaud	4	PPI, PPI Advanced, MPI und PROFIBUS[1]
Schnittstelle 1	9,6 kBaud, 19,2 kBaud oder 187,5 kBaud	4	PPI, PPI Advanced, MPI und PROFIBUS[1]
EM 277	9,6 kBaud bis 12 MBaud	6 je Modul	PPI Advanced, MPI und PROFIBUS

[1] Wird eine CP-Karte verwendet, um STEP 7-Micro/WIN über Schnittstelle 0 oder Schnittstelle 1 mit der S7-200 CPU zu verbinden, können Sie ein MPI- oder PROFIBUS-Profil nur wählen, wenn das S7-200 Gerät als Slave konfiguriert ist.

7

Komplexe Netze

Bei der S7-200 verfügen komplexe Netze typischerweise über mehrere S7-200 Master, die mit den Operationen Aus Netz lesen (NETR) und In Netz schreiben (NETW) mit anderen Geräten im PPI-Netz kommunizieren. Komplexe Netze weisen üblicherweise bestimmte Probleme auf, die die Kommunikation eines Master mit einem Slave verhindern können.

Wenn im Netz eine niedrige Baudrate (z.B. 9,6 kBaud oder 19,2 kBaud) eingestellt ist, dann beendet jeder Master die Operation (Lesen oder Schreiben), bevor der Token weitergegeben wird. Bei einer Baudrate von 187,5 kBaud jedoch gibt der Master eine Anforderung an den Slave aus und gibt dann den Token weiter, so dass die Anforderung vom Slave noch zu bearbeiten ist.

Bild 7-33 zeigt ein Netz mit möglichen Kommunikationskonflikten. In diesem Netz sind Teilnehmer 1, Teilnehmer 2 und Teilnehmer 3 Master, die mit den Operationen Aus Netz lesen und In Netz schreiben mit Teilnehmer 4 kommunizieren. Die Operationen Aus Netz lesen und In Netz schreiben verwenden das PPI-Protokoll, so dass alle S7-200 Geräte die einzige PPI-Verbindung an Teilnehmer 4 gemeinsam nutzen.

In diesem Beispiel gibt Teilnehmer 1 eine Anforderung an Teilnehmer 4 aus. Bei Baudraten über 19,2 kBaud gibt Teilnehmer 1 den Token dann an Teilnehmer 2 weiter. Wenn Teilnehmer 2 versucht, eine Anforderung an Teilnehmer 4 abzugeben, wird die Anforderung von Teilnehmer 2 abgewiesen, weil die Anforderung von Teilnehmer 1 noch vorhanden ist. Alle Anforderungen an Teilnehmer 4 werden zurückgewiesen, bis Teilnehmer 4 die Antwort an Teilnehmer 1 beendet hat. Erst nach der Fertigstellung der Antwort kann ein anderer Master eine Anforderung an Teilnehmer 4 senden.

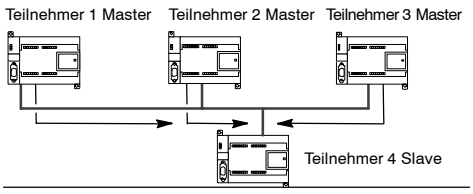


Bild 7-33 Kommunikationskonflikt

Um diesen Konflikt der Kommunikationsschnittstelle an Teilnehmer 4 zu vermeiden, können Sie Teilnehmer 4 als einzigen Master im Netz definieren (siehe Bild 7-34). Teilnehmer 4 gibt dann die Lese-/Schreibanforderungen an die anderen S7-200 Geräte aus.

Diese Konfiguration stellt nicht nur sicher, dass es nicht zu Kommunikationskonflikten kommt, sondern es wird außerdem die Bearbeitungszeit durch das Vorhandensein mehrerer Master verkürzt und der Netzbetrieb läuft effizienter ab.

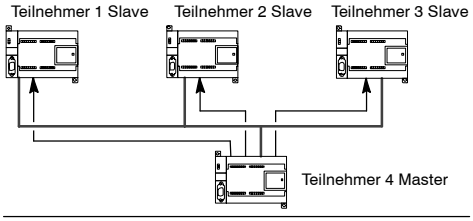


Bild 7-34 Vermeiden von Konflikten

In einigen Anwendungen ist das Verringern der Anzahl der Master im Netz jedoch nicht möglich. Gibt es mehrere Master, müssen Sie die Token-Umlaufzeit verwalten und darauf achten, dass das Netz die Ziel-Token-Umlaufzeit nicht überschreitet. (Die Token-Umlaufzeit ist die Zeit, die zwischen dem Zeitpunkt abläuft, zu dem ein Master den Token weitergibt und dem Zeitpunkt, zu dem der Master den Token erneut erhält.)

Tabelle 7-13 HSA und Ziel-Token-Umlaufzeit

HSA	9,6 kBaud	19,2 kBaud	187,5 kBaud
HSA = 15	0,613 s	0,307 s	31 ms
HSA = 31	1,040 s	0,520 s	53 ms
HSA = 63	1,890 s	0,950 s	97 ms
HSA = 126	3,570 s	1,790 s	183 ms

Ist die Zeit, die erforderlich ist, damit der Token zum Master zurückkehrt, größer als die Ziel-Token-Umlaufzeit, dann darf der Master keine Anforderung senden. Der Master darf nur dann eine Anforderung senden, wenn die tatsächliche Token-Umlaufzeit kleiner als die Ziel-Token-Umlaufzeit ist.

Die Einstellungen für die höchste Teilnehmeradresse (HSA) und die Baudrate für die S7-200 beeinflussen die Ziel-Token-Umlaufzeit. Tabelle 7-13 führt die Ziel-Umlaufzeiten auf.

Bei den niedrigen Baudraten wie 9,6 kBaud und 19,2 kBaud wartet der Master auf eine Antwort auf die Anforderung, bevor der Token weitergegeben wird. Weil die Bearbeitung des Anforderungs-/Antwortzyklus eine relativ lange Zeit in Bezug auf die Zykluszeit einnehmen kann, ist die Wahrscheinlichkeit groß, dass jeder Master im Netz jedesmal, wenn er im Besitz des Token ist, eine Anforderung zum Senden verfügbar hat. Die tatsächliche Token-Umlaufzeit würde sich dadurch erhöhen, und es kann sein, dass einige Master überhaupt keine Anforderungen mehr bearbeiten könnten. In einigen Fällen kann es sein, dass ein Master nur sehr selten Anforderungen bearbeiten darf.

Beispiel: Stellen Sie sich ein Netz mit 10 Mastern vor, die 1 Byte bei 9,6 kBaud und einer HSA von 15 übertragen. In diesem Beispiel hat jeder Master jedesmal eine Anforderung zum Senden. Wie Sie in Tabelle 7-13 sehen, beträgt die Ziel-Umlaufzeit für dieses Netz 0,613 s. Entsprechend der Leistungsfähigkeit jedoch (siehe Tabelle 7-11) beträgt die tatsächlich erforderliche Token-Umlaufzeit für dieses Netz 1,48 s. Weil die tatsächliche Token-Umlaufzeit größer ist als die Ziel-Token-Umlaufzeit können einige Master erst bei späteren Umläufen des Token Meldungen senden.

Sie haben zwei Möglichkeiten, um eine Situation zu verbessern, in der die tatsächliche Token-Umlaufzeit größer als die Ziel-Token-Umlaufzeit ist:

- Sie können die tatsächliche Token-Umlaufzeit durch Verringern der Anzahl der Master im Netz verringern. Je nach Anwendung kann diese Lösung nicht machbar sein.
- Sie können die Ziel-Token-Umlaufzeit durch Erhöhen der HSA für alle Master-Gerät im Netz erhöhen.

Wenn Sie die HSA erhöhen, kann dies zu einem weiteren Problem im Netz führen, weil sich dies auf die Zeit auswirkt, die die S7-200 benötigt, um in den Mastermodus umzuschalten und ins Netz zu gehen. Wenn Sie eine Zeitoperation verwenden, um sicherzustellen, dass die Operationen Aus Netz lesen und In Netz schreiben innerhalb einer angegebenen Zeit vollständig bearbeitet werden, kann es sein, dass die Verzögerung beim Initialisieren des Mastermodus und beim Aufnehmen der S7-200 als Master ins Netz ein Timeout der Operation hervorruft. Sie können die Verzögerung beim Aufnehmen von Mastern gering halten, indem Sie den GAP-Aktualisierungsfaktor für alle Master im Netz verringern.

Wegen der Art und Weise, wie Anforderungen bei 187,5 kBaud an den Slave gesendet und dort gelassen werden, sollten Sie zusätzliche Zeit einplanen, wenn Sie die Ziel-Token-Umlaufzeit einstellen. Bei 187,5 kBaud muss die tatsächliche Token-Umlaufzeit ungefähr die Hälfte der Ziel-Token-Umlaufzeit betragen.

Verwenden Sie zum Ermitteln der Token-Umlaufzeit die Leistungsdaten in Tabelle 7-11, um die Zeit zu berechnen, die zum vollständigen Bearbeiten der Operationen Aus Netz lesen und In Netz schreiben erforderlich ist. Wenn Sie die erforderliche Zeit für HMI-Geräte (wie das TD 200) berechnen möchten, ziehen Sie die Leistungsdaten für die Übertragung von 16 Bytes hinzu. Sie berechnen die Token-Umlaufzeit, indem Sie die Zeit für jedes Gerät im Netz addieren. Wenn Sie alle Zeitwerte addieren, erhalten Sie den schlimmsten Anwendungsfall, in dem alle Geräte während des gleichen Token-Umlaufs eine Anforderung bearbeiten möchten. Dies ist die maximale Token-Umlaufzeit für ein Netz.

Beispiel: Stellen Sie sich ein Netz vor mit einer Baudrate von 9,6 kBaud und vier Textdisplays TD 200 sowie S7-200 Geräten, wobei jede S7-200 jede Sekunde 10 Bytes an Daten in eine andere S7-200 schreibt. Berechnen Sie mit Hilfe von Tabelle 7-11 die spezifischen Übertragungszeiten für das Netz.

4 TDs 200 übertragen 16 Bytes an Daten =	0,66 s
4 S7-200 Geräte übertragen 10 Bytes an Daten =	0,63 s
Gesamt-Token-Umlaufzeit =	1,29 s

Damit im Netz genügend Zeit ist, alle Anforderungen während eines Token-Umlaufs zu bearbeiten, stellen Sie als HSA den Wert 63 ein (siehe Tabelle 7-13). Wenn Sie eine Ziel-Token-Umlaufzeit (1,89 s) einstellen, die größer ist als die maximale Token-Umlaufzeit (1,29 s), stellen Sie dadurch sicher, dass jedes Gerät bei jedem Umlauf des Token Daten übertragen kann.

Um die Zuverlässigkeit des Multi-Master-Netzes zu optimieren, sollten Sie auch folgende Maßnahmen berücksichtigen:

- Ändern Sie die Aktualisierungsrate für die HMI-Geräte, damit mehr Zeit zwischen den Aktualisierungen zur Verfügung steht. Ändern Sie die Aktualisierungsrate z.B. für ein TD 200 von “So schnell wie möglich” in “Jede Sekunde”.
- Verringern Sie die Anzahl der Anforderungen (und die Netzzeiten zum Bearbeiten der Anforderungen) durch Zusammenfassen der jeweiligen Operationen Aus Netz lesen und In Netz schreiben. Verwenden Sie beispielsweise statt zwei Operationen Aus Netz lesen, die jeweils 4 Bytes lesen, nur eine Operation Aus Netz lesen, die 8 Bytes liest. Die Zeit für die Bearbeitung von zwei Anforderungen von 4 Bytes ist viel höher als die Zeit für die Bearbeitung einer Anforderung von 8 Bytes.
- Ändern Sie die Aktualisierungsrate der S7-200 Master so, dass diese nicht versuchen, schneller zu aktualisieren als die Token-Umlaufzeit.

Konfigurieren des RS-232/PPI-Multi-Master-Kabels für den entfernten Betrieb

HyperTerminal als Konfigurationswerkzeug

Wenn Ihnen STEP 7-Micro/WIN nicht für die Konfiguration des RS-232/PPI-Multi-Master-Kabels für den entfernten Betrieb zur Verfügung steht, können Sie das HyperTerminal oder ein beliebiges anderes Terminal-Programm verwenden. Das RS-232/PPI-Multi-Master-Kabel verfügt über integrierte Menüs, die Sie durch die Konfiguration des Kabels für den entfernten Betrieb führen.

Wenn Sie das RS-232/PPI-Multi-Master-Kabel mit dem HyperTerminal konfigurieren, müssen Sie den RS-485-Steckverbinder an die S7-200 CPU anschließen. Dies ist die Quelle für die 24-V-Spannung für den Betrieb des Kabels. Achten Sie darauf, dass Sie die S7-200 CPU mit Spannung versorgen.

Zum Aufrufen des HyperTerminals auf Ihrem PC wählen Sie **Start > Programme > Zubehör > Kommunikation > HyperTerminal**.

Das HyperTerminal wird aufgerufen und fordert Sie auf, eine Beschreibung für die Verbindung einzugeben. Sie müssen einen Namen für die Verbindung eingeben (z.B. Multi-Master). Bestätigen Sie mit ”OK”. Sie können ein Symbol wählen oder das Standardsymbol für die neue Verbindung übernehmen (siehe Bild 7-35).

Bild 7-35 HyperTerminal: ”Beschreibung der Verbindung”

Angezeigt wird das Dialogfeld ”Verbinden mit”. Wählen Sie die Kommunikationsschnittstelle, die Sie verwenden möchten, und bestätigen Sie mit ”OK”. Daraufhin wird das Dialogfeld ”COMx-Eigenschaften” angezeigt. Übernehmen Sie die Voreinstellung und bestätigen Sie mit ”OK” (siehe Bild7-36).

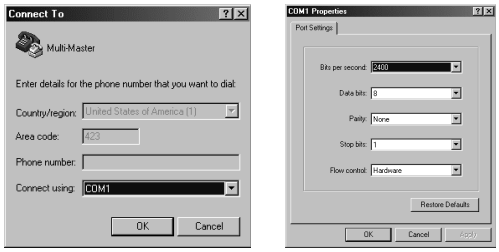


Bild 7-36 HyperTerminal: ”Verbinden mit” und ”COMx-Eigenschaften”

Nachdem Sie mit ”OK” bestätigt haben, wird der Cursor im Bearbeitungsfenster des HyperTerminals platziert (siehe Bild 7-37). Beachten Sie, dass in der Statuszeile am unteren Rand des HyperTerminal-Fensters angezeigt wird, dass die Verbindung hergestellt ist. Außerdem zeigt eine Zeit die Dauer der Verbindung an.

Wählen Sie den Menübefehl **Anrufen > Trennen**. Die Statuszeile zeigt jetzt an, dass die Verbindung getrennt wurde.

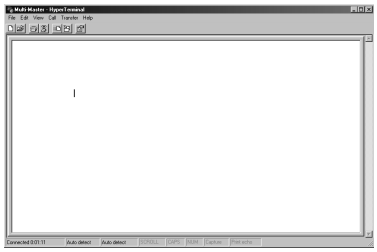


Bild 7-37 HyperTerminal: Bearbeitungsfenster

Wählen Sie **Ansicht > Schriftart**. Wählen Sie ”Courier New” und bestätigen Sie mit ”OK”.

Wählen Sie **Datei > Eigenschaften**. Wählen Sie im Register ”Verbinden mit” die Schaltfläche **Konfigurieren...**, um die Eigenschaften der Kommunikationsschnittstelle anzuzeigen (siehe Bild7-38).

Wählen Sie im Dialogfeld ”COMx-Eigenschaften” im aufklappbaren Listenfeld die Baudrate in Bit/s. Sie müssen eine Baudrate zwischen 9600 und 115200 Bit/s wählen (typischerweise 9600). Wählen Sie 8 Datenbits, keine Parität, ein Stoppbit und keine Flusskontrolle in den entsprechenden aufklappbaren Listenfeldern.

Wählen Sie ”OK”, um ins Register ”Verbinden mit” zurückzukehren.

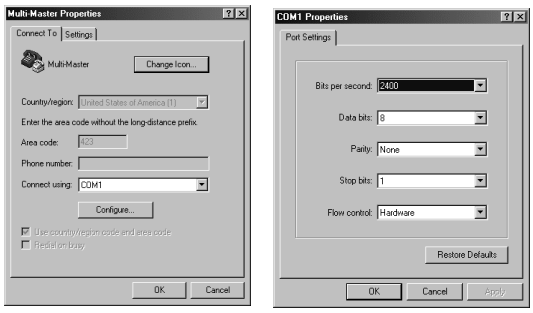


Bild 7-38 Multi-Master-Eigenschaften und COMx-Eigenschaften

Öffnen Sie das Register ”Einstellungen”. Wählen Sie im aufklappbaren Listenfeld unter ”Emulation” die Option ”ANSI” und bestätigen Sie mit ”OK”. Daraufhin kehren Sie ins Bearbeitungsfenster des HyperTerminals zurück. Die Statuszeile am unteren Rand des Dialogfelds sollte folgendes anzeigen: “Verbindung getrennt ANSI 9600 8-N-1” (siehe Bild 7-39).

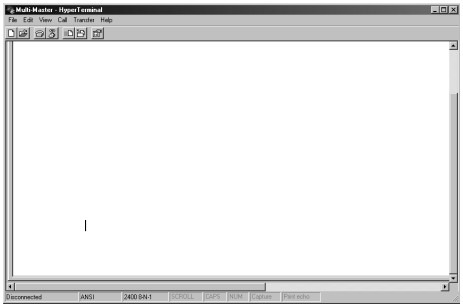


Bild 7-39 Bearbeitung im HyperTerminal - ANSI getrennt

Zum Initiieren der Kommunikation mit dem RS-232/PPI-Multi-Master-Kabel geben Sie “hhh” ein. Die LED Rx muss beim Eingeben von ”hhh” für ca. 1 Sekunde aufleuchten. Die LED Tx wird kurz eingeschaltet, wenn das Kabel eine Sprachauswahl zur Verfügung stellt.

Geben Sie die Nummer ein, die der gewünschten Sprache entspricht (mit der Rücktaste löschen Sie die Voreinstellung), und drücken Sie die Eingabetaste. Bild 7-40 zeigt die Anzeige für die Sprachauswahl sowie die Anzeige für die Einstellung des entfernten Betriebs für das RS232/PPI-Kabel.

In dieser Anzeige wird außerdem der Ausgabestand der Firmware des Kabels angezeigt.

Bild 7-40 HyperTerminal: Sprachauswahl und RS-232/PPI-Kabeleinrichtung

Die Einrichtung des RS232/PPI-Kabels für den entfernten Betrieb führt Sie durch die erforderlichen Schritte, um das Kabel für den von Ihnen gewünschten entfernten Betrieb einzustellen.

- ❑ Wenn Sie mit einer früheren Version von STEP 7-Micro/WIN arbeiten, wählen Sie Option 2 “PPI-Single-Master-Netz mit einem Modem”.
- ❑ Wenn Sie in der frei programmierbaren Kommunikation mit einem Modem arbeiten, wählen Sie Option 3.

Wählen Sie beispielsweise Option 1 für ein PPI-Multi-Master-Netz mit einem Modem, wenn Sie STEP 7-Micro/WIN 3.2 Service Pack 4 oder höher verwenden.

Die HyperTerminal-Anzeige, die in Bild 7-41 dargestellt ist, zeigt die Schalterstellungen an, die Sie am Kabel einstellen müssen. Durch die Schalterstellungen kann STEP 7-Micro/WIN über Modems an einem entfernten Netz mit einem oder mehreren Mastern und einem oder mehreren S7-200 Zielsystemen teilnehmen. Ein solches Netz wird in Bild 7-41 gezeigt.

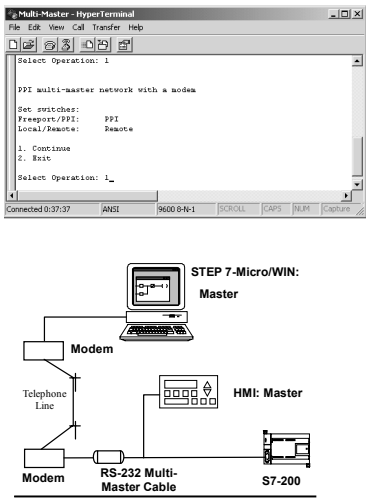


Bild 7-41 HyperTerminal: RS-232/PPI-Kabeleinrichtung

Nachdem Sie die Schalter wie angezeigt eingestellt haben, wählen Sie ”Fortfahren”. Die darauf folgende Anzeige im HyperTerminal wird in Bild 7-42 gezeigt.

Für das entfernte Modem (das Modem, das an das RS-232/PPI-Multi-Master-Kabel angeschlossen ist), müssen die werkseitigen Voreinstellungen eingestellt werden. Wenn für das entfernte Modem die werkseitigen Voreinstellungen eingestellt sind, geben Sie die AT-Befehle ein, die erforderlich sind, um das Modem für den Betrieb mit dem RS-232/PPI-Multi-Master-Kabel zu programmieren. Typischerweise muss nur der Befehl ATS0=1 gesendet werden, wodurch das Modem so konfiguriert wird, dass es automatisch beim ersten Klingeln auf eingehende Anrufe reagiert.

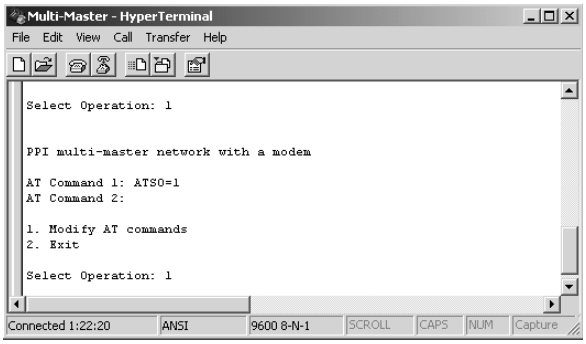


Bild 7-42 HyperTerminal: entfernte Modem

Wenn Sie ein Mobilmodem verwenden, für das eine PIN-Nummer eingegeben werden muss, liefern Sie im zweiten AT-Befehl diese PIN (die von Ihrem Modem unterstützten AT-Befehle finden Sie in der Dokumentation des Modems). Wenn Sie die AT-Befehle ändern müssen, treffen Sie die entsprechende Auswahl und geben die erforderlichen Befehle ein, wenn Sie dazu aufgefordert werden. Die Eingabeaufforderungen liefern Beispiele für AT-Befehle, die Ihnen bei der Formatierung der Befehle behilflich sind.

Das RS-232/PPI-Multi-Master-Kabel sendet diese AT-Befehle immer dann an das Modem, wenn das Kabel eingeschaltet wird. Achten Sie darauf, dass das Modem vor dem Kabel oder gleichzeitig mit dem Kabel eingeschaltet wird. Wenn Sie das Modem aus- und wieder einschalten, müssen Sie dies auch für das Kabel durchführen. Dadurch kann das Kabel das Modem ordnungsgemäß konfigurieren und mit der höchsten verfügbaren Baudrate betrieben werden.

Die Anzeigen des HyperTerminal in Bild 7-43 zeigen, wie Sie die AT-Befehle eingeben. Wenn Sie an der Eingabeaufforderung keinen zweiten AT-Befehl eingeben müssen, drücken Sie die Eingabetaste. Dadurch kommen Sie wieder zu dem Punkt zurück, an dem Sie auswählen können, ob Sie die AT-Befehle ändern oder die Bearbeitung beenden möchten. Wenn Sie die AT-Befehle vollständig eingegeben haben, wählen Sie ”Beenden”.

Nachdem Sie die Konfiguration des RS-232/PPI-Multi-Master-Kabels im HyperTerminal beendet haben, trennen Sie das Kabel vom PC und schließen es an das Modem an. Schalten Sie das Modem und das Kabel aus und wieder ein. Jetzt können Sie das Kabel für den entfernten Betrieb in einem PPI-Multi-Master-Netz einsetzen.

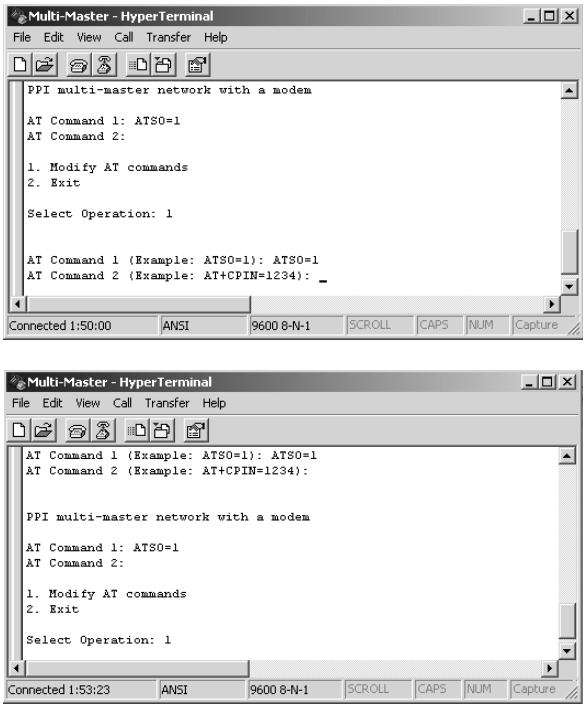


Bild 7-43 HyperTerminal: AT-Befehle

Betrieb in der frei programmierbaren Kommunikation mit dem HyperTerminal

Wenn Sie das RS-232/PPI-Multi-Master-Kabel im HyperTerminal für den Betrieb in der frei programmierbaren Kommunikation konfigurieren möchten, gehen Sie ähnlich wie in dem Beispiel oben vor. Befolgen Sie die Eingabeaufforderungen, um das Kabel entsprechend Ihren Anforderungen zu konfigurieren.

Hardware-Fehlerbehebung und Werkzeuge für den Software-Test

8

STEP 7-Micro/WIN bietet Software-Werkzeuge, mit denen Sie Ihr Programm testen können. Diese Funktionen bieten die Beobachtung des Status während der Ausführung des Programms durch die S7-200, die Ausführung einer bestimmten Anzahl von Zyklen in der S7-200 und das Forcen von Werten.

Ermitteln Sie mit Hilfe der Tabelle 8-1 die Ursache und die mögliche Behebung von Fehlern in der Hardware der S7-200.

In diesem Kapitel

Funktionen zum Testen Ihres Programms 246
Anzeigen des Programmstatus 248
Beobachten und Ändern von Daten in der S7-200 mit Hilfe einer Statustabelle 249
Forcen von Werten 250
Ausführen des Programms für eine bestimmte Anzahl von Zyklen 250
Hinweise zur Fehlerbehebung der Hardware 251

Funktionen zum Testen Ihres Programms

STEP 7-Micro/WIN bietet verschiedene Funktionen, mit denen Sie Ihr Programm testen können: Lesezeichen, Tabelle der Querverweise und Bearbeitungsmöglichkeit zur Laufzeit.

Lesezeichen zum schnellen Zugriff auf das Programm

Sie können in Ihrem Programm Lesezeichen setzen, um bestimmte Zeilen in einem Programm leichter wiederfinden zu können. Sie können in Ihrem Programm zur nächsten oder vorherigen Zeile mit Lesezeichen springen.

Prüfen der Programmreferenzen mit der Tabelle der Querverweise

Die Tabelle der Querverweise zeigt die Querverweise und Informationen zur Verwendung der Elemente in Ihrem Programm an.

Querverweise

Die Tabelle der Querverweise führt alle im Programm verwendeten Operanden auf und gibt alle Vorkommen der Operanden mit Programmbaustein, Netzwerk oder Zeile sowie die Operation an.

Sie können zwischen symbolischer und absoluter Anzeige umschalten, um die Darstellung aller Operanden zu ändern.

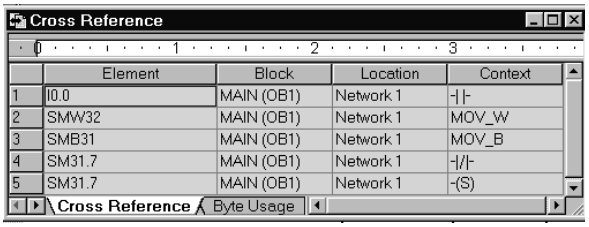


Bild 8-1 Tabelle der Querverweise

Tipp

Wenn Sie auf ein Element in der Tabelle der Querverweise doppelklicken, springt der Cursor an die entsprechende Stelle im Programm oder Baustein.

Bearbeiten Ihres Programms im Betriebszustand RUN

Die CPU 224 ab Ausgabestand 1.10 und die CPU 226 ab Ausgabestand 1.00 unterstützen die Bearbeitung im Betriebszustand RUN. Die Fähigkeit, das Programm im Betriebszustand RUN zu bearbeiten, dient dazu, kleinere Änderungen am Anwenderprogramm vornehmen zu können, ohne den vom Programm gesteuerten Prozess zu stören. Diese Funktion ermöglicht jedoch auch größere Programmänderungen, die den Prozess beeinträchtigen oder sogar gefährlich werden können.

Warnung

Wenn Sie im Betriebszustand RUN Änderungen in die S7-200 laden, wirken sich die Änderungen sofort auf den Prozess aus. Wenn Sie das Programm im Betriebszustand RUN ändern, kann dies zu unerwartetem Verhalten im Prozess führen, wodurch es zu Tod, schweren Körperverletzungen und/oder Sachschaden kommen kann.

Nur geschultes Personal mit Kenntnis der Auswirkungen einer Programmbearbeitung im Betriebszustand RUN auf das Prozessverhalten darf das Programm im Betriebszustand RUN ändern.

Wenn Sie Ihr Programm im Betriebszustand RUN bearbeiten möchten, muss die S7-200 Online-CPU die Bearbeitung im Betriebszustand RUN unterstützen und sich im Betriebszustand RUN befinden.

1. Wählen Sie den Menübefehl **Testen > Programm in RUN bearbeiten**.
2. Unterscheidet sich das Programm vom Programm in der S7-200, werden Sie aufgefordert, es zu speichern. Die Programmbearbeitung im Betriebszustand RUN wird nur mit dem Programm in der S7-200 ausgeführt.
3. STEP 7-Micro/WIN warnt Sie vor der Programmbearbeitung im Betriebszustand RUN und fragt Sie, ob Sie fortfahren oder abbrechen möchten. Wenn Sie fortfahren, lädt STEP 7-Micro/WIN das Programm aus der S7-200. Dann können Sie Ihr Programm im Betriebszustand RUN bearbeiten. Es bestehen während der Bearbeitung keine Einschränkungen.

Tipp

In KOP werden die Operationen Steigende Flanke (EU) und Fallende Flanke (ED) mit Operanden dargestellt. Wenn Sie sich Informationen zu Flanken anzeigen lassen möchten, wählen Sie in der Ansicht die Schaltfläche ”Querverweise”. Im Register ”Verwendete Flanken” werden die Nummern der Flankenoperationen in Ihrem Programm aufgeführt. Achten Sie darauf, dass Sie während der Programmbearbeitung keine Nummern für Flanken doppelt vergeben.

Laden des Programms im Betriebszustand RUN

In der Funktion ”Programm in RUN bearbeiten” können Sie nur den Programmbaustein bearbeiten, während sich die S7-200 im Betriebszustand RUN befindet. Bevor Sie den Programmbaustein im Betriebszustand RUN laden, denken Sie einen Moment über die Auswirkungen Ihrer im Betriebszustand RUN vorgenommen Änderungen auf den Betrieb der S7-200 nach:

- Wenn Sie die Steuerungslogik für einen Ausgang gelöscht haben, behält die S7-200 den letzten Zustand des Ausgangs bei, bis die S7-200 ausgeschaltet oder in den Betriebszustand STOP versetzt wird.
- Wenn Sie einen schnellen Zähler oder eine Impulsausgabe gelöscht haben, und eine der beiden Funktionen in Betrieb ist, läuft der schnelle Zähler bzw. die Impulsausgabe bis zum nächsten Einschalten bzw. bis zum nächsten Übergang in STOP weiter.
- Wenn Sie eine Operation Interrupt zuordnen gelöscht haben, aber das Interruptprogramm nicht gelöscht haben, führt die S7-200 das Interruptprogramm solange aus, bis die S7-200 ausgeschaltet oder in den Betriebszustand STOP versetzt wird. Auch wenn Sie eine Operation Interrupt trennen gelöscht haben, werden die Interrupts weiterhin ausgeführt, bis die S7-200 ausgeschaltet oder in den Betriebszustand STOP versetzt wird.
- Wenn Sie eine Operation Interrupt zuordnen hinzugefügt haben, die durch den Merker des ersten Zyklus eingeschaltet wird, wird das Ereignis erst aktiviert, nachdem die S7-200 ausgeschaltet oder von RUN in STOP versetzt wurde.
- Wenn Sie eine Operation Alle Interruptereignisse freigeben gelöscht haben, arbeiten die Interrupts solange weiter, bis die S7-200 ausgeschaltet oder von RUN in STOP versetzt wird.
- Wenn Sie die Adresse einer Empfangsbox in der Tabelle geändert haben und die Empfangsbox ist während des Umschaltens vom alten zum geänderten Programm aktiv, dann schreibt die S7-200 die empfangenen Daten weiterhin in die alte Adresse. Die Operationen Aus Netz lesen und In Netz schreiben funktionieren auf die gleiche Art und Weise.
- Logik, die durch den Merker des ersten Zyklus aktiviert wird, wird erst nach dem nächsten Einschalten bzw. nach dem nächsten Übergang von STOP in RUN ausgeführt. Der Merker des ersten Zyklus wird nur durch den Wechsel in den Betriebszustand RUN gesetzt und wird nicht durch eine Bearbeitung im Betriebszustand RUN beeinflusst.

Tipp

Damit Sie Ihr Programm im Betriebszustand RUN laden können, muss die S7-200 die Bearbeitung im Betriebszustand RUN unterstützen. Das Programm muss fehlerfrei übersetzt werden und die Kommunikation zwischen STEP 7-Micro/WIN und der S7-200 muss fehlerfrei sein.

Sie können nur den Programmbaustein in die S7-200 laden.

Klicken Sie in der Funktionsleiste auf das Symbol ”Laden in CPU” oder wählen Sie den Menübefehl **Zielsystem > Laden in CPU**, um das Programm zu laden Wird das Programm erfolgreich übersetzt, lädt STEP 7-Micro/WIN den Programmbaustein in die S7-200.

Beenden der Bearbeitung im Betriebszustand RUN

Zum Beenden der Bearbeitung im Betriebszustand RUN wählen Sie den Menübefehl **Testen > Programm in RUN bearbeiten** erneut, um das Häkchen neben dem Menübefehl zu entfernen. Wurden Änderungen noch nicht gespeichert, können Sie wählen, ob Sie die Bearbeitung fortsetzen möchten, ob Sie die Änderungen laden und die Programmbearbeitung im Betriebszustand RUN beenden möchten oder ob Sie die Programmbearbeitung beenden möchten, ohne die Änderungen in die S7-200 zu laden.

Anzeigen des Programmstatus

In STEP 7-Micro/WIN können Sie den Status des Anwenderprogramms während der Ausführung beobachten. Wenn Sie den Programmstatus beobachten, zeigt der Programm-Editor den Status der Operanden von Operationen an.

Zum Anzeigen des Status klicken Sie auf das Symbol für Programmstatus oder wählen den Menübefehl **Testen > Programmstatus**.

Anzeigen des Programmstatus in KOP und FUP

In STEP 7-Micro/WIN gibt es zwei Möglichkeiten, wie der Status von KOP- und FUP-Programmen angezeigt werden kann.

- Status am Zyklusende: STEP 7-Micro/WIN erfasst die Werte für die Statusanzeige in mehreren Zyklen und aktualisiert dann die Statusanzeige auf dem Bildschirm. Der Status zeigt nicht den tatsächlichen Zustand der einzelnen Elemente zur Zeit der Ausführung an. Der Zyklusende-Status zeigt nicht den Status des Lokaldatenspeichers und den der Akkumulatoren an.

  Für den Status am Zyklusende werden die Statuswerte in allen Betriebszuständen der CPU aktualisiert.

- Status während der Ausführung: STEP 7-Micro/WIN zeigt die Werte der Netzwerke während der Ausführung der Elemente in der S7-200 an. Zum Anzeigen des Ausführungsstatus wählen Sie den Menübefehl **Testen > Ausführungsstatus**.

  Für den Ausführungsstatus werden die Statuswerte nur aktualisiert, wenn sich die CPU im Betriebszustand RUN befindet.

Tipp

In STEP 7-Micro/WIN gibt es eine einfache Methode zum Ändern des Zustands einer Variablen: Wählen Sie einfach die Variable und klicken Sie mit der rechten Maustaste, um das Kontextmenü aufzurufen.

Einrichten der Statusanzeige in KOP- und FUP-Programmen

In STEP 7-Micro/WIN gibt es verschiedene Möglichkeiten, wie der Status im Programm angezeigt werden kann.

Zum Einrichten der Anzeigeoptionen für den Status wählen Sie den Menübefehl **Extras > Optionen**. Öffnen Sie dann das Register "Programm-Editor" (siehe Bild 8-2).

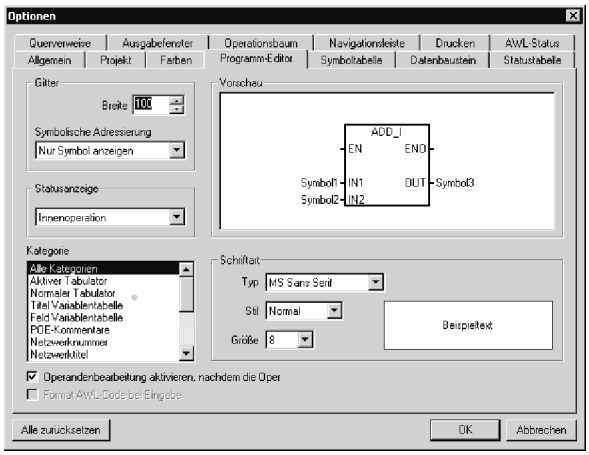


Bild 8-2 Optionen für die Statusanzeige

Anzeigen des Programmstatus in AWL

Sie können den Ausführungsstatus Ihres AWL-Programms operationsweise beobachten. In einem AWL-Programm zeigt STEP 7-Micro/WIN den Status der Operationen an, die auf dem Bildschirm angezeigt werden.

STEP 7-Micro/WIN erfasst die Statusinformationen von der S7-200 und beginnt bei der ersten AWL-Anweisung am oberen Rand des Editor-Fensters. Wenn Sie im Editor-Fenster nach unten blättern, werden neue Informationen aus der S7-200 erfasst.

STEP 7-Micro/WIN aktualisiert die Werte auf dem Bildschirm ununterbrochen. Zum Anhalten der Aktualisierung wählen Sie die Schaltfläche "Ausgelöster Halt". Die aktuellen Daten bleiben auf dem Bildschirm, bis Sie die Schaltfläche "Ausgelöster Halt" erneut anwählen.

Einrichten der im AWL-Programm angezeigten Parameter

In STEP 7-Micro/WIN können Sie den Status verschiedener Parameter für AWL-Anweisungen anzeigen. Wählen Sie den Menübefehl **Extras > Optionen** und öffnen Sie das Register "AWL-Status" (siehe Bild 8-3).

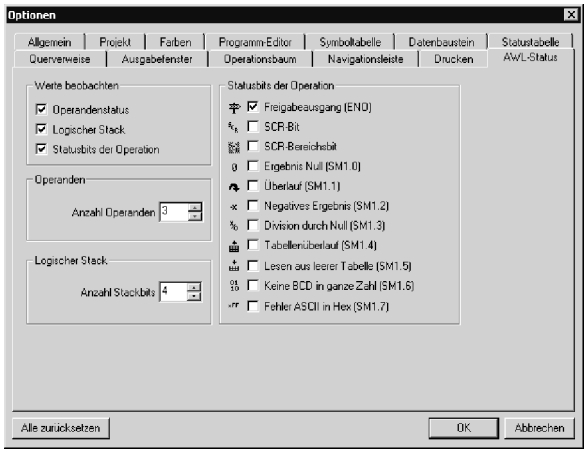


Bild 8-3 Optionen zum Anzeigen des AWL-Status

8

Beobachten und Ändern von Daten in der S7-200 mit Hilfe einer Statustabelle

Mit Hilfe der Statustabelle können Sie Variablen lesen, schreiben, forcen und beobachten, während die S7-200 Ihr Programm ausführt. Wählen Sie den Menübefehl **Ansicht > Komponente > Statustabelle**, um eine Statustabelle anzulegen. Bild 8-4 zeigt ein Beispiel für eine Statustabelle.

Sie können sich den Status mehrfach anzeigen lassen.

STEP 7-Micro/WIN verfügt über Symbole in der Funktionsleiste, mit denen Sie die Statustabelle bearbeiten können: Aufsteigend sortieren, Absteigend sortieren, Einfaches Lesen, Alle schreiben, Forcen, Entforcen, Alle entforcen und Force lesen.

Zum Auswählen eines Formats für ein Feld wählen Sie das Feld und klicken mit der rechten Maustaste, um das Kontextmenü anzuzeigen.

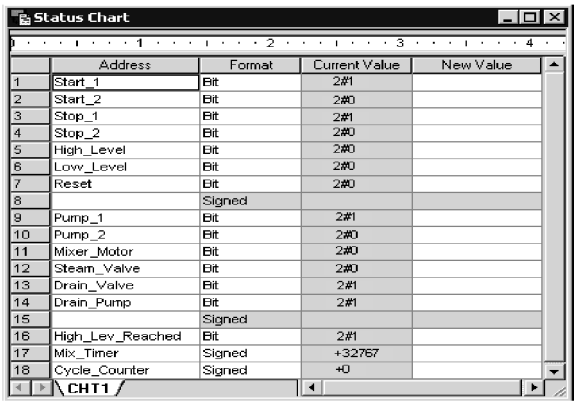


Bild 8-4 Statustabelle

Forcen von Werten

Mit der S7-200 können Sie einige oder alle Ein- und Ausgänge (E- und A-Bits) forcen. Zusätzlich können Sie insgesamt 16 Merker (V oder M) oder Analogeingänge 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 S7-200 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 S7-200 die geforcten Werte immer wieder zu verschiedenen Zeiten im Zyklus.

- *Lesen der Eingänge:* Die S7-200 forct die Eingänge beim Lesen.
- *Ausführen der Steuerungslogik im Programm:* Die S7-200 forct alle direkten E/A-Zugriffe. Nach Bearbeitung des Programms werden maximal 16 Werte im Speicher geforct.
- *Bearbeiten von Kommunikationsanforderungen:* Die S7-200 forct alle Lese-/Schreib-Kommunikationszugriffe.
- *Schreiben in die Ausgänge:* Die S7-200 forct die Ausgänge beim Schreiben.

Sie können die Statustabelle nutzen, um Werte zu forcen. Zum Forcen eines neuen Werts tragen Sie den gewünschten Wert in die Spalte "Neuer Wert" in die Statustabelle ein und klicken auf die Schaltfläche zum Forcen in der Funktionsleiste. Zum Forcen eines vorhandenen Werts markieren Sie den gewünschten Wert in der Spalte "Aktueller Wert" und klicken auf die Schaltfläche zum Forcen.

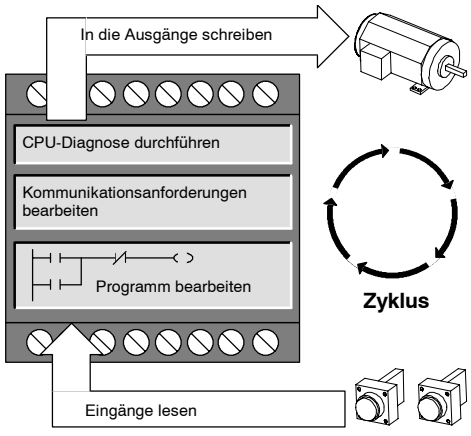


Bild 8-5 Zyklus der S7-200

Tipp

Die Force-Funktion übersteuert eine Operation zum direkten Lesen bzw. Schreiben von Ein- und Ausgängen. Die Force-Funktion übersteuert auch die Tabelle der Ausgänge, die für den Übergang in den Betriebszustand STOP konfiguriert wurde. Geht die S7-200 in den Betriebszustand STOP, stellt der Ausgang den geforcten Wert dar und nicht den Wert, der in der Tabelle der Ausgänge konfiguriert wurde.

Ausführen des Programms für eine bestimmte Anzahl von Zyklen

Zur Unterstützung beim Testen Ihres Programms können Sie in STEP 7-Micro/WIN das Programm für eine bestimmte Anzahl von Zyklen ausführen.

Sie können bewirken, dass die S7-200 nur den ersten Zyklus ausführt. Auf diese Weise können Sie die Daten in der S7-200 nach dem ersten Zyklus beobachten. Wählen Sie den Menübefehl **Testen > Erster Zyklus**, um den ersten Zyklus auszuführen.

Sie können angeben, dass die S7-200 Ihr Programm für eine bestimmte Anzahl von Zyklen bearbeiten soll (von 1 Zyklus bis zu 65.535 Zyklen). Auf diese Weise können Sie das Programm beobachten, wenn die Variablen verändert werden. Mit dem Menübefehl **Testen > Mehrere Zyklen** geben Sie die Anzahl der Zyklen an, die ausgeführt werden soll.

Hinweise zur Fehlerbehebung der Hardware

Tabelle 8-1 Fehlerbehebung der S7-200 Hardware

Symptom	Mögliche Ursachen	Mögliche Behebung
Die Ausgänge arbeiten nicht mehr.	■ Im gesteuerten Gerät ist Überspannung aufgetreten, wodurch der Ausgang beschädigt wurde. ■ Fehler im Anwenderprogramm ■ Verdrahtung lose oder fehlerhaft ■ Überlast ■ Der Ausgang ist geforct.	■ Beim Anschließen an induktive Lasten (z.B. Motoren oder Relais) müssen entsprechende Schutzbeschaltungen eingesetzt werden (siehe Kapitel 3). ■ Korrigieren Sie das Anwenderprogramm. ■ Prüfen Sie die Verdrahtung. ■ Prüfen Sie die Last gegen den Nennwert der Ausgänge. ■ Prüfen Sie die S7-200 auf geforcte E/A.
Die Leuchte SF (Systemfehler) auf der S7-200 wird eingeschaltet.	Die folgende Liste führt die häufigsten Fehlercodes mit ihren Ursachen auf: ■ Fehler im Anwenderprogramm - 0003 Zeitüberwachungsfehler - 0011 Indirekte Adressierung - 0012 Unzulässiger Gleitpunktwert - 0014 Bereichsfehler ■ Elektrische Störungen (0001 bis 0009) ■ Bauteilschaden (0001 bis 0010)	Lesen Sie den Fehlercode des schweren Fehlers und lesen Sie im Anhang C weitere Informationen zur Fehlerart: ■ Bei Programmierfehlern prüfen Sie die Verwendung der Operationen FOR, NEXT, JMP, LBL und CMP. ■ Bei elektrischen Störungen: - Lesen Sie die Verdrahtungsrichtlinien im Kapitel 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.
Keine der LEDs wird eingeschaltet.	■ Sicherung durchgebrannt ■ Umgedrehte 24-V-Leitungen ■ Falsche Spannung	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 finden Sie in den Verdrahtungsrichtlinien im Kapitel 3.
Intermittierender Betrieb durch Hochgeschwindigkeitsgeräte.	■ Schlechte Erdung ■ Weiterleitung der Verdrahtung innerhalb des Schaltschranks ■ Zu kurze Verzögerungszeit für die Eingabefilter	Lesen Sie die Verdrahtungsrichtlinien im Kapitel 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. Erhöhen Sie die Eingabefilterverzögerung im Systemdatenbaustein.
Das Kommunikationsnetz ist beim Anschließen eines externen Geräts nicht mehr funktionsfähig. Entweder ist die Schnittstelle am Computer, die Schnittstelle an der S7-200 oder das PC/PPI-Kabel defekt.	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.	■ Ausführliche Informationen hierzu finden Sie in den Verdrahtungsrichtlinien in Kapitel 3 und in den Richtlinien für Kommunikationsnetze in Kapitel 7. ■ Verwenden Sie ein elektrisch getrenntes PC/PPI-Kabel. ■ Verwenden Sie den galvanisch getrennten RS-485/RS-485-Busverstärker, wenn Sie Geräte anschließen, die nicht den gleichen Bezugsleiter im Stromkreis haben. Informationen zu Bestellnummern für S7-200 Geräte finden Sie in Anhang E.
Sonstige Kommunikationsfehler (STEP 7-Micro/WIN)	Ausführliche Informationen zur Kommunikation im Netz finden Sie in Kapitel 7.	
Fehlerbehebung	Informationen zu Fehlercodes finden Sie in Anhang C.	

8

Erstellen eines Programms für das Positioniermodul

9

Das Positioniermodul EM 253 ist ein S7-200 Sonderfunktionsmodul, das die Impulsfolgen für Drehzahl- und Positionssteuerung von Schrittmotoren und Servomotoren erzeugt. Das Modul kommuniziert mit der S7-200 über den Erweiterungs-E/A-Bus und erscheint in der E/A-Konfiguration als intelligentes Modul mit acht digitalen Ausgängen.

Das Positioniermodul erzeugt die Impulsfolgen für die Positioniersteuerung nach den Konfigurationsinformationen, die im Variablenspeicher der S7-200 abgelegt sind.

Zur Vereinfachung der Positioniersteuerung in Ihren Anwendungen bietet STEP 7-Micro/WIN einen Positionier-Assistenten, mit dem Sie das Positioniermodul innerhalb von wenigen Minuten komplett konfigurieren können. STEP 7-Micro/WIN verfügt auch über ein Steuer-Panel, in dem Sie die Vorgänge der Positioniersteuerung steuern, beobachten und testen können.

In diesem Kapitel

Funktionen des Positioniermoduls 254
Konfigurieren des Positioniermoduls 256
Vom Positionier-Assistenten erzeugte Positionieroperationen 267
Beispielprogramme für das Positioniermodul 279
Beobachten des Positioniermoduls mit em EM 253 Steuer-Panel 284
Fehlercodes für Positioniermodul und Positionieroperationen 286
Für erfahrene Anwender 288

Funktionen des Positioniermoduls

Das Positioniermodul bietet die Funktionalität und Leistungsfähigkeit, die Sie für die einachsige Positioniersteuerung benötigen:

- ❑ Hochgeschwindigkeitssteuerung mit einem Bereich von zwischen 12 Impulsen pro Sekunde bis zu 200.000 Impulsen pro Sekunde
- ❑ Unterstützung von ruckfreier (S-Kurve) und linearer Beschleunigung und Verzögerung
- ❑ Konfigurierbares Messsystem, so dass Sie Daten in physika-
  lische Maßeinheiten (z.B. Zoll oder Zentimeter) oder als Anzahl an Impulsen angeben können
- ❑ Einstellbarer Spielausgleich
- ❑ Unterstützung von absoluten, relativen und manuellen Methoden der Positioniersteuerung
- ❑ Fortlaufende Operation
- ❑ Bis zu 25 Bewegungsprofile mit bis zu 4 Drehzahländer-
  ungen je Profil
- ❑ Vier verschiedene Arten der Referenzpunktsuche mit Aus-
  wahl der Anfangssuchrichtung und der Richtung für das abschließende Anfahren für jede Sequenz
- ❑ Steckbare Feldverdrahtungsklemmenblöcke für einfa-
  chen Ein- und Ausbau

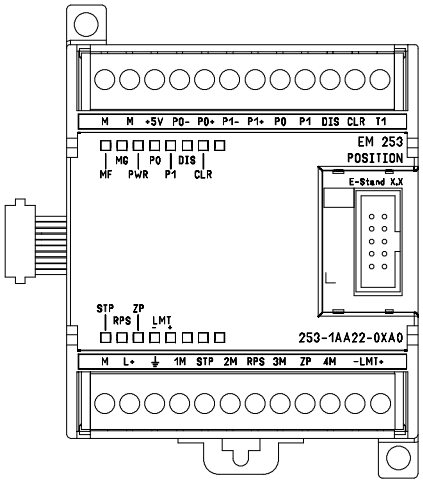


Bild 9-1 Positioniermodul EM 253

In STEP 7-Micro/WIN erstellen Sie sämtliche Konfigurations- und Profilinformationen für das Positioniermodul. Diese Informationen werden mit den Programmbausteinen in die S7-200 geladen. Weil alle Informationen für die Positioniersteuerung in der S7-200 gespeichert werden, können Sie das Positioniermodul austauschen, ohne das Modul neu programmieren oder konfigurieren zu müssen.

Die S7-200 hat 8 Bits im Prozessabbild der Ausgänge (Speicherbereich A) für die Schnittstelle zum Positioniermodul reserviert. Ihr Anwendungsprogramm in der S7-200 nutzt diese Bits für die Steuerung des Betriebs des Positioniermoduls. Diese 8 Ausgangsbits sind nicht mit den physikalischen Feldausgängen des Positioniermoduls verbunden.

Das Positioniermodul verfügt über fünf digitale Eingänge und vier digitale Ausgänge, die die Schnittstelle zu Ihrer Positionieranwendung darstellen (siehe Tabelle 9-1). Diese Eingänge und Ausgänge befinden sich lokal auf dem Positioniermodul. Anhang A liefert die ausführlichen technischen Daten für das Positioniermodul sowie Schaltpläne zum Anschluss des Positioniermoduls an einige Motorantrieb-/Verstärkergeräte.

Tabelle 9-1 Eingänge und Ausgänge des Positioniermoduls

Signal	Beschreibung
STP	Der Eingang STP bewirkt, dass das Modul die laufende Bewegung zum Stillstand bringt. Sie können den gewünschten Betrieb des STP-Eingangs im Positionier-Assistenten einstellen.
RPS	Der Eingang RPS (Referenzpunktschalter) richtet den Referenzpunkt bzw. die Ruhelage für absolute Bewegungsoperationen ein.
ZP	Der Eingang ZP (Nullimpuls) unterstützt die Einrichtung des Referenzpunkts bzw. der Ruhelage. Typischerweise gibt die Motoransteuerung/Verstärker einen Nullimpuls pro Motorumdrehung aus.
LMT+ LMT-	Die Eingänge LMT+ und LMT- richten die Verfahrweggrenzen ein. Den Betrieb der Eingänge LMT+ und LMT- können Sie im Positionier-Assistenten einrichten.
P0 P1 P0+, P0- P1+, P1-	P0 und P1 sind durch Open-Drain-Transistoren erzeugte Impulsausgänge, die Bewegung und Bewegungsrichtung des Motors steuern. P0+, P0- und P1+, P1- sind Differential-Impulsausgänge, die die gleichen Funktionen wie P0 und P1 liefern, jedoch eine höhere Signalqualität bieten. Die Open-Drain-Ausgänge und die Differential- ausgänge sind alle gleichzeitig aktiv. Sie wählen die Impulsausgänge nach den Schnittstellenanforderungen von Motoransteuerung/Verstärker aus.
DIS	DIS ist ein Open-Drain-Transistorausgang, mit dem Sie Motoransteuerung/Verstärker aktivieren bzw. deaktivieren können.
CLR	CLR ist ein Open-Drain-Transistorausgang, mit dem Sie das Servoimpulszählregister löschen können.

Programmierung des Positioniermoduls

STEP 7-Micro/WIN bietet bedienerfreundliche Werkzeuge zum Konfigurieren und Programmieren des Positioniermoduls. Gehen Sie wie folgt vor:

1. Konfigurieren Sie das Positioniermodul. STEP 7-Micro/WIN verfügt über einen Positionier-Assistenten, mit dem Sie die Konfigurations-/Profiltabelle und die Positionieroperationen erstellen können. Auf Seite 256 finden Sie weitere Informationen zum Konfigurieren des Positioniermoduls.
2. Testen Sie den Betrieb des Positioniermoduls. STEP 7-Micro/WIN verfügt über ein EM 253 Steuer-Panel, mit dem Sie die Verdrahtung der Eingänge und Ausgänge, die Konfiguration des Positioniermoduls und den Betrieb der Bewegungsprofile testen können. Auf Seite 284 finden Sie weitere Informationen zum EM 253 Steuer-Panel.
3. Erstellen Sie das Programm für die Ausführung in der S7-200. Der Positionier-Assistent erstellt die Positionieroperationen, die Sie in Ihr Programm einfügen, automatisch. Auf Seite 267 finden Sie weitere Informationen zu den Positionieroperationen. Fügen Sie die folgenden Operationen in Ihr Programm ein:
   - Zum Aktivieren des Positioniermoduls fügen Sie eine Operation POSx_CTRL ein. Mit SM0.0 (Ständig ein) stellen Sie sicher, dass diese Operation in jedem Zyklus ausgeführt wird.
   - Zum Bewegen des Motors in eine bestimmte Stellung nutzen Sie eine der Operationen POSx_GOTO oder POSx_RUN. Die Operation POSx_GOTO fährt eine Lage an, die von den Eingängen des Programms angegeben wird. Die Operation POSx_RUN führt die Bewegungsprofile aus, die Sie im Positionier-Assistenten erstellt haben.
   - Wenn Sie absolute Koordinaten für die Positioniersteuerung verwenden möchten, müssen Sie in Ihrer Anwendung die Nulllage einrichten. Die Nulllage richten Sie mit einer der Operationen POSx_RSEEK oder POSx_LDPOS ein.
   - Die anderen Operationen, die vom Positionier-Assistenten erstellt werden, bieten Funktionen für typische Anwendungen und können je nach Bedarf in Ihren spezifischen Anwendungen eingesetzt werden.
4. Übersetzen Sie das Programm und laden Sie den Systemdatenbaustein, den Datenbaustein und den Programmbaustein in die S7-200.

Tipp

Weitere Informationen zum Anschließen des Positioniermoduls an verschiedene Schrittmotorsteuerungen finden Sie im Anhang A.

Tipp

Stellen Sie die DIP-Schalter an der Schrittmotorsteuerung zur Übereinstimmung mit den Voreinstellungen im Positionier-Assistent auf 10.000 Impulse pro Umdrehung.

Konfigurieren des Positioniermoduls

Positionier-steuerung

Sie müssen eine Konfigurations-/Profiltabelle für das Positioniermodul erstellen, damit das Modul Ihre Positionieranwendung steuern kann. Im Positionier-Assistenten können Sie die Konfiguration schnell und einfach vornehmen. Sie werden schrittweise durch die Konfiguration geführt. Ausführliche Informationen zur Konfigurations-/Profiltabelle finden Sie unter ”Für erfahrene Anwender” auf Seite 288.

Im Positionier-Assistenten können Sie die Konfigurations-/Profiltabelle auch offline erstellen. Sie können die Konfiguration anlegen, ohne mit einer S7-200 CPU mit angeschlossenem Positioniermodul verbunden zu sein.

Damit Sie den Positionier-Assistenten starten können, muss Ihr Projekt übersetzt und die symbolische Adressierung eingestellt sein.

Zum Aufrufen des Positionier-Assistenten klicken Sie in der Navigationsleiste auf das Symbol für Werkzeuge und dann doppelklicken Sie auf das Symbol des Positionier-Assistenten, oder Sie wählen den Menübefehl **Extras > Positionier-Assistent**.

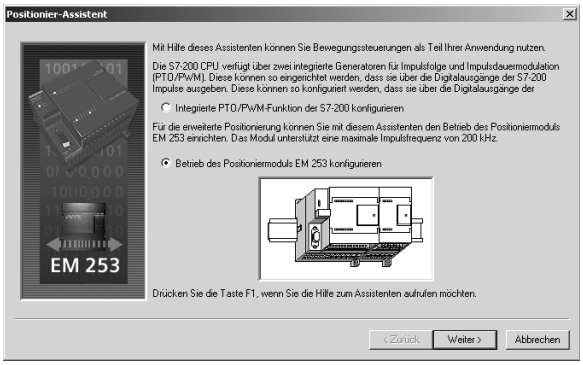


Bild 9-2 Positionier-Assistent

Im Positionier-Assistenten können Sie entweder den Betrieb des Positioniermoduls oder die Funktionen PTO/PWM der Operation Impulsausgabe konfigurieren. Nachdem Sie ausgewählt haben, dass Sie das Positioniermodul einrichten möchten, klicken Sie auf ”Weiter”. Daraufhin führt der Assistent Sie durch die Schritte, die für die Konfiguration des Positioniermoduls erforderlich sind.

9

Eingeben der Position des Positioniermoduls

Sie müssen die Parameter für das Modul und die Bewegungsprofile für Ihre Anwendung definieren, indem Sie den Modultyp und die Modulposition eingeben. Der Positionier-Assistent vereinfacht dies, indem die Position des intelligenten Moduls automatisch gelesen wird. Sie müssen nur auf die Schaltfläche ”Module lesen” klicken.

Bei einer S7-200 CPU mit Firmware vor Version 1.2 müssen Sie das intelligente Modul neben der CPU einbauen, damit der Positionier-Assistent das Modul konfigurieren kann.

Einstellen der Messart

Sie müssen das Messsystem einstellen, das während der Konfiguration verwendet werden soll. Sie können zwischen physikalischen Maßeinheiten und Impulsen wählen. Wenn Sie sich für Impulse entscheiden, müssen Sie keine weiteren Informationen eingeben. Wenn Sie sich für physikalische Maßeinheiten entscheiden, müssen Sie die folgenden Daten eingeben: die Anzahl an Impulsen, die für eine Motorumdrehung benötigt wird (siehe Datenblatt Ihres Motors bzw. Antriebs), die Basis-Maßeinheit (z.B. Zoll, Fuß, Millimeter oder Zentimeter) und den Umfang der Bewegung (bzw. die “Einheiten”) bei einer Motorumdrehung.

STEP 7-Micro/WIN verfügt über ein EM 253 Steuer-Panel, in dem Sie die Anzahl der Einheiten pro Umdrehung ändern können, nachdem das Positioniermodul konfiguriert wurde.

Wenn Sie das Messsystem später ändern, müssen Sie die gesamte Konfiguration einschließlich der vom Positionier-Assistenten erzeugten Operationen löschen. Sie müssen dann die Angaben für das neue Messsystem vornehmen.

Bearbeiten der voreingestellten Eingangs- und Ausgangskonfigurationen

Der Positionier-Assistent verfügt über erweiterte Optionen, mit denen Sie die voreingestellten Eingangs- und Ausgangskonfigurationen für das Positioniermodul anzeigen und bearbeiten können:

- Im Register ”Aktivierungspegel Eingang” können Sie die Einstellungen für die Aktivierungspegel ändern. Ist der Pegel High eingestellt, wird logisch 1 gelesen, wenn am Eingang Signalfluss vorhanden ist. Ist der Pegel Low eingestellt, wird logisch 1 gelesen, wenn am Eingang kein Signalfluss vorhanden ist. Der Signalpegel 1 wird immer so interpretiert, dass die Bedingung aktiv ist. Die LEDs leuchten unabhängig vom Aktivierungspegel, wenn am Eingang Signalfluss vorhanden ist. (Voreinstellung = Aktiv High)
- Im Register ”Zeiten Eingabefilter” können Sie eine Verzögerungszeit (Bereich von 0,20 ms bis 12,80 ms) zum Filtern der Eingänge STP, RPS, LMT+ und LMT- eingeben. Diese Verzögerung dient dazu, ein Rauschen, welches unbeabsichtigte Signalzustandsänderungen an den Eingängen verursachen kann, in der Eingangsverdrahtung zu filtern. (Voreinstellung = 6,4 ms)
- Im Register ”Impulsausgänge und Richtungsausgänge” können Sie die Methode zum Steuern der Richtung eingeben. Sie müssen zunächst die Polarität der Ausgänge eingeben.

Einstellen von positiver Polarität

In einer Anwendung, die mit positiver Polarität arbeitet, wählen Sie eine der folgenden Methoden (siehe Bild 9-3) zum Einrichten Ihres Antriebs und der Ausrichtung Ihrer Anwendung:

- Das Positioniermodul gibt am Ausgang P0 Impulse für positive Drehung und am Ausgang P1 Impulse für negative Drehung aus.
- Das Positioniermodul gibt am Ausgang P0 Impulse aus. Das Modul schaltet den Ausgang P1 für positive Drehung ein und den Ausgang P1 für negative Drehung aus. (Dies ist die Voreinstellung.)

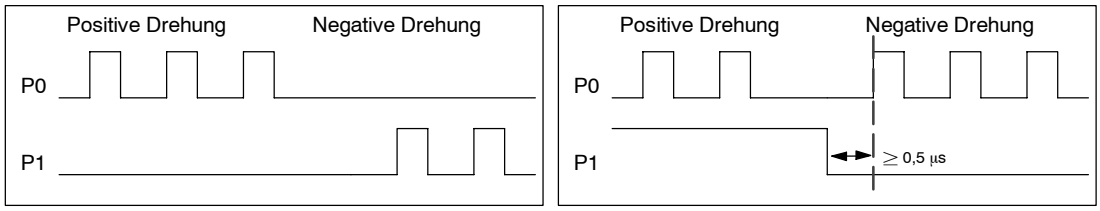


Bild 9-3 Drehmöglichkeiten bei positiver Polarität

Einstellen von negativer Polarität

In einer Anwendung, die mit negativer Polarität arbeitet, wählen Sie eine der folgenden Methoden (siehe Bild 9-4) zum Einrichten Ihres Antriebs und der Ausrichtung Ihrer Anwendung:

- Das Positioniermodul gibt am Ausgang P0 Impulse für negative Drehung und am Ausgang P1 Impulse für positive Drehung aus.
- Das Positioniermodul gibt am Ausgang P0 Impulse aus. Das Modul schaltet den Ausgang P1 für positive Drehung aus und den Ausgang P1 für negative Drehung ein.

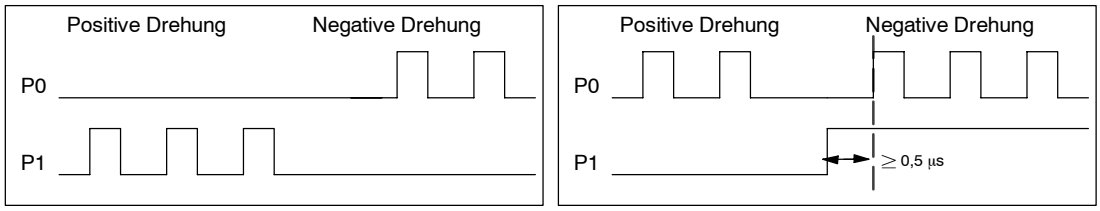


Bild 9-4 Drehmöglichkeiten bei negativer Polarität

Richtungswechsel

Um einen Richtungswechsel herbeizuführen, müssen die Impulsausgänge durch ein Profil oder einen STOP-Befehl gestoppt werden, und es muss ein neuer RUN-Befehl mit der neuen Richtung abgesetzt werden. Das Richtungssignal wird vor dem ersten Impuls der neuen Sequenz mindestens 0,5 µs lang gewechselt.

Einrichten der Reaktion des Moduls auf die physikalischen Eingänge

Sie müssen angeben, wie das Positioniermodul auf die Schalter LMT+ und LMT- und auf den Eingang STP reagiert: keine Aktion (Eingangsbedingung ignorieren), verzögerter Stopp (Voreinstellung) oder sofortiger Stopp.

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.

Bei den Grenzwert- und Stoppfunktionen des Positioniermoduls handelt es sich um elektronische Logik, die nicht den Schutzgrad bietet wie elektromechanische Steuerelemente. Sorgen Sie daher für eine NOT-AUS-Funktion oder elektromechanische oder redundante Sicherheitseinrichtungen, die von Ihrem Positioniermodul und der S7-200 CPU unabhängig sind.

Eingeben der maximalen Drehzahl sowie der Start-/Stoppdrehzahl

Sie müssen für Ihre Anwendung die maximale Drehzahl (MAX_SPEED) sowie die Start-/Stoppdrehzahl (SS_SPEED) eingeben:

- MAX_SPEED: Geben Sie den Wert für die optimale Betriebsdrehzahl Ihrer Anwendung im Bereich des möglichen Motordrehmoments ein. Das für den Antrieb der Last erforderliche Drehmoment ist abhängig von Reibung, Trägheit und den Beschleunigungs-/Verzögerungszeiten. Der Positionier-Assistent berechnet die Mindestdrehzahl, die vom Positioniermodul für die angegebene MAX_SPEED gesteuert werden kann.
- SS_SPEED: Geben Sie einen Wert in dem Bereich ein, in dem der Motor in der Lage ist, die Last mit niedrigen Drehzahlen anzutreiben. Ist der Wert von SS_SPEED zu klein, kommt es möglicherweise zu Schwingungen von Motor und Last oder zu kurzen Sprüngen am Anfang und am Ende des Wegs. Ist der Wert von SS_SPEED zu groß, entgehen dem Motor beim Anlauf evtl. Impulse oder die Last reißt den Motor mit, wenn dieser versucht, die Last zum Stillstand zu bringen.

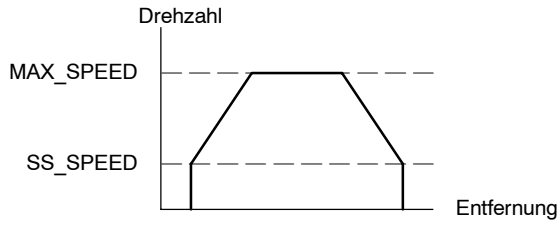


Bild 9-5 Maximale Drehzahl und Start-/Stoppdrehzahl

In Motordatenblättern wird die Start-/Stoppdrehzahl für den Motor bei einer bestimmten Last auf verschiedene Arten angegeben. Üblicherweise beträgt der Wert für die SS_SPEED 5% bis 15% des Werts der MAX_SPEED. Der Wert für die SS_SPEED muss größer sein als die Mindestdrehzahl, die für MAX_SPEED angegeben ist.

Damit Sie die richtigen Drehzahlen für Ihre Anwendung einstellen, ziehen Sie das Datenblatt des Motors hinzu. Bild 9-6 zeigt eine typische Kurve Motordrehmoment/Drehzahl.

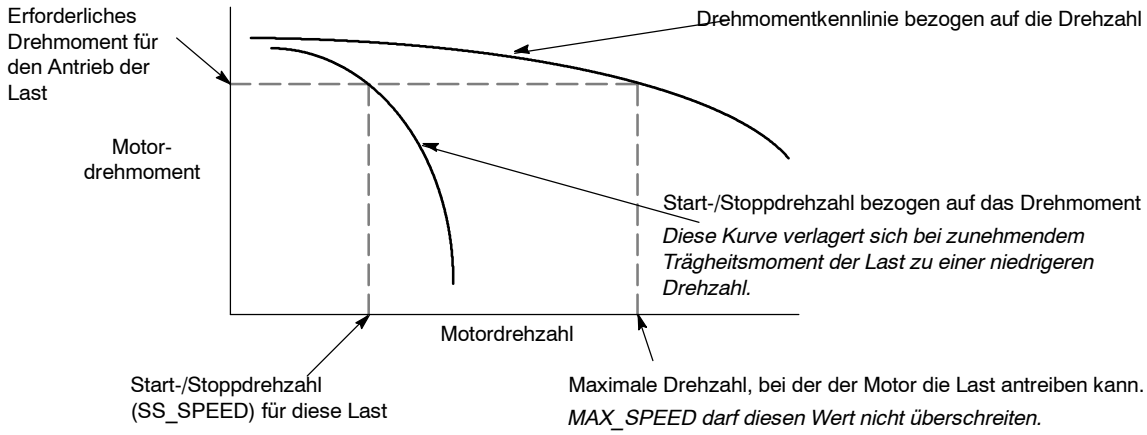


Bild 9-6 Typische Kurve Drehmoment/Drehzahl für einen Motor

Eingeben der Parameter für Tippbetrieb

Der Tippbefehl dient dazu, das Werkzeug manuell in eine gewünschte Position zu bringen. Mit dem Positionier-Assistenten geben Sie die folgenden Werte für die Tippbetriebsparameter ein:

- JOG_SPEED: Bei JOG_SPEED (Tippdrehzahl für den Motor) handelt es sich um die maximale Drehzahl, die bei aktivem Befehl JOG erreicht werden kann.
- JOG_INCREMENT: Dies ist die Entfernung, die das Werkzeug durch einen Befehl JOG bewegt wird.

Bild 9-7 zeigt die Funktionsweise des Tippbefehls. Wenn das Positioniermodul einen Tippbefehl erhält, wird eine Zeit gestartet. Wird der Tippbefehl vor dem Ablauf von 0,5 Sekunden beendet, bewegt das Positioniermodul das Werkzeug mit der von SS_SPEED angegebenen Drehzahl um die von JOG_INCREMENT angegebene Entfernung. Ist der Tippbefehl nach Ablauf von 0,5 Sekunden noch aktiv, beschleunigt das Positioniermodul auf die von JOG_SPEED angegebene Drehzahl. Die Bewegung wird fortgesetzt, bis der Tippbefehl beendet ist. Das Positioniermodul führt dann einen verzögerten Stopp aus. Sie können den Tippbefehl im EM 253 Steuer-Panel oder mit einer Positionieroperation aktivieren.

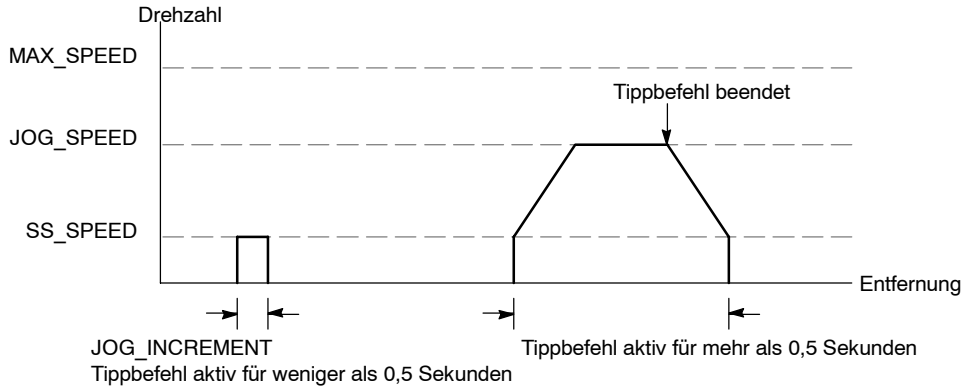


Bild 9-7 Darstellung des Tippbetriebs

Eingeben von Beschleunigungs- und Verzögerungszeiten

Als Teil der Konfiguration des Positioniermoduls stellen Sie die Beschleunigungs- und Verzögerungszeiten ein. Die Voreinstellung für die Beschleunigungszeit und für die Verzögerungszeit ist 1 Sekunde. Typischerweise können Motoren mit weniger als 1 Sekunde arbeiten. Die folgenden Zeiten geben Sie in Millisekunden ein:

- ACCEL_TIME: Zeit, die der Motor für die Beschleunigung von SS_SPEED auf MAX_SPEED benötigt.
  Voreinstellung = 1000 ms
- DECEL_TIME: Zeit, die der Motor für die Verzöge-
  rung von MAX_SPEED auf SS_SPEED benötigt.
  Voreinstellung = 1000 ms

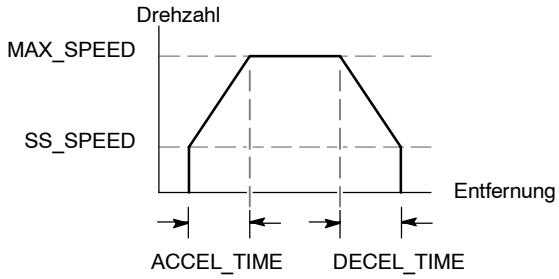


Bild 9-8 Beschleunigungs- und Verzögerungszeiten

Tipp
Die Motorbeschleunigungs- und -verzögerungszeiten werden durch Versuch und Irrtum ermittelt. Geben Sie zunächst einen größeren Wert im Positionier-Assistenten ein. Wenn Sie Ihre Anwendung testen, können Sie die Werte im EM 253 Steuer-Panel nach Bedarf ändern. Optimieren Sie diese Einstellungen für die Anwendung, indem Sie die Zeiten allmählich verkürzen, bis der Motor anfängt zu blockieren.

Eingeben der Ruckausgleichszeit

Der Ruckausgleich führt zu einer ruckfreieren Positioniersteuerung, indem die Änderungsgeschwindigkeit der Bewegung in den Beschleunigungs- und Verzögerungsphasen des Bewegungsprofils verringert wird (siehe Bild 9-9). Ruckverringerung verbessert das Nachführverhalten. Der Ruckausgleich wird auch als “S-Kurvenprofilierung” bezeichnet. Der Ruckausgleich kann nur bei einfachen Einzelschrittprofilen angewendet werden. Dieser Ruckausgleich erfolgt zu Beginn und am Ende der Beschleunigungs- und Verzögerungskurve. Der Ruckausgleich wird nicht auf den ersten und auf den letzten Schritt zwischen der Drehzahl Null und SS_SPEED angewendet.

Sie geben für den Ruckausgleich einen Zeitwert ein (JERK_TIME). Hierbei handelt es sich um die Zeit, die für die Beschleunigung von Null auf die von MAX_SPEED, SS_SPEED und ACCEL_TIME oder entsprechend von DECEL_TIME angegebene Drehzahl benötigt wird. Eine längere Ruckausgleichszeit sorgt für ruckfreieren Betrieb bei geringerem Anstieg der Gesamtzykluszeit als dies durch einfache Erhöhung von ACCEL_TIME oder DECEL_TIME erreicht werden könnte. Der Wert Null zeigt an, dass kein Ruckausgleich angewendet werden soll.

(Voreinstellung = 0 ms)

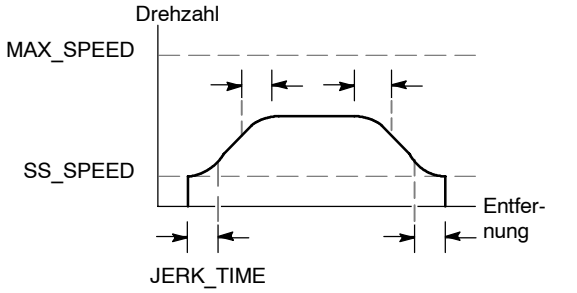


Bild 9-9 Ruckausgleich

Tipp
Ein guter Einstiegswert für JERK_TIME liegt bei 40% von ACCEL_TIME.

9

Konfigurieren eines Referenzpunkts und der Suchparameter

Wenn in Ihrer Anwendung Bewegungen ab einer absoluten Position angegeben werden, müssen Sie eine Nulllage einrichten, die die Maße der Position auf einen bekannten Punkt im physikalischen System fixiert. Eine Vorgehensweise ist die, dass Sie einen Referenzpunkt (RP) in Ihrem physikalischen System liefern. Das Positioniermodul verfügt über einen externen Eingang Referenzpunktschalter (RPS), der bei der Suche des RP zum Einsatz kommt.

Sie können die Parameter für die Referenzpunktsuche (RP-Suche) konfigurieren, die steuern, wie Ihre Positionieranwendung den RP sucht. Der RP kann im Bereich RPS aktiv zentriert sein, der RP kann sich am Rand des Bereichs RPS aktiv befinden oder der RP kann sich eine bestimmte Anzahl an Nullimpuls-Eingangsflanken (ZP) vom Rand des Bereichs RPS aktiv entfernt befinden. Zum Konfigurieren des RP geben Sie die folgenden Informationen ein:

- Sie geben die RP-Suchdrehzahlen für den Motor ein:
  - RP_FAST ist die Anfangsdrehzahl, die das Modul beim Ausführen des RP-Suchbefehls verwendet. Typischerweise beträgt der Wert von RP_FAST ca. 2/3 des Werts von MAX_SPEED.
  - RP_SLOW ist die Drehzahl beim abschließenden Anfahren des RP. Beim Anfahren des RP wird eine langsamere Drehzahl verwendet, damit der RP nicht verpasst wird. Typischerweise ist der Wert von RP_SLOW gleich dem Wert von SS_SPEED.
- Sie geben die anfängliche Suchrichtung (RP_SEEK_DIR) und die Richtung für das abschließende Anfahren (RP_APPR_DIR) bei der RP-Suche ein. Diese Richtungen werden als negative oder positive Richtungen angegeben.
  - RP_SEEK_DIR ist die anfängliche Richtung für die RP-Suche. Typischerweise ist dies die Richtung vom Arbeitsbereich in die Umgebung des Referenzpunkts. Endschalter spielen eine wichtige Rolle beim Definieren des Bereichs, der nach dem RP durchsucht wird. Wird während einer RP-Suche ein Endschalter erreicht, kann dies zu Richtungsumkehr führen, damit die Suche fortgesetzt werden kann. (Voreinstellung = negativ)
  - RP_APPR_DIR ist die Richtung beim abschließenden Anfahren des Referenzpunkts. Um Spiel zu vermeiden und die Genauigkeit zu erhöhen, bewegt sich RP_APPR_DIR in die gleiche Richtung wie der normale Arbeitszyklus. (Voreinstellung = positiv)

Der Positionier-Assistent verfügt über erweiterte Referenzpunktoptionen, mit denen Sie einen RP-Versatz (RP_OFFSET) angeben können. Beim RP-Versatz handelt es sich um die Entfernung zwischen dem RP und der Nulllage (siehe Bild 9-10). Der RP wird über eine Methode identifiziert, bei der eine genaue Position bezüglich des RPS festgelegt wird. Zum Konfigurieren des RP-Versatzes geben Sie die folgenden Werte ein:

- RP_OFFSET: Entfernung vom RP zur Nulllage des physikalischen Messsystems. Voreinstellung = 0
- Spielausgleich: Entfernung, die sich der Motor bewegen muss, um bei Richtungswechsel das Spiel im System zu beseitigen. Der Spielausgleich ist immer ein positiver Wert. Voreinstellung = 0

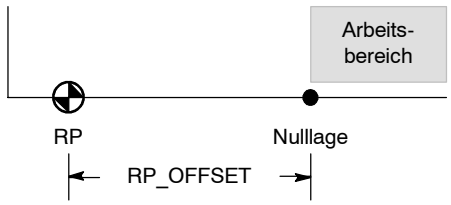


Bild 9-10 Beziehung zwischen RP und Nulllage

Konfigurieren der RP-Suchsequenz

Sie können die Sequenz konfigurieren, mit der das Positioniermodul den Referenzpunkt sucht. Bild 9-11 zeigt eine vereinfachte Darstellung der standardmäßigen RP-Suchsequenz. Sie können die folgenden Optionen für die RP-Suchsequenz einstellen:

- RP-Suchmodus 0: Es wird keine RP-Suchsequenz ausgeführt.
- RP-Suchmodus 1: Der RP liegt an der Stelle, an der der Eingang RPS beim Anfahren von der Arbeitsbereichsseite aktiv wird. (Voreinstellung)
- RP-Suchmodus 2: Der RP liegt zentriert im aktiven Bereich des Eingangs RPS.
- RP-Suchmodus 3: Der RP liegt außerhalb des aktiven Bereichs des Eingangs RPS. RP_Z_CNT gibt an, wie viele ZP (Nullimpulse) gezählt werden sollen, nachdem der Eingang RPS deaktiviert wurde.
- RP-Suchmodus 4: Der RP liegt im allgemeinen im aktiven Bereich des Eingangs RPS. RP_Z_CNT gibt an, wie viele ZP (Nullimpulse) gezählt werden sollen, nachdem der Eingang RPS aktiviert wurde.

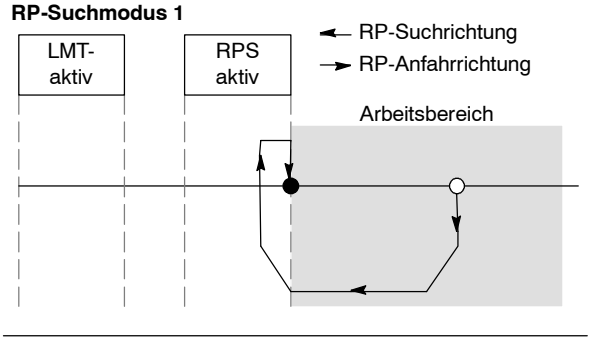


Bild 9-11 Voreingestellte RP-Suchsequenz (vereinfacht)

Tipp

Der Bereich RPS aktiv (hierbei handelt es sich um die Entfernung, die der Eingang RPS aktiv bleibt) muss größer sein als die Entfernung, die erforderlich ist, um von der Drehzahl RP_FAST auf die Drehzahl RP_SLOW zu verzögern. Ist die Entfernung zu gering, erzeugt das Positioniermodul einen Fehler.

Weitere Informationen zu den verschiedenen RP-Suchsequenzen des Positioniermoduls finden Sie in den Bildern 9-14 bis 9-17 auf den Seiten 264 und 265.

Konfigurieren der Bewegungsprofile für das Positioniermodul

Ein Profil ist eine vordefinierte Bewegungsbeschreibung, die sich aus einer oder mehreren Bewegungsdrehzahlen zusammensetzt, die eine Bewegung von einem Anfangspunkt zu einem Endpunkt beeinflusst. Sie brauchen für den Einsatz des Moduls kein Profil zu definieren. Der Positionier-Assistent bietet ein Unterprogramm (POSx_GOTO), mit dem Sie Bewegungen steuern können.

- Anzahl der Profile: Sie können bis zu 25 Profile auswählen.
- Adresse für das Befehlsbyte: Sie müssen die Adresse im Speicherbereich der Ausgänge (A) für das Befehlsbyte des Positioniermoduls eingeben. Im Bild 4-10 auf Seite 31 finden Sie eine Beschreibung der E/A-Nummerierung.
- Adresse für die Konfigurations-/Profiltabelle: Sie müssen die Anfangsadresse im Speicher für die Konfigurations-/Profiltabelle eingeben, die die Konfigurationsdaten für das Positioniermodul und die Daten für alle Profile speichert. Die Konfigurationsdaten für das Positioniermodul benötigen 92 Bytes im Variablenspeicher und jedes Profil benötigt 34 Bytes im Variablenspeicher. Insgesamt werden z.B. für die Konfigurations-/Profiltabelle für ein Positioniermodul mit einem Profil 126 Bytes im Variablenspeicher benötigt.

  Der Positionier-Assistent kann die Adresse eines freien Bereichs im Variablenspeicher in der entsprechenden Größe vorschlagen.

Definieren des Bewegungsprofils

Der Positionier-Assistent verfügt über eine Bewegungsprofildefinition, in der Sie alle Bewegungsprofile für Ihre Anwendung definieren. Für jedes Profil wählen Sie den Betriebszustand und definieren die einzelnen Schritte des Profils. Im Positionier-Assistenten können Sie außerdem für jedes Profil einen symbolischen Namen definieren, indem Sie den symbolischen Namen einfach beim Definieren des Profils eingeben. Nachdem Sie ein Profil konfiguriert haben, können Sie die Konfiguration speichern und die Parameter ausdrucken.

9

Einstellen der Betriebsart für das Profil

Sie konfigurieren des Profil entsprechend der Betriebsart: absolute Position, relative Position, kontinuierlicher Lauf mit fester Drehzahl oder kontinuierlicher Lauf mit zwei Drehzahlen. Bild 9-12 zeigt die verschiedenen Betriebsarten.

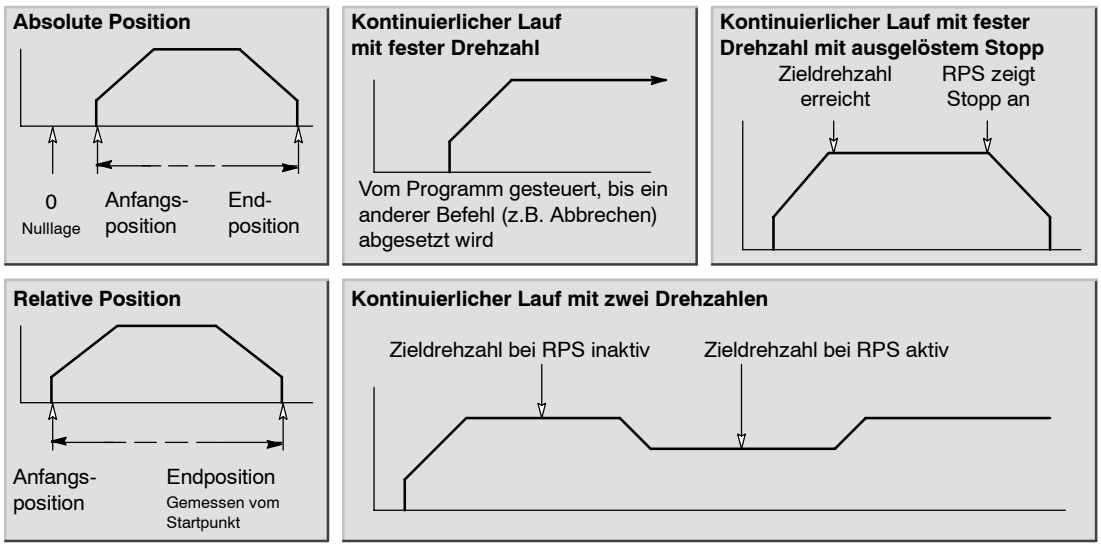


Bild 9-12 Einstellen der Betriebsart für das Positioniermodul

Erstellen der Schritte für das Profil

Ein Schritt ist eine feste Entfernung, die ein Werkzeug bewegt wird. Ein Schritt umfasst auch die Entfernung, die während der Beschleunigungs- und Verzögerungszeiten zurückgelegt wird. Jedes Profil kann aus bis zu 4 einzelnen Schritten bestehen.

Sie geben die Zieldrehzahl und die Endposition für jeden Schritt ein. Wenn Sie mehrere Schritte definieren, klicken Sie auf die Schaltfläche ”Neuer Schritt” und geben für jeden Schritt des Profils Informationen ein. Bild 9-13 zeigt vier mögliche Profile. Es gibt jedoch auch noch andere mögliche Kombinationen.

Wenn Sie auf die Schaltfläche ”Schritt grafisch darstellen” klicken, können Sie sich eine grafische Darstellung des Schritts ansehen, die vom Positionier-Assistenten berechnet wird. Auf diese Weise können Sie jeden Schritt interaktiv und ganz bequem prüfen und bearbeiten.

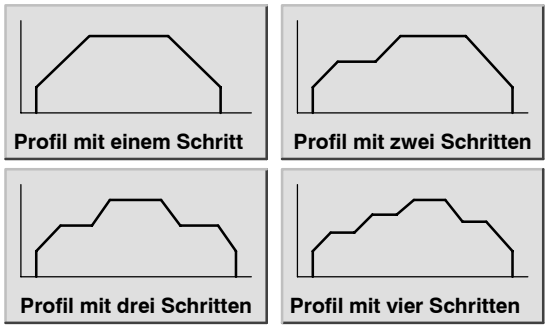


Bild 9-13 Beispiele für Bewegungsprofile

Beenden der Konfiguration des Positioniermoduls

Nachdem Sie den Betrieb des Positioniermoduls konfiguriert haben, klicken Sie auf ”Beenden”. Daraufhin führt der Positionier-Assistent folgende Aktivitäten aus:

- ❏ Die Modulkonfiguration und die Profiltabelle werden in den Datenbaustein Ihres S7-200 Programms eingefügt.
- ❏ Es wird eine globale Symboltabelle für die Bewegungsparameter angelegt.
- ❏ Die Unterprogramme für die Positioniersteuerung werden in den Programmbaustein des Projekts eingefügt, damit sie in der Anwendung verwendet werden können.

Sie können den Positionier-Assistenten erneut ausführen, wenn Sie Konfigurations- oder Profilinformationen ändern möchten.

Tipp
Da der Positionier-Assistent den Programmbaustein, den Datenbaustein und den Systemdatenbaustein ändert, müssen Sie darauf achten, dass Sie alle drei Bausteine in die S7-200 CPU laden. Ansonsten fehlen dem Positioniermodul möglicherweise einige Programmkomponenten für den ordnungsgemäßen Betrieb.

Vom Positioniermodul unterstützte Modi für die RP-Suche

Das folgende Bild zeigt die verschiedenen Möglichkeiten für die einzelnen RP-Suchmodi.

- ❏ Bild 9-14 zeigt zwei der Optionen für den RP-Suchmodus 1. In diesem Modus wird die Lage des RP an der Stelle ermittelt, an der der Eingang RPS beim Anfahren von der Arbeitsbereichsseite aktiv wird.
- ❏ Bild 9-15 zeigt zwei der Optionen für den RP-Suchmodus 2. In diesem Modus wird die Lage des RP im Mittelpunkt des aktiven Bereichs des Eingangs RPS ermittelt.
- ❏ Bild 9-16 zeigt zwei der Optionen für den RP-Suchmodus 3. In diesem Modus wird die Lage des RP als angegebene Anzahl Nullimpulse (ZP) außerhalb des aktiven Bereichs des Eingangs RPS ermittelt.
- ❏ Bild 9-17 zeigt zwei der Optionen für den RP-Suchmodus 4. In diesem Modus wird die Lage des RP als angegebene Anzahl Nullimpulse (ZP) innerhalb des aktiven Bereichs des Eingangs RPS ermittelt.

In jedem Modus gibt es vier Kombinationen aus RP-Suchrichtung und RP-Anfahrrichtung. (Es werden nur zwei der Kombinationen gezeigt.) Diese Kombinationen geben das Muster für die RP-Suche an. Für jede Kombination gibt es auch vier verschiedene Anfangspunkte:

Die Lage der Arbeitsbereiche wurde in den Darstellungen so gewählt, dass die Bewegung vom Referenzpunkt zum Arbeitsbereich eine Bewegung in die gleiche Richtung wie die RP-Anfahrrichtung erfordert. Indem die Lage des Arbeitsbereichs auf diese Weise gewählt wird, wird das gesamte Spiel in der Antriebsmechanik für die erste Bewegung zum Arbeitsbereich im Anschluss an eine Referenzpunktsuche beseitigt.

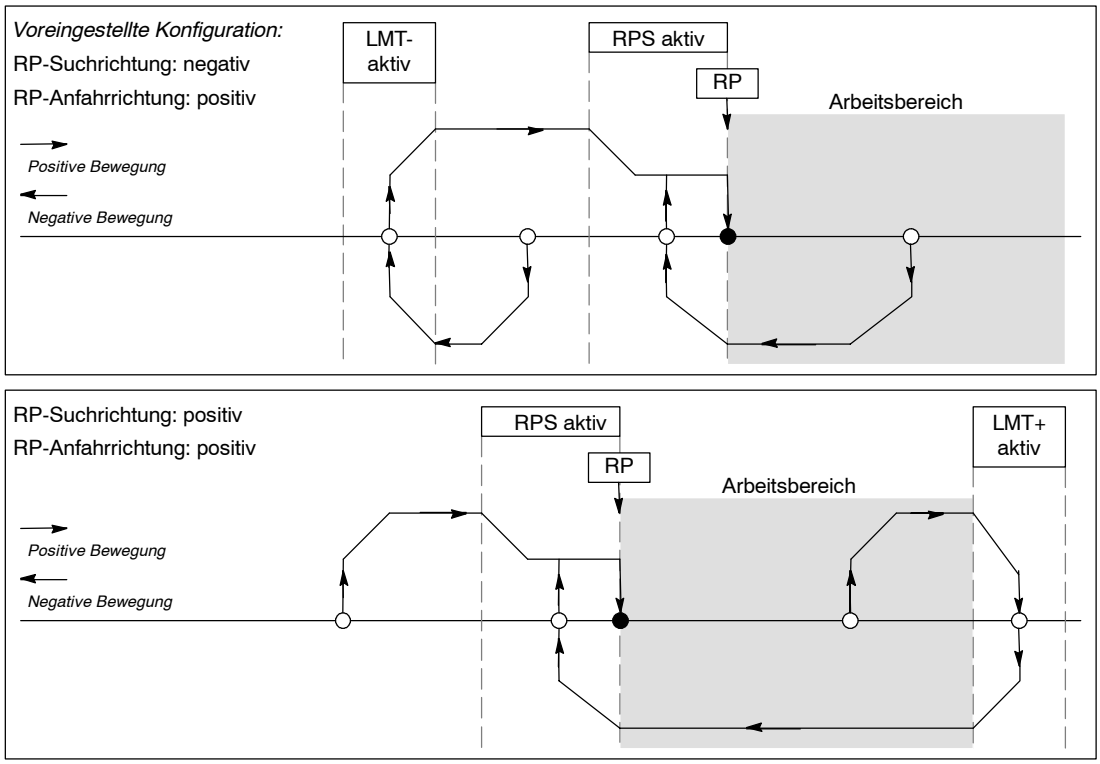


Bild 9-14 RP-Suche: Modus 1

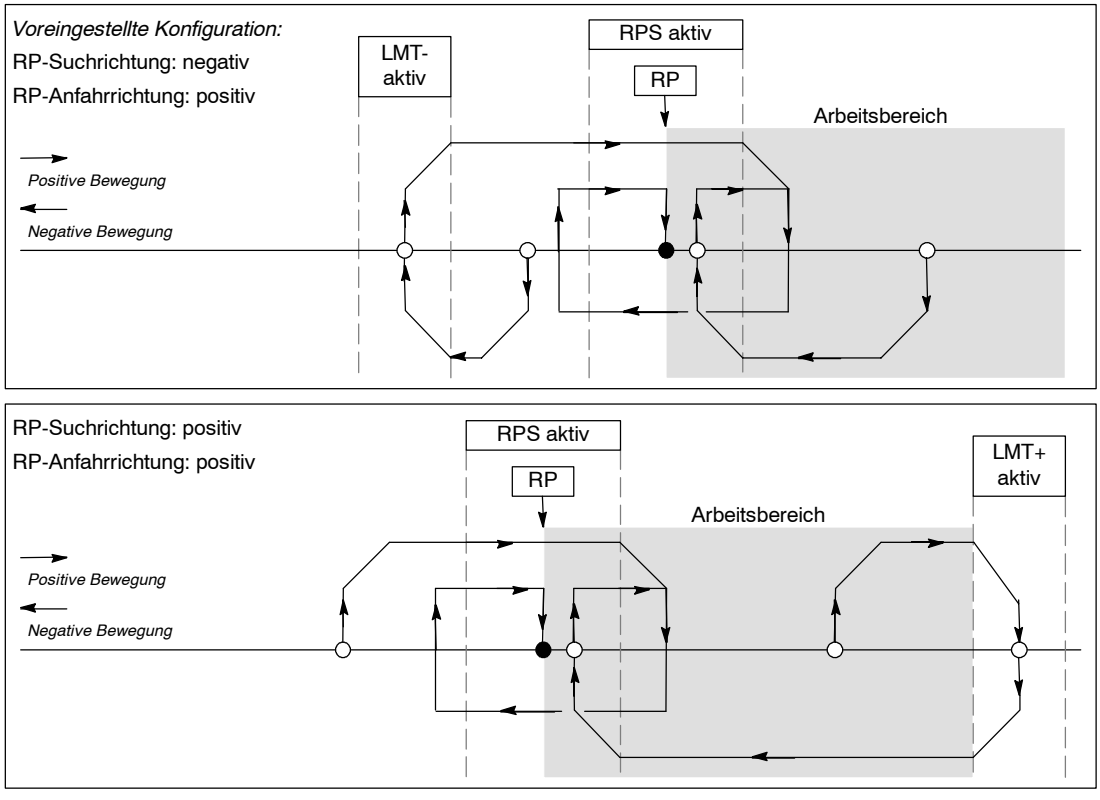


Bild 9-15 RP-Suche: Modus 2

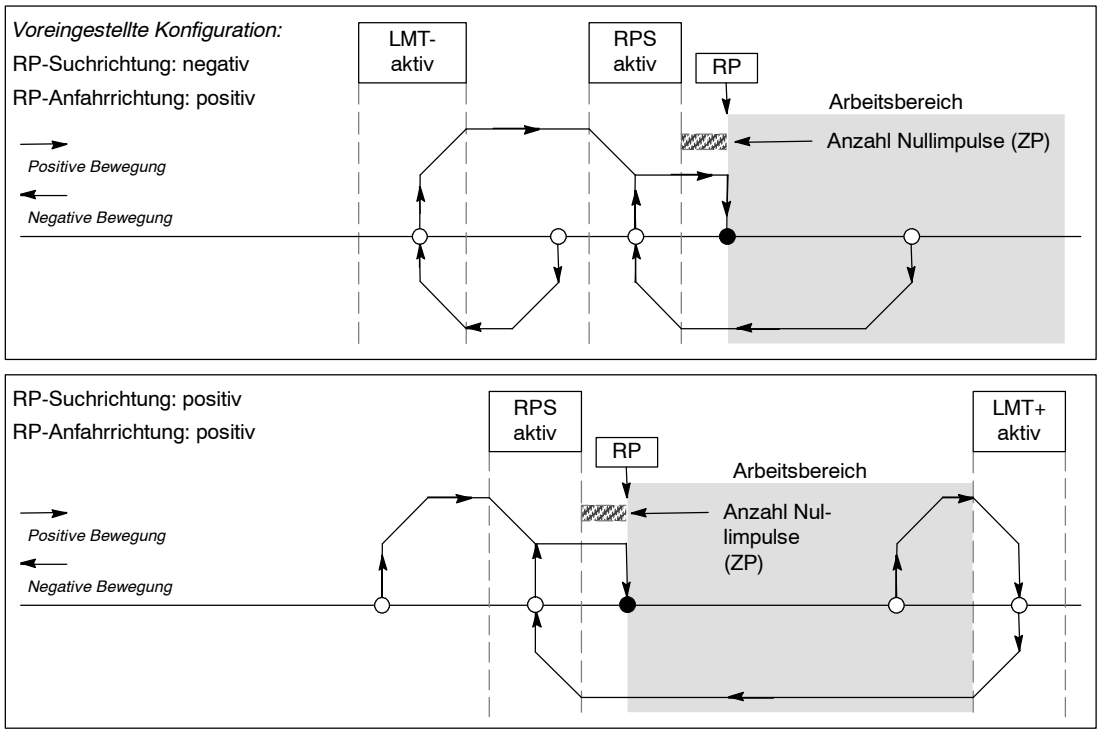


Bild 9-16 RP-Suche: Modus 3

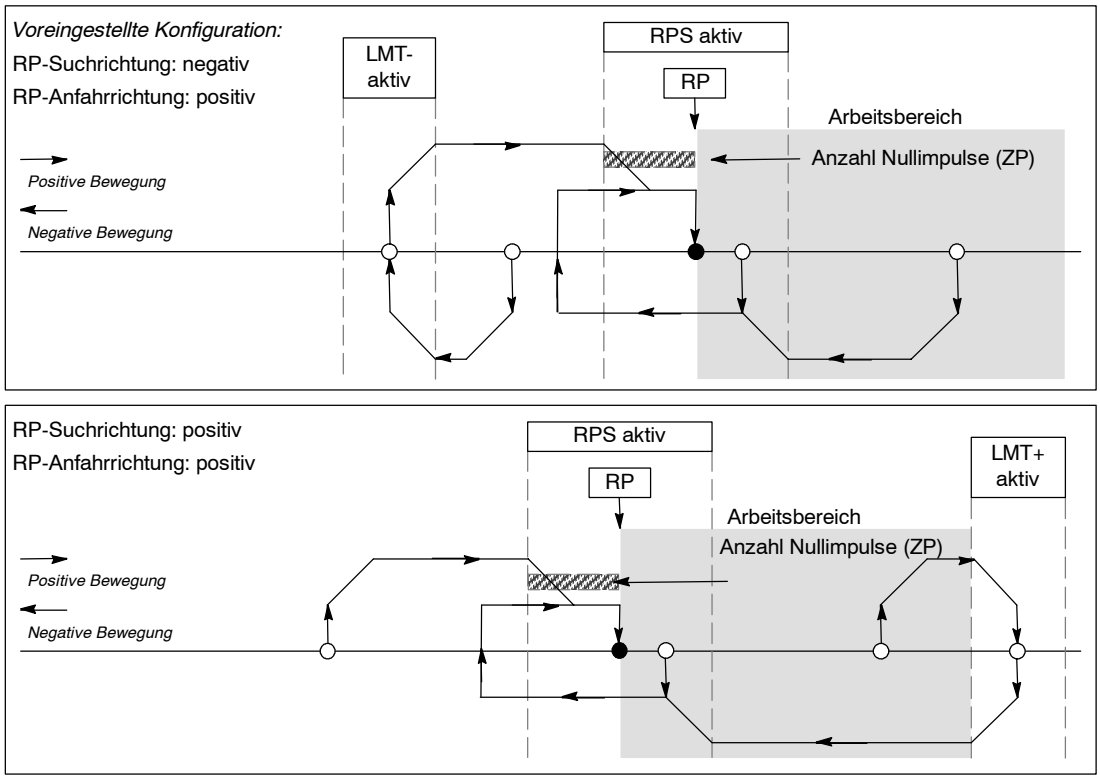


Bild 9-17 RP-Suche: Modus 4

Wählen der Lage des Arbeitsbereichs zur Spielbeseitigung

Bild 9-18 zeigt den Arbeitsbereich in Beziehung zum Referenzpunkt (RP), den Bereich RPS aktiv und die Endschalter (LMT+ und LMT-) für eine Anfahrrichtung, bei der das Spiel beseitigt wird. Im zweiten Teil der Abbildung ist der Arbeitsbereich so angeordnet, dass das Spiel nicht beseitigt wird. Bild 9-18 zeigt den RP-Suchmodus 3. Eine ähnliche Anordnung des Arbeitsbereichs ist für jede Suchsequenz der anderen RP-Suchmodi möglich, wenn auch nicht ratsam.

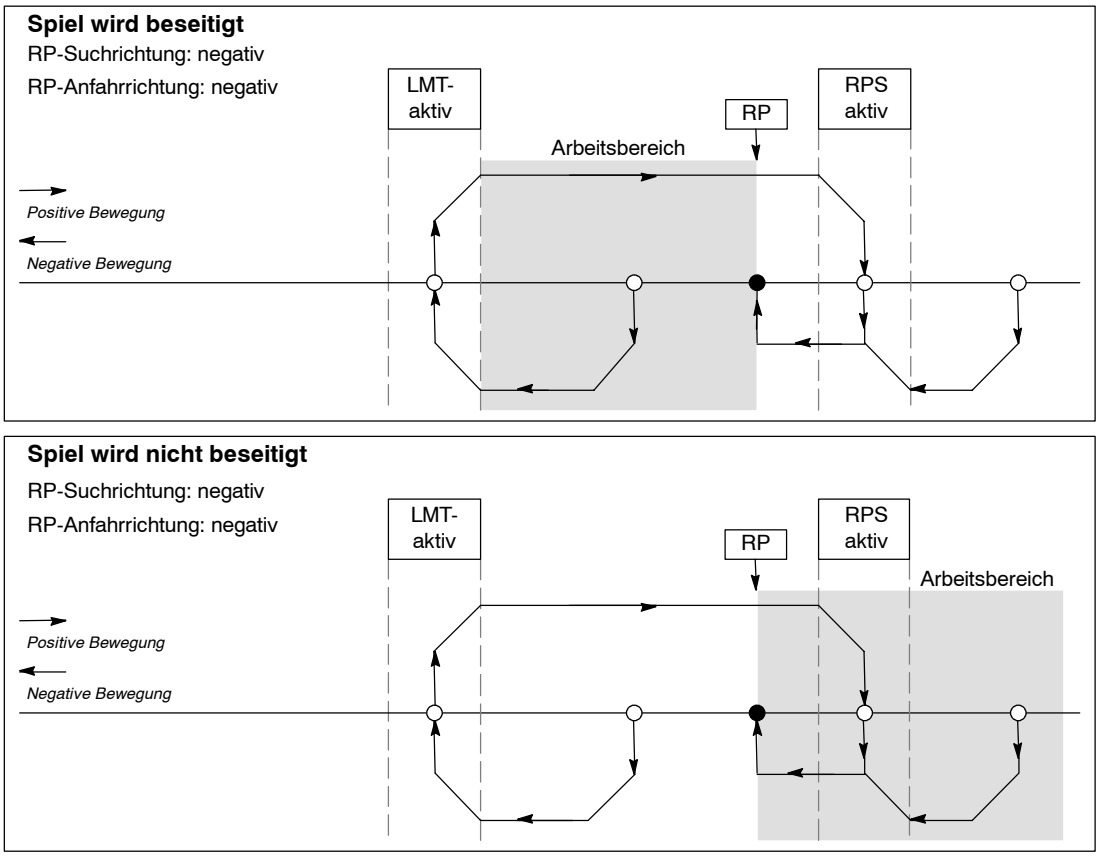


Bild 9-18 Lage des Arbeitsbereichs mit und ohne Spielbeseitigung

Vom Positionier-Assistenten erzeugte Positionieroperationen

Mit dem Positionier-Assistenten können Sie das Positioniermodul auf einfache Weise steuern, indem Sie eindeutige Unterprogramme erstellen, die auf der von Ihnen eingestellten Position des Moduls und auf den von Ihnen gewählten Konfigurationsoptionen beruhen. Alle Positionieroperationen haben das Präfix "POS*x*_", wobei das *x* die Position des Moduls angibt. Weil es sich bei jeder Positionieroperation um ein Unterprogramm handelt, gibt es 11 Positionieroperationen mit 11 Unterprogrammen.

Tipp

Die Positionieroperationen erhöhen den Speicherbedarf für Ihr Programm um bis zu 1700 Bytes. Sie können nicht verwendete Positionieroperationen löschen, um den Speicherbedarf zu verringern. Wenn Sie eine gelöschte Positionieroperation wiederherstellen möchten, führen Sie einfach den Positionier-Assistenten erneut aus.

Richtlinien für den Einsatz von Positionieroperationen

Sie müssen darauf achten, dass immer nur eine Positionieroperation zur Zeit aktiv ist.

Sie können die Operationen POSx_RUN und POSx_GOTO in einem Interruptprogramm ausführen. Es ist jedoch äußerst wichtig, dass Sie keine Operation in einem Interruptprogramm starten, wenn das Modul gerade einen anderen Befehl bearbeitet. Wenn Sie eine Operation in einem Interruptprogramm starten, können Sie mit Hilfe der Ausgänge der Operation POSx_CTRL beobachten, wann das Positioniermodul die Bewegung beendet hat.

Der Positionier-Assistent konfiguriert die Werte für die Drehzahlparameter (Speed und C_Speed) und die Positionsparameter (Pos oder C_Pos) automatisch entsprechend dem von Ihnen gewählten Messsystem. Bei Impulsen sind diese Parameter Werte vom Typ DINT. Bei physikalischen Maßeinheiten sind die Parameter Werte vom Typ REAL in der von Ihnen eingestellten Einheit. Beispiel: Wenn Sie Zentimeter (cm) einstellen, werden die Positionsparameter als Werte vom Typ REAL in Zentimetern gespeichert und die Drehzahlparameter werden als Werte vom Typ REAL in Zentimetern pro Sekunde (cm/s) gespeichert.

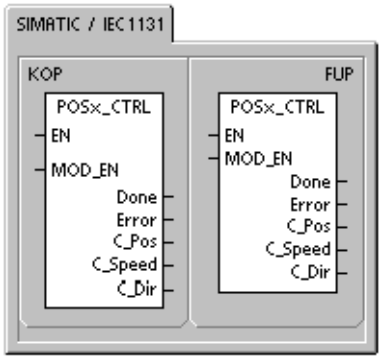
Die folgenden Positionieroperationen werden für bestimmte Positionieraufgaben benötigt:

- Fügen Sie die Operation POSx_CTRL in Ihr Programm ein und führen Sie sie mit dem Kontakt SM0.0 in jedem Zyklus aus.
- Wenn Sie Bewegung zu einer absoluten Position angeben möchten, müssen Sie zunächst mit einer Operation POSx_RSEEK oder POSx_LDPOS die Nulllage festlegen.
- Zum Anfahren einer bestimmten Lage entsprechend den Eingängen Ihres Programms verwenden Sie die Operation POSx_GOTO.
- Zum Ausführen der mit dem Positionier-Assistenten konfigurierten Bewegungsprofile verwenden Sie die Operation POSx_RUN.

Die anderen Positionieroperationen sind optional.

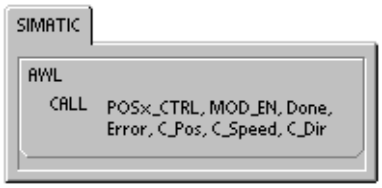
Operation POSx_CTRL

Die Operation POSx_CTRL (Steuerung) aktiviert und initialisiert das Positioniermodul, indem es dem Positioniermodul jedesmal, wenn die S7-200 in den Betriebszustand RUN wechselt, automatisch befiehlt, die Konfigurations-/Profiltabelle zu laden.



Verwenden Sie diese Operation in Ihrem Projekt nur einmal und achten Sie darauf, dass das Programm diese Operation in jedem Zyklus aufruft. Verwenden Sie SM0.0 (Ständig ein) als Eingang für den Parameter MOD_EN.

Der Parameter MOD_EN muss eingeschaltet sein, damit die anderen Positionieroperationen Befehle an das Positioniermodul senden können. Wird der Parameter MOD_EN ausgeschaltet, bricht das Positioniermodul alle in Bearbeitung befindlichen Befehle ab.



Die Ausgangsparameter der Operation POSx_CTRL liefern den aktuellen Status des Positioniermoduls.

Der Parameter Done wird eingeschaltet, wenn das Positioniermodul eine Operation beendet.

Der Parameter Error enthält das Ergebnis dieser Operation. In Tabelle 9-13 auf Seite 286 werden die Fehlercodes definiert.

Der Parameter C_Pos gibt die aktuelle Position des Moduls an. Je nach den Maßeinheiten handelt es sich entweder um eine Anzahl an Impulsen (DINT) oder um die Anzahl der physikalischen Maßeinheiten (REAL).

Der Parameter C_Speed liefert die aktuelle Drehzahl des Moduls. Wenn Sie für das Messsystem des Positioniermoduls Impulse eingestellt haben, ist C_Speed ein Wert vom Typ DINT, der die Anzahl der Impulse/Sekunde enthält. Wenn Sie für das Messsystem physikalische Maßeinheiten eingestellt haben, ist C_Speed ein Wert vom Typ REAL, der die ausgewählte physikalische Maßeinheit/Sekunde (REAL) enthält.

Der Parameter C_Dir gibt die aktuelle Richtung des Motors an.

Tabelle 9-2 Parameter der Operation POSx_CTRL

Eingänge/Ausgänge	Datentyp	Operanden
MOD_EN	BOOL	E, A, V, M, SM, S, T, Z, L, Signalfluss
Done, C_Dir	BOOL	E, A, V, M, SM, S, T, Z, L
Error	BYTE	EB, AB, VB, MB, SMB, SB, LB, AC, *VD, *AC, *LD
C_Pos, C_Speed	DINT, REAL	ED, AD, VD, MD, SMD, SD, LD, AC, *VD, *AC, *LD

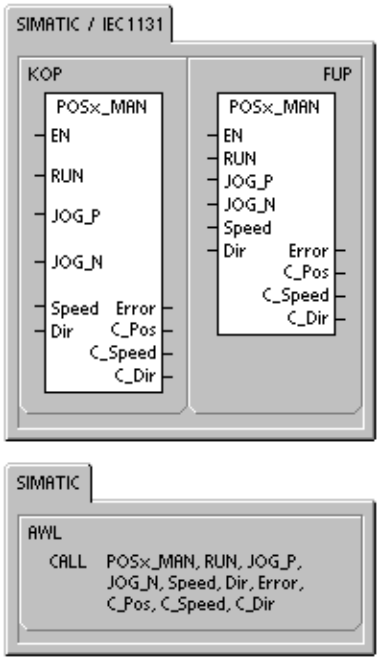
Tipp

Das Positioniermodul liest die Informationen der Konfigurations-/Profiltabelle nur beim Einschalten oder wenn es den Befehl erhält, die Konfiguration zu laden.

- Wenn Sie die Konfiguration mit dem Positionier-Assistenten ändern, befiehlt die Operation POSx_CTRL dem Positioniermodul jedesmal, wenn die S7-200 CPU in den Betriebszustand RUN wechselt, automatisch die Konfigurations-/Profiltabelle zu laden.
- Wenn Sie die Konfiguration im EM 253 Steuer-Panel ändern und auf die Schaltfläche ”Konfiguration aktualisieren” klicken, wird dem Positioniermodul befohlen, die neue Konfigurations-/Profiltabelle zu laden.
- Wenn Sie die Konfiguration auf eine andere Weise ändern, müssen Sie dem Positioniermodul ebenfalls befehlen, die Konfiguration neu zu laden, damit die Konfigurations-/Profiltabelle geladen wird. Ansonsten nutzt das Positioniermodul weiterhin die alte Konfigurations-/Profiltabelle.

Operation POSx_MAN

Die Operation POSx_MAN (Handbetrieb) versetzt das Positioniermodul in den Handbetrieb. Im Handbetrieb kann der Motor mit verschiedenen Drehzahlen oder im Tippbetrieb in positiver oder negativer Richtung betrieben werden. Wenn die Operation POSx_MAN aktiviert ist, sind nur die Operation POSx_CTRL und POSx_DIS zulässig.



Sie können nur einen der Eingänge RUN, JOG_P oder JOG_N zur Zeit aktivieren.

Wenn Sie den Parameter RUN (Run/Stopp) aktivieren, wird dem Positio-
niermodul befohlen, auf die angegebene Drehzahl (Parameter Speed) und in die angegebene Richtung (Parameter Dir) zu beschleunigen. Sie können den Wert des Parameters Speed bei laufendem Motor ändern, doch der Parameter Dir muss konstant bleiben. Wenn Sie den Parameter RUN deaktivieren, wird dem Positioniermodul befohlen, den Motor bis zum Stillstand zu verzögern.

Wenn Sie den Parameter JOG_P (Tippbetrieb positive Drehung) oder den Parameter JOG_N (Tippbetrieb negative Drehung) aktivieren, wird dem Positioniermodul befohlen, in positive oder negative Richtung vorzurücken. Bleibt einer der Parameter JOG_P oder JOG_N weniger als 0,5 Sekunden aktiviert, gibt das Positioniermodul Impulse aus, um die in JOG_INCREMENT angegebene Entfernung zu fahren. Bleibt einer der Parameter JOG_P oder JOG_N 0,5 Sekunden oder länger aktiviert, beschleunigt das Positioniermodul auf die von JOG_SPEED angegeben Drehzahl.

Der Parameter Speed gibt die Drehzahl an, wenn RUN aktiviert ist. Wenn Sie für das Messsystem des Positioniermoduls Impulse eingestellt haben, ist die Drehzahl ein Wert vom Typ DINT mit der Anzahl der Impulse/Sekunde. Wenn Sie für das Messsystem des Positioniermoduls physikalische Einheiten eingestellt haben, ist die Drehzahl ein Wert vom Typ REAL mit den Einheiten/Sekunde. Sie können diesen Parameter bei laufendem Motor ändern.

Tipp
Das Positioniermodul reagiert möglicherweise nicht auf geringfügige Änderungen des Parameters Speed, insbesondere dann nicht, wenn die eingerichtete Beschleunigungs- oder Verzögerungszeit kurz ist und der Unter-
schied zwischen der eingerichteten Höchstdrehzahl und der Start-/Stoppdrehzahl groß ist.

Der Parameter Dir gibt die Bewegungsrichtung an, wenn RUN aktiviert ist. Sie können diesen Wert nicht ändern, wenn der Parameter RUN aktiviert ist.

Der Parameter Error enthält das Ergebnis dieser Operation. In Tabelle 9-13 auf Seite 286 werden die Fehlercodes definiert.

Der Parameter C_Pos gibt die aktuelle Position des Moduls an. Je nach den eingestellten Maßeinheiten handelt es sich entweder um eine Anzahl an Impulsen (DINT) oder um die Anzahl der physikalischen Maßeinheiten (REAL).

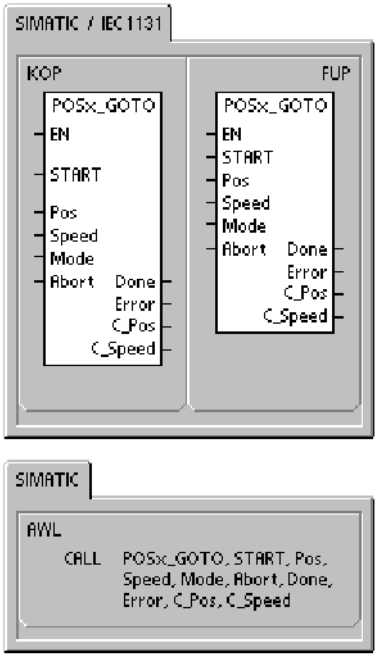
Der Parameter C_Speed liefert die aktuelle Drehzahl des Moduls. Je nach den eingestellten Maßeinheiten handelt es sich entweder um eine Anzahl an Impulsen/Sekunde (DINT) oder um die Anzahl der physikalischen Maßeinheiten/Sekunde (REAL).

Der Parameter C_Dir gibt die aktuelle Richtung des Motors an.

Tabelle 9-3 Parameter der Operation POSx_MAN

Eingänge/Ausgänge	Datentyp	Operanden
RUN, JOG_P, JOG_N	BOOL	E, A, V, M, SM, S, T, Z, L, Signalfluss
Speed	DINT, REAL	ED, AD, VD, MD, SMD, SD, LD, AC, *VD, *AC, *LD, Konstante
Dir, C_Dir	BOOL	E, A, V, M, SM, S, T, Z, L
Error	BYTE	EB, AB, VB, MB, SMB, SB, LB, AC, *VD, *AC, *LD
C_Pos, C_Speed	DINT, REAL	ED, AD, VD, MD, SMD, SD, LD, AC, *VD, *AC, *LD

Operation POSx_GOTO



Die Operation POSx_GOTO befiehlt dem Positioniermodul, in eine gewünschte Lage zu gehen.

Wenn das Bit EN eingeschaltet wird, wird die Operation aktiviert. Stellen Sie sicher, dass das Bit EN eingeschaltet bleibt, bis das Bit Done anzeigt, dass die Ausführung der Operation beendet ist.

Wenn der Parameter START eingeschaltet wird, wird der Befehl GOTO an das Positioniermodul gesendet. In jedem Zyklus, in dem der Parameter START eingeschaltet ist und das Positioniermodul nicht gerade belegt ist, sendet die Operation einen Befehl GOTO an das Positioniermodul. Um sicherzustellen, dass nur ein Befehl GOTO gesendet wird, müssen Sie den Parameter START über Flankenerkennung einschalten.

Der Parameter Pos enthält einen Wert, der entweder die Lage angibt, die angefahren werden soll (bei einer absoluten Bewegung) oder der die Entfernung angibt, die zurückgelegt werden soll (bei einer relativen Bewegung). Je nach den gewählten Maßeinheiten handelt es sich entweder um eine Anzahl an Impulsen (DINT) oder um die physikalischen Maßeinheiten (REAL).

Der Parameter Speed gibt die maximale Drehzahl für diese Bewegung an. Je nach den Maßeinheiten handelt es sich entweder um eine Anzahl an Impulsen/Sekunde (DINT) oder um die Anzahl der physikalischen Maßeinheiten/Sekunde (REAL).

Der Parameter Mode wählt die Bewegungsart aus:
- 0 - Absolute Position
- 1 - Relative Position
- 2 - Kontinuierlicher positiver Lauf mit fester Drehzahl
- 3 - Kontinuierlicher negativer Lauf mit fester Drehzahl

Der Parameter Done wird eingeschaltet, wenn das Positioniermodul diese Operation beendet.

Wenn der Parameter Abort eingeschaltet wird, wird dem Positioniermodul befohlen, das aktuelle Profil zu beenden und den Motor bis zum Stillstand zu verzögern.

Der Parameter Error enthält das Ergebnis dieser Operation. In Tabelle 9-13 auf Seite 286 werden die Fehlercodes definiert.

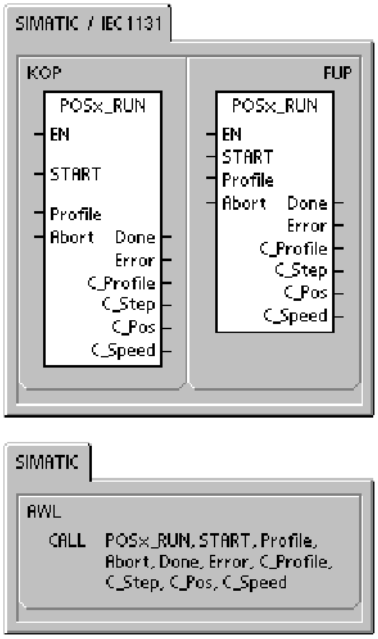
Der Parameter C_Pos gibt die aktuelle Position des Moduls an. Je nach den Maßeinheiten handelt es sich entweder um eine Anzahl an Impulsen (DINT) oder um die Anzahl der physikalischen Maßeinheiten (REAL).

Der Parameter C_Speed liefert die aktuelle Drehzahl des Moduls. Je nach den Maßeinheiten handelt es sich entweder um eine Anzahl an Impulsen/Sekunde (DINT) oder um die Anzahl der physikalischen Maßeinheiten/Sekunde (REAL).

Tabelle 9-4 Parameter der Operation POSx_GOTO

Eingänge/Ausgänge	Datentyp	Operanden
START	BOOL	E, A, V, M, SM, S, T, Z, L, Signalfluss
Pos, Speed	DINT, REAL	ED, AD, VD, MD, SMD, SD, LD, AC, *VD, *AC, *LD, Konstante
Mode	BYTE	EB, AB, VB, MB, SMB, SB, LB, AC, *VD, *AC, *LD, Konstante
Abort, Done	BOOL	E, A, V, M, SM, S, T, Z, L
Error	BYTE	EB, AB, VB, MB, SMB, SB, LB, AC, *VD, *AC, *LD
C_Pos, C_Speed	DINT, REAL	ED, AD, VD, MD, SMD, SD, LD, AC, *VD, *AC, *LD

Operation POSx_RUN



Die Operation POSx_RUN (Profil ausführen) befiehlt dem Positioniermodul, die Bewegung in einem bestimmten Profil auszuführen, das in der Konfigurations-/Profiltabelle gespeichert ist.

Wenn das Bit EN eingeschaltet wird, wird die Operation aktiviert. Stellen Sie sicher, dass das Bit EN eingeschaltet bleibt, bis das Bit Done anzeigt, dass die Ausführung der Operation beendet ist.

Wenn der Parameter START eingeschaltet wird, wird der Befehl RUN an das Positioniermodul gesendet. In jedem Zyklus, in dem der Parameter START eingeschaltet ist und das Positioniermodul nicht gerade belegt ist, sendet die Operation einen Befehl RUN an das Positioniermodul. Um
sicherzustellen, dass nur ein Befehl RUN gesendet wird, müssen Sie den Parameter START über Flankenerkennung einschalten.

Der Parameter Profile enthält die Nummer oder den symbolischen Namen des Bewegungsprofils. Sie können auch die erweiterten Positionierbefehle (118 bis 127) wählen. Weitere Informationen zu den Positionierbefehlen finden Sie in Tabelle 9-19 auf Seite 294.

Wenn der Parameter Abort eingeschaltet wird, wird dem Positioniermodul befohlen, das aktuelle Profil zu beenden und den Motor bis zum Stillstand zu verzögern.

Der Parameter Done wird eingeschaltet, wenn das Positioniermodul diese Operation beendet.

Der Parameter Error enthält das Ergebnis dieser Operation. In Tabelle 9-13 auf Seite 286 werden die Fehlercodes definiert.

Der Parameter C_Profile enthält das Profil, das gerade vom Positioniermodul ausgeführt wird.

Der Parameter C_Step enthält den Schritt des Profils, der gerade ausgeführt wird.

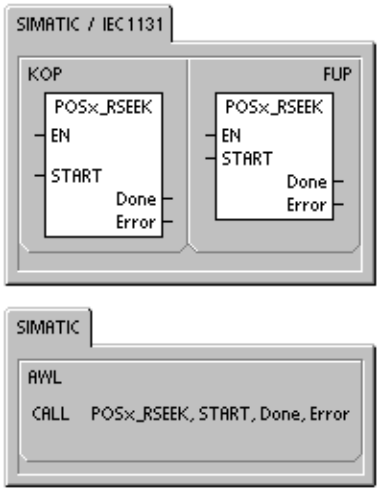
Der Parameter C_Pos gibt die aktuelle Position des Moduls an. Je nach den Maßeinheiten handelt es sich entweder um eine Anzahl an Impulsen (DINT) oder um die Anzahl der physikalischen Maßeinheiten (REAL).

Der Parameter C_Speed liefert die aktuelle Drehzahl des Moduls. Je nach den Maßeinheiten handelt es sich entweder um eine Anzahl an Impulsen/Sekunde (DINT) oder um die Anzahl der physikalischen Maßeinheiten/Sekunde (REAL).

Tabelle 9-5 Parameter der Operation POSx_RUN

Eingänge/Ausgänge	Datentyp	Operanden
START	BOOL	E, A, V, M, SM, S, T, Z, L, Signalfluss
Profile	BYTE	EB, AB, VB, MB, SMB, SB, LB, AC, *VD, *AC, *LD, Konstante
Abort, Done	BOOL	E, A, V, M, SM, S, T, Z, L
Error, C_Profile, C_Step	BYTE	EB, AB, VB, MB, SMB, SB, LB, AC, *VD, *AC, *LD
C_Pos, C_Speed	DINT, REAL	ED, AD, VD, MD, SMD, SD, LD, AC, *VD, *AC, *LD

Operation POSx_RSEEK



Die Operation POSx_RSEEK (Referenzpunktlage suchen) startet eine Referenzpunktsuche nach der Suchmethode, die in der Konfigurations-/Profiltabelle angegeben ist. Wenn das Positioniermodul den Referenzpunkt ermittelt und die Bewegung zum Stillstand gebracht hat, lädt das Positioniermodul den Parameterwert RP_OFFSET in die aktuelle Position und erzeugt einen Impuls von 50 ms am Ausgang CLR.

Der voreingestellte Wert für RP_OFFSET ist 0. Sie können im Positionier-Assistenten, im EM 253 Steuer-Panel oder mit der Operation POSx_LDOFF (Versatz laden) den Wert von RP_OFFSET ändern.

Wenn das Bit EN eingeschaltet wird, wird die Operation aktiviert. Stellen Sie sicher, dass das Bit EN eingeschaltet bleibt, bis das Bit Done anzeigt, dass die Ausführung der Operation beendet ist.

Wenn der Parameter START eingeschaltet wird, wird der Befehl RSEEK an das Positioniermodul gesendet. In jedem Zyklus, in dem der Parameter START eingeschaltet ist und das Positioniermodul nicht gerade belegt ist, sendet die Operation einen Befehl RSEEK an das Positioniermodul. Um sicherzustellen, dass nur ein Befehl RUN gesendet wird, müssen Sie den Parameter START über Flankenerkennung einschalten.

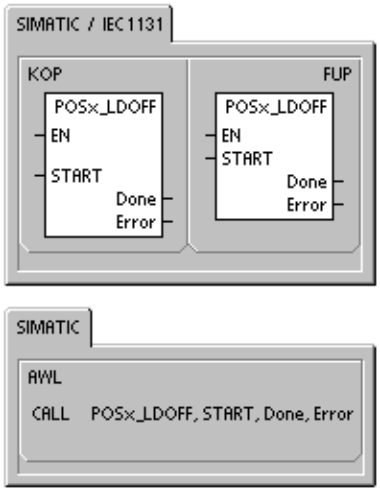
Der Parameter Done wird eingeschaltet, wenn das Positioniermodul diese Operation beendet.

Der Parameter Error enthält das Ergebnis dieser Operation. In Tabelle 9-13 auf Seite 286 werden die Fehlercodes definiert.

Tabelle 9-6 Parameter der Operation POSx_RSEEK

Eingänge/Ausgänge	Datentyp	Operanden
START	BOOL	E, A, V, M, SM, S, T, Z, L, Signalfluss
Done	BOOL	E, A, V, M, SM, S, T, Z, L
Error	BYTE	EB, AB, VB, MB, SMB, SB, LB, AC, *VD, *AC, *LD

Operation POSx_LDOFF



Die Operation POSx_LDOFF (Referenzpunktversatz laden) richtet eine neue Nulllage ein, die sich an einer anderen Stelle befindet als der Refe-
renzpunkt.

Vor der Ausführung dieser Operation müssen Sie zunächst die Lage des Referenzpunkts ermitteln. Sie müssen auch die Maschine in die Ausgangs-
position fahren. Wenn die Operation den Befehl LDOFF sendet, berechnet das Positioniermodul den Versatz zwischen der Ausgangsposition (der aktuellen Position) und der Lage des Referenzpunkts. Das Positioniermodul speichert dann den berechneten Versatz zum Parameter RP_OFFSET und setzt die aktuelle Position auf 0. Dadurch wird die Nulllage als Ausgangs-
position eingerichtet.

In dem Fall, dass der Motor seine Lage ”vergisst” (z.B. bei Spannungsausfall oder wenn der Motor manuell neu positioniert wird), können Sie mit der Operation POSx_RSEEK die Nulllage automatisch neu einrichten.

Wenn das Bit EN eingeschaltet wird, wird die Operation aktiviert. Stellen Sie sicher, dass das Bit EN eingeschaltet bleibt, bis das Bit Done anzeigt, dass die Ausführung der Operation beendet ist.

Wenn der Parameter START eingeschaltet wird, wird der Befehl LDOFF an das Positioniermodul gesendet. In jedem Zyklus, in dem der Parameter START eingeschaltet ist und das Positioniermodul nicht gerade belegt ist, sendet die Operation einen Befehl LDOFF an das Positioniermodul. Um sicherzustellen, dass nur ein Befehl RUN gesendet wird, müssen Sie den Parameter START über Flankenerkennung einschalten.

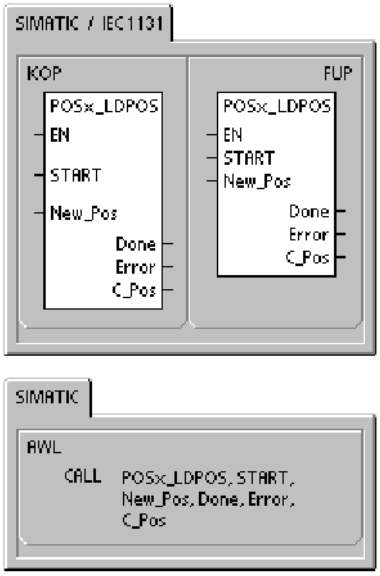
Der Parameter Done wird eingeschaltet, wenn das Positioniermodul diese Operation beendet.

Der Parameter Error enthält das Ergebnis dieser Operation. In Tabelle 9-13 auf Seite 286 werden die Fehlercodes definiert.

Tabelle 9-7 Parameter der Operation POSx_LDOFF

Eingänge/Ausgänge	Datentyp	Operanden
START	BOOL	E, A, V, M, SM, S, T, Z, L, Signalfluss
Done	BOOL	E, A, V, M, SM, S, T, Z, L
Error	BYTE	EB, AB, VB, MB, SMB, SB, LB, AC, *VD, *AC, *LD

Operation POSx_LDPOS



Die Operation POSx_LDPOS (Position laden) ändert den aktuellen Positionswert im Positioniermodul in einen neuen Wert. Sie können mit dieser Operation auch eine neue Nulllage für einen absoluten Fahrbefehl einrichten.

Wenn das Bit EN eingeschaltet wird, wird die Operation aktiviert. Stellen Sie sicher, dass das Bit EN eingeschaltet bleibt, bis das Bit Done anzeigt, dass die Ausführung der Operation beendet ist.

Wenn der Parameter START eingeschaltet wird, wird der Befehl LDPOS an das Positioniermodul gesendet. In jedem Zyklus, in dem der Parameter START eingeschaltet ist und das Positioniermodul nicht gerade belegt ist, sendet die Operation einen Befehl LDPOS an das Positioniermodul. Um sicherzustellen, dass nur ein Befehl RUN gesendet wird, müssen Sie den Parameter START über Flankenerkennung einschalten.

Der Parameter New_Pos liefert den neuen Wert, durch den der aktuelle Positionswert, der vom Positioniermodul gemeldet und für absolute Bewegungen verwendet wird, ersetzt werden soll. Je nach den Maßeinheiten handelt es sich entweder um eine Anzahl an Impulsen (DINT) oder um die physikalischen Maßeinheiten (REAL).

Der Parameter Done wird eingeschaltet, wenn das Positioniermodul diese Operation beendet.

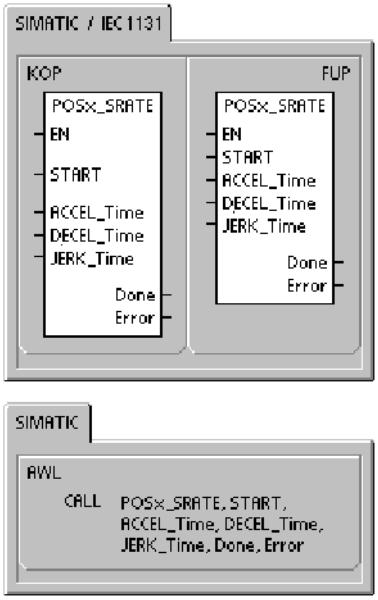
Der Parameter Error enthält das Ergebnis dieser Operation. In Tabelle 9-13 auf Seite 286 werden die Fehlercodes definiert.

Der Parameter C_Pos gibt die aktuelle Position des Moduls an. Je nach den Maßeinheiten handelt es sich entweder um eine Anzahl an Impulsen (DINT) oder um die Anzahl der physikalischen Maßeinheiten (REAL).

Tabelle 9-8 Parameter der Operation POSx_LDPOS

Eingänge/Ausgänge	Datentyp	Operanden
START	BOOL	E, A, V, M, SM, S, T, Z, L, Signalfluss
New_Pos, C_Pos	DINT, REAL	ED, AD, VD, MD, SMD, SD, LD, AC, *VD, *AC, *LD
Done	BOOL	E, A, V, M, SM, S, T, Z, L
Error	BYTE	EB, AB, VB, MB, SMB, SB, LB, AC, *VD, *AC, *LD

Operation POSx_SRATE



Die Operation POSx_SRATE (Geschwindigkeit einstellen) befiehlt dem Positioniermodul, die Beschleunigungs-, Verzögerungs- oder Ruckausgleichszeit zu ändern.

Wenn das Bit EN eingeschaltet wird, wird die Operation aktiviert. Stellen Sie sicher, dass das Bit EN eingeschaltet bleibt, bis das Bit Done anzeigt, dass die Ausführung der Operation beendet ist.

Wenn der Parameter START eingeschaltet wird, wird der neue Zeitwert in die Konfigurations-/Profiltabelle kopiert und der Befehl SRATE an das Positioniermodul gesendet. In jedem Zyklus, in dem der Parameter START eingeschaltet ist und das Positioniermodul nicht gerade belegt ist, sendet die Operation einen Befehl SRATE an das Positioniermodul. Um sicherzustellen, dass nur ein Befehl RUN gesendet wird, müssen Sie den Parameter START über Flankenerkennung einschalten.

Die Parameter ACCEL_Time, DECEL_Time und JERK_Time geben die neue Beschleunigungszeit, Verzögerungszeit und Ruckausgleichszeit in Millisekunden (ms) an.

Der Parameter Done wird eingeschaltet, wenn das Positioniermodul diese Operation beendet.

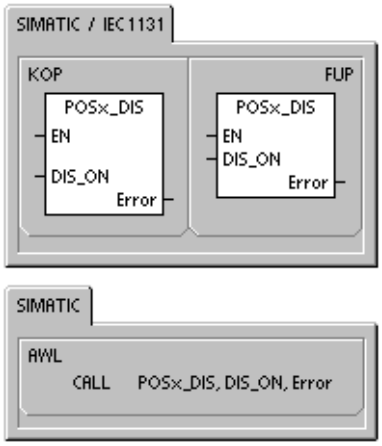
Der Parameter Error enthält das Ergebnis dieser Operation. In Tabelle 9-13 auf Seite 286 werden die Fehlercodes definiert.

Tabelle 9-9 Parameter der Operation POSx_SRATE

Eingänge/Ausgänge	Datentyp	Operanden
START	BOOL	E, A, V, M, SM, S, T, Z, L
ACCEL_Time, DECEL_Time, JERK_Time	DINT	ED, AD, VD, MD, SMD, SD, LD, AC, *VD, *AC, *LD, Konstante
Done	BOOL	E, A, V, M, SM, S, T, Z, L
Error	BYTE	EB, AB, VB, MB, SMB, SB, LB, AC, *VD, *AC, *LD

Operation POSx_DIS

Die Operation POSx_DIS schaltet den Ausgang DIS des Positioniermoduls ein oder aus. Auf diese Weise können Sie den Ausgang DIS zum Deaktivieren oder zum Aktivieren einer Motorsteuerung verwenden. Wenn Sie den Ausgang DIS am Positioniermodul verwenden, dann kann diese
Operation in jedem Zyklus aufgerufen werden oder sie kann nur dann aufgerufen werden, wenn Sie den Wert des Ausgangs DIS ändern möchten.



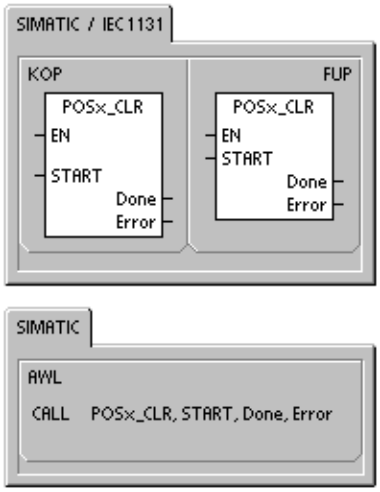
Wenn das Bit EN eingeschaltet wird, um die Operation zu aktivieren, steuert der Parameter DIS_ON den Ausgang DIS des Positioniermoduls. Ausführliche Informationen zum Ausgang DIS finden Sie in Tabelle 9-1 auf Seite 254 oder in den technischen Daten des Positioniermoduls in Anhang A.

Der Parameter Error enthält das Ergebnis dieser Operation. In Tabelle 9-13 auf Seite 286 werden die Fehlercodes definiert.

Tabelle 9-10 Parameter der Operation POSx_DIS

Eingänge/Ausgänge	Datentyp	Operanden
DIS_ON	BYTE	EB, AB, VB, MB, SMB, SB, LB, AC, *VD, *AC, *LD, Konstante
Error	BYTE	EB, AB, VB, MB, SMB, SB, LB, AC, *VD, *AC, *LD

Operation POSx_CLR



Die Operation POSx_CLR (Impuls an Ausgang CLR erzeugen) befiehlt dem Positioniermodul, einen Impuls von 50 ms an Ausgang CLR zu erzeugen.

Wenn das Bit EN eingeschaltet wird, wird die Operation aktiviert. Stellen Sie sicher, dass das Bit EN eingeschaltet bleibt, bis das Bit Done anzeigt, dass die Ausführung der Operation beendet ist.

Wenn der Parameter START eingeschaltet wird, wird der Befehl CLR an das Positioniermodul gesendet. In jedem Zyklus, in dem der Parameter START eingeschaltet ist und das Positioniermodul nicht gerade belegt ist, sendet die Operation einen Befehl CLR an das Positioniermodul. Um
sicherzustellen, dass nur ein Befehl RUN gesendet wird, müssen Sie den Parameter START über Flankenerkennung einschalten.

Der Parameter Done wird eingeschaltet, wenn das Positioniermodul diese Operation beendet.

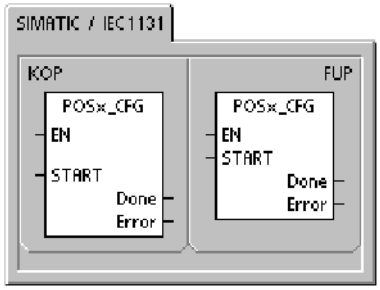
Der Parameter Error enthält das Ergebnis dieser Operation. In Tabelle 9-13 auf Seite 286 werden die Fehlercodes definiert.

Tabelle 9-11 Parameter der Operation POSx_CLR

Eingänge/Ausgänge	Datentyp	Operanden
START	BOOL	E, A, V, M, SM, S, T, Z, L, Signalfluss
Done	BOOL	E, A, V, M, SM, S, T, Z, L
Error	BYTE	EB, AB, VB, MB, SMB, SB, LB, AC, *VD, *AC, *LD

Operation POSx_CFG

Die Operation POSx_CFG (Konfiguration neu laden) befiehlt dem Positio-
niermodul, den Konfigurationsbaustein an der Adresse, die vom Pointer auf die Konfigurations-/Profiltabelle angegeben wird, zu lesen. Das Positioniermodul vergleicht dann die neue Konfiguration mit der vorhandenen Konfiguration und führt alle erforderlichen Setup-Änderungen oder Neuberechnungen durch.



Wenn das Bit EN eingeschaltet wird, wird die Operation aktiviert. Stellen Sie sicher, dass das Bit EN eingeschaltet bleibt, bis das Bit Done anzeigt, dass die Ausführung der Operation beendet ist.

Wenn der Parameter START eingeschaltet wird, wird der Befehl CFG an das Positioniermodul gesendet. In jedem Zyklus, in dem der Parameter START eingeschaltet ist und das Positioniermodul nicht gerade belegt ist, sendet die Operation einen Befehl CFG an das Positioniermodul. Um sicherzustellen, dass nur ein Befehl RUN gesendet wird, müssen Sie den Parameter START über Flankenerkennung einschalten.

Der Parameter Done wird eingeschaltet, wenn das Positioniermodul diese Operation beendet.

Der Parameter Error enthält das Ergebnis dieser Operation. In Tabelle 9-13 auf Seite 286 werden die Fehlercodes definiert.

Tabelle 9-12 Parameter der Operation POSx_CFG

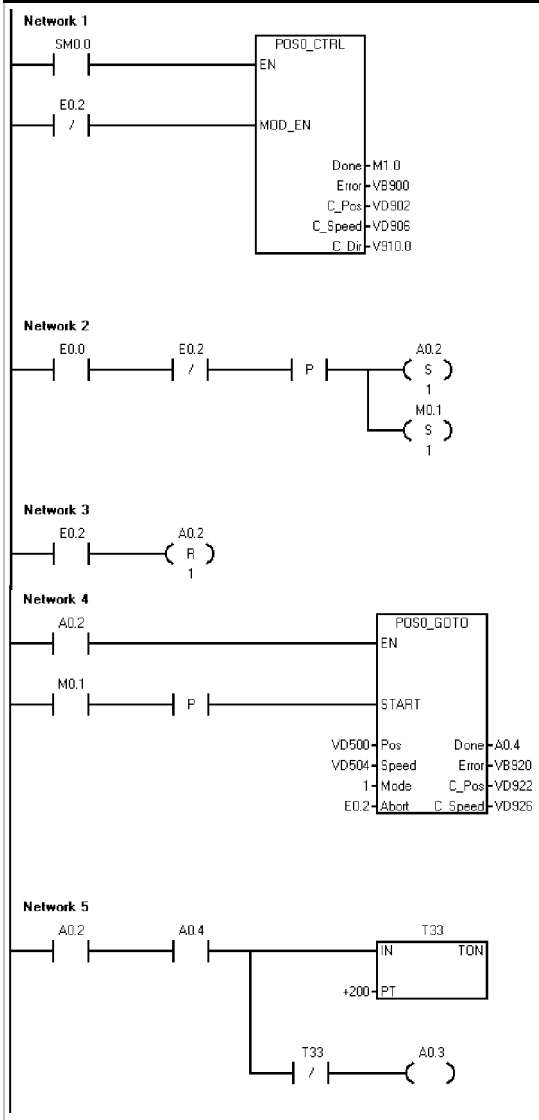
Eingänge/Ausgänge	Datentyp	Operanden
START	BOOL	E, A, V, M, SM, S, T, Z, L, Signalfluss
Done	BOOL	E, A, V, M, SM, S, T, Z, L
Error	BYTE	EB, AB, VB, MB, SMB, SB, LB, AC, *VD, *AC, *LD

Beispielprogramme für das Positioniermodul

Das erste Beispielprogramm zeigt eine einfache relative Bewegung, die mit den Operationen POSx_CTRL und POSx_GOTO Längenzuschnitte durchführt. Dieses Programm benötigt keinen RP-Suchmodus und auch kein Bewegungsprofil, und die Länge kann entweder in Impulsen oder in physikalischen Maßeinheiten gemessen werden. Geben Sie die Länge (VD500) und die Zieldrehzahl (VD504) ein. Wenn E0.0 (Start) eingeschaltet wird, läuft die Maschine an. Wenn E0.1 (Stop) eingeschaltet wird, beendet die Maschine den aktuellen Vorgang und kommt zum Stillstand. Wenn E0.2 (E_Stop) eingeschaltet wird, bricht die Maschine ab und kommt sofort zum Stillstand.

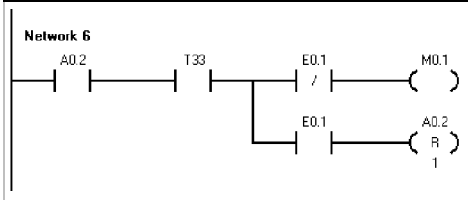
Das zweite Beispielprogramm liefert ein Beispiel für den Einsatz der Operationen POSx_CTRL, POSx_RUN, POSx_RSEEK und POSx_MAN. Sie müssen den RP-Suchmodus und ein Bewegungsprofil konfigurieren.

Beispielprogramm 1: Einfache relative Bewegung (Längenzuschnitt)



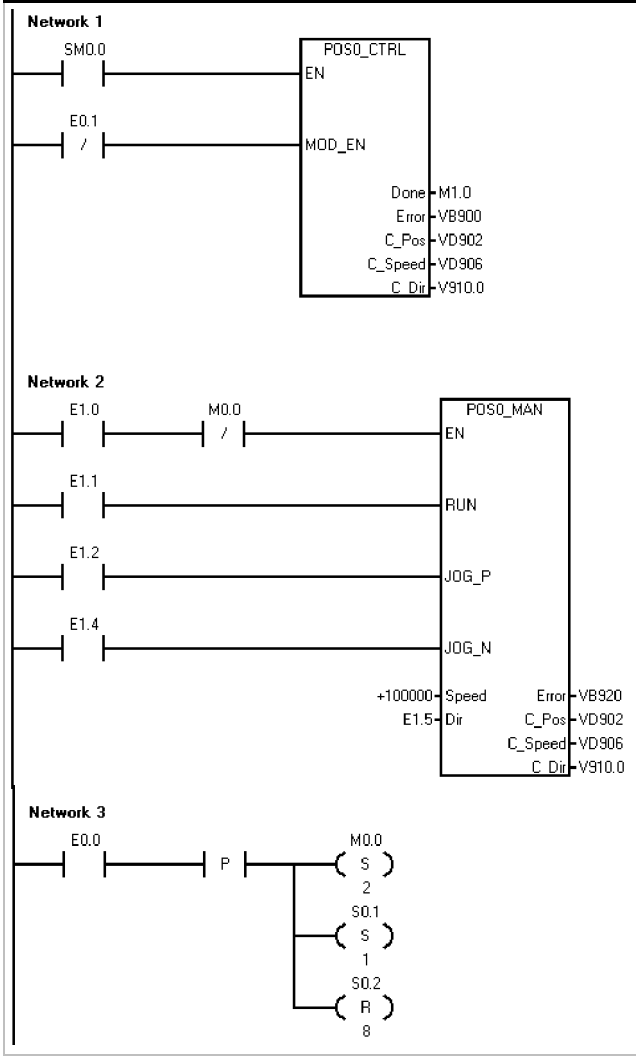

```
Netzwerk 1      //Steuerungsoperation (Modul in Steckplatz 0).
LD      SM0.0
=       L60.0
LDN     E0.2
=       L63.7
LD      L60.0
CALL    POS0_CTRL, L63.7, M1.0, VB900, VD902, VD906,
        V910.0

Netzwerk 2      //Der Startbefehl versetzt die Maschine in den
                //Automatikbetrieb.
LD      E0.0
UN      E0.2
EU
S       A0.2, 1
S       M0.1, 1

Netzwerk 3      //E_Stop: Sofortiger Stillstand und
                //Ausschalten des Automatikbetriebs.
LD      E0.2
R       A0.2, 1

Netzwerk 4      //Bewegung zu einem bestimmten Punkt:
                //Länge für Zuschnitt eingeben.
                //Zieldrehzahl in Speed eingeben.
                //Modus auf 1 setzen (relativer Modus).
LD      A0.2
=       L60.0
LD      M0.1
EU
=       L63.7
LD      L60.0
CALL    POS0_GOTO, L63.7, VD500, VD504, 1, E0.2, A0.4,
        VB920, VD922, VD926

Netzwerk 5      //Wenn Position eingenommen, Schneidwerkzeug

                //2 Sekunden lang einschalten, um den Schnitt
                //durchzuführen.
LD      A0.2
U       A0.4
TON     T33, +200
UN      T33
=       A0.3
```

Beispielprogramm 1: Einfache relative Bewegung (Längenzuschnitt), Fortsetzung



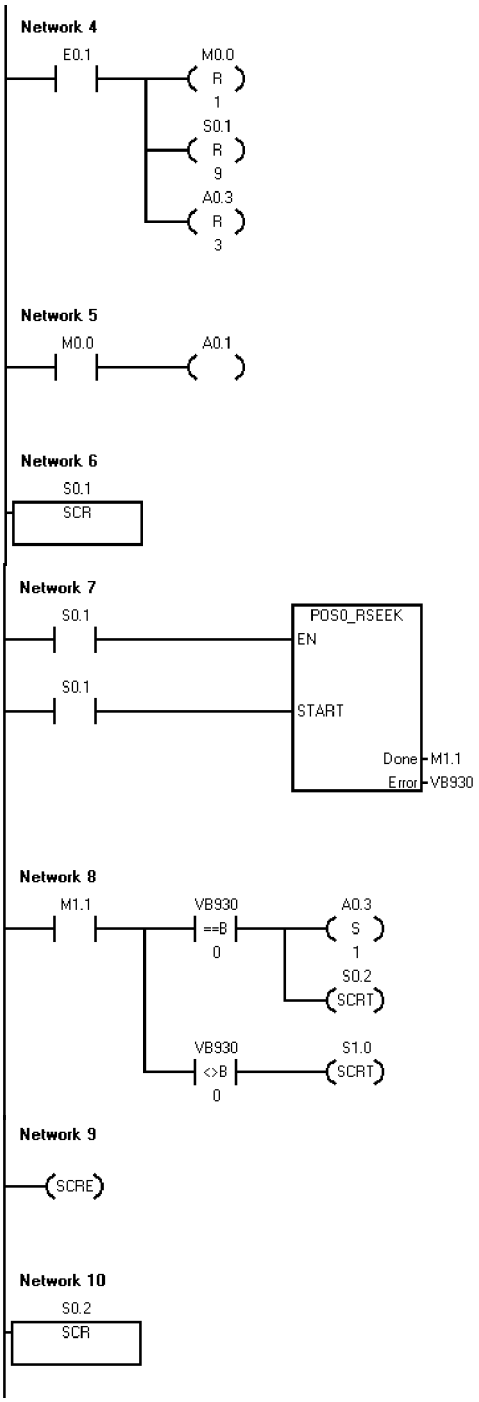

```
Netzwerk 6      //Nach dem Schnitt neu starten,
                //es sei denn, der Stoppbefehl ist aktiv.
LD      A0.2
U       T33
LPS
UN      E0.1
=       M0.1
LPP
U       E0.1
R       A0.2, 1
```

Beispielprogramm 2: Programm mit den Operationen POSx_CTRL, POSx_RUN, POSx_SEEK und POSx_MAN



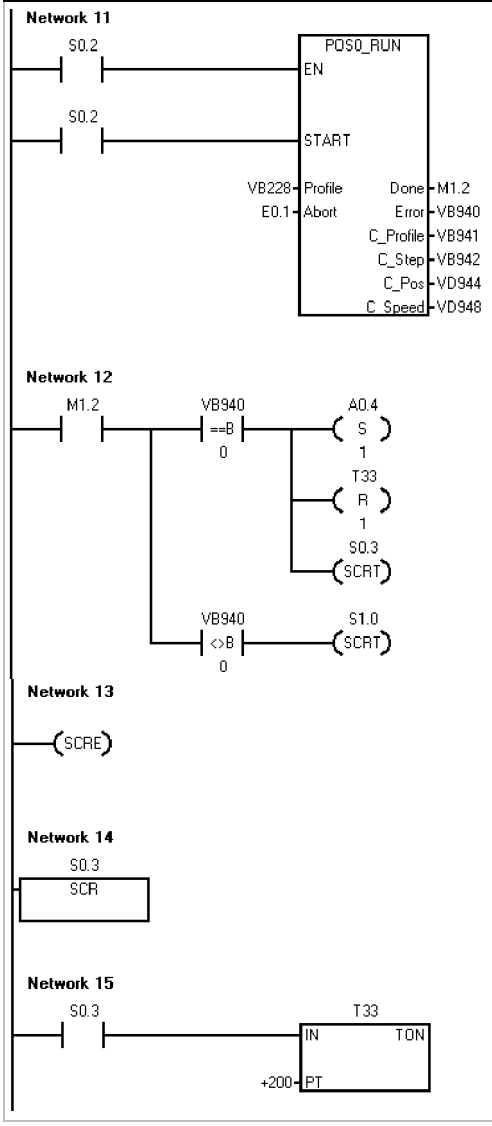

```
Netzwerk 1      //Positioniermodul aktivieren.
LD      SM0.0
=       L60.0
LDN     E0.1
=       L63.7
LD      L60.0
CALL    POS0_CTRL, L63.7, M1.0, VB900, VD902,
        VD906, V910.0

Netzwerk 2      //Handbetrieb, wenn nicht im Automatikbetrieb.
LD      E1.0
UN      M0.0
=       L60.0
LD      E1.1
=       L63.7
LD      E1.2
=       L63.6
LD      E1.4
=       L63.5
LD      L60.0
CALL    POS0_MAN, L63.7, L63.6, L63.5, +100000,
        E1.5, VB920, VD902, VD906, V910.0

Netzwerk 3      //Automatikbetrieb aktivieren.
LD      E0.0
EU
S       M0.0, 2
S       S0.1, 1
R       S0.2, 8
```

Beispielprogramm 2: Programm mit den Operationen POSx_CTRL, POSx_RUN, POSx_SEEK und POSx_MAN, Fortsetzung



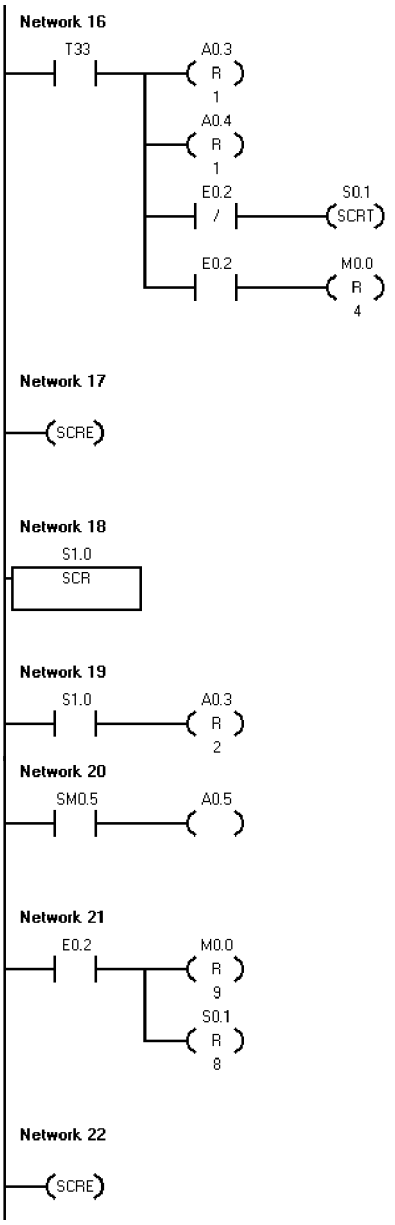

```
Netzwerk 4      //Not-AUS:
                //Modul und Automatikbetrieb deaktivieren.
LD      E0.1
R       M0.0, 1
R       S0.1, 9
R       A0.3, 3

Netzwerk 5      //Wenn im Automatikbetrieb:
                //Betriebsleuchte einschalten.
LD      M0.0
=       A0.1

Netzwerk 6
LSCR    S0.1

Netzwerk 7      //Referenzpunkt (RP) suchen.
LD      S0.1
=       L60.0
LD      S0.1
=       L63.7
LD      L60.0
CALL    POS0_RSEEK, L63.7, M1.1, VB930

Netzwerk 8      //Wenn am Referenzpunkt (RP):
                //Material einspannen und
                //zum nächsten Schritt gehen.
LD      M1.1
LPS
UB=     VB930, 0
S       A0.3, 1
SCRT    S0.2
LPP
UB<>    VB930, 0
SCRT    S1.0

Netzwerk 9
SCRE

Netzwerk 10
LSCR    S0.2
```

Beispielprogramm 2: Programm mit den Operationen POSx_CTRL, POSx_RUN, POSx_SEEK und POSx_MAN, Fortsetzung




```
Netzwerk 11    //Mit Profil 1 in Position fahren.
LD      S0.2
=       L60.0
LD      S0.2
=       L63.7
LD      L60.0
CALL    POS0_RUN, L63.7, VB228, E0.1, M1.2, VB940,
        VB941, VB942, VD944, VD948
Netzwerk 12    //Wenn positioniert, Schneidwerkzeug einschal
               //ten
               //und zum nächsten Schritt gehen.
LD      M1.2
LPS
UB=     VB940, 0
S       A0.4, 1
R       T33, 1
SCRT    S0.3
LPP
UB<>    VB940, 0
SCRT    S1.0

Netzwerk 13
SCRE

Netzwerk 14    //Auf Fertigstellung des Schnitts warten.
LSCR    S0.3

Netzwerk 15
LD      S0.3
TON     T33, +200
```

Beispielprogramm 2: Programm mit den Operationen POSx_CTRL, POSx_RUN, POSx_SEEK und POSx_MAN, Fortsetzung




```
Netzwerk 16    //Wenn STOP nicht eingeschaltet ist, nach
               //Beendigung des Schnitts neu starten.
LD     T33
LPS
R      A0.3, 1
R      A0.4, 1
UN     E0.2
SCRT   S0.1
LPP
U      E0.2
R      M0.0, 4
Netzwerk 17
SCRE

Netzwerk 18
LSCR   S1.0

Netzwerk 19    //Ausgänge zurücksetzen.
LD     S1.0
R      A0.3, 2

Netzwerk 20    //Fehlerleuchte blinken lassen.
LD     SM0.5
=      A0.5

Netzwerk 21    //Fehlerprogramm beenden, wenn STOP
               //eingeschaltet ist.
LD     E0.2
R      M0.0, 9
R      S0.1, 8

Netzwerk 22
SCRE
```

Beobachten des Positioniermoduls mit em EM 253 Steuer-Panel

Als Unterstützung für die Entwicklung Ihrer Positionierlösung gibt es in STEP 7-Micro/WIN das EM 253 Steuer-Panel. In den Registern ”Betrieb”, ”Konfiguration” und ”Diagnose” können Sie den Betrieb des Positioniermoduls während der Anlauf- und Testphasen Ihres Entwicklungsprozesses beobachten und steuern.

Im EM 253 Steuer-Panel können Sie überprüfen, dass das Positioniermodul ordnungsgemäß verdrahtet ist, Sie können die Konfigurationsdaten anpassen und die Bewegungsprofile testen.

Anzeigen und Steuern des Betriebs des Positioniermoduls

Im Register ”Betrieb” im Steuer-Panel können Sie den Betrieb des Positioniermoduls beeinflussen. Das Steuer-Panel zeigt die aktuelle Drehzahl, die aktuelle Position und die aktuelle Richtung des Positioniermoduls an. Es wird auch der Zustand der Eingangs- und Ausgangs-LEDs angezeigt (mit Ausnahme der Impuls-LEDs).

Im Steuer-Panel können Sie das Positioniermodul beeinflussen, indem Sie die Drehzahl und die Richtung ändern, indem Sie die Bewegung starten und stoppen und indem Sie das Werkzeug im Tippbetrieb betreiben (sofern die Bewegung zum Stillstand gekommen ist).

Sie können außerdem die folgenden Positionierbefehle erzeugen:

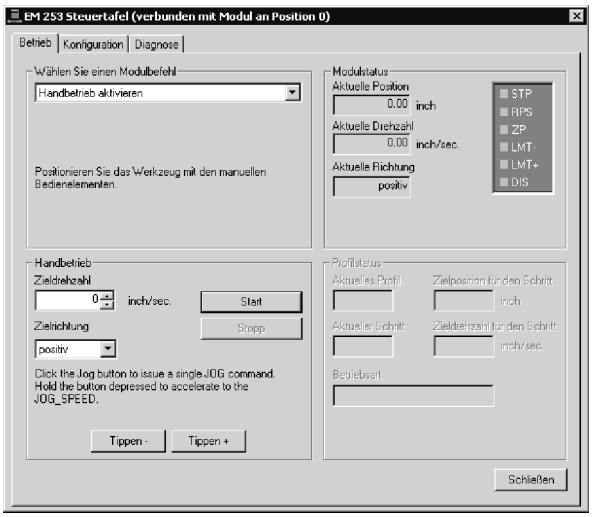


Bild 9-19 Register ”Betrieb” im EM 253 Steuer-Panel

- Handbetrieb aktivieren. Dieser Befehl ermöglicht die Positionierung des Werkzeugs mit Hilfe der manuellen Bedienelemente.
- Bewegungsprofil ausführen. Mit diesem Befehl können Sie ein Profil zur Ausführung auswählen. Das Steuer-Panel zeigt den Status des Profils an, das vom Positioniermodul ausgeführt wird.
- Referenzpunkt suchen. Dieser Befehl sucht den Referenzpunkt mit Hilfe des konfigurierten Suchmodus.
- Referenzpunktversatz laden. Nachdem Sie das Werkzeug mit den manuellen Bedienelementen in eine neue Nulllage vorgerückt haben, laden Sie den Referenzpunktversatz.
- Aktuelle Position neu laden. Dieser Befehl aktualisiert den aktuellen Positionswert und richtet eine neue Nulllage ein.
- Ausgang DIS aktivieren und Ausgang DIS deaktivieren. Diese Befehle schalten den Ausgang DIS des Positioniermoduls ein und aus.
- Impuls am Ausgang CLR. Dieser Befehl erzeugt einen Impuls von 50 ms am Ausgang CLR des Positioniermoduls.
- Bewegungsprofil einlernen. Mit diesem Befehl speichern Sie beim manuellen Positionieren des Werkzeugs die Zielposition und die Zieldrehzahl für ein Bewegungsprofil und einen Schritt. Das Steuer-Panel zeigt den Status des Profils an, das vom Positioniermodul ausgeführt wird.
- Modulkonfiguration laden. Dieser Befehl lädt eine neue Konfiguration, indem dem Positioniermodul befohlen wird, den Konfigurationsbaustein aus dem Variablenspeicher der S7-200 zu lesen.
- Absolute Position anfahren. Mit diesem Befehl fahren Sie mit einer Zieldrehzahl eine bestimmte Position an. Bevor Sie den Befehl verwenden, müssen Sie die Nulllage eingerichtet haben.
- Relativen Weg fahren. Mit diesem Befehl fahren Sie einen bestimmten Weg ab der aktuellen Position mit einer Zieldrehzahl. Sie können eine positive oder eine negative Entfernung angeben.
- Befehlsschnittstelle zurücksetzen. Dieser Befehl löscht das Befehlsbyte des Positioniermoduls und setzt das Bit Done. Verwenden Sie diesen Befehl, wenn das Positioniermodul anscheinend nicht auf Befehle reagiert.

Anzeigen und Ändern der Konfiguration des Positioniermoduls

Im Register ”Konfiguration” im Steuer-Panel können Sie sich die Konfigurationseinstellungen des Positioniermoduls, die im Datenbaustein der S7-200 gespeichert sind, ansehen und auch ändern.

Nachdem Sie die Konfigurationseinstellungen geändert haben, klicken Sie einfach auf eine Schaltfläche, um die Einstellungen im Projekt in STEP 7-Micro/Win und im Datenbaustein der S7-200 zu aktualisieren.

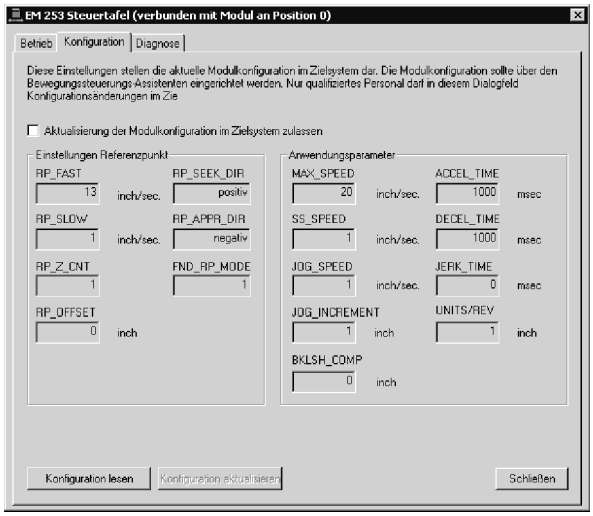


Bild 9-20 Register ”Konfiguration” im EM 253 Steuer-Panel

Anzeigen der Diagnoseinformationen für das Positioniermodul

Im Register ”Diagnose” im Steuer-Panel können Sie sich die Diagnoseinformationen zum Positioniermodul ansehen.

Sie können bestimmte Informationen zum Positioniermodul einsehen, z.B. die Position des Moduls in der E/A-Kette, den Modultyp, die Version der Firmware und das als Befehlsbyte für das Modul verwendete Ausgangsbyte.

Das Steuer-Panel zeigt alle Fehlerbedingungen an, die durch einen befohlenen Vorgang hervorgerufen wurden. Die Fehlerbedingungen der Operationen finden Sie in Tabelle 9-13 auf Seite 286.

Sie können sich außerdem die Fehlerbedingungen ansehen, die vom Positioniermodul gemeldet wurden. Die Fehlerbedingungen des Moduls finden Sie in Tabelle 9-14 auf Seite 287.

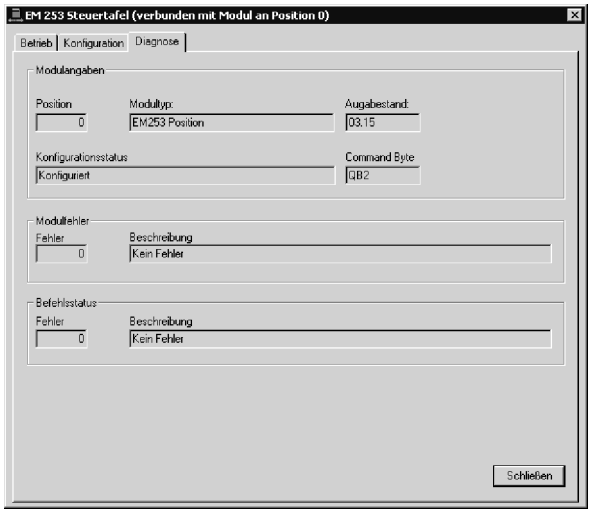


Bild 9-21 Register ”Diagnose” im EM 253 Steuer-Panel

Fehlercodes für Positioniermodul und Positionieroperationen

Tabelle 9-13 Fehlercodes der Operationen

Fehlercode	Beschreibung
0	Kein Fehler aufgetreten
1	Anwenderabbruch
2	Konfigurationsfehler Sehen Sie sich die Fehlercodes im EM 253 Steuer-Panel im Register ”Diagnose” an.
3	Unzulässiger Befehl
4	Abbruch wegen ungültiger Konfiguration Sehen Sie sich die Fehlercodes im EM 253 Steuer-Panel im Register ”Diagnose” an.
5	Abbruchen wegen fehlender Anwenderspannung
6	Abbruch wegen nicht definiertem Referenzpunkt
7	Abbruch wegen aktivem Eingang STP
8	Abbruch wegen aktivem Eingang LMT-
9	Abbruch wegen aktivem Eingang LMT+
10	Abbruch wegen Problem bei Bewegungsausführung
11	Für das angegebene Profil ist kein Profilbaustein konfiguriert
12	Unzulässige Betriebsart
13	Betriebsart für diesen Befehl nicht unterstützt
14	Unzulässige Anzahl Schritte im Profilbaustein
15	Unzulässiger Richtungswechsel
16	Unzulässige Entfernung
17	RPS vor Erreichen der Zieldrehzahl ausgelöst
18	Breite des Bereichs RPS aktiv nicht ausreichend
19	Drehzahl außerhalb des Bereichs
20	Entfernung reicht nicht aus, um gewünschte Drehzahländerung durchzuführen
21	Unzulässige Position
22	Nulllage unbekannt
23 bis 127	Reserviert
128	Das Positioniermodul kann diese Operation nicht bearbeiten: entweder ist das Positioniermodul durch die Bearbeitung einer anderen Operation belegt oder es gab für diese Operation keinen Startimpuls.
129	Positioniermodulfehler: Modulkennung falsch oder Modul abgemeldet. In SMB8 bis SMB21 (E/A-Modulkennung und Fehlerregister) finden Sie weitere Fehlerbedingungen.
130	Positioniermodul ist nicht aktiviert
131	Positioniermodul ist nicht verfügbar: Modulfehler oder Modul nicht aktiviert (siehe Status POSx_CTRL)
132	Die im Positionier-Assistenten eingerichtete Adresse im Speicherbereich der Ausgänge entspricht nicht der tatsächlichen Speicheradresse des Moduls an dieser Position.

Tabelle 9-14 Fehlercodes des Moduls

Fehlercode	Beschreibung
0	Kein Fehler aufgetreten
1	Keine Anwenderspannung
2	Konfigurationsbaustein nicht vorhanden
3	Pointerfehler Konfigurationsbaustein
4	Größe des Konfigurationsbausteins überschreitet den Variablenspeicher
5	Unzulässiges Format des Konfigurationsbausteins
6	Zu viele Profile angegeben
7	Unzulässige Angabe STP_RSP
8	Unzulässige Angabe LMT-_RPS
9	Unzulässige Angabe LMT+_RPS
10	Unzulässige Angabe FILTER_TIME
11	Unzulässige Angabe MEAS_SYS
12	Unzulässige Angabe RP_CFG
13	Unzulässiger Wert PLS/REV
14	Unzulässiger Wert UNITS/REV
15	Unzulässiger Wert RP_ZP_CNT
16	Unzulässiger Wert JOG_INCREMENT
17	Unzulässiger Wert MAX_SPEED
18	Unzulässiger Wert SS_SPD
19	Unzulässiger Wert RP_FAST
20	Unzulässiger Wert RP_SLOW
21	Unzulässiger Wert JOG_SPEED
22	Unzulässiger Wert ACCEL_TIME
23	Unzulässiger Wert DECEL_TIME
24	Unzulässiger Wert JERK_TIME
25	Unzulässiger Wert BKLSH_COMP

Für erfahrene Anwender

Beschreibung der Konfigurations-/Profiltabelle

Der Positionier-Assistent wurde entwickelt, um Positionieranwendungen zu vereinfachen, indem die Konfigurations- und Profilinformationen anhand der von Ihnen eingegebenen Antworten zu Ihrem Positioniersystem automatisch erzeugt werden. Die Informationen in der Konfigurations-/Profiltabelle dienen erfahrenen Anwendern, die ihre eigenen Unterprogramme für die Positioniersteuerung erstellen möchten.

Die Konfigurations-/Profiltabelle befindet sich im Variablenspeicher der S7-200. Wie Sie in Tabelle 9-15 sehen, werden die Konfigurationseinstellungen in den folgenden Arten von Informationen gespeichert:

- ❏ Der Konfigurationsbaustein enthält Informationen, mit denen das Modul auf die Ausführung von Positionierbefehlen vorbereitet wird.
- ❏ Der interaktive Baustein unterstützt das direkte Einrichten von Bewegungsparametern durch das Anwenderprogramm.
- ❏ Jeder Profilbaustein beschreibt eine vordefinierte Bewegung, die vom Positioniermodul ausgeführt werden soll. Sie können bis zu 25 Profilbausteine konfigurieren.

Tipp

Wenn Sie mehr als 25 Bewegungsprofile erstellen möchten, können Sie Konfigurations-/Profiltabellen austauschen, indem Sie den im Pointer auf die Konfigurations-/Profiltabelle gespeicherten Wert ändern.

Tabelle 9-15 Konfigurations-/Profiltabelle

Versatz	Name	Funktionsbeschreibung	Datentyp
Konfigurationsbaustein			
0	MOD_ID	Feld für die Modulkennung	--
5	CB_LEN	Länge des Konfigurationsbausteins in Bytes (1 Byte)	--
6	IB_LEN	Länge des interaktiven Bausteins in Bytes (1 Byte)	--
7	PF_LEN	Länge eines einzelnen Profils in Bytes (1 Byte)	--
8	STP_LEN	Länge eines einzelnen Schritts in Bytes (1 Byte)	--
9	STEPS	Anzahl der je Profil zugelassenen Schritte (1 Byte)	--
10	PROFILES	Anzahl der Profile von 0 bis 25 (1 Byte)	--
11	Reserviert	Auf 0x0000 gesetzt	--
13	IN_OUT_CFG	Gibt die Verwendung der Eingänge und Ausgänge des Moduls an (1 Byte) MSB 7 6 5 4 3 2 1 LSB 0 P/D | POL | 0 | 0 | STP | RPS | LMT- | LMT+ P/D Dieses Bit gibt die Verwendung von P0 und P1 an. Positive Polarität (POL=0): 0 - P0-Impulse für positive Drehung P1-Impulse für negative Drehung 1 - P0-Impulse für Drehung P1 steuert die Drehrichtung (0 - positiv, 1 - negativ) Negative Polarität (POL=1): 0 - P0-Impulse für positive Drehung P1-Impulse für negative Drehung 1 - P0-Impulse für Drehung P1 steuert die Drehrichtung (0 - positiv, 1 - negativ) POL Dieses Bit stellt die Polaritätskonvention für P0 und P1 ein. (0 - positive Polarität, 1 - negative Polarität) STP Dieses Bit steuert den Aktivierungspegel für den Eingang zum Stoppen. RPS Dieses Bit steuert den Aktivierungspegel für den Eingang RPS. LMT- Dieses Bit steuert den Aktivierungspegel für den Eingang der negativen Verfahrweggrenze. LMT+ Dieses Bit steuert den Aktivierungspegel für den Eingang der positiven Verfahrweggrenze. 0 - Aktiv High 1 - Aktiv Low	--

Tabelle 9-15 Konfigurations-/Profiltabelle, Fortsetzung

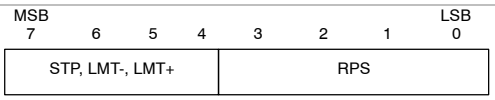
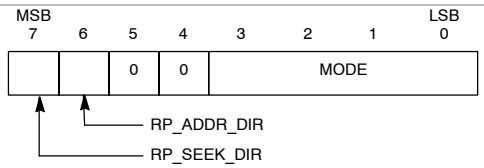
Versatz	Name	Funktionsbeschreibung	Datentyp
14	STP_RSP	Gibt die Reaktion des Antriebs auf den Eingang STP an (1 Byte). 0 Keine Aktion. Eingangsbedingung ignorieren. 1 Bis zum Stillstand verzögern und anzeigen, dass der Eingang STP aktiv ist. 2 Impulse beenden und Eingang STP anzeigen. 3 bis 255 Reserviert (Fehler, wenn angegeben).	--
15	LMT-_RSP	Gibt die Reaktion des Antriebs auf den Eingang negative Verfahrweggrenze an (1 Byte). 0 Keine Aktion. Eingangsbedingung ignorieren. 1 Bis zum Stillstand verzögern und anzeigen, dass der Grenzwert erreicht wurde. 2 Impulse beenden und anzeigen, dass der Grenzwert erreicht wurde. 3 bis 255 Reserviert (Fehler, wenn angegeben).	--
16	LMT+_RSP	Gibt die Reaktion des Antriebs auf den Eingang positive Verfahrweggrenze an (1 Byte). 0 Keine Aktion. Eingangsbedingung ignorieren. 1 Bis zum Stillstand verzögern und anzeigen, dass der Grenzwert erreicht wurde. 2 Impulse beenden und anzeigen, dass der Grenzwert erreicht wurde. 3 bis 255 Reserviert (Fehler, wenn angegeben).	--
17	FILTER_TIME	Gibt die Filterzeit für die Eingänge STP, LMT-, LMT+ und RPS an (1 Byte).  '0000' 200 µs '0101' 3200 µs '0001' 400 µs '0110' 6400 µs '0010' 800 µs '0111' 12800 µs '0011' 1600 µs '1000' Kein Filter '0100' 1600 µs '1001 ' bis '1111' Reserviert (Fehler, wenn angegeben).	--
18	MEAS_SYS	Gibt das Messsystem an (1 Byte). 0 Impulse (Drehzahl wird in Impulsen/Sekunde angegeben und die Positionswerte werden in Impulsen gemessen). Werte werden als Datentyp DINT gespeichert. 1 Physikalische Maßeinheiten (Drehzahl wird in Einheiten/Sekunde gemessen und die Positionswerte werden in Einheiten gemessen). Werte werden als einfachgenauer Datentyp REAL gespeichert. 2 bis 255 Reserviert (Fehler, wenn angegeben).	--
19	--	Reserviert (auf 0 gesetzt).	--
20	PLS/REV	Gibt die Anzahl an Impulsen pro Motorumdrehung an (4 Bytes). Gilt nur, wenn MEAS_SYS auf 1 gesetzt ist.	DINT
24	UNITS/REV	Gibt die physikalischen Maßeinheiten pro Motorumdrehung an (4 Bytes). Gilt nur, wenn MEAS_SYS auf 1 gesetzt ist.	REAL
28	UNITS	Reserviert für STEP 7-Micro/WIN, zum Speichern einer anwenderspezifischen Einheit (Zeichenkette) (4 Bytes).	--
32	RP_CFG	Gibt die Konfiguration für die Referenzpunktsuche an (1 Byte).  RP_SEEK_DIR Dieses Bit gibt die Anfangsrichtung für die Referenzpunktsuche an. (0 - positive Richtung, 1 - negative Richtung) RP_APPR_DIR Dieses Bit gibt die Anfahrrichtung zum Beenden der Referenzpunktsuche an. (0 - positive Richtung, 1 - negative Richtung) MODE Gibt die Methode für die Referenzpunktsuche an. '0000' Referenzpunktsuche deaktiviert. '0001' Der Referenzpunkt ist die Stelle, an der der Eingang RPS aktiv wird. '0010' Der Referenzpunkt liegt im Mittelpunkt des aktiven Bereichs des Eingangs RPS. '0011' Der Referenzpunkt liegt außerhalb des aktiven Bereichs des Eingangs RPS. '0100' Der Referenzpunkt liegt innerhalb des aktiven Bereichs des Eingangs RPS. '0101' bis '1111' Reserviert (Fehler, wenn ausgewählt).	--

Tabelle 9-15 Konfigurations-/Profiltabelle, Fortsetzung

Versatz	Name	Funktionsbeschreibung	Datentyp
33	--	Reserviert (auf 0 gesetzt).	--
34	RP_Z_CNT	Anzahl an Impulsen des Eingangs ZP für die Definition des Referenzpunkts (4 Bytes).	DINT
38	RP_FAST	Schnelle Drehzahl für die RP-Suche: MAX_SPD oder kleiner (4 Bytes).	DINT REAL
42	RP_SLOW	Langsame Drehzahl für die RP-Suche: maximale Drehzahl, aus der der Motor sofort zum Stillstand kommen kann (4 Bytes).	DINT REAL
46	SS_SPEED	Start-/Stoppdrehzahl (4 Bytes). Die Startdrehzahl ist die maximale Drehzahl, die der Motor sofort aus dem Stillstand erreichen kann, sowie die maximale Drehzahl, aus der der Motor sofort zum Stillstand kommen kann. Der Betrieb mit einer kleineren als dieser Drehzahl ist zulässig, doch die Beschleunigungs- und Verzögerungzeiten gelten dann nicht.	DINT REAL
50	MAX_SPEED	Maximale Betriebsdrehzahl des Motors (4 Bytes).	DINT REAL
54	JOG_SPEED	Tippdrehzahl. MAX_SPEED oder kleiner (4 Bytes).	
58	JOG_INCREMENT	Dieser Wert ist die Entfernung (bzw. die Anzahl an Impulsen), die in Reaktion auf einen einzelnen Tippimpuls vorgerückt wird (4 Bytes).	DINT REAL
62	ACCEL_TIME	Zeit in Millisekunden, die erforderlich ist, um von der Mindest- zur Höchstdrehzahl zu beschleunigen (4 Bytes).	DINT
66	DECEL_TIME	Zeit in Millisekunden, die erforderlich ist, um von der Höchst- zur Mindestdrehzahl zu verzögern (4 Bytes).	DINT
70	BKLSH_COMP	Spielausgleich: die Entfernung, mit der das Spiel im System bei Richtungswechsel ausgeglichen wird (4 Bytes).	DINT REAL
74	JERK_TIME	Zeit, während der der Ruckausgleich zu Beginn und am Ende der Beschleunigungs- und Verzögerungskurve (S-Kurve) angewendet wird. Der Wert 0 deaktiviert den Ruckausgleich. Die Ruckausgleichszeit wird in Millisekunden angegeben (4 Bytes).	DINT
Interaktiver Baustein			
78	MOVE_CMD	Stellt die Betriebsart ein (1 Byte). 0 Absolute Position 1 Relative Position 2 Kontinuierlicher positiver Lauf mit fester Drehzahl 3 Kontinuierlicher negativer Lauf mit fester Drehzahl 4 Manuelle Drehzahlsteuerung, positive Drehung 5 Manuelle Drehzahlsteuerung, negative Drehung 6 Kontinuierlicher positiver Lauf mit fester Drehzahl und ausgelöstem Stopp (Eingang RPS signalisiert Stopp) 7 Kontinuierlicher negativer Lauf mit fester Drehzahl und ausgelöstem Stopp (Eingang RPS signalisiert Stopp) 8 bis 255 - Reserviert (Fehler, wenn angegeben)	--
79	--	Reserviert. Auf 0 gesetzt.	--
80	TARGET_POS	Zielposition für diese Bewegung (4 Bytes).	DINT REAL
84	TARGET_SPEED	Zieldrehzahl für diese Bewegung (4 Bytes).	DINT REAL
88	RP_OFFSET	Absolute Position des Referenzpunkts (4 Bytes).	DINT REAL
Profilbaustein 0			
92 (+0)	STEPS	Anzahl Schritte in dieser Bewegungssequenz (1 Byte).	--
93 (+1)	MODE	Stellt die Betriebsart für diesen Profilbaustein ein (1 Byte). 0 Absolute Position 1 Relative Position 2 Kontinuierlicher positiver Lauf mit fester Drehzahl 3 Kontinuierlicher negativer Lauf mit fester Drehzahl 4 Reserviert (Fehler, wenn angegeben) 5 Reserviert (Fehler, wenn angegeben) 6 Kontinuierlicher positiver Lauf mit fester Drehzahl und ausgelöstem Stopp (RPS stellt Drehzahl ein) 7 Kontinuierlicher negativer Lauf mit fester Drehzahl und ausgelöstem Stopp (Eingang RPS signalisiert Stopp) 8 Kontinuierlicher positiver Lauf mit zwei Drehzahlen (RPS stellt Drehzahl ein) 9 Kontinuierlicher negativer Lauf mit zwei Drehzahlen (RPS stellt Drehzahl ein) 10 bis 255 - Reserviert (Fehler, wenn angegeben).	--

Tabelle 9-15 Konfigurations-/Profiltabelle, Fortsetzung

Versatz	Name		Funktionsbeschreibung	Datentyp
94 (+2)	0	POS	Position für Bewegungsschritt 0 (4 Bytes).	DINT REAL
98 (+6)		SPEED	Zieldrehzahl für Bewegungsschritt 0 (4 Bytes).	DINT REAL
102 (+10)	1	POS	Position für Bewegungsschritt 1 (4 Bytes).	DINT REAL
106 (+14)		SPEED	Zieldrehzahl für Bewegungsschritt 1 (4 Bytes).	DINT REAL
110 (+18)	2	POS	Position für Bewegungsschritt 2 (4 Bytes).	DINT REAL
114 (+22)		SPEED	Zieldrehzahl für Bewegungsschritt 2 (4 Bytes).	DINT REAL
118 (+26)	3	POS	Position für Bewegungsschritt 3 (4 Bytes).	DINT REAL
122 (+30)		SPEED	Zieldrehzahl für Bewegungsschritt 3 (4 Bytes).	DINT REAL
Profilbaustein 1				
126 (+34)	STEPS		Anzahl Schritte in dieser Bewegungssequenz (1 Byte).	--
127 (+35)	MODE		Stellt die Betriebsart für diesen Profilbaustein ein (1 Byte).	--
128 (+36)	0	POS	Position für Bewegungsschritt 0 (4 Bytes).	DINT REAL
132 (+40)		SPEED	Zieldrehzahl für Bewegungsschritt 0 (4 Bytes).	DINT REAL
...	...	...	...	...

Sondermerker für das Positioniermodul

Die S7-200 ordnet jedem intelligenten Modul 50 Bytes im Speicherbereich der Sondermerker (SM) zu, die sich nach der physikalischen Position des Moduls im E/A-System richten (siehe Tabelle 9-16). Wenn das Modul eine Fehlerbedingung oder eine Zustandsänderung der Daten erkennt, aktualisiert das Modul diese Sondermerker. Das erste Modul aktualisiert SMB200 bis SMB249 so, dass der Fehler gemeldet wird, das zweite Modul aktualisiert SMB250 bis SMB299 usw.

Hinweis

Ab Version 2.2 werden die Sondermerker für intelligente Module auf andere Art und Weise zugeordnet.

Wenn Sie mit einer CPU vor Version 2.2 arbeiten, müssen Sie alle intelligenten Module in Steckplätzen neben der CPU und vor allen nicht intelligenten Modulen anordnen, um die Kompatibilität sicherzustellen.

Tabelle 9-16 Sondermerkerbytes SMB200 bis SMB549

SM-Bytes für ein intelligentes Modul in:						
Steckplatz 0	**Steckplatz 1**	**Steckplatz 2**	**Steckplatz 3**	**Steckplatz 4**	**Steckplatz 5**	**Steckplatz 6**
SMB200 bis SMB249	SMB250 bis SMB299	SMB300 bis SMB349	SMB350 bis SMB399	SMB400 bis SMB449	SMB450 bis SMB499	SMB500 bis SMB549

Tabelle 9-17 zeigt den Aufbau des SM-Datenbereichs für ein intelligentes Modul. Die Definition wird so beschrieben, als ob es sich um das intelligente Modul in Steckplatz 0 des E/A-Systems handeln würde.

Tabelle 9-17 Definition der Sondermerker für das Positioniermodul EM 253

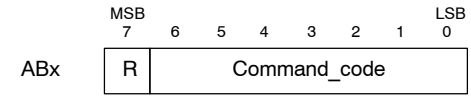
Adresse der Sonder-merker	Beschreibung
SMB200 bis SMB215	Modulname (16 ASCII-Zeichen). SMB200 ist das erste Zeichen: “EM 253 Position”.
SMB216 bis SMB219	Software-Versionsnummer (4 ASCII-Zeichen). SMB216 ist das erste Zeichen.
SMW220	Fehlercode für das Modul. In Tabelle 9-14 finden Sie eine Beschreibung der Fehlercodes.
SMB222	Eingangs-/Ausgangszustand. Gibt den Zustand der Eingänge und Ausgänge des Moduls an. MSB 7 6 5 4 3 2 1 0 LSB DIS | 0 | 0 | STP | LMT- | LMT+ | RPS | ZP DIS Ausgänge deaktivieren 0 = Kein Signalfluss 1 = Signalfluss STP Stoppeingang 0 = Kein Signalfluss 1 = Signalfluss LMT- Eingang negative Verfahrweggrenze 0 = Kein Signalfluss 1 = Signalfluss LMT+ Eingang positive Verfahrweggrenze 0 = Kein Signalfluss 1 = Signalfluss RPS Eingang Referenzpunktschalter 0 = Kein Signalfluss 1 = Signalfluss ZP Eingang Nullimpuls 0 = Kein Signalfluss 1 = Signalfluss
SMB223	Sofortiger Modulstatus. Gibt den Zustand der Modul-konfiguration und den Zustand der Drehrichtung an. MSB 7 6 5 4 3 2 1 0 LSB 0 | 0 | 0 | 0 | 0 | ODER | R | CFG OR Zieldrehzahl außerhalb des Bereichs 0 = Im Bereich 1 = Außerhalb des Bereichs R Drehrichtung 0 = Positive Drehung 1 = Negative Drehung CFG Module konfiguriert 0 = Nicht konfiguriert 1 = Konfiguriert
SMB224	CUR_PF ist ein Byte, das das Profil angibt, das gerade ausgeführt wird.
SMB225	CUR_STP ist ein Byte, das den Schritt angibt, der gerade im Profil ausgeführt wird.
SMD226	CUR_POS ist ein Doppelwortwert, der die aktuelle Position des Moduls angibt.
SMD230	CUR_SPD ist ein Doppelwortwert, der die aktuelle Drehzahl des Moduls angibt.
SMB234	Ergebnis der Operation. In Tabelle 9-13 finden Sie eine Beschreibung der Fehlercodes. Die Fehlerbedingungen über 127 werden von den Unterprogrammen erzeugt, die vom Assistenten erstellt wurden. MSB 7 6 LSB 0 d | ERROR d Bit Done 0 = In Bearbeitung 1 = Beendet (wird während der Initialisierung vom Modul gesetzt)
SMB235 bis SMB244	Reserviert
SMB245	Versatz zum ersten A-Byte, das als Befehlsschnittstelle dieses Moduls verwendet wird. Der Versatz wird zur Erleichte- rung des Anwenders automatisch von der S7-200 angegeben und wird vom Modul nicht benötigt.
SMD246	Pointer auf die Adresse im Variablenspeicher der Konfigurations-/Profiltabelle. Ein Pointerwert auf einen anderen Bereich als den Variablenspeicher ist nicht gültig. Das Positioniermodul beobachtet diese Adresse, bis es einen Pointerwert erhält, der nicht Null ist.

Beschreibung des Befehlsbyte für das Positioniermodul

Das Positioniermodul verfügt über ein Byte an Digitalausgängen, das als Befehlsbyte verwendet wird. Bild 9-22 zeigt die Definition des Befehlsbyte. Tabelle 9-18 zeigt die Definitionen im Command_code.

Wenn in das Befehlsbyte geschrieben wird und das Bit R von 0 nach 1 wechselt, wird dies vom Modul als neuer Befehl interpretiert.

Erkennt das Modul einen Übergang in den Leerlauf (Bit R geht auf 0), während ein Befehl aktiv ist, dann wird die in Bearbeitung befindliche Operation abgebrochen und es wird, sofern eine Bewegung in Bearbeitung ist, ein verzögerter Stopp durchgeführt.



R 0 = Leerlauf
1 = Befehl ausführen, der im Command_code angegeben ist (siehe Tabelle 9-18)

Bild 9-22 Definition des Befehlsbyte

Nach Beendigung eines Vorgangs muss das Modul zunächst in den Leerlauf wechseln, damit ein neuer Befehl akzeptiert wird. Wird ein Vorgang abgebrochen, muss das Modul alle Verzögerungen beenden, bevor ein neuer Befehl akzeptiert wird. Jede Änderung im Wert des Command_code wird ignoriert, solange ein Befehl aktiv ist.

Die Reaktion des Positioniermoduls auf einen Betriebszustandswechsel der S7-200 oder auf eine Fehlerbedingung wird durch die Tatsache geregelt, dass die S7-200 die Digitalausgänge entsprechend der vorhandenen Funktion der S7-200 steuert:

- *Wechselt die S7-200 von STOP in RUN:* Das Programm in der S7-200 steuert den Betrieb des Positioniermoduls.
- *Wechselt die S7-200 von RUN in STOP:* Sie können den Zustand, den die Digitalausgänge bei einem Wechsel in STOP einnehmen sollen, einstellen oder Sie können wählen, dass die Ausgänge im letzten Zustand bleiben.
  - *Wird das Bit R beim Wechsel in STOP ausgeschaltet:* Das Positioniermodul verzögert alle laufenden Bewegungen bis zum Stillstand.
  - *Wird das Bit R beim Wechsel in STOP eingeschaltet:* Das Positioniermodul beendet alle laufenden Befehle. Ist kein Befehl in Bearbeitung, dann führt das Positioniermodul den Befehl aus, der von den Bits im Command_code angegeben wird.
  - *Wird das Bit R in seinem letzten Zustand gehalten:* Das Positioniermodul beendet alle laufenden Bewegungen.
- *Wenn die S7-200 einen schweren Fehler erkennt und alle Digitalausgänge ausschaltet:*Das Positioniermodul verzögert alle laufenden Bewegungen bis zum Stillstand.

Tabelle 9-18 Definitionen Command_code

Command_code	Befehl	
000 0000 bis 000 1111	0 bis 24	Bewegung ausführen, die in den Profilbausteinen 0 bis 24 angegeben ist
100 0000 bis 111 0101	25 bis 117	Reserviert (Fehler, wenn angegeben)
111 0110	118	Ausgang DIS aktivieren
111 0111	119	Ausgang DIS deaktivieren
111 1000	120	Impuls am Ausgang CLR
111 1001	121	Aktuelle Position neu laden
111 1010	122	Bewegung ausführen, die im interaktiven Baustein angegeben ist
111 1011	123	Referenzpunktversatz erfassen
111 1100	124	Tippbetrieb positive Drehung
111 1101	125	Tippbetrieb negative Drehung
111 1110	126	Referenzpunkt suchen
111 1111	127	Konfiguration neu laden

Das Positioniermodul implementiert eine Zeitüberwachung, die die Ausgänge ausschaltet, wenn die Kommunikation mit der S7-200 verloren geht. Läuft die Zeitüberwachung der Ausgänge ab, verzögert das Positioniermodul alle laufenden Bewegungen bis zum Stillstand.

Wird in der Hardware oder in der Firmware des Moduls ein schwerer Fehler erkannt, setzt das Positioniermodul die Ausgänge P0, P1, DIS und CLR in den inaktiven Zustand.

Tabelle 9-19 Positionierbefehle

Befehl	Beschreibung
Befehle 0 bis 24: *Bewegung ausführen, die in den Profilbausteinen 0 bis 24 angegeben ist*	Wenn dieser Befehl ausgeführt wird, führt das Positioniermodul die Bewegung aus, die im Feld MODE im Profilbaustein angegeben ist, welches wiederum im Command_code des Befehls angegeben wird. ■ Im Modus 0 (absolute Position) definiert der Profilbaustein bis zu vier Schritte, wobei jeder Schritt sowohl die Position (POS) als auch die Drehzahl (SPEED) enthält, die das Bewegungssegment definieren. Die Angabe POS stellt eine absolute Lage dar, die auf der Lage basiert, die als Referenzpunkt bezeichnet wird. Die Bewegungsrichtung wird durch die Beziehung zwischen der aktuellen Position und der Position des ersten Schritts im Profil festgelegt. In einer Bewegung mit mehreren Schritten ist ein Fahrrichtungswechsel verboten und führt zu einer Fehlerbedingung. ■ Im Modus 1 (relative Position) definiert der Profilbaustein bis zu vier Schritte, wobei jeder Schritt sowohl die Position (POS) als auch die Drehzahl (SPEED) enthält, die das Bewegungssegment definieren. Das Vorzeichen des Positionswerts (POS) legt die Bewegungsrichtung fest. In einer Bewegung mit mehreren Schritten ist ein Fahrrichtungswechsel verboten und führt zu einer Fehlerbedingung. ■ In den Modi 2 und 3 (Betriebsarten mit kontinuierlichem Lauf und fester Drehzahl) wird die Angabe der Position (POS) ignoriert und das Modul beschleunigt auf die im Feld SPEED des ersten Schritts angegebene Drehzahl. Modus 2 dient für positive Drehung und Modus 3 für negative Drehung. Die Bewegung kommt zum Stillstand, wenn das Befehlsbyte in den Leerlauf geht. ■ In den Modi 6 und 7 (Betriebsarten mit kontinuierlichem Lauf mit fester Drehzahl und ausgelöstem Stopp) beschleunigt das Modul auf die im Feld SPEED des ersten Schritts angegebene Drehzahl. Wenn der Eingang RPS aktiv wird, kommt die Bewegung zum Stillstand, nachdem die im Feld POS des ersten Schritts angegebene Entfernung zurückgelegt wurde. (Die im Feld POS angegebene Entfernung muss die Verzögerungsstrecke umfassen.) Ist das Feld POS Null, wenn der Eingang RPS aktiv wird, verzögert das Positioniermodul bis zum Stillstand. Modus 6 dient für positive Drehung und Modus 7 für negative Drehung. ■ In den Modi 8 und 9 stellt der Binärwert des Eingangs RPS einen von zwei Drehzahlwerten ein, der in den ersten beiden Schritten im Profilbaustein angegeben ist. - Wenn der Eingang RPS inaktiv ist: Schritt 0 steuert die Drehzahl des Antriebs. - Wenn der Eingang RPS aktiv ist: Schritt 1 steuert die Drehzahl des Antriebs. Modus 8 dient für positive Drehung und Modus 9 für negative Drehung. Der Wert SPEED steuert die Drehzahl der Bewegung. Der Wert POS wird in diesem Modus ignoriert.
Befehl 118 *Ausgang DIS aktivieren*	Wenn dieser Befehl ausgeführt wird, aktiviert das Positioniermodul den Ausgang DIS.
Befehl 119 *Ausgang DIS deaktivieren*	Wenn dieser Befehl ausgeführt wird, deaktiviert das Positioniermodul den Ausgang DIS.
Befehl 120 *Impuls am Ausgang CLR*	Wenn dieser Befehl ausgeführt wird, erzeugt das Positioniermodul einen Impuls von 50 ms am Ausgang CLR.
Befehl 121 *Aktuelle Position neu laden*	Wenn dieser Befehl ausgeführt wird, setzt das Positioniermodul die aktuelle Position auf den Wert, der im Feld TARGET_POS des interaktiven Bausteins angegeben ist.
Befehl 122 *Bewegung ausführen, die im interaktiven Baustein angegeben ist*	Wenn dieser Befehl ausgeführt wird, führt das Positioniermodul die Bewegung aus, die im Feld MOVE_CMD des interaktiven Bausteins angegeben ist. ■ In den Modi 0 und 1 (absolute und relative Betriebsarten) wird ein einzelner Bewegungsschritt auf der Basis der Informationen zu Zieldrehzahl und Position ausgeführt, die in den Feldern TARGET_SPEED und TARGET_POS des interaktiven Bausteins angegeben werden. ■ In den Modi 2 und 3 (Betriebsarten mit kontinuierlichem Lauf mit fester Drehzahl) wird die Angabe der Position ignoriert und das Positioniermodul beschleunigt auf die im Feld TARGET_SPEED des interaktiven Bausteins angegebene Drehzahl. Die Bewegung kommt zum Stillstand, wenn das Befehlsbyte in den Leerlauf geht. ■ In den Modi 4 und 5 (Betriebsarten mit manuell einstellbarer Drehzahl) wird die Angabe der Position ignoriert und Ihr Programm lädt den Wert der Drehzahländerungen in das Feld TARGET_SPEED des interaktiven Bausteins. Das Positioniermodul beobachtet diese Lage kontinuierlich und reagiert entsprechend, wenn sich der Drehzahlwert ändert.

Tabelle 9-19 Positionierbefehle, Fortsetzung

Befehl	Beschreibung
Befehl 123 *Referenzpunktversatz erfassen*	Wenn dieser Befehl ausgeführt wird, richtet das Positioniermodul eine Nulllage ein, bei der es sich um eine andere Lage als den Referenzpunkt handelt. Bevor dieser Befehl abgesetzt wird, müssen Sie die Position des Referenzpunkts ermittelt haben und Sie müssen außerdem die Maschine im Tippbetrieb in die Ausgangsposition gebracht haben. Nachdem das Positioniermodul diesen Befehl erhalten hat, berechnet es den Versatz zwischen der Ausgangsposition (der aktuellen Position) und dem Referenzpunkt und schreibt den berechneten Versatz in das Feld RP_OFFSET des interaktiven Bausteins. Die aktuelle Position wird dann auf 0 gesetzt, um die Ausgangsposition als Nulllage einzurichten. Für den Fall, dass der Schrittmotor seine Lage ”vergisst” (z.B. bei Spannungsausfall oder wenn der Schrittmotor manuell neu positioniert wird), kann der Befehl Referenzpunkt suchen abgesetzt werden, um die Nulllage automatisch wieder einzurichten.
Befehl 124 *Tippbetrieb positive Drehung*	Mit diesem Befehl können Sie manuell Impulse ausgeben, um den Schrittmotor in die positive Richtung zu bewegen. Bleibt der Befehl weniger als 0,5 Sekunden lang aktiv, gibt das Positioniermodul Impulse aus, um die in JOG_INCREMENT angegebene Entfernung zu fahren. Bleibt der Befehl 0,5 Sekunden oder länger aktiv, beschleunigt das Positioniermodul auf die von JOG_SPEED angegebene Drehzahl. Wird ein Wechsel in den Leerlauf erkannt, verzögert das Positioniermodul bis zum Stillstand.
Befehl 125 *Tippbetrieb negative Drehung*	Mit diesem Befehl können Sie manuell Impulse ausgeben, um den Schrittmotor in die negative Richtung zu bewegen. Bleibt der Befehl weniger als 0,5 Sekunden lang aktiv, gibt das Positioniermodul Impulse aus, um die in JOG_INCREMENT angegebene Entfernung zu fahren. Bleibt der Befehl 0,5 Sekunden oder länger aktiv, beschleunigt das Positioniermodul auf die von JOG_SPEED angegebene Drehzahl. Wird ein Wechsel in den Leerlauf erkannt, verzögert das Positioniermodul bis zum Stillstand.
Befehl 126 *Referenzpunkt suchen*	Wenn dieser Befehl ausgeführt wird, startet das Positioniermodul eine Referenzpunktsuche nach der angegebenen Suchmethode. Wenn der Referenzpunkt ermittelt und die Bewegung zum Stillstand gebracht wurde, lädt das Positioniermodul den Wert aus dem Feld RP_OFFSET des interaktiven Bausteins in die aktuelle Position und erzeugt am Ausgang CLR einen Impuls von 50 ms.
Befehl 127 *Konfiguration neu laden*	Wenn dieser Befehl ausgeführt wird, liest das Positioniermodul den Pointer auf die Konfigurations-/Profiltabelle aus der entsprechenden Adresse im Speicher der Sondermerker. Anschließend liest das Modul den Konfigurationsbaustein aus der Adresse, die vom Pointer auf die Konfigurations-/Profiltabelle angegeben wird. Das Positioniermodul vergleicht die gerade erhaltenen Konfigurationsdaten mit der vorhandenen Modulkonfiguration und führt alle erforderlichen Setup-Änderungen oder Neuberechnungen durch. Alle gepufferten Profile werden verworfen.

Beschreibung des Profilpuffers des Positioniermoduls

Das Positioniermodul speichert die Ausführungsdaten von maximal 4 Profilen im Pufferspeicher. Wenn das Positioniermodul einen Befehl zur Ausführung eines Profils empfängt, prüft es, ob das angeforderte Profil im Pufferspeicher abgelegt ist. Befinden sich die Ausführungsdaten für das Profil im Pufferspeicher, führt das Positioniermodul das Profil sofort aus. Befinden sich die Ausführungsdaten für das Profil nicht im Pufferspeicher, liest das Positioniermodul die Daten des Profilbausteins aus der Konfigurations-/Profiltabelle der S7-200 und berechnet die Ausführungsdaten für das Profil, bevor es das Profil ausführt.

Befehl 122 (Im interaktiven Baustein angegebene Bewegung ausführen) speichert die Ausführungsdaten nicht im Pufferspeicher, sondern liest den interaktiven Baustein jedesmal aus der Konfigurations-/Profiltabelle der S7-200 und berechnet jedesmal die Ausführungsdaten für die Bewegung.

Wenn Sie das Positioniermodul neu konfigurieren, werden alle im Pufferspeicher abgelegten Ausführungsdaten gelöscht.

Erstellen Ihrer eigenen Positionieroperationen

Der Positionier-Assistent erstellt die Positionieroperationen zum Steuern des Betriebs des Positioniermoduls. Sie können allerdings auch Ihre eigenen Operationen erstellen. Der folgende AWL-Code zeigt ein Beispiel dafür, wie Sie Ihre eigenen Steuerungsoperationen für das Positioniermodul erstellen können.

In diesem Beispiel wird eine S7-200 CPU 224 mit einem Positioniermodul in Steckplatz 0 verwendet. Das Positioniermodul wird beim Anlauf konfiguriert. BEF_ZUST ist ein Symbol für SMB234, BEF ist ein Symbol für AB2 und BEF_NEU ist ein Symbol für das Profil.

Beispielprogramm: Steuern des Positioniermoduls

```
Netzwerk 1      //Zustand: Neuer Positionierbefehl
LSCR    Zustand_0

Netzwerk 2      //BEF_ZUST ist ein Symbol für SMB234
                //BEF ist ein Symbol für AB2
                //BEF_NEU ist ein Symbol für das Profil
                //
                //1. Bit Done des Positioniermoduls löschen.
                //2. Befehlsbyte des Positioniermoduls löschen.
                //3. Neuen Befehl absetzen.
                //4. Auf Ausführung des Befehls warten.
LD      SM0.0
MOVB    0, BEF_ZUST
BIW     0, CMD
BIW     BEF_NEU, BEF
SCRT    Zustand_1

Netzwerk 3
SCRE

Netzwerk 4      //Auf Beendigung der Befehlsausführung warten.
LSCR    Zustand_1

Netzwerk 5      //Wird der Befehl fehlerfrei beendet, in Leerlauf gehen.
LDB=    BEF_ZUST, 16#80
SCRT    Leerlauf_Zustand

Netzwerk 6      //Wird der Befehl mit Fehler beendet, in Fehlerbearbeitungszustand gehen.
LDB>    BEF_ZUST, 16#80
SCRT    Fehler_Zustand

Netzwerk 7
SCRE
```

Erstellen eines Programms für das Modemmodul

10

Mit Hilfe des Modemmoduls EM 241 können Sie Ihre S7-200 direkt an eine Analogtelefonleitung anschließen. Das Modemmodul unterstützt die Kommunikation zwischen Ihrer S7-200 und STEP 7-Micro/WIN. Das Modemmodul unterstützt außerdem das Modbus-Slave-RTU-Protokoll. Die Kommunikation zwischen dem Modemmodul und der S7-200 wird über den Erweiterungs-E/A-Bus aufgebaut.

STEP 7-Micro/WIN bietet einen erweiterten Modemassistenten, mit dem Sie einen entfernten Modus oder ein Modemmodul für die Verbindung zwischen einer lokalen S7-200 und einem entfernten Gerät einrichten können.

In diesem Kapitel

Funktionen des Modemmoduls 298
Konfigurieren des Modemmoduls EM mit dem erweiterten Modemassistenten 304
Übersicht über Modemoperationen und Einschränkungen 308
Operationen für das Modemmodul 309
Beispielprogramm für das Modemmodul 313
S7-200 CPUs, die intelligente Module unterstützen 313
Sondermerker für das Modemmodul 314
Für erfahrene Anwender 316
Format von Telefonnummern für die Nachrichtenübermittlung 318
Format von Textnachrichten 319
Format von CPU-Datenübertragungsnachrichten 320

Funktionen des Modemmoduls

Mit Hilfe des Modemmoduls können Sie Ihre S7-200 direkt an eine Analogtelefonleitung anschließen. Das Modemmodul bietet die folgenden Leistungsmerkmale:

- ❑ Internationale Schnittstelle zu Telefonleitungen
- ❑ Modemschnittstelle zu STEP 7-Micro/WIN für die Programmierung und Fehlerbehebung (Teleservice)
- ❑ Unterstützung des Modbus-RTU-Protokolls
- ❑ Unterstützung von Nummern- und Textfunkrufen
- ❑ Unterstützung von SMS-Nachrichtenübermittlung
- ❑ Datenübertragungen von CPU zu CPU und von CPU zu Modbus
- ❑ Passwortschutz
- ❑ Sicherheitsrückruf
- ❑ Speicherung der Konfiguration des Modemmoduls in der CPU

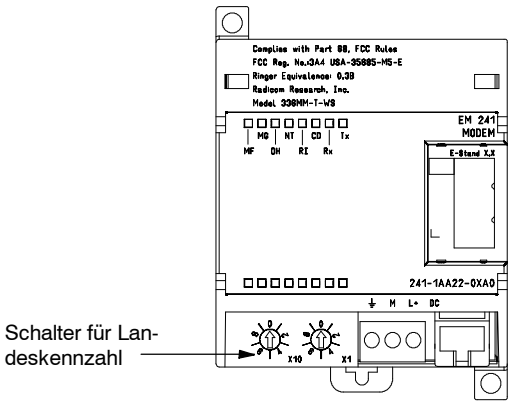
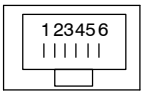


Bild 10-1 Modemmodul EM 241

Mit dem erweiterten Modemassistenten in STEP 7-Micro/WIN können Sie das Modemmodul konfigurieren. Die technischen Daten des Modemmoduls finden Sie in Anhang A.

Internationale Schnittstelle zu Telefonleitungen

Das Modemmodul ist ein herkömmliches 10-Bit-Modem V.34 (33,6 kBaud) und ist mit den meisten internen und externen PC-Modems kompatibel. Das Modemmodul kommuniziert nicht mit 11-Bit-Modems.



Pin	Beschreibung	
3	Klingeln	Umgekehrter Anschluss ist zulässig.
4	Tipp	

Bild 10-2 Ansicht des RJ11-Steckers

Sie schließen das Modemmodul über den sechspoligen RJ11-Vierdrahtstecker, der sich auf der Vorderseite des Moduls befindet, an die Telefonleitung an. (siehe Bild 10-2).

Je nach Land kann es sein, dass Sie einen Adapter benötigen, um den RJ11-Stecker für den Anschluss an eine Standard-Telefonleitung umzuwandeln. Ausführliche Informationen finden Sie in der Dokumentation zu Ihrem Adapterstecker.

Das Modem und die Schnittstelle zur Telefonleitung wird über eine externe 24-V-DC-Quelle mit Spannung versorgt. Diese kann an die CPU-Geberversorgung oder an eine externe Versorgung angeschlossen werden. Schließen Sie die Erdungsklemme des Modemmoduls an Systemerde an.

Das Modemmodul richtet die Telefonschnittstelle automatisch für länderspezifischen Betrieb ein, wenn das Modul mit Spannung versorgt wird. Mit den beiden Drehschaltern auf der Vorderseite des Moduls stellen Sie die Landeskennzahl ein. Sie müssen mit den Schaltern das entsprechende Land einstellen, bevor Sie das Modemmodul einschalten. Die Schalterstellungen für die unterstützten Länder sind in Tabelle 10-1 angegeben.

Tabelle 10-1 Vom EM 241 unterstützte Länder

Schalterstellung	Land
00	Australien
01	Österreich
02	Belgien
05	Kanada
08	Dänemark
09	Finnland
10	Frankreich
11	Deutschland
12	Griechenland
16	Irland
18	Italien
22	Luxemburg
25	Niederlande
26	Neuseeland
27	Norwegen
30	Portugal
34	Spanien
35	Schweden
36	Schweiz
38	Großbritannien
39	U.S.A.

Schnittstelle zu STEP 7-Micro/WIN

Mit dem Modemmodul können Sie über eine Telefonleitung mit STEP 7-Micro/WIN kommunizieren (Teleservice). Sie müssen die S7-200 CPU nicht konfigurieren oder programmieren, um das Modemmodul als entferntes Modem zu nutzen, wenn Sie mit STEP 7-Micro/WIN arbeiten.

Gehen Sie folgendermaßen vor, um das Modemmodul mit STEP 7-Micro/WIN zu nutzen:

1. Schalten Sie die Spannung der S7-200 CPU aus und schließen Sie das Modemmodul an den E/A-Erweiterungsbus an. Schließen Sie keine E/A-Module an, wenn die S7-200 CPU eingeschaltet ist.
2. Schließen Sie die Telefonleitung an das Modemmodul an. Verwenden Sie bei Bedarf einen Adapter.
3. Schließen Sie die Klemmenblöcke des Modemmoduls an eine 24-V-DC-Versorgung an.
4. Verbinden Sie den Erdanschluss des Klemmenblock des Modemmoduls mit Systemerde.
5. Stellen Sie die Schalter für die Landeskennzahl ein.
6. Schalten Sie die S7-200 CPU und das Modemmodul ein.
7. Konfigurieren Sie STEP 7-Micro/WIN für die Kommunikation mit einem 10-Bit-Modem.

Modbus-RTU-Protokoll

Sie können das Modemmodul so einrichten, dass es als Modbus-RTU-Slave fungiert. Das Modemmodul empfängt über die Modemschnittstelle Modbus-Anforderungen, es wertet diese Anforderungen aus und überträgt Daten an die oder von der CPU. Das Modemmodul erzeugt dann eine Modbus-Antwort und sendet diese über die Modemschnittstelle.

Tipp
Ist das Modemmodul als Modbus-RTU-Slave eingerichtet, kann STEP 7-Micro/WIN nicht über die Telefonleitung mit dem Modemmodul kommunizieren.

Das Modemmodul unterstützt die in Tabelle 10-2 aufgeführten Modbus-Funktionen.

Mit den Modbus-Funktionen 4 und 16 können maximal 125 Halteregister (250 Bytes im Variablenspeicher) in einer Anforderung gelesen oder geschrieben werden. Die Funktionen 5 und 15 schreiben in die Prozessabbilder der Ausgänge der CPU. Diese Werte können vom Anwenderprogramm überschrieben werden.

Modbus-Adressen werden üblicherweise als Werte aus 5 oder 6 Zeichen geschrieben, die den Datentyp und den Versatz angeben. Das erste Zeichen oder die ersten beiden Zeichen geben den Datentyp an, die letzten vier Zeichen geben den richtigen Wert innerhalb des Datentyps an. Das Modbus-Master-Gerät bildet die Adressen auf die entsprechenden Modbus-Funktionen ab.

Tabelle 10-2 Vom Modemmodul unterstützte Modbus-Funktionen

Funktion	Beschreibung
Funktion 01	Ausgangszustand lesen
Funktion 02	Eingangszustand lesen
Funktion 03	Halteregister lesen
Funktion 04	Prozessabbilder der Eingänge (analog) lesen
Funktion 05	Einen Ausgang schreiben
Funktion 06	Ein Register voreinstellen
Funktion 15	Mehrere Ausgänge schreiben
Funktion 16	Mehrere Register voreinstellen

Tabelle 10-3 zeigt die vom Modemmodul unterstützten Modbus-Adressen und die Abbildung der Modbus-Adressen auf die Adressen der S7-200 CPU.

Erstellen Sie mit dem erweiterten Modemassistenten einen Konfigurationsbaustein für das Modemmodul, damit das Modbus-RTU-Protokoll unterstützt wird. Der Konfigurationsbaustein des Modemmoduls muss in den Datenbaustein der CPU geladen werden, bevor Sie mit dem Modbus-Protokoll arbeiten können.

Tabelle 10-3 Abbilden von Modbus-Adressen auf die S7-200 CPU

Modbus-Adresse	Adresse der S7-200 CPU
000001	A0.0
000002	A0.1
000003	A0.2
...	...
000127	A15.6
000128	A15.7
010001	E0.0
010002	E0.1
010003	E0.2
...	...
010127	E15.6
010128	E15.7
030001	AEW0
030002	AEW2
030003	AEW4
...	...
030032	AEW62
040001	VW0
040002	VW2
040003	VW4
...	...
04xxxx	VW 2*(xxxx-1)

Funkruf und SMS-Nachrichtenübermittlung

Mit dem Modemmodul können Sie Nummern- und Textfunkrufnachrichten sowie SMS-Mitteilungen an Mobiltelefone senden (sofern vom Anbieter unterstützt). Die Nachrichten und Telefonnummern sind im Konfigurationsbaustein des Modemmoduls enthalten, der in den Datenbaustein in die S7-200 CPU geladen werden muss. Mit dem erweiterten Modemassistenten können Sie die Nachrichten und Telefonnummern für den Konfigurationsbaustein des Modemmoduls eingeben. Der erweiterte Modemassistent erstellt auch den Programmcode, damit Ihr Programm auslösen kann, wann Nachrichten gesendet werden.

Nummernfunkruf

Beim Nummernfunkruf werden über die Töne eines Tastentelefons numerische Werte an einen Pager gesendet. Das Modemmodul wählt den angegebenen Funkrufanbieter, wartet auf das Ende der Ansage und sendet dann die Töne, die den Ziffern in der Funkrufnachricht entsprechen. Eine Funkrufnachricht darf die Ziffern 0 bis 9, Sternchen (*), A, B, C und D enthalten. Die tatsächlich auf dem Pager angezeigten Zeichen anstelle des Sternchens bzw. der Zeichen A, B, C und D sind nicht genormt und werden vom Pager und dem Funkrufanbieter festgelegt.

Textfunkruf

Beim Textfunkruf können alphanumerische Nachrichten an einen Funkrufanbieter und von dort an einen Pager übermittelt werden. Textfunkrufanbieter verfügen im allgemeinen über eine Modemleitung, die Textnachrichten annimmt. Das Modemmodul nutzt das Telelocator Alphanumeric Protocol (TAP), um die Textnachrichten an den Funkrufanbieter zu übertragen. Viele Textfunkruf-Anbieter nutzen dieses Protokoll für die Annahme von Nachrichten.

Kurzmitteilungsdienst (SMS, Short Message Service)

Die SMS-Nachrichtenübermittlung wird von einigen Mobiltelefonanbietern unterstützt, im allgemeinen sind diese GSM-kompatibel. Über den Kurzmitteilungsdienst kann das Modemmodul eine Kurzmitteilung (SMS) per Telefonleitung an einen SMS-Anbieter senden. Der SMS-Anbieter sendet die Mitteilung dann an das Mobiltelefon und die Nachricht wird im Text-Display des Telefons angezeigt. Das Modemmodul nutzt die beiden Protokolle Telelocator Alphanumeric Protocol (TAP) und Universal Computer Protocol (UCP), um Nachrichten an den SMS-Anbieter zu senden. Sie können SMS-Kurzmitteilungen nur an SMS-Anbieter senden, die diese Protokolle für Modemleitungen unterstützen.

Eingebettete Variablen in Textnachrichten und SMS-Kurzmitteilungen

Das Modemmodul kann Datenwerte von der CPU in Textnachrichten einbetten und die Datenwerte wie angegeben in der Nachricht formatieren. Sie können die Anzahl der Ziffern links und rechts vom Dezimalpunkt angeben, und Sie können angeben, ob der Dezimalpunkt als Punkt oder Komma dargestellt werden soll. Wenn das Anwenderprogramm das Modemmodul anweist, eine Textnachricht zu senden, ruft das Modemmodul die Nachricht aus der CPU ab, ermittelt, welche CPU-Werte in der Nachricht benötigt werden und ruft diese aus der CPU ab und formatiert dann die Werte und legt sie in der Textnachricht ab, bevor die Nachricht an den entsprechenden Anbieter gesendet wird.

Die Telefonnummer des Nachrichtenübermittlungsanbieters, die Nachricht und die in die Nachricht eingebetteten Variablen werden während mehrerer CPU-Zyklen aus der CPU gelesen. Ihr Programm darf Telefonnummern und Nachrichten nicht ändern, während eine Nachricht gesendet wird. Die in die Nachricht eingebetteten Variablen dürfen beim Senden der Nachricht weiterhin aktualisiert werden. Enthält eine Nachricht mehrere Variablen, werden diese Variablen in mehreren Zyklen der CPU gelesen. Wenn Sie möchten, dass alle in eine Nachricht eingebetteten Variablen konsistent sind, dann dürfen Sie keine der eingebetteten Variablen ändern, nachdem Sie eine Nachricht gesendet haben.

Datenübertragungen

Mit Hilfe des Modemmoduls kann Ihr Programm über die Telefonleitung Daten an eine andere CPU oder an ein Modbus-Gerät übertragen. Die Datenübertragungen und Telefonnummern werden mit dem erweiterten Modemassistenten eingerichtet und im Konfigurationsbaustein des Modemmoduls gespeichert. Der Konfigurationsbaustein wird dann in den Datenbaustein in die S7-200 CPU geladen. Der erweiterte Modemassistent erstellt auch Programmcode, damit Ihr Programm die Datenübertragungen auslösen kann.

Eine Datenübertragung kann entweder eine Anforderung sein, Daten aus einem entfernten Gerät zu lesen, oder es kann eine Anforderung sein, Daten in ein entferntes Gerät zu schreiben. Eine Datenübertragung kann zwischen 1 Wort und 100 Wörtern an Daten lesen oder schreiben. Datenübertragungen übertragen Daten in den oder aus dem Variablenspeicher der angeschlossenen CPU.

Im erweiterten Modemassistenten können Sie eine Datenübertragung erstellen, die einmal aus einem entfernten Gerät Daten liest, die einmal in ein entferntes Gerät Daten schreibt oder die sowohl Daten aus dem entfernten Gerät liest als auch Daten in das entfernte Gerät schreibt.

Datenübertragungen nutzen das konfigurierte Protokoll des Modemmoduls. Ist das Modemmodul für das PPI-Protokoll eingerichtet (dann kann es auf STEP 7-Micro/WIN reagieren), dann nutzt das Modemmodul das PPI-Protokoll für die Datenübertragung. Ist das Modemmodul für das Modbus-RTU-Protokoll eingerichtet, dann werden die Datenübertragungen über das Modbus-Protokoll gesendet.

Die Telefonnummer des entfernten Geräts, die Datenübertragungsanforderung und die zu übertragenden Daten werden während mehrerer CPU-Zyklen aus der CPU gelesen. Ihr Programm darf Telefonnummern und Nachrichten nicht ändern, während eine Nachricht gesendet wird. Sie sollten auch die Daten, die übertragen werden, nicht ändern, während eine Nachricht gesendet wird.

Handelt es sich bei dem entfernten Gerät um ein anderes Modemmodul, kann die Passwortfunktion für die Datenübertragungen genutzt werden, indem das Passwort des entfernten Modemmoduls in die Konfiguration der Telefonnummer eingegeben wird. Die Rückruffunktion kann bei Datenübertragungen nicht verwendet werden.

Passwortschutz

Der Passwortschutz des Modemmoduls ist optional und wird im erweiterten Modemassistenten aktiviert. Das Passwort für das Modemmodul ist ein anderes als das für die CPU. Das Passwort für das Modemmodul ist ein eigenes Passwort mit 8 Zeichen. Dieses Passwort muss der Anrufer für das Modemmodul eingeben, damit er Zugriff auf die angeschlossene CPU erhält. Das Passwort wird im Variablenspeicher der CPU als Teil des Konfigurationsbausteins für das Modemmodul gespeichert. Der Konfigurationsbaustein des Modemmoduls muss in den Datenbaustein der angeschlossenen CPU geladen werden.

Ist der Passwortschutz für die CPU im Systemdatenbaustein aktiviert, muss der Anrufer das CPU-Passwort eingeben, damit er Zugriff auf die passwortgeschützten Funktionen der CPU erhält.

Sicherheitsrückruf

Die Rückruffunktion des Modemmoduls ist optional und wird im erweiterten Modemassistenten eingerichtet. Die Rückruffunktion bietet zusätzliche Sicherheit für die angeschlossene CPU, indem nur vordefinierten Telefonnummern Zugriff auf die CPU gewährt wird. Ist die Rückruffunktion aktiviert, beantwortet das Modemmodul alle eingehenden Anrufe, prüft den Anrufer und trennt dann die Verbindung. Handelt es sich um einen berechtigten Anrufer, wählt das Modemmodul die vordefinierte Telefonnummer für den Anrufer und gewährt diesem Zugriff auf die CPU.

Das Modemmodul unterstützt drei Rückrufarten:

- Rückruf einer vordefinierten Telefonnummer
- Rückruf mehrerer vordefinierter Telefonnummern
- Rückruf beliebiger Telefonnummern

Der Rückrufmodus wird durch Aktivieren der entsprechenden Option im erweiterten Modemassistenten und durch Definieren der Telefonnummern für die Rückruffunktion eingestellt. Die Rückruftelefonnummern werden im Konfigurationsbaustein des Modemmoduls im Datenbaustein der angeschlossenen CPU gespeichert.

Die einfachste Form eines Rückrufs geht an eine einzige vordefinierte Telefonnummer. Ist nur eine Rückrufnummer im Konfigurationsbaustein des Modemmoduls gespeichert, wird immer, wenn das Modemmodul einen eingehenden Anruf beantwortet, der Anrufer informiert, dass die Rückruffunktion aktiviert ist. Dann wird die Verbindung zum Anrufer getrennt und die im Konfigurationsbaustein angegebene Nummer gewählt.

Das Modemmodul unterstützt die Rückruffunktion auch für mehrere vordefinierte Telefonnummern. In diesem Modus wird der Anrufer nach einer Telefonnummer gefragt. Entspricht die angegebene Nummer einer der vordefinierten Telefonnummern im Konfigurationsbaustein des Modemmoduls, dann trennt das Modemmodul die Verbindung zum Anrufer und ruft den Anrufer über die entsprechende Telefonnummer im Konfigurationsbaustein zurück. Sie können bis zu 250 Rückrufnummern einrichten.

Sind mehrere Rückrufnummern vordefiniert, muss die Telefonnummer, die während der Verbindung zum Modemmodul angegeben wird, genau der Telefonnummer entsprechen, die im Konfigurationsbaustein des Modemmoduls gespeichert ist. Nur die ersten zwei Ziffern dürfen sich unterscheiden. Ist die eingerichtete Rückrufnummer 91(123)4569999, weil eine Amtsleitung (9) sowie ein Ferngespräch (1) gewählt werden müssen, kann es sich bei der für die Rückruffunktion angegebenen Nummer um eine der folgenden handeln:

- 91(123)4569999
- 1(123)4569999
- (123)4569999

Alle oben angegebenen Telefonnummern werden für die Rückruffunktion als Entsprechung aufgefasst. Das Modemmodul nutzt die Rückrufnummer aus dem Konfigurationsbaustein, wenn der Rückruf durchgeführt wird, in diesem Beispiel als 91(123)4569999. Wenn Sie mehrere Rückrufnummern einrichten, achten Sie darauf, dass alle Telefonnummern eindeutig sind, mit Ausnahme der ersten zwei Ziffern. Nur die numerischen Zeichen einer Telefonnummer werden bei Rückrufnummern verglichen. Zeichen wie Kommas oder Klammern werden beim Vergleich der Rückrufnummern ignoriert.

Den Rückruf beliebiger Telefonnummern aktivieren Sie im erweiterten Modemassistenten über die Option “Rückruf für alle Telefonnummern freigeben” beim Einrichten der Rückruffunktion. Ist diese Option aktiviert, beantwortet das Modemmodul einen eingehenden Anruf und verlangt die Rückruftelefonnummer. Nachdem der Anrufer die Telefonnummer angegeben hat, trennt das Modemmodul die Verbindung und wählt diese Telefonnummer. Dieser Rückrufmodus bietet die Möglichkeit, dass alle Verbindungskosten auf den Telefonanschluss des Modemmoduls laufen, er bietet keine Sicherheit für die S7-200 CPU. Das Passwort des Modemmoduls sollte aus Sicherheitsgründen bei diesem Rückrufmodus aktiviert werden.

Das Passwort des Modemmoduls und die Rückruffunktionen können gleichzeitig aktiviert werden. Bei aktiviertem Passwortschutz verlangt das Modemmodul vom Anrufer das Passwort, bevor der Rückruf getätigt wird.

Konfigurationstabelle für das Modemmodul

Alle Textnachrichten, Telefonnummern, Datenübertragungsinformationen, Rückrufnummern und andere Optionen werden in der Konfigurationstabelle des Modemmoduls gespeichert, die in den Variablenspeicher der S7-200 CPU geladen werden muss. Der erweiterte Modemassistent führt Sie durch die Erstellung einer Konfigurationstabelle für das Modemmodul. STEP 7-Micro/WIN legt die Konfigurationstabelle für das Modemmodul dann im Datenbaustein ab, der in die S7-200 CPU geladen wird.

Das Modemmodul liest diese Konfigurationstabelle beim Anlauf aus der CPU und innerhalb von fünf Sekunden bei jedem CPU-Betriebszustandswechsel von STOP in RUN. Das Modemmodul liest keine neue Konfigurationstabelle aus der CPU, solange das Modemmodul mit STEP 7-Micro/WIN online ist. Wird eine neue Konfigurationstabelle geladen, während das Modemmodul online ist, liest das Modemmodul die neue Konfigurationstabelle, wenn die Online-Sitzung beendet ist.

Erkennt das Modemmodul einen Fehler in der Konfigurationstabelle, blinkt die LED ”Modul gut (MG)” auf der Vorderseite des Moduls. Prüfen Sie die Informationen des Zielsystems in STEP 7-Micro/WIN oder lesen Sie den Wert in SMW220 (für Modulsteckplatz 0), um Informationen zum Konfigurationsfehler zu erhalten. Die Konfigurationsfehler des Modemmoduls werden in Tabelle 10-4 aufgeführt. Wenn Sie die Konfigurationstabelle des Modemmoduls im erweiterten Modemassistenten erstellen, prüft STEP 7-Micro/WIN die Daten, bevor die Konfigurationstabelle erzeugt wird.

Tabelle 10-4 Konfigurationsfehler beim EM 241 (hexadezimal)

Fehler	Beschreibung
0000	Kein Fehler aufgetreten
0001	Keine externe 24-V-DC-Versorgung
0002	Modemausfall
0003	Keine Konfigurationsbaustein-ID - Die Kennung des EM 241 am Anfang der Konfigurationstabelle ist für dieses Modul nicht gültig.
0004	Konfigurationsbaustein außerhalb des Bereichs - Der Pointer auf die Konfigurationstabelle zeigt nicht auf den Variablenspeicher oder ein Teil der Tabelle liegt außerhalb des Variablenspeicherbereichs der angeschlossenen CPU.
0005	Konfigurationsfehler - Rückruf ist aktiviert und die Anzahl der Rückrufnummern ist gleich 0 oder sie ist größer als 250. Die Anzahl der Nachrichten ist größer als 250. Die Anzahl der Telefonnummern für die Nachrichtenübermittlung ist größer als 250 oder die Länge derTelefonnummern für die Nachrichtenübermittlung ist größer als 120 Bytes.
0006	Fehlerhafte Landeskennzahl - Die Ländereinstellung über die beiden Drehschalter enthält einen Wert, der nicht unterstützt wird.
0007	Telefonnummer zu groß - Die Rückruffunktion ist aktiviert und die Länge der Rückrufnummer ist größer als das Maximum.
0008 bis 00FF	Reserviert
01xx	Fehler in Rückrufnummer xx - Die Rückrufnummer xx enthält unzulässige Zeichen. Der Wert xx ist 1 für die erste Rückrufnummer, 2 für die zweite Nummer usw.
02xx	Fehler in Telefonnummer xx - Eines der Felder in einer Telefonnummer xx für eine Nachrichtenübermittlung oder in einer Telefonnummer xx für eine Datenübertragung enthält einen unzulässigen Wert. Der Wert xx ist 1 für die erste Telefonnummer, 2 für die zweite Nummer usw.
03xx	Fehler in Nachricht xx - Die Nachricht bzw. die Datenübertragung mit der Nummer xx überschreitet die maximale Länge. Der Wert xx ist 1 für die erste Nachricht, 2 für die zweite usw.
0400 bis FFFF	Reserviert

Status-LEDs des Modemmoduls

Das Modemmodul verfügt über 8 Status-LEDs auf der Vorderseite. Tabelle 10-5 beschreibt die Status-LEDs.

Tabelle 10-5 Status-LEDs des EM 241

LED	Beschreibung
MF	Modulfehler - Diese LED ist eingeschaltet, wenn das Modul eine Fehlerbedingung erkennt wie: ■ Keine externe 24-V-DC-Versorgung ■ E/A-Zeitüberwachung abgelaufen ■ Modemausfall ■ Kommunikationsfehler mit lokaler CPU
MG	Modul gut - Diese LED ist eingeschaltet, wenn keine Fehlerbedingung vorliegt. Die LED "Modul gut" blinkt, wenn ein Fehler in der Konfigurationstabelle vorhanden ist oder wenn der Anwender eine unzulässige Ländereinstellung für die Schnittstelle der Telefonleitung ausgewählt hat. Prüfen Sie die Informationen des Zielsystems in STEP 7-Micro/WIN oder lesen Sie den Wert in SMW220 (für Modulsteckplatz 0), um Informationen zum Konfigurationsfehler zu erhalten.
OH	Besetzt - Diese LED ist eingeschaltet, wenn das EM 241 die Telefonleitung aktiv nutzt.
NT	Kein Wählton - Diese LED zeigt eine Fehlerbedingung an und wird eingeschaltet, wenn das EM 241 den Befehl erhalten hat, eine Nachricht zu senden, doch die Telefonleitung weist keinen Wählton auf. Diese Fehlerbedingung wird nur dann angezeigt, wenn das EM 241 so eingerichtet wurde, dass es vor dem Wählen auf einen Wählton warten soll. Die LED bleibt ca. 5 Sekunden lang nach einem fehlgeschlagenen Wählversuch eingeschaltet.
RI	Klingelerkennung - Diese LED zeigt an, dass beim EM 241 ein Anruf eingeht.
CD	Trägererkennung - Diese LED zeigt an, dass eine Verbindung zu einem entfernten Modem aufgebaut wurde.
Rx	Daten empfangen - Diese LED blinkt, wenn das Modem Daten empfängt.
Tx	Daten senden - Diese LED blinkt, wenn das Modem Daten sendet.

10 Konfigurieren des Modemmoduls EM mit dem erweiterten Modemassistenten

Rufen Sie im Menü "Extras" von STEP 7-Micro/WIN oder über den entsprechenden Bereich in der Navigationsleiste den erweiterten Modemassistenten auf.

Modem-erweiterung

Damit Sie diesen Assistenten nutzen können, muss das Projekt übersetzt und die symbolische Adressierung eingeschaltet sein. Wenn Sie Ihr Programm noch nicht übersetzt haben, tun Sie dies jetzt.

1. Wählen Sie in der ersten Maske des erweiterten Modemassistenten die Option "Modemmodul EM 241 einrichten" und klicken Sie auf "Weiter >".
2. Der erweiterte Modemassistent benötigt die Position des Modemmoduls relativ zur S7-200 CPU, damit der richtige Programmcode erzeugt werden kann. Klicken Sie auf die Schaltfläche "Module lesen", um die Positionen der intelligenten Module, die an die CPU angeschlossen sind, automatisch auszulesen. Erweiterungsmodule werden der Reihe nach durchnummeriert, begonnen wird bei Null. Doppelklicken Sie auf das Modemmodul, das Sie konfigurieren möchten, oder stellen Sie im Feld "Modulposition" die Position des Modemmoduls ein. Klicken Sie auf "Weiter >".

   Bei einer S7-200 CPU mit Firmware vor Version 1.2 müssen Sie das intelligente Modul neben der CPU einbauen, damit der Modem-Assistent das Modul konfigurieren kann.
3. In der Maske für den Passwortschutz können Sie den Passwortschutz für das Modemmodul aktivieren und dem Modul ein Passwort aus 1 bis 8 Zeichen zuweisen. Dieses Passwort ist unabhängig vom Passwort für die S7-200 CPU. Ist das Modul durch ein Passwort geschützt, muss jeder, der versucht, über das Modemmodul eine Verbindung zur S7-200 CPU aufzubauen, das Passwort eingeben. Stellen Sie den Passwortschutz bei Bedarf ein und geben Sie ein Passwort ein. Klicken Sie auf "Weiter >".
4. Das Modemmodul unterstützt zwei Kommunikationsprotokolle: das PPI-Protokoll (für die Kommunikation mit STEP 7-Micro/WIN) und das Modbus-RTU-Protokoll. Die Protokollauswahl richtet sich nach der Art des Geräts, das als entfernter Kommunikationspartner dienen soll. Diese Einstellung steuert das Kommunikationsprotokoll, das verwendet wird, wenn das Modemmodul auf einen Anruf antwortet und auch wenn das Modemmodul eine CPU-Datenübertragung initiiert. Wählen Sie das entsprechende Protokoll und klicken Sie auf "Weiter >".

5. Sie können das Modul so einrichten, dass es Nummer- und Textnachrichten an Pager sowie Kurzmitteilungen an Mobiltelefone senden kann. Aktivieren Sie das Kontrollkästchen ”Nachrichtenübermittlung freigeben” und wählen Sie die Schaltfläche ”Nachrichtenübermittlung einrichten...”, um die Nachrichten und die Telefonnummer des Empfängers zu definieren.

6. Wenn Sie eine Nachricht einrichten, die an einen Pager oder ein Mobiltelefon gesendet werden soll, müssen Sie die Nachricht und die Telefonnummer definieren. Öffnen Sie in der Maske ”Nachrichtenübermittlung konfigurieren” das Register ”Nachrichten” und wählen Sie die Schaltfläche ”Neue Nachricht”. Geben Sie den Text für die Nachricht ein und geben Sie CPU-Datenwerte ein, die in die Nachricht eingefügt werden sollen. Zum Einfügen eines CPU-Datenwerts in die Nachricht positionieren Sie den Cursor an der Stelle, an der die Daten eingefügt werden sollen, und klicken auf die Schaltfläche ”Daten einfügen...”. Geben Sie die Adresse des CPU-Datenwerts (z.B. VW100), das Anzeigeformat (z.B. vorzeichenlose Ganzzahl) und die Ziffern links und rechts vom Dezimalpunkt an. Sie können außerdem angeben, ob es sich bei dem Dezimalpunkt um einen Punkt oder um ein Komma handeln soll.

   - Nachrichten für den Nummernfunkruf dürfen nur die Ziffern 0 bis 9, die Buchstaben A, B, C und D und Sternchen (*) enthalten. Die maximal zulässige Länge einer Nachricht für einen Nummernfunkruf ist bei den verschiedenen Anbietern unterschiedlich.
   - Textnachrichten dürfen bis zu 119 Zeichen lang sein und können beliebige alphanumerische Zeichen enthalten.
   - Textnachrichten können eine beliebige Anzahl eingebetteter Variablen enthalten.
   - Eingebettete Variablen können aus den Speicherbereichen V, M, SM, E, A, S, T, Z oder AE der angeschlossenen CPU stammen.
   - Hexadezimaldaten wird die Kennzeichnung ‘16#’ vorangestellt. Die Zeichenzahl des Werts richtet sich nach der Größe der Variablen. VW100 beispielsweise wird angezeigt als 16#0123.
   - Die Anzahl der Ziffern links vom Dezimalpunkt müssen groß genug sein, um den erwarteten Wertebereich anzuzeigen, einschließlich negativem Vorzeichen, sofern es sich bei dem Datenwert um eine Ganzzahl mit Vorzeichen oder um eine Gleitpunktzahl handelt.
   - Ist das Datenformat Ganzzahl und die Anzahl der Ziffern rechts vom Dezimalpunkt ist nicht gleich Null, wird der ganzzahlige Wert als skalierte Ganzzahl angezeigt. Ist VW100 beispielsweise gleich 1234 und es gibt zwei Ziffern rechts vom Dezimalpunkt, dann werden die Daten angezeigt als ‘12.34’.
   - Ist der Datenwert größer als er in der angegebenen Feldgröße angezeigt werden kann, legt das Modemmodul das Zeichen # in allen Stellen des Datenwerts ab.

7. Telefonnummern werden in der Maske ”Nachrichtenübermittlung einrichten” im Register ”Telefonnummern” eingerichtet. Klicken Sie auf die Schaltfläche ”Neue Telefonnummer...”, um eine neue Telefonnummer zu ergänzen. Nachdem Sie eine Telefonnummer konfiguriert haben, muss sie ins Projekt aufgenommen werden. Markieren Sie die Telefonnummer in der Spalte ”Verfügbare Telefonnummern” und klicken Sie auf den Pfeil nach rechts, um die Telefonnummer in das aktuelle Projekt aufzunehmen. Wenn Sie die Telefonnummer ins aktuelle Projekt aufgenommen haben, können Sie die Telefonnummer auswählen und ihr einen symbolischen Namen für die Verwendung im Programm zuweisen.

   Die Telefonnummer besteht aus mehreren Feldern, die sich je nach der vom Anwender eingestellten Nachrichtenübermittlung unterscheiden.

   - Das Nachrichtenübermittlungsprotokoll teilt dem Modemmodul mit, welches Protokoll zum Senden der Nachricht an den Nachrichtenübermittlungsanbieter verwendet werden soll. Ziffernpager unterstützen nur das numerische Protokoll. Textfunkrufanbieter unterstützen meist das TAP (Telelocator Alphanumeric Protocol). SMS-Nachrichtenübermittlungsanbieter unterstützten das TAP oder das UCP (Universal Computer Protocol). Es gibt drei verschiedene UCP-Dienste, die üblicherweise für die SMS-Nachrichtenübermittlung eingesetzt werden. Die meisten Anbieter unterstützen die Befehle 1 und 51. Prüfen Sie, welches Protokoll und welche Befehle Ihr SMS-Anbieter unterstützt.
   - Im Feld ”Beschreibung” können Sie eine Textbeschreibung für die Telefonnummer eingeben.

- Das Feld ”Telefonnummer” enthält die Telefonnummer des Nachrichtenübermittlungsanbieters. Bei Textnachrichten ist dies die Telefonnummer der Modemleitung, über die der Anbieter die Textnachrichten annimmt. Beim Nummernfunkruf ist dies die Telefonnummer des Pagers selbst. Sie können beim Modemmodul bis zu 40 Zeichen in das Feld ”Telefonnummer” eingeben. Die folgenden Zeichen sind in Telefonnummern zugelassen, mit denen das Modemmodul eine Verbindung anwählt:

0 bis 9	zugelassen in der Telefontastatur
A, B, C, D, *, #	DTMF-Ziffern (nur Tonwahl)
,	Wählpause von 2 Sekunden
!	Befehl an das Modem, einen Hook Flash zu erzeugen
@	Warten auf 5 Sekunden Ruhe
W	Warten auf Wählton, dann weiter
( )	Werden ignoriert (können zum Formatieren der Telefonnummer verwendet werden)

- Im Feld ”Bestimmte Pager-Kennung oder Mobiltelefonnummer” geben Sie die Pagernummer oder die Mobiltelefonnummer des Empfängers der Nachricht ein. Diese Nummer darf außer den Ziffern 0 bis 9 keine anderen Zeichen enthalten. Es sind maximal 20 Zeichen zugelassen.
- Das Feld ”Passwort” ist bei TAP-Nachrichten optional. Bei einigen Anbietern müssen Sie ein Passwort eingeben, doch meistens bleibt dieses Feld leer. Das Modemmodul lässt ein Passwort mit maximal 15 Zeichen zu.
- Über das Feld ”Ursprungstelefonnummer” können Sie das Modemmodul in SMS-Kurzmitteilungen angeben. Diese Feld ist bei einigen Anbietern, die mit UCP-Befehlen arbeiten, erforderlich. Bei einigen Anbietern ist in diesem Feld eine Mindestlänge erforderlich. Das Modemmodul lässt maximal 15 Zeichen zu.
- Das Feld ”Modem-Standard” ist in solchen Situationen nützlich, in denen das Modemmodul und das Modem des Anbieters keinen gemeinsamen Modem-Standard finden können. Die Voreinstellung ist V.34 (33,6 kBaud).
- In den Feldern ”Datenformat” können Sie die Datenbits und die Parität anpassen, die vom Modem beim Senden einer Nachricht an einen Anbieter verwendet werden. TAP nutzt normalerweise 7 Datenbits und gerade Parität, doch einige Anbieter nutzen 8 Datenbits und keine Parität. UCP nutzt immer 8 Datenbits ohne Parität. Erfragen Sie beim Anbieter, welche Einstellungen zu verwenden sind.

8. Sie können das Modemmodul so einrichten, dass Daten an eine andere S7-200 CPU übertragen werden können (sofern das PPI-Protokoll eingestellt ist) oder dass Daten an ein Modbus-Gerät übertragen werden können (sofern das Modbus-Protokoll eingestellt ist). Aktivieren Sie das Kontrollkästchen ”CPU-Datenübertragungen freigeben” und wählen Sie die Schaltfläche ”CPU zu xx einrichten...”, um die Datenübertragungen und die Telefonnummern der entfernten Geräte zu definieren.
9. Wenn Sie eine Datenübertragung von CPU zu CPU oder von CPU zu Modbus einrichten, müssen Sie die Daten für die Übertragung definieren und Sie müssen die Telefonnummer des entfernten Geräts angeben. Öffnen Sie in der Maske ”Datenübertragungen einrichten” das Register ”Datenübertragungen” und wählen Sie die Schaltfläche ”Neue Übertragung”. Eine Datenübertragung besteht darin, Daten aus dem entfernten Gerät zu lesen, Daten in das entfernte Gerät zu schreiben oder sowohl Daten aus dem entfernten Gerät zu lesen als auch Daten in das entfernte Gerät zu schreiben. Sollen Daten gelesen und geschrieben werden, wird zuerst gelesen und dann wird geschrieben.

   Bis zu 100 Wörter können pro Lese- oder Schreibvorgang übertragen werden. Datenübertragungen übertragen Daten in den oder aus dem Variablenspeicher der lokalen CPU. Der Assistent beschreibt die Adressen im Speicher des entfernten Geräts immer so, als ob es sich bei dem entfernten Gerät um eine S7-200 CPU handelt. Ist das entfernte Gerät ein Modbus-Gerät, werden die Daten aus den oder in die Halteregister im Modbus-Gerät übertragen (Adresse 04xxxx). Die entsprechende Modbus-Adresse (xxxx) wird wie folgt ermittelt:

Modbus-Adresse	= 1 + (Adresse im Variablenspeicher / 2)
Adresse im Variablenspeicher	= (Modbus-Adresse - 1) * 2

10. In der Maske ”CPU-Datenübertragungen einrichten” können Sie im Register ”Telefonnummern” die Telefonnummern für die Datenübertragungen von CPU zu CPU oder von CPU zu Modbus einrichten. Klicken Sie auf die Schaltfläche ”Neue Telefonnummer...”, um eine neue Telefonnummer zu ergänzen. Nachdem Sie eine Telefonnummer konfiguriert haben, muss sie ins Projekt aufgenommen werden. Markieren Sie die Telefonnummer in der Spalte ”Verfügbare Telefonnummern” und klicken Sie auf den Pfeil nach rechts, um die Telefonnummer in das aktuelle Projekt aufzunehmen. Wenn Sie die Telefonnummer ins aktuelle Projekt aufgenommen haben, können Sie die Telefonnummer auswählen und ihr einen symbolischen Namen für die Verwendung im Programm zuweisen.

    Die Felder ”Beschreibung” und ”Telefonnummer” haben die gleiche Funktion wie weiter oben für die Nachrichtenübermittlung beschrieben. Das Feld ”Passwort” ist erforderlich, wenn das entfernte Gerät ein Modemmodul ist und der Passwortschutz aktiviert ist. Im Feld ”Passwort” des lokalen Modemmoduls muss das Passwort des entfernten Modemmoduls eingestellt werden. Das lokale Modemmodul liefert dieses Passwort, wenn es vom entfernten Modemmodul dazu aufgefordert wird.

11. Durch die Rückruffunktion trennt das Modemmodul automatisch die Verbindung und wählt eine vordefinierte Telefonnummer, nachdem ein Anruf von einem entfernten STEP 7-Micro/WIN eingegangen ist. Aktivieren Sie das Kontrollkästchen ”Rückruf freigeben” und wählen Sie die Schaltfläche ”Rückruf einrichten...”, um Telefonnummern für die Rückruffunktion einzugeben. Klicken Sie auf ”Weiter >”.

12. In der Maske ”Rückruf einrichten...” können Sie die Telefonnummern eingeben, die das Modemmodul in Beantwortung eines eingehenden Anrufs wählen soll. Aktivieren Sie die Option ”Rückruf nur für bestimmte Telefonnummern freigeben”, wenn die Rückrufnummern vordefiniert werden sollen. Wenn das Modemmodul jede Rückrufnummer vom eingehenden Anrufer akzeptieren soll (um die Verbindungsentgelte zu übernehmen), aktivieren Sie die Option ”Rückruf für alle Telefonnummern freigeben”.

    Wenn nur bestimmte Rückruftelefonnummern zugelassen sind, wählen Sie die Schaltfläche ”Neue Telefonnummer” und ergänzen die Rückrufnummern. In der Maske ”Eigenschaften Rückruf” können Sie die vordefinierten Rückruftelefonnummern und eine Beschreibung der Rückrufnummer eingeben. Die hier eingegebene Rückrufnummer ist die Telefonnummer, die das Modemmodul wählt, wenn es den Rückruf tätigt. Diese Telefonnummer muss alle Ziffern enthalten, die erforderlich sind, um die Verbindung über eine Amtsleitung herzustellen, z.B. Pause zum Warten auf die Amtsleitung, Ferngespräch usw.

    Nachdem Sie eine Rückruftelefonnummer eingegeben haben, muss diese ins Projekt aufgenommen werden. Markieren Sie die Telefonnummer in der Spalte ”Verfügbare Rückruftelefonnummern” und klicken Sie auf den Pfeil nach rechts, um die Telefonnummer in das aktuelle Projekt aufzunehmen.

13. Sie können die Anzahl der Wählversuche einstellen, die das Modemmodul zum Senden einer Nachricht oder für eine Datenübertragung unternimmt. Das Modemmodul meldet nur dann einen Fehler an das Anwenderprogramm, wenn alle Versuche, zu wählen und die Nachricht zu senden, erfolglos waren.

    Einige Telefonleitungen haben keinen Wählton, wenn der Telefonhörer abgenommen wird. Normalerweise gibt das Modemmodul einen Fehler an das Anwenderprogramm aus, wenn kein Wählton vorhanden ist und das Modemmodul eine Nachricht senden oder einen Rückruf tätigen soll. Damit über eine Leitung ohne Wählton gewählt werden kann, aktivieren Sie das Kontrollkästchen ”Wählen ohne Wählton freigeben”.

14. Der erweiterte Modemassistent erstellt einen Konfigurationsbaustein für das Modemmodul, für den Sie die Anfangsadresse im Speicher angeben müssen, an der die Konfigurationsdaten des Modemmoduls gespeichert werden. Der Konfigurationsbaustein des Modemmoduls wird im Variablenspeicher der CPU abgelegt. STEP 7-Micro/WIN schreibt den Konfigurationsbaustein in den Datenbaustein des Projekts. Die Größe des Konfigurationsbausteins richtet sich nach der Anzahl der Nachrichten und Telefonnummern, die Sie eingerichtet haben. Sie können die Adresse im Variablenspeicher eingeben, an der Sie den Konfigurationsbaustein speichern möchten. Sie können auch auf die Schaltfläche ”Adresse vorschlagen” klicken, wenn Sie möchten, dass der Assistent einen freien Baustein im Variablenspeicher mit der richtigen Größe vorschlägt. Klicken Sie auf ”Weiter >”.

15. Der letzte Schritt in der Konfiguration des Modemmoduls ist die Angabe der Adresse im Speicher der Ausgänge für das Befehlsbyte des Modemmoduls. Sie können die Adresse im Speicherbereich der Ausgänge ermitteln, indem Sie die Ausgangsbytes zählen, die von den Modulen mit digitalen Ausgängen verwendet werden, die vor dem Modemmodul auf der S7-200 montiert sind. Klicken Sie auf ”Weiter >”.

16. Der erweiterte Modemassistent erzeugt nun die Projektkomponenten für die von Ihnen eingegebene Konfiguration (Programmbaustein und Datenbaustein) und stellt diesen Code zur Verwendung in Ihrem Anwenderprogramm zur Verfügung. In der letzten Maske im Assistenten werden die von Ihnen benötigten Projektkomponenten für die Konfiguration angezeigt. Sie müssen den Konfigurationsbaustein des Modemmoduls (Datenbaustein) und den Programmbaustein in die S7-200 CPU laden.

Übersicht über Modemoperationen und Einschränkungen

Mit dem erweiterten Modemassistenten können Sie das Modemmodul auf einfache Weise steuern, indem Sie eindeutige Unterprogramme erstellen, die auf der von Ihnen eingestellten Position des Moduls und auf den von Ihnen gewählten Konfigurationsoptionen beruhen. Alle Operationen verfügen über das Präfix “MODx_”, wobei das x die Modulposition angibt.

Anforderungen für den Einsatz der Operationen für das Modemmodul EM 241

Berücksichtigen Sie diese Anforderungen, wenn Sie die Operationen für das Modemmodul verwenden:

- ❑ Die Operationen für das Modemmodul nutzen drei Unterprogramme.
- ❑ Die Operationen für das Modemmodul erhöhen den Speicherbedarf für Ihr Programm um bis zu 370 Bytes. Wenn Sie eine nicht benötigte Operation löschen, können Sie den erweiterten Modemassistenten erneut ausführen, um die Operation bei Bedarf neu zu erstellen.
- ❑ Sie müssen darauf achten, dass immer nur eine Operation zur Zeit aktiv ist.
- ❑ Die Operationen dürfen nicht in Interruptprogrammen verwendet werden.
- ❑ Das Modemmodul liest die Informationen in der Konfigurationstabelle beim Einschalten und nach einem Betriebszustandswechsel von STOP in RUN. Alle Änderungen, die Ihr Programm in der Konfigurationstabelle vornimmt, werden vom Modul erst beim Betriebszustandswechsel oder beim nächsten Einschalten erkannt.

Einsatz der Operationen für das Modemmodul EM 241

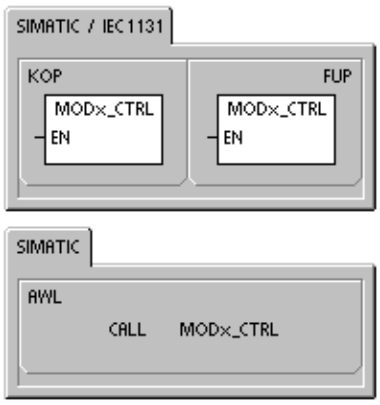
Wenn Sie die Operationen für das Modemmodul in Ihrem S7-200 Programm verwenden möchten, gehen Sie folgendermaßen vor:

1. Mit dem erweiterten Modemassistenten erstellen Sie die Konfigurationstabelle für das Modemmodul.
2. Fügen Sie die Operation MODx_CTRL in Ihr Programm ein und führen Sie sie mit dem Kontakt SM0.0 in jedem Zyklus aus.
3. Fügen Sie für jede Nachricht, die Sie senden möchten, eine Operation MODx_MSG ein.
4. Fügen Sie für jede Datenübertragung eine Operation MODx_XFR ein.

Operationen für das Modemmodul

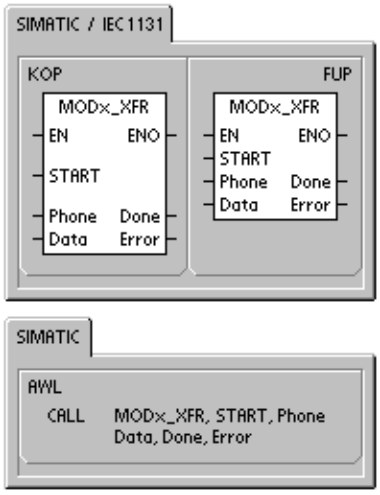
Operation MODx_CTRL

Die Operation MODx_CTRL (Steuerung) dient zur Aktivierung und Initialisierung des Modemmoduls. Diese Operation muss in jedem Zyklus
aufgerufen werden und darf nur einmal im Projekt verwendet werden.



Operation MODx_XFR

Die Operation MODx_XFR (Datenübertragung) befiehlt dem Modemmodul Daten einer anderen S7-200 CPU oder eines Modbus-Geräts zu lesen bzw. Daten in ein anderes Gerät zu schreiben. Diese Operation benötigt 20 bis 30 Sekunden ab dem Zeitpunkt, zu dem der Eingang START ausgelöst wird, bis zu dem Zeitpunkt, zu dem das Bit Done gesetzt wird.



Das Bit EN muss eingeschaltet sein, damit ein Befehl für das Modul abgesetzt werden kann. Es muss so lange eingeschaltet bleiben, bis das Bit Done gesetzt und dadurch das Ende des Prozesses angezeigt wird. Der Befehl XFR wird dem Modemmodul in jedem Zyklus gesendet, wenn der Eingang START eingeschaltet ist und das Modul nicht gerade besetzt ist. Der Eingang START kann durch die Erkennung einer steigenden Flanke eingeschaltet werden, so dass nur ein Befehl gesendet wird.

Phone ist eine der Telefonnummern für die Datenübertragungen. Sie können den symbolischen Namen verwenden, den Sie den Telefonnummern für die Datenübertragungen im erweiterten Modemassistenten zugewiesen haben.

Data ist die Nummer einer der definierten Datenübertragungen. Sie können den symbolischen Namen verwenden, den Sie der Datenübertragung im erweiterten Modemassistenten zugewiesen haben.

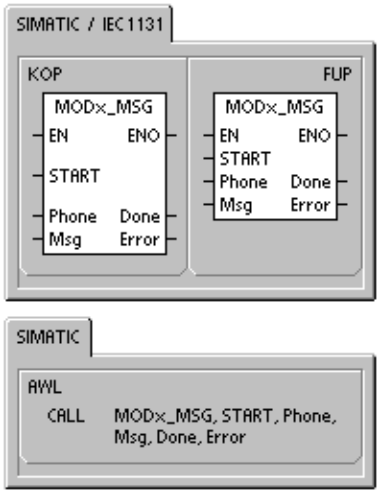
Done ist ein Bit, das eingeschaltet wird, wenn das Modemmodul die Datenübertragung beendet hat.

Error ist ein Byte, das das Ergebnis der Datenübertragung enthält. Tabelle 10-4 zeigt die möglichen Fehlerbedingungen an, die während der Ausführung dieser Operationen auftreten können.

Tabelle 10-6 Parameter der Operation MODx_XFR

Eingänge/Ausgänge	Datentyp	Operanden
START	BOOL	E, A, M, S, SM, T, Z, V, L, Signalfluss
Phone, Data	BYTE	VB, EB, AB, MB, SB, SMB, LB, AC, Konstante, *VD, *AC, *LD
DONE	BOOL	E, A, M, S, SM, T, Z, V, L
Error	BYTE	VB, EB, AB, MB, SB, SMB, LB, AC, *VD, *AC, *LD

Operation MODx_MSG



Die Operation MODx_MSG (Nachricht senden) sendet eine Funkrufnachricht oder eine SMS-Kurzmitteilung vom Modemmodul. Diese Operation benötigt 20 bis 30 Sekunden ab dem Zeitpunkt, zu dem der Eingang START ausgelöst wird, bis zu dem Zeitpunkt, zu dem das Bit Done gesetzt wird.

Das Bit EN muss eingeschaltet sein, damit ein Befehl für das Modul abgesetzt werden kann. Es muss so lange eingeschaltet bleiben, bis das Bit Done gesetzt und dadurch das Ende des Prozesses angezeigt wird. Der Befehl MSG wird dem Modemmodul in jedem Zyklus gesendet, wenn der Eingang START eingeschaltet ist und das Modul nicht gerade besetzt ist. Der Eingang START kann durch die Erkennung einer steigenden Flanke eingeschaltet werden, so dass nur ein Befehl gesendet wird.

Phone ist eine der Telefonnummern für die Nachrichtenübermittlung. Sie können den symbolischen Namen verwenden, den Sie den Telefonnummern der Nachrichten im erweiterten Modemassistenten zugewiesen haben.

Msg ist die Nummer von einer der definierten Nachrichten. Sie können den symbolischen Namen verwenden, den Sie der Nachricht im erweiterten Modemassistenten zugewiesen haben.

Done ist ein Bit, das eingeschaltet wird, wenn das Modemmodul das Senden der Nachricht an den Anbieter beendet hat.

Error ist ein Byte, das das Ergebnis dieser Anforderung an das Modul enthält. Tabelle 10-8 zeigt die möglichen Fehlerbedingungen an, die während der Ausführung dieser Operationen auftreten können.

Tabelle 10-7 Parameter der Operation MODx_MSG

Eingänge/Ausgänge	Datentyp	Operanden
START	BOOL	E, A, M, S, SM, T, Z, V, L, Signalfluss
Phone, Msg	BYTE	VB, EB, AB, MB, SB, SMB, LB, AC, Konstante, *VD, *AC, *LD
DONE	BOOL	E, A, M, S, SM, T, Z, V, L
Error	BYTE	VB, EB, AB, MB, SB, SMB, LB, AC, *VD, *AC, *LD

Tabelle 10-8 Fehlerwerte der Operationen MODx_MSG und MODx_XFR

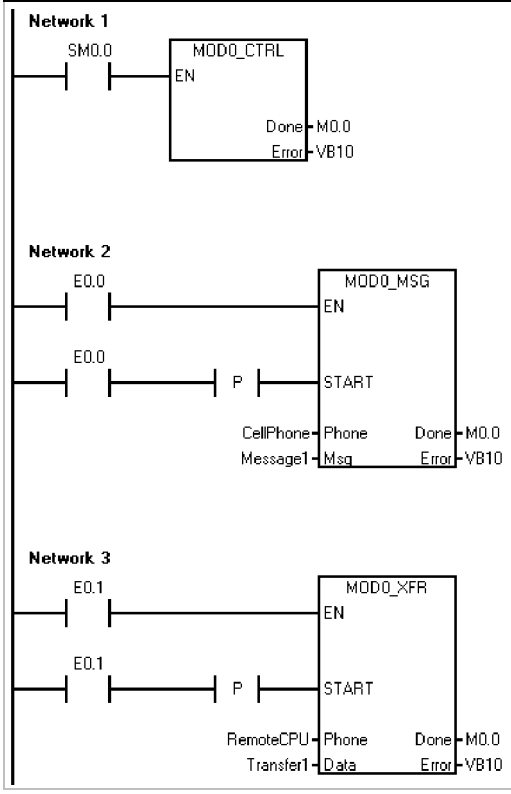
Fehler	Beschreibung
0	Kein Fehler aufgetreten
Telefonleitungsfehler	
1	Kein Wählton vorhanden
2	Leitung besetzt
3	Wählfehler
4	Keine Antwort
5	Verbindungs-Timeout (keine Verbindung innerhalb 1 Minute)
6	Verbindung abgebrochen oder unbekannte Antwort
Fehler im Befehl	
7	Nachricht für Nummernfunkruf enthält unzulässige Ziffern
8	Telefonnummer (Eingang Phone) außerhalb des Bereichs
9	Nachricht oder Datenübertragung (Eingang Msg oder Data) außerhalb des Bereichs
10	Fehler in Textnachricht oder Datenübertragungsnachricht
11	Fehler in Telefonnummer für Nachrichtenübermittlung oder Datenübertragung
12	Operation nicht zugelassen (z.b. Versuch, auf Null zu setzen)
Fehler beim Anbieter	
13	Keine Antwort (Timeout) von Nachrichtenübermittlungsanbieter
14	Verbindung zum Nachrichtenübermittlungsanbieter aus unbekanntem Grund getrennt
15	Vom Anwender abgebrochene Nachricht (Befehlsbit deaktiviert)
TAP - Textfunkruf- und SMS-Kurzmitteilungsfehler beim Anbieter	
16	Entfernte Trennung der Verbindung (Sitzungsabbruch durch Anbieter)
17	Login vom Anbieter nicht angenommen (falsches Passwort)
18	Baustein vom Nachrichtenübermittlungsanbieter nicht angenommen (Prüfsummen- oder Übertragungsfehler)
19	Baustein vom Nachrichtenübermittlungsanbieter nicht angenommen (Grund unbekannt)
UCP - SMS-Kurzmitteilungsfehler beim Anbieter	
20	Unbekannter Fehler
21	Prüfsummenfehler
22	Syntaxfehler
23	Operation vom System nicht unterstützt (unzulässiger Befehl)
24	Operation zu diesem Zeitpunkt nicht zulässig
25	Anrufsperrung aktiv (schwarze Liste)
26	Anruferadresse ungültig
27	Berechtigungsfehler
28	Legitimierungscodefehler
29	GA ungültig
30	Wiederholung unzulässig
31	Legitimierungscode für Wiederholung, Fehler
32	Prioritätsanruf nicht zulässig
33	Legitimierungscode für Prioritätsanruf, Fehler
34	Dringende Nachricht nicht zulässig
35	Legitimierungscode für dringende Nachricht, Fehler
36	Gebührenübernahme nicht zulässig
37	Legitimierungscode für Gebührenübernahme, Fehler

Tabelle 10-8 Fehlerwerte der Operationen MODx_MSG und MODx_XFR, Fortsetzung

Fehler	Beschreibung
UCP - SMS-Kurzmitteilungsfehler beim Anbieter (Fortsetzung)	
38	Zeitversetzte Lieferung nicht zugelassen
39	Neues AC nicht gültig
40	Neuer Legitimierungscode nicht zulässig
41	Standardtext nicht gültig
42	Zeitraum ungültig
43	Nachrichtentyp vom System nicht unterstützt
44	Nachricht zu lang
45	Geforderter Standardtext ungültig
46	Nachrichtentyp ungültig für Pager-Typ
47	Nachricht in SMSC nicht gefunden
48	Reserviert
49	Reserviert
50	Teilnehmer hat aufgelegt
51	Faxgruppe nicht unterstützt
52	Faxnachrichtentyp nicht unterstützt
Datenübertragungsfehler	
53	Nachrichten-Timeout (keine Antwort von entferntem Gerät)
54	Entfernte CPU durch Ladevorgang belegt
55	Zugriffsfehler (Speicher außerhalb des Bereichs, unzulässiger Datentyp)
56	Kommunikationsfehler (unbekannte Antwort)
57	Prüfsummen- oder CRC-Fehler in Antwort
58	Entferntes EM 241 für Rückruf eingerichtet (unzulässig)
59	Entferntes EM 241 hat Passwort zurückgewiesen
60 bis 127	Reserviert
Fehler in Verwendung von Operationen	
128	Diese Anforderung kann nicht bearbeitet werden. Entweder ist das Modemmodul durch die Bearbeitung einer anderen Anforderung belegt oder es gab für diese Operation keine Anforderung.
129	Modemmodulfehler: ■ Die Anordnung des Modemmoduls oder die im erweiterten Modemassistenten eingerichtete Adresse im Speicherbereich der Ausgänge entspricht nicht der tatsächlichen Anordnung bzw. der Adresse im Speicher. ■ Siehe SMB8 bis SMB21 (E/A-Modulkennung Fehlerregister).

Beispielprogramm für das Modemmodul

Beispiel: Modemmodul




```
Netzwerk 1      //Unterprogramm MOD0_CTRL
                // in jedem Zyklus aufrufen.
LD      SM0.0
CALL    MOD0_CTRL

Netzwerk 2      //Textnachricht an Mobiltelefon senden.
LD      E0.0
EU
=       L63.7
LD      E0.0
CALL    MOD0_MSG, L63.7, CellPhone, Message1,
        M0.0, VB10

Netzwerk 3      //Daten in entfernte CPU übertragen.
LD      E0.1
EU
=       L63.7
LD      E0.1
CALL    MOD0_XFR, L63.7, RemoteCPU, Transfer1,
M0.0,
        VB10
```

S7-200 CPUs, die intelligente Module unterstützen

Das Modemmodul ist ein intelligentes Erweiterungsmodul, das mit den S7-200 CPUs, die in Tabelle 10-9 aufgeführt sind, eingesetzt werden kann.

Tabelle 10-9 Kompatibilität des Modemmoduls EM 241 mit S7-200 CPUs

CPU	Beschreibung
CPU 222 ab Ausgabestand 1.10	CPU 222 DC/DC/DC
	CPU 222 AC/DC/Relais
CPU 224 ab Ausgabestand 1.10	CPU 224 DC/DC/DC
	CPU 224 AC/DC/Relais
CPU 226 ab Ausgabestand 1.00	CPU 226 DC/DC/DC
	CPU 226 AC/DC/Relais
CPU 226XM ab Ausgabestand 1.00	CPU 226XM DC/DC/DC
	CPU 226XM AC/DC/Relais

Sondermerker für das Modemmodul

Fünfzig Bytes im Speicherbereich der Sondermerker (SM) sind jedem intelligenten Modul entsprechend der physikalischen Position auf dem E/A-Erweiterungsbus zugeordnet. Wenn eine Fehlerbedingung oder eine Zustandsänderung erkannt wird, wird dies vom Modul angezeigt, indem die Sondermerker, die der Modulposition entsprechen, aktualisiert werden. Handelt es sich um das erste Modul, werden SMB200 bis SMB249 aktualisiert, um Status- und Fehlerinformationen zu melden. Das zweite Modul aktualisiert SMB250 bis SMB299 usw. (siehe Tabelle 10-10).

Hinweis

Ab Version 2.2 werden die Sondermerker für intelligente Module auf andere Art und Weise zugeordnet.

Wenn Sie mit einer CPU vor Version 2.2 arbeiten, müssen Sie alle intelligenten Module in Steckplätzen neben der CPU und vor allen nicht intelligenten Modulen anordnen, um die Kompatibilität sicherzustellen.

Tabelle 10-10 Sondermerkerbytes SMB200 bis SMB549

Sondermerkerbytes SMB200 bis SMB549						
Intelligentes Modul in Steckplatz 0	**Intelligentes Modul in Steckplatz 1**	**Intelligentes Modul in Steckplatz 2**	**Intelligentes Modul in Steckplatz 3**	**Intelligentes Modul in Steckplatz 4**	**Intelligentes Modul in Steckplatz 5**	**Intelligentes Modul in Steckplatz 6**
SMB200 bis SMB249	SMB250 bis SMB299	SMB300 bis SMB349	SMB350 bis SMB399	SMB400 bis SMB449	SMB450 bis SMB499	SMB500 bis SMB549

Tabelle10-11 zeigt den Sondermerker-Datenbereich, der dem Modemmodul zugeordnet ist. Dieser Bereich ist definiert, als ob es sich um das intelligente Modul in Steckplatz 0 des E/A-Systems handeln würde.

Tabelle 10-11 Sondermerker für das Modemmodul EM 241

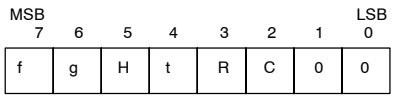
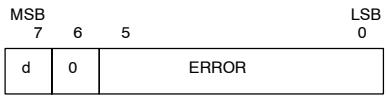
Adresse des Sondermerkers	Beschreibung
SMB200 bis SMB215	Modulname (16 ASCII-Zeichen) SMB200 ist das erste Zeichen. “EM241 Modem”
SMB216 bis SMB219	Software-Ausgabestand (4 ASCII-Zeichen) SMB216 ist das erste Zeichen.
SMW220	Fehlercode 0000 - Kein Fehler 0001 - Keine Spannung 0002 - Modemausfall 0003 - Keine Konfigurationsbaustein-ID 0004 - Konfigurationsbaustein außerhalb des Bereichs 0005 - Konfigurationsfehler 0006 - Fehlerhafte Landeskennzahl 0007 - Telefonnummer zu lang 0008 - Nachricht zu lang 0009 bis 00FF - Reserviert 01xx - Fehler in Rückrufnummer xx 02xx - Fehler in Pagernummer xx 03xx - Fehler in Nachricht Nummer xx 0400 bis FFFF - Reserviert
SMB222	Modulstatus - reflektiert den LED-Status  F - EM_FAULT 0 - kein Fehler 1 - Fehler G - EM_GOOD 0 - nicht gut 1 - gut H - OFF_HOOK 0 - aufgelegt, 1 - besetzt T - NO DIALTONE 0 - Wählton 1 - kein Wählton R - RING 0 - klingelt nicht 1 - klingelt C - CONNECT 0 - nicht verbunden 1 - verbunden

Tabelle 10-11 Sondermerker für das Modemmodul EM 241, Fortsetzung

Adresse des Sondermerkers	Beschreibung
SMB223	Landeskennzahl über Schalter eingestellt (Dezimalwert)
SMW224	Baudrate, mit der die Verbindung aufgebaut wurde (vorzeichenloser Dezimalwert).
SMB226	Ergebnis des Anwenderbefehls  D - Bit Done; 0 - in Bearbeitung 1 - fertig ERROR : Fehlercodebeschreibung, siehe Tabelle 10-8
SMB227	Auswahl Telefonnummer - Dieses Byte gibt an, welche Telefonnummer für die Nachrichtenübermittlung verwendet werden soll, wenn eine Nachricht gesendet wird. Gültig sind Werte von 1 bis 250.
SMB228	Auswahl Nachricht - Dieses Byte gibt an, welche Nachricht gesendet werden soll. Gültig sind Werte von 1 bis 250.
SMB229 bis SMB244	Reserviert
SMB245	Versatz zum ersten A-Byte, das als Befehlsschnittstelle dieses Moduls verwendet wird. Der Versatz wird zur Erleichterung des Anwenders von der CPU angegeben und wird vom Modul nicht benötigt.
SMD246	Pointer auf die Konfigurationstabelle für das Modemmodul im Variablenspeicher. Ein Pointerwert auf einen Bereich, bei dem es sich nicht um Variablenspeicher handelt, wird nicht akzeptiert, und das Modul prüft diese Adresse weiterhin, weil es auf einen Pointerwert wartet, der nicht Null ist.

Für erfahrene Anwender

Überblick über die Konfigurationstabelle

Der erweiterte Modemassistent wurde entwickelt, um Modemanwendungen zu vereinfachen, indem die Konfigurationstabelle anhand der von Ihnen eingegebenen Antworten zu Ihrem System automatisch erzeugt wird. Die Informationen in der Konfigurationstabelle dienen fortgeschrittenen Anwendern, die ihre eigenen Unterprogramme für die Steuerung von Modemmodulen erstellen und ihre eigenen Nachrichten formatieren möchten.

Die Konfigurationstabelle befindet sich im Variablenspeicher der S7-200. In Tabelle 10-12 gibt die Spalte Byteversatz den Byteversatz von der Adresse an, auf die der Pointer auf den Konfigurationsbereich im Sondermerker zeigt. Die Informationen in der Konfigurationstabelle sind in vier Abschnitte unterteilt.

- ❏ Der Konfigurationsbaustein enthält Informationen zum Einrichten des Moduls.
- ❏ Der Baustein für die Rückruftelefonnummern enthält die vordefinierten Telefonnummern für die Rückruffunktion.
- ❏ Der Baustein für die Telefonnummern für die Nachrichtenübermittlung enthält die Telefonnummern, die für die Nachrichtenübermittlung oder für CPU-Datenübertragungen gewählt werden.
- ❏ Der Nachrichtenbaustein enthält die vordefinierten Nachrichten, die an die Nachrichtenübermittlungsanbieter gesendet werden sollen.

Tabelle 10-12 Konfigurationstabelle für das Modemmodul

Konfigurationsbaustein	
Byteversatz	**Beschreibung**
0 bis 4	Modulkennung - Fünf ASCII-Zeichen für die Zuweisung der Konfigurationstabelle zu einem intelligenten Modul. Ausgabestand 1.00 des Modemmoduls EM 241 erwartet die Kennung “M241A”.
5	Länge des Konfigurationsbausteins - Derzeit 24.
6	Länge der Rückruftelefonnummer - Gültige Werte liegen im Bereich von 0 bis 40.
7	Länge der Telefonnummer für die Nachrichtenübermittlung - Gültige Werte liegen im Bereich von 0 bis 120.
8	Anzahl der Rückruftelefonnummern - Gültige Werte liegen im Bereich von 0 bis 250.
9	Anzahl der Telefonnummern für die Nachrichtenübermittlung - Gültige Werte liegen im Bereich von 0 bis 250.
10	Anzahl der Nachrichten - Gültige Werte liegen im Bereich von 0 bis 250.
11 bis 12	Reserviert (2 Bytes)
13	Dieses Byte enthält das Freigabebit für die unterstützten Funktionen. MSB 7 6 5 4 3 2 1 LSB 0 PD | CB | PW | MB | BD | 0 | 0 | 0 PD - 0 = Tonwahl 1 = Impulswahl CB - 0 = Rückruffunktion deaktiviert 1 = Rückruffunktion aktiviert PW - 0 = Passwort deaktiviert 1 = Passwort aktiviert MB - 0 = PPI-Protokoll aktiviert 1 = Modbus-Protokoll aktiviert BD - 0 = Blindwahl deaktiviert 1 = Blindwahl aktiviert Bits 2, 1 und 0 werden vom Modul ignoriert
14	Reserviert
15	Versuche - Dieser Wert gibt die Anzahl der Versuche an, die das Modem versuchen soll, zu wählen und eine Nachricht zu senden, bevor ein Fehler ausgegeben wird. Der Wert 0 verhindert, dass das Modem wählt.
16 bis 23	Passwort - Acht ASCII-Zeichen

Tabelle 10-12 Konfigurationstabelle für das Modemmodul, Fortsetzung

Baustein für die Rückruftelefonnummern (optional)	
Byteversatz	**Beschreibung**
24	Rückruftelefonnummer 1 - Eine Zeichenkette, die die erste Telefonnummer angibt, die für den Zugriff auf das Modemmodul EM 241 per Rückruf berechtigt ist. Jeder Rückruftelefonnummer muss der gleiche Speicherplatz zugewiesen werden, der für die Länge der Rückruftelefonnummer angegeben ist (Versatz 6 im Konfigurations-baustein).
24+ Rückruf-nummer	Rückruftelefonnummer 2
:	:
:	Rückruftelefonnummer n
Baustein für die Telefonnummern für die Nachrichtenübermittlung (optional)	
Byteversatz	**Beschreibung**
M	Telefonnummer für die Nachrichtenübermittlung 1 - Eine Zeichenkette, die eine Telefonnummer für die Nachrichtenübermittlung, das Protokoll und die Wähloptionen angibt. Jeder Telefonnummer muss der gleiche Speicherplatz zugewiesen werden, der für die Länge der Telefonnummer für die Nachrich-tenübermittlung angegeben ist (Versatz 7 im Konfigurationsbaustein). Das Format von Telefonnummern für die Nachrichtenübermittlung wird im folgenden beschrieben.
M + Länge der Nachrich-tenübermitt-lungsnummer	Telefonnummer für die Nachrichtenübermittlung 2
:	:
:	Telefonnummer für die Nachrichtenübermittlung n
Nachrichtenbaustein (optional)	
Byteversatz	**Beschreibung**
N	Versatz im Variablenspeicher (relativ zu VB0) für die erste Nachricht (2 Bytes)
N+2	Länge der Nachricht 1
N+3	Länge der Nachricht 2
:	
:	Länge der Nachricht n
P	Nachricht 1 - Eine Zeichenkette (max. 120 Bytes), die die erste Nachricht darstellt. Diese Zeichenkette umfasst Text und eingebettete Variablen oder sie gibt eine CPU-Datenübertragung an. Das Format für die Textnachricht und das Format für die CPU-Datenübertragung ist im folgenden beschrieben.
P + Länge von Nachricht 1	Nachricht 2
:	:
:	Nachricht n

Das Modemmodul liest die Konfigurationstabelle in den folgenden Fällen erneut:

- ❏ Innerhalb von fünf Sekunden nach einem Betriebszustandswechsel von STOP nach RUN der S7-200 CPU (sofern das Modem nicht online ist)
- ❏ Alle fünf Sekunden, bis eine gültige Konfiguration gefunden wird (sofern das Modem nich online ist)
- ❏ Jedesmal, wenn das Modem vom Online- in den Offline-Zustand wechselt

Format von Telefonnummern für die Nachrichtenübermittlung

Die Telefonnummer für die Nachrichtenübermittlung ist eine Struktur, die die Informationen enthält, die vom Modemmodul benötigt werden, um eine Nachricht zu senden. Die Telefonnummer für die Nachrichtenübermittlung ist eine ASCII-Zeichenkette mit einem führenden Längenbyte gefolgt von ASCII-Zeichen. Die maximale Länge der Telefonnummer für die Nachrichtenübermittlung beträgt 120 Bytes (einschließlich des Längenbyte).

Die Telefonnummer für die Nachrichtenübermittlung enthält bis zu 6 Felder, die durch einen Schrägstrich (/) getrennt sind. Vorwärts-/Rückwärtsschrägstriche zeigen ein Nullfeld (leer) an. Nullfelder werden im Modemmodul auf Standardwerte gesetzt.

Format: <Telefonnummer>/<ID>/<Passwort/<Protokoll>/<Standard>/<Format>

Das Telefonnummernfeld enthält die Telefonnummer, die das Modemmodul wählt, wenn eine Nachricht gesendet wird. Handelt es sich bei der zu sendenden Nachricht um einen Text oder eine SMS-Kurzmitteilung, dann ist dies die Telefonnummer des Anbieters. Handelt es sich um einen Nummernfunkruf, gibt dieses Feld die Telefonnummer des Pagers an. Handelt es sich bei der Nachricht um eine CPU-Datenübertragung, ist dies die Telefonnummer des entfernten Geräts. Die maximale Zeichenzahl in diesem Feld beträgt 40 Zeichen.

Die ID ist die Pager-Nummer oder die Mobiltelefonnummer. Dieses Feld darf nur die Ziffern 0 bis 9 enthalten. Handelt es sich bei dem Protokoll um eine CPU-Datenübertragung, wird in diesem Feld die Adresse des entfernten Geräts angegeben. Dieses Feld darf maximal 20 Zeichen enthalten.

Das Passwortfeld gibt das Passwort an für Nachrichten, die über TAP gesendet werden, wenn das Passwort vom Anbieter gefordert wird. Bei Nachrichten, die über UCP gesendet werden, wird in diesem Feld die Ursprungsadresse oder die Ursprungstelefonnummer angegeben. Handelt es sich bei der Nachricht um eine CPU-Datenübertragung an ein anderes Modemmodul, kann in diesem Feld das Passwort für das entfernte Modemmodul angegeben werden. Das Passwort kann bis zu 15 Zeichen lang sein.

Das Protokollfeld enthält ein ASCII-Zeichen, das dem Modemmodul angibt, wie die Nachricht formatiert und gesendet werden soll. Folgende Werte sind zulässig:

1 - Nummernfunkrufprotokoll (Voreinstellung)
2 - TAP
3 - UCP-Befehl 1
4 - UCP-Befehl 30
5 - UCP-Befehl 51
6 - CPU-Datenübertragung

Das Standardfeld zwingt das Modemmodul, einen bestimmten Modemstandard einzusetzen. Das Standardfeld enthält ein ASCII-Zeichen. Folgende Werte sind zulässig:

1 - Bell 103
2 - Bell 212
3 - V.21
4 - V.22
5 - V.22 Bit
6 - V.23c
7 - V.32
8 - V.32 Bit
9 - V.34 (Voreinstellung)

Das Formatfeld besteht aus drei ASCII-Zeichen, die die Anzahl Datenbits und die Parität zum Senden von Nachrichten angeben. Dieses Feld ist ungültig, wenn für das Protokoll der Nummernfunkruf eingestellt ist. Nur die folgenden beiden Einstellungen sind zulässig:

8N1 - 8 Datenbits, keine Parität, ein Stoppbit (Voreinstellung)
7E1 - 7 Datenbits, gerade Parität, ein Stoppbit

Format von Textnachrichten

Das Textnachrichtenformat definiert das Format von Textfunkrufen oder SMS-Kurzmitteilungen. Diese Arten von Nachrichten können Text und eingebettete Variablen enthalten. Die Textnachricht ist eine ASCII-Zeichenkette mit einem führenden Längenbyte gefolgt von ASCII-Zeichen. Die maximale Länge der Textnachricht beträgt 120 Bytes (einschließlich des Längenbyte).

Format: <Text><Variable><Text><Variable>...

Das Textfeld besteht aus ASCII-Zeichen.

Das Variablenfeld definiert einen eingebetteten Datenwert, den das Modemmodul aus der lokalen CPU liest, formatiert und in die Nachricht einfügt. Das Prozentzeichen (%) kennzeichnet den Beginn und das Ende eines Variablenfelds. Die Adresse und die Felder links werden durch einen Doppelpunkt getrennt. Der Begrenzer zwischen den Feldern ”Links” und ”Rechts” kann ein Punkt oder ein Komma sein und wird in der formatierten Variable als Dezimalpunkt verwendet. Das Variablenfeld hat folgende Syntax:

%Adresse:Format Links.Rechts%

Das Adressfeld gibt die Adresse, den Datentyp und die Größe des eingebetteten Datenwerts an (z.B. VD100, VW50, MB20 oder T10). Folgende Datentypen sind zugelassen: E, A, M, SM, V, T (nur Wort), Z (nur Wort) und AE (nur Wort). Byte-, Wort- und Doppelwortgrößen sind möglich.

Das Feld ”Links” definiert die Anzahl der Ziffern, die links vom Dezimalpunkt angezeigt werden. Dieser Wert muss groß genug sein, um den erwarteten Bereich der eingebetteten Variable einschließlich Vorzeichen (sofern erforderlich) zu verarbeiten. Ist das Feld ”Links” Null, wird der Wert mit einer führenden Null angezeigt. Der gültige Bereich liegt zwischen 0 und 10.

Das Feld ”Rechts” definiert die Anzahl der Ziffern, die rechts vom Dezimalpunkt angezeigt werden. Nullen rechts vom Dezimalpunkt werden immer angezeigt. Ist das Feld ”Rechts” Null, wird der Wert ohne Dezimalpunkt angezeigt. Der gültige Bereich liegt zwischen 0 und 10.

Das Formatfeld gibt das Anzeigeformat des eingebetteten Werts an. Die folgenden Zeichen sind im Formatfeld zugelassen:

i - Ganzzahl mit Vorzeichen
u - Vorzeichenlose Ganzzahl
h - Hexadezimal
f - Gleitpunkt/Real

Beispiel: “Temperatur = %VW100:3,1i% Druck = %VD200:4,3f%”

Format von CPU-Datenübertragungsnachrichten

Eine CPU-Datenübertragung, entweder von CPU zu CPU oder von CPU zu Modbus, wird im Format von CPU-Datenübertragungsnachrichten angegeben. Eine CPU-Datenübertragungsnachricht ist eine ASCII-Zeichenkette, die eine beliebige Anzahl an Datenübertragungen zwischen Geräten angeben kann, und zwar bis zu der maximalen Anzahl, die sich in der maximalen Nachrichtenlänge von 120 Bytes (119 Zeichen plus ein Längenbyte) unterbringen lassen. Mit einem ASCII-Leerzeichen können die einzelnen Datenübertragungen getrennt werden, doch dies ist nicht unbedingt erforderlich. Alle angegebenen Datenübertragungen werden in einer Verbindung ausgeführt. Datenübertragungen werden in der Reihenfolge ausgeführt, die in der Nachricht definiert ist. Wird in einer Datenübertragung ein Fehler erkannt, wird die Verbindung zum entfernten Gerät beendet und nachfolgende Transaktionen werden nicht bearbeitet.

Wird eine Leseoperation angegeben, wird die im Zählwert angegebene Anzahl Wörter aus dem entfernten Gerät mit Beginn an der entfernten Adresse gelesen und in den Variablenspeicher in der lokalen CPU mit Beginn an der lokalen Adresse geschrieben.

Wird eine Schreiboperation angegeben, wird die im Zählwert angegebene Anzahl Wörter in der lokalen CPU mit Beginn an der lokalen Adresse gelesen und ins entfernte Gerät mit Beginn an der entfernten Adresse geschrieben.

Format: <Operation>=<Zählwert>,<lokale_Adresse>,<entfernte_Adresse>

Das Operationsfeld enthält ein ASCII-Zeichen, das die Art der Übertragung angibt.

R - Daten aus dem entfernten Gerät lesen
W - Daten ins entfernte Gerät schreiben

Der Zählwert gibt die Anzahl der Wörter an, die übertragen werden sollen. Der gültige Bereich für den Zählwert liegt zwischen 1 Wort und 100 Wörtern.

Das Feld ”Lokale_Adresse” gibt die Adresse im Variablenspeicher der lokalen CPU für die Datenübertragung an (z.B. VW100).

Das Feld ”Entfernte_Adresse” gibt die Adresse im entfernten Gerät für die Datenübertragung an (z.B. VW500). Diese Adresse wird immer als Adresse im Variablenspeicher angegeben, auch wenn die Datenübertragung zu einem Modbus-Gerät stattfindet. Handelt es sich bei dem entfernten Gerät um ein Modbus-Gerät, wird die Variablenspeicheradresse wie folgt in eine Modbus-Adresse umgewandelt:

Modbus-Adresse = 1 + (Adresse im Variablenspeicher / 2)
Adresse im Variablenspeicher = (Modbus-Adresse - 1) * 2

Beispiel: R=20,VW100, VW200 W=50,VW500,VW1000 R=100,VW1000,VW2000

11 Steuern eines MicroMaster-Antriebs mit der Bibliothek für das USS-Protokoll

Durch die Operationsbibliotheken in STEP 7-Micro/WIN wird die Steuerung von MicroMaster-Antrieben einfacher. STEP 7-Micro/WIN bietet Ihnen vorkonfigurierte Unterprogramme und Interruptprogramme, die speziell für die Kommunikation mit dem Antrieb über das USS-Protokoll ausgelegt sind. Mit diesen neuen USS-Operationen können Sie den physikalischen Antrieb und die Parameter zum Lesen und Schreiben des Antriebs steuern.

Diese Operationen finden Sie im Ordner Bibliotheken im Operationsbaum von STEP 7-Micro/WIN. Wenn Sie eine USS-Operation auswählen, wird automatisch ein zugehöriges Unterprogramm (oder mehrere) eingefügt (USS1 bis USS7).

In diesem Kapitel

Anforderungen für den Einsatz des USS-Protokolls 322
Berechnen der Zeit für die Kommunikation mit dem Antrieb 323
Verwenden der USS-Operationen 324
Operationen für das USS-Protokoll 325
Beispielprogramme für das USS-Protokoll 332
Fehlercodes für die Ausführung der USS-Operationen 333
Anschließen und Einrichten des MicroMaster-Antriebs der Serie 3 334
Anschließen und Einrichten des MicroMaster-Antriebs der Serie 4 337

Anforderungen für den Einsatz des USS-Protokolls

Die Operationsbibliothek von STEP 7-Micro/WIN bieten 14 Unterprogramme, 3 Interruptprogramme und 8 Operationen zur Unterstützung des USS-Protokolls. Die USS-Operationen nutzen die folgenden Ressourcen der S7-200:

- Durch die Initialisierung des USS-Protokolls wird die Schnittstelle 0 für die USS-Kommunikation bereitgestellt.

  Mit der Operation USS_INIT wählen Sie entweder USS oder PPI für die Schnittstelle 0 aus. (USS bezieht sich auf das USS-Protokoll für SIMOTION MicroMaster-Antriebe.) Wenn Sie das USS-Protokoll für die Kommunikation mit den Antrieben eingestellt haben, können Sie die Schnittstelle 0 nicht für andere Zwecke verwenden, auch nicht für die Kommunikation mit STEP 7-Micro/WIN.

  Wenn Sie ein Programm für eine Anwendung mit dem USS-Protokoll entwickeln, müssen Sie eine CPU 226, CPU 226XM oder ein EM 277 PROFIBUS-DP-Modul verwenden, die bzw. das an eine PROFIBUS CP-Karte in Ihrem PC angeschlossen ist. So kann STEP 7-Micro/WIN über die zweite Kommunikationsschnittstelle die Anwendung beobachten, während das USS-Protokoll in Betrieb ist.

- Die USS-Operationen wirken sich auf alle Sondermerker aus, die der frei programmierbaren Kommunikation an Schnittstelle 0 zugeordnet sind.
- Die USS-Operationen nutzen 14 Unterprogramme und 3 Interruptprogramme.
- Die USS-Operationen erhöhen den Speicherbedarf für Ihr Programm um bis zu 3450 Bytes. Je nach den verwendeten USS-Operationen können die unterstützenden Unter- und Interruptprogramme das Steuerungsprogramm um mindestens 2150 Bytes und um maximal 3450 Bytes erhöhen.
- Die Variablen für die USS-Operationen benötigen einen 400 Byte großen Baustein im Variablenspeicher. Die Anfangsadresse dieses Bausteins wird vom Anwender zugewiesen und ist für USS-Variablen reserviert.
- Einige USS-Operationen benötigen außerdem einen 16 Byte großen Kommunikationspuffer. Als Parameter für die Operation geben Sie eine Anfangsadresse im Variablenspeicher für diesen Puffer an. Es ist empfehlenswert, für jede Instanz der USS-Operationen einen eindeutigen Puffer zuzuordnen.
- Für Berechnungen nutzen die USS-Operationen die Akkumulatoren AC0 bis AC3. Sie können die Akkumulatoren auch in Ihrem Programm verwenden, doch die Werte in den Akkumulatoren werden von den USS-Operationen geändert.
- Die USS-Operationen dürfen nicht in Interruptprogrammen verwendet werden.

Tipp

Wenn Sie für den Betrieb von Schnittstelle 0 wieder PPI einstellen möchten, so dass Sie mit STEP 7-Micro/WIN kommunizieren können, müssen Sie die Schnittstelle 0 mit einer anderen Operation USS_INIT erneut zuweisen.

Sie können den Betriebsartenschalter auf der S7-200 auch in den Betriebszustand STOP versetzen. Dadurch werden die Parameter für die Schnittstelle 0 zurückgesetzt. Beachten Sie, dass auch die Antriebe gestoppt werden, wenn Sie die Kommunikation zu den Antrieben stoppen.

Berechnen der Zeit für die Kommunikation mit dem Antrieb

Die Kommunikation mit dem Antrieb läuft asynchron zum Zyklus der S7-200 ab. Die S7-200 durchläuft üblicherweise mehrere Zyklen, bevor eine Transaktion für die Antriebskommunikation beendet wird. Die folgenden Faktoren helfen dabei, die erforderliche Zeit zu ermitteln: die Anzahl der vorhandenen Antriebe, die Baudrate und die Zykluszeit der S7-200.

Einige Antriebe benötigen längere Verzögerungen, wenn Operationen mit Parameterzugriff verwendet werden. Der Zeitbedarf für den Parameterzugriff richtet sich nach der Art des Antriebs und danach, auf welchen Parameter zugegriffen wird.

Hat die Operation USS_INIT der Schnittstelle 0 das USS-Protokoll zugeordnet, fragt die S7-200 regelmäßig alle aktiven Antriebe in den in Tabelle 11-1 aufgeführten Abständen ab. Sie müssen den Timeout-Parameter der Antriebe setzen, damit diese Aufgabe durchgeführt werden kann.

Tabelle 11-1 Kommunikationszeiten

Baudrate	Zeit zwischen den Abfragen der aktiven Antriebe (keine Operationen mit Parameterzugriff aktiv)
1200	240 ms (max.) multipliziert mit der Anzahl der Antriebe
2400	130 ms (max.) multipliziert mit der Anzahl der Antriebe
4800	75 ms (max.) multipliziert mit der Anzahl der Antriebe
9600	50 ms (max.) multipliziert mit der Anzahl der Antriebe
19.200	35 ms (max.) multipliziert mit der Anzahl der Antriebe
38400	30 ms (max.) multipliziert mit der Anzahl der Antriebe
57600	25 ms (max.) multipliziert mit der Anzahl der Antriebe
115200	25 ms (max.) multipliziert mit der Anzahl der Antriebe

Tipp

Es darf immer nur eine Operation USS_RPM_x oder USS_WPM_x zur Zeit aktiv sein. Der Ausgang Done einer Operation muss das Ende der Bearbeitung anzeigen, bevor das Anwenderprogramm die nächste Operation initiiert.

Verwenden Sie nur eine Operation USS_CTRL pro Antrieb.

Verwenden der USS-Operationen

Wenn Sie die USS-Operationen in Ihrem S7-200 Programm verwenden möchten, gehen Sie folgendermaßen vor:

1. Fügen Sie die Operation USS_INIT in Ihr Programm ein. Führen Sie die Operation USS_INIT nur in einem Zyklus aus. Mit der Operation USS_INIT können Sie die USS-Kommunikationsparameter initialisieren oder ändern.

   Wenn Sie die Operation USS_INIT einfügen, werden verschiedene versteckte Unterprogramme und Interruptprogramme automatisch in Ihr Programm aufgenommen.

2. Fügen Sie für jeden aktiven Antrieb in Ihrem Programm nur eine Operation USS_CTRL ein.

   Sie können so viele Operationen USS_RPM_x und USS_WPM_x einfügen, wie Sie benötigen, doch es darf immer nur eine der Operationen zur Zeit aktiv sein.

3. Ordnen Sie den Variablenspeicher für die Bibliotheksoperationen zu, indem Sie mit der rechten Maustaste auf das Symbol des Programmbausteins im Operationsbaum klicken (zum Aufblenden des Kontextmenüs).

   Wählen Sie die Option ”Speicher für Bibliothek”, um das Dialogfeld ”Speicher für Bibliothek” aufzurufen.

Bild 11-1 Zuordnen von Variablenspeicher für die Operationsbibliothek

4. Konfigurieren Sie die Antriebsparameter so, dass diese der im Programm verwendeten Baudrate und Adresse entsprechen.

5. Schließen Sie die Kommunikationskabel an die S7-200 und an die Antriebe an.

   Achten Sie darauf, dass alle Steuerungsgeräte, die an den Antrieb angeschlossen sind (z.B. die S7-200) durch ein kurzes, dickes Kabel mit der gleichen Erde oder dem gleichen Sternpunkt verbunden sind wie der Antrieb.

Vorsicht

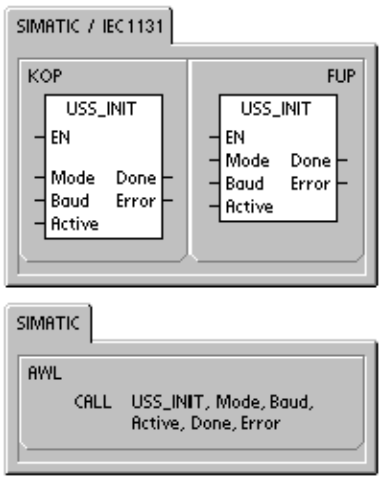
Wenn Sie Geräte miteinander verbinden, die nicht die gleiche Bezugsspannung haben, kann dies unerwünschte Ströme im Verbindungskabel hervorrufen. Diese unerwünschten Ströme können Kommunikationsfehler verursachen oder Sachschaden in den Geräten hervorrufen.

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.

Die Schirmung muss mit Masse oder Pin 1 des 9-poligen Steckverbinders verbunden werden. Die Verdrahtungsklemme 2-0V des MicroMaster-Antriebs sollte mit Masse verbunden werden.

Operationen für das USS-Protokoll

Operation USS_INIT



Mit der Operation USS_INIT wird die Kommunikation zum MicroMaster-Antrieb aktiviert und initialisiert oder deaktiviert. Bevor eine andere USS-Operation verwendet werden kann, muss die Operation USS_INIT fehlerfrei ausgeführt werden. Die Operation wird beendet und das Bit Done wird sofort gesetzt, bevor die nächste Operation ausgeführt wird.

Die Operation wird in jedem Zyklus ausgeführt, wenn der Eingang EN eingeschaltet ist.

Führen Sie die Operation USS_INIT bei jeder Änderung des Kommunikationszustands genau einmal aus. Schalten Sie den Eingang EN durch die Erkennung einer steigenden Flanke ein. Zum Ändern der Initialisierungsparameter führen Sie eine neue Operation USS_INIT aus.

Der Wert von Mode stellt das Kommunikationsprotokoll ein: Der Eingangswert 1 weist der Schnittstelle 0 das USS-Protokoll zu und aktiviert das Protokoll. Der Eingangswert 0 weist der Schnittstelle 0 das PPI-Protokoll zu und deaktiviert das USS-Protokoll.

Der Parameter Baud stellt eine Baudrate von 1200, 2400, 4800, 9600, 19200, 38400, 57600 oder 115200 ein.

ACTIVE zeigt die aktiven Antriebe an. Einige Antriebe unterstützen nur die Adressen 0 bis 30.

Tabelle 11-2 Parameter der Operation USS_INIT

Eingänge/Ausgänge	Datentyp	Operanden
Modus	BYTE	VB, EB, AB, MB, SB, SMB, LB, AC, Konstante, *VD, *AC, *LD
Baud, Active	DWORD	VD, ED, AD, MD, SD, SMD, LD, Konstante, AC *VD, *AC, *LD
Done	BOOL	E, A, M, S, SM, T, Z, V, L
Error	BYTE	VB, EB, AB, MB, SB, SMB, LB, AC, *VD, *AC, *LD

Bild 11-2 zeigt eine Beschreibung und das Format des Eingangs für den aktiven Antrieb**.** Jeder Antrieb, der als aktiv gekennzeichnet ist, wird automatisch im Hintergrund abgefragt, um den Antrieb zu steuern, den Status zu erfassen und Timeouts der seriellen Verbindung im Antrieb zu verhindern.

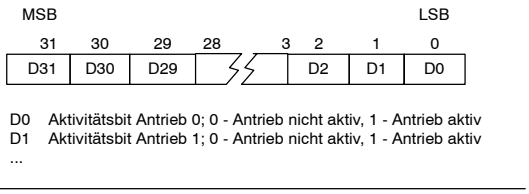
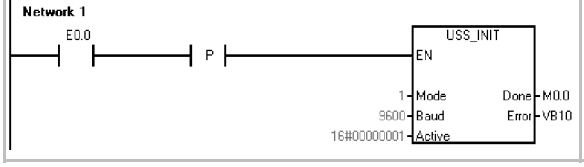


Bild 11-2 Format des Parameters für aktiven Antrieb

Berechnen Sie mit Hilfe der Tabelle 11-1 die Zeit zwischen den Statusabfragen.

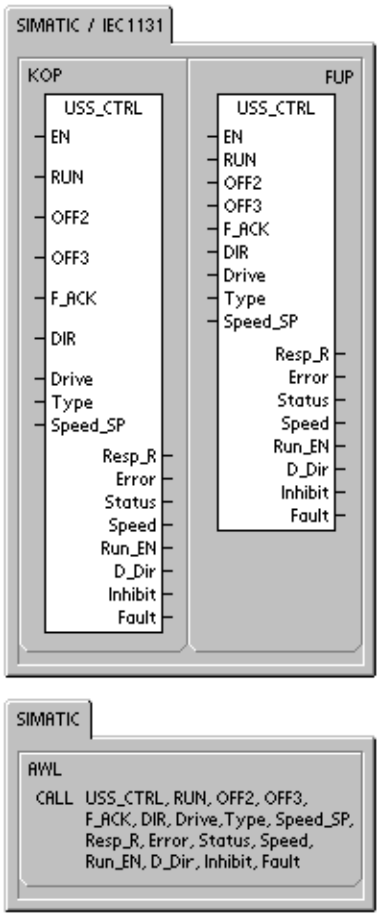
Ist die Bearbeitung der Operation USS_INIT beendet, wird der Ausgang Done eingeschaltet. Das Ausgangsbyte Error enthält das Ergebnis der Ausführung der Operation. Tabelle 11-6 zeigt die Fehlerbedingungen, die während der Ausführung der Operationen auftreten können.

Beispiel: Unterprogramm USS_INIT




```
Netzwerk 1
LD      E0.0
EU
CALL    USS_INIT, 1, 9600, 16#00000001, M0.0, VB10
```

Operation USS_CTRL



Mit der Operation USS_CTRL wird ein aktiver MicroMaster-Antrieb gesteuert. Die Operation USS_CTRL legt die ausgewählten Befehle in einem Kommunikationspuffer ab, der dann an den adressierten Antrieb (Parameter Drive) gesendet wird, sofern dieser Antrieb im Parameter Active der Operation USS_INIT eingestellt ist.

Jedem Antrieb darf nur eine Operation USS_CTRL zugeordnet werden.

Einige Antriebe geben die Drehzahl nur als positiven Wert an. Ist die Drehzahl negativ, gibt der Antrieb die Drehzahl als positiven Wert an und kehrt das Bit D_DIR (Richtung) um.

Das Bit EN muss eingeschaltet sein, um die Operation USS_CTRL zu aktivieren. Diese Operation muss immer aktiviert sein.

RUN (RUN/STOP) zeigt an, ob der Antrieb ein- (1) oder ausgeschaltet (0) ist. Ist das Bit RUN eingeschaltet, empfängt der MicroMaster-Antrieb einen Befehl für den Betrieb mit bestimmter Drehzahl und Laufrichtung. Damit der Antrieb anläuft, müssen folgende Voraussetzungen erfüllt sein:

- ❏ Der Antrieb (Drive) muss in der Operation USS_INIT aktiv (Active) sein.
- ❏ OFF2 und OFF3 müssen auf 0 gesetzt sein.
- ❏ Fault und Inhibit müssen 0 sein.

Ist RUN ausgeschaltet, wird dem MicroMaster-Antrieb ein Befehl geschickt, die Motordrehzahl herunterzufahren, bis der Motor stoppt. Das Bit OFF2 lässt den MicroMaster-Antrieb bis zum Stopp auslaufen. Mit dem Bit OFF3 wird der MicroMaster-Antrieb angewiesen, den Motor schnell zu stoppen.

Das Bit Resp_R (Antwort empfangen) quittiert eine Reaktion des Antriebs. Alle aktiven (ACTIVE) Antriebe werden nach den aktuellen Statusinformationen abgefragt. Jedesmal, wenn die S7-200 eine Reaktion des Antriebs erhält, wird das Bit Resp_R einen Zyklus lang eingeschaltet und alle folgenden Werte werden aktualisiert.

Das Bit F_ACK (Fehlerquittierung) quittiert einen Fehler im Antrieb. Der Antrieb löscht den Fehler (FAULT), wenn F_ACK von 0 auf 1 gesetzt wird.

Das Bit DIR (Richtung) gibt die Drehrichtung des Antriebs an.

Der Eingang Drive (Antriebsadresse) ist die Adresse des MicroMaster-Antriebs, an den der Befehl USS_CTRL gesendet werden soll. Gültige Adressen: 0 bis 31

Der Eingang Type (Antriebstyp) stellt die Art des Antriebs ein. Bei einem Antrieb MicroMaster 3 (oder früher) stellen Sie für Type 0 ein. Bei einem Antrieb MicroMaster 4 stellen Sie Type auf 1 ein.

Tabelle 11-3 Parameter der Operation USS_CTRL

Eingänge/Ausgänge	Datentypen	Operanden
RUN, OFF 2, OFF 3, F_ACK, DIR	BOOL	E, A, M, S, SM, T, Z, V, L, Signalfluss
Resp_R, Run_EN, D_Dir, Inhibit, Fault	BOOL	E, A, M, S, SM, T, Z, V, L
Drive, Type	BYTE	VB, EB, AB, MB, SB, SMB, LB, AC, *VD, *AC, *LD, Konstante
Error	BYTE	VB, EB, AB, MB, SB, SMB, LB, AC, *VD, *AC, *LD
Status	WORD	VW, T, Z, EW, AW, SW, MW, SMW, LW, AC, AAW, *VD, *AC, *LD
Speed_SP	REAL	VD, ED, AD, MD, SD, SMD, LD, AC, *VD, *AC, *LD, Konstante
Speed	REAL	VD, ED, AD, MD, SD, SMD, LD, AC, *VD, *AC, *LD

Speed_SP (Sollwert Drehzahl) gibt die Drehzahl des Antriebs als Prozentwert der Höchstdrehzahl an. Negative Werte von Speed_SP wechseln die Drehrichtung des Antriebs. Bereich: -200,0% bis 200,0%.

Error ist ein Fehlerbyte, das das Ergebnis der letzten Kommunikationsanforderung an den Antrieb enthält. Tabelle 11-6 zeigt die Fehlerbedingungen, die während der Ausführung der Operationen auftreten können.

STATUS ist der Rohwert des Statusworts, das vom Antrieb ausgegeben wird. Bild 11-3 zeigt die Statusbits für das Standard-Statuswort und die Hauptrückmeldung.

Speed ist die Drehzahl des Antriebs als Prozentwert der Höchstdrehzahl. Bereich: -200,0% bis 200,0%.

RUN_EN (RUN aktivieren) gibt an, ob der Antrieb in Betrieb (1) oder gestoppt (0) ist.

D_Dir gibt die Drehrichtung des Antriebs an.

INHIBIT zeigt den Zustand des Sperrbits des Antriebs an (0 - nicht gesperrt, 1 - gesperrt). Zum Löschen des Bits INHIBIT muss das Bit FAULT ausgeschaltet sein und die Eingänge RUN, OFF2 und OFF3 müssen auch ausgeschaltet sein.

FAULT zeigt den Zustand des Fehlerbits an (0 - kein Fehler, 1 - Fehler). Der Antrieb zeigt den Fehlercode an. (Hinweise hierzu entnehmen Sie dem Handbuch zu Ihrem Antrieb). Zum Löschen des Bits FAULT beheben Sie die Fehlerursache und schalten das Bit F_ACK ein.

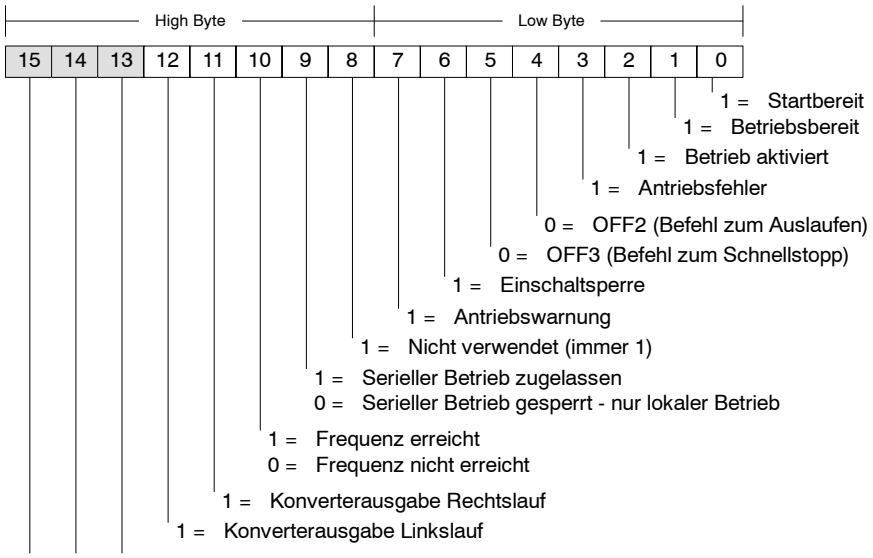


Bild 11-3 Statusbits für das Standard-Statuswort für MicroMaster 3 und die Hauptrückmeldung

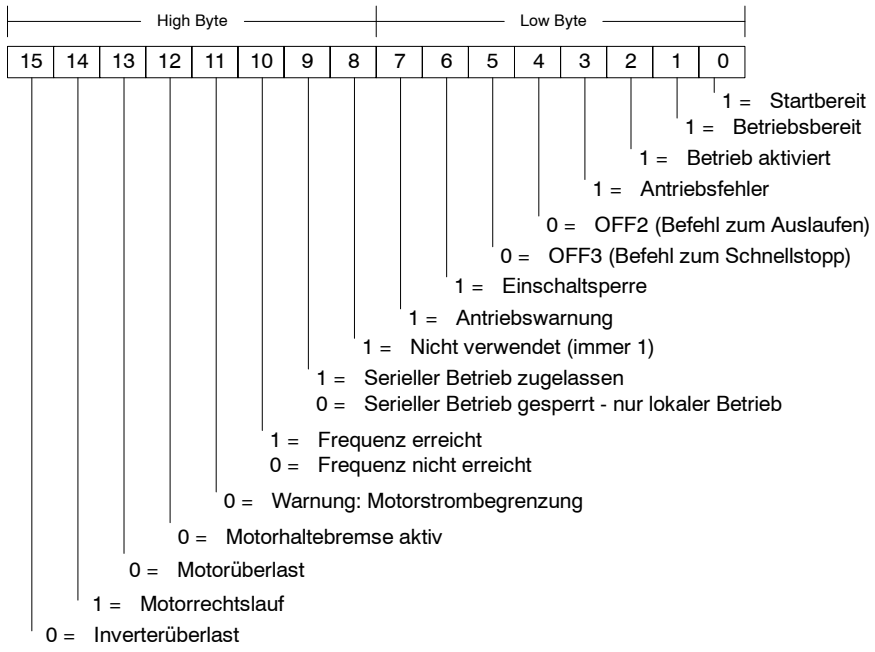
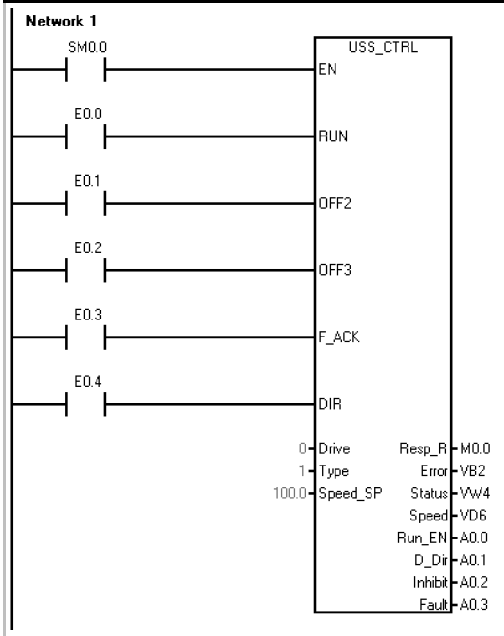


Bild 11-4 Statusbits für das Standard-Statuswort für MicroMaster 4 und die Hauptrückmeldung

Beispiel: Unterprogramm USS_CTRL



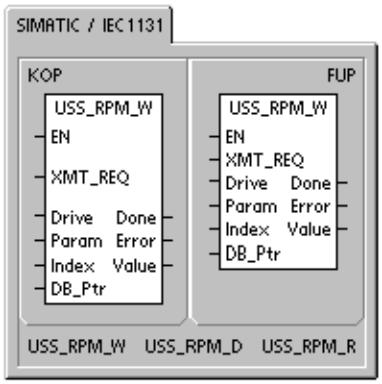
Anzeige nur in AWL:

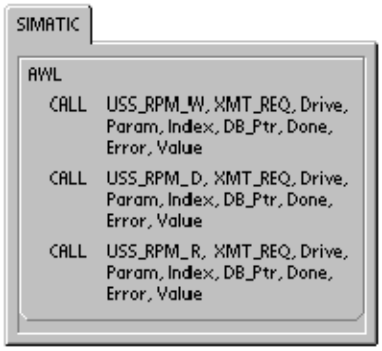
```
Netzwerk 1      //Steuerbox für Antrieb 0
LD          SM0.0
CALL        USS_CTRL, E0.0, E0.1, E0.2, E0.3, E0.4, 0, 1, 100.0, M0.0, VB2, VW4, VD6, A0.0, A0.1, A0.2, A0.3
```

Anzeige in KOP oder FUP:

```
Netzwerk 1      //Steuerbox für Antrieb 0
LD          SM0.0
=           L60.0
LD          E0.0
=           L63.7
LD          E0.1
=           L63.6
LD          E0.2
=           L63.5
LD          E0.3
=           L63.4
LD          E0.4
=           L63.3
LD          L60.0
CALL        USS_CTRL, L63.7, L63.6, L63.5, L63.4, L63.3, 0, 1, 100.0, M0.0, VB2, VW4, VD6, A0.0, A0.1, A0.2, A0.3
```

Operation USS_RPM_x





Es gibt drei Leseoperationen für das USS-Protokoll:

- Die Operation USS_RPM_W liest einen vorzeichenlosen Wortparameter.
- Die Operation USS_RPM_D liest einen vorzeichenlosen Doppelwortparameter.
- Die Operation USS_RPM_R liest einen Gleitpunktparameter.

Es darf immer nur eine Leseoperation (USS_RPM_x) oder eine Schreiboperation (USS_WPM_x) zur Zeit aktiv sein.

Die Transaktionen USS_RPM_x werden beendet, wenn der MicroMaster-Antrieb den Empfang des Befehls quittiert oder wenn eine Fehlerbedingung gemeldet wird. Der Zyklus wird weiter ausgeführt, während dieser Prozess auf eine Reaktion wartet.

Das Bit EN muss eingeschaltet sein, damit eine Anforderung gesendet werden kann. Es muss so lange eingeschaltet bleiben, bis das Bit Done gesetzt und dadurch das Ende der Bearbeitung angezeigt wird. Beispiel: In jedem Zyklus, in dem der Eingang XMT_REQ eingeschaltet ist, wird eine Anforderung USS_RPM_x an den MicroMaster-Antrieb gesendet. Deshalb muss der Eingang XMT_REQ durch die Erkennung einer Flanke eingeschaltet werden, weil so bei jeder steigenden Flanke am Eingang EN eine Anforderung gesendet wird.

Der Eingang Drive ist die Adresse des MicroMaster-Antriebs, an den der Befehl USS_RPM_x gesendet werden soll. Gültige Adressen einzelner Antriebe liegen zwischen 0 und 31.

Param ist die Nummer des Parameters. Index ist der Indexwert des Parameters, der gelesen werden soll. Value ist der ausgegebene Parameterwert. Am Eingang DB_Ptr muss die Adresse eines 16 Byte langen Puffers angegeben werden. In diesem Puffer speichert die Operation USS_RPM_x die Ergebnisse des Befehls an den MicroMaster-Antrieb.

Ist die Bearbeitung der Operation USS_RPM_x beendet, wird der Ausgang Done eingeschaltet und das Ausgangsbyte Error und der Ausgang Value enthalten das Ergebnis der Ausführung der Operation. Tabelle 11-6 zeigt die Fehlerbedingungen, die während der Ausführung der Operationen auftreten können. Die Ausgänge Error und Value sind erst gültig, wenn der Ausgang Done eingeschaltet wird.

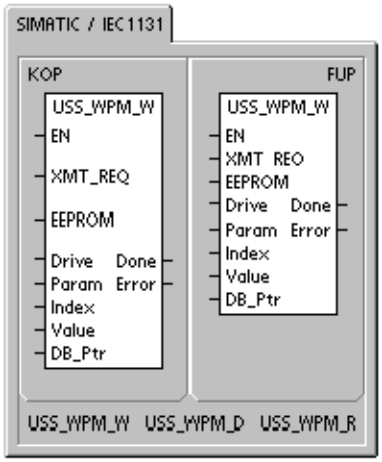
Tabelle 11-4 Gültige Operanden für USS_RPM_x

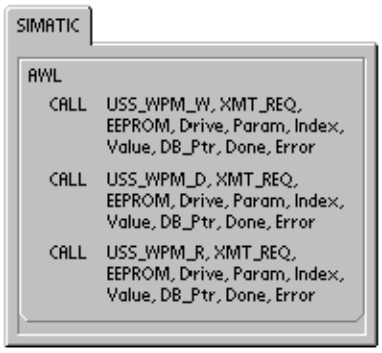
Eingänge / Ausgänge	**Datentyp**	**Operanden**
XMT_REQ	BOOL	E, A, M, S, SM, T, Z, V, L, Signalfluss bedingt durch die Erkennung einer steigenden Flanke
Drive	BYTE	VB, EB, AB, MB, SB, SMB, LB, AC, *VD, *AC, *LD, Konstante
Param, Index	WORD	VW, EW, AW, MW, SW, SMW, LW, T, Z, AC, AEW, *VD, *AC, *LD, Konstante
DB_Ptr	DWORD	&VB
Value	WORD DWORD, REAL	VW, EW, AW, MW, SW, SMW, LW, T, Z, AC, AAW, *VD, *AC, *LD VD, ED, AD, MD, SD, SMD, LD, *VD, *AC, *LD
Done	BOOL	E, A, M, S, SM, T, Z, V, L
Error	BYTE	VB, EB, AB, MB, SB, SMB, LB, AC. *VD, *AC, *LD

Operation USS_WPM_x

Es gibt drei Schreiboperationen für das USS-Protokoll:

- Die Operation USS_WPM_W schreibt einen vorzeichenlosen Wortparameter.
- Die Operation USS_WPM_D schreibt einen vorzeichenlosen Doppelwortparameter.
- Die Operation USS_WPM_R schreibt einen Gleitpunktparameter.





Es darf immer nur eine Leseoperation (USS_RPM_x) oder eine Schreiboperation (USS_WPM_x) zur Zeit aktiv sein.

Die Transaktionen USS_WPM_x werden beendet, wenn der MicroMaster-Antrieb den Empfang des Befehls quittiert oder wenn eine Fehlerbedingung gemeldet wird. Der Zyklus wird weiter ausgeführt, während dieser Prozess auf eine Reaktion wartet.

Das Bit EN muss eingeschaltet sein, damit eine Anforderung gesendet werden kann. Es muss so lange eingeschaltet bleiben, bis das Bit Done gesetzt und dadurch das Ende der Bearbeitung angezeigt wird. Beispiel: In jedem Zyklus, in dem der Eingang XMT_REQ eingeschaltet ist, wird eine Anforderung USS_WPM_x an den MicroMaster-Antrieb gesendet. Deshalb muss der Eingang XMT_REQ durch die Erkennung einer Flanke eingeschaltet werden, weil so bei jeder steigenden Flanke am Eingang EN eine Anforderung gesendet wird.

Der Eingang Drive ist die Adresse des MicroMaster-Antriebs, an den der Befehl USS_WPM_x gesendet werden soll. Gültige Adressen einzelner Antriebe liegen zwischen 0 und 31.

Param ist die Nummer des Parameters. Index ist der Indexwert des Parameters, der geschrieben werden soll. Value ist der Parameterwert, der in den RAM im Antrieb geschrieben werden soll. Bei Antrieben MicroMaster 3 können Sie diesen Wert auch in den EEPROM des Antriebs schreiben, je nachdem, wie Sie P971 (EEPROM-Speichersteuerung) konfiguriert haben.

Am Eingang DB_Ptr muss die Adresse eines 16 Byte langen Puffers angegeben werden. In diesem Puffer speichert die Operation USS_WPM_x die Ergebnisse des Befehls an den MicroMaster-Antrieb.

Ist die Bearbeitung der Operation USS_WPM_x beendet, wird der Ausgang Done eingeschaltet und das Ausgangsbyte Error enthält das Ergebnis der Ausführung der Operation. Tabelle 11-6 zeigt die Fehlerbedingungen, die während der Ausführung der Operationen auftreten können.

Ist der Eingang EEPROM eingeschaltet, schreibt die Operation in den RAM und den EEPROM des Antriebs. Ist der Eingang ausgeschaltet, schreibt die Operation nur in den RAM des Antriebs. Weil der Antrieb MicroMaster 3 diese Funktion nicht unterstützt, müssen Sie darauf achten, dass der Eingang ausgeschaltet ist, wenn Sie diese Operation mit einem Antrieb MicroMaster 3 verwenden möchten.

Tabelle 11-5 Gültige Operanden für die Operationen USS_WPM_x

Eingänge / Ausgänge	Datentyp	Operanden
XMT_REQ	BOOL	E, A, M, S, SM, T, Z, V, L, Signalfluss bedingt durch die Erkennung einer steigenden Flanke
EEPROM	BOOL	E, A, M, S, SM, T, Z, V, L, Signalfluss
Drive	BYTE	VB, EB, AB, MB, SB, SMB, LB, AC, *VD, *AC, *LD, Konstante
Param, Index	WORD	VW, EW, AW, MW, SW, SMW, LW, T, Z, AC, AEW, *VD, *AC, *LD, Konstante
DB_Ptr	DWORD	&VB
Value	WORD DWORD, REAL	VW, EW, AW, MW, SW, SMW, LW, T, Z, AC, AAW, *VD, *AC, *LD VD, ED, AD, MD, SD, SMD, LD, *VD, *AC, *LD
Done	BOOL	E, A, M, S, SM, T, Z, V, L
Error	BYTE	VB, EB, AB, MB, SB, SMB, LB, AC. *VD, *AC, *LD

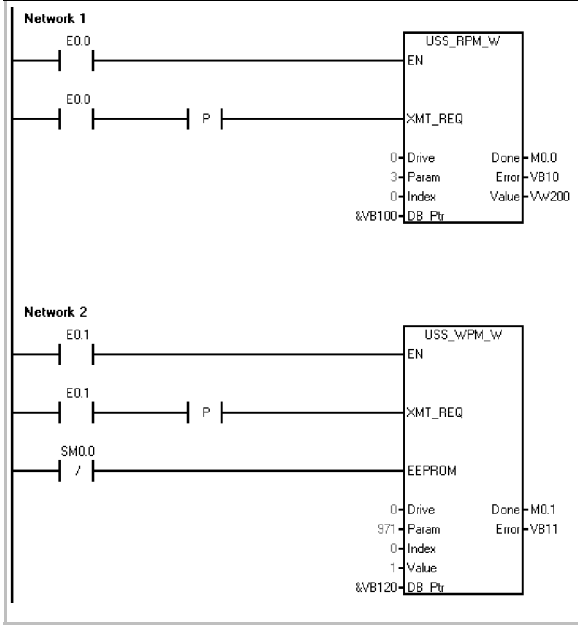
Vorsicht

Wenn Sie mit einer Operation USS_WPM_x den Parametersatz im EEPROM des Antriebs aktualisieren, müssen Sie darauf achten, dass die maximal zulässige Anzahl Schreibzyklen (ca. 50.000) für den EEPROM nicht überschritten wird.

Wenn Sie die maximal zulässige Anzahl Schreibzyklen überschreiten, führt dies zu korrupten Daten und somit zu Datenverlust. Die Anzahl der Lesezyklen ist nicht begrenzt.

Wenn häufig in die Antriebsparameter geschrieben werden muss, müssen Sie zunächst den Parameter für die EEPROM-Speichersteuerung des Antriebs auf Null setzen (bei Antrieben MicroMaster 3) und bei Antrieben MicroMaster 4 den Eingang EEPROM ausschalten.

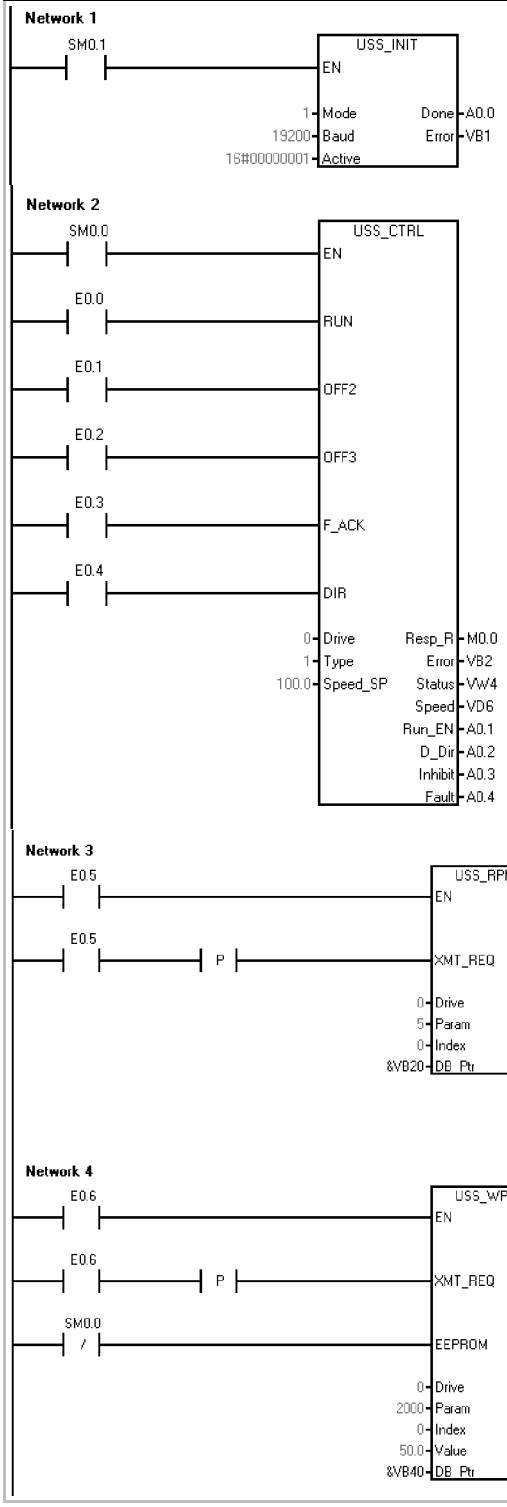
Beispiel: USS_RPM_x und USS_WPM_x




```
Netzwerk 1      //Die beiden Kontakte müssen die gleiche Adresse
                //haben.
LD      E0.0
=       L60.0
LD      E0.0
EU
=       L63.7
LD      L60.0
CALL    USS_RPM_W, L63.7, 0, 3, 0, &VB100, M0.0, VB10,
VW200

Netzwerk 2      //Die beiden Kontakte müssen die gleiche Adresse
haben.
LD      E0.1
=       L60.0
LD      E0.1
EU
=       L63.7
LDN     SM0.0
=       L63.6
LD      L60.0
CALL    USS_WPM_W, L63.7, L63.6, 0, 971, 0, 1, &VB120, M0.1,
VB11
```

Beispielprogramme für das USS-Protokoll

Beispiel: USS-Operationen Beispielprogramm für die Anzeige in AWL




```
Netzwerk 1      //USS-Protokoll initialisieren
                //Im ersten Zyklus USS-Protokoll für die
                //Schnittstelle 0 und 19200 aktivieren,
                //wobei Antriebsadresse "0" aktiv ist.
LD      SM0.1
CALL    USS_INIT, 1, 19200, 16#00000001, A0.0, VB1

Netzwerk 2      //Steuerparameter für Antrieb 0
LD      SM0.0
CALL    USS_CTRL, E0.0, E0.1, E0.2, E0.3, E0.4, 0,
1, 100.0, M0.0, VB2, VW4, VD6, A0.1, A0.2, A0.3, A0.4

Netzwerk 3      //Wortparameter aus Antrieb 0 lesen.
                //Parameter 5 Index 0 lesen.
                //1. Zustand von E0.5 im temporären
                //   Speicher ablegen, so dass dieses
                //   Netzwerk in KOP angezeigt wird.
                //2. Steigende Flanke an E0.5
                //   im temporären Lokaldatenspeicher
                //   ablegen, so dass der Impuls an das
                //   Unterprogramm übergeben werden
                //   kann.
LD      E0.5
=       L60.0
LD      E0.5
EU
=       L63.7
LD      L60.0
CALL    USS_RPM_W, L63.7, 0, 5, 0, &VB20, M0.1,
VB10, VW12

Netzwerk 4      //Wortparameter in Antrieb 0 schreiben.
                //Parameter 2000 Index 0 schreiben.
LD      E0.6
=       L60.0
LD      E0.6
EU
=       L63.7
LDN     SM0.0
=       L63.6
LD      L60.0
CALL    USS_WPM_R, L63.7, L63.6, 0, 2000, 0, 50.0,
&VB40, M0.2, VB14

Hinweis:  Dieser AWL-Code kann nicht in KOP oder
FUP übersetzt werden.
```

Fehlercodes für die Ausführung der USS-Operationen

Tabelle 11-6 Fehlercodes für die Ausführung der USS-Operationen

Fehlermeldungen	Beschreibung
0	Kein Fehler aufgetreten
1	Antrieb reagiert nicht
2	In der Antwort des Antriebs trat ein Prüfsummenfehler auf
3	In der Antwort des Antriebs trat ein Paritätsfehler auf
4	Fehler durch Störung vom Anwenderprogramm
5	Ungültiger Befehl
6	Ungültige Antriebsadresse
7	Die Kommunikationsschnittstelle war nicht für das USS-Protokoll eingerichtet
8	Die Kommunikationsschnittstelle bearbeitet eine andere Operation
9	Die angegebene Antriebsdrehzahl liegt außerhalb des Bereichs
10	Die Länge der Antwort des Antriebs ist falsch
11	Das erste Zeichen in der Antwort des Antriebs ist falsch
12	Das Längenzeichen in der Antwort des Antriebs wird von USS-Operationen nicht unterstützt
13	Der falsche Antrieb hat reagiert
14	Die Adresse DB_Ptr ist falsch
15	Die Parameterzahl ist falsch
16	Es wurde ein ungültiges Protokoll ausgewählt
17	USS ist aktiv; Änderung nicht zulässig
18	Es wurde eine unzulässige Baudrate angegeben
19	Keine Kommunikation: der Antrieb ist nicht aktiv
20	Der Parameter oder der Wert in der Antwort des Antriebs ist falsch oder enthält einen Fehlercode
21	Statt des angeforderten Wortwerts wurde ein Doppelwortwert ausgegeben
22	Statt des angeforderten Doppelwortwerts wurde ein Wortwert ausgegeben

Anschließen und Einrichten des MicroMaster-Antriebs der Serie 3

Anschließen des MicroMaster-Antriebs 3

Sie können das herkömmliche PROFIBUS-Kabel und die üblichen Steckverbinder nutzen, um die S7-200 an den MicroMaster-Antrieb der Serie 3 (MM3) anzuschließen. In Bild 11-5 sehen Sie das Verbindungskabel mit Abschlusswiderstand.

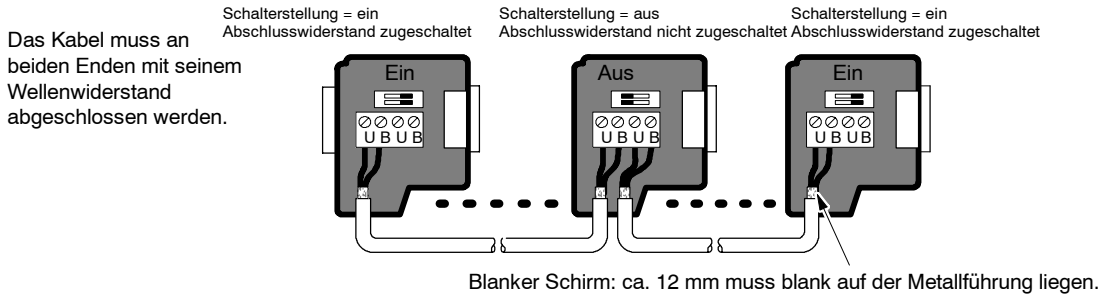
Vorsicht

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.

Die Schirmung muss mit Masse oder Pin 1 des 9-poligen Steckverbinders verbunden werden. Die Verdrahtungsklemme 2-0V des MicroMaster-Antriebs sollte mit Masse verbunden werden.



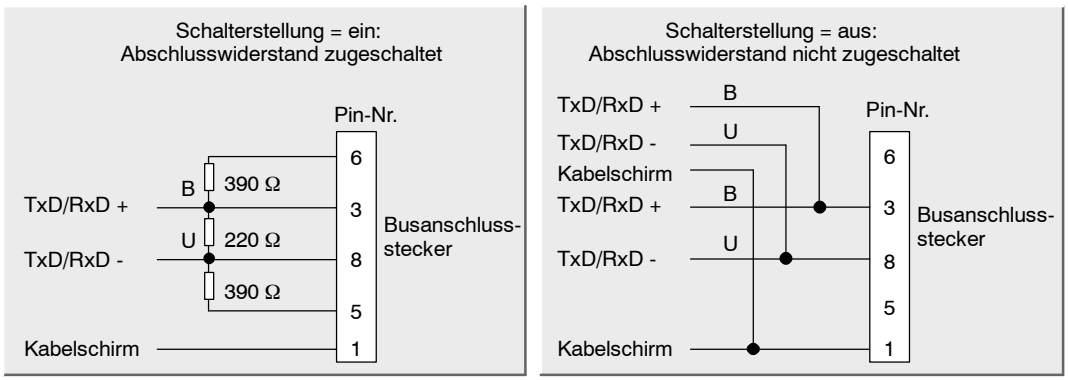


Bild 11-5 Abschließen des Netzwerkkabels

Einrichten des MicroMaster-Antriebs 3

Bevor Sie einen Antrieb an die S7-200 anschließen, müssen Sie sicherstellen, dass der Antrieb über folgende Systemparameter verfügt. Sie stellen die Parameter mit der Tastatur des Antriebs ein:

1. Setzen Sie den Antrieb auf die Werkseinstellungen zurück (optional). Drücken Sie die Taste P: P000 wird angezeigt. Drücken Sie eine der Pfeiltasten nach oben/nach unten, bis das Anzeigefeld P944 anzeigt. Drücken Sie die Taste P, um den Parameter einzugeben.

   P944 = 1

2. Aktivieren Sie den Lese-/Schreibzugriff für alle Parameter. Drücken Sie die Taste P. Drücken Sie eine der Pfeiltasten nach oben/nach unten, bis das Anzeigefeld P009 anzeigt. Drücken Sie die Taste P, um den Parameter einzugeben.

   P009 = 3

3. Prüfen Sie die Motoreinstellungen Ihres Antriebs. Die Einstellungen richten sich nach dem jeweiligen Motor. Drücken Sie die Taste P. Drücken Sie eine der Pfeiltasten nach oben/nach unten, bis das Anzeigefeld die Motoreinstellungen für Ihren Antrieb anzeigt. Drücken Sie die Taste P, um den Parameter einzugeben.

   P081 = Nennfrequenz des Motors (Hz)
   P082 = Nenndrehzahl des Motors (U/min)
   P083 = Nennstrom des Motors (A)
   P084 = Nennspannung des Motors (V)
   P085 = Nennleistung des Motors (kW/PS)

4. 4. Stellen Sie den lokalen/entfernten Modus ein. Drücken Sie die Taste P. Drücken Sie eine der Pfeiltasten nach oben/nach unten, bis das Anzeigefeld P910 anzeigt. Drücken Sie die Taste P, um den Parameter einzugeben.

   P910 = 1 Modus für entfernte Steuerung

5. Stellen Sie die Baudrate der seriellen Schnittstelle RS-485 ein. Drücken Sie die Taste P. Drücken Sie eine der Pfeiltasten nach oben/nach unten, bis das Anzeigefeld P902 anzeigt. Drücken Sie die Taste P, um den Parameter einzugeben. Drücken Sie eine der Pfeiltasten nach oben/nach unten, bis das Anzeigefeld die Nummer anzeigt, die der Baudrate Ihrer seriellen Schnittstelle RS-485 entspricht. Drücken Sie die Taste P, um den Parameter einzugeben.

P092	3	(1200 Baud)
	4	(2400 Baud)
	5	(4800 Baud)
	6	(9600 Baud - Voreinstellung)
	7	(19200 Baud)

6. Geben Sie die Slave-Adresse ein. Jeder Antrieb (maximal 31) kann über den Bus betrieben werden. Drücken Sie die Taste P. Drücken Sie eine der Pfeiltasten nach oben/nach unten, bis das Anzeigefeld P091 anzeigt. Drücken Sie die Taste P, um den Parameter einzugeben. Drücken Sie eine der Pfeiltasten nach oben/nach unten, bis das Anzeigefeld die gewünschte Slave-Adresse anzeigt. Drücken Sie die Taste P, um den Parameter einzugeben.

   P091 = 0 bis 31.

7. Hochfahrzeit (optional). Zeit in Sekunden, die der Motor braucht, um auf maximale Frequenz zu beschleunigen. Drücken Sie die Taste P. Drücken Sie eine der Pfeiltasten nach oben/nach unten, bis das Anzeigefeld P002 anzeigt. Drücken Sie die Taste P, um den Parameter einzugeben. Drücken Sie eine der Pfeiltasten nach oben/nach unten, bis das Anzeigefeld die gewünschte Zeit zum Hochfahren anzeigt. Drücken Sie die Taste P, um den Parameter einzugeben.

   P002 = 0 - 650.00

8. Herunterfahrzeit (optional) Zeit in Sekunden, die der Motor braucht, um bis zum Stillstand abzubremsen. Drücken Sie die Taste P. Drücken Sie eine der Pfeiltasten nach oben/nach unten, bis das Anzeigefeld P003 anzeigt. Drücken Sie die Taste P, um den Parameter einzugeben. Drücken Sie eine der Pfeiltasten nach oben/nach unten, bis das Anzeigefeld die gewünschte Zeit zum Herunterfahren anzeigt. Drücken Sie die Taste P, um den Parameter einzugeben.

   P003 = 0 - 650.00

9. Timeout der seriellen Verbindung. Dies ist der maximal zulässige Zeitraum zwischen zwei eingehenden Datentelegrammen. Diese Funktion schaltet den Inverter bei Kommunikationsausfall aus.

   Die Zeit wird gemessen, nachdem ein gültiges Telegramm empfangen wurde. Wird kein weiteres Datentelegramm innerhalb der angegebenen Zeit empfangen, schaltet der Inverter ab und zeigt Fehlercode F0008 an. Wenn Sie den Wert auf Null setzen, wird die Steuerung ausgeschaltet. Die Zeit zwischen den Statusabfragen am Antrieb kann mit Hilfe der Tabelle 11-1 berechnet werden.

   Drücken Sie die Taste P. Drücken Sie eine der Pfeiltasten nach oben/nach unten, bis das Anzeigefeld P093 anzeigt. Drücken Sie die Taste P, um den Parameter einzugeben. Drücken Sie eine der Pfeiltasten nach oben/nach unten, bis das Anzeigefeld den gewünschten Timeout-Wert für die serielle Verbindung anzeigt. Drücken Sie die Taste P, um den Parameter einzugeben.

   P093 = 0 - 240 (0 ist die Voreinstellung, die Zeit wird in Sekunden angegeben)

10. Sollwert serielle Verbindung Nennsystem. Dieser Wert kann variieren, doch entspricht er typischerweise 50 Hz oder 60 Hz. Dies definiert den entsprechenden 100%-Wert für voreingestellte Werte (PV) oder Sollwerte (SP). Drücken Sie die Taste P. Drücken Sie eine der Pfeiltasten nach oben/nach unten, bis das Anzeigefeld P094 anzeigt. Drücken Sie die Taste P, um den Parameter einzugeben. Drücken Sie eine der Pfeiltasten nach oben/nach unten, bis das Anzeigefeld den gewünschten Sollwert serielle Verbindung Nennsystem anzeigt. Drücken Sie die Taste P, um den Parameter einzugeben.

    P094 = 0 - 400.00

11. USS-Kompatibilität (optional). Drücken Sie die Taste P. Drücken Sie eine der Pfeiltasten nach oben/nach unten, bis das Anzeigefeld P095 anzeigt. Drücken Sie die Taste P, um den Parameter einzugeben. Drücken Sie eine der Pfeiltasten nach oben/nach unten, bis das Anzeigefeld die Nummer der gewünschten USS-Kompatibilität anzeigt. Drücken Sie die Taste P, um den Parameter einzugeben.

    P095 = 0 0,1 Hz Auflösung (Voreinstellung)
    1 0,01 Hz Auflösung

12. EEPROM-Speichersteuerung (optional). Drücken Sie die Taste P. Drücken Sie eine der Pfeiltasten nach oben/nach unten, bis das Anzeigefeld P971 anzeigt. Drücken Sie die Taste P, um den Parameter einzugeben. Drücken Sie eine der Pfeiltasten nach oben/nach unten, bis das Anzeigefeld die gewünschte EEPROM-Speichersteuerung anzeigt. Drücken Sie die Taste P, um den Parameter einzugeben.

    P971 = 0 Änderungen an Parametereinstellungen (einschließlich P971) gehen bei Spannungsausfall verloren.
    1 (Voreinstellung) Änderungen an Parametereinstellungen bleiben in Zeiträumen mit Spannungsausfall erhalten.

13. Betriebsanzeige. Drücken Sie die Taste P, um den Parametermodus zu verlassen.

Anschließen und Einrichten des MicroMaster-Antriebs der Serie 4

Anschließen des MicroMaster-Antriebs 4

Wenn Sie den MicroMaster-Antrieb der Serie 4 (MM4) anschließen möchten, stecken Sie die Enden des RS-485-Kabels in die beiden schraubenlosen Druckklemmen für den USS-Betrieb. Die S7-200 kann mit dem herkömmlichen PROFIBUS-Kabel und den Steckverbindern an den MicroMaster-Antrieb angeschlossen werden.

Vorsicht

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

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

Stellen Sie sicher, dass alle Geräte, die Sie über ein Kommunikationskabel miteinander verbinden, entweder den gleichen Bezugsleiter im Stromkreis haben oder galvanisch getrennt sind, damit keine unerwünschten Ströme auftreten.

Die Schirmung muss mit Masse oder Pin 1 des 9-poligen Steckverbinders verbunden werden. Die Verdrahtungsklemme 2-0V des MicroMaster-Antriebs sollte mit Masse verbunden werden.

Wie Sie in Bild 11-6 sehen, müssen Sie die beiden Drähte am gegenüberliegenden Ende des RS-485-Kabels in den Klemmenblock des MM4-Antriebs stecken. Zum Herstellen der Kabelverbindung am MM4-Antrieb entfernen Sie die Abdeckung(en) des Antriebs, damit Sie Zugriff auf die Klemmenblöcke haben. Im Benutzerhandbuch des MM4-Antriebs finden Sie eine ausführliche Beschreibung, wie Sie die Abdeckung(en) an Ihrem spezifischen Antrieb entfernen.

Die Anschlüsse am Klemmenblock sind durchnummeriert. Verwenden Sie auf der Seite der S7-200 einen PROFIBUS-Anschlussstecker und schließen Sie die Klemme A des Kabels an Klemme 15 des Antriebs (bei einem MM420) oder an Klemme 30 des Antriebs (bei einem MM440) an. Schließen Sie die Klemme B des Kabelverbinders an Klemme 14 (bei einem MM420) oder an Klemme 29 (bei einem MM440) an.

Handelt es sich bei der S7-200 um einen abschließenden Teilnehmer im Netz, oder handelt es sich um eine Punkt-zu-Punkt-Verbindung, müssen Sie die Klemmen A1 und B1 (nicht A2 und B2) des Steckverbinders verwenden, weil diese Klemmen die Abschlusseinstellungen ermöglichen (z.B. beim DP-Steckverbinder 6ES7 972-0BA40-0X40).

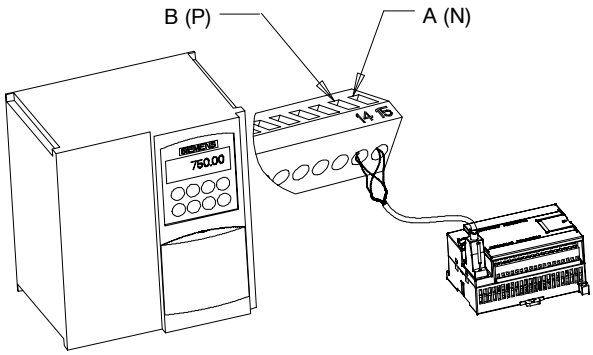


Bild 11-6 Anschließen des MM420-Klemmenblocks

Vorsicht

Achten Sie darauf, dass Sie die Abdeckungen des Antriebs sorgfältig wieder einbauen, bevor Sie das Gerät mit Spannung versorgen.

Ist der Antrieb als abschließender Teilnehmer im Netz konfiguriert, dann müssen außerdem Abschlusswiderstände mit den entsprechenden Klemmen verdrahtet werden. Bild 11-7 zeigt ein Beispiel für die erforderlichen Abschlussschaltungen an einem Antrieb MM4.

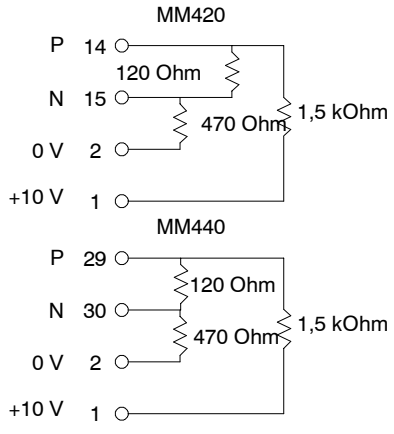


Bild 11-7 Beispiel für den Abschluss

Einrichten des MM4-Antriebs

Bevor Sie einen Antrieb an die S7-200 anschließen, müssen Sie sicherstellen, dass der Antrieb über folgende Systemparameter verfügt. Sie stellen die Parameter mit der Tastatur des Antriebs ein:

1. Setzen Sie den Antrieb auf die Werkseinstellungen zurück (optional): P0010 = 30
   P0970 = 1

   Wenn Sie diesen Schritt überspringen, müssen Sie darauf achten, dass die folgenden Parameter auf diese Werte gesetzt sind:
   USS PZD-Länge: P2012 Index 0 = 2
   USS PKW-Länge: P2013 Index 0 = 127

2. Aktivieren Sie den Lese-/Schreibzugriff für alle Parameter (Expertenmodus): P0003 = 3

3. Prüfen Sie die Motoreinstellungen Ihres Antriebs: P0304 = Motornennspannung (V)
   P0305 = Motornennstrom (A)
   P0307 = Motornennleistung (W)
   P0310 = Motornennfrequenz (Hz)
   P0311 = Motornenndrehzahl (U/MIN)

   Die Einstellungen richten sich nach dem jeweiligen Motor.

   Damit Sie die Parameter P304, P305, P307, P310 und P311 einstellen können, müssen Sie zunächst den Parameter P010 auf 1 setzen (Modus für Schnellinbetriebsetzung). Nachdem Sie die Parameter eingestellt haben, setzen Sie den Parameter P010 auf 0. Die Parameter P304, P305, P307, P310 und P311 können nur im Modus für Schnellinbetriebsetzung geändert werden.

4. Stellen Sie den lokalen/entfernten Modus ein: P0700 Index 0 = 5

5. Stellen Sie den Frequenzsollwert der COM-Verbindung auf USS ein: P1000 Index 0 = 5

6. Hochfahrzeit (optional): P1120 = 0 bis 650.00

   Zeit in Sekunden, die der Motor braucht, um auf maximale Frequenz zu beschleunigen.

7. Herunterfahrzeit (optional): P1121 = 0 bis 650.00

   Zeit in Sekunden, die der Motor braucht, um bis zum Stillstand abzubremsen.

8. Stellen Sie die Referenzfrequenz der seriellen Verbindung ein: P2000 = 1 bis 650 Hz

9. Stellen Sie die USS-Normalisierung ein: P2009 Index 0 = 0

10. Stellen Sie die Baudrate der seriellen Schnittstelle RS-485 ein: P2010 Index 0 =
    4 (2400 Baud)
    5 (4800 Baud)
    6 (9600 Baud)
    7 (19200 Baud
    8 (38400 Baud)
    9 (57600 Baud)
    12 (115200 Baud)

11. Geben Sie die Slave-Adresse ein: P2011 Index 0 = 0 bis 31

    Jeder Antrieb (maximal 31) kann über den Bus betrieben werden.

12. Stellen Sie das Timeout für die serielle Verbindung ein:
    P2014 Index 0 = 0 bis 65.535 ms
    (0 = Timeout deaktiviert)

    Dies ist der maximal zulässige Zeitraum zwischen zwei eingehenden Datentelegrammen. Diese Funktion schaltet den Inverter bei Kommunikationsausfall aus. Die Zeit wird gemessen, nachdem ein gültiges Telegramm empfangen wurde. Wird kein weiteres Datentelegramm innerhalb der angegebenen Zeit empfangen, schaltet der Inverter ab und zeigt Fehlercode F0070 an. Wenn Sie den Wert auf Null setzen, wird die Steuerung ausgeschaltet. Die Zeit zwischen den Statusabfragen am Antrieb kann mit Hilfe der Tabelle 11-1 berechnet werden.

13. Übertragen Sie die Daten vom RAM zum EEPROM:

    P0971 = 1 (Übertragung starten). Speichern Sie die Änderungen an den Parametereinstellungen im EEPROM.

12 Bibliothek für das Modbus-Protokoll

Durch die Operationsbibliotheken von STEP 7 Micro/WIN wird die Kommunikation mit Modbus-Master-Geräten einfacher. STEP 7-Micro/WIN bietet Ihnen vorkonfigurierte Unterprogramme und Interruptprogramme, die speziell für die Modbus-Kommunikation ausgelegt sind. Mit den Operationen des Modbus-Slave-Protokolls können Sie die S7-200 als Modbus-RTU-Slave-Gerät konfigurieren, das mit Modbus-Master-Geräten kommunizieren kann.

Diese Operationen finden Sie im Ordner Bibliotheken im Operationsbaum von STEP 7-Micro/WIN. Mit diesen neuen Operationen können Sie die S7-200 als Modbus-Slave einrichten. Wenn Sie eine Modbus-Slave-Operation auswählen, wird automatisch ein zugehöriges Unterprogramm (oder mehrere) in Ihr Projekt aufgenommen.

In diesem Kapitel

Anforderungen für den Einsatz des Modbus-Protokolls 340
Initialisierung und Ausführungszeit des Modbus-Protokolls 340
Modbus-Adressierung 341
Verwenden von Operationen für das Modbus-Slave-Protokoll 342
Operationen für das Modbus-Slave-Protokoll 343

Anforderungen für den Einsatz des Modbus-Protokolls

Die Operationen für das Modbus-Slave-Protokoll nutzen die folgenden Ressourcen der S7-200:

- Durch die Initialisierung des Modbus-Slave-Protokolls wird die Schnittstelle 0 für die Kommunikation mittels Modbus-Slave-Protokoll bereitgestellt.

  Wenn die Schnittstelle 0 für die Kommunikation mittels Modbus-Slave-Protokoll genutzt wird, kann sie nicht für andere Zwecke eingesetzt werden, auch nicht für die Kommunikation mit STEP 7-Micro/WIN. Die Operation MBUS_INIT regelt die Zuordnung von Schnittstelle 0 zum Modbus-Slave-Protokoll oder PPI.

- Die Operationen für das Modbus-Slave-Protokoll wirken sich auf alle Sondermerker aus, die der frei programmierbaren Kommunikation an Schnittstelle 0 zugeordnet sind.
- Die Operationen für das Modbus-Slave-Protokoll nutzen 3 Unterprogramme und 2 Interruptprogramme.
- Die Operationen für das Modbus-Slave-Protokoll benötigen 1857 Bytes Programmspeicher für zwei Modbus-Slave-Operationen und die unterstützenden Programme.
- Die Variablen für die Operationen für das Modbus-Slave-Protokoll benötigen einen 779 Byte großen Baustein im Variablenspeicher. Die Anfangsadresse dieses Bausteins wird vom Anwender zugewiesen und ist für Modbus-Variablen reserviert.

Tipp

Wenn Sie für den Betrieb von Schnittstelle 0 wieder PPI einstellen möchten, so dass Sie mit STEP 7-Micro/WIN kommunizieren können, müssen Sie die Schnittstelle 0 mit einer anderen Operation MBUS_INIT erneut zuweisen.

Sie können den Betriebsartenschalter auf der S7-200 auch in den Betriebszustand STOP versetzen. Dadurch werden die Parameter für die Schnittstelle 0 zurückgesetzt.

Initialisierung und Ausführungszeit des Modbus-Protokolls

Die Modbus-Kommunikation umfasst eine zyklische Redundanzprüfung (CRC = cyclic redundancy check), um die Integrität der Kommunikationsmeldungen sicherzustellen. Das Modbus-Slave-Protokoll nutzt eine Tabelle mit zuvor berechneten Werten, um die für die Bearbeitung einer Meldung erforderliche Zeit zu verringern. Die Initialisierung dieser CRC-Tabelle benötigt ca. 425 Millisekunden. Diese Initialisierung wird im Unterprogramm MBUS_INIT vorgenommen und läuft normalerweise im ersten Zyklus des Anwenderprogramms ab, nachdem der Betriebszustand RUN eingenommen wurde. Sie sind dafür verantwortlich, die Zeitüberwachung zurückzusetzen und die Ausgänge aktiviert zu halten (sofern dies für Erweiterungsmodule erforderlich ist), wenn die Zeit für das Unterprogramm MBUS_INIT und für andere Anwenderinitialisierungen die 500 Millisekunden der Zeitüberwachung überschreitet. Die Zeitüberwachung des Ausgabemoduls wird durch Schreiben in die Ausgänge des Moduls zurückgesetzt. Lesen Sie hierzu die Beschreibung der Operation Überwachungszeit rücksetzen in Kapitel 6.

Die Zykluszeit wird verlängert, wenn das Unterprogramm MBUS_SLAVE eine Anforderung bearbeitet. Weil die meiste Zeit dafür benötigt wird, den Modbus-CRC zu berechnen, wird die Zykluszeit für jedes Byte in der Anforderung und in der Antwort um ca. 650 Mikrosekunden verlängert. Maximal wird die Zykluszeit pro Anforderung/Antwort (Lesen oder Schreiben von 120 Wörtern) um ca. 165 Millisekunden verlängert.

Modbus-Adressierung

Modbus-Adressen werden üblicherweise als Werte aus 5 oder 6 Zeichen geschrieben, die den Datentyp und den Versatz angeben. Das erste Zeichen oder die ersten beiden Zeichen geben den Datentyp an, die letzten vier Zeichen geben den richtigen Wert innerhalb des Datentyps an. Das Modbus-Master-Gerät bildet dann die Adressen auf die entsprechenden Funktionen ab. Die folgenden Adressen werden von den Modbus-Slave-Operationen unterstützt:

- ❏ 000001 bis 000128 sind digitale Ausgänge, die auf A0.0 bis A15.7 abgebildet werden.
- ❏ 010001 bis 010128 sind digitale Eingänge, die auf E0.0 bis E15.7 abgebildet werden.
- ❏ 030001 bis 030032 sind analoge Prozessabbilder der Eingänge, die auf AEW0 bis AEW62 abgebildet werden.
- ❏ 040001 bis 04xxxx sind Halteregister, die auf den Variablenspeicher abgebildet werden.

Alle Modbus-Adressen haben eine Basis von 1. Tabelle 12-1 zeigt die Abbildung der Modbus-Adressen auf die S7-200 Adressen.

Beim Modbus-Slave-Protokoll können Sie die Anzahl der Eingänge, Ausgänge, Analogeingänge und Halteregister (Variablenspeicher), auf die ein Modbus-Master zugreifen kann, einschränken.

Der Parameter MaxIQ der Operation MBUS_INIT gibt die maximale Anzahl der digitalen Eingänge oder Ausgänge (E oder A) an, auf die der Modbus-Master zugreifen darf.

Der Parameter MaxAI der Operation MBUS_INIT gibt die maximale Anzahl der Prozessabbilder der Eingänge (AEW) an, auf die der Modbus-Master zugreifen darf.

Der Parameter MaxHold der Operation MBUS_INIT gibt die maximale Anzahl der Halteregister (Worte im Variablenspeicher) an, auf die der Modbus-Master zugreifen darf.

Weitere Informationen zum Einrichten der Speichereinschränkungen für den Modbus-Slave finden Sie in der Beschreibung der Operation MBUS_INIT.

Tabelle 12-1 Abbilden von Modbus-Adresse auf die S7-200

Modbus-Adresse	S7-200 Adresse
000001	A0.0
000002	A0.1
000003	A0.2
...	...
000127	A15.6
000128	A15.7
010001	E0.0
010002	E0.1
010003	E0.2
...	...
010127	E15.6
010128	E15.7
030001	AEW0
030002	AEW2
030003	AEW4
...	...
030032	AEW62
040001	HoldStart
040002	HoldStart+2
040003	HoldStart+4
...	...
04xxxx	HoldStart+2 x (xxxx-1)

Konfigurieren der Symboltabelle

Nachdem Sie die Adresse für das erste Symbol eingetragen haben, berechnet die Tabelle automatisch die übrigen Symbole und ordnet sie in der Tabelle zu.

Sie ordnen der Tabelle, die 779 Bytes belegt, eine Anfangsadresse im Variablenspeicher zu. Achten Sie darauf, dass sich die Zuweisung der Modbus-Slave-Symbole nicht mit dem Variablenspeicher überschneidet, der über die Parameter HoldStart und MaxHold der Operation MBUS_INIT den Modbus-Halteregistern zugeordnet ist. Überschneiden sich die Speicherbereiche, gibt die Operation MBUS_INIT einen Fehler aus.

Verwenden von Operationen für das Modbus-Slave-Protokoll

Wenn Sie die Operationen für das Modbus-Slave-Protokoll in Ihrem S7-200 Programm verwenden möchten, gehen Sie folgendermaßen vor:

1. Fügen Sie die Operation MBUS_INIT in Ihr Programm ein. Führen Sie die Operation MBUS_INIT nur in einem Zyklus aus. Mit der Operation MBUS_INIT können Sie die Modbus-Kommunikationsparameter initialisieren oder ändern.

   Wenn Sie die Operation MBUS_INIT einfügen, werden verschiedene versteckte Unterprogramme und Interruptprogramme automatisch in Ihr Programm aufgenommen.

2. Weisen Sie den 779 aufeinander folgenden Bytes im Variablenspeicher für die Operationen des Modbus-Slave-Protokolls eine Anfangsadresse zu.

3. Ordnen Sie nur eine Operation MBUS_SLAVE in Ihrem Programm an. Diese Operation wird in jedem Zyklus aufgerufen, um empfangene Anforderungen zu bearbeiten.

4. Schließen Sie das Kommunikationskabel an die Schnittstelle 0 der S7-200 und an die Modbus-Master-Geräte an.

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.

Die Akkumulatoren (AC0, AC1, AC2, AC3) werden von den Modbus-Slave-Operationen verwendet und erscheinen in der Auflistung der Querverweise. Vor der Ausführung einer Modbus-Slave-Operation werden die Werte in den Akkumulatoren gespeichert und vor Beendigung der Operation werden diese Werte in den Akkumulatoren wiederhergestellt. Dadurch wird sichergestellt, dass alle Anwenderdaten in den Akkumulatoren erhalten bleiben, während eine Modbus-Slave-Operation ausgeführt wird.

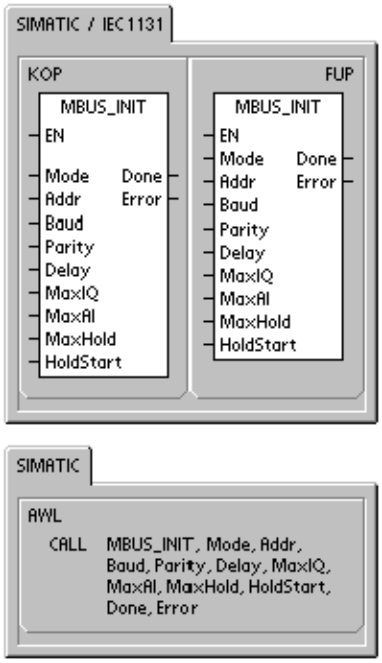
Die Operationen für das Modbus-Slave-Protokoll unterstützen das Modbus-RTU-Protokoll. Diese Operationen nutzen die frei programmierbare Kommunikation der S7-200, um die häufigsten Modbus-Funktionen zu unterstützen. Folgende Modbus-Funktionen werden unterstützt:

Tabelle 12-2 Unterstützte Funktionen des Modbus-Slave-Protokolls

Funktion	Beschreibung
1	Zustand von einem/mehreren digitalen Ausgängen lesen. Funktion 1 gibt den Zustand (ein/aus) von einer beliebigen Anzahl an Ausgängen (A) aus.
2	Zustand von einem/mehreren digitalen Eingängen lesen. Funktion 2 gibt den Zustand (ein/aus) von einer beliebigen Anzahl an Eingängen (E) aus.
3	Ein/mehrere Halteregister lesen. Funktion 3 gibt den Inhalt des Variablenspeichers aus. Halteregister sind Wortwerte unter Modbus und dienen dazu, bis zu 120 Wörter in einer Anforderung zu lesen.
4	Ein/mehrere Prozessabbilder der Eingänge lesen. Funktion 4 gibt analoge Eingangswerte aus.
5	Einen digitalen Ausgang schreiben. Funktion 5 setzt einen digitalen Ausgang auf den angegebenen Wert. Der Ausgang wird nicht geforct und das Programm kann den von der Modbus-Anforderung geschriebenen Wert überschreiben.
6	Ein Halteregister schreiben. Funktion 6 schreibt einen einzigen Halteregisterwert in den Variablenspeicher der S7-200.
15	Mehrere digitale Ausgänge schreiben. Funktion 15 schreibt mehrere digitale Ausgangswerte in das Prozessabbild der Ausgänge der S7-200. Der erste Ausgang muss auf einer Bytegrenze beginnen (z.B. A0.0 oder A2.0) und die Anzahl der zu schreibenden Ausgänge muss ein Vielfaches von 8 sein. Dies ist eine Einschränkung der Operationen für das Modbus-Slave-Protokoll. Die Ausgänge werden nicht geforct und das Programm kann die von der Modbus-Anforderung geschriebenen Werte überschreiben.
16	Mehrere Halteregister schreiben. Funktion 16 schreibt mehrere Halteregister in den Variablenspeicher der S7-200. In einer Anforderung können bis zu 120 Wörter geschrieben werden.

Operationen für das Modbus-Slave-Protokoll

Operation MBUS_INIT



Mit der Operation MBUS_INIT wird die Modbus-Kommunikation aktiviert und initialisiert oder deaktiviert. Bevor die Operation MBUS_SLAVE verwendet werden kann, muss die Operation MBUS_INIT fehlerfrei ausgeführt werden. Die Operation wird beendet und das Bit Done wird sofort gesetzt, bevor die nächste Operation ausgeführt wird.

Die Operation wird in jedem Zyklus ausgeführt, wenn der Eingang EN eingeschaltet ist.

Die Operation MBUS_INIT muss bei jeder Änderung des Kommunikationszustands genau einmal ausgeführt werden. Deshalb muss der Eingang EN durch die Erkennung einer steigenden Flanke eingeschaltet oder nur im ersten Zyklus ausgeführt werden.

Der Wert des Eingangs Mode stellt das Kommunikationsprotokoll ein: der Eingangswert 1 weist der Schnittstelle 0 das Modbus-Protokoll zu und aktiviert das Protokoll. Der Eingangswert 0 weist der Schnittstelle 0 das PPI-Protokoll zu und deaktiviert das Modbus-Protokoll.

Der Parameter Baud stellt eine Baudrate von 1200, 2400, 4800, 9600, 19200, 38400, 57600 oder 115200 ein.

Der Parameter Addr stellt für die Adresse einen Wert zwischen 1 und 247 (jeweils inklusive) ein.

Tabelle 12-3 Parameter der Operation MBUS_INIT

Eingänge/Ausgänge	Datentyp	Operanden
Mode, Addr, Parity	BYTE	VB, EB, AB, MB, SB, SMB, LB, AC, Konstante, *VD, *AC, *LD
Baud, HoldStart	DWORD	VD, ED, AD, MD, SD, SMD, AC, LD, Konstante, *VD, *AC, *LD
Delay, MaxIQ, MaxAI, MaxHold	WORD	VW, EW, AW, MW, SW, SMW, LW, AC, Konstante, *VD, *AC, *LD
Done	BOOL	E, A, M, S, SM, T, Z, V, L
Error	BYTE	VB, EB, AB, MB, SB, SMB, LB, AC, *VD, *AC, *LD

Der Parameter Parity entspricht der Parität des Modbus-Master. Folgende Werte sind gültig:

- 0 - keine Parität
- 1 - ungerade Parität
- 2 - gerade Parität

Der Parameter Delay verlängert die übliche Modbus-Bedingung für die Zeitüberwachung einer Meldung um die angegebene Anzahl Millisekunden. Der typische Wert für diesen Parameter ist bei Betrieb im verdrahteten Netz 0. Wenn Sie ein Modem mit Fehlerkorrektur verwenden, stellen Sie für die Verzögerung einen Wert zwischen 50 und 100 Millisekunden ein. Wenn Sie Spreizbandfunkmodems verwenden, stellen Sie für die Verzögerung einen Wert zwischen 10 und 100 Millisekunden ein. Der Wert für den Parameter Delay kann zwischen 0 und 32767 Millisekunden betragen.

Der Parameter MaxIQ stellt die Anzahl der Eingänge und Ausgänge, die für die Modbus-Adressen 00xxxx und 01xxxx zur Verfügung stehen, auf Werte zwischen 0 und 128 ein. Der Wert 0 deaktiviert alle Lese- und Schreiboperationen für die Eingänge und Ausgänge. Der vorgeschlagene Wert für MaxIQ beträgt 128, wodurch der Zugriff auf alle Eingänge und Ausgänge der S7-200 möglich ist.

Der Parameter MaxAI stellt die Anzahl der Worteingänge (AE), die für die Modbus-Adresse 03xxx zur Verfügung stehen, auf Werte zwischen 0 und 32 ein. Der Wert 0 deaktiviert alle Leseoperationen der Analogeingänge. Der vorgeschlagene Wert für MaxAI, der den Zugriff auf alle Analogeingänge der S7-200 zulässt, lautet wie folgt:

- 0 bei der CPU 221
- 16 bei der CPU 222
- 32 bei den CPUs 224, 226 und 226XM

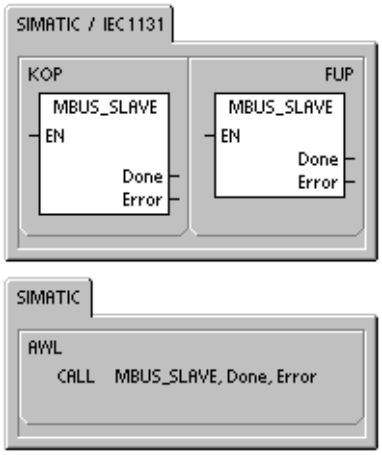
Der Parameter MaxHold stellt die Anzahl der Wort-Halteregister im Variablenspeicher ein, die für die Modbus-Adresse 04xxx zur Verfügung stehen. Damit beispielsweise ein Master auf 2000 Bytes im Variablenspeicher zugreifen kann, müssen Sie für MaxHold einen Wert von 1000 Wörtern (Halteregister) einstellen.

Der Parameter HoldStart ist die Adresse, an der die Halteregister im Variablenspeicher beginnen. Dieser Wert wird üblicherweise auf VB0 gesetzt, so dass der Parameter HoldStart auf &VB0 (Adresse von VB0) gesetzt wird. Es können andere Adressen im Variablenspeicher als Anfangsadresse für die Halteregister angegeben werden, damit VB0 an anderer Stelle im Projekt verwendet werden kann. Der Modbus-Master hat Zugriff auf die in MaxHold angegebene Anzahl Wörter im Variablenspeicher, die an der in HoldStart angegebenen Adresse beginnen.

Ist die Bearbeitung der Operation MBUS_INIT beendet, wird der Ausgang Done eingeschaltet. Das Ausgangsbyte Error enthält das Ergebnis der Ausführung der Operation. Tabelle 12-5 zeigt die Fehlerbedingungen, die während der Ausführung der Operationen auftreten können.

Operation MBUS_SLAVE

Mit der Operation MBUS_SLAVE wird eine Anforderung eines Modbus-Master bearbeitet. Die Operation muss in jedem Zyklus ausgeführt werden, damit auf Modbus-Anforderungen geprüft und geantwortet wird.



Die Operation wird in jedem Zyklus ausgeführt, wenn der Eingang EN eingeschaltet ist.

Die Operation MBUS_SLAVE hat keine Eingangsparameter.

Der Ausgang Done ist eingeschaltet, wenn die Operation MBUS_SLAVE auf eine Modbus-Anforderung reagiert. Der Ausgang Done ist ausgeschaltet, wenn keine Anforderung bearbeitet wurde.

Der Ausgang Error enthält das Ergebnis der Ausführung der Operation. Dieser Ausgang ist nur gültig, wenn Done eingeschaltet ist. Ist Done ausgeschaltet, wird der Parameter Error nicht verändert. Tabelle 12-5 zeigt die Fehlerbedingungen, die während der Ausführung der Operation auftreten können.

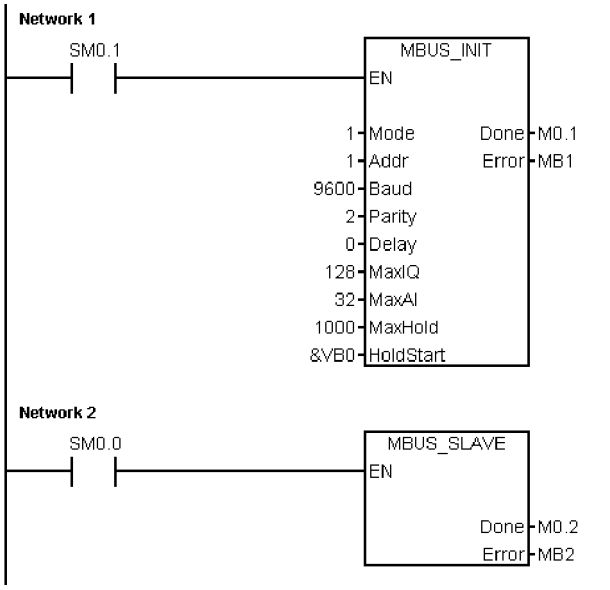
Tabelle 12-4 Parameter der Operation MBUS_SLAVE

Parameter	Datentyp	Operanden
Done	BOOL	E, A, M, S, SM, T, Z, V, L
Error	BYTE	VB, EB, AB, MB, SB, SMB, LB, AC, *VD, *AC, *LD

Tabelle 12-5 Fehlercodes für die Ausführung des Modbus-Slave-Protokolls

Fehlermeldungen	Beschreibung
0	Kein Fehler aufgetreten
1	Speicherbereichsfehler
2	Unzulässige Baudrate oder Parität
3	Unzulässige Slave-Adresse
4	Unzulässiger Wert für Modbus-Parameter
5	Halteregister überschneiden sich mit Modbus-Slave-Symbolen
6	Paritätsfehler beim Empfangen
7	CRC-Fehler beim Empfangen
8	Unzulässige Funktionsanforderung / Funktion nicht unterstützt
9	Unzulässige Speicheradresse in Anforderung
10	Slave-Funktion nicht aktiviert

Beispiel für die Programmierung des Modbus-Slave-Protokolls




```
Netzwerk 1
        //Modbus-Slave-Protokoll im ersten
        //Zyklus initialisieren. Slave-Adresse auf 1
        //setzen, für Schnittstelle 0 9600 Baud mit
        //gerader Parität einstellen, gesamter
        //Zugriff auf alle E-, A- und AE-Werte,
        //Zugriff auf 1000 Halteregister (2000 Bytes)
        //mit Beginn an VB0 zulassen.
LD      SM0.1
CALL    MBUS_INIT,1,1,9600,2,0,128,32,1000,
        &VB0,M0.1,MB1
Netzwerk 2
        //Modbus-Slave-Protokoll in
        //jedem Zyklus ausführen.
LD      SM0.0
CALL    MBUS_SLAVE,M0.2,MB2
```

Technische Daten

A

In diesem Kapitel

Allgemeine technische Daten 348
Technische Daten der CPUs 351
Technische Daten der digitalen Erweiterungsmodule 357
Technische Daten der analogen Erweiterungsmodule 363
Technische Daten der Thermoelement- und RTD-Erweiterungsmodule 373
Technische Daten des EM 277 PROFIBUS-DP-Moduls 385
Technische Daten des Modemmoduls EM 241 397
Technische Daten des Positioniermoduls EM 253 399
Technische Daten des Ethernet-Moduls (CP 243-1) 405
Technische Daten des Internet-Moduls (CP 243-1 IT) 407
Technische Daten des AS-Interface-Moduls (CP 243-2) 410
Optionale Steckmodule 412
Steckleitung für Erweiterungsmodule 412
RS-232/PPI-Multi-Master-Kabel und USB/PPI-Multi-Master-Kabel 413
Eingangssimulatoren 417

Allgemeine technische Daten

Normen

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.

- ❑ EG-Richtlinie zu Niederspannungen
  EN 61131-2: Automatisierungssysteme - Anforderungen an die Geräte
- ❑ EG-Richtlinie zur elektromagnetischen Verträglichkeit 89/336/EWG

  Elektromagnetische Aussendung
  EN 61000-6-3: Wohn-, Gewerbe- und Leichtindustrieumgebungen
  EN 61000-6-4: Industrieumgebung

  Elektromagnetische Störfestigkeit
  EN 61000-6-2: Industrieumgebung
- ❑ Underwriters Laboratories, Inc.: UL 508 Listed (Industriesteuerungsgeräte)
  Registrierungsnummer E75310
- ❑ Canadian Standards Association: CSA C22.2 Nummer 142 (Prozesssteuerungsgeräte)
- ❑ Factory Mutual Research: FM Klasse I, Division 2, Gruppen A, B, C, & D Explosionsgefährdete Betriebsstätten, T4A und Klasse I, Zone 2, IIC, T4.

Tipp

Die Produktreihe SIMATIC S7-200 entspricht der CSA-Norm.

Das cULus-Logo macht kenntlich, dass die S7-200 von Underwriters Laboratories (UL) gemäß den Normen UL 508 und CSA 22.2 Nr. 142 geprüft und zertifiziert wurde.

Zulassungen für das Seewesen

Die S7-200 Produkte werden regelmäßig für die Zulassungen hinsichtlich bestimmter Märkte und Anwendungen bei bestimmten Behörden eingereicht. Diese Tabelle gibt die Behörden und Zertifikatsnummern der S7-200 Produkte an. Nicht alle S7-200 Produkte in diesem Handbuch tragen eine besondere behördliche Zulassung. Wenden Sie sich an Ihre Siemens Vertretung, wenn Sie eine Liste mit den aktuellen Zulassungen für die einzelnen Bestellnummern benötigen.

Behörde	Zertifikatsnummer
Lloyds Register of Shipping (LRS)	99 / 20018(E1)
American Bureau of Shipping (ABS)	01-HG20020-PDA
Germanischer Lloyd (GL)	12 045 - 98 HH
Det Norske Veritas (DNV)	A-8071
Bureau Veritas (BV)	09051 / A2 BV
Nippon Kaiji Kyokai (NK)	A-534

A

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.
Ein externer Schutzkreis, der der Last angepasst ist, verlängert die Lebensdauer der Kontakte.

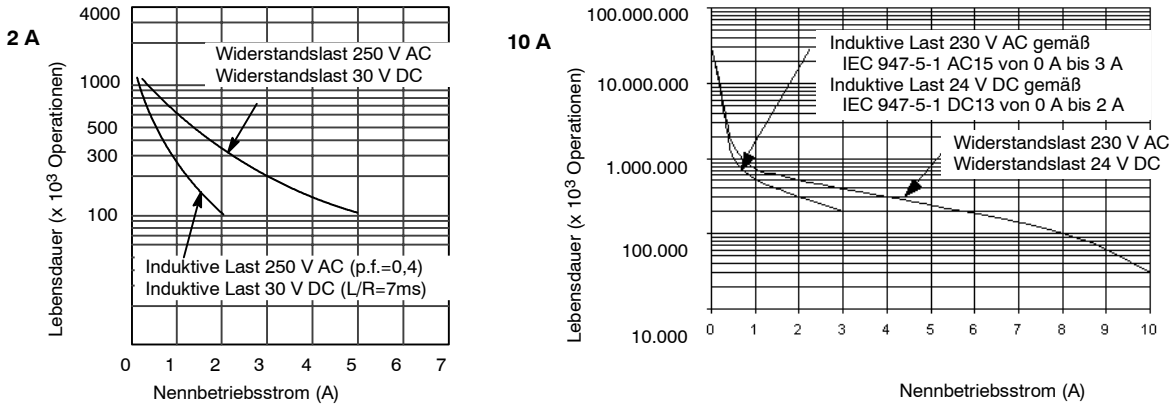


Bild A-1 Lebensdauer von Relais bei 2 A und 10 A

Technische Daten

Alle S7-200 CPUs und Erweiterungsmodule entsprechen den in Tabelle A-1 aufgeführten technischen Daten.

Hinweis

Wenn ein mechanischer Kontakt die Ausgangsspannung zur S7-200 CPU oder einem digitalen Erweiterungs-
modul einschaltet, wird ca. 50 Mikrosekunden lang das Signal 1 an die Digitalausgänge gesendet. Dies müssen Sie berücksichtigen, vor allem, wenn Sie mit Geräten arbeiten, die auf kurze Impulse reagieren.

Tabelle A-1 Technische Daten

Umgebungsbedingungen - Transport und Lagerung	
IEC 60068-2-2 Test Bb, trockene Wärme und IEC 60068-2-1 Test Ab, Kälte	-40 °C bis +70 °C
IEC 60068-2-30 Test Db, feuchte Wärme	25 °C bis 55 °C, 95% Luftfeuchtigkeit
EN 60068-2-14, Test Na, Temperaturschock	-40 °C bis +70 °C, Verweilzeit 3 Stunden, 2 Zyklen
EN 60068-2-31 Umkippen	100 mm, viermal, ohne Verpackung
EN 60068-2-32 Freier Fall	1 m, fünfmal, in Versandverpackung
Umgebungsbedingungen - Betrieb	
Umgebungstemperatur (Luftzufuhr 25 mm unterhalb des Geräts)	0 °C bis 55 °C waagerechter Einbau, 0 °C bis 45 °C senkrechter Einbau 95 % Luftfeuchtigkeit, nicht kondensierend
Atmosphärischer Druck	1080 bis 795 hPa (entspricht einer Höhe von -1000 bis 2000 m)
Konzentration von Schmutzstoffen	SO_2: < 0,5 ppm; H_2S: < 0,1 ppm; RH < 60% nicht kondensierend
EN 60068-2-14, Test Nb, Temperaturwechsel	5 °C bis 55 °C, 3 °C/Minute
EN 60068-2-27 Mechanische Stoßbeanspruchung	15 G, Impuls 11 ms, 6 Stöße auf jeder der 3 Achsen
EN 60068-2-6 Sinusschwingung	Schaltschrankmontage: 0,30 mm von 10 bis 57 Hz; 2 G von 57 bis 150 Hz Hutschienenmontage: 0,15 mm von 10 bis 57 Hz; 1 G von 57 bis 150 Hz 10 Ablenkungen je 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 < 12,5 mm.
Elektromagnetische Verträglichkeit - Störfestigkeit nach EN61000-6-2[1]	
EN 61000-4-2 Elektrostatische Entladung	8 kV Entladung durch die Luft an allen Oberflächen und Kommunikationsschnittstellen, 4 kV Entladung durch Kontakt mit freiliegenden leitenden Oberflächen
EN 61000-4-3 Abgestrahltes elektromagnetisches Feld	80 MHz bis 1 GHz 10 V/m, 80% Modulation mit 1 kHz Signal
EN 61000-4-4 Schnelle transiente Störgröße	2 kV, 5 kHz bei Kopplungsnetz zu AC- und DC-Systemspannung 2 kV, 5 kHz bei Kopplungsklemme zu Digitalein-/Digitalausgängen 1 kV, 5 kHz bei Kopplungsklemme zu Kommunikation
EN 61000-4-5 Stoßwellenfestigkeit	Spannungsversorgung: 2 kV asymmetrisch, 1 kV symmetrisch E/A 1 kV symmetrisch (24-V-DC-Kreise benötigen externen Schutz vor Stromstößen)
EN 61000-4-6 Leitungsgeführte Störungen	0,15 bis 80 MHz 10 V effektiv 80 % Amplitudenmodulation bei 1 kHz
EN 61000-4-11 Spannungseinbrüche, kurze Unterbrechungen und Spannungsänderungen	Verringerung von > 95 % bei 8,3 ms, 83 ms, 833 ms und 4167 ms
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
Elektromagnetische Verträglichkeit — geleitete und abgestrahlte Aussendung gemäß EN 61000-6-3[2] und EN 61000-6-4	
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 MHz 230 MHz bis 1 GHz	 Quasi-Spitze 40 dB (μV/m), gemessen bei 10 m Quasi-Spitze 47 dB (μV/m), gemessen bei 10 m
EN 55011, Klasse B, Gruppe 1, geleitet[2] 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. Frequenz bis 56 dB (μV), < 56 dB (μV) abnehmender Mittelwert mit log. Frequenz bis 46 dB (μV) < 56 dB (μV) Quasi-Spitze, < 46 dB (μV) Mittelwert < 60 dB (μV) Quasi-Spitze, < 50 dB (μV) Mittelwert
EN 55011, Klasse B, Gruppe 1, abgestrahlt[2] 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-Stromkreis an 115/230-V-Stromkreis 230-V-Stromkreis an 24/5-V-Stromkreis 115-V-Stromkreis an 24/5-V-Stromkreis	500 V DC (Grenzwerte Potentialtrennung) 1500 V AC 1500 V AC 1500 V AC 1500 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 Das Gerät muss in einem geerdeten Metallgehäuse montiert sein. Die AC-Eingangsleitung muss mit einem Filter EPCOS B84115-E-A30 (oder vergleichbar) ausgerüstet sein.
Die maximale Kabellänge von den Filtern zur S7-200 darf 25 cm betragen. Die Verkabelung der 24-V-DC-Stromquelle und der Geberversorgung muss geschirmt sein.

A

Technische Daten der CPUs

Tabelle A-2 Bestellnummern der CPUs

Bestellnummer	CPU-Variante	CPU-Spannungsversorgung (Nennspannung)	CPU-Eingänge	CPU-Ausgänge	Steckbarer Klemmenblock
6ES7 211-0AA22-0XB0	CPU 221	24 V DC	6 x 24 V DC	4 x 24 V DC	Nein
6ES7 211-0BA22-0XB0	CPU 221	120 bis 240 V AC	6 x 24 V DC	4 x Relais	Nein
6ES7 212-1AB22-0XB0	CPU 222	24 V DC	8 x 24 V DC	6 x 24 V DC	Nein
6ES7 212-1BB22-0XB0	CPU 222	120 bis 240 V AC	8 x 24 V DC	6 x Relais	Nein
6ES7 214-1AD22-0XB0	CPU 224	24 V DC	14 x 24 V DC	10 x 24 V DC	Ja
6ES7 214-1BD22-0XB0	CPU 224	120 bis 240 V AC	14 x 24 V DC	10 x Relais	Ja
6ES7 216-2AD22-0XB0	CPU 226	24 V DC	24 x 24 V DC	16 x 24 V DC	Ja
6ES7 216-2BD22-0XB0	CPU 226	120 bis 240 V AC	24 x 24 V DC	16 x Relais	Ja
6ES7 216-2AF22-0XB0	CPU 226XM	24 V DC	24 x 24 V DC	16 x 24 V DC	Ja
6ES7 216-2BF22-0XB0	CPU 226XM	120 bis 240 V AC	24 x 24 V DC	16 x Relais	Ja

Tabelle A-3 Allgemeine technische Daten der CPUS

Bestellnummer	Name und Beschreibung des Moduls	Abmessungen (mm) (B x H x T)	Gewicht	Verlust-leistung	Verfügbare V DC +5 V DC	+24 V DC
6ES7 211-0AA22-0XB0	CPU 221, DC/DC/DC, 6 Eingänge / 4 Ausgänge	90 x 80 x 62	270 g	3 W	0 mA	180 mA
6ES7 211-0BA22-0XB0	CPU 221 AC/DC/Relais, 6 Eingänge / 4 Relais	90 x 80 x 62	310 g	6 W	0 mA	180 mA
6ES7 212-1AB22-0XB0	CPU 222, DC/DC/DC, 8 Eingänge / 6 Ausgänge	90 x 80 x 62	270 g	5 W	340 mA	180 mA
6ES7 212-1BB22-0XB0	CPU 222 AC/DC/Relais 8 Eingänge / 6 Relais	90 x 80 x 62	310 g	7 W	340 mA	180 mA
6ES7 214-1AD22-0XB0	CPU 224, DC/DC/DC, 14 Eingänge / 10 Ausgänge	120,5 x 80 x 62	360 g	7 W	660 mA	280mA
6ES7 214-1BD22-0XB0	CPU 224 AC/DC/Relais 14 Eingänge / 10 Relais	120,5 x 80 x 62	410 g	10 W	660 mA	280mA
6ES7 216-2AD22-0XB0	CPU 226, DC/DC/DC, 24 Eingänge / 16 Ausgänge	196 x 80 x 62	550 g	11 W	1000 mA	400 mA
6ES7 216-2BD22-0XB0	CPU 226 AC/DC/Relais 24 Eingänge / 16 Relais	196 x 80 x 62	660 g	17 W	1000 mA	400 mA
6ES7 216-2AF22-0XB0	CPU 226XM DC/DC/DC 24 Eingänge / 16 Ausgänge	196 x 80 x 62	550 g	11 W	1000 mA	400 mA
6ES7 216-2BF22-0XB0	CPU 226XM AC/DC/Relais 24 Eingänge / 16 Relais	196 x 80 x 62	660 g	17 W	1000 mA	400 mA

Tabelle A-4 Technische Daten der CPUs

	CPU 221	CPU 222	CPU 224	CPU 226	CPU 226XM
Speicher					
Größe Anwenderprogramm (EEPROM)	4096 Bytes		8192 Bytes	8192 Bytes	16384 Bytes
Anwenderdaten (EEPROM)	2048 Bytes (nullspannungsfest gespeichert)		5120 Bytes (nullspannungsfest gespeichert)	5120 Bytes (nullspannungsfest gespeichert)	10240 Bytes (nullspannungsfest gespeichert)
Pufferung (Hochleistungskondensator) (optionale Batterie)	Typ. 50 Stunden (min. 8 Stunden bei 40 °C) Typ. 200 Tage		Typ. 100 Stunden (min. 70 Stunden bei 40 °C) Typ. 200 Tage		Typ. 190 Stunden (min. 120 Stunden bei 40 °C) Typ. 200 Tage
E/A					
Integrierte digitale Eingänge/Ausgänge	6 Eingänge / 4 Ausgänge	8 Eingänge / 6 Ausgänge	14 Eingänge / 10 Ausgänge	24 Eingänge / 16 Ausgänge	
Digitale E/A (Prozessabbild)	256 (128 Eingänge und 128 Ausgänge)				
Analoge E/A (Prozessabbild)	Keine	32 (16 Eingänge / 16 Ausgänge)	64 (32 Eingänge / 32 Ausgänge)		
Max. Anzahl Erweiterungsmodule	Keine	2 Module[1]	7 Module[1]		
Max. Anzahl zugelassene intelligente Module	Keine	2 Module[1]	7 Module[1]		
Eingänge für Impulsabgriff	6	8	14		
Schnelle Zähler Einphasig Zweiphasig	4 Zähler gesamt 4 bei 30 kHz / 2 bei 20 kHz		6 Zähler gesamt 6 bei 30 kHz / 4 bei 20 kHz		
Impulsausgänge	2 bei 20 kHz (nur DC-Ausgänge)				

	CPU 221	CPU 222	CPU 224	CPU 226	CPU 226XM
Allgemein					
Zeiten	256 Zeiten gesamt; 4 Zeiten (1 ms); 16 Zeiten (10 ms); 236 Zeiten (100 ms)				
Zähler	256 (gepuffert durch Hochleistungskondensator oder Batterie)				
Interne Merker beim Ausschalten gespeichert	256 (gepuffert durch Hochleistungskondensator oder Batterie) 112 (im EEPROM gespeichert)				
Zeitgesteuerte Interrupts	2 mit Auflösung von 1 ms				
Flankeninterrupts	4 steigende und/oder 4 fallende				
Analogpotentiometer	1 mit Auflösung von 8 Bit		2 mit Auflösung von 8 Bit		
Boolesche Ausführungszeiten	0,37 µs pro Operation				
Echtzeituhr	Optionales Steckmodul		Integriert		
Optionale Steckmodule	Speicher, Batterie oder Echtzeituhr		Speicher und Batterie		
Integrierte Kommunikation					
Anschlüsse	1 RS-485-Anschluss			2 RS-485-Anschlüsse	
PPI, DP/T Baudraten	9,6, 19,2, 187,5 kBaud				
Baudraten frei programmierbare Kommunikation	1,2 kBaud bis 115,2 kBaud				
Max. Kabellänge pro Segment	Mit potentialgetrenntem Busverstärker: 1000 m bis zu 187,5 kBaud, 1200 m bis zu 38,4 kBaud Ohne potentialgetrennten Busverstärker: 50 m				
Max. Anzahl Stationen	32 je Segment, 126 je Netz				
Max. Anzahl Master	32				
Punkt-zu-Punkt (PPI-Master-Modus)	Ja (NETR/NETW)				
MPI-Verbindungen	4 gesamt, 2 reserviert (1 für ein PG und 1 für ein OP)				

1 Sie müssen die Leistungsbilanz berechnen, um zu ermitteln, wieviel Leistung (bzw. Strom) die S7-200 CPU für Ihre Konfiguration liefern kann. Wird die Leistungsbilanz der CPU überschritten, können Sie evtl. nicht die maximal zulässige Anzahl Module anschließen. Im Anhang A finden Sie den Leistungsbedarf der CPUs und der Erweiterungsmodule, und im Anhang B können Sie die Leistungsbilanz berechnen.

Tabelle A-5 Leistungsdaten der CPUs

	DC		AC	
Eingangsleistung				
Eingangsspannung	20,4 bis 28,8 V DC		85 bis 264 V AC (47 bis 63 Hz)	
Eingangsstrom	CPU nur bei 24 V DC	Max. Last bei 24 V DC	Nur CPU	Max. Last
CPU 221	80 mA	450 mA	30/15 mA bei 120/240 V AC	120/60 mA bei 120/240 V AC
CPU 222	85 mA	500 mA	40/20 mA bei 120/240 V AC	140/70 mA bei 120/240 V AC
CPU 224	110 mA	700 mA	60/30 mA bei 120/240 V AC	200/100 mA bei 120/240 V AC
CPU 226/CPU 226XM	150 mA	1050 mA	80/40 mA bei 120/240 V AC	320/160 mA bei 120/240 V AC
Einschaltstromstoß	10 A bei 28,8 V DC		20 A bei 264 V AC	
Elektrische Trennung (Feld zu Logik)	Nicht elektrisch getrennt		1500 V AC	
Verzögerungszeit (Spannungsverlust)	10 ms bei 24 V DC		20/80 ms bei 120/240 V AC	
Sicherung (nicht austauschbar)	3 A, 250 V, träge		2 A, 250 V, träge	
24-V-DC-Geberspannung				
Geberspannung	L+ minus 5 V		20,4 bis 28,8 V DC	
Strombegrenzung	1,5 A Spitze, zerstörungsfreie Wärmegrenze (Nennlast in Tabelle A-3.)			
Welligkeit/Störströme	Abgeleitet aus Eingangsspannung		Weniger als 1 V Spitze zu Spitze	
Elektrische Trennung (Geber zu Logik)	Nicht elektrisch getrennt			

Tabelle A-6 Technische Daten der CPU-Eingänge

Allgemein	24-V-DC-Eingang
Datentyp	Stromziehend/stromliefernd (IEC Typ 1 stromziehend)
Nennspannung	Typ. 24 V DC bei 4 mA
Max. zulässige Dauerspannung	30 V DC
Stoßspannung	35 V DC für 0,5 s
Logik 1 (min.)	15 V DC bei 2,5 mA
Logik 0 (max.)	5 V DC bei 1 mA
Eingangsverzögerung	Einstellbar (0,2 bis 12,8 ms) CPU 226, CPU 226XM: Eingänge E1.6 bis E2.7 haben eine feste Verzögerung (4,5 ms)

Tabelle A-6 Technische Daten der CPU-Eingänge, Fortsetzung

Allgemein	24-V-DC-Eingang	
Anschluss 2-Draht-Näherungssensor (Bero)		
Zulässiger Kriechstrom (max.)	1 mA	
Elektrische Trennung (Feld zu Logik)	Ja	
Galvanische Trennung	500 V AC für 1 Minute	
Potentialgetrennte Gruppen	Siehe Schaltplan	
Frequenz Schnelle Zähler (max.)	Einphasen-Zähler	Zweiphasen-Zähler
Logik 1 = 15 bis 30 V DC	20 kHz	10 kHz
Logik 1 = 15 bis 26 V DC	30 kHz	20 kHz
Gleichzeitig eingeschaltete Eingänge	Alle bei 55 °C	
Kabellänge (max.)		
Geschirmt	500 m normale Eingänge, 50 m HSC-Eingänge	
Nicht geschirmt	300 m normale Eingänge	

Tabelle A-7 Technische Daten der CPU-Ausgänge

Allgemein	24-V-DC-Ausgang	Relaisausgang
Datentyp	MOSFET, elektronisch[1]	Schwachstromkontakt
Nennspannung	24 V DC	24 V DC oder 250 V AC
Spannungsbereich	20,4 bis 28,8 V DC	5 bis 30 V DC oder 5 bis 250 V AC
Stoßstrom (max.)	8 A für 100 ms	7 A bei geschlossenen Kontakten
Logik 1 (min.)	20 V DC bei Maximalstrom	-
Logik 0 (max.)	0,1 V DC bei 10 kΩ Last	-
Nennstrom je Ausgang (max.)	0,75 A	2,0 A
Nennstrom je Leiter (max.)	6 A	10 A
Kriechstrom (max.)	10 µ A	-
Lampenlast (max.)	5 W	30 W DC; 200 W AC
Induktive Klemmspannung	L+ minus 48 V DC, 1 W Verlustleistung	-
Widerstand bei EIN (Kontakt)	0,3 Ω typ.	0,2 Ω (max. wenn neu)
Trennung		
Galvanische Trennung (Feld zu Logik)	500 V AC für 1 Minute	-
Logik zu Kontakt	-	1500 V AC für 1 Minute
Widerstand (Logik zu Kontakt)	-	100 M Ω
Potentialgetrennte Gruppen	Siehe Schaltplan	Siehe Schaltplan
Verzögerung Aus - Ein/Ein - Aus (max.)	2/10 µs (A0.0 und A0.1) 15/100 µs (alle anderen)	-
Schaltvorgänge (max.)	-	10 ms
Impulsfrequenz (max.) A0.0 und A0.1	20 kHz	1 Hz
Schaltzyklen mechanische Lebensdauer	-	10.000.000 (ohne Last)
Lebensdauer Kontakte	-	100.000 (Bemessungslast)
Gleichzeitig eingeschaltete Ausgänge	Alle bei 55 °C	Alle bei 55 °C
Parallelschalten von 2 Ausgängen	Ja	Nein
Kabellänge (max.)		
Geschirmt	500 m	500 m
Ungeschirmt	150 m	150 m

1 Wenn ein mechanischer Kontakt die Ausgangsspannung zur S7-200 CPU oder einem digitalen Erweiterungsmodul einschaltet, wird ca. 50 Mikrosekunden lang das Signal 1 an die Digitalausgänge gesendet. Dies müssen Sie berücksichtigen, vor allem, wenn Sie mit Geräten arbeiten, die auf kurze Impulse reagieren.

Schaltpläne

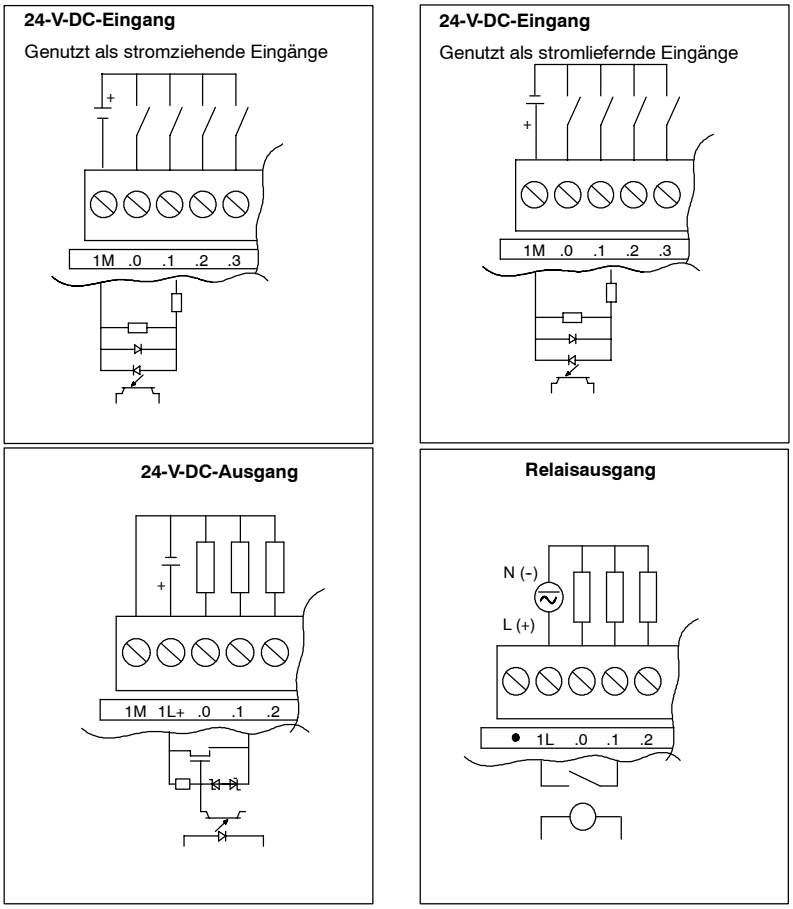


Bild A-2 Eingänge und Ausgänge der CPU

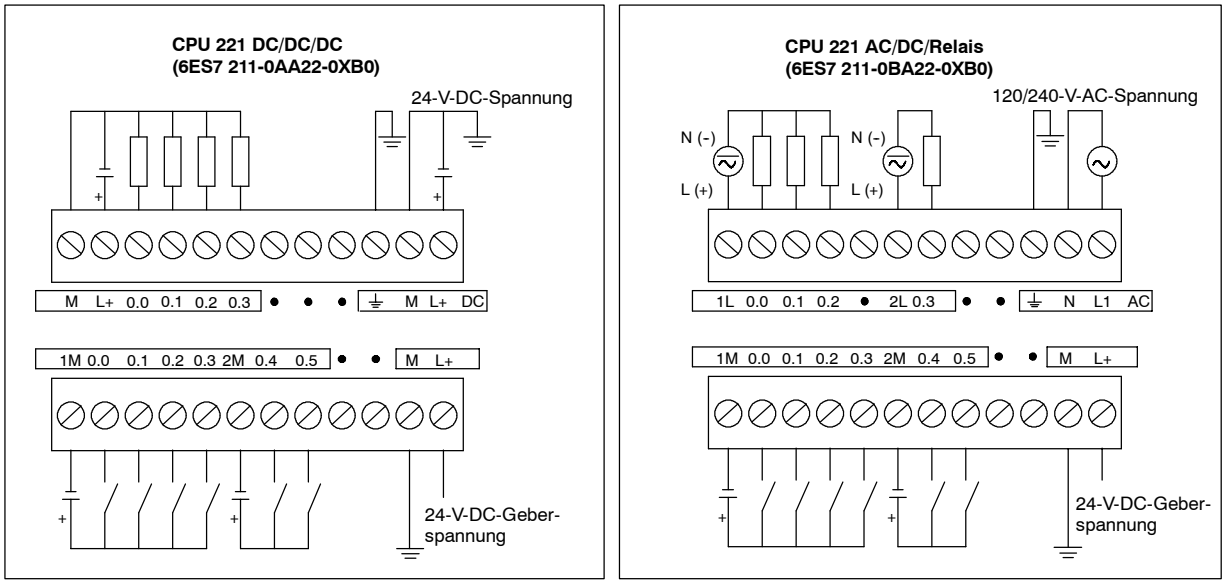


Bild A-3 Schaltpläne CPU 221

A

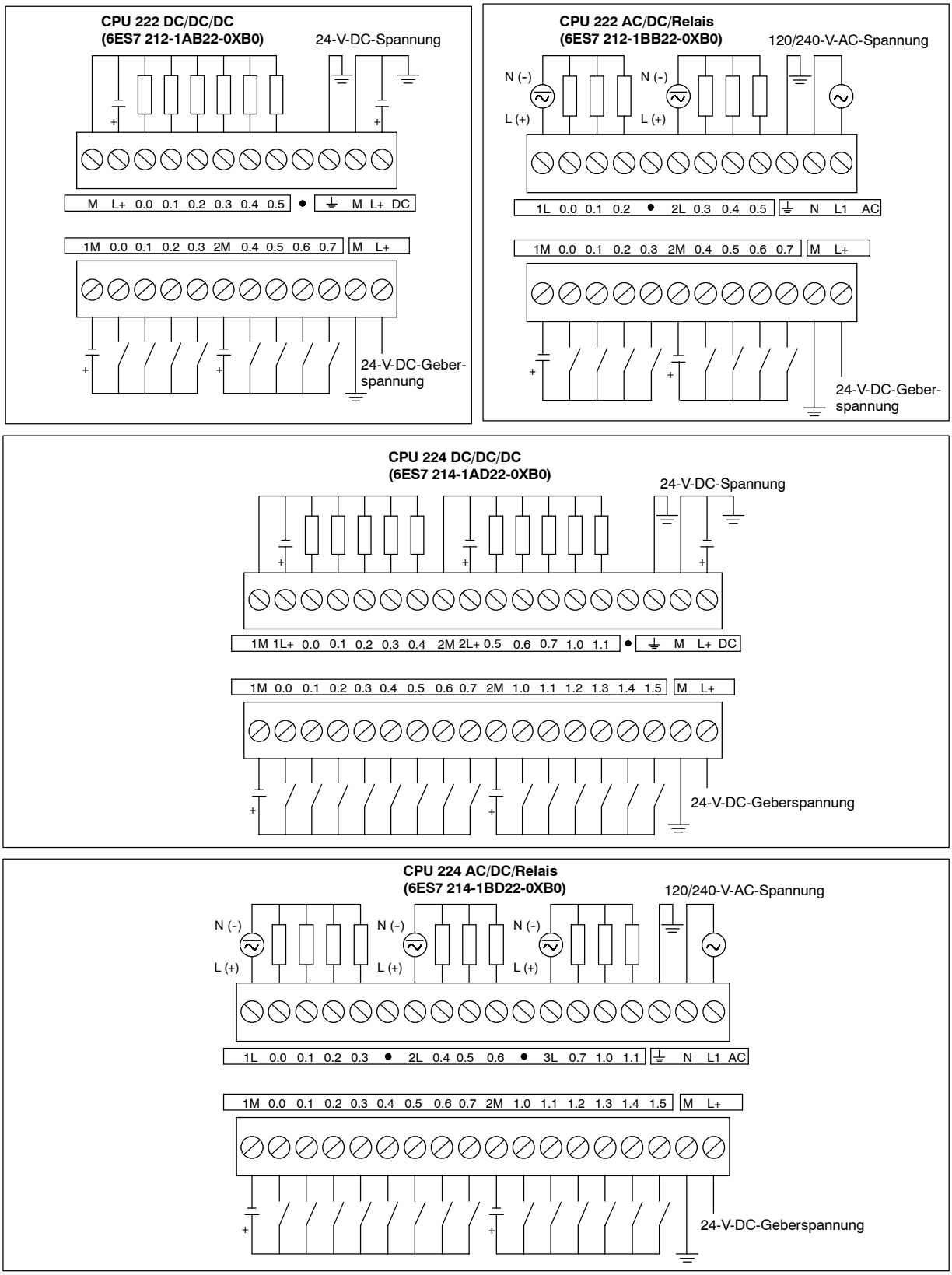


Bild A-4 Schaltpläne der CPU 222 und der CPU 224

A

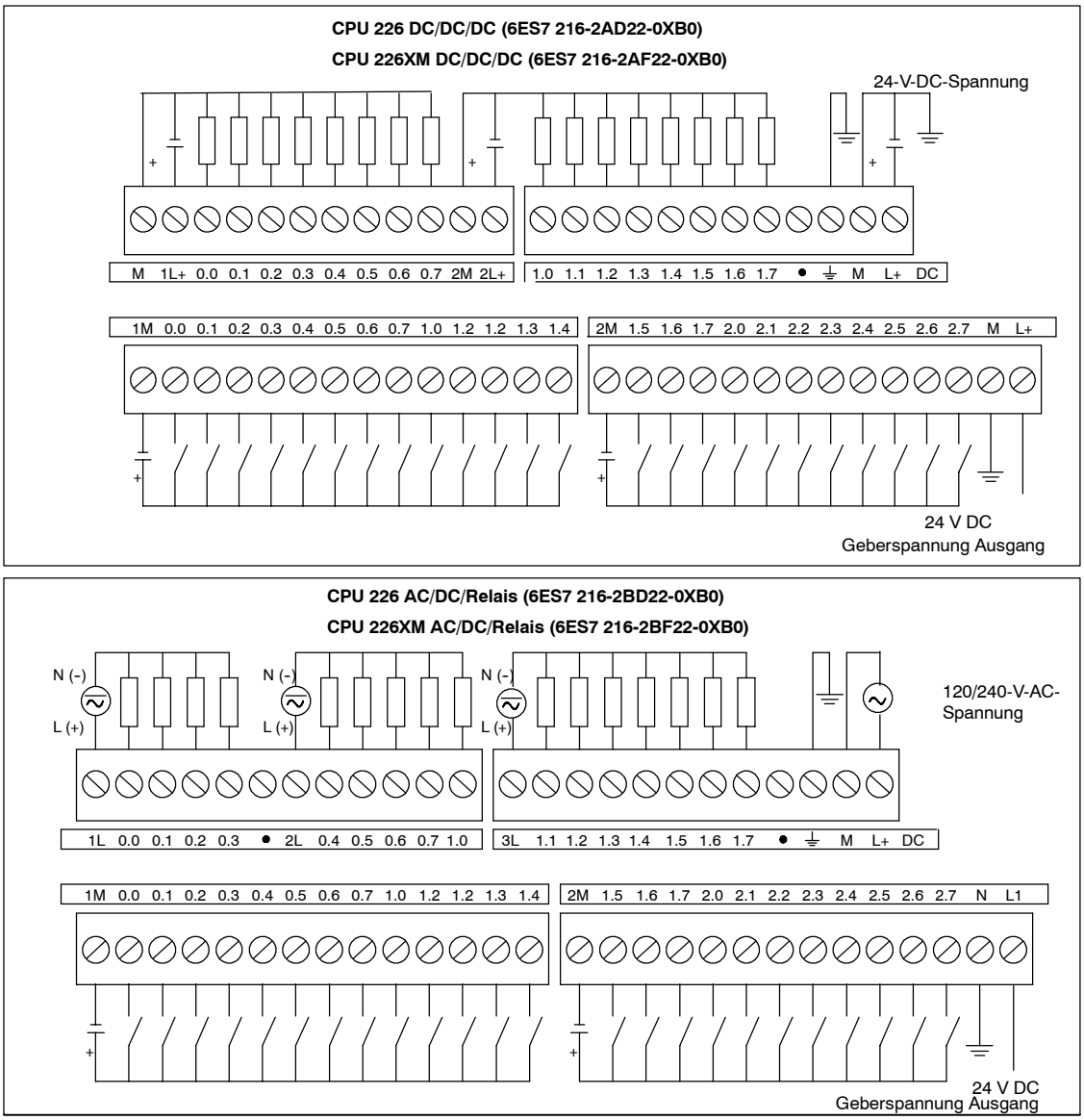
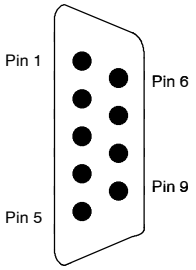


Bild A-5 Schaltpläne der CPU 226 und der CPU226XM

Tabelle A-8 Anschlussbelegung der S7-200 Kommunikationsschnittstelle

Klemmenblock	Pin	PROFIBUS-Signal	Schnittstelle 0/Schnittstelle 1
	1	Schirmung	Erdungsanschluss
	2	24-V-Rückleiter	Logischer Leiter
	3	RS-485 Signal B	RS-485 Signal B
	4	Anforderung zum Senden	RTS (TTL)
	5	5-V-Rückleiter	Logischer Leiter
	6	+5 V	+5 V, 100-Ω-Reihenwiderstand
	7	+24 V	+24 V
	8	RS-485 Signal A	RS-485 Signal A
	9	-/-	10-Bit-Protokoll (Eingang)
	Gehäuse des Steckverbinders	Schirmung	Erdungsanschluss

Technische Daten der digitalen Erweiterungsmodule

Tabelle A-9 Bestellnummern der digitalen Erweiterungsmodule

Bestellnummer	Erweiterungsmodule	EM-Eingänge	EM-Ausgänge	Steckbarer Klemmenblock
6ES7 221-1BF22-0XA0	EM 221 Digitaleingabe 8 x 24 V DC	8 x 24 V DC	–	Ja
6ES7 221-1EF22-0XA0	EM 221 Digitaleingabe 8 x AC 120/230 V	8 x AC 120/230V	–	Ja
6ES7 221-1BH22-0XA0	EM 221 Digitaleingabe 16 x 24 V DC	16 x 24 V DC	–	Ja
6ES7 222-1BD22-0XA0	EM 222 Digitalausgabe 4 x 24 V DC - 5 A	–	4 x 24 V DC - 5 A	Ja
6ES7 222-1BF22-0XA0	EM 222 Digitalausgabe 8 x 24 V DC	–	8 x 24 V DC - 0,75 A	Ja
6ES7 222-1HD22-0XA0	EM 222 Digitalausgabe 4 x Relais -10 A	–	4 x Relais - 10 A	Ja
6ES7 222-1HF22-0XA0	EM 222 Digitalausgabe 8 x Relais	–	8 x Relais - 10 A	Ja
6ES7 222-1EF22-0XA0	EM 222 Digitalausgabe 8 x AC 120/230 V	–	8 x AC 120/230 V	Ja
6ES7 223-1BF22-0XA0	EM 223 24 V DC, Digitalein-/Digitalausgabe 4 Eingänge / 4 Ausgänge	4 x 24 V DC	4 x 24 V DC - 0,75 A	Ja
6ES7 223-1HF22-0XA0	EM 223 24 V DC, Digitalein-/Digitalausgabe 4 Eingänge / 4 Relaisausgänge	4 x 24 V DC	4 x Relais - 2 A	Ja
6ES7 223-1BH22-0AX0	EM 223 24 V DC, Digitalein-/Digitalausgabe 8 Eingänge / 8 Ausgänge	8 x 24 V DC	8 x 24 V DC - 0,75 A	Ja
6ES7 223-1PH22-0XA0	EM 223 24 V DC, Digitalein-/Digitalausgabe 8 Eingänge / 8 Relaisausgänge	8 x 24 V DC	8 x Relais - 2 A	Ja
6ES7 223-1BL22-0XA0	EM 223 24 V DC, Digitalein-/Digitalausgabe 16 Eingänge / 16 Ausgänge	16 x 24 V DC	16 x 24 V DC - 0,75 A	Ja
6ES7 223-1PL22-0XA0	EM 223 24 V DC, Digitalein-/Digitalausgabe 16 Eingänge / 16 Relaisausgänge	16 x 24 V DC	16 x Relais - 2 A	Ja

Tabelle A-10 Allgemeine technische Daten der digitalen Erweiterungsmodule

Bestellnummer	Name und Beschreibung des Moduls	Abmessungen (mm) (B x H x T)	Gewicht	Verlust-leistung	V-DC-Bedarf +5 V DC	V-DC-Bedarf +24 V DC
6ES7 221-1BF22-0XA0	EM 221 DI 8 x 24 V DC	46 x 80 x 62	150 g	2 W	30 mA	–
6ES7 221-1EF22-0XA0	EM 221 DI 8 x AC 120/230 V	71,2 x 80 x 62	160 g	3 W	30 mA	–
6ES7 221-1BH22-0XA0	EM 221 DI 16 x 24 V DC	71,2 x 80 x 62	160 g	3 W	70 mA	–
6ES7 222-1BD22-0XA0	EM 222 DO 4 x 24 VDC-5A	46 x 80 x 62	120 g	3 W	40 mA	–
6ES7 222-1BF22-0XA0	EM 222 DO 8 x 24 V DC	46 x 80 x 62	150 g	2 W	50 mA	–
6ES7 222-1HD22-0XA0	EM 222 DO 4 x Relais - 10 A	46 x 80 x 62	150 g	4 W	30 mA	EIN: 20 mA / Ausgang, 20,4 bis 28,8 V DC
6ES7 222-1HF22-0XA0	EM 222 DO 8 x Relais	46 x 80 x 62	170 g	2 W	40 mA	EIN: 9 mA / Ausgang, 20,4 bis 28,8 V DC
6ES7 222-1EF22-0XA0	EM 222 DO 8 x AC 120/230 V	71,2 x 80 x 62	165 g	4 W	110 mA	–
6ES7 223-1BF22-0XA0	EM 223 24 V DC 4 Eingänge/4 Ausgänge	46 x 80 x 62	160 g	2 W	40 mA	–
6ES7 223-1HF22-0XA0	EM 223 24 V DC 4 Eingänge/4 Relais	46 x 80 x 62	170 g	2 W	40 mA	EIN: 9 mA / Ausgang, 20,4 bis 28,8 V DC
6ES7 223-1BH22-0AX0	EM 223 24 V DC 8 Eingänge/8 Ausgänge	71,2 x 80 x 62	200 g	3 W	80 mA	–
6ES7 223-1PH22-0XA0	EM 223 24 V DC 8 Eingänge/8 Relais	71,2 x 80 x 62	300 g	3 W	80 mA	EIN: 9 mA / Ausgang, 20,4 bis 28,8 V DC
6ES7 223-1BL22-0XA0	EM 223 24 V DC 16 Eingänge/16 Ausgänge	137,3 x 80 x 62	360 g	6 W	160 mA	–
6ES7 223-1PL22-0XA0	EM 223 24 V DC 16 Eingänge/16 Relais	137,3 x 80 x 62	400 g	6 W	150 mA	EIN: 9 mA / Ausgang, 20,4 bis 28,8 V DC

Tabelle A-11 Technische Daten der Eingänge von Digitalerweiterungsmodulen

Allgemein	24-V-DC-Eingang	120/230-V-AC-Eingang (47 bis 63 HZ)
Datentyp	Stromziehend/stromliefernd (IEC Typ 1, wenn stromziehend)	IEC Typ I
Nennspannung	24 V DC bei 4 mA	120 V AC bei 6 mA oder 230 V AC bei 9 mA Bemessung
Maximal zulässige Dauerspannung	30 V DC	264 V AC
Stoßspannung (max.)	35 V DC für 0,5 s	–
Logik 1 (min.)	15 V DC bei 2,5 mA	79 V AC bei 2,5 mA
Logik 0 (max.)	5 V DC bei 1 mA	20 V AC oder 1 mA AC
Eingabeverzögerung (max.)	4,5 ms	15 ms
Anschluss 2-Draht-Näherungssensor (Bero) Zulässiger Kriechstrom (max.)	 1 mA	 1 mA AC
Potentialtrennung Galvanische Trennung (Feld zu Logik) Potentialgetrennte Gruppen	 500 V AC für 1 Minute Siehe Schaltplan	 1500 V AC für 1 Minute 1 Eingang
Gleichzeitig eingeschaltete Eingänge	Alle bei 55 °C	Alle bei 55 °C
Kabellänge (max.) Geschirmt Ungeschirmt	 500 m 300 m	 500 m 300 m

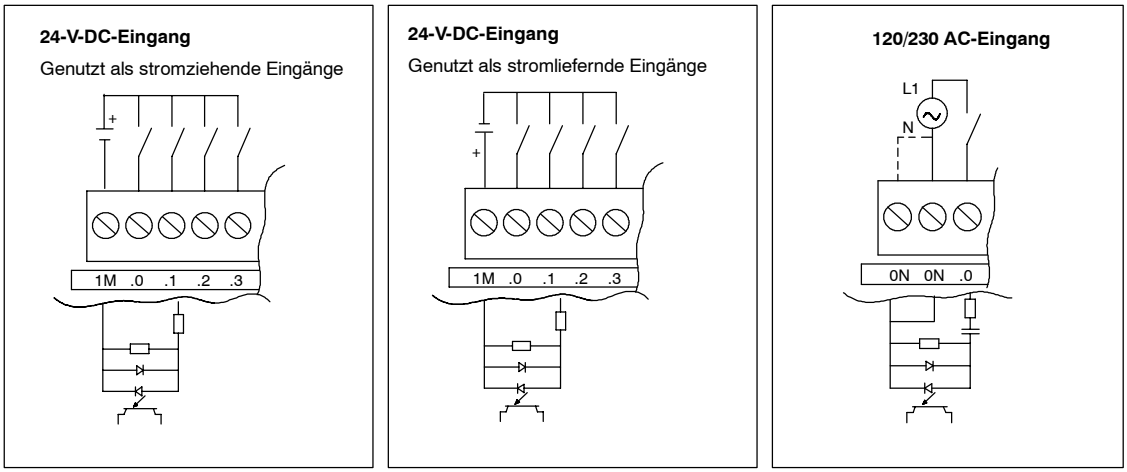


Bild A-6 Eingänge der digitalen S7-200 Erweiterungsmodule

A

Tabelle A-12 Technische Daten der Ausgänge von Digitalerweiterungsmodulen

Allgemein	24-V-DC-Ausgang		Relaisausgang		120/230-V-AC-Ausgang
	0,75 A	5 A	2 A	10 A	
Datentyp	MOSFET, elektronisch[1]		Schwachstromkontakt		Triac, Nulldurchgang-Einschaltung[2]
Nennspannung	24 V DC		24 V DC oder 250 V AC		120/230 V AC
Spannungsbereich	20,4 bis 28,8 V DC		5 bis 30 V DC oder 5 bis 250 V AC	12 bis 30 V DC oder 12 bis 250 V AC	40 bis 264 V AC (47 bis 63 Hz)
24-V-DC-Spulenspannungsbereich	–		20,4 bis 28,8 V DC		-
Stoßstrom (max.)	8 A für 100 ms	30 A	7 A bei geschlossenen Kontakten	15 A für 4 s bei Betriebsspiel von 10%	5 A Effektivwert für 2 AC-Zyklen
Logik 1 (min.)	20 V DC		–		L1 (-0,9 V Effektivwert)
Logik 0 (max.)	0,1 V DC	0,2 V DC	–		–
Nennstrom je Ausgang (max.)	0,75 A	5 A	2,00 A	10 A ohmsch; 2 A DC induktiv; 3 A AC induktiv	0,5 A AC[3]
Nennstrom je Leiter (max.)	6 A	5 A	8 A	10 A	0,5 A AC
Kriechstrom (max.)	10 µA	30 µA	–		1,1 mA Effektivwert bei 132 V AC und 1,8 mA Effektivwert bei 264 V AC
Lampenlast (max.)	5 W	50 W	30 W DC/ 200 W AC	100 W DC/ 1000 W AC	60 W
Induktive Klemmspannung	L+ minus 48 V	L + minus 47 V[4]	–		–
Widerstand EIN (Kontakt)	0,3 Ω typ.	0,05 Ω max.	0,2 Ω max. wenn neu	0,1 Ω max. wenn neu	410 Ω µ max. wenn Ladestrom kleiner als 0,05 A
Trennung					
Galvanische Trennung (Feld zu Logik)	500 V AC für 1 Minute		–		1500 V AC für 1 Minute
Spule zu Logik	–		Keine		–
Spule zu Kontakt	–		1500 V AC für 1 Minute		–
Widerstand (Spule zu Kontakt)	–		Min. 100 MΩ , wenn neu		–
Potentialgetrennte Gruppen	Siehe Schaltplan		Siehe Schaltplan		1 Ausgang
Verzögerung Aus-Ein/Ein-Aus (max.)	50 µs / 200 µs	500 µs	–	–	0,2 ms + 1/2 AC-Zyklus
Schaltvorgänge (max.)	–	–	10 ms	15 ms	–
Schaltfrequenz (max.)	–		1 Hz		10 Hz
Schaltzyklen mechanische Lebensdauer	–		10.000.000 (ohne Last)	30.000.000 (ohne Last)	–
Lebensdauer Kontakte	–		100.000 (Bemessungslast)	30.000 (Bemessungslast)	–
Gleichzeitig eingeschaltete Ausgänge	Alle bei 55 °C		Alle bei 55 °C	Alle bei 55 °C mit 20 A max. Modulstrom[5] Alle bei 40 °C mit 10 A je Ausgang	Alle bei 55 °C
Parallelschalten von 2 Ausgängen	Ja		Nein		Nein
Kabellänge (max.)					
Geschirmt	500 m		500 m		500 m
Ungeschirmt	150 m		150 m		150 m

1 Wenn ein mechanischer Kontakt die Ausgangsspannung zur S7-200 CPU oder einem digitalen Erweiterungsmodul einschaltet, wird ca. 50 Mikrosekunden lang das Signal 1 an die Digitalausgänge gesendet. Dies müssen Sie berücksichtigen, vor allem, wenn Sie mit Geräten arbeiten, die auf kurze Impulse reagieren.

2 Wenn ein mechanischer Kontakt die Ausgangsspannung zum AC-Erweiterungsmodul einschaltet, wird ca. 1/2 AC-Zyklus lang das Signal 1 an die AC-Ausgänge gesendet. Dies müssen Sie berücksichtigen.

3 Beim Laststrom muss es sich wegen der Nulldurchgangsschaltung um Vollwellen-Gleichspannung und nicht um Halbwellen-Gleichspannung handeln. Der Mindestlaststrom beträgt 0,05 A AC. Bei einem Laststrom zwischen 5 mA und 50 mA AC kann der Strom gesteuert werden, doch es gibt einen zusätzlichen Spannungsabfall aufgrund eines Reihenwiderstands von 410 Ohm.

4 Wird der Ausgang aufgrund übermäßigen induktiven Schaltens oder unnormaler Bedingungen zu heiß, schaltet sich der Ausgang möglicherweise aus oder wird beschädigt. Der Ausgang kann überhitzt oder beschädigt werden, wenn er beim Ausschalten einer induktiven Laster einer größeren Energie als 0,7 J ausgesetzt wird. Um diesen Grenzwert ignorieren zu können, kann eine Schutzbeschaltung, die in Kapitel 3 beschrieben ist, parallel zur Last geschaltet werden. Diese Komponenten müssen entsprechend der jeweiligen Anwendung bemessen werden.

5 Das EM 222 DO 4 x Relais verfügt über eine andere FM-Bemessung als der Rest der S7-200 Module. Dieses Modul hat eine T4-Bemessung und keine T4A für Gefahrenbereiche der FM-Klasse I, Divisionsgruppen A, B, C und D.

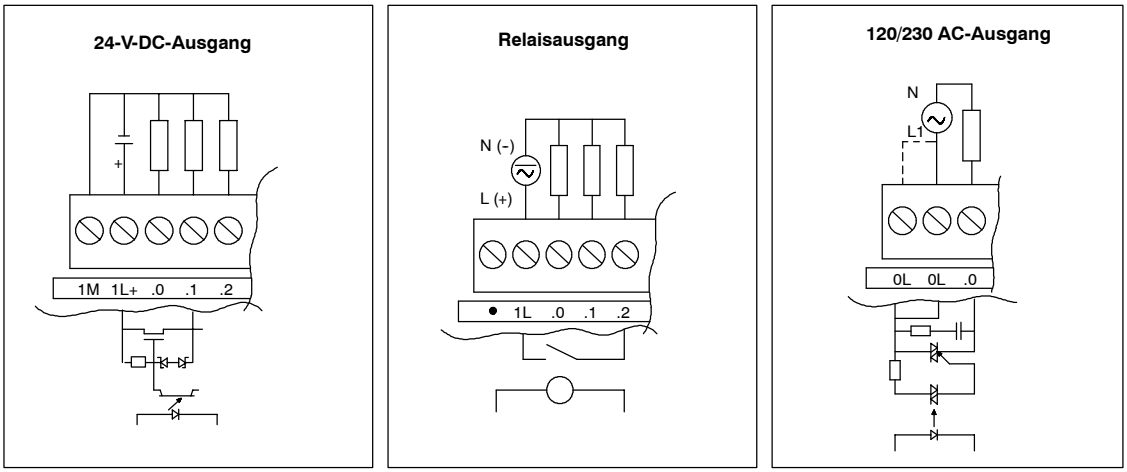


Bild A-7 Ausgänge der digitalen S7-200 Erweiterungsmodule

Schaltpläne

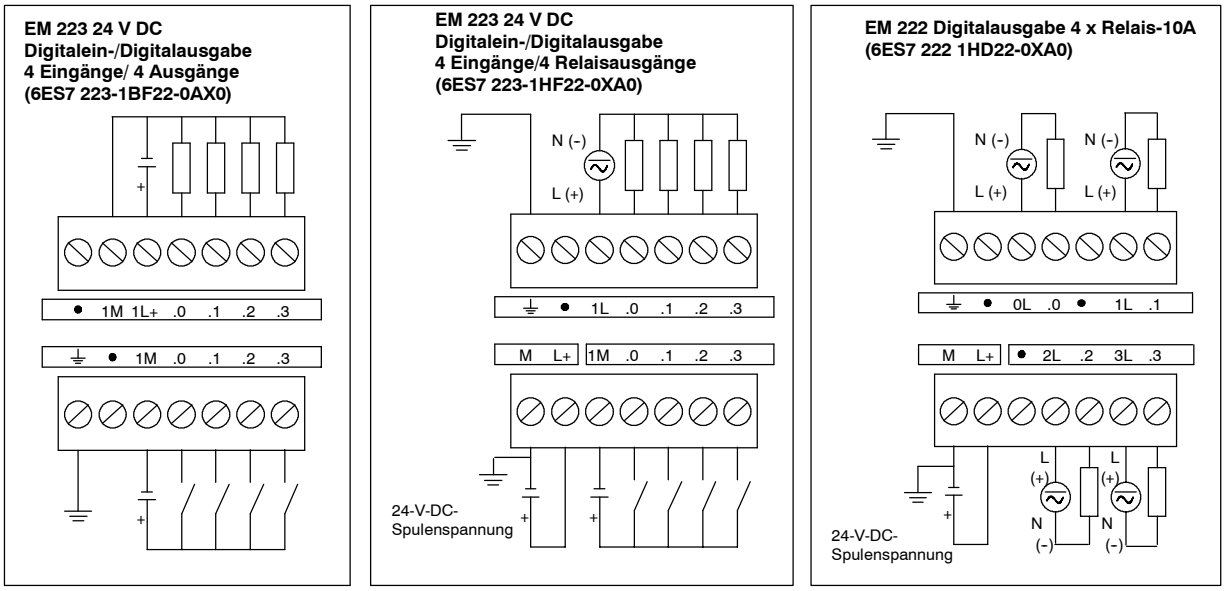


Bild A-8 Schaltpläne für die Erweiterungsmodule EM 222 und EM 223

A

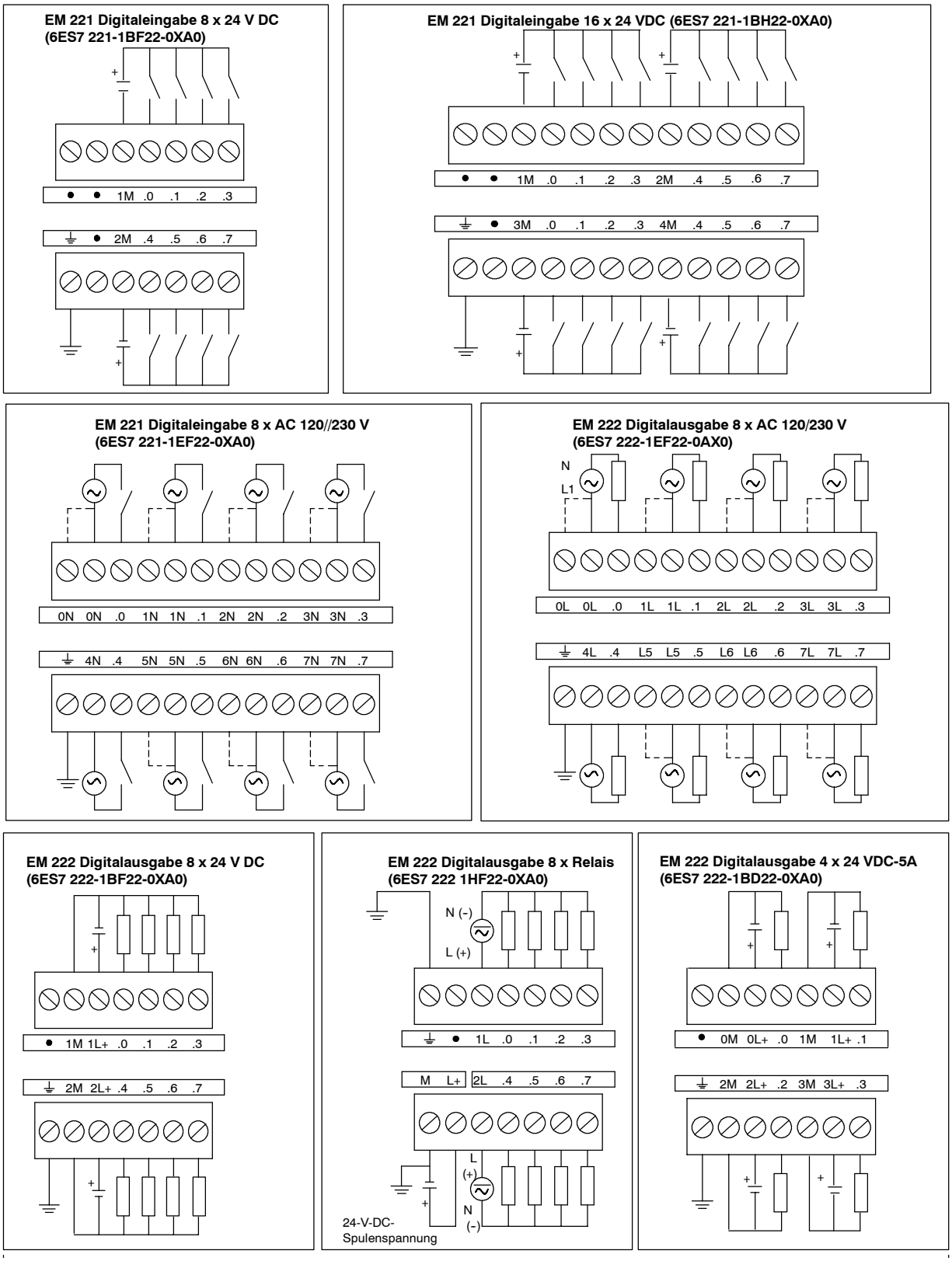
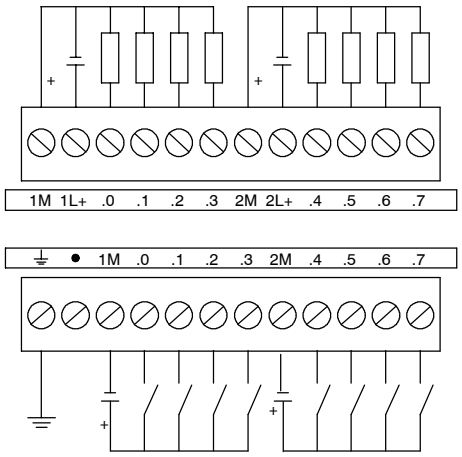


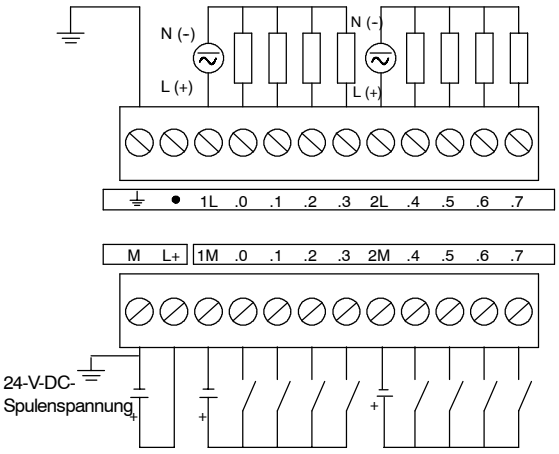
Bild A-9 Schaltpläne für die Erweiterungsmodule EM 221 und EM 222

A

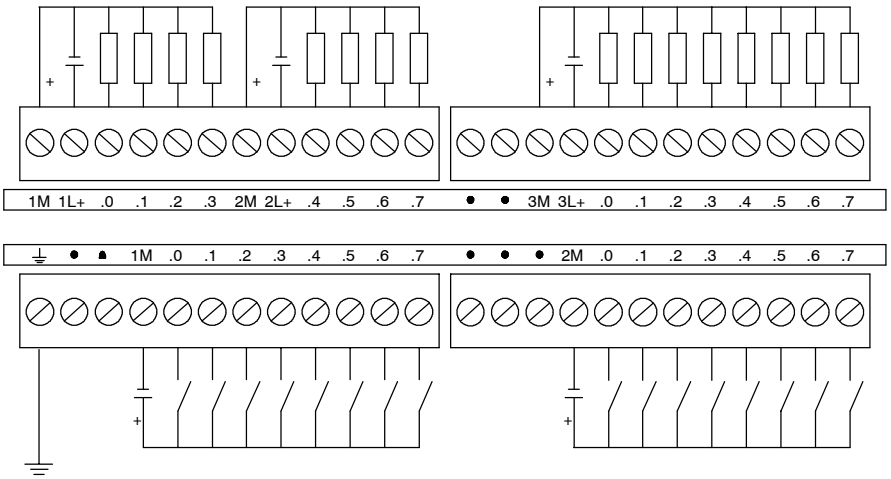
EM 223 24 V DC Digitalein-/Digitalausgabe
8 Eingänge/8 Ausgänge (6ES7 223-1BH22-0XA0)



EM 223 24 V DC Digitalein-/Digitalausgabe
8 Eingänge/8 Relaisausgänge (6ES7 223-1PH22-0XA0)



EM 223 24 V DC Digitalein-/Digitalausgabe 16 Eingänge/16 Ausgänge
(6ES7 223-1BL22-0XA0)



EM 223 24 V DC Digitalein-/Digitalausgabe 16 Eingänge/16 Relaisausgänge
(6ES7 223-1PL22-0XA0)

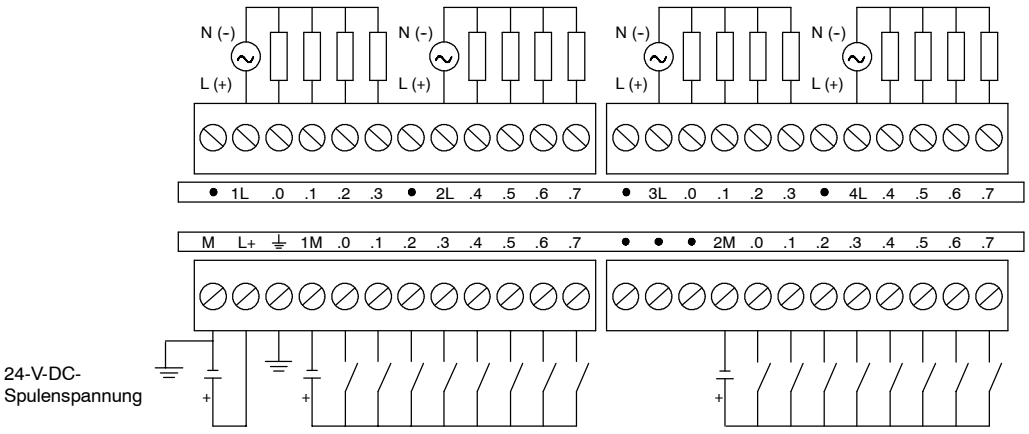


Bild A-10 Schaltpläne für das Erweiterungsmodul EM 223

Technische Daten der analogen Erweiterungsmodule

Tabelle A-13 Bestellnummern der analogen Erweiterungsmodule

Bestellnummer	Erweiterungsmodule	EM-Eingänge	EM-Ausgänge	Steckbarer Klemmenblock
6ES7 231-0HC22-0XA0	EM 231 Analogeingabe, 4 Eingänge	4	–	Nein
6ES7 232-0HB22-0XA0	EM 232 Analogausgabe, 2 Ausgänge	–	2	Nein
6ES7 235-0KD22-0XA0	EM 235 Analogein-/Analogausgabe 4 Eingänge/1 Ausgang	4	1[1]	Nein

1 In der CPU sind 2 Analogausgänge für dieses Modul vorgesehen.

Tabelle A-14 Allgemeine technische Daten der analogen Erweiterungsmodule

Bestellnummer	Name und Beschreibung des Moduls	Abmessungen (mm) (B x H x T)	Ge-wicht	Verlust-leistung	V-DC-Bedarf +5 V DC	V-DC-Bedarf +24 V DC
6ES7 231-0HC22-0XA0	EM 231 Analogeingabe, 4 Eingänge	71,2 x 80 x 62	183 g	2 W	20 mA	60 mA
6ES7 232-0HB22-0XA0	EM 232 Analogausgabe, 2 Ausgänge	46 x 80 x 62	148 g	2 W	20 mA	70 mA (mit beiden Ausgängen bei 20 mA)
6ES7 235-0KD22-0XA0	EM 235 Analogein-/Analogausgabe 4 Eingänge/1 Ausgang	71,2 x 80 x 62	186 g	2 W	30 mA	60 mA (mit Ausgang bei 20 mA)

Tabelle A-15 Technische Daten der Eingänge von analogen Erweiterungsmodulen

Allgemein	6ES7 231-0HC22-0XA0	6ES7 235-0KD22-0XA0
Datenwortformat	(siehe Bild A-13)	(siehe Bild A-13)
Zweipolig, Vollausschlagsbereich	-32000 bis +32000	-32000 bis +32000
Einpolig, Vollausschlagsbereich	0 bis 32000	0 bis 32000
DC-Eingangsimpedanz	≥ 10 MΩ Spannungseingang, 250 Ω Stromeingang	≥ 10 MΩ Spannungseingang, 250 Ω Stromeingang
Eingabefilterdämpfung	-3 db bei 3,1 kHz	-3 db bei 3,1 kHz
Max. Eingangsspannung	30 V DC	30 V DC
Max. Eingangsstrom	32 mA	32 mA
Auflösung	12-Bit-A/D-Umsetzung	12-Bit-A/D-Umsetzung
Elektrische Trennung (Feld zu Logik)	Keine	Keine
Eingangstyp	Differentialeingang	Differentialeingang
Eingangsbereiche		
Spannung (einpolig)	0 bis 10 V, 0 bis 5 V	0 bis 10 V, 0 bis 5 V, 0 bis 1 V, 0 bis 500 mV, 0 bis 100 mV, 0 bis 50 mV
Spannung (zweipolig)	±5 V, ± 2,5 V	±10 V, ±5 V, ±2,5 V, ±1 V, ±500 mV, ±250 mV, ±100 mV, ±50 mV, ±25 mV
Strom	0 bis 20 mA	0 bis 20 mA
Eingangsauflösung	Siehe Tabelle A-18	Siehe Tabelle A-19
Spannung (einpolig)		
Spannung (zweipolig)		
Strom		
Analog-Digital-Umsetzzeit	< 250 µs	< 250 µs
Schrittreaktion Analogeingang	1,5 ms bis 95%	1,5 ms bis 95%
Gleichtaktunterdrückung	40 dB, Nennwert bei 60 Hz	40 dB, Nennwert bei 60 Hz
Gleichtaktspannung	Signalspannung plus Gleichtaktspannung muss ≤ ±12 V betragen	Signalspannung plus Gleichtaktspannung muss ≤ ±12 V betragen
24-V-DC-Versorgungsspannungsbereich	20,4 bis 28,8	20,4 bis 28,8

Tabelle A-16 Technische Daten der Ausgänge von analogen Erweiterungsmodulen

Allgemein	6ES7 232-0HB22-0XA0	6ES7 235-0KD22-0XA0
Elektrische Trennung (Feld zu Logik)	Keine	Keine
Signalbereich		
Spannungsausgabe	± 10 V	± 10 V
Stromausgabe	0 bis 20 mA	0 bis 20 mA
Auflösung, Vollausschlag		
Spannung	12 Bits	12 Bits
Strom	11 Bits	11 Bits
Datenwortformat		
Spannung	-32000 bis +32000	-32000 bis +32000
Strom	0 bis +32000	0 bis +32000
Genauigkeit		
Ungünstigster Fall, 0 °C bis 55 °C		
Spannungsausgabe	± 2% des Vollausschlags	± 2% des Vollausschlags
Stromausgabe	± 2% des Vollausschlags	± 2% des Vollausschlags
Typ. 25 °C		
Spannungsausgabe	± 0,5% des Vollausschlags	± 0,5% des Vollausschlags
Stromausgabe	± 0,5% des Vollausschlags	± 0,5% des Vollausschlags
Ausregelzeit		
Spannungsausgabe	100 µS	100 µS
Stromausgabe	2 ms	2 ms
Maximalantrieb		
Spannungsausgabe	Min. 5000 Ω	Min. 5000 Ω
Stromausgabe	Max. 500 Ω	Max. 500 Ω

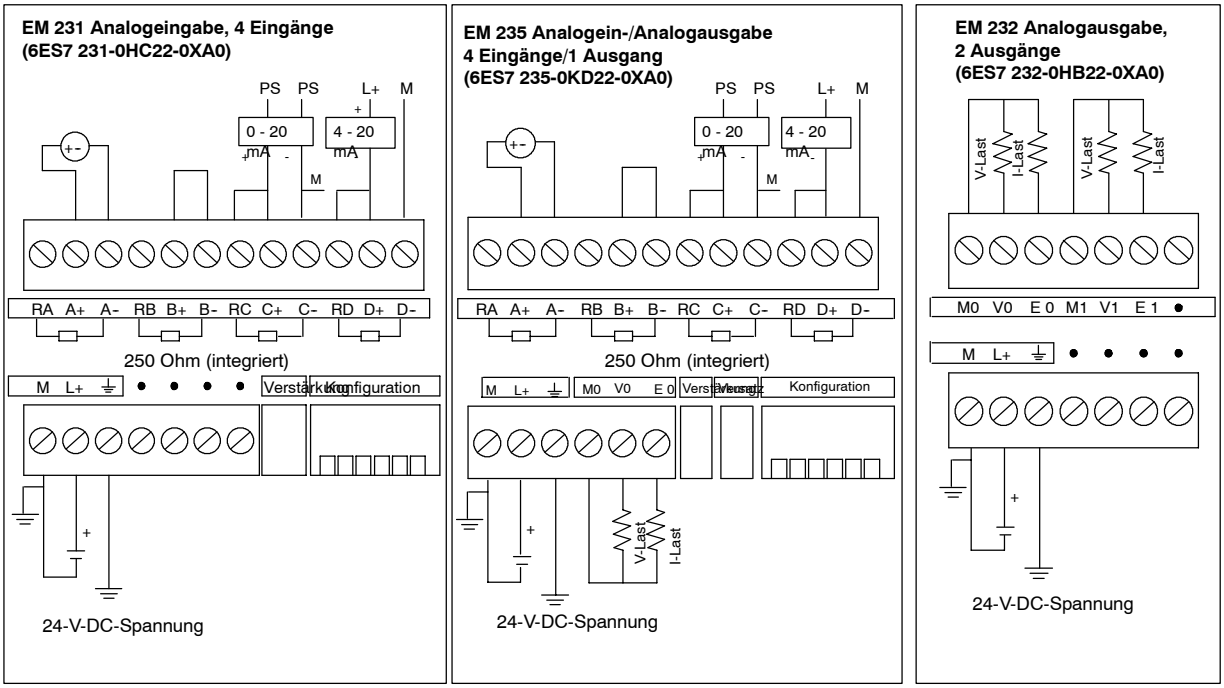


Bild A-11 Schaltpläne für Analogerweiterungsmodule

Analoge LED-Anzeigen

Die LED-Anzeigen der Analogmodule werden in Tabelle A-17 aufgeführt.

Tabelle A-17 Analoge LED-Anzeigen

LED-Anzeige	EIN	AUS
24-V-DC-Spannungsversorgung vorhanden	Fehlerfrei	Keine 24-V-DC-Spannung

Tipp

Der Zustand der Anwenderspannung wird auch in Sondermerkern (SM) gemeldet. Ausführliche Informationen finden Sie in Anhang D, SMB8 bis SMB21 Kenn- und Fehlerregister E/A-Modul.

Kalibrierung der Eingänge

Die Kalibrierung wirkt sich auf die Verstärkung der Instrumente aus, die auf den Analog-Multiplexer folgen (siehe Schaltbild der Eingänge für das EM 231 in Bild A-14 und für das EM 235 in Bild A-15). Deshalb wirkt sich die Kalibrierung auf alle Eingangskanäle aus. Abweichungen in den Komponentenwerten der einzelnen Eingangskreise, die dem Analog-Multiplexer vorgeschaltet sind, verursachen auch nach der Kalibrierung geringfügige Unterschiede in den gelesenen Werten zwischen den Kanälen, die an das gleiche Eingangssignal angeschlossen sind.

Um die technischen Daten einzuhalten, sollten Sie die Analogeingabefilter für alle Eingänge des Moduls aktivieren. 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.
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.

Kalibrierung und Konfiguration beim EM 231 und EM 235

Das Kalibrierpotentiometer und die DIP-Schalter für die Konfiguration befinden sich auf der rechten Seite des unteren Klemmenblocks des Moduls (siehe Bild A-12).

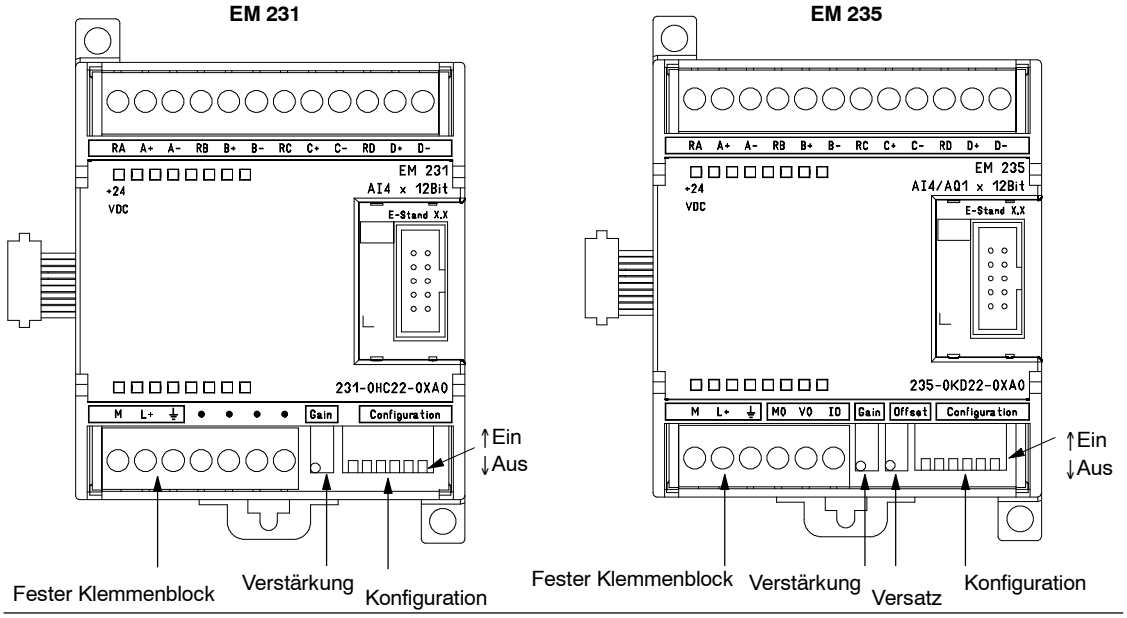


Bild A-12 Kalibrierungspotentiometer und DIP-Konfigurationsschalter beim EM 231 und EM 235

Konfiguration des EM 231

Tabelle A-18 zeigt, wie Sie das Modul EM 231 mit den DIP-Schaltern konfigurieren. Die Schalter 1, 2 und 3 wählen den Bereich der Analogeingänge aus. Alle Eingänge werden auf den gleichen Bereich der Analogeingänge gesetzt. In dieser Tabelle bedeutet EIN geschlossen und AUS geöffnet. Die Schalterstellungen sind schreibgeschützt, wenn die Spannung eingeschaltet ist.

Tabelle A-18 Konfigurationsschalter des EM 231 zum Einstellen des Analogeingabebereichs

Einpolig			Eingabe Vollausschlag	Auflösung
Schalter 1	**Schalter 2**	**Schalter 3**		
EIN	AUS	EIN	0 bis 10 V	2,5 mV
	EIN	AUS	0 bis 5 V	1,25 mV
			0 bis 20 mA	5 µA
Zweipolig			**Eingabe Vollausschlag**	**Auflösung**
Schalter 1	**Schalter 2**	**Schalter 3**		
AUS	AUS	EIN	± 5 V	2,5 mV
	EIN	AUS	±2,5 V	1,25 mV

A

Konfiguration des EM 235

Tabelle A-19 zeigt, wie Sie das Modul EM 235 mit den DIP-Schaltern konfigurieren. Die Schalter 1 bis 6 stellen den Analogeingabebereich und die Auflösung ein. Alle Eingänge werden auf den gleichen Bereich der Analogeingänge und auf das gleiche Datenformat gesetzt. Tabelle A-20 zeigt, wie Sie die Einstellungen für Einpolig/Zweipolig (Schalter 6), Verstärkung (Schalter 4 und 5) und Dämpfung (Schalter 1, 2 und 3) vornehmen. In diesen Tabellen bedeutet EIN geschlossen und AUS geöffnet. Die Schalterstellungen sind schreibgeschützt, wenn die Spannung eingeschaltet ist.

Tabelle A-19 Konfigurationsschalter des EM 235 zum Einstellen des Analogbereichs und der Auflösung

Einpolig						Eingabe Vollausschlag	Auflösung
Schalter 1	**Schalter 2**	**Schalter 3**	**Schalter 4**	**Schalter 5**	**Schalter 6**		
EIN	AUS	AUS	EIN	AUS	EIN	0 bis 50 mV	12,5 µV
AUS	EIN	AUS	EIN	AUS	EIN	0 bis 100 mV	25 µV
EIN	AUS	AUS	AUS	EIN	EIN	0 bis 500 mV	125 µV
AUS	EIN	AUS	AUS	EIN	EIN	0 bis 1 V	250 µV
EIN	AUS	AUS	AUS	AUS	EIN	0 bis 5 V	1,25 mV
EIN	AUS	AUS	AUS	AUS	EIN	0 bis 20 mA	5 µA
AUS	EIN	AUS	AUS	AUS	EIN	0 bis 10 V	2,5 mV
Zweipolig						**Eingabe Vollausschlag**	**Auflösung**
Schalter 1	**Schalter 2**	**Schalter 3**	**Schalter 4**	**Schalter 5**	**Schalter 6**		
EIN	AUS	AUS	EIN	AUS	AUS	±25 mV	12,5 µV
AUS	EIN	AUS	EIN	AUS	AUS	±50 mV	25 µV
AUS	AUS	EIN	EIN	AUS	AUS	±100 mV	50 µV
EIN	AUS	AUS	AUS	EIN	AUS	±250 mV	125 µV
AUS	EIN	AUS	AUS	EIN	AUS	±500 mV	250 µV
AUS	AUS	EIN	AUS	EIN	AUS	±1 V	500 µV
EIN	AUS	AUS	AUS	AUS	AUS	±2,5 V	1,25 mV
AUS	EIN	AUS	AUS	AUS	AUS	±5 V	2,5 mV
AUS	AUS	EIN	AUS	AUS	AUS	±10 V	5 mV

Tabelle A-20 Konfigurationsschalter des EM 235 zum Einstellen von Einpolig/Zweipolig, Verstärkung, Dämpfung

Konfigurationsschalter des EM 235						Einpolig/Zweipolig	Verstärkung	Dämpfung
Schalter 1	**Schalter 2**	**Schalter 3**	**Schalter 4**	**Schalter 5**	**Schalter 6**			
					EIN	Einpolig		
					AUS	Zweipolig		
			AUS	AUS			x1	
			AUS	EIN			x10	
			EIN	AUS			x100	
			EIN	EIN			Ungültig	
EIN	AUS	AUS						0,8
AUS	EIN	AUS						0,4
AUS	AUS	EIN						0,2

Datenwortformat der Eingänge beim EM 231 und EM 235

Bild A-13 zeigt die Anordnung des 12-Bit-Datenwerts im Analogeingangswort der CPU.

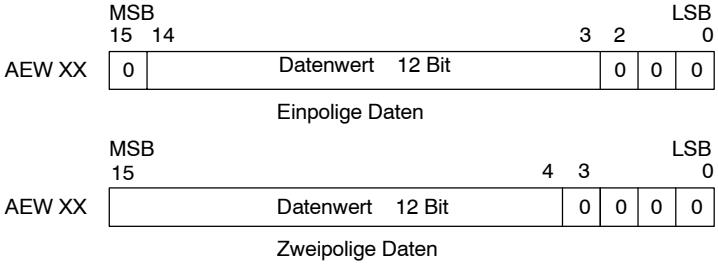


Bild A-13 Datenwortformat der Eingänge beim EM 231 und EM 235

Tipp

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 beim EM 231 und EM 235

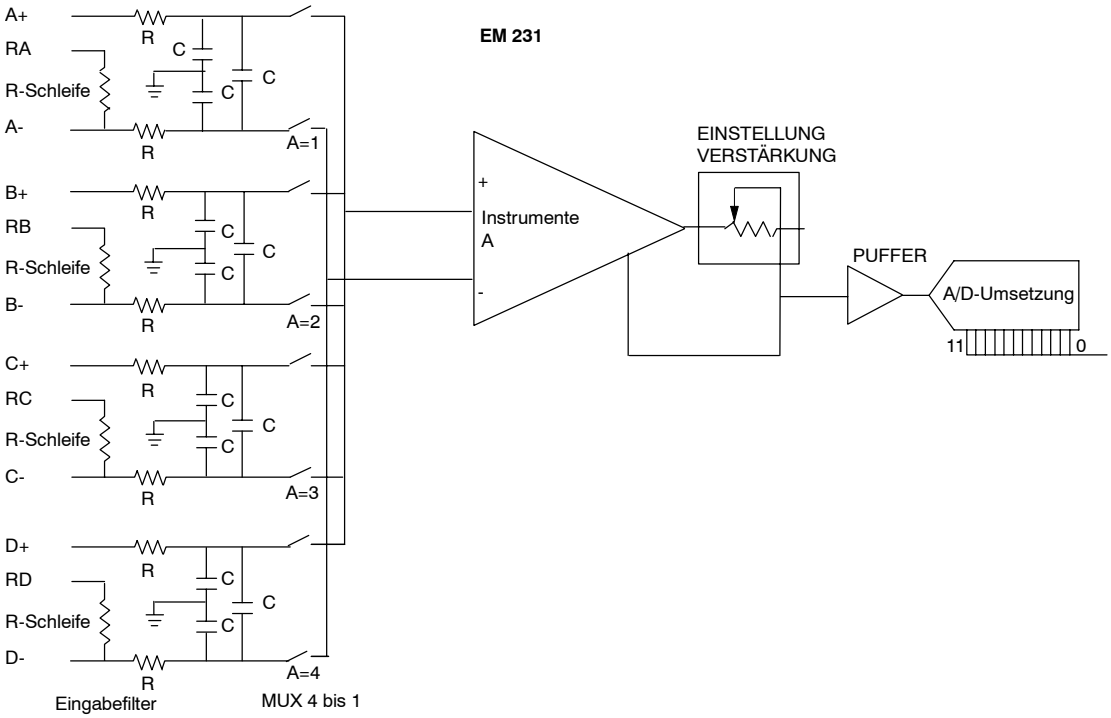


Bild A-14 Schaltbild der Eingänge beim EM 231

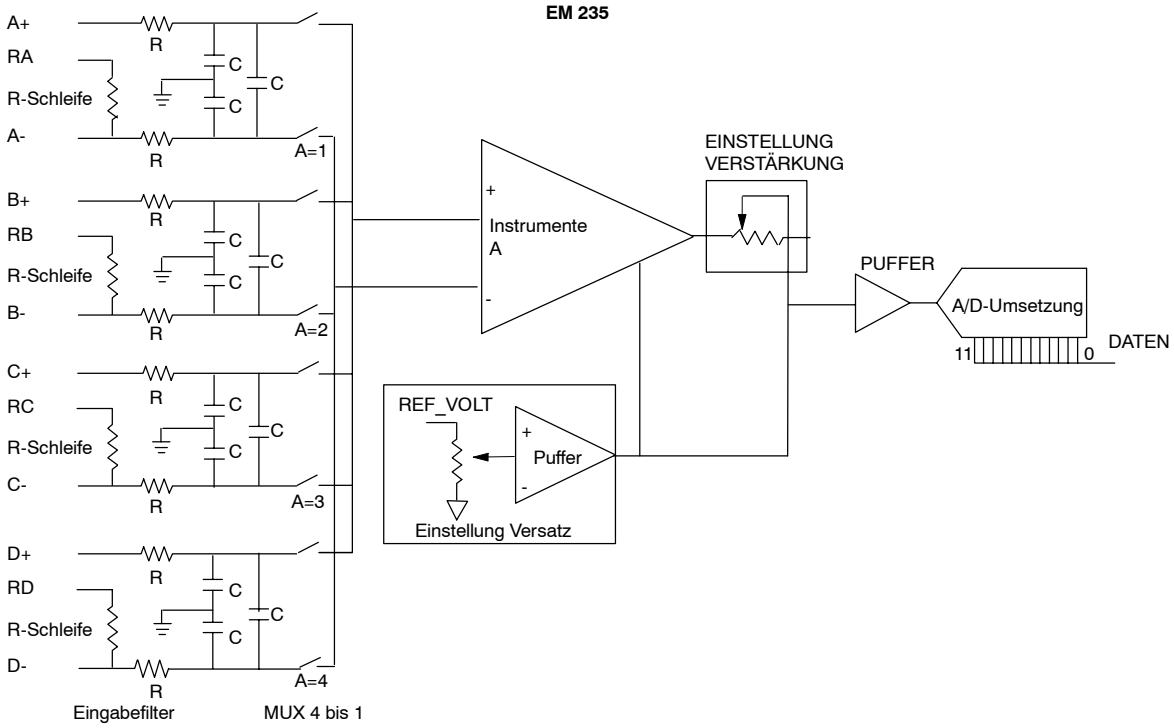


Bild A-15 Schaltbild der Eingänge beim EM 235

Datenwortformat der Ausgänge beim EM 232 und EM 235

Bild A-16 zeigt die Anordnung des 12-Bit-Datenwerts im Analogausgangswort der CPU.

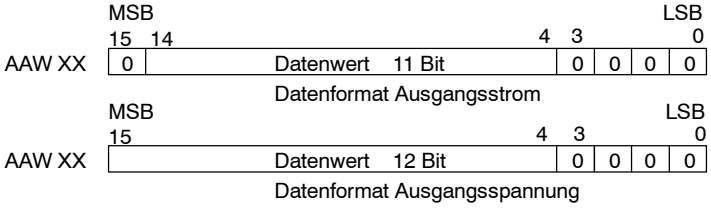


Bild A-16 Datenwortformat der Ausgänge beim EM 232 und EM 235

Tipp

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 Bits wirken sich nicht auf den Wert des Ausgangssignals aus.

Schaltbild der Ausgänge beim 232 und EM 235

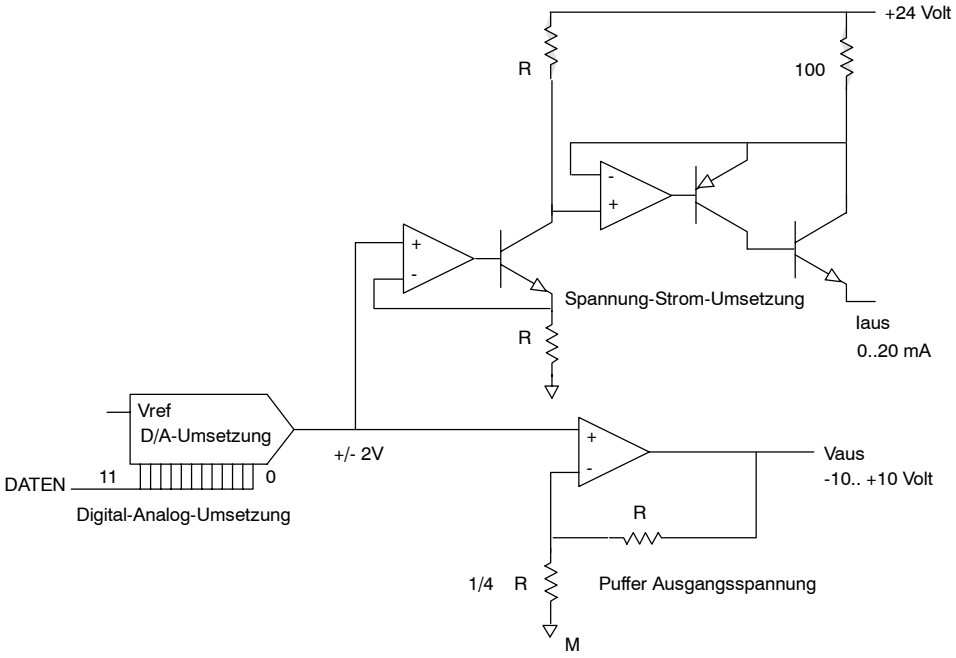


Bild A-17 Schaltbild der Ausgänge beim EM 232 und EM 235

Richtlinien für den Einbau

Beachten Sie die folgenden Richtlinien, damit Genauigkeit und Wiederholbarkeit sichergestellt sind:

- ❑ Achten Sie darauf, dass die 24-V-DC-Geberversorgung störfest ist.
- ❑ 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-11).
- ❑ 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.
- ❑ Stellen Sie sicher, dass die Eingangssignale den Angaben für die Gleichtaktspannung entsprechen, indem Sie die Eingangssignale elektrisch trennen oder als Bezug den externen 24-V-Leiter des Analogmoduls angeben.

Tipp

Die Erweiterungsmodule EM 231 und EM 235 werden nicht für die Verwendung mit Thermoelementen empfohlen.

A

Arbeiten mit dem Analogeingabemodul: Genauigkeit und Wiederholbarkeit

Die Analogeingabemodule EM 231 und EM 235 sind preiswerte Hochgeschwindigkeits-Analogeingabe- / Analogausgabemodule (12 Bit). Die Module können ein Analogsignal innerhalb von 149 µs in den entsprechenden Digitalwert umwandeln. Das Analogsignal wird dann jedesmal zur Verfügung gestellt, wenn Ihr Programm auf den Analogeingang zugreift. Diese Umwandlungszeiten müssen zu der grundlegenden Ausführungszeit der Operation, die auf den Analogeingang zugreift, addiert werden.

Die Erweiterungsmodule EM 231 und EM 235 liefern einen nicht bearbeiteten Digitalwert (keine Linearisierung oder Filterung), der der Analogspannung bzw. dem Analogstrom an den Eingangsklemmen des Moduls entspricht. Da es sich bei den Modulen um Hochgeschwindigkeitsmodule handelt, können sie schnellen Änderungen des Analogeingabesignals 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 des Eingabesignals feststellen.

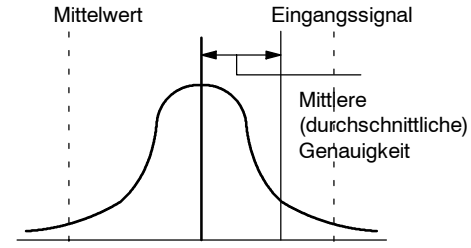


Bild A-18 Definition der Genauigkeit

Bild A-18 zeigt die Grenzwerte für die Wiederholbarkeit von 99%, den mittleren bzw. durchschnittlichen Wert aus den einzelnen Werten und die mittlere Genauigkeit in grafischer Form.

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 Wiederholbarkeit wird in diesem Bild durch die Kurve dargestellt.

Die mittlere Genauigkeit beschreibt den durchschnittlichen Wert des Fehlers (die Differenz zwischen dem Durchschnittswert der einzelnen Werte und dem genauen Wert des tatsächlichen Analogeingabesignals).

Tabelle A-21 enthält die Angaben zur Wiederholbarkeit und zur mittleren Genauigkeit in bezug auf die konfigurierbaren Bereiche.

Definitionen der Angaben zu Analogmodulen

- Genauigkeit: Abweichung von einem erwarteten Wert an einem bestimmten Eingang bzw. Ausgang.
- Auflösung: Auswirkungen der Änderung eines niederwertigsten Byte im Ausgang.

Tabelle A-21 Technische Daten des EM 231 und des EM 235

Eingangsbereich bei Vollausschlag	Wiederholbarkeit[1]		Mittlere (durchschnittliche) Genauigkeit[1, 2, 3, 4]	
	% des Vollausschlags	Zählimpulse	% des Vollausschlags	Zählimpulse
Technische Daten EM 231				
0 bis 5 V	± 0,075%	± 24	± 0,1%	± 32
0 bis 20 mA				
0 bis 10 V				
± 2,5 V		± 48	± 0,05%	
± 5 V				
Technische Daten EM 235				
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				

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

Technische Daten der Thermoelement- und RTD-Erweiterungsmodule

Tabelle A-22 Bestellnummern der Thermoelement- und RTD-Module

Bestellnummer	Erweiterungsmodule	EM-Eingänge	EM-Ausgänge	Steckbarer Klemmenblock
6ES7 231-7PD22-0XA0	EM 231 Analogeingabe Thermoelement, 4 Eingänge	4 Thermoelemente	-	Nein
6ES7 231-7PB22-0XA0	EM 231 Analogeingabe RTD, 2 Eingänge	2 RTD	-	Nein

Tabelle A-23 Allgemeine technische Daten der Thermoelement- und RTD-Module

Bestellnummer	Name und Beschreibung des Moduls	Abmessungen (mm) (B x H x T)	Gewicht	Verlustleistung	V-DC-Bedarf +5 V DC	V-DC-Bedarf +24 V DC
6ES7 231-7PD22-0XA0	EM 231 Analogeingabe Thermoelement, 4 Eingänge	71,2 x 80 x 62	210 g	1,8 W	87 mA	60 mA
6ES7 231-7PB22-0XA0	EM 231 Analogeingabe RTD, 2 Eingänge	71,2 x 80 x 62	210 g	1,8 W	87 mA	60 mA

Tabelle A-24 Technische Daten der Thermoelement- und RTD-Module

Allgemein	6ES7 231-7PD22-0XA0 Thermoelement	6ES7 231-7PB22-0XA0 RTD
Trennung		
Feld zu Logik	500 V AC	500 V AC
Feld zu 24 V DC	500 V AC	500 V AC
24 V DC zu Logik	500 V AC	500 V AC
Gleichtakt-Eingangsbereich (Eingangskanal zu Eingangskanal)	120 V AC	0
Gleichtaktunterdrückung	> 120 dB bei 120 V AC	> 120 dB bei 120 V AC
Eingangstyp	Erdfreies Thermoelement	Auf Modulerde bezogener Widerstandstemperaturfühler
Eingangsbereiche[1]	TC-Typen (1 je Modul) S, T, R, E, N, K, J Spannungsbereich: +/- 80 mV	RTD-Typen (1 je Modul) Pt -100 Ω, 200 Ω, 500 Ω, 1000 Ω (mit α = 3850 ppm, 3920 ppm, 3850,55 ppm, 3916 ppm, 3902 ppm) Pt -10000 Ω (α = 3850 ppm) Cu -9.035 Ω (α = 4720 ppm) Ni -10 Ω, 120 Ω, 1000 Ω (mit α = 6720 ppm, 6178 ppm) R -150 Ω, 300 Ω, 600 Ω FS
Eingangsauflösung		
Temperatur	0,1 °C / 0,1 °F	0,1 °C / 0,1 °F
Spannung	15 Bit plus Vorzeichen	-
Widerstand	-	15 Bit plus Vorzeichen
Messprinzip	Sigma-Delta	Sigma-Delta
Aktualisierungszeit Modul: Alle Kanäle	405 ms	405 ms (700 ms bei Pt10000)
Leitungslänge	Max. 100 m zum Geber	Max. 100 m zum Geber.
Schleifenwiderstand Leitung	Max. 100 Ω	20 Ω, max. 2,7 Ω für Cu
Störunterdrückung	85 dB bei 50 Hz/60 Hz/ 400 Hz	85 dB bei 50 Hz/60 Hz/ 400 Hz
Datenwortformat	Spannung: -27648 bis + 27648	Widerstand: -27648 bis +27648
Maximale Verlustleistung Geber	-	1 mW
Eingangsimpedanz	≥1 MΩ	≥ 10 MΩ
Max. Eingangsspannung	30 V DC	30 V DC (Fühler), 5 V DC (Quelle)
Eingabefilterdämpfung	-3 dB bei 21 kHz	-3 dB bei 3,6 kHz
Basisfehler	0,1% FS (Spannung)	0,1% FS (Widerstand)
Wiederholbarkeit	0,05% FS	0,05% FS
Fehler kalte Verbindungsstelle	±1,5 ° C	-
24-V-DC-Versorgungsspannungsbereich	20,4 bis 28,8 V DC	20,4 bis 28,8 V DC

[1] Der eingestellte Eingangsbereich (Temperatur, Spannung bei Widerstand) gilt für alle Kanäle des Moduls.

A

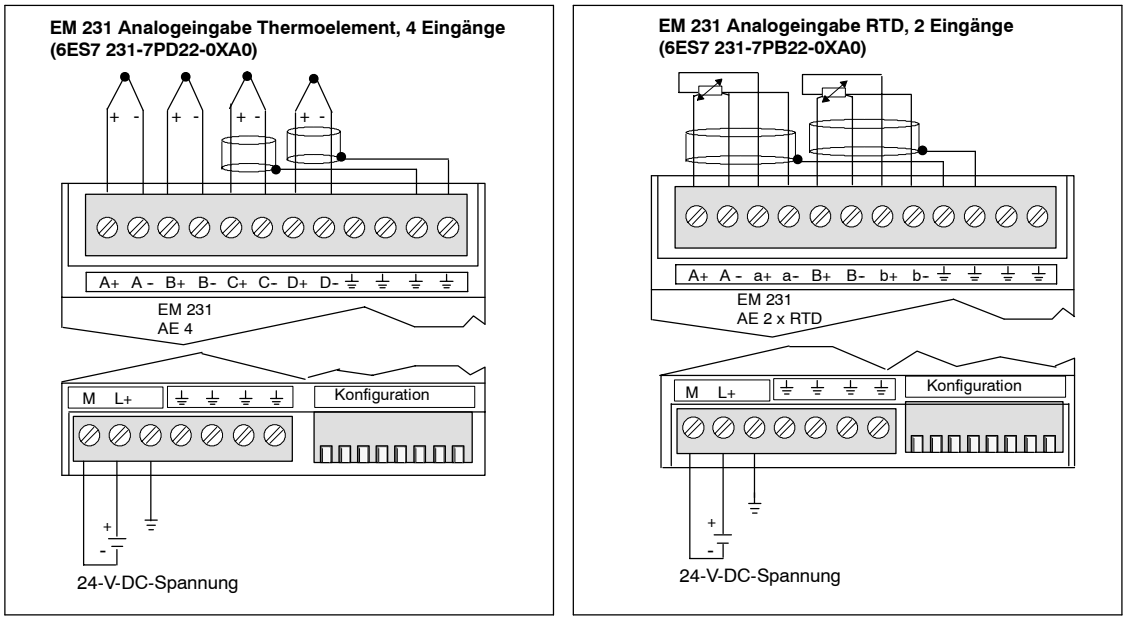


Bild A-19 Kennzeichnung der Anschlüsse beim EM 231 Thermoelement und beim EM 231 RTD

Kompatibilität

Die RTD- und Thermoelement-Module können zusammen mit der CPU 222, CPU 224, CPU 226 und der CPU 226XM eingesetzt werden.

Tipp

Das RTD- und das Thermoelement-Modul bieten maximale Leistungsfähigkeit, wenn sie in einer Umgebung mit stabiler Temperatur eingebaut werden.

Das EM 231 Thermoelement-Modul beispielsweise verfügt über eine Kompensationsschaltung für die kalte Verbindungsstelle, die die Temperatur an den Anschlüssen des Moduls misst und die erforderlichen Änderungen an der Messung vornimmt, um die Temperaturunterschiede zwischen der Bezugstemperatur und der Temperatur am Modul auszugleichen. Ändert sich die Umgebungstemperatur in dem Bereich, in dem das EM 231 Thermoelement-Modul installiert ist, schnell, werden weitere Fehler eingeführt.

Um höchste Genauigkeit und Wiederholbarkeit zu erreichen, empfiehlt Siemens, die S7-200 RTD- und Thermoelement-Module an Standorten zu montieren, an denen die Umgebungstemperatur stabil ist.

A

Störfestigkeit

Setzen Sie geschirmte Leitungen für optimale Störfestigkeit ein. Wird ein Eingangskanal eines Thermoelements nicht verwendet, müssen Sie den nicht verwendeten Eingangskanal kurzschließen oder zu einem anderen Kanal parallel schalten.

EM 231 Thermoelement-Modul

Das EM 231 Thermoelement-Modul bietet der Produktreihe S7-200 eine komfortable, elektrisch getrennte Schnittstelle zu sieben Typen von Thermoelementen: J, K, E, N, S, T und R. Außerdem ermöglicht es der S7-200 den Anschluss an Analogsignale mit Pegel Low im Bereich ±80 mV. Alle an das Modul angeschlossenen Thermoelemente müssen vom gleichen Typ sein.

Hintergrundinformationen zu Thermoelementen

Thermoelemente entstehen, wenn zwei unterschiedliche Metalle elektrisch miteinander verbunden werden. Dadurch wird eine Spannung erzeugt, die proportional zu der Temperatur der Verbindungsstelle ist. Es handelt sich um eine geringe Spannung. Ein Mikrovolt kann viele Grade darstellen. Grundlage für die Temperaturmessung mit Thermoelementen sind das Messen der Spannung eines Thermoelements, das Kompensieren von zusätzlichen Verbindungsstellen und das Linearisieren der Ergebnisse.

Wenn Sie ein Thermoelement an das EM 231 Thermoelement-Modul anschließen, werden die beiden Leitungen der unterschiedlichen Metalle am Signalanschluss des Moduls angeschlossen. Die Stelle, an der die beiden unterschiedlichen Leitungen miteinander verbunden werden, bildet den Sensor des Thermoelements.

Zwei weitere Thermoelemente entstehen an der Stelle, an der die unterschiedlichen Leitungen an den Signalanschluss angeschlossen werden. Die Temperatur des Anschlusses erzeugt eine Spannung, die zu der Spannung des Thermoelement-Sensors addiert wird. Wird diese Spannung nicht ausgeglichen, weicht die ausgegebene Temperatur von der Temperatur des Sensors ab.

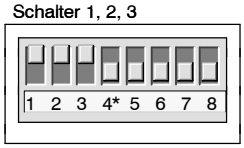
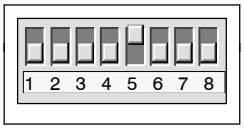
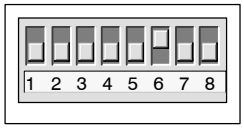
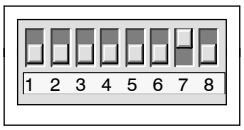
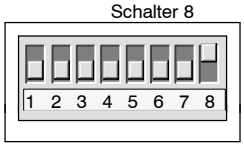
Durch die Kompensation der kalten Verbindungsstelle wird das Thermoelement am Anschluss ausgeglichen. Tabellen für Thermoelemente basieren auf einer Bezugstemperatur an der Verbindungsstelle, üblicherweise Null Grad Celsius. Durch die Kompensation der kalten Verbindungsstelle wird der Anschluss auf Null Grad Celsius kompensiert. Die Spannung, die durch das Thermoelement des Anschlusses addiert wird, wird durch die Kompensation der kalten Verbindungsstelle korrigiert. Die Temperatur des Moduls wird innen gemessen und dann in einen Wert umgewandelt, der zur Sensor-Umsetzung addiert wird. Die korrigierte Sensor-Umsetzung wird dann mittels der Thermoelement-Tabellen linearisiert.

Einrichten des EM 231 Thermoelement-Moduls

Mit den DIP-Schaltern für die Konfiguration, die sich am Boden des Moduls befinden, können Sie den Typ des Thermoelements, die Prüfung auf offene Leitungen, die Einheit der Temperatur und die Kompensation der kalten Verbindungsstelle einstellen. Damit die Einstellung der DIP-Schalter wirksam wird, müssen Sie das Zielsystem und/oder die 24-V-Versorgung aus- und wieder einschalten.

DIP-Schalter 4 ist für zukünftige Verwendung reserviert. Stellen Sie DIP-Schalter 4 auf 0 (unten/aus) ein. Weitere DIP-Schalterstellungen finden Sie in Tabelle A-25.

Tabelle A-25 Einstellen der DIP-Schalter des Thermoelement-Moduls

Schalter 1, 2, 3	Thermoelement-Typ	Einstellung	Beschreibung
Schalter 1, 2, 3  Konfiguration ↑1 - ein ↓0 - aus * Stellen Sie DIP-Schalter 4 auf 0 (unten/aus) ein.	J (Voreinstellung)	000	Schalter 1 bis 3 stellen den Thermoelement-Typ (oder mV-Betrieb) für alle Kanäle des Moduls ein. Beispiel: Zum Einstellen des Thermoelement-Typs E stellen Sie Schalter 1 = 0, Schalter 2 = 1, Schalter 3 = 1 ein.
	K	001	
	T	010	
	E	011	
	R	100	
	S	101	
	N	110	
	+/- 80 mV	111	
Schalter 5	**Prüfung auf offene Leitungen - Richtung**	**Einstellung**	**Beschreibung**
Schalter 5  Konfiguration ↑1 - ein ↓0 - aus	Aufwärts (+3276,7 Grad)	0	0 Prüfung auf offene Leitungen positiv 1 Prüfung auf offene Leitungen negativ
	Abwärts (-3276,8 Grad)	1	
Schalter 6	**Prüfung auf offene Leitungen - Aktivierung**	**Einstellung**	**Beschreibung**
Schalter 6  Konfiguration ↑1 - ein ↓0 - aus	Aktiviert	0	Die Prüfung auf offene Leitungen wird mittels eines Stroms von 25 µA an den Eingangsklemmen durchgeführt. Der Schalter zum Aktivieren der Prüfung aktiviert oder deaktiviert die Stromquelle. Die Bereichsprüfung für offene Leitungen wird immer ausgeführt, auch wenn die Stromquelle deaktiviert ist. Das EM 231 Thermoelement-Modul erkennt offene Leitungen, wenn das Eingangssignal ca. ±200 mV überschreitet. Wird eine offene Leitung erkannt, wird der ausgelesene Wert auf den Wert gesetzt, der von der Prüfung auf offene Leitungen eingestellt ist.
	Deaktiviert	1	
Schalter 7	**Einheit der Temperatur**	**Einstellung**	**Beschreibung**
Schalter 7  Konfiguration ↑1 - ein ↓0 - aus	Celsius (°C)	0	Das EM 231 Thermoelement-Modul kann Temperaturen in Celsius oder Fahrenheit angeben. Die Umrechnung von Celsius in Fahrenheit wird im Modul vorgenommen.
	Fahrenheit (°F)	1	
Schalter 8	**Kalte Verbindungsstelle**	**Einstellung**	**Beschreibung**
Schalter 8  Konfiguration ↑1 - ein ↓0 - aus	Kompensation der kalten Verbindungsstelle aktiviert	0	Die Kompensation der kalten Verbindungsstelle muss aktiviert sein, wenn Sie mit Thermoelementen arbeiten. Ist die Kompensation der kalten Verbindungsstelle nicht aktiviert, sind die Umwandlungen des Moduls aufgrund der Spannung, die beim Anschließen der Leitung des Thermoelements an den Anschluss des Moduls erzeugt wird, fehlerhaft. Die Kompensation der kalten Verbindungsstelle wird automatisch deaktiviert, wenn Sie einen Bereich von ±80 mV einstellen.
	Kompensation der kalten Verbindungsstelle deaktiviert	1	

A

Tipp

- Die Stromquelle für die Leitungsprüfung kann Signale aus Quellen mit Pegel Low z.B. Thermoelement-Simulatoren stören.
- Eingangsspannungen über ca. ±200 mV lösen die Prüfung auf offene Leitungen aus, auch wenn die Stromquelle für die Leitungsprüfung deaktiviert ist.

Tipp

- Der Modulfehler kann die Angaben überschreiten, wenn sich die Umgebungstemperatur ändert.
- Wenn der angegebene Bereich der Umgebungstemperatur des Moduls überschritten wird, kann die Kompensation der kalten Verbindungsstelle des Moduls fehlerhaft sein.

Einsetzen des Thermoelements: Statusanzeigen

Das EM 231 Thermoelement-Modul liefert dem Automatisierungssystem Datenwörter, die Temperaturen oder Fehlerbedingungen anzeigen. Statusbits zeigen Bereichsfehler und Anwenderspannungs-/Modulausfälle an. LEDs zeigen den Status des Moduls an. Ihr Programm muss Logik umfassen, die Fehlerbedingungen erkennt und entsprechend der Anwendung reagiert. Tabelle A-26 zeigt die Statusanzeigen des EM 231 Thermoelements.

Tabelle A-26 Statusanzeigen des EM 231 Thermoelements

Fehlerbedingung	Kanaldaten	SF LED Rot	24 V LED Grün	Statusbit Bereich[1]	Fehlerhafte 24-V-DC-Spannung[2]
Keine Fehler	Umwandlungsdaten	AUS	EIN	0	0
24 V fehlt	32766	AUS	AUS	0	1
Leitungsprüfung und Stromquelle aktiviert	-32768/32767	BLINKT	EIN	1	0
Bereichsüberschreitung	-32768/32767	BLINKT	EIN	1	0
Diagnosefehler[3]	0000	EIN	AUS	0	Hinweis [3]

1 Das Bereichs-Statusbit ist Bit 3 im Byte des Fehlerregisters des Moduls (SMB9 für Modul 1, SMB11 für Modul 2 usw.)
2 Das Statusbit Fehlerhafte Spannung ist Bit 2 im Byte des Fehlerregisters des Moduls (SMB 9, SMB 11 usw., siehe Anhang D.)
3 Diagnosefehler bewirken einen Modul-Konfigurationsfehler. Das Statusbit Fehlerhafte Spannung kann oder kann nicht vor dem Modul-Konfigurationsfehler gesetzt werden.

Tipp

Das Format der Kanaldaten ist ein Zweierkomplement, 16-Bit-Wörter. Die Temperatur wird in Einheiten von 0,1 Grad dargestellt. Beispiel: Beträgt die gemessene Temperatur 100,2 Grad, wird der Wert 1002 ausgegeben. Spannungsdaten werden auf 27648 skaliert. Beispiel: -60,0 mV wird ausgegeben als -20736 (= -60 mV/80 mV * 27648).

Alle vier Kanäle werden alle 405 ms aktualisiert, wenn das Automatisierungssystem die Daten gelesen hat. Liest das Automatisierungssystem die Daten nicht innerhalb einer Aktualisierungszeit, gibt das Modul alte Daten aus, bis das Modul erneut nach dem Lesen des Automatisierungssystems aktualisiert wird. Damit die Kanaldaten immer aktuell bleiben, ist es empfehlenswert, dass das Programm des Automatisierungssystems die Daten mindestens genauso häufig liest, wie das Modul aktualisiert wird.

Tipp

Wenn Sie mit dem EM 231 Thermoelement-Modul arbeiten, müssen Sie die Analogeingabefilter im Automatisierungssystem ausschalten. Durch die Analogfilter kann es sein, dass Fehlerbedingungen nicht rechtzeitig erkannt werden.

Tabelle A-27 Temperaturbereiche (° C) und Genauigkeit für Arten von Thermoelementen

Datenwort (1 Ziffer = 0,1 °C)		Typ J	Typ K	Typ T	Typ E	Typen R, S	Typ N	±80 mV	
Dez.	Hex.								
32767	7FFF	>1200,0 °C	>1372,0 °C	>400,0 °C	>1000,0 °C	>1768,0 °C	>1300,0 °C	>94,071 mV	OF
↑	↑							↑	↑
32511	7EFF							94,071 mV	
:	:								ODER
27649	6C01							80,0029 mV	
27648	6C00					↑		80 mV	
:	:								
17680	4510		↑			1768,0 °C			
:	:								NR
13720	3598		1372,0 °C				↑		
:	:		oberhalb des Bereichs						
13000	32C8	↑	1300,0 °C				1300,0 °C		
:	:								
12000	2EE0	1200,0 °C			↑				
:	:								
10000	2710			↑	1000,0 °C				
:	:								
4000	0FA0			400,0 °C		400,0 °C			
:	:								
1	0001	0,1 °C	0,1 °C	0,1 °C	0,1° C	0,1 °C	0,1 °C	0,0029 mV	
0	0000	0,0 °C	0,0 °C	0,0 °C	0,0° C	0,0 °C	0,0 °C	0,0 mV	
-1	FFFF	-0,1 °C	-0,1 °C	-0,1 °C	-0,1° C	-0,1 °C	-0,1 °C	-0,0029 mV	
:	:					unterhalb des Bereichs			
-500	FE0C					-50,0 °C			
-1500	FA24	-150,0 °C				↓			
:	:								
-2000	F830	unterhalb des Bereichs	-200,0 °C						
:	:								
-2100	F7CC	-210,0 °C							
:	:		unterhalb des Bereichs						
-2550	F60A			-255,0 °C	-255,0° C				
:	:			unterhalb des Bereichs	unterhalb des Bereichs				
-2700	F574	↓	-270,0 °C	-270,0 °C	-270,0° C		-270,0 °C		
:	:								
-27648	9400		↓	↓	↓		↓	-80 mV	
-27649	93FF							-80,0029 mV	
:	:								
-32512	8100							-94,071 mV	UR
↓	↓							↓	↓
-32768	8000	<-210,0 °C	<-270,0 °C	<-270,0 °C	<-270,0 °C	<-50,0 °C	<-270,0 °C	<-94,071 mV	UF
Genauigkeit über Vollbereich		±0,1%	±0,3%	±0,6%	±0,1%	±0,6%	±0,1%	±0,1%	
Genauigkeit (Nennbereich ohne kalte Verbindungsstelle)		±1,5 °C	±1,7 °C	±1,4 °C	±1,3 °C	±3,7 °C	±1,6 °C	±0,10%	
Fehler kalte Verbindungsstelle		±1,5 °C	±1,5 °C	±1,5 °C	±1,5 °C	±1,5 °C	±1,5 °C	-/-	

*OF = Überlauf; OR = Oberhalb des Bereichs; NR = Nennbereich; UR = Unterhalb des Bereichs; UF = Unterschreitung

↑ bedeutet, dass alle größeren Analogwerte und alle Analogwerte, die unterhalb des Grenzwerts für die Leitungsprüfung liegen, den Datenwert für Überlauf ausgeben, 32767 (0x7FFF).
↓ bedeutet, dass alle kleineren Analogwerte und alle Analogwerte, die oberhalb des Grenzwerts für die Leitungsprüfung liegen, den Datenwert für Unterschreitung ausgeben, -32768 (0x8000).

A

Tabelle A-28 Temperaturbereiche (°F) von Thermoelement-Typen

Datenwort (1 Ziffer = 0,1°F)		Typ J	Typ K	Typ T	Typ E	Typen R, S	Typ N	±80 mV	
Dez.	Hex.								
32767	7FFF	>2192,0 °F	>2502,0 °F	>752,0 °F	>1832,0 °F	>3214,0 °F	>2372,0 °F	>94,071 mV	OF
↑	↑	↑	↑	↑	↑	↑	↑	↑	↑
32511	7EFF							94,071 mV	ODER
32140	7D90					3214,0 °F			
27649	6C01							80,0029 mV	
27648	6C00		↑			2764,8 °F		80 mV	NR
:	:								
25020	61B8		2502,0 °F				↑		
:	:		oberhalb des Bereichs						
23720	5CA8	↑	2372,0 °F				2372,0 °F		
:	:								
21920	55A0	2192,0 °F			↑				
:	:								
18320	4790			↑	1832,0 °F				
:	:								
7520	1D60			752,0 °F		752,0 °F			
:	:								
320	0140					unterhalb des Bereichs	32,0 °F		
:	:								
1	0001	0,1 °F	0,1 °F	0,1 °F	0,1 °F	0,1 °F	0,1 °F	0,0029 mV	
0	0000	0,0 °F	0,0 °F	0,0 °F	0,0 °F	0,0 °F	0,0 °F	0,0 mV	
-1	FFFF	-0,1 °F	-0,1 °F	-0,1 °F	-0,1 °F	-0,1 °F	-0,1 °F	-0,0029 mV	
:	:								
-580	FDBC					-58,0 °F			
:	:								
-2380	F6B4	-238,0 °F							
:	:								
-3280	F330	unterhalb des Bereichs	-328,0 °F				unterhalb des Bereichs		
:	:								
-3460	F27C	-346,0 °F				↓			
:	:		unterhalb des Bereichs						
-4270	EF52			-427,0 °F	-427,0 °F				
:	:			unterhalb des Bereichs	unterhalb des Bereichs				
-4540	EE44	↓	-454,0 °F	-454,0 °F	-454,0 °F		-454,0 °F		
:	:								
-27648	9400		↓	↓	↓		↓	-80 mV	
-27649	93FF							-80,0029 mV	ODER
:	:								
-32512	8100							-94,071 mV	
↓	↓							↓	↓
-3268	8000	<-346,0 ° F	<-454,0 ° F	<-454,0 ° F	<-454,0 ° F	<-58,0 ° F	<-454,0 ° F	<-94,07 mV	UF

*OF = Überlauf; OR = Oberhalb des Bereichs; NR = Normalbereich; UR = Unterhalb des Bereichs; UF = Unterschreitung
↑ bedeutet, dass alle größeren Analogwerte und alle Analogwerte, die unterhalb des Grenzwerts für die Leitungsprüfung liegen, den Datenwert für Überlauf ausgeben, 32767 (0x7FFF)
↓ bedeutet, dass alle kleineren Analogwerte und alle Analogwerte, die oberhalb des Grenzwerts für die Leitungsprüfung liegen, den Datenwert für Unterschreitung ausgeben, -32768 (0x8000).

A

EM 231 RTD-Modul

Das EM 231 RTD-Modul bietet der Produktreihe S7-200 eine komfortable, elektrisch getrennte Schnittstelle zu verschiedenen Widerstandstemperaturfühlern. Außerdem ermöglicht es der Produktreihe S7-200, drei verschiedene Widerstandsbereiche zu messen. Beide an das Modul angeschlossenen Widerstandstemperaturfühler müssen vom gleichen Typ sein.

Einrichten des EM 231 RTD-Moduls

Mit den DIP-Schaltern stellen Sie RTD-Typ, Verdrahtungskonfiguration, Einheit der Temperatur und Burnout-Richtung ein. Die DIP-Schalter befinden sich auf der Unterseite des Moduls (siehe Bild A-20). Damit die Einstellung der DIP-Schalter wirksam wird, müssen Sie das Zielsystem und/oder die 24-V-Versorgung aus- und wieder einschalten.

Sie stellen den RTD-Typ mit den DIP-Schaltern 1, 2, 3, 4 und 5 ein (siehe Tabelle A-29). Weitere DIP-Schalterstellungen finden Sie in Tabelle A-30.

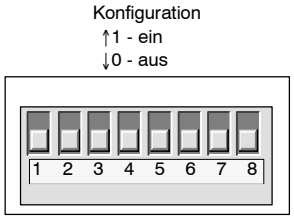


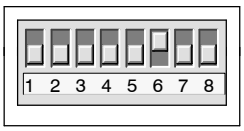
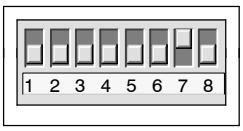
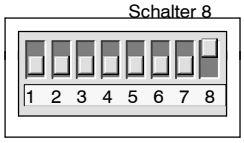
Bild A-20 DIP-Schalter des EM 231 RTD-Modul

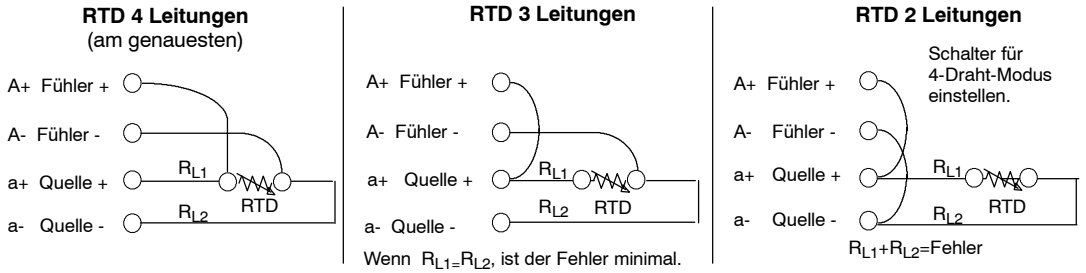
Tabelle A-29 Einstellen des RTD-Typs: DIP-Schalter 1 bis 5

RTD-Typ und Alpha	Schalter 1	Schalter 2	Schalter 3	Schalter 4	Schalter 5	RTD-Typ und Alpha	Schalter 1	Schalter 2	Schalter 3	Schalter 4	Schalter 5
100 Ω Pt 0,003850 (Voreinstellung)	0	0	0	0	0	100 Ω Pt 0,00302	1	0	0	0	0
200 Ω Pt 0,003850	0	0	0	0	1	200 Ω Pt 0,003902	1	0	0	0	1
500 Ω Pt 0,003850	0	0	0	1	0	500 Ω Pt 0,003902	1	0	0	1	0
1000 Ω Pt 0,003850	0	0	0	1	1	1000 Ω Pt 0,003902	1	0	0	1	1
100 Ω Pt 0,003920	0	0	1	0	0	ERSATZ	1	0	1	0	0
200 Ω Pt 0,003920	0	0	1	0	1	100 Ω Ni 00,00672	1	0	1	0	1
500 Ω Pt 0,003920	0	0	1	1	0	120 Ω Ni 0,00672	1	0	1	1	0
1000 Ω Pt 0,003920	0	0	1	1	1	1000 Ω Ni 0,00672	1	0	1	1	1
100 Ω Pt 0,00385055	0	1	0	0	0	100 Ω Ni 00,006178	1	1	0	0	0
200 Ω Pt 0,00385055	0	1	0	0	1	120 Ω Ni 0,006178	1	1	0	0	1
500 Ω Pt 0,00385055	0	1	0	1	0	1000 Ω Ni 0,006178	1	1	0	1	0
1000 Ω Pt 0,00385055	0	1	0	1	1	10000 Ω Pt 0,003850	1	1	0	1	1
100 Ω Pt 0,003916	0	1	1	0	0	10 Ω Cu 0,004270	1	1	1	0	0
200 Ω Pt 0,003916	0	1	1	0	1	150 Ω FS Widerstand	1	1	1	0	1
500 Ω Pt 0,003916	0	1	1	1	0	300 Ω FS Widerstand	1	1	1	1	0
1000 Ω Pt 0,003916	0	1	1	1	1	600 ΩFS Widerstand	1	1	1	1	1

A

Tabelle A-30 Einstellen der RTD DIP-Schalter

Schalter 6	**Prüfung auf offene Leitungen**	**Einstel-lung**	**Beschreibung**
Schalter 6  Konfiguration ↑1 - ein ↓0 - aus	Aufwärts (+3276,7 Grad)	0	Prüfung auf offene Leitungen positiv
	Abwärts (-3276,8 Grad)	1	Prüfung auf offene Leitungen negativ
Schalter 7	**Einheit der Temperatur**	**Einstel-lung**	**Beschreibung**
Schalter 7  Konfiguration ↑1 - ein ↓0 - aus	Celsius (°C)	0	Das RTD-Modul kann Temperaturen in Celsius oder Fahrenheit angeben. Die Umrechnung von Celsius in Fahrenheit wird im Modul vorgenommen.
	Fahrenheit (°F)	1	
Schalter 8	**Verdrahtungsschema**	**Einstel-lung**	**Beschreibung**
Schalter 8  Konfiguration ↑1 - ein ↓0 - aus	3 Leitungen	0	Sie können das RTD-Modul auf drei Arten an den Geber anschließen (siehe Bild). Am genauesten sind 4 Leitungen. Am ungenauesten sind 2 Leitungen. Diese Verdrahtung wird nur empfohlen, wenn Fehler aufgrund der Verdrahtung in Ihrer Anwendung ignoriert werden können.
	2 Leitungen oder 4 Leitungen	1	



Hinweis: R_{L1} = Leitungswiderstand von Klemme a+ zum RTD
R_{L2} = Leitungswiderstand von Klemme a- zum RTD

Bild A-21 Verdrahten des Widerstandstemperaturfühlers mit dem Geber über 4, 3 oder 2 Leitungen

A

Statusanzeigen des EM 231 RTD-Moduls

Das RTD-Modul liefert dem Automatisierungssystem Datenwörter, die Temperaturen oder Fehlerbedingungen anzeigen. Statusbits zeigen Bereichsfehler und Anwenderspannungs-/Modulausfälle an. LEDs zeigen den Status des Moduls an. Ihr Programm muss Logik umfassen, die Fehlerbedingungen erkennt und entsprechend der Anwendung reagiert. Tabelle A-31 zeigt die Statusanzeigen des EM 231 RTD-Moduls.

Tipp

Das Format der Kanaldaten ist ein Zweierkomplement, 16-Bit-Wörter. Die Temperatur wird in Einheiten von 0,1 Grad dargestellt. (Beispiel: Beträgt die gemessene Temperatur 100,2 Grad, wird der Wert 1002 ausgegeben.) Widerstandsdaten werden auf 27648 skaliert. Beispiel: 75% des Vollausschlag-Widerstands wird ausgegeben als 20736.

(225 Ω / 300 Ω * 27648 = 20736)

Tabelle A-31 Statusanzeigen des EM 231 RTD-Moduls

Fehlerbedingung	Kanaldaten	SF LED Rot	24 V LED Grün	Statusbit Bereich[1]	Fehlerhafte 24-V-DC-Spannung[2]
Keine Fehler	Umwandlungsdaten	AUS	EIN	0	0
24 V fehlt	32766	AUS	AUS	0	1
SW erkennt offene Leitungen	-32768/32767	BLINKT	EIN	1	0
Bereichsüberschreitung	-32768/32767	BLINKT	EIN	1	0
Diagnosefehler[3]	0000	EIN	AUS	0	Hinweis [3]

1 Das Bereichs-Statusbit ist Bit 3 im Byte des Fehlerregisters des Moduls (SMB9 für Modul 1, SMB11 für Modul 2 usw.)
2 Das Statusbit Fehlerhafte Spannung ist Bit 2 im Byte des Fehlerregisters des Moduls (z.B. SMB 9, SMB 11 usw., siehe Anhang D).
3 Diagnosefehler bewirken einen Modul-Konfigurationsfehler. Das Statusbit Fehlerhafte Spannung kann oder kann nicht vor dem Modul-Konfigurationsfehler gesetzt werden.

Die Kanaldaten werden alle 405 ms aktualisiert, wenn das Automatisierungssystem die Daten gelesen hat. Liest das Automatisierungssystem die Daten nicht innerhalb einer Aktualisierungszeit, gibt das Modul alte Daten aus, bis das Modul erneut nach dem Lesen des Automatisierungssystems aktualisiert wird. Damit die Kanaldaten immer aktuell bleiben, ist es empfehlenswert, dass das Programm des Automatisierungssystems die Daten mindestens genauso häufig liest, wie das Modul aktualisiert wird.

Tipp

Wenn Sie mit dem RTD-Modul arbeiten, müssen Sie die Analogeingabefilter im Automatisierungssystem ausschalten. Durch die Analogfilter kann es sein, dass Fehlerbedingungen nicht rechtzeitig erkannt werden.

A

Die Prüfung auf offene Leitungen wird von interner Software des RTD-Moduls ausgeführt. Eingaben außerhalb des Bereichs werden angezeigt und Daten offener Leitungen werden als Burnout-Daten ausgegeben. Die Prüfung auf offene Leitungen benötigt mindestens drei Zyklen des Moduls. Sie kann auch länger dauern, je nachdem, welche Leitungen offen sind. Offene Leitungen Quelle+ und/oder Quelle- werden in der Mindestzeit erkannt. Offene Leitungen Fühler+ und/oder Fühler- werden innerhalb von 5 oder mehr Sekunden erkannt. Offene Fühlerleitungen können zufälligerweise gültige Daten darstellen, wenn die offene Leitung zwischenzeitlich erkannt wird. Dies tritt insbesondere in Umgebungen mit elektrischen Störungen auf. Elektrische Störungen können auch die Zeit verlängern, die benötigt wird, um eine offene Leitung zu erkennen. Es ist empfehlenswert, die Anzeigen für offene Leitungen/Werte außerhalb des Bereichs im Anwendungsprogramm zu speichern und zu halten, nachdem gültige Daten ausgegeben wurden.

Bereiche des EM 231 RTD-Moduls

Die Temperaturbereiche und die Genauigkeit für jede Art von RTD-Modul beim EM 231 RTD sind in den Tabellen A-32 und A-33 aufgeführt.

Tabelle A-32 Temperaturbereiche (°C) und Genauigkeit von RTD-Typen

Systemwort (1 Ziffer = 0,1 °C)		PT10000	Pt100, Pt200, Pt500, Pt1000	Ni100, Ni120, Ni1000	Cu9,035	0 - 150 Ω	0 - 300 Ω	0 - 600Ω	
Dezimal	Hexadezimal								
32767	7FF.								
32766	7FFE					↑	↑	↑	
32511	7EFF					176,383 Ω	352,767 Ω	705,534 Ω	
29649	6C01					150,005 Ω	300,011 Ω	600,022 Ω	
27648	6C00					150,000 Ω	300,000 Ω	600,000 Ω	
25000	61A8								↑
18000	4650								ODER
15000	3A98								
13000	32C8	↑	↑						
10000	2710	1000,0 °C	1000,0 °C						
8500	2134		850,0 °C						
6000	1770	600,0 °C			↑				
3120	0C30			↑	312,0 °C				
2950	0B86			295,0 °C					
2600	0A28				260,0 °C				
2500	09C4			250,0 °C					
1	0001	0,1 °C	0,1 °C	0,1 °C	0,1 °C	0,005 Ω	0,011 Ω	0,022 Ω	
0	0000	0,0 °C	0,0 °C	0,0 °C	0,0 °C	0,000 Ω	0,000 Ω	0,000 Ω	
-1	FFFF	-0,1 °C	-0,1 °C	-0,1 °C	-0,1 °C	(negative Werte sind nicht möglich)			
						↓	↓	↓	NR
-600	FDA8			-60,0 °C					
-1050	FBE6			-105,0 °C					
				↓					
-2000	F830	-200,0 °C	-200,0 °C		-200,0 °C				
-2400	F6A0				-240,0 °C				
-2430	F682	-243,0 °C	-243,0 °C		↓				
		↓	↓						
-5000	EC78								
-6000	E890								UR
-10500	D6FC								↓
-12000	D120								
-20000	4E20								
-32767	8001								
-32768	8000								
Genauigkeit über Vollbereich		±0,4%	±0,1%	±0,2%	±0,5%	±0,1%	±0,1%	±0,1%	
Genauigkeit (Nennbereich)		±4°C	±1°C	±0,6° C	±2,8° C	±0,15 Ω	±0,3 Ω	±0,6 Ω	

*OF = Überlauf; OR = Überhang; KNURL = Nennbereich; OUR = Unterhang; OUI = Unterlauf

↑ und ↓ bedeuten, dass alle Analogwerte, die den Bereich überschreiten, den eingestellten Burnout-Wert ausgeben, 32767 (0x7FF) oder -32768 (0x8000).

Tabelle A-33 Temperaturbereiche (°F) für RTD-Typen

Systemwort (1 Ziffer = 0,1 °F)		PT1000	Pt100, Pt200, Pt500, Pt1000	Ni100, Ni120, Ni1000	Cu 9,035	
Dezimal	Hexadezimal					
32767	7FF.					
32766	7PHAGE					
						↑
						Überhang
		↑	↑			
18320	4790	1832,0 °F	1832,0 °F			
15620	3D04		1562,0°F			
11120	2B70	1112,0°F				
					↑	
5936	1730			↑	593,6°F	
5630	15FE			563,0°F		
5000	1388				500,0°F	
4820	12D4			482,0°F		
						Normalbereich
1	0001	0,1 °F	0,1 °F	0,1 °F	0,1 °F	
0	0000	0,0 °F	0,0 °F	0,0 °F	0,0 °F	
-1	FFFF	-0,1 °F	-0,1 °F	-0,1 °F	-0,1 °F	
-760	FD08			-76,0°F		
-1570	F9DE			-157,0°F		
				↓		
-3280	F330	-328,0 °F	-328,0 °F		-328,0 °F	
-4000	F060				-400,0°F	
-4054	F02A	-405,4°F	-405,4°F		↓	
		↓	↓			
-5000	EC78					
-6000	E890					unterhalb des Bereichs
-10500	D6FC					↓
-32767	8001					
-32768	8000					

↑ und ↓ bedeuten, dass alle Analogwerte, die den Bereich überschreiten, den eingestellten Burnout-Wert ausgeben, 32767 (0x7FFF) oder -32768 (0x8000).

A

Technische Daten des EM 277 PROFIBUS-DP-Moduls

Tabelle A-34 Bestellnummer EM 277 PROFIBUS-DP-Modul

Bestellnummer	Erweiterungsmodule	EM-Eingänge	EM-Ausgänge	Steckbarer Klemmenblock
6ES7 277-0AA22-0XA0	EM 277 PROFIBUS-DP	–	–	Nein

Tabelle A-35 Allgemeine technische Daten des EM 277 PROFIBUS-DP-Moduls

Bestellnummer	Name und Beschreibung des Moduls	Abmessungen (mm) (B x H x T)	Gewicht	Verlust-leistung	V-DC-Bedarf +5 V DC	V-DC-Bedarf +24 V DC
6ES7 277-0AA22-0XA0	EM 277 PROFIBUS-DP	71 x 80 x 62	175 g	2,5 W	150 mA	Siehe unten

Tabelle A-36 Technische Daten des EM 277 PROFIBUS-DP-Moduls

Allgemein	6ES7 277-0AA22-0XA0
Anzahl Schnittstellen	1
Elektrische Schnittstelle	RS-485
PROFIBUS-DP/MPI Baudraten (automatisch eingestellt)	9,6 / 19,2 / 45,45 / 93,75 / 187,5 und 500 kBaud; 1 / 1,5 / 3 / 6 und 12 MBaud
Protokolle	PROFIBUS-DP-Slave und MPI-Slave
Kabellänge	
Bis zu 93,75 kBaud	1200 m
187,5 kBaud	1000 m
500 kBaud	400 m
1 bis 1,5 MBaud	200 m
3 bis 12 MBaud	100 m
Netzfähigkeiten	
Einstellungen Stationsadresse	0 bis 99 (einstellbar über Drehschalter)
Max. Anzahl Stationen pro Segment	32
Max. Anzahl Stationen pro Netz	126, max. 99 EM 277-Stationen
MPI-Verbindungen	6 gesamt, 2 reserviert (1 für PG und 1 für OP)
Leistungsbedarf 24-V-DC-Eingang	
Spannungsbereich	20,4 bis 28,8 V DC (Klasse 2 oder Geberspannung aus CPU)
Maximalstrom Modul nur mit Anschluss aktiv 90 mA von 5 V Anschlusslast addieren 120 mA von 24 V Anschlusslast addieren	 30 mA 60 mA 180 mA
Welligkeit/Störströme (<10 MHz)	Max. < 1 V Spitze-Spitze
Elektrische Trennung (Feld zu Logik)[1]	500 V AC für 1 Minute
5-V-DC-Leistung an Kommunikationsschnittstelle	
Maximalstrom pro Schnittstelle	90 mA
Elektrische Trennung (24 V DC zu Logik)	500 V AC für 1 Minute
24-V-DC-Leistung an Kommunikationsschnittstelle	
Spannungsbereich	20,4 bis 28,8 V DC
Maximalstrom pro Schnittstelle	120 mA
Strombegrenzung	0,7 bis 2,4 A
Trennung	Nicht elektrisch getrennt, gleicher Stromkreis wie Eingang 24 V DC

[1] Die 24-V-DC-Versorgung liefert der Logik des Moduls keine Spannung. Die 24-V-DC-Versorgung liefert die Spannung für die Kommunikationsschnittstelle.

S7-200 CPUs, die intelligente Module unterstützen

Das EM 277 PROFIBUS-DP Slave-Modul ist ein intelligentes Erweiterungsmodul und kann zusammen mit den in Tabelle A-37 aufgeführten S7-200 CPUs eingesetzt werden.

Tabelle A-37 Kompatibilität des EM 277 PROFIBUS-DP-Moduls mit den S7-200 CPUs

CPU	Beschreibung
CPU 222 ab Ausgabestand 1.10	CPU 222 DC/DC/DC
	CPU 222 AC/DC/Relais
CPU 224 ab Ausgabestand 1.10	CPU 224 DC/DC/DC
	CPU 224 AC/DC/Relais
CPU 226 ab Ausgabestand 1.00	CPU 226 DC/DC/DC
	CPU 226 AC/DC/Relais
CPU 226XM ab Ausgabestand 1.00	CPU 226XM DC/DC/DC
	CPU 226XM AC/DC/Relais

Adressschalter und LEDs

Die Adressschalter und Status-LEDs befinden sich auf der Vorderseite des Moduls (siehe Bild A-22). Die Anschlussbelegung des Schnittstellenanschlusses für den DP-Slave wird auch dargestellt.

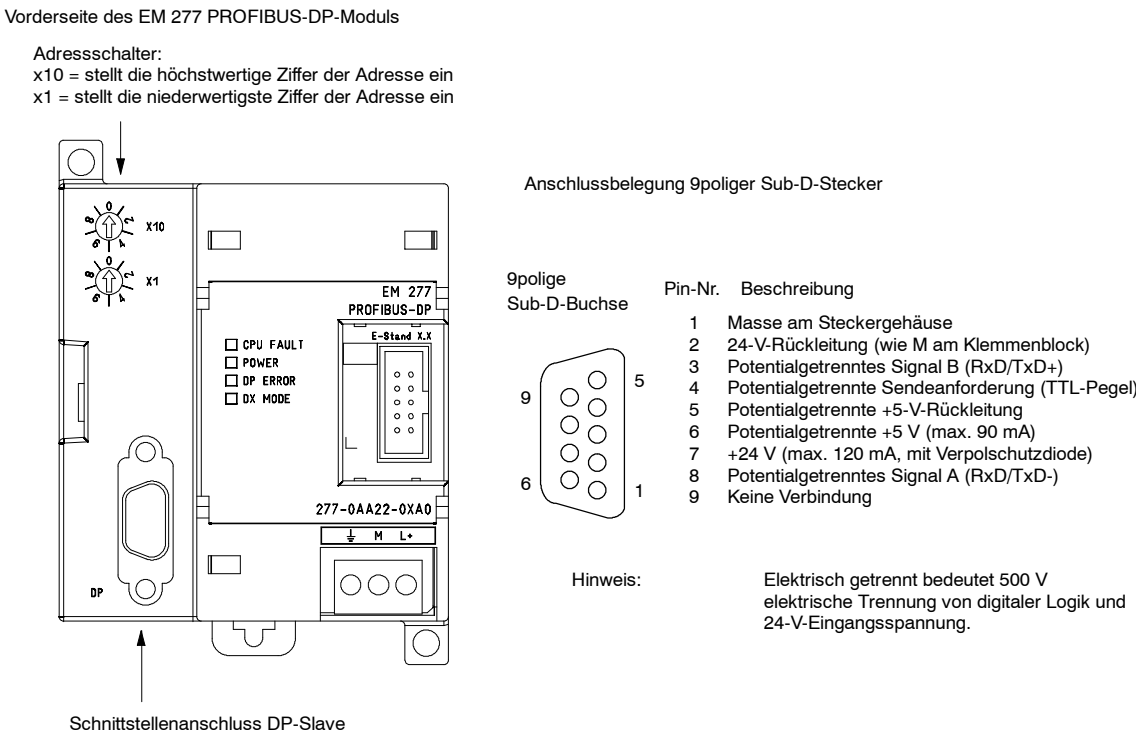


Bild A-22 EM 277 PROFIBUS-DP

DP-Standardkommunikation (dezentrale Peripherie)

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 Eingänge und Ausgänge. PROFIBUS bedeutet “Process Field Bus.”

Im Modul EM 277 PROFIBUS-DP 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.

Anschließen einer S7-200 als DP-Slave mittels EM 277

Die S7-200 CPU kann über ein Slave-Erweiterungsmodul EM 277 PROFIBUS-DP an ein PROFIBUS-DP-Netz angeschlossen werden. Das EM 277 wird über den seriellen E/A-Bus an die S7-200 CPU angeschlossen. Das PROFIBUS-Netz wird über die DP-Kommunikationsschnittstelle an das Modul EM 277 PROFIBUS-DP angeschlossen. Diese Schnittstelle arbeitet mit einer beliebigen PROFIBUS-Baudrate zwischen 9600 Baud und 12 MBaud. Die unterstützten Baudraten finden Sie in den technischen Daten des EM 277 PROFIBUS-DP-Moduls.

Als DP-Slave akzeptiert das EM 277 verschiedene E/A-Konfigurationen vom Master, so dass Sie die Menge der gesendeten Daten an die Anforderungen Ihrer Anwendung anpassen können. Im Gegensatz zu vielen DP-Geräten überträgt das Modul EM 277 nicht nur Ein- und Ausgabedaten. Eingänge, Zählerwerte, Zeitwerte oder andere berechnete Werte können zum Master übertragen werden, wenn sie zunächst im Variablenspeicher der S7-200 CPU gespeichert werden. Auch die vom Master empfangenen Daten werden im Variablenspeicher der S7-200 CPU abgelegt und können von dort in andere Speicherbereiche übertragen werden.

Die DP-Schnittstelle des Moduls EM 277 PROFIBUS-DP 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 A-23 zeigt ein PROFIBUS-Netz mit einer CPU 224 und einem EM 277 PROFIBUS-DP-Modul.

- Die CPU 315-2 ist der DP-Master und wurde von einem SIMATIC Programmiergerät mit der Programmiersoftware STEP 7 konfiguriert.
- Die CPU 224 ist ein DP-Slave, der der CPU 315-2 ”gehört”. Das E/A-Modul 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 224.

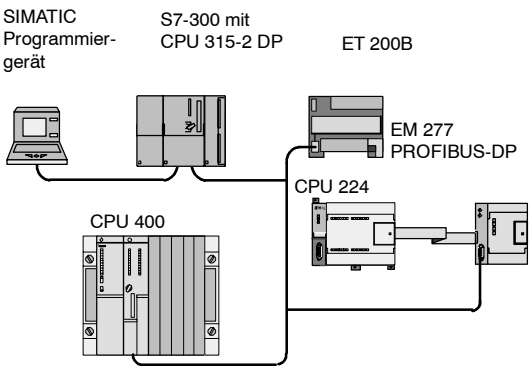


Bild A-23 Modul EM 277 PROFIBUS-DP und CPU 224 in einem PROFIBUS-Netz

A

Konfiguration

Damit Sie das EM 277 PROFIBUS-DP als DP-Slave einsetzen können, müssen Sie die Stationsadresse der DP-Schnittstelle so einstellen, dass sie der Adresse in der Konfiguration des Master entspricht. Die Stations-
adresse wird mit den Drehschaltern am Modul EM 277 eingestellt. Nachdem Sie mit den Schaltern eine neue Adresse eingestellt haben, müssen Sie die CPU aus- und wieder einschalten, damit die neue Adresse wirksam wird.

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

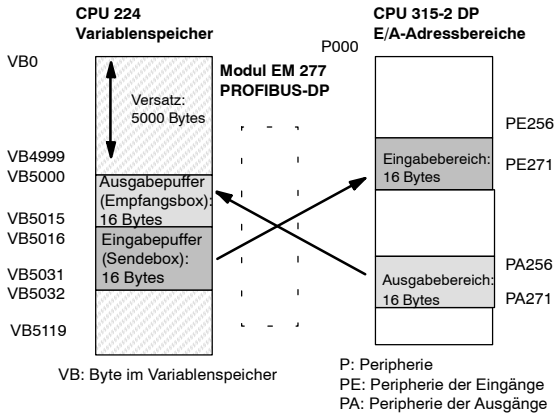


Bild A-24 Variablenspeicher und Adressbereich der Ein- und Ausgänge

Bild A-24 zeigt ein Beispiel für den Variablenspeicher und die Adressbereiche der Ein- und Ausgänge eines PROFIBUS-DP-Master.

Das EM 277 PROFIBUS-DP kann vom DP-Master so konfiguriert werden, dass es vom Master einen Ausgabepuffer mit Daten annimmt und einen Eingabepuffer mit Daten zurücksendet. Die Ausgabe- und Eingabedatenpuffer befinden sich beide im Variablenspeicher der S7-200 CPU. Beim Konfigurieren des DP-Master definieren Sie im Rahmen der Parametrierung des EM 277 eine Byteadresse im Variablenspeicher, an der der Ausgabedatenpuffer beginnen soll. Sie definieren auch die Konfiguration der Ein- und Ausgänge als Ausgabedatenmenge, die in die S7-200 CPU geschrieben werden soll und als Eingabedatenmenge, die von der S7-200 CPU empfangen werden soll. Das EM 277 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 das Modul EM 277 PROFIBUS-DP. Das EM 277 überträgt dann die Adresse des Variablenspeichers sowie die Länge der Ein- und Ausgabedaten an die S7-200 CPU.

Bild A-24 zeigt ein Modell des Variablenspeichers in einer CPU 224 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 224 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.

A

Tipp

Wenn Sie mit einer Dateneinheit (konsistente Daten) von drei Bytes oder mit Dateneinheiten von mehr als vier Bytes arbeiten, müssen Sie mit SFC14 die Eingänge des DP-Slave lesen und mit SFC15 die Ausgänge des DP-Slave adressieren. Weitere Informationen hierzu finden Sie im Referenzhandbuch *System- und Standardfunktionen für S7-300/400*.

Tabelle A-38 führt die vom Modul EM 277 PROFIBUS-DP unterstützten Konfigurationen auf. Die Standard-Konfiguration des Moduls EM 277 besteht aus zwei Eingabe- und zwei Ausgabewörtern.

Tabelle A-38 Konfigurationsmöglichkeiten beim EM 277

Konfiguration	Eingaben in den Master	Ausgaben aus dem Master	Datenkonsistenz
1	1 Wort	1 Wort	Wortkonsistenz
2	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	

Sie können die Adressen der Eingabe- und Ausgabepuffer so einstellen, dass sich die Puffer an einer beliebigen Stelle im Variablenspeicher der S7-200 CPU 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 S7-200 CPU schreibt. Sie konfigurieren den Master so, 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 PROFIBUS
- Für SIMATIC S7-Master: Programmiersoftware STEP 7
- Für SIMATIC 505-Master: COM PROFIBUS und TISOFT2 oder SoftShop

Ausführliche Informationen zur Verwendung dieser Konfigurations-/Programmierpakete finden Sie in den Handbüchern zu den jeweiligen Werkzeugen. Das Handbuch *ET 200 Dezentrale Peripheriesysteme* bietet ausführliche Informationen zu PROFIBUS-Netzen und den Komponenten.

Datenkonsistenz

PROFIBUS unterstützt drei Arten der Datenkonsistenz:

- Durch die Bytekonsistenz wird sichergestellt, dass die Bytes als ganze Einheiten übertragen werden.
- Durch Wortkonsistenz wird sichergestellt, dass die Übertragung von Worten nicht durch andere Vorgänge in der CPU unterbrochen werden kann (die beiden Bytes, aus denen sich ein Wort zusammensetzt, werden immer zusammen übertragen und können nicht getrennt werden). Sie sollten mit Wortkonsistenz arbeiten, wenn es sich bei den zu übertragenden Datenwerten um ganze Zahlen handelt.
- 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. Pufferkonsistenz sollten Sie verwenden, wenn es sich bei den zu übertragenden Datenwerten um Doppelwörter oder Gleitpunktwerte handelt oder wenn sich eine Gruppe von Werten auf eine Berechnung bzw. ein Objekt bezieht.

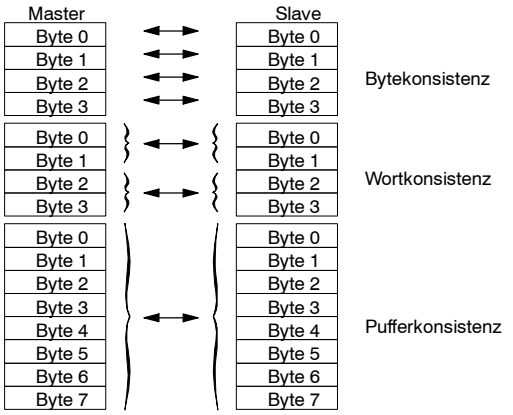


Bild A-25 Byte-, Wort- und Pufferkonsistenz

Die Datenkonsistenz stellen Sie als Teil der E/A-Konfigration des Master 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. Die verschiedenen Konsistenzarten werden in Bild A-25 gezeigt.

Hinweise zum Anwenderprogramm

Nachdem das Modul EM 277 PROFIBUS-DP erfolgreich von einem DP-Master konfiguriert wurde, gehen das EM 277 und der DP-Master in den Modus zum Datenaustausch. Im Modus für Datenaustausch schreibt der Master Ausgabedaten in das Modul EM 277 PROFIBUS-DP, und das EM 277 reagiert, indem es die aktuellen Eingabedaten der S7-200 CPU sendet. Das Modul EM 277 aktualisiert die Eingaben der S7-200 CPU ständig, um dem DP-Master immer die aktuellen Eingabedaten zur Verfügung stellen zu können. Das Modul überträgt dann die Ausgabedaten an die S7-200 CPU. 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.

A

Das Anwenderprogramm der S7-200 CPU 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 zum EM 277 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.

Die Anfangsadressen der Datenpuffer im Variablenspeicher und deren Länge müssen bekannt sein, wenn ein Anwenderprogramm für die S7-200 CPU erstellt werden soll.

Statusinformationen

Es gibt 50 Bytes an Sondermerkern (SM), die je nach physikalischer Anordnung den intelligenten Modulen zugeordnet werden. Das Modul aktualisiert die SM-Adressen entsprechend der relativen Anordnung des Moduls zur CPU (mit Rücksicht auf andere Module). Das erste Modul aktualisiert SMB200 bis SMB249. Das zweite Modul aktualisiert SMB250 bis SMB299 usw. (siehe Tabelle A-39).

Hinweis

Ab Version 2.2 werden die Sondermerker für intelligente Module auf andere Art und Weise zugeordnet.

Wenn Sie mit einer CPU vor Version 2.2 arbeiten, müssen Sie alle intelligenten Module in Steckplätzen neben der CPU und vor allen nicht intelligenten Modulen anordnen, um die Kompatibilität sicherzustellen.

Tabelle A-39 Sondermerkerbytes SMB200 bis SMB549

Sondermerkerbytes SMB200 bis SMB549						
Intelligentes Modul in Steckplatz 0	**Intelligentes Modul in Steckplatz 1**	**Intelligentes Modul in Steckplatz 2**	**Intelligentes Modul in Steckplatz 3**	**Intelligentes Modul in Steckplatz 4**	**Intelligentes Modul in Steckplatz 5**	**Intelligentes Modul in Steckplatz 6**
SMB200 bis SMB249	SMB250 bis SMB299	SMB300 bis SMB349	SMB350 bis SMB399	SMB400 bis SMB449	SMB450 bis SMB499	SMB500 bis SMB549

Diese Sondermerker verfügen über voreingestellte Werte, wenn die DP-Kommunikation mit dem Master nicht besteht. Nachdem ein Master die Parametrierung und die E/A-Konfiguration in das Modul EM 277 PROFIBUS-DP geschrieben hat, speichern diese Sondermerker die vom DP-Master eingestellte Konfiguration. Sie müssen das Protokoll-Statusbyte prüfen (z.B. SMB224 für Steckplatz 0), um sicherzustellen, dass sich das EM 277 im Modus für Datenaustausch mit dem Master befindet, bevor Sie die Informationen in den Sondermerkern in Tabelle A-40 oder Daten im Variablenspeicher nutzen.

Tipp

Sie können die Puffergrößen und die Pufferadressen für das EM 277 PROFIBUS-DP nicht dadurch ändern, dass Sie die Sondermerker ändern. Nur der DP-Master kann das Modul EM 277 PROFIBUS-DP für den DP-Betrieb einrichten.

Tabelle A-40 Sondermerkerbytes für das EM 277 PROFIBUS-DP

Intelligentes Modul in Steckplatz 0	**...**	**Intelligentes Modul in Steckplatz 6**	**Beschreibung**
SMB200 bis SMB215	...	SMB500 bis SMB515	Modulname (16 ASCII-Zeichen) “EM277 ProfibusDP”
SMB216 bis SMB219	...	SMB516 bis SMB519	S/W-Versionsnummer (4 ASCII-Zeichen) xxxx
SMW220	...	SMW520	Fehlercode 16#0000 Kein Fehler 16#0001 Keine Anwenderspannung 16#0002 bis 16#FFFF Reserviert.
SMB222	...	SMB522	Stationsadresse des DP-Slave-Moduls, eingestellt durch Adressschalter (0 - 99 dezimal)
SMB223	...	SMB523	Reserviert.
SMB224	...	SMB524	Statusbyte des DP-Standardprotokolls MSB LSB 0 0 0 0 0 0 S1 S0 s1 S0 DP-Standard Statusbyte 0 0 DP-Kommunikation nach Einschalten nicht initiiert 0 1 Konfigurations-/Parametrierungsfehler 1 0 Im Modus für Datenaustausch 1 1 Modus zum Datenaustausch verloren
SMB225	...	SMB525	DP-Standardprotokoll - Adresse des Master des Slave (0 bis 126)
SMW226	...	SMW526	DP-Standardprotokoll - Adresse im Variablenspeicher des Ausgabepuffers als Versatz von VB0
SMB228	...	SMB528	DP-Standardprotokoll - Anzahl der Bytes an Ausgabedaten
SMB229	...	SMB529	DP-Standardprotokoll - Anzahl der Bytes an Eingabedaten
SMB230 bis SMB249	...	SMB530 bis SMB549	Reserviert - wird beim Anlauf gelöscht

Hinweis: Die Sondermerker werden jedesmal aktualisiert, wenn das DP-Slave-Modul Konfigurations- bzw. Parametrierungsinformationen übernimmt. Diese Adressen werden aktualisiert, auch wenn ein Konfigurations- bzw. Parametrierungsfehler erkannt wird. Die Adressen werden bei jedem Anlauf gelöscht.

LED-Statusanzeigen des EM 277 PROFIBUS-DP

Das Modul EM 277 PROFIBUS-DP besitzt vier Status-LEDs auf der Vorderseite des Moduls, die den Betriebszustand der DP-Schnittstelle anzeigen:

- Nach dem Einschalten der S7-200 CPU bleibt die Anzeige DX MODE solange ausgeschaltet, bis die DP-Kommunikation aufgerufen wird.
- Wird die DP-Kommunikation erfolgreich initiiert (das Modul EM 277 PROFIBUS-DP geht in den Modus zum Datenaustausch mit dem Master), dann leuchtet die Anzeige DX MODE grün auf und bleibt solange eingeschaltet, bis der Modus zum Datenaustausch wieder verlassen wird.
- Geht die DP-Kommunikation verloren, wodurch das Modul EM 277 den Modus für Datenaustausch verlassen muss, wird die Anzeige DX MODE ausgeschaltet und die Anzeige DP ERROR leuchtet rot auf. Dieser Zustand bleibt solange erhalten, bis die S7-200 CPU ausgeschaltet wird oder der Modus zum Datenaustauch wieder aufgenommen wird.
- Liegt in der E/A-Konfiguration oder in der Parametrierung, die der DP-Master in das Modul EM 277 schreibt, ein Fehler vor, dann blinkt die Anzeige DP ERROR rot.
- Wird keine 24-V-DC-Spannung geliefert, ist die Anzeige POWER ausgeschaltet.

Tabelle A-41 führt die verschiedenen Zustände der Status-LEDs des EM 277 auf.

Tabelle A-41 Status-LEDs des Moduls EM 277 PROFIBUS-DP

LED	AUS	Rot	Rot blinkend	Grün
CPU FAULT	Modul ist in Ordnung	Interner Modulfehler	--	--
POWER	Keine 24-V-DC-Spannung	--	--	24-V-DC-Spannung vorhanden
DP ERROR	Kein Fehler aufgetreten	Modus für Datenaustausch verlassen	Parametrierungs- / Konfigurationsfehler	--
DX MODE	Nicht im Modus für Datenaustausch	--	--	Im Modus für Datenaustausch

Hinweis: Wird das Modul EM 277 PROFIBUS-DP nur als MPI-Slave eingesetzt, ist nur die grüne Anzeige POWER eingeschaltet.

Weitere Konfigurationsfunktionen

Das Modul EM 277 PROFIBUS-DP kann als Kommunikationsschnittstelle zu anderen MPI-Mastern eingesetzt werden, ganz gleich ob es als PROFIBUS-DP-Slave eingesetzt wird oder nicht. Das Modul kann mittels der Funktionen XGET/XPUT der S7-300/400 eine Verbindung von der S7-300/400 zur S7-200 herstellen. Mit STEP 7-Micro/WIN und einer Netzkarte (z.B. der CP5611) und dem MPI- oder PROFIBUS-Parametersatz, mit einem OP oder dem TD 200 (Version 2.0 oder höher, Bestellnummer 6ES7 272-0AA20-0YA0) können Sie über das Modul EM 277 PROFIBUS-DP mit der S7-200 kommunizieren.

Maximal sechs Verbindungen (sechs Geräte) können zusätzlich zum DP-Master an das Modul EM 277 PROFIBUS-DP angeschlossen werden. Eine Verbindung ist reserviert für ein Programmiergerät (PG) und eine ist reserviert für ein Operator Panel (OP). Die anderen vier Verbindungen können von einem beliebigen MPI-Master genutzt werden. Damit das Modul EM 277 PROFIBUS-DP mit mehreren Mastern kommunizieren kann, müssen alle Master mit der gleichen Baudrate betrieben werden. Eine mögliche Netzkonfiguration zeigt Bild A-26.

Wird das Modul EM 277 PROFIBUS-DP für die MPI-Kommunikation eingesetzt, muss der MPI-Master die Stationsadresse des Moduls für alle Meldungen verwenden, die an die S7-200 gesendet werden, an die das Modul angeschlossen ist. MPI-Meldungen, die an das Modul EM 277 PROFIBUS-DP gesendet werden, werden an die S7-200 weitergeleitet.

Das Modul EM 277 PROFIBUS-DP ist ein Slave-Modul und kann nicht mit den Funktionen NETR und NETW für die Kommunikation zwischen S7-200 CPUs eingesetzt werden. Das EM 277 PROFIBUS-DP-Modul kann nicht für die frei programmierbare Kommunikation eingesetzt werden.

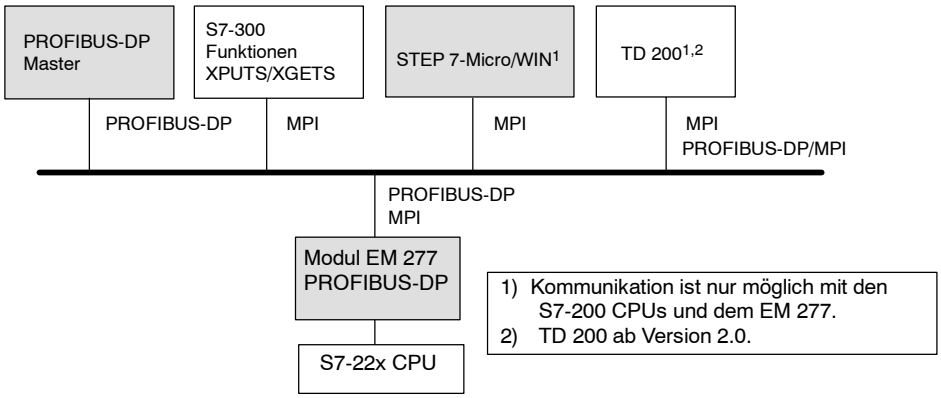


Bild A-26 PROFIBUS-DP/MPI-Netz

Gerätestammdaten: GSD

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 Gerätestammdaten (GSD-Datei), angegeben. Konfigurationswerkzeuge, denen die Gerätestammdaten 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 PROFIBUS bzw. der Software STEP 7 enthalten die Konfigurationsdateien für das Modul EM 277 PROFIBUS-DP. Enthält Ihre Version der Software keine Konfigurationsdatei für das EM 277, finden Sie die aktuelle GSD-Datei (SIEM089D.GSD) auf der Web-Seite www.profibus.com.

Arbeiten Sie mit einem Master nicht von Siemens, erfahren Sie in der Dokumentation des Herstellers, wie Sie den Master mit Hilfe der GSD-Datei konfigurieren.

```
;================================================
; GSD File for the EM 277 PROFIBUS-DP with a DPC31
; MLFB  : 6ES7 277-0AA2.-0XA0
; DATE  : 26-March-2001
;================================================
#Profibus_DP
;General parameters
GSD_Revision          = 1
Vendor_Name           = "Siemens"
Model_Name            = "EM 277 PROFIBUS-DP"
Revision              = "V1.02"
Ident_Number          = 0x089D
Protocol_Ident        = 0
Station_Type          = 0
FMS_supp              = 0
Hardware_Release      = "1.00"
Software_Release      = "1.02"
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
MaxTsdr_9.6           = 60
MaxTsdr_19.2          = 60
MaxTsdr_45.45         = 250
MaxTsdr_93.75         = 60
MaxTsdr_187.5         = 60
MaxTsdr_500           = 100
MaxTsdr_1.5M          = 150
MaxTsdr_3M            = 250
MaxTsdr_6M            = 450
MaxTsdr_12M           = 800
Redundancy            = 0
Repeater_Ctrl_Sig     = 2
24V_Pins              = 2

; Slave-Specification:
OrderNumber="6ES7 277-0AA2.-0XA0"
Periphery="SIMATIC S5"
Slave_Family=10@TdF@SIMATIC

Freeze_Mode_supp      = 1
Sync_Mode_supp        = 1
Set_Slave_Add_Supp    = 0
Auto_Baud_supp        = 1
Min_Slave_Intervall   = 1
Fail_Safe             = 0
Max_Diag_Data_Len     = 6
Modul_Offset          = 0
Modular_Station       = 1
Max_Module            = 1
Max_Input_len         = 128
Max_Output_len        = 128
Max_Data_len          = 256

; UserPrmData-Definition
ExtUserPrmData=1 "I/O Offset in the V-memory"
Unsigned16 0 0-10239
EndExtUserPrmData
; UserPrmData: Length and Preset:
User_Prm_Data_Len=3
User_Prm_Data= 0,0,0
Max_User_Prm_Data_Len=3
Ext_User_Prm_Data_Const(0)=0x00,0x00,0x00
Ext_User_Prm_Data_Ref(1)=1
```

```
;================================================
; Continuation of GSD File
;================================================

; Module Definition List
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
```

Bild A-27 GSD-Datei des EM 277 PROFIBUS-Moduls

A

Programmierbeispiel für die DP-Kommunikation mit einer CPU

Im folgenden finden Sie ein Beispielprogramm in der Anweisungsliste für das PROFIBUS-DP-Modul in Steckplatz 0 für eine CPU, die die DP-Schnittstelleninformationen der Sondermerker nutzt. In diesem Programm werden die Adressen der DP-Puffer über SMW226 ermittelt und die Größen der Puffer aus SMB226 und SMB229 ausgelesen. Diese Informationen werden dazu verwendet, die Daten aus dem DP-Ausgabepuffer in das Prozessabbild der Ausgänge der CPU zu kopieren. Gleichermaßen werden die Daten aus dem Prozessabbild der Eingänge der CPU in den Eingabepuffer im Variablenspeicher kopiert.

Hinweis

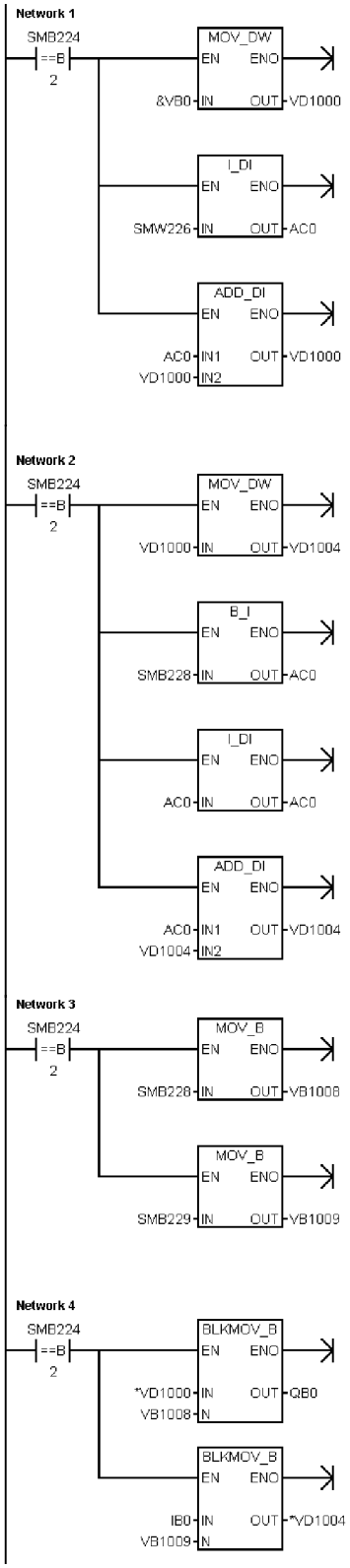
Ab Version 2.2 werden die Sondermerker für intelligente Module auf andere Art und Weise zugeordnet.

Wenn Sie mit einer CPU vor Version 2.2 arbeiten, müssen Sie alle intelligenten Module in Steckplätzen neben der CPU und vor allen nicht intelligenten Modulen anordnen, um die Kompatibilität sicherzustellen.

Im folgenden Beispielprogramm für ein DP-Modul in Steckplatz 0 liefern die DP-Konfigurationsdaten im Speicherbereich der Sondermerker die Konfiguration des DP-Slave. Das Programm verwendet folgende Daten:

SMW220	Fehlerstatus DP-Modul
SMB224	DP-Status
SMB225	Masteradresse
SMW226	Versatz im Variablenspeicher für die Ausgänge
SMB228	Anzahl der Bytes an Ausgabedaten
SMB229	Anzahl der Bytes an Eingabedaten
VD1000	Pointer auf Ausgabedaten
VD1004	Pointer auf Eingabedaten

Beispiel für die DP-Kommunikation mit einer CPU



Netzwerk 1 //Pointer auf Ausgabedaten berechnen. Wenn im Modus für //Datenaustausch:
//1. Der Ausgabepuffer ist ein Versatz von VB0.
//2. Versatz im Variablenspeicher in ganze Zahl (32 Bit) // umwandeln.
//3. Zu Adresse VB0 addieren, um Pointer auf //Ausgabedaten zu //erhalten.

```
LDB =   SMB224, 2
MOVD    &VB0, VD1000
ITD     SMW226, AC0
+D      AC0, VD1000
```

Netzwerk 2 //Pointer auf Eingabedaten berechnen. Wenn im Modus für //Datenaustausch:
//1. Pointer auf Ausgabedaten kopieren.
//2. Anzahl der Ausgabebytes holen.
//3. Pointer auf Ausgabedaten addieren, um Pointer auf //erste // Eingabedaten zu erhalten.

```
LDB =   SMB224, 2
MOVD    VD1000, VD1004
BTI     SMB228, AC0
ITD     AC0, AC0
+D      AC0, VD1004
```

Netzwerk 3 //Zu kopierende Datenmenge einrichten. Wenn im Modus für //Datenaustausch:
//1. Anzahl zu kopierender Ausgabebytes holen.
//2. Anzahl zu kopierender Eingabebytes holen.

```
LDB=  SMB224, 2
MOVB  SMB228, VB1008
MOVB  SMB229, VB1009
```

Netzwerk 4 //Ausgaben des Master in Ausgaben der CPU übertragen. //Eingaben der CPU in Eingaben des Master kopieren. //Wenn im Modus für Datenaustausch:
//1. Ausgaben des Master in Ausgaben der CPU kopieren.
//2. Eingaben der CPU in Eingaben des Master kopieren.

```
LDB =   SMB224, 2
BMB     *VD1000, AB0, VB1008
BMB     EB0, *VD1004, VB1009
```

A

Technische Daten des Modemmoduls EM 241

Tabelle A-42 Bestellnummer Modemmodul EM 241

Bestellnummer	Erweiterungsmodule	EM-Eingänge	EM-Ausgänge	Steckbarer Klemmenblock
6ES7 241-1AA22-0XA0	Modemmodul EM 241	–	8[1]	Nein

1 Acht Ausgänge werden zur logischen Steuerung der Modemfunktion verwendet, sie steuern nicht direkt externe Signale.

Tabelle A-43 Allgemeine technische Daten des Modemmoduls EM 241

Bestellnummer	Name und Beschreibung des Moduls	Abmessungen (mm) (B x H x T)	Gewicht	Verlust-leistung	V-DC-Bedarf +5 V DC	V-DC-Bedarf +24 V DC
6ES7 241-1AA22-0XA0	Modemmodul EM 241	71,2 x 80 x 62	190 g	2,1 W	80 mA	70 mA

Tabelle A-44 Technische Daten des Modemmoduls EM 241

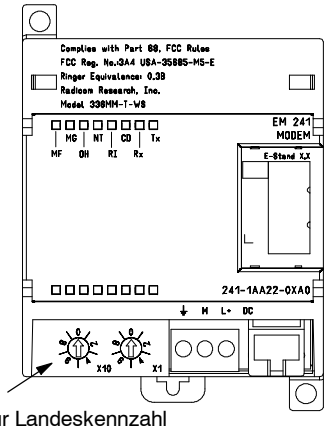
Allgemein	6ES7 241-1AA22-0XA0
Telefonanschluss	
Trennung (Telefonleitung zu Logik und Feldleistung)	1500 V AC (galvanisch)
Physikalischer Anschluss	RJ11 (6 Position, 4 Leitungen)
Modem-Normen	Bell 103, Bell 212, V.21, V.22, V.22 bis, V.23c, V.32, V.32 bis, V.34 (Voreinstellung)
Zugriffsschutz	Passwort Rückruf
Wählverfahren	Impuls oder Ton
Nachrichtenübermittlungsprotokolle	Numerisch TAP (alphanumerisch) UCP-Befehle 1, 30, 51
Industrie-Protokolle	Modbus PPI
Leistungsbedarf 24-V-DC-Eingang	
Spannungsbereich	20,4 bis 28,8 V DC
Elektrische Trennung (Feldspannung zu Logik)	500 V AC für 1 Minute

Das Modemmodul EM 241 ersetzt die Funktion eines externen Modems, das an die Kommunikationsschnittstelle der CPU angeschlossen ist. Wenn Sie ein EM 241 in Ihrem S7-200 System haben, brauchen Sie nur noch einen PC mit einem externen Modem und STEP 7-Micro/WIN, um von einem entfernten Standort mit Ihrer CPU zu kommunizieren.

In Kapitel 7, Kommunikation im Netz, finden Sie Informationen zur Konfiguration. In Kapitel 10, Erstellen eines Programms für das Modemmodul, finden Sie Informationen zum Programmieren und zu den erweiterten Funktionen des Moduls.

Modem Assistent

Mit dem Modem-Assistenten in STEP 7-Micro/WIN können Sie ein Modemmodul EM 241 konfigurieren. Weitere Informationen zum Modem-Assistenten finden Sie in Kapitel 10.



Schalter für Landeskennzahl

Bild A-28 Schaltbild des Modemmoduls EM 241

S7-200 CPUs, die intelligente Module unterstützen

Das Modemmodul EM 241 ist ein intelligentes Erweiterungsmodul und kann zusammen mit den in Tabelle A-45 aufgeführten S7-200 CPUs eingesetzt werden.

Tabelle A-45 Kompatibilität des Modemmoduls EM 241 mit S7-200 CPUs

CPU	Beschreibung
CPU 222 ab Ausgabestand 1.10	CPU 222 DC/DC/DC und CPU 222 AC/DC/Relais
CPU 224 ab Ausgabestand 1.10	CPU 224 DC/DC/DC und CPU 224 AC/DC/Relais
CPU 226 ab Ausgabestand 1.00	CPU 226 DC/DC/DC und CPU 226 AC/DC/Relais
CPU 226XM ab Ausgabestand 1.00	CPU 226XM DC/DC/DC und CPU 226XM AC/DC/Relais

Einbauen des EM 241

Zum Einbauen des EM 241 gehen Sie folgendermaßen vor:

1. Lassen Sie das EM 241 auf der Standard-Hutschiene einrasten und stecken Sie das Flachkabel.
2. Schließen Sie 24 V DC von der CPU-Geberversorgung oder von einer externen Quelle an und verbinden Sie die Erdungsklemme mit Systemerde.
3. Stecken Sie die Telefonleitung in die RJ11-Buchse.
4. Stellen Sie die Schalter für die Landeskennzahl wie in Tabelle A-46 ein. Sie müssen die Schalter einstellen, bevor Sie die CPU mit Spannung versorgen, damit die richtige Landeskennzahl gelesen wird.
5. Schalten Sie die CPU ein. Die grüne Leuchte MG (Modul gut) leuchtet auf.

Ihr EM 241 ist jetzt kommunikationsbereit.

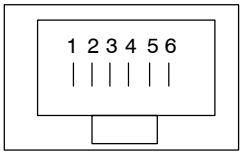
Tabelle A-46 Vom EM 241 unterstützte Landeskennzahlen

Code	Land	Telekom-Standard
00	Australien	ACA TS-002
01	Österreich	CTR21
02	Belgien	CTR21
05	Kanada	IC CS03
06	China	GB3482
08	Dänemark	CTR21
09	Finnland	CTR21
10	Frankreich	CTR21
11	Deutschland	CTR21
12	Griechenland	CTR21
16	Irland	CTR21
18	Italien	CTR21
22	Luxemburg	CTR21
25	Niederlande	CTR21
26	Neuseeland	
27	Norwegen	CTR21
30	Portugal	CTR21
34	Spanien	CTR21
35	Schweden	CTR21
36	Schweiz	CTR21
38	Großbritannien	CTR21
39	U.S.A.	FCC Teil 68

A

RJ11-Stecker

Bild A-29 zeigt die Einzelheiten des RJ11-Steckers. Sie können einen Adapter für andere genormte Telefon-
anschlüsse verwenden. Ausführliche Informationen finden Sie in der Dokumentation zu Ihrem Adapterstecker.



Pin	Beschreibung
3	Klingeln
4	Tipp

Umgekehrter Anschluss ist zulässig.

Bild A-29 Ansicht des RJ11-Steckers

Vorsicht

Blitzstöße oder andere unerwartete hohe Spannungen in der Telefonleitung können Ihr Modemmodul EM 241 beschädigen.

Verwenden Sie eine handelsübliche Überspannungsschutzeinrichtung für die Telefonleitung, wie sie auch häufig zum Schutz von PC-Modems verkauft wird. Der Überspannungsschutz kann beim Schützen Ihres Modemmoduls EM 241 beschädigt werden. Wählen Sie deshalb eine Überspannungsschutzeinrichtung, die anzeigt, ob sie funktionsfähig ist oder nicht.

Prüfen Sie Ihren Überspannungsschutz regelmäßig, damit Ihr Modemmodul EM 241 immer geschützt ist.

Technische Daten des Positioniermoduls EM 253

Tabelle A-47 Bestellnummer Positioniermodul EM 253

Bestellnummer	Erweiterungsmodule	EM-Eingänge	EM-Ausgänge	Steckbarer Klemmenblock
6ES7 253-1AA22-0XA0	Positioniermodul EM 253	–	8[1]	Ja

[1] Acht Ausgänge werden zur logischen Steuerung der Bewegungsfunktion verwendet, sie steuern nicht direkt externe Signale.

Tabelle A-48 Allgemeine technische Daten des Positioniermoduls EM 253

Bestellnummer	Name und Beschreibung des Moduls	Abmessungen (mm) (B x H x T)	Gewicht	Verlust-leistung	V-DC-Bedarf +5 V DC	V-DC-Bedarf +24 V DC
6ES7 253-1AA22-0XA0	Positioniermodul EM 253	71,2 x 80 x 62	0,190 kg	2,5 W	190 mA	Siehe unten

Tabelle A-49 Technische Daten des Positioniermoduls EM 253

Allgemein	6ES7 253-1AA22-0XA0
Eingangsfunktionen	
Anzahl Eingänge	5 Eingänge
Eingangstyp	Stromziehend/stromliefernd (IEC Typ 1, stromziehend, bis auf ZP)
Eingangsspannung	
Maximal zulässige Dauerspannung	
STP, RPS, LMT+, LMT-	30 V DC
ZP	Max. 30 V DC bei 20 mA
Stoß (alle Eingänge)	35 V DC für 0,5 s
Nennwert	
STP, RPS, LMT+, LMT-	24-V-DC-Nennspannung bei 4 mA
ZP	24-V-DC-Nennspannung bei 15 mA
Signal logisch 1 (min.)	
STP, RPS, LMT+, LMT-	Min. 15 V DC bei 2,5 mA
ZP	Min. 3 V DC bei 8,0 mA
Signal logisch 0 (max.)	
STP, RPS, LMT+, LMT-	Max. 5 V DC bei 1 mA
ZP	Max. 1 V DC bei 1 mA
Elektrische Trennung (Feld zu Logik)	
Galvanische Trennung	500 V AC für 1 Minute
Potentialgetrennte Gruppen	1 Eingang für STP, RPS und ZP 2 Eingänge für LMT+ und LMT-
Eingabeverzögerungszeiten	
STP, RPS, LMT+, LMT-	0,2 ms bis 12,8 ms, einstellbar
ZP (zählbare Pulsbreite)	Min. 2 µs
Anschluss 2-Draht-Näherungssensor (Bero)	
Zulässiger Kriechstrom	Max. 1 mA
Kabellänge	
Nicht geschirmt	
STP, RPS, LMT+, LMT-	30 m
ZP	Nicht empfehlenswert
Geschirmt	
STP, RPS, LMT+, LMT-	100 m
ZP	10 m
Anzahl gleichzeitig eingeschalteter Eingänge	
55 Grad C	5

Tabelle A-49 Technische Daten des Positioniermoduls EM 253, Fortsetzung

Allgemein	6ES7 253-1AA22-0XA0	
Ausgangsfunktionen		
Anzahl integrierter Ausgänge	6 Ausgänge (4 Signale)	
Ausgangstyp P0+, P0-, P1+, P1- P0, P1, DIS, CLR	 RS422/485-Treiber Open Drain	
Ausgangsspannung P0, P1, RS-422-Treiber, Differenzausgangsspannung Offener Stromkreis In Optokopplerdiode mit Reihenwiderstand von 200 Ω 100 Ω Last 54 Ω Last P0, P1, DIS, CLR Open Drain empfohlene Spannung, offener Stromkreis zulässige Spannung, offener Stromkreis Stromsenke Widerstand bei EIN Kriechstrom im ausgeschalteten Zustand, 30 V DC Interner Pull-up-Widerstand, Ausgangs-Drain nach T1	 Typ. 3,5 V Min. 2,8 V Min. 1,5 V Min. 1,0 V 5 V DC vom Modul 30 V DC[1] Max. 50 mA Max. 15 Ω Max. 10 µA 3,3 KΩ[2]	
Ausgangsstrom Anzahl Ausgangsgruppen Max. Anzahl eingeschalteter Ausgänge Kriechstrom pro Ausgang P0, P1, DIS, CLR Überlastschutz	 1 6 Max. 10 µA Nein	
Elektrische Trennung (Feld zu Logik) Galvanische Trennung	 500 V AC für 1 Minute	
Ausgabeverzögerung DIS, CLR: Aus - Ein / Ein - Aus	 Max. 30 µs	
Impulsverzerrung P0, P1, Ausgänge, RS-422-Treiber, 100 Ω externe Last Ausgänge P0, P1, Open Drain, 5 V / 470 Ω externe Last	 Max. 75 ns Max. 300 ns	
Schaltfrequenz P0+, P0-, P1+, P1-, P0 und P1	 200 kHz	
Kabellänge Nicht geschirmt Geschirmt	 Nicht empfohlen 10 m	
Spannungsversorgung		
L+ Versorgungsspannung	11 bis 30 V DC	
Versorgungsausgabe Logik	Max. +5 V DC +/- 10 %, 200 mA	
L+ Versorgungsstrom vs. 5-V-DC-Last Laststrom 0 mA (ohne Last) 200 mA (Nennlast)	 <u>12-V-DC-Eingang</u> 120 mA 300 mA	 <u>24-V-DC-Eingang</u> 70 mA 130 mA
Potentialtrennung L+ Spannung zu Logik L+ Spannung zu Eingängen L+ Spannung zu Ausgängen	 500 V AC für 1 Minute 500 V AC für 1 Minute Keine	
Verpolschutz	Der Eingang L+ und der Ausgang +5V sind durch Dioden geschützt. Wenn Sie an eine Klemme M eine positive Spannung mit Bezug auf die Ausgangsanschlüsse anlegen, kann dies zu möglicherweise schädlichem Stromfluss führen.	

[1] Durch den Betrieb der Open-Drain-Ausgänge mit über 5 V DC können sich die Hochfrequenzaussendungen auf Werte erhöhen, die die Grenzwerte überschreiten. Dadurch können in Ihrem System bzw. für die Verdrahtung Hochfrequenz-Sicherheitmaßnahmen erforderlich sein.

[2] Je nach Impulsempfänger und -kabel kann durch einen zusätzlichen externen Pull-up-Widerstand die Signalqualität der Impulse und die Störfestigkeit verbessert werden.

S7-200 CPUs, die intelligente Module unterstützen

Das Positioniermodul EM 253 ist ein intelligentes Erweiterungsmodul und kann zusammen mit den in Tabelle A-50 aufgeführten S7-200 CPUs eingesetzt werden.

Tabelle A-50 Kompatibilität des Positioniermoduls EM 253 mit S7-200 CPUs

CPU	Beschreibung
CPU 222 ab Ausgabestand 1.10	CPU 222 DC/DC/DC und CPU 222 AC/DC/Relais
CPU 224 ab Ausgabestand 1.10	CPU 224 DC/DC/DC und CPU 224 AC/DC/Relais
CPU 226 ab Ausgabestand 1.00	CPU 226 DC/DC/DC und CPU 226 AC/DC/Relais
CPU 226XM ab Ausgabestand 1.00	CPU 226XM DC/DC/DC und CPU 226XM AC/DC/Relais

Status-LEDs des Positioniermoduls EM 253

Die Status-LEDs des Positioniermoduls werden in Tabelle A-51 aufgeführt.

Tabelle A-51 Status-LEDs des Positioniermoduls

Integrierte E/A	LED	Farbe	Funktionsbeschreibung
–	MF	Rot	Leuchtet, wenn das Modul einen schweren Fehler erkennt.
–	MG	Grün	Leuchtet, wenn kein Modulfehler vorliegt, und blinkt mit einer Frequenz von 1 Hz, wenn ein Konfigurationsfehler erkannt wurde.
–	PWR	Grün	Leuchtet, wenn an den Klemmen L+ und M des Moduls eine Spannung von 24 V DC anliegt.
Eingang	STP	Grün	Leuchtet, wenn der Stopp-Eingang eingeschaltet ist.
Eingang	RPS	Grün	Leuchtet, wenn der Referenzpunkteingang eingeschaltet ist.
Eingang	ZP	Grün	Leuchtet, wenn der Nullimpulseingang eingeschaltet ist.
Eingang	LMT-	Grün	Leuchtet, wenn der Eingang für den negativen Grenzwert eingeschaltet ist.
Eingang	LMT+	Grün	Leuchtet, wenn der Eingang für den positiven Grenzwert eingeschaltet ist.
Ausgang	P0	Grün	Leuchtet, wenn der Ausgang P0 im Impulsbetrieb ist.
Ausgang	P1	Grün	Leuchtet, wenn der Ausgang P1 im Impulsbetrieb ist oder wenn dieser Ausgang positive Bewegungen anzeigt.
Ausgang	DIS	Grün	Leuchtet, wenn der Ausgang DIS aktiv ist.
Ausgang	CLR	Grün	Leuchtet, wenn der Ausgang zum Löschen des Abweichungszählers aktiv ist.

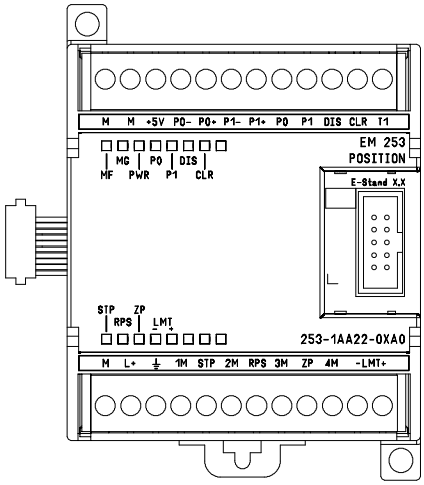


Bild A-30 Positioniermodul EM 253

A

Schaltpläne

In der folgenden Abbildung sind die Klemmen nicht in der richtigen Reihenfolge. Die Anordnung der Klemmen finden Sie in Bild A-30.

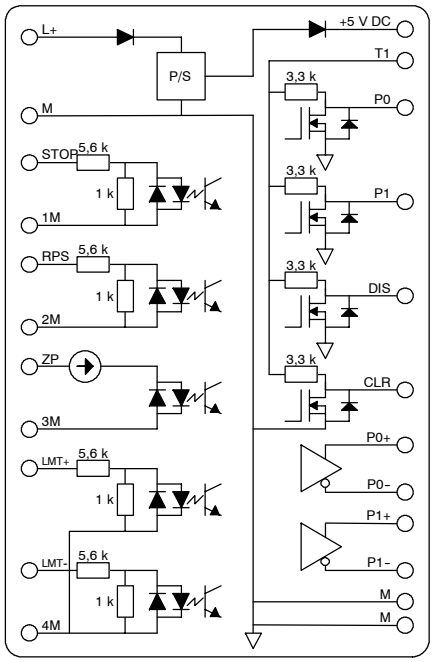
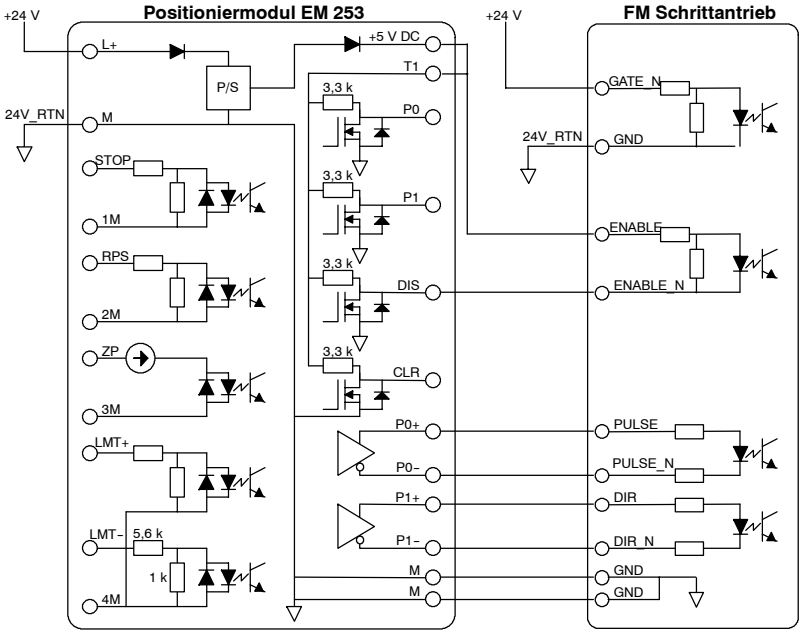


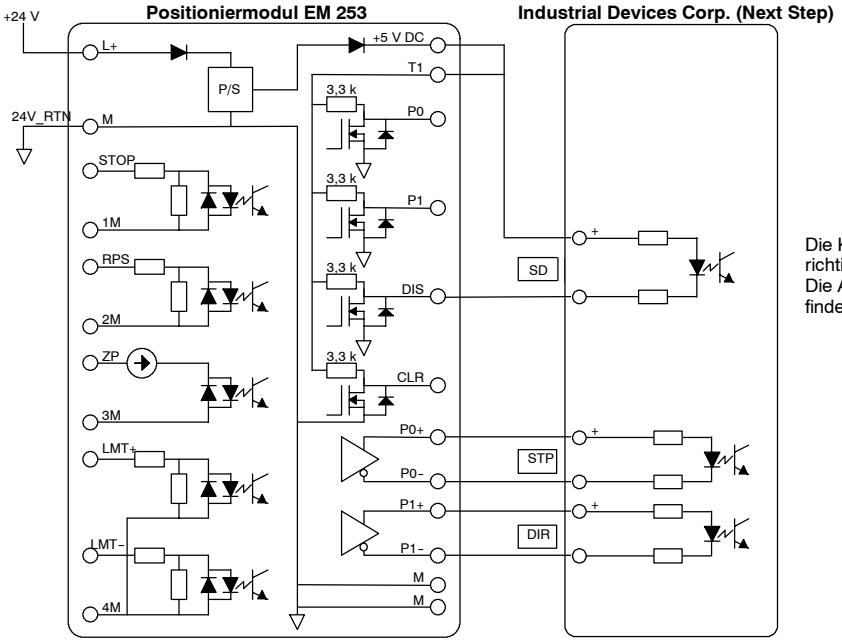
Bild A-31 Schaltplan der Eingänge und Ausgänge des Positioniermoduls EM 253



Die Klemmen sind nicht in der richtigen Reihenfolge.
Die Anordnung der Klemmen finden Sie in Bild A-30.

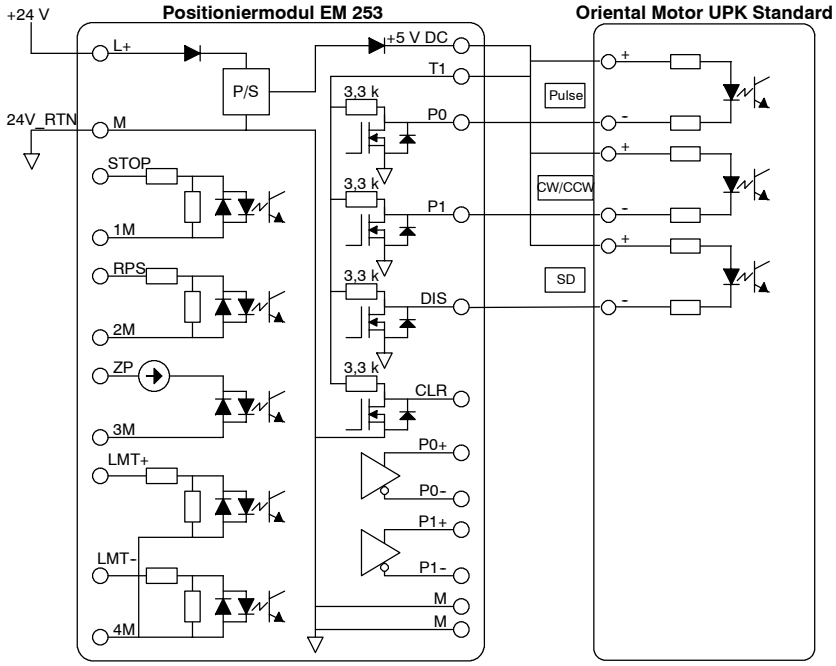
Bild A-32 Anschließen eines Positioniermodul EM 253 an einen SIMATIC FM Schrittantrieb

A



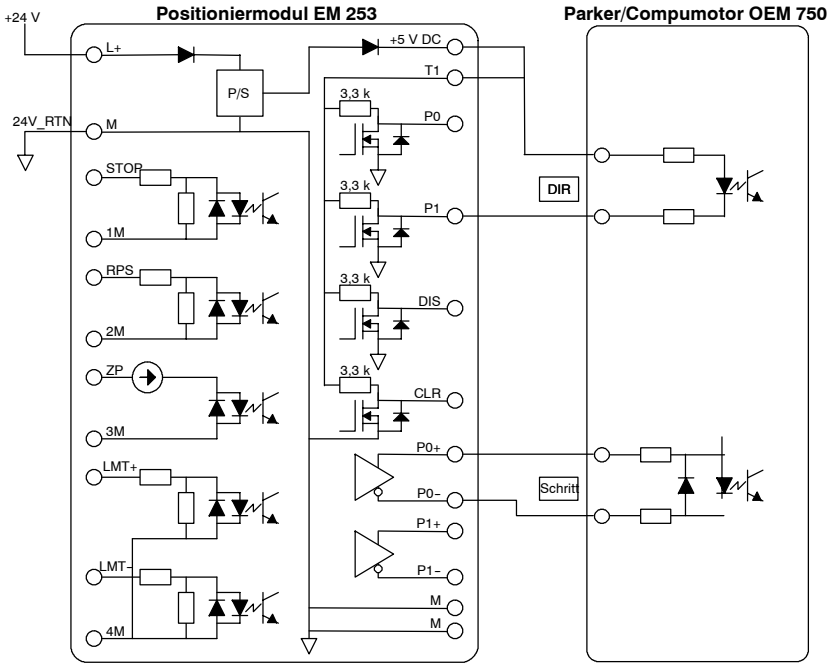
Die Klemmen sind nicht in der richtigen Reihenfolge.
Die Anordnung der Klemmen finden Sie in Bild A-30.

Bild A-33 Anschließen eines Positioniermoduls EM 253 an ein Industrial Devices Corp. (Next Step)



Die Klemmen sind nicht in der richtigen Reihenfolge.
Die Anordnung der Klemmen finden Sie in Bild A-30.

Bild A-34 Anschließen eines Positioniermoduls EM 253 an ein Oriental Motor UPK Standard



Die Klemmen sind nicht in der richtigen Reihenfolge.
Die Anordnung der Klemmen finden Sie in Bild A-30.

Bild A-35 Anschließen eines Positioniermoduls EM 253 an ein Parker/Compumotor OEM 750

A

Technische Daten des Ethernet-Moduls (CP 243-1)

Tabelle A-52 Bestellnummer Ethernet-Modul (CP 243-1)

Bestellnummer	Erweiterungsmodul	EM-Eingänge	EM-Ausgänge	Steckbarer Klemmenblock
6GK7 243-1EX00-OXE0	Ethernet-Modul (CP 243-1)	–	8[1]	Nein

[1] Acht Ausgänge werden zur logischen Steuerung der Ethernetfunktion verwendet, sie steuern nicht direkt externe Signale.

Tabelle A-53 Allgemeine technische Daten des Ethernet-Moduls (CP 243-1)

Bestellnummer	Name und Beschreibung des Moduls	Abmessungen (mm) (B x H x T)	Gewicht	Verlust-leistung	V-DC-Bedarf +5 V DC	V-DC-Bedarf Aus AS-Interface
6GK7 243-1EX00-OXE0	Ethernet-Modul (CP 243-1)	71,2 x 80 x 62	ca. 150 g	1,75 W	55 mA	60 mA

Tabelle A-54 Technische Daten des Ethernet-Moduls (CP 243-1)

Allgemein	6GK7 243-1EX00-0XE0
Übertragungsgeschwindigkeit	10 MBits/s und 100 MBits/s
Größe Flash-Speicher	1 MB
Größe SDRAM-Speicher	8 MB
Schnittstellen verbindung zu Industrial Ethernet (10/100 MBit/s)	 8polige RJ45-Buchse
Eingangsspannung	20,4 bis 28,8 V DC
Maximale Anzahl Verbindungen	Maximal 8 S7-Verbindungen (XPUT/XGET und READ/WRITE) plus 1 Verbindung zu STEP 7-Micro/WIN je Ethernet-Modul (CP 243-1)[2]
Startzeit bzw. Neustartzeit nach Rücksetzen	ca. 10 Sekunden
Quantität Anwenderdaten	Als Client: max. 212 Bytes für XPUT/XGET Als Server: max. 222 Byte für XGET oder READ max. 212 Byte für XPUT oder WRITE

[2] Es darf nur ein Ethernet-Modul (CP 243-1) je S7-200 CPU angeschlossen werden.

Das Ethernet-Modul (CP 243-1) ist ein Kommunikationsprozessor, mit dem das S7-200 System an Industrial Ethernet (IE) angeschlossen wird. Die S7-200 kann mit STEP 7-Micro/WIN über Ethernet entfernt konfiguriert, programmiert und diagnostiziert werden. Die S7-200 kann über Ethernet mit anderen Steuerungen vom Typ S7-200, S7-300 oder S7-400 kommunizieren. Sie kann auch mit einem OPC-Server kommunizieren.

Industrial Ethernet ist für die Industrie ausgelegt. Es kann entweder mit rauschfreier ITP-Technologie (Industrial Twisted Pair) oder mit Industrie-Standard-Technologie TP (Twisted Pair) verwendet werden. Industrial Ethernet kann implementiert werden, um eine große Bandbreite anwendungsspezifischer Einsatzzwecke zu bieten, z.B. Schaltung, Hochgeschwindigkeitsredundanz, Schnellverbindungen und redundante Netze. Durch das Ethernet-Modul (CP 243-1) wird das Zielsystem S7-200 kompatibel mit einer Vielzahl vorhandener Produkte, die Ethernet unterstützen.

S7-200 CPUs, die intelligente Module unterstützen

Das Ethernet-Modul (CP 243-1) ist ein intelligentes Erweiterungsmodul und kann zusammen mit den in Tabelle A-45 aufgeführten S7-200 CPUs eingesetzt werden.

Tabelle A-55 Kompatibilität des Ethernet-Moduls (CP 243-1) mit S7-200 CPUs

CPU	Beschreibung
CPU 222 ab Ausgabestand 1.10	CPU 222 DC/DC/DC und CPU 222 AC/DC/Relais
CPU 224 ab Ausgabestand 1.10	CPU 224 DC/DC/DC und CPU 224 AC/DC/Relais
CPU 226 ab Ausgabestand 1.00	CPU 226 DC/DC/DC und CPU 226 AC/DC/Relais
CPU 226XM ab Ausgabestand 1.00	CPU 226XM DC/DC/DC und CPU 226XM AC/DC/Relais

Das Ethernet-Modul (CP 243-1) wird mit einer voreingestellten, eindeutigen, weltweiten MAC-Adresse ausgeliefert, die nicht geändert werden kann.

Funktionen

Das Ethernet-Modul (CP 243-1) bearbeitet Datenverkehr unabhängig über das Industrial Ethernet.

- ❑ Die Kommunikation basiert auf TCP/IP.
- ❑ Für die Kommunikation zwischen S7-200 CPUs und anderen S7-Steuerungssystemen bzw. PCs über Ethernet stehen Kommunikationsdienste als Client und Server zur Verfügung. Es können bis zu acht Verbindungen betrieben werden.
- ❑ Die Implementierung von PC-Anwendungen ist durch Integration des S7-OPC-Servers möglich.
- ❑ Das Ethernet-Modul (CP 243-1) ermöglicht der S7-200 Programmiersoftware STEP 7-Micro/WIN direkten Zugriff auf die S7-200 über Ethernet.

Ethernet Assistent

Konfiguration

Mit dem Ethernet-Assistenten in STEP 7-Micro/WIN können Sie das Ethernet-Modul (CP 243-1) für den Anschluss eines S7-200 Zielsystems über ein Ethernet-Netz einrichten. Mit dem Ethernet-Assistenten können Sie die Parameter für das Ethernet-Modul (CP 243-1) definieren und die Konfigurationsoperationen in Ihrem Projektordner ablegen. Zum Aufrufen des Ethernet-Assistenten wählen Sie den Menübefehl **Extras > Ethernet-Assistent**. Der Assistent benötigt die folgenden Angaben: IP-Adresse, Subnetzmaske, Gateway-Adresse und Art der Kommunikationsverbindung.

Anschlüsse

Das Ethernet-Modul (CP 243-1) verfügt über die folgenden Anschlüsse. Die Anschlüsse befinden sich hinter den Fronttüren.

- ❑ Klemmenblock für die 24-V-DC-Spannungsversorgung und den Erdanschluss
- ❑ 8polige RJ45-Buchse für den Ethernet-Anschluss
- ❑ Steckverbinder für E/A-Bus
- ❑ Integriertes Flachkabel mit Buchse für E/A-Bus

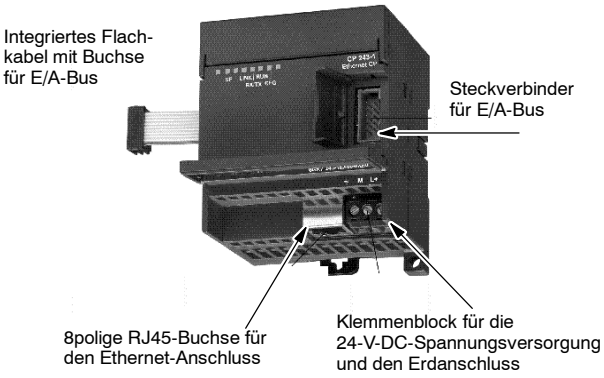


Bild A-36 Anschließen des Ethernet-Moduls (CP 243-1)

Weitere Informationen

Ausführliche Informationen zum Ethernet-Modul (CP 243-1) finden Sie im Technischen Handbuch *SIMATIC NET CP 243-1 Kommunikationsprozessor für Industrial Ethernet.*

A

Technische Daten des Internet-Moduls (CP 243-1 IT)

Tabelle A-56 Bestellnummer Internet-Modul (CP 243-1 IT)

Bestellnummer	Erweiterungsmodul	EM-Eingänge	EM-Ausgänge	Steckbarer Klemmenblock
6GK7 243-1GX00-OXE0	Internet-Modul (CP 243-1 IT)	–	8[1]	Nein

[1] Acht Ausgänge werden zur logischen Steuerung der IT-Funktion verwendet, sie steuern nicht direkt externe Signale.

Tabelle A-57 Allgemeine technische Daten des Internet-Moduls (CP 243-1 IT)

Bestellnummer	Name und Beschreibung des Moduls	Abmessungen (mm) (B x H x T)	Gewicht	Verlust-leistung	V-DC-Bedarf +5 V DC	V-DC-Bedarf Aus AS-Interface
6GK7 243-1GX00-OXE0	Internet-Modul (CP 243-1 IT)	71,2 x 80 x 62	ca. 150 g	1,75 W	55 mA	60 mA

Tabelle A-58 Technische Daten des Internet-Moduls (CP 243-1 IT)

Allgemein	6GK7 243-1GX00-0XE0
Übertragungsgeschwindigkeit	10 MBits/s und 100 MBits/s
Größe Flash-Speicher	8 MB als ROM für die Firmware des Internet-Moduls (CP 243-1 IT), 8 MB als RAM für das Dateisystem
Größe SDRAM-Speicher	16 MB
Garantierte Lebensdauer des Flash-Speichers für das Dateisystem	1 Mio. Schreib- oder Löschoperationen
Schnittstellen verbindung zu Industrial Ethernet (10/100 MBit/s)	 8polige RJ45-Buchse
Eingangsspannung	20,4 bis 28,8 V DC
Maximale Anzahl Verbindungen	Maximal 8 S7-Verbindungen (XPUT/XGET und READ/WRITE) plus 1 Verbindung zu STEP 7-Micro/WIN je Internet-Modul (CP 243-1 IT)[2]
Maximale Anzahl IT-Verbindungen	1 für FTP-Server 1 für FTP-Client 1 für E-Mail-Client 4 für HTTP-Verbindungen
Startzeit bzw. Neustartzeit nach Rücksetzen	ca. 10 Sekunden
Quantität Anwenderdaten	Client: max. 212 Bytes für XPUT/XGET Server: max. 222 Byte für XGET oder READ max. 212 Byte für XPUT oder WRITE
Max. E-Mail-Größe	1024 Zeichen
Dateisystem: Pfadlänge einschließlich Dateigröße und Laufwerksnamen Länge Dateiname Schachtelungstiefe Verzeichnisse	 Max. 254 Zeichen Max. 99 Zeichen Max. 49
Verfügbare Server-Schnittstellen: HTTP FTP-Befehlskanal FTP-Datenkanäle für FTP-Server S7-Verbindungsaufbau S7-Server	 80 21 3100 bis 3199 102 3000 bis 3008

[2] Es darf nur ein Internet-Modul (CP 243-1 IT) je S7-200 CPU angeschlossen werden.

Das Internet-Modul (CP 243-1 IT) ist ein Kommunikationsprozessor, mit dem das S7-200 System an Industrial Ethernet (IE) angeschlossen wird. Die S7-200 kann mit STEP 7-Micro/WIN über Ethernet entfernt konfiguriert, programmiert und diagnostiziert werden. Die S7-200 kann über Ethernet mit anderen Steuerungen vom Typ S7-200, S7-300 oder S7-400 kommunizieren. Sie kann auch mit einem OPC-Server kommunizieren.

Die IT-Funktionen des Internet-Moduls (CP 243-1 IT) bilden die Basis für die Überwachung und bei Bedarf auch für die Bearbeitung von Automatisierungssystemen mit einem Web-Browser über einen vernetzten PC. Diagnosemeldungen können von einem System per E-Mail versendet werden. Mit den IT-Funktionen ist es einfach, komplette Dateien mit anderen Computer- und Steuerungssystemen auszutauschen.

Industrial Ethernet ist das Netz für die Prozesssteuerungsebene und die Zellebene des offenen Kommunikationssystems SIMATIC NET. Physikalisch ist Industrial Ethernet ein elektrisches Netz, das auf geschirmten, koaxialen Leitungen, verdrillten Doppelleitungen und einem optischen Netz aus Lichtwellenleitern basiert. Industrial Ethernet wird von der Internationalen Norm IEEE 802.3 definiert.

S7-200 CPUs, die intelligente Module unterstützen

Das Internet-Modul (CP 243-1 IT) ist ein intelligentes Erweiterungsmodul und kann zusammen mit den in Tabelle A-45 aufgeführten S7-200 CPUs eingesetzt werden.

Tabelle A-59 Kompatibilität des Internet-Moduls (CP 243-1 IT) mit S7-200 CPUs

CPU	Beschreibung
CPU 222 ab Ausgabestand 1.10	CPU 222 DC/DC/DC und CPU 222 AC/DC/Relais
CPU 224 ab Ausgabestand 1.10	CPU 224 DC/DC/DC und CPU 224 AC/DC/Relais
CPU 226 ab Ausgabestand 1.00	CPU 226 DC/DC/DC und CPU 226 AC/DC/Relais
CPU 226XM ab Ausgabestand 1.00	CPU 226XM DC/DC/DC und CPU 226XM AC/DC/Relais

Das Internet-Modul (CP 243-1 IT) verfügt über die folgenden Leistungsmerkmale:

- Das Internet-Modul (CP 243-1 IT) ist vollständig mit dem Ethernet-Modul (CP 243-1) kompatibel. Anwenderprogramme, die für das Ethernet-Modul (CP 243-1) geschrieben wurden, können auch auf dem Internet-Modul (CP 243-1 IT) ausgeführt werden.

Das Internet-Modul (CP 243-1 IT) wird mit einer voreingestellten, eindeutigen, weltweiten MAC-Adresse ausgeliefert, die nicht geändert werden kann.

Tipp

Es darf nur ein Internet-Modul (CP 243-1 IT) je S7-200 CPU angeschlossen werden. Werden mehrere Internet-Module (CP 243-1 IT) angeschlossen, funktioniert die S7-200 CPU möglicherweise nicht ordnungsgemäß.

A

Funktionen

Das Internet-Modul (CP 243-1 IT) bietet die folgenden Funktionen:

- Die S7-Kommunikation basiert auf TCP/IP.
- IT-Kommunikation
- Konfiguration
- Zeitüberwachung
- Voreingestellte MAC-Adressen (48-Bit-Werte) können adressiert werden.

Internet Assistent

Konfiguration

Mit dem Internet-Assistenten in STEP 7-Micro/WIN können Sie das Internet-Modul (CP 243-1 IT) für den Anschluss eines S7-200 Zielsystems über ein Ethernet/Internet-Netz einrichten. Das Internet-Modul (CP 243-1 IT) verfügt über zusätzliche Web-Server-Funktionalität, die mit dem Internet-Assistenten eingerichtet werden kann. Zum Aufrufen des Internet-Assistenten wählen Sie den Menübefehl **Extras > Internet-Assistent**.

Anschlüsse

Das Internet-Modul (CP 243-1 IT) verfügt über die folgenden Anschlüsse. Die Anschlüsse befinden sich hinter den Fronttüren.

- ❑ Klemmenblock für die 24-V-DC-Spannungsversorgung und den Erdanschluss
- ❑ 8polige RJ45-Buchse für den Ethernet-Anschluss
- ❑ Steckverbinder für E/A-Bus
- ❑ Integriertes Flachkabel mit Buchse für E/A-Bus

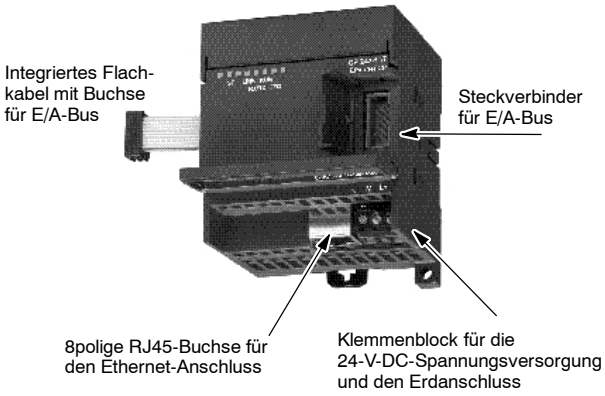


Bild A-37 Anschließen des Internet-Moduls (CP 243-1 IT)

Weitere Informationen

Ausführliche Informationen zum Internet-Modul (CP 243-1 IT) finden Sie im Technischen Handbuch *SIMATIC NET CP 243-1 IT Kommunikationsprozessor für Industrial Ethernet und Informationstechnologie.*

A

Technische Daten des AS-Interface-Moduls (CP 243-2)

Tabelle A-60 Bestellnummer des AS-Interface-Moduls (CP 243-2)

Bestellnummer	Erweiterungsmodule	EM-Eingänge	EM-Ausgänge	Steckbarer Klemmenblock
6GK7 243-2AX01-0XA0	AS-Interface-Modul (CP 243-2)	8 digital und 8 analog	8 digital und 8 analog	Ja

Tabelle A-61 Allgemeine technische Daten des AS-Interface-Moduls (CP 243-2)

Bestellnummer	Name und Beschreibung des Moduls	Abmessungen (mm) (B x H x T)	Gewicht	Verlust-leistung	V-DC-Bedarf +5 V DC	V-DC-Bedarf Aus AS-Interface
6GK7 243-2AX01-0XA0	AS-Interface-Modul (CP 243-2)	71 x 80 x 62	ca. 250 g	3,7 W	220 mA	100 mA

Tabelle A-62 Technische Daten des AS-Interface-Moduls (CP 243-2)

Allgemein	6GK7 243-2AX01-0XA0
Zykluszeit	5 ms mit 31 Slaves 10 ms mit 62 AS-I-Slaves und erweiterter Adressierung
Konfiguration	Schalter auf Vorderseite verwenden oder Befehl für Gesamtkonfiguration verwenden (siehe Beschreibung der Befehle für das AS-I im Handbuch *CP 243-2 AS-I Interface-Master*)
Unterstützte AS-i-Master-Profile	M1e
Anschluss an AS-i-Kabel	Über einen S7-200 Klemmenblock. Zulässige Stromlast von Klemmen 1 bis 3 oder von Klemmen 2 bis 4 beträgt maximal 3 A.
Adressbereich	Ein Digitalmodul mit 8 Digitaleingängen und 8 Digitalausgängen und Ein Analogmodul mit 8 Analogeingängen und 8 Analogausgängen.

Leistungsmerkmale

Sie können bis zu zwei AS-Interface-Module gleichzeitig auf der S7-200 betreiben, wodurch sich die Anzahl der verfügbaren digitalen und analogen Eingänge und Ausgänge erheblich erhöht (maximal 124 Digitaleingänge /
124 Digitalausgänge am AS-Interface je CP). Die Einrichtungszeiten werden durch Konfiguration per Knopfdruck verkürzt. Die LEDs verkürzen die Ausfallzeiten bei Auftreten von Fehlern, weil der Status des CP und aller angeschlossenen Slaves angezeigt wird und weil die Hauptspannung des AS-Interface überwacht wird.

Das AS-Interface-Modul verfügt über die folgenden Leistungsmerkmale:

- Es unterstützt Analogmodule.
- Es unterstützt alle Masterfunktionen und ermöglicht den Anschluss von bis zu 62 AS-Interface-Slaves.
- Die LEDs auf der Vorderseite zeigen den Betriebszustand und die Verfügbarkeit der angeschlossenen Slaves an.
- LEDs auf der Vorderseite zeigen Fehler an (einschließlich Spannungsfehler, Konfigurationsfehler des AS-Interface).
- Zwei Klemmen für direkten Anschluss des AS-Interface-Kabels.
- Zwei Schaltflächen zum Anzeigen von Statusinformationen der Slaves, zum Umschalten des Betriebszustands und zum Übernehmen der vorhandenen Konfiguration als Konfiguration SET.

AS-i Assistent

Mit dem AS-i-Assistenten in STEP 7-Micro/WIN können Sie das AS-Interface-Modul (CP 243-2) konfigurieren. Mit dem AS-i-Assistenten können Sie die Daten aus einem AS-Interface-Netz in Ihre Konfiguration aufnehmen. Zum Aufrufen des AS-i-Assistenten wählen Sie den Menübefehl **Extras > AS-i-Assistent**.

Betrieb

Im Prozessabbild der S7-200 belegt das AS-Interface-Modul ein digitales Eingangsbyte (Statusbyte), ein digitales Ausgangsbyte (Steuerbyte) und 8 analoge Eingangswörter sowie 8 analoge Ausgangswörter. Das AS-Interface-Modul nutzt zwei logische Modulpositionen. Mit dem Status- und dem Steuerbyte kann der Betriebszustand des AS-Interface-Moduls über ein Anwenderprogramm eingestellt werden. Je nach Betriebszustand speichert das AS-Interface die E/A-Daten des AS-Interface-Slave oder Diagnosewerte oder es aktiviert Aufrufe vom Master (z.B. Ändern einer Slave-Adresse) im analogen Adressbereich der S7-200.

Alle angeschlossenen AS-Interface Slaves können per Knopfdruck konfiguriert werden. Eine weitere Konfiguration des CP ist nicht erforderlich.

Vorsicht

Wenn Sie das AS-Interface-Modul einsetzen, müssen Sie die Analogeingabefilter in der CPU ausschalten.

Deaktivieren Sie die Analogfilter in der CPU nicht, werden die digitalen Daten zerstört und Fehlerbedingungen werden nicht als Bitwerte im Analogwort ausgegeben.

Achten Sie darauf, dass die Analogfilter in der CPU deaktiviert sind.

Funktionen

Der CP 243-2 ist AS-Interface Master der Masterklasse M1e. Dies bedeutet, dass alle angegebenen Funktionen unterstützt werden. Deshalb können mittels doppelter Adresszuordnung (A-B) maximal 31 digitale Slaves auf dem AS-Interface betrieben werden. Der CP 243-2 verfügt über zwei Betriebsarten:

- Standardmodus: Zugriff auf die E/A-Daten des AS-Interface Slave
- Erweiterter Modus: Aufrufe des Master (z.B. Schreiben von Parametern) oder Anforderungen von Diagnosewerten

Anschlüsse

Das AS-Interface-Modul verfügt über die folgenden Anschlüsse:

- Zwei Anschlüsse ans AS-Interface-Modulkabel (intern überbrückt)
- Einen Anschluss an Funktionserde

Die Klemmen befinden sich unter der Abdeckung auf der Vorderseite (siehe Bild A-38).

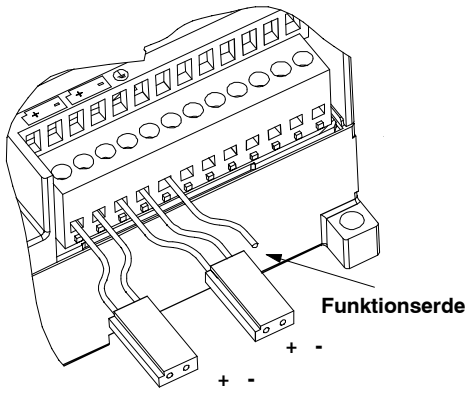


Bild A-38 Anschließen des AS-Interface-Modulkabels

Vorsicht

Die Ladekapazität der Kontakte des AS-Interface-Moduls beträgt maximal 3 A. Wird dieser Wert am AS-Interface-Modulkabel überschritten, darf das AS-Interface nicht in einer Schleife ans AS-I-Kabel angeschlossen werden, sondern muss über ein getrenntes Kabel angeschlossenen werden (in diesem Fall
wird nur ein Klemmenpaar des AS-Interface-Moduls verwendet). Das AS-Interface-Modul muss über die Erdungsklemme an den Erdleiter angeschlossen werden.

Tipp

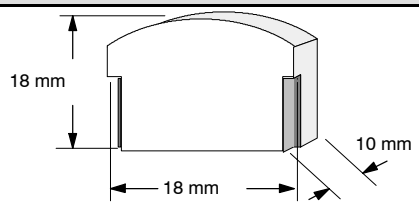
Das AS-Interface-Modul hat einen Anschluss für Funktionserde. Dieser Anschluss muss mit dem kleinstmöglichen Widerstand an den PE-Leiter angeschlossen werden.

Weitere Informationen

Weitere Informationen zum CP 243-2 AS-Interface-Master finden Sie im Handbuch *SIMATIC NET CP 243-2 AS-Interface-Master.*

Optionale Steckmodule

Steckmodul	Beschreibung	Bestellnummer
Speichermodul	Speicher Speichermodul: Programm, Daten und Konfiguration	6ES7 291-8GE20-0XA0
Echtzeituhr mit Batterie	Genauigkeit Uhrmodul:2 Minuten/Monat bei 25 °C, 7 Minuten/Monat bei 0 °C bis 55 °C	6ES7 297-1AA20-0XA0
Batteriemodul	Batteriemodul (Speicherzeit): Typ. 200 Tage	6ES7 291-8BA20-0XA0

Allgemein		Abmessungen
Batterie	3 V, 30 mA Hr, Renata CR 1025	
Größe	9,9 mm x 2,5 mm	
Datentyp	Lithium < 0,6 g	

Das Speichermodul speichert das gesamte Programm und den Datenbaustein für alle CPUs (CPU 221, CPU 222, CPU 224, CPU 226 und CPU 226XM).

Steckleitung für Erweiterungsmodule

Allgemeine Leistungsmerkmale (6ES7 290-6AA20-0XA0)	
Kabellänge	0,8 m
Gewicht	25 g
Anschlussart	10poliges Flachkabel

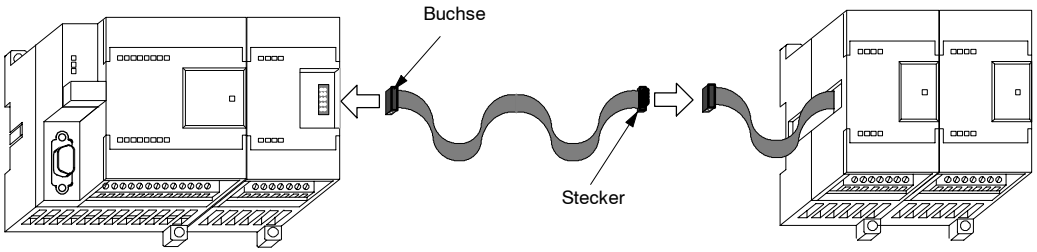


Bild A-39 Typischer Einbau der Steckleitung für Erweiterungsmodule

Tipp
Bei mehrzeiligem Aufbau dürfen Sie maximal eine Steckleitung pro CPU-/Erweiterungsmodulkette verwenden.

RS-232/PPI-Multi-Master-Kabel und USB/PPI-Multi-Master-Kabel

Tabelle A-63 Technische Daten des RS-232/PPI-Multi-Master-Kabels und des USB/PPI-Multi-Master-Kabels

Beschreibung **Bestellnummer**	**S7-200 RS-232/PPI-Multi-Master-Kabel** **6ES7 901-3CB30-0XA0**	**S7-200 USB/PPI-Multi-Master-Kabel** **6ES7-901-3DB30-0XA0**
Allgemeine Eigenschaften		
Versorgungsspannung	14,4 bis 28,8 V DC	14,4 bis 28,8 V DC
Versorgungsstrom bei 24-V-Nennspannung	Max. 60 mA Effektivwert	Max. 50 mA Effektivwert
Verzögerung Richtungsänderung: RS-232 Stoppbit-Flanke empfangen zu RS-485 Übertragung deaktiviert	-	-
Trennung	RS-485 zu RS-232: 500 V DC	RS-485 zu USB: 500 V DC
Elektrische Eigenschaften RS-485-Seite		
Gleichtaktspannungsbereich	-7 V bis +12 V, 1 s, V fortlaufender Effektivwert	-7 V bis +12 V, 1 s, V fortlaufender Effektivwert
Eingangsimpedanz Empfänger	Min. 5,4 kΩ einschließlich Abschluss	Min. 5,4 kΩ einschließlich Abschluss
Abschluss/Bias	10 kΩ bis +5 V auf B, PROFIBUS-Pin 3 10 kΩ bis Erde auf A, PROFIBUS-Pin 8	10 kΩ bis +5 V auf B, PROFIBUS-Pin 3 10 kΩ bis Erde auf A, PROFIBUS-Pin 8
Ansprechgrenze/Sensibilität Empfänger	+/- 0,2 V, 60 mV typ. Hysterese	+/- 0,2 V, 60 mV typ. Hysterese
Differentialausgangsspannung Sender	Min. 2 V bei R_L = 100 Ω, min. 1,5 V bei R_L = 54 Ω	Min. 2 V bei R_L = 100 Ω, min. 1,5 V bei R_L = 54 Ω
Elektrische Eigenschaften RS-232-Seite		
Eingangsimpedanz Empfänger	Min. 3 kΩ	-
Ansprechgrenze/Sensibilität Empfänger	Min. 0,8 V Low, max. 2,4 V High, typ. Hysterese 0,5 V	-
Ausgangsspannung Sender	Min. +/- 5 V bei R_L = 3 kΩ	-
Elektrische Eigenschaften USB-Seite		
Volle Geschwindigkeit (12 MB/s), Human Interface Device (HID)		
Versorgungsspannung 5 V	-	Max. 50 mA
Strom im ausgeschalteten Zustand	-	Max. 400 uA

Leistungsmerkmale

Das S7-200 RS-232/PPI-Multi-Master-Kabel ist werkseitig für optimale Leistungsfähigkeit mit dem Programmierpaket STEP 7-Micro/WIN 3.2 Service Pack 4 (oder höher) eingestellt. Die werkseitigen Voreinstellungen dieses Kabels unterscheiden sich von denen der PC/PPI-Kabel. Weitere Informationen zum Konfigurieren des Kabels für Ihre Anwendung finden Sie in Bild 1.

Sie können das S7-200 RS-232/PPI-Multi-Master-Kabel so einrichten, dass es wie das PC/PPI-Kabel betrieben werden kann und dass es mit allen Versionen des Programmierpakets STEP 7-Micro/WIN kompatibel ist. Hierzu wählen Sie für Schalter 5 die Einstellung PPI/frei programmierbare Kommunikation und stellen dann die erforderliche Baudrate ein.

Für den Betrieb des USB-Kabels wird das Programmierpaket STEP 7-Micro/WIN 3.2 Service Pack 4 (oder höher) benötigt.

Tipp

Informationen zum PC/PPI-Kabel finden Sie in Ausgabe 3 des *S7-200 Systemhandbuchs*, Bestellnummer (6ES7 298-8FA22-8BH0).

S7-200 RS-232/PPI-Multi-Master-Kabel

Tabelle A-64 S7-200 RS-232/PPI-Multi-Master-Kabel - Anschlussbelegung für RS-485-Steckverbinder zu lokalem RS-232-Steckverbinder

Anschlussbelegung RS-485-Steckverbinder		Anschlussbelegung lokaler RS-232-Steckverbinder	
Pin	**Signalbeschreibung**	**Pin**	**Signalbeschreibung**
1	Kein Anschluss	1	Erkennung Data Carrier (DCD) (nicht verwendet)
2	24-V-Rückleitung (RS-485 logische Erde)	2	Daten empfangen (RD) (Ausgang des PC/PPI-Kabels)
3	Signal B (RxD/TxD+)	3	Daten senden (TD) (Eingang des PC/PPI-Kabels)
4	RTS (TTL-Level)	4	Datenterminal bereit (DTR)[1]
5	Kein Anschluss	5	Erde (RS-232 logische Erde)
6	Kein Anschluss	6	Betriebsbereitschaft (DSR)[1]
7	24-V-Spannungsversorgung	7	Anforderung zum Senden (RTS) (nicht verwendet)
8	Signal A (RxD/TxD-)	8	Bereit zum Senden (CTS) (nicht verwendet)
9	Protokollauswahl	9	Ringanzeige (RI) (nicht verwendet)

[1] Die Pins 4 und 6 sind intern angeschlossen.

Tabelle A-65 S7-200 RS-232/PPI-Multi-Master-Kabel - Anschlussbelegung für RS-485-Steckverbinder zu entferntem RS-232-Steckverbinder

Anschlussbelegung RS-485-Steckverbinder		Anschlussbelegung entfernter RS-232-Steckverbinder[1]	
Pin	**Signalbeschreibung**	**Pin**	**Signalbeschreibung**
1	Kein Anschluss	1	Erkennung Data Carrier (DCD) (nicht verwendet)
2	24-V-Rückleitung (RS-485 logische Erde)	2	Daten empfangen (RD) (Eingang des PC/PPI-Kabels)
3	Signal B (RxD/TxD+)	3	Daten senden (TD) (Ausgang des PC/PPI-Kabels)
4	RTS (TTL-Level)	4	Datenterminal bereit (DTR)[2]
5	Kein Anschluss	5	Erde (RS-232 logische Erde)
6	Kein Anschluss	6	Betriebsbereitschaft (DSR)[2]
7	24-V-Spannungsversorgung	7	Sendeanforderung (RTS) (Ausgang PC/PPI-Kabel)
8	Signal A (RxD/TxD-)	8	Bereit zum Senden (CTS) (nicht verwendet)
9	Protokollauswahl	9	Ringanzeige (RI) (nicht verwendet)

[1] Für Modems ist eine Wandlung Buchse/Stecker sowie 9polig/25polig erforderlich.
[2] Die Pins 4 und 6 sind intern angeschlossen.

A

Verwendung des S7-200 RS-232/PPI-Multi-Master-Kabels mit STEP 7-Micro/WIN als Ersatz für das PC/PPI-Kabel bzw. für den Betrieb in der frei programmierbaren Kommunikation

Bei direktem Anschluss an den PC

- ❏ Stellen Sie den Modus PPI/frei programmierbare Kommunikation ein (Schalter 5=0).
- ❏ Stellen Sie die Baudrate ein (Schalter 1, 2 und 3).
- ❏ Stellen Sie 'Lokal' ein (Schalter 6=0). Die Einstellung 'Lokal' ist die gleiche Einstellung wie DCE beim PC/PPI-Kabel.
- ❏ Stellen Sie 11 Bit ein (Schalter 7=0).

Bei Anschluss an ein Modem:

- ❏ Stellen Sie den Modus PPI/frei programmierbare Kommunikation ein (Schalter 5=0).
- ❏ Stellen Sie die Baudrate ein (Schalter 1, 2 und 3).
- ❏ Stellen Sie 'Entfernt' ein (Schalter 6=1). Die Einstellung 'Entfernt' ist die gleiche Einstellung wie DTE beim PC/PPI-Kabel.
- ❏ Stellen Sie 10 Bit oder 11 Bit ein (Schalter 7). Diese Einstellung muss der Einstellung für die Anzahl der Bits pro Zeichen Ihres Modems entsprechen.

Verwendung des S7-200 RS-232/PPI-Multi-Master-Kabels mit STEP 7-Micro/WIN 3.2 Service Pack 4 (oder höher)

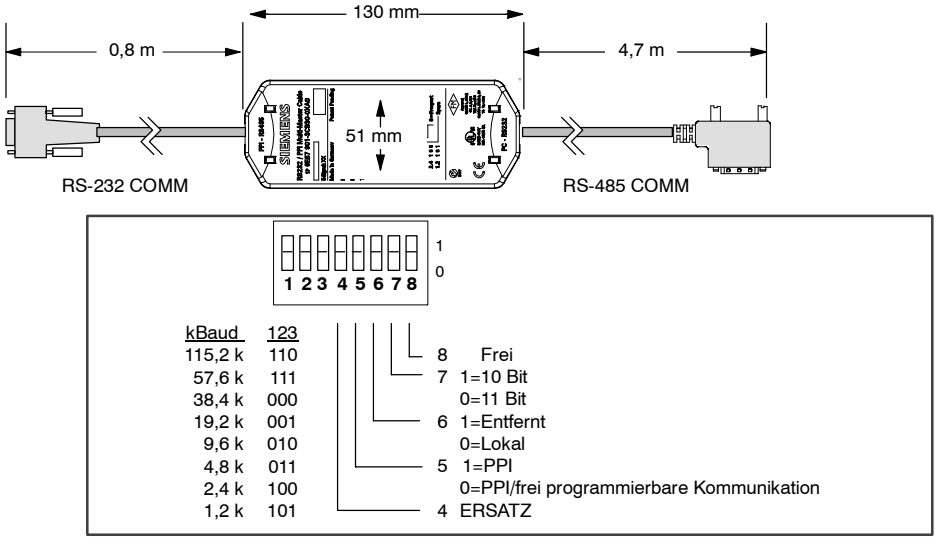
Bei direktem Anschluss an den PC

- ❑ Stellen Sie den PPI-Modus ein (Schalter 5=1).
- ❑ Stellen Sie 'Lokal' ein (Schalter 6=0).

Bei Anschluss an ein Modem:

- ❑ Stellen Sie den PPI-Modus ein (Schalter 5=1).
- ❑ Stellen Sie 'Entfernt' ein (Schalter 6=1).

Bild A-40 zeigt die Abmessungen, Kennzeichnung und LEDs des S7-200 RS-232/PPI-Multi-Master-Kabels.



LED	Farbe	Beschreibung
Tx	Grün	RS-232-Sendeanzeige
Rx	Grün	RS-232-Empfangsanzeige
PPI	Grün	RS-485-Sendeanzeige

Bild A-40 Abmessungen, Kennzeichnung und LEDs des S7-200 RS-232/PPI-Multi-Master-Kabels

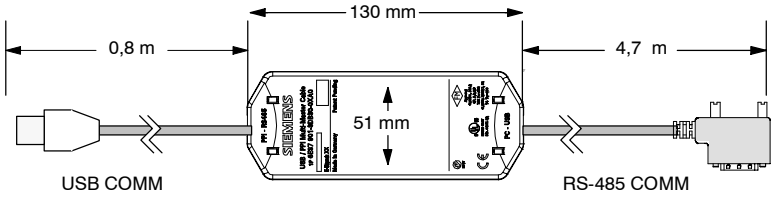
S7-200 USB/PPI-Multi-Master-Kabel

Für die Verwendung des USB-Kabels müssen Sie STEP 7-Micro/WIN 3.2 Service Pack 4 (oder höher) installiert haben. Das USB-Kabel unterstützt die frei programmierbare Kommunikation nicht.

Tabelle A-66 S7-200 USB/PPI-Multi-Master-Kabel - Anschlussbelegung RS-485 zu USB-Steckverbinder Serie “A”

Anschlussbelegung RS-485-Steckverbinder		Anschlussbelegung USB-Steckverbinder	
Pin	**Signalbeschreibung**	**Pin**	**Signalbeschreibung**
1	Kein Anschluss	1	USB – DataP
2	24-V-Rückleitung (RS-485 logische Erde)	2	USB – DataM
3	Signal B (RxD/TxD+)	3	USB 5 V
4	RTS (TTL-Level)	4	USB logische Erde
5	Kein Anschluss		
6	Kein Anschluss		
7	24-V-Spannungsversorgung		
8	Signal A (RxD/TxD-)		
9	Protokolleinstellung (Low = 10 Bit)		

Bild A-41 zeigt die Abmessungen und LEDs des S7-200 USB/PPI-Multi-Master-Kabels.



LED	Farbe	Beschreibung
Tx	Grün	USB-Sendeanzeige
Rx	Grün	USB-Empfangsanzeige
PPI	Grün	RS-485-Sendeanzeige

Bild A-41 Abmessungen und LEDs des S7-200 USB/PPI-Multi-Master-Kabels

A

Eingangssimulatoren

Bestellnummer	Simulator mit 8 Anschlussklemmen 6ES7 274-1XF00-0XA0	Simulator mit 14 Anschlussklemmen 6ES7 274-1XH00-0XA0	Simulator mit 24 Anschlussklemmen 6ES7 274-1XK00-0XA0
Größe (L x B x T)	61 x 36 x 22 mm	91 x 36 x 22 mm	147 x 36 x 25 mm
Gewicht	0,02 kg	0,03 kg	0,04 kg
Ein- und Ausgänge	8	14	24

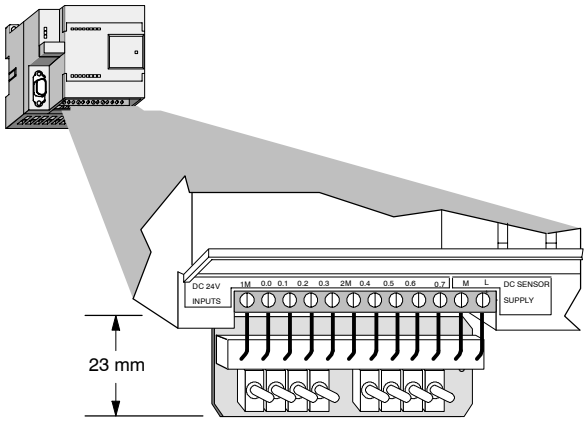


Bild A-42 Einbau des Eingangssimulators

Vorsicht

Diese Eingangssimulatoren sind nicht geeignet für die Nutzung in explosionsgefährdeten Betriebsstätten der Gefahrengruppen Klasse I DIV 2 und Klasse I Zone 2. Die Schalter können zu Funkenbildung führen.

Setzen Sie keine Eingangssimulatoren in explosionsgefährdeten Betriebsstätten der Gefahrengruppen Klasse I DIV 2 und Klasse I Zone 2 ein.

A

A

Berechnen der Leistungsbilanz

B

Die S7-200 CPU besitzt eine interne Spannungsversorgung, die neben der CPU selbst die Erweiterungsmodule und andere 24-V-DC-Verbraucher versorgt. Mit Hilfe der folgenden Informationen können Sie berechnen, wieviel Leistung die S7-200 CPU für Ihre Konfiguration zur Verfügung stellen kann.

Leistungsbedarf

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

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

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

Tipp

Wird die Leistungsbilanz der CPU überschritten, können Sie evtl. nicht die maximal zulässige Anzahl Module für Ihre CPU anschließen.

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.

Berechnen eines Beispiels für den Leistungsbedarf

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

- S7-200 CPU 224 AC/DC/Relais
- 3 EM 223 8 DC-Eingänge/8 Relaisausgänge
- 1 EM 221 8 DC-Eingänge

Dieser Aufbau verfügt insgesamt über 46 Eingänge und 34 Ausgänge.

Die S7-200 CPU in diesem Beispiel liefert genügend 5-V-DC-Spannung für die Erweiterungsmodule, doch die Geberversorgung liefert nicht genügend 24-V-DC-Spannung für alle Eingänge und Erweiterungsrelais. Die Ein- und Ausgänge benötigen 400 mA und die S7-200 CPU liefert nur 280 mA. Dieser Aufbau benötigt zusätzlich mindestens 120 mA an 24-V-DC-Spannung, um alle 24-V-DC-Ein- und Ausgänge zu versorgen.

Tabelle B-1 Beispiel für die Berechnung der Leistungsbilanz einer S7-200

Leistung der CPU	**5 V DC**	**24 V DC**
CPU 224 AC/DC/Relais	660 mA	280mA

minus

Leistungsbedarf des Systems	**5 V DC**	**24 V DC**
CPU 224, 14 Eingänge		14 * 4 mA = 56 mA
3 EM 223, 5-V-Spannung erforderlich	3 * 80 mA = 240 mA	
1 EM 221, 5-V-Spannung erforderlich	1 * 30 mA = 30 mA	
3 EM 223, je 8 Eingänge		3 * 8 * 4 mA = 96 mA
3 EM 223, je 8 Relaisausgänge		3 * 8 * 9 mA = 216 mA
1 EM 221, 8 Eingänge		8 * 4 mA = 32 mA
Gesamtbedarf	270 mA	400 mA

gleich

Leistungsbilanz	**5 V DC**	**24 V DC**
Gesamt-Leistungsbilanz	390 mA	[120 mA]

B

Berechnen des Leistungsbedarfs

Mit Hilfe dieser Tabelle können Sie berechnen, wieviel Leistung die S7-200 CPU für Ihre Konfiguration liefern kann. Die Datenblätter in Anhang A bieten Informationen zu der verfügbaren Leistung der einzelnen Ausführungen der CPUs und zum Leistungsbedarf der Erweiterungsmodule.

Leistung der CPU	**5 V DC**	**24 V DC**

minus

Leistungsbedarf des Systems	**5 V DC**	**24 V DC**
Gesamtbedarf		

gleich

Leistungsbilanz	**5 V DC**	**24 V DC**
Gesamt-Leistungsbilanz		

B

Fehlermeldungen

C

Die folgenden Informationen zu möglichen Fehlern sollen Ihnen bei der Fehlerbehebung mit Ihrer S7-200 CPU behilflich sein.

In diesem Kapitel

Fehlercodes und Meldungen von schweren Fehlern 424
Fehler zur Laufzeit 425
Verletzungen der Übersetzungsregeln 426

Fehlercodes und Meldungen von schweren Fehlern

Tritt ein schwerer Fehler auf, beendet die S7-200 die Bearbeitung des Programms. Je nach Schweregrad des Fehlers kann die S7-200 einige oder auch gar keine Funktionen mehr ausführen. Ziel der Behebung von schweren Fehlern ist es, die S7-200 in einen sicheren Zustand zu bringen, so dass Informationen zu der Fehlerbedingung in der S7-200 abgefragt werden können.

Die S7-200 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 S7-200 bleibt in diesem Zustand, bis der schwere Fehler behoben ist. Wenn Sie sich die Fehlercodes ansehen möchten, wählen Sie in der Menüleiste den Menübefehl **Zielsystem > Informationen**. Tabelle C-1 führt für alle Fehlercodes, die die S7-200 zu schweren Fehlern ausgibt, eine Beschreibung auf.

Tabelle C-1 Fehlercodes und Meldungen von schweren Fehlern in der S7-200

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 (SDB0).
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 (SDB0).
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[1]	Fehler bei indirekter Adressierung des Vergleichskontakts.
0012[1]	Unzulässiger Gleitpunktwert in Vergleichskontakt.
0013	Speichermodul ist leer oder das Programm wird von der S7-200 nicht unterstützt.
0014[1]	Bereichsfehler Vergleichskontakt.

[1] Die Vergleichskontaktfehler sind die einzigen Fehler, die sowohl schwere als auch leichte Fehlerbedingungen generieren. Der leichte Fehler wird generiert, damit die Programmadresse des Fehlers gespeichert wird.

Fehler zur Laufzeit

Während der Bearbeitung Ihres Programms können leichte Fehlerbedingungen auftreten (z. B. Adressierungsfehler). In diesem Fall gibt die S7-200 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	Beschreibung
0000	Keine schweren Fehler vorhanden; kein Fehler.
0001	Box HSC vor Bearbeitung der Box HDEF freigegeben.
0002	Eingangsinterrupt, der bereits einem HSC zugeordnet ist, wurde einem Eingang zugewiesen.
0003	Zuordnung von Eingängen zu einem HSC, der bereits einem Eingangsinterrupt oder einem anderen HSC zugeordnet ist.
0004	Versuch, eine der Operationen ENI, DISI, SPA oder HDEF im Interruptprogramm auszuführen.
0005	Versuch, vor Beenden der ersten Operation HSC/PLS eine zweite Operation HSC/PLS mit der gleichen Nummer auszuführen (HSC im Interruptprogramm in Konflikt mit HSC/PLS im Hauptprogramm).
0006	Fehler bei indirekter Adressierung.
0007	Datenfehler in TODW (Echtzeituhr schreiben) oder TODR (Echtzeituhr lesen).
0008	Maximale Schachtelungstiefe für Unterprogramme überschritten.
0009	Gleichzeitige Ausführung der Befehle XMT/RCV an Schnittstelle 0.
000A	Versuch, HSC durch zweite Operation HDEF für denselben HSC neu zu definieren.
000B	Gleichzeitige Ausführung der Befehle XMT/RCV an Schnittstelle 1.
000C	Uhrmodul nicht vorhanden für Zugriff durch TODR, TODW oder Kommunikation.
000D	Versuch, Impulsausgabe neu zu definieren, während die Funktion aktiv ist.
000E	Nummer des PTO-Profilsegments wurde auf 0 gesetzt.
000F	Unzulässiger numerischer Wert in Vergleichskontakt.
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).
009A	Versuch, aus einem Anwenderinterrupt in die frei programmierbare Kommunikation zu schalten.
009B	Unzulässiger Index (Zeichenkette mit Wert 0 in Anfangsposition).

Verletzungen der Übersetzungsregeln

Wenn Sie ein Programm in die S7-200 laden, übersetzt die CPU das Programm. Erkennt die S7-200, dass das Programm eine Übersetzungsregel verletzt (z.B. eine Operation ungültig ist), dann bricht die S7-200 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. Verringern Sie die Programmgröße.
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	Unzulässige Operation JMP in oder aus einem SCR-Segment.
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.
0097	Anwenderprogramm enthält sowohl unnummerierte als auch nummerierte Operationen EV/ED.
0098	Unzulässige Bearbeitung im Betriebszustand RUN (versuchte Bearbeitung im Programm mit unnummerierten Operationen EV/ED).
0099	Zu viele versteckte Programmsegmente (Operationen HIDE).
009B	Unzulässiger Index (Zeichenkette mit Wert 0 in Anfangsposition).
009C	Maximale Operationslänge überschritten.

Sondermerker

Sondermerker bieten verschiedene Status- und Steuerungsfunktionen und dienen dazu, Informationen zwischen der S7-200 und Ihrem Programm auszutauschen. Sondermerker können als Bits, Bytes, Wörter und Doppelwörter verwendet werden.

In diesem Kapitel

SMB0: Statusbits 428
SMB1: Statusbits 428
SMB2: Empfangene Zeichen in der frei programmierbaren Kommunikation 429
SMB3: Paritätsfehler in der frei programmierbaren Kommunikation 429
SMB4: Überlauf der Warteschlange 429
SMB5: E/A-Status 430
SMB6: Kennregister der CPU 430
SMB7: Reserviert. 430
SMB8 bis SMB21: Kenn- und Fehlerregister des E/A-Moduls 431
SMW22 bis SMW26: Zykluszeiten 432
SMB28 und SMB29: Analogpotentiometer 432
SMB30 und SMB130: Steuerungsregister der frei programmierbaren Kommunikation 432
SMB31 und SMW32: Schreibsteuerung nullspannungsfester Speicher (EEPROM) 433
SMB34 und SMB35: Intervallregister für zeitgesteuerte Interrupts 433
SMB36 bis SMB65: Register HSC0, HSC1 und HSC2 433
SMB66 bis SMB85: PTO/PWM-Register 435
SMB86 bis SMB94 und SMB186 bis SMB194: Steuerung des Meldungsempfangs 436
SMW98: Fehler im Erweiterungsbus 437
SMB130: Steuerungsregister der frei programmierbaren Kommunikation (siehe SMB30) 437
SMB131 bis SMB165: Register HSC3, HSC4 und HSC5 437
SMB166 bis SMB185: Profildefinitionstabelle PTO0, PTO1 438
SMB186 bis SMB194: Steuerung des Meldungsempfangs (siehe SMB86 bis SMB94) 438
SMB200 bis SMB549: Status intelligentes Modul 439

SMB0: Statusbits

Wie in Tabelle D-1 beschrieben, enthält SMB0 acht Statusbits, die von der S7-200 am Ende eines jeden Zyklus aktualisiert werden.

Tabelle D-1 Sondermerker SMB0 (SM0.0 bis SM0.7)

Sondermerker	Beschreibung (schreibgeschützt)
SM0.0	Dieses Bit ist immer eingeschaltet.
SM0.1	Dieses Bit ist im ersten Zyklus eingeschaltet. Es wird z.B. zum Aufrufen eines Unterprogramms für die Initialisierung 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 freigeben, 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 (schreibgeschützt)
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: Empfangene Zeichen in 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.

Tipp

SMB2 und SMB3 werden von Schnittstelle 0 und Schnittstelle 1 gemeinsam genutzt. Löst der Empfang eines Zeichens an Schnittstelle 0 die Ausführung des Interruptprogramms aus, das dem Ereignis (Interruptereignis 8) zugeordnet ist, dann enthält SMB2 das an Schnittstelle 0 empfangene Zeichen und SMB3 enthält den Paritätsstatus des Zeichens. Löst der Empfang eines Zeichens an Schnittstelle 1 die Ausführung des Interruptprogramms aus, das dem Ereignis (Interruptereignis 25) zugeordnet ist, dann enthält SMB2 das an Schnittstelle 1 empfangene Zeichen und SMB3 enthält den Paritätsstatus des Zeichens.

Tabelle D-3 Sondermerker SMB2

Sondermerker	Beschreibung (schreibgeschützt)
SMB2	Dieses Byte enthält alle Zeichen, die in der frei programmierbaren Kommunikation über Schnittstelle 0 oder Schnittstelle 1 empfangen werden.

SMB3: Paritätsfehler in 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 (schreibgeschützt)
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 (schreibgeschützt)
SM4.0[1]	Wird aktiviert, wenn in der Warteschlange für Kommunikationsinterrupts ein Überlauf auftritt.
SM4.1[1]	Wird aktiviert, wenn in der Warteschlange für Eingangsinterrupts 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	Dieses Bit wird eingeschaltet, wenn etwas geforct ist.

[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: E/A-Status

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 (schreibgeschützt)
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	Dieses Bit wird eingeschaltet, wenn zu viele intelligente Erweiterungsmodule an den E/A-Bus angeschlossen sind.
SM5.4 bis SM5.7	Reserviert.

SMB6: Kennregister der CPU

Wie in Tabelle D-7 beschrieben, ist SMB6 das Kennregister der S7-200 CPU. SM6.4 bis SM6.7 enthalten die Kennung der S7-200 CPU. SM6.0 bis SM6.3 sind für zukünftige Funktionen reserviert.

Tabelle D-7 Sondermerker SMB6

Sondermerker	Beschreibung (schreibgeschützt)
Format	MSB 7 … LSB 0 x x x x r r r r Kennregister der CPU
SM6.0 bis SM6.3	Reserviert.
SM6.4 bis SM6.7	xxxx = 0000 = CPU 222 0010 = CPU 224 0110 = CPU 221 1001 = CPU 226/CPU 226XM

SMB7: Reserviert.

SMB7 ist für zukünftige Funktionen reserviert.

D

SMB8 bis SMB21: Kenn- und Fehlerregister des E/A-Moduls

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 (schreibgeschützt)
Format	Byte mit ger. Nummer: Kennregister des Moduls MSB 7 … LSB 0 m t t a i i Q Q M: Modul vorhanden 0 = Vorhanden, 1 = Nicht vorhanden tt: Modultyp 00 Nicht intelligentes Erweiterungsmodul 01 Intelligentes Modul 10 Reserviert. 11 Reserviert. A: Art der E/A 0 = Digital, 1 = Analog ii: Eingänge 00 Keine Eingänge 01 2 AE oder 8 DE 10 4 AE oder 16 DE 11 8 AE oder 32 DE QQ: Ausgänge 00 Keine Ausgänge 01 2 AA oder 8 DA 10 4 AA oder 16 DA 11 8 AA oder 32 DA Byte mit unger. Nummer: Fehlerregister des Moduls MSB 7 … LSB 0 c 0 0 b r p f t c: Konfigurationsfehler 0 = kein Fehler, 1 = Fehler b: Busfehler o. Paritätsfehler r: Bereichsüberschreitung P: Kein Anwenderstrom f: Sicherung durchgebrannt t: Klemmenblock lose
SMB8 SMB9	Kennregister Modul 0 Fehlerregister Modul 0
SMB10 SMB11	Kennregister Modul 1 Fehlerregister Modul 1
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 (schreibgeschützt)
SMW22	Zykluszeit des letzten Zyklus in Millisekunden
SMW24	Kürzeste Zykluszeit seit Beginn des Betriebszustands RUN
SMW26	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 (schreibgeschützt)
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	Steuerbyte für die frei programmierbare Kommunikation MSB 7 … LSB 0 p p d b b b m m
SM30.0 und SM30.1	SM130.0 und SM130.1	mm: Protokollauswahl (PPI/Slave-Modus) 00 = Protokoll der Pkt-zu-Pkt-Schnittstelle 01 = Protokoll der frei programmierbaren Kommunikation 10 = PPI/Master-Modus 11 = Reserviert (Voreinstellung: PPI/Slave-Modus) Hinweis: Wählen Sie mm = 10 (PPI-Master), dann wird die S7-200 zum Master im Netz und die Operationen NETR und NETW können ausgeführt werden. Bits 2 bis 7 werden im PPI-Modus ignoriert.
SM30.2 bis SM30.4	SM130.2 bis SM130.4	bbb: Baudrate frei progr. Komm. 000 = 38.400 Baud 001 = 19.200 Baud 010 = 9.600 Baud 011 = 4800 Baud 100 = 2.400 Baud 101 = 1.200 Baud 110 = 115.200 Baud 111 = 57.600 Baud
SM30.5	SM130.5	d: Datenbits pro Zeichen 0 = 8 Bits pro Zeichen 1 = 7 Bits pro Zeichen
SM30.6 und SM30.7	SM130.6 und SM130.7	pp: Paritätsauswahl 00 = keine Parität 01 = gerade Parität 10 = ungerade Parität 11 = ungerade Parität

D

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 S7-200 den Sondermerker SM31.7 zurücksetzt und dadurch das Ende der Speicheroperation anzeigt.

Am Ende eines jeden Zyklus prüft die S7-200, 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 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	SMB31: Software-Befehl MSB 7 … LSB 0 c 0 0 0 0 0 s s SMW32 Adresse im Variablen-speicher MSB 15 … LSB 0 Adresse im Variablenspeicher
SM31.0 und SM31.1	ss: Größe der Daten 00 = Byte 01 = Byte 10 = Wort 11 = Doppelwort
SM31.7	c: Im EEPROM speichern 0 = Keine Anforderung zu speichern 1 = Anwenderprogramm fordert Speicherung von Daten an Die S7-200 setzt dieses Bit nach jeder Speicheroperation zurück.
SMW32	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.

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 1 und 255 ms angeben. Die S7-200 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 1 und 255 ms) für den zeitgesteuerten Interrupt 0 an.
SMB35	Dieses Byte gibt das Intervall (in Inkrementen von 1 ms zwischen 1 und 255 ms) für den zeitgesteuerten Interrupt 1 an.

SMB36 bis SMB65: Register HSC0, HSC1 und HSC2

Wie in Tabelle D-14 beschrieben, werden SMB36 bis SMB65 dazu verwendet, den Betrieb der schnellen Zähler HSC0, HSC1 und HSC2 zu überwachen und zu steuern.

Tabelle D-14 Sondermerker SMB36 bis SMB62

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	Steuerbit für Pegel bei Rücksetzen aktiv: 0 = Rücksetzen ist Aktiv High, 1 = Rücksetzen ist Aktiv Low
SM37.1	Reserviert.
SM37.2	Zählgeschwindigkeit A/B-Zähler: 0 - vierfache Zählgeschwindigkeit, 1 - einfache Zählgeschwindigkeit.
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
SMD38	HSC0 - Neuer aktueller Wert
SMD42	HSC0 - Neuer voreingestellter Wert
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 - Pegel bei aktiv zum Rücksetzen: 0 = Aktiv High; 1 = Aktiv Low
SM47.1	Steuerbit HSC1 - Pegel für Starten aktiv: 0 = Aktiv High; 1 = Aktiv Low
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
SMD48	HSC1 - Neuer aktueller Wert
SMD52	HSC1 - Neuer voreingestellter Wert
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 - Pegel für Rücksetzen aktiv: 0 = Aktiv High; 1 = Aktiv Low
SM57.1	Steuerbit HSC2 - Pegel für Starten aktiv: 0 = Aktiv High; 1 = Aktiv Low
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

Tabelle D-14 Sondermerker SMB36 bis SMB62 (continued)

Sondermerker	Beschreibung
SMD58	HSC2 - Neuer aktueller Wert
SMD62	HSC2 - Neuer voreingestellter Wert

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. In der Beschreibung der schnellen Ausgangsoperationen in Kapitel 6 finden Sie ausführliche Informationen zu diesen Bits.

Tabelle D-15 Sondermerker SMB66 bis SMB85

Sondermerker	Beschreibung
SM66.0 bis SM66.3	Reserviert.
SM66.4	PTO0 - Profil abgebrochen: 0 = kein Fehler, 1 = abgebrochen wegen Delta-Berechnungsfehler
SM66.5	PTO0 - Profil abgebrochen: 0 = nicht durch Anwenderbefehl abgebrochen, 1 = durch Anwenderbefehl abgebrochen
SM66.6	PTO0 - Pipeline-Überlauf (vom System gelöscht bei Verwendung von externem Profil; andernfalls vom Anwender zurückzusetzen): 0 = kein Überlauf; 1 = Pipeline-Überlauf
SM66.7	PTO0 - Leerlaufbit: 0 = PTO wird bearbeitet; 1 = PTO im Leerlauf
SM67.0	PTO0/PWM0 - Zykluszeit aktualisieren: 1 = neue Zykluszeit schreiben
SM67.1	PWM0: Zeitwert der Impulsdauer aktualisieren: 1 = neue Impulsdauer schreiben
SM67.2	PTO0 - Impulszählwert aktualisieren: 1 = neuen Impulszählwert schreiben
SM67.3	PTO0/PWM0 - Zeitbasis: 0 = 1 μs/Takt, 1 = 1 ms/Takt
SM67.4	PWM0 synchron aktualisieren: 0 = asynchron aktualisieren, 1 = synchron aktualisieren
SM67.5	Funktion PTO0: 0 = Einzel-Segment-Betrieb (Zykluszeit und Impulszählwert in Sondermerkern gespeichert), 1 = Multi-Segment-Betrieb (Profiltabelle im Variablenspeicher abgelegt)
SM67.6	PTO0/PWM0 - Modus auswählen: 0 = PTO; 1 = PWM
SM67.7	PTO0/PWM0 - Freigabebit: 1 = Freigeben
SMW68	PTO0/PWM0 - Zykluszeit (2 bis 65.535 Einheiten der Zeitbasis);
SMW70	PWM0 - Impulsdauer (0 bis 65.535 Einheiten der Zeitbasis);
SMD72	PTO0 - Impulszählwert (1 bis 2^{32} -1);
SM76.0 bis SM76.3	Reserviert.
SM76.4	PTO1 - Profil abgebrochen: 0 = kein Fehler, 1 = abgebrochen wegen Delta-Berechnungsfehler
SM76.5	PTO1 - Profil abgebrochen: 0 = nicht durch Anwenderbefehl abgebrochen, 1 = durch Anwenderbefehl abgebrochen
SM76.6	PTO1 - Pipeline-Überlauf (vom System gelöscht bei Verwendung von externem Profil; andernfalls vom Anwender zurückzusetzen): 0 = kein Überlauf; 1 = Pipeline-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: Zeitwert der Impulsdauer aktualisieren: 1 = neue Impulsdauer schreiben
SM77.2	PTO1 - Impulszählwert aktualisieren: 1 = neuen Impulszählwert schreiben
SM77.3	PTO1/PWM1 - Zeitbasis: 0 = 1 μs/Takt, 1 = 1 ms/Takt
SM77.4	PWM1 synchron aktualisieren: 0 = asynchron aktualisieren, 1 = synchron aktualisieren
SM77.5	Funktion PTO1: 0 = Einzel-Segment-Betrieb (Zykluszeit und Impulszählwert in Sondermerkern gespeichert), 1 = Multi-Segment-Betrieb (Profiltabelle im Variablenspeicher abgelegt)
SM77.6	PTO1/PWM1 - Modus auswählen: 0 = PTO; 1 = PWM
SM77.7	PTO1/PWM1 - Freigabebit: 1 = Freigeben
SMW778	PTO1/PWM1 - Zykluszeit (2 bis 65.535 Einheiten der Zeitbasis);

Tabelle D-15 Sondermerker SMB66 bis SMB85 (continued)

Sondermerker	Beschreibung
SMW80	PWM1 - Impulsdauer (0 bis 65.535 Einheiten der Zeitbasis);
SMD82	PTO1 - Impulszählwert (1 bis 2^{32} -1);

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

Schnitt-stelle 0	Schnitt-stelle 1	Beschreibung
SMB86	SMB186	Statusbyte zum Empfangen von Meldungen MSB 7 … LSB 0 n r e 0 0 t c p n: 1 = Meldungsempfang vom Anwender gesperrt r: 1 = Meldungsempfang beendet: Fehler in Eingabeparameter oder fehlende Start- oder Endebedingung e: 1 = Endezeichen empfangen t: 1 = Meldungsempfang beendet: Zeit abgelaufen c: 1 = Meldungsempfang beendet: maximale Zeichenzahl erreicht p 1 = Meldungsempfang beendet: Paritätsfehler
SMB87	SMB187	Steuerbyte zum Empfangen von Meldungen MSB 7 … LSB 0 EN sc EC il c/m tmr bk 0 en: 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. sc: 0 = SMB88 oder SMB188 ignorieren. 1 = Mit dem Wert von SMB88 oder von SMB188 den Beginn der Meldung erkennen. ec: 0 = SMB89 oder SMB189 ignorieren. 1 = Mit dem Wert von SMB89 oder von SMB189 das Ende der Meldung erkennen. il: 0 = SMW90 oder SMW190 ignorieren. 1 = Mit dem Wert von SMW90 oder SMW190 Leerlauf erkennen. c/m: 0 = Zeit misst die Zeit zwischen den Zeichen. 1 = Zeit misst die Zeit der Meldungen. tmr: 0 = SMW92 oder SMW192 ignorieren. 1 = Empfang beenden, wenn die Zeitdauer von SMW92 bzw. SMW192 überschritten ist. bk: 0 = Pausenbedingungen ignorieren. 1 = Mit Pausenbedingung Meldungsbeginn erkennen.
SMB88	SMB188	Zeichen für den Beginn einer Meldung
SMB89	SMB189	Zeichen für das Ende einer Meldung
SMW90	SMW190	Leerlaufzeit in Millisekunden. Das ersten Zeichen, das nach Ablauf der Leerlaufzeit empfangen wird, kennzeichnet den Beginn einer neuen Meldung.
SMW92	SMW192	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.
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.

D

SMW98: Fehler im Erweiterungsbus

Wie in Tabelle D-17 beschrieben, liefert SMW98 Informationen zu der Anzahl der Fehler im Erweiterungsbus.

Tabelle D-17 Sondermerkerbyte SMW98

Sondermerker	Beschreibung
SMW98	Diese Adresse wird jedesmal, wenn auf dem Erweiterungsbus ein Paritätsfehler erkannt wird, um 1 erhöht. Gelöscht wird die Adresse beim Anlauf oder durch den Anwender.

SMB130: Steuerungsregister der frei programmierbaren Kommunikation *(siehe SMB30)*

Siehe Tabelle D-11.

SMB131 bis SMB165: Register HSC3, HSC4 und HSC5

Wie in Tabelle D-18 beschrieben, werden SMB130 bis SMB165 dazu verwendet, den Betrieb der schnellen Zähler HSC3, HSC4 und HSC5 zu überwachen und zu steuern.

Tabelle D-18 Sondermerker SMB131 bis SMB165

Sondermerker	Beschreibung
SMB131 bis SMB135	Reserviert.
SM136.0 bis SM136.4	Reserviert.
SM136.5	Statusbit HSC3 - Aktuelle Zählrichtung: 1 = Vorwärtszählen
SM136.6	Statusbit HSC3 - Aktueller Wert gleich voreingestellter Wert 1 = gleich
SM136.7	Statusbit HSC3 - Aktueller Wert ist größer als voreingestellter Wert 1 = größer als
SM137.0 bis SM137.2	Reserviert.
SM137.3	Steuerbit HSC3 - Zählrichtung: 1 = Vorwärtszählen
SM137.4	HSC3 - Richtung aktualisieren: 1 = Richtung aktualisieren
SM137.5	HSC3 - Voreingestellten Wert aktualisieren: 1 = neuen Wert in die Voreinstellung von HSC3 schreiben
SM137.6	HSC3 - Aktuellen Wert aktualisieren: 1 = neuen Wert in aktuellen Wert von HSC3 schreiben
SM137.7	HSC3 - Freigabebit: 1 = Freigeben
SMD138	HSC3 - Neuer aktueller Wert:
SMD142	HSC3 - Neuer voreingestellter Wert:
SM146.0 bis SM146.4	Reserviert.
SM146.5	Statusbit HSC4 - Aktuelle Zählrichtung: 1 = Vorwärtszählen
SM146.6	Statusbit HSC4 - Aktueller Wert gleich voreingestellter Wert: 1 = gleich
SM146.7	Statusbit HSC4 - Aktueller Wert ist größer als voreingestellter Wert: 1 = größer als
SM147.0	Steuerbit für Pegel bei Rücksetzen aktiv: 0 = Rücksetzen ist Aktiv High, 1 = Rücksetzen ist Aktiv Low
SM147.1	Reserviert.
SM147.2	Zählgeschwindigkeit A/B-Zähler: 0 - vierfache Zählgeschwindigkeit, 1 - einfache Zählgeschwindigkeit.
SM147.3	Steuerbit HSC4 - Zählrichtung: 1 = Vorwärtszählen
SM147.4	HSC4 - Richtung aktualisieren: 1 = Richtung aktualisieren
SM147.5	HSC4 - Voreingestellten Wert aktualisieren: 1 = neuen Wert in die Voreinstellung von HSC4 schreiben
SM147.6	HSC4 - Aktuellen Wert aktualisieren: 1 = neuen Wert in aktuellen Wert von HSC4 schreiben
SM147.7	HSC4 - Freigabebit: 1 = Freigeben
SMD148	HSC4 - Neuer aktueller Wert

Tabelle D-18 Sondermerker SMB131 bis SMB165, Fortsetzung

Sondermerker	Beschreibung
SMD152	HSC4 - Neuer voreingestellter Wert
SM156.0 bis SM156.4	Reserviert.
SM156.5	Statusbit HSC5 - Aktuelle Zählrichtung: 1 = Vorwärtszählen
SM156.6	Statusbit HSC5 - Aktueller Wert gleich voreingestellter Wert: 1 = gleich
SM156.7	Statusbit HSC5 - Aktueller Wert ist größer als voreingestellter Wert: 1 = größer als
SM157.0 bis SM157.2	Reserviert.
SM157.3	Steuerbit HSC5 - Zählrichtung: 1 = Vorwärtszählen
SM157.4	HSC5 - Richtung aktualisieren: 1 = Richtung aktualisieren
SM157.5	HSC5 - Voreingestellten Wert aktualisieren: 1 = neuen Wert in die Voreinstellung von HSC5 schreiben
SM157.6	HSC5 - Aktuellen Wert aktualisieren: 1 = neuen Wert in aktuellen Wert von HSC5 schreiben
SM157.7	HSC5 - Freigabebit: 1 = Freigeben
SMD158	HSC5 - Neuer aktueller Wert
SMD162	HSC5 - Neuer voreingestellter Wert

SMB166 bis SMB185: Profildefinitionstabelle PTO0, PTO1

Wie in Tabelle D-19 beschrieben, werden SMB166 bis SMB185 dazu verwendet, die Anzahl der aktiven Profilschritte und die Adresse der Profiltabelle im Variablenspeicher anzuzeigen.

Tabelle D-19 Sondermerker SMB166 bis SMB185

Sondermerker	Beschreibung
SMB166	Aktuelle Eintragsnummer des aktiven Profilschritts für PTO0.
SMB167	Reserviert.
SMW168	Adresse im Variablenspeicher der Profiltabelle für PTO0, angegeben als Versatz von V0.
SMB170 bis SMB175	Reserviert.
SMB176	Aktuelle Eintragsnummer des aktiven Profilschritts für PTO1.
SMB177	Reserviert.
SMW178	Adresse im Variablenspeicher der Profiltabelle für PTO1, angegeben als Versatz von V0.
SMB180 bis SMB185	Reserviert.

SMB186 bis SMB194: Steuerung des Meldungsempfangs *(siehe SMB86 bis SMB94)*

Siehe Tabelle D-16.

SMB200 bis SMB549: Status intelligentes Modul

Wie in Tabelle D-20 beschrieben, sind SMB200 bis SMB549 reserviert für Informationen von intelligenten Erweiterungsmodulen, z.B. dem Modul EM 277 PROFIBUS-DP. In den technischen Daten in Anhang A erfahren Sie, wie Ihr Modul SMB200 bis SMB549 nutzt.

Ab Version 2.2 werden die Sondermerker für intelligente Module auf andere Art und Weise zugeordnet.

Bei einer S7-200 CPU mit Firmware vor Version 1.2 müssen Sie das intelligente Modul aus Kompatibilitätsgründen neben der CPU einbauen.

Tabelle D-20 Sondermerkerbytes SMB200 bis SMB549

Sondermerkerbytes SMB200 bis SMB549							
Intelligentes Modul in Steckplatz 0	**Intelligentes Modul in Steckplatz 1**	**Intelligentes Modul in Steckplatz 2**	**Intelligentes Modul in Steckplatz 3**	**Intelligentes Modul in Steckplatz 4**	**Intelligentes Modul in Steckplatz 5**	**Intelligentes Modul in Steckplatz 6**	**Beschreibung**
SMB200 bis SMB215	SMB250 bis SMB265	SMB300 bis SMB315	SMB350 bis SMB365	SMB400 bis SMB415	SMB450 bis SMB465	SMB500 bis SMB515	Modulname (16 ASCII-Zeichen)
SMB216 bis SMB219	SMB266 bis SMB269	SMB316 bis SMB319	SMB366 bis SMB369	SMB416 bis SMB419	SMB466 bis SMB469	SMB516 bis SMB519	S/W-Versionsnummer (4 ASCII-Zeichen)
SMW220	SMW270	SMW320	SMW370	SMW420	SMW470	SMW520	Fehlercode
SMB222 bis SMB249	SMB272 bis SMB299	SMB322 bis SMB349	SMB372 bis SMB399	SMB422 bis SMB449	SMB472 bis SMB499	SMB522 bis SMB549	Informationen zu bestimmten Modultypen.

D

S7-200 Bestellnummern

E

CPUs	Bestellnummer
CPU 221, DC/DC/DC, 6 Eingänge / 4 Ausgänge	6ES7 211-0AA22-0XB0
CPU 221 AC/DC/Relais, 6 Eingänge / 4 Relais	6ES7 211-0BA22-0XB0
CPU 222, DC/DC/DC, 8 Eingänge / 6 Ausgänge	6ES7 212-1AB22-0XB0
CPU 222 AC/DC/Relais 8 Eingänge / 6 Relais	6ES7 212-1BB22-0XB0
CPU 224, DC/DC/DC, 14 Eingänge / 10 Ausgänge	6ES7 214-1AD22-0XB0
CPU 224 AC/DC/Relais 14 Eingänge /10 Relais	6ES7 214-1BD22-0XB0
CPU 226, DC/DC/DC, 24 Eingänge / 16 Ausgänge	6ES7 216-2AD22-0XB0
CPU 226 AC/DC/Relais 24 Eingänge / 16 Relais	6ES7 216-2BD22-0XB0
CPU 226XM DC/DC/DC 24 Eingänge / 16 Relais	6ES7 216-2AF22-0XB0
CPU 226XM AC/DC/Relais 24 Eingänge / 16 Relais	6ES7 216-2BF22-0XB0
Erweiterungsmodule	**Bestellnummer**
EM 221 24 V DC 8 Digitaleingänge	6ES7 221-1BF22-0XA0
EM 221 8 AC-Digitaleingänge (8 x 120/230 V AC)	6ES7 221-1EF22-0XA0
EM 221 Digitaleingabe 16 x 24 V DC	6ES7 221-1BH22-0XA0
EM 222 24 V DC 8 Digitalausgänge	6ES7 222-1BF22-0XA0
EM 222, Digitalausgabe 8 x Relais	6ES7 222-1HF22-0XA0
EM 222 8 AC-Digitalausgänge (8 x 120/230 V AC)	6ES7 222-1EF22-0XA0
EM 222 Digitalausgabe 4 x 24 VDC-5 A	6ES7 222-1BD22-0XA0
EM 222 Digitalausgabe 4 x Relais -10 A	6ES7 222-1HD22-0XA0
EM 223 24 V DC, Digitalein-/Digitalausgabe 4 Eingänge / 4 Ausgänge	6ES7 223-1BF22-0XA0
EM 223 24 V DC, Digitalein-/Digitalausgabe 4 Eingänge / 4 Relaisausgänge	6ES7 223-1HF22-0XA0
EM 223 24 V DC, Digitalein-/Digitalausgabe 8 Eingänge / 8 Ausgänge	6ES7 223-1BH22-0XA0
EM 223 24 V DC, Digitalein-/Digitalausgabe 8 Eingänge / 8 Relaisausgänge	6ES7 223-1PH22-0XA0
EM 223 24 V DC, Digitalein-/Digitalausgabe 16 Eingänge / 16 Ausgänge	6ES7 223-1BL22-0XA0
EM 223 24 V DC, Digitalein-/Digitalausgabe 16 Eingänge / 16 Relaisausgänge	6ES7 223-1PL22-0XA0
EM 231 Analogeingabe, 4 Eingänge	6ES7 231-0HC22-0XA0
EM 231 Analogeingabe RTD, 2 Eingänge	6ES7 231-7PB22-0XA0
EM 231 Analogeingabe Thermoelement, 4 Eingänge	6ES7 231-7PD22-0XA0
EM 232 Analogausgabe, 2 Ausgänge	6ES7 232-0HB22-0XA0
EM 235 Analogein-/Analogausgabe 4 Eingänge / 1 Ausgang	6ES7 235-0KD22-0XA0
Modemmodul EM 241	6ES7 241-1AA22-0XA0
Positioniermodul EM 253	6ES7 253-1AA22-0XA0
EM 277 PROFIBUS-DP	6ES7 277-0AA22-0XA0
(CP 243-2) AS-Interface-Modul	6GK7 243-2AX01-0XA0
(CP 243-I) Internet-Modul (mit elektronischer Dokumentation auf CD)	6GKY 243-1GX00-0XE0
(CP 243-1) Ethernet-Modul (mit elektronischer Dokumentation auf CD)	6GKY 243-1EX00-0XE0

Steckmodule und Kabel	Bestellnummer
MC 291, 32 K x 8 EEPROM Speichermodul	6ES7 291-8GE20-0XA0
CC 292, CPU 22x Echtzeituhr mit Batteriemodul	6ES7 297-1AA20-0XA0
BC 293, CPU 22x Batteriemodul	6ES7 291-8BA20-0XA0
Steckleitung für Erweiterungsmodule, 0,8 m, CPU 22x/EM	6ES7 290-6AA20-0XA0
Kabel, RS-232/PPI-Multi-Master	6ES7 901-3CB30-0XA0
Kabel, USB/PPI-Multi-Master	6ES7 901-3DB30-0XA0
Programmiersoftware	**Bestellnummer**
STEP 7-Micro/WIN 32 (V3.2) Einzellizenz (CD-ROM)	6ES7 810-2BC02-0YX0
STEP 7-Micro/WIN 32 (V3.2) Upgrade-Lizenz (CD-ROM)	6ES7 810-2BC02-0YX3
S7-200 Toolbox: TP-Designer für TP070, Version 1.0 (CD-ROM) Update	6ES7 850-2BC00-0YX0
STEP 7-Micro/WIN Optionspaket: STEP 7-Micro/WIN 32 Operationsbibliothek, V1.1 (CD-ROM)	6ES7 830-2BC00-0YX0

Kommunikationskarten	Bestellnummer
CP 5411: Short AT ISA	6GK 1 541-1AA00
CP 5511: PCMCIA, Typ II	6GK 1 551-1AA00
CP 5611: PCI-Karte (Version 3.0 oder höher)	6GK 1 561-1AA00
Handbücher	**Bestellnummer**
Textdisplay TD 200, Benutzerhandbuch	6ES7 272-0AA20-8BA0
Benutzerhandbuch TP070 Touch Panel (Englisch)	6AV6 591-1DC01-0AB0
S7-200 Punkt-zu-Punkt-Schnittstelle, Kommunikationshandbuch (Englisch/Deutsch)	6ES7 298-8GA00-8XH0
Handbuch CP 243-2 SIMATIC NET AS-Interface Master (Englisch)	6GK7 243-2AX00-8BA0
Automatisierungssystem S7-200, Systemhandbuch (Deutsch)	6ES7 298-8FA23-8AH0
Automatisierungssystem S7-200, Systemhandbuch (Englisch)	6ES7 298-8FA23-8BH0
Automatisierungssystem S7-200, Systemhandbuch (Französisch)	6ES7 298-8FA23-8CH0
Automatisierungssystem S7-200, Systemhandbuch (Spanisch)	6ES7 298-8FA23-8DH0
Automatisierungssystem S7-200, Systemhandbuch (Italienisch)	6ES7 298-8FA23-8EH0
Kabel, Busanschlussstecker und Busverstärker	**Bestellnummer**
MPI-Kabel	6ES7 901-0BF00-0AA0
PROFIBUS-Netzkabel	6XVI 830-0AH10
Busanschlussstecker mit Stecker für die Programmierschnittstelle, axialer Kabelabgang	6ES7 972-0BB11-0XA0
Busanschlussstecker ohne Stecker für die Programmierschnittstelle, axialer Kabelabgang	6ES7 972-0BA11-0XA0
RS-485 Busanschlussstecker mit 35°-Kabelabgang (keine Programmierschnittstelle)	6ES7 972-0BA40-0XA0
RS-485 Busanschlussstecker mit 35°-Kabelabgang (mit Programmierschnittstelle)	6ES7 972-0BB40-0XA0
CPU 22x/EM Klemmenblock, 7 Klemmen, steckbar	6ES7 292-1AD20-0AA0
CPU 22x/EM Klemmenblock, 12 Klemmen, steckbar	6ES7 292-1AE20-0AA0
CPU 22x/EM Klemmenblock, 14 Klemmen, steckbar	6ES7 292-1AF20-0AA0
CPU 22x/EM Klemmenblock, 18 Klemmen, steckbar	6ES7 292-1AG20-0AA0
RS-485 IP 20 Busverstärker, elektrisch getrennt	6ES7 972-0AA00-0XA0

Bedienerschnittstellen	Bestellnummer
Textdisplay TD 200	6ES7 272-0AA30-0YA0
Operator Panel OP3	6AV3 503-1DB10T
Operator Panel OP7	6AV3 607-1JC20-0AX1
Operator Panel OP17	6AV3 617-1JC20-0AX1
Touch Panel TP070	6AV6 545-0AA15-2AX0
Touch Panel TP170A	6AV6 545-0BA15-2AX0
Sonstiges	**Bestellnummer**
Erdungsklemmen für Hutschiene	6ES5 728-8MAll
Klemmenblock mit 12 Anschlussklemmen für Feldverdrahtung (CPU 221, CPU 222): 10 Stück	6ES7 290-2AA00-0XA0
Bausatz Ersatzabdeckklappen, enthält jeweils 4 der folgenden: Abdeckklappen für Klemmenblöcke mit 7, 12, 14, 18, 2x12, 2x14 Klemmen; CPU-Abdeckklappen, EM-Abdeckklappen	6ES7 291-3AX20-0XA0
Simulator mit 8 Anschlussklemmen	6ES7 274 1XF00-0XA0
Simulator mit 14 Anschlussklemmen	6ES7 274 1XH00-0XA0
Simulator mit 24 Anschlussklemmen	6ES7 274 1XK00-0XA0

E

Ausführungszeiten von AWL-Operationen

F

Die Ausführungszeiten der Operationen sind von großer Bedeutung, wenn Ihre Anwendung zeitkritische Funktionen enthält. Die Ausführungszeiten der Operationen werden in Tabelle F-3 aufgeführt.

Tipp

Wenn Sie mit den in Tabelle F-3 aufgeführten Ausführungszeiten arbeiten, müssen Sie die Auswirkungen des Signalflusses auf die Operation, die Auswirkungen der indirekten Adressierung und die Auswirkungen durch Zugriff auf bestimmte Speicherbereiche bei diesen Ausführungszeiten berücksichtigen. Diese Faktoren können sich direkt auf die aufgeführten Ausführungszeiten auswirken.

Auswirkungen des Signalflusses

Tabelle F-3 zeigt die Zeiten, die erforderlich sind, um die Logik oder die Funktion einer Operation auszuführen, wenn für die Operation Signalfluss vorhanden ist (oberster Stackwert = 1 oder EIN).

Ist kein Signalfluss vorhanden, dann beträgt die Ausführungszeit für die Operation 3 µs.

Auswirkungen der indirekten Adressierung

Tabelle F-3 zeigt die Zeiten, die erforderlich sind, um die Logik oder die Funktion einer Operation auszuführen, wenn Sie Operanden und Konstanten direkt adressieren.

Wird für Operationen indirekte Adressierung verwendet, erhöht sich die Ausführungszeit um 22 µs für jeden indirekt adressierten Operanden in der Operation.

Auswirkungen durch Zugriff auf bestimmte Speicherbereiche

Der Zugriff auf bestimmte Speicherbereiche, z.B. AE, AA, L und Akkumulatoren, bedarf zusätzlicher Ausführungszeit.

Tabelle F-1 zeigt die zusätzlichen Zeiten, die zu der Ausführungszeit einer Operation addiert werden müssen, wenn diese Speicherbereiche in einem Operanden angegeben werden.

Tabelle F-1 Zu addierender Wert für Zugriff auf bestimmte Speicherbereiche

Speicherbereich	Zur Ausführungszeit zu addierender Wert
Analogeingänge (AE)	
Ohne aktivierten Analogfilter:	149 µs
Mit aktiviertem Analogfilter:	0 µs
Analogausgänge (AA)	73 µs
Lokaldatenspeicher (L)	5,4 µs
Akkumulatoren (AC)	4,4 µs

Auswirkungen bestimmter Operationen für die CPU 226XM

Die Ausführung bestimmter verzweigender Operationen auf einer CPU 226XM benötigt zusätzliche Ausführungszeit.

Tabelle F-2 zeigt den Faktor, der bei jeder der aufgeführten Operationen zu der Grundausführungszeit addiert werden muss.

Tabelle F-2 Zu addierender Wert bei Operationen für die CPU 226XM

Operation	Zur Ausführungszeit zu addierender Wert
ATCH	1,0 µs
CALL	4,3 µs
CSCRE	3,1 µs
FOR (zur Grundausführungszeit zu addieren) (zum Schleifenmultiplikator zu addieren)	3,1 µs 3,1 µs
INT	1,7 µs
JMP	3,1 µs
RET	2,8 µs

Tabelle F-3 Ausführungszeiten der Operationen

Operation			µs
=	Verwendet:	E SM, T, Z, V, S, A, M L	0,37 1,8 19,2
+D			55
-D			55
*D			92
/D			376
+I			46
-I			47
*I			71
/I			115
=I	Verwendet:	Integrierte Ausgänge Erweiterungsausgänge	29 39
+R			110 163 (max.)
-R			113 166 (max.)
*R			100 130 (max.)
/R			300 360 (max.)
ATCH			20
ATH	Zeit = Grundausführungszeit + (Länge * LM) Grundausführungszeit (konstante Länge) Grundausführungszeit (variable Länge) Längenmultiplikator (LM)		 41 55 20
ATT			70
BCDI			66
BIR	Verwendet:	Integrierte Eingänge Erweiterungsausgänge	45 53
BIW	Verwendet:	Integrierte Ausgänge Erweiterungsausgänge	46 56
BMB	Zeit = Grundausführungszeit + (Länge * LM) Grundausführungszeit (konstante Länge) Grundausführungszeit (variable Länge) Längenmultiplikator (LM)		 21 51 11
BMD	Zeit = Grundausführungszeit + (Länge * LM) Grundausführungszeit (konstante Länge) Grundausführungszeit (variable Länge) Längenmultiplikator (LM)		 21 51 20
BMW	Zeit = Grundausführungszeit + (Länge * LM) Grundausführungszeit (konstante Länge) Grundausführungszeit (variable Länge) Längenmultiplikator (LM)		 21 51 16

Operation		µs
BTI		27
CALL	Ohne Parameter: Mit Parametern: Zeit = Grundausführungszeit + Σ (Operandenzeit) Grundausführungszeit Operandenzeit Bit (Eingang, Ausgang) Byte (Eingang, Ausgang) Wort (Eingang, Ausgang) Doppelwort (Eingang, Ausgang) Hinweis: Die Ausgangsoperanden werden beim Rücksprung vom Unterprogramm verarbeitet.	15 32 23, 21 21, 14 24, 18 27, 20
CFND	Max. Zeit = Grundausführungszeit + (N1-N2) * ((LM1*N2)+LM2) Grundausführungszeit Längenmultiplikator 1 (LM1) Längenmultiplikator 2 (LM2) N1 ist die Länge der Quell-Zeichenkette N2 ist die Länge des Zeichensatzes Zeichenkette	79 79 9,2 4,4
COS		1525 1800 (max.)
CRET		13
CRETI		23
CSCRE		0,9
DECB		30
DECD		42
DECO		36
DECW		37
DISI		18
DIV		119
DTA		540
DTCH		18
DTI		36
DTR		60 70 (max.)
DTS		540
ED		15
ENCO		39 43 (max.)
END		0,9
ENI		53
EU		15
EXP		1170 1375 (max.)

F

Operation			µs
FIFO	Zeit = Grundausführungszeit + (Länge ∗ LM) Grundausführungszeit Längenmultiplikator (LM)		 70 14
FILL	Zeit = Grundausführungszeit + (Länge ∗ LM) Grundausführungszeit (konstante Länge) Grundausführungszeit (variable Länge) Längenmultiplikator (LM)		 29 50 7
FND <, =, >, <>	Zeit = Grundausführungszeit + (Länge ∗ LM) Grundausführungszeit Längenmultiplikator (LM)		 85 12
FOR	Zeit = Grundausführungszeit + (Schleifenzahl ∗ LM) Grundausführungszeit Schleifenmultiplikator (LM)		 64 50
GPA			31
HDEF			35
HSC			37
HTA	Zeit = Grundausführungszeit + (Länge ∗ LM) Grundausführungszeit (konstante Länge) Grundausführungszeit (variable Länge) Längenmultiplikator (LM)		 38 48 11
IBCD			114
INCB			29
INCD			42
INCW			37
INT	Typisch bei 1 Interrupt		47
INVB			31
INVD			42
INVW			38
ITA			260
ITB			27
ITD			36
ITS			260
JMP			0,9
LBL			0,37
LD	Verwendet:	E, SM0.0 SM, T, Z, V, S, A, M L	0,37 1,1 10,9
LDB <=, =, >=, >, <, <>			35
LDD <=, =, >=, >, <, <>			52
LDI	Verwendet:	Integrierte Eingänge Erweiterungsausgänge	26 34
LDN	Verwendet:	E SM, T, Z, V, S, A, M L	0,37 1,1 10,9
LDNI	Verwendet:	Integrierte Eingänge Erweiterungsausgänge	26 34
LDR <=, =, >=, >, <, <>			55
LDS			0,37

Operation			µs
LDS=, <>	Zeit = Grundausführungszeit + (LM * N) Grundausführungszeit Längenmultiplikator (LM) N ist die verglichene Zeichenzahl		 51 9,2
LDW <=, =, >=, >, <, <>			42
LIFO			70
LN			1130 1275 (max.)
LPP			0,37
LPS			0,37
LRD			0,37
LSCR			12
MEND			0,5
MOVB			29
MOVD			38
MOVR			38
MOVW			34
MUL			70
NETR			179
NETW	Zeit = Grundausführungszeit + (Länge ∗ LM) Grundausführungszeit Längenmultiplikator (LM)		 175 8
NEXT			0
NOP			0,37
NOT			0,37
O	Verwendet:	E SM, T, Z, V, S, A, M L	0,37 1,1 10,8
OB <=, =, >=, >, <, <>			35
OD <=, =, >=, >, <, <>			53
OI	Verwendet:	Integrierte Eingänge Erweiterungsausgänge	27 35
OLD			0,37
ON	Verwendet:	E SM, T, Z, V, S, A, M L	0,37 1,1 10,8
ONI	Verwendet:	Integrierte Eingänge Erweiterungsausgänge	27 35
OR <=, =, >=, >, <, <>			55
ORB			37
ORD			55
ORW			48
OS=, < >	Zeit = Grundausführungszeit + (LM * N) Grundausführungszeit Längenmultiplikator (LM) N ist die verglichene Zeichenzahl		 51 9,2
OW <=, =, >=, >, <, <>			45

F

Operation		µs
PID	Grundausführungszeit Zu addierender Wert für Neuberechnung der Integral- und proportionalen Differentialkonstanten	750 1000
PLS:	Verwendet: PWM PTO, Einzel-Segment-Betrieb PTO, Multi-Segment-Betrieb	57 67 92
R	Länge = 1 und als Konstante angegeben Mit Operand = Z, T Mit allen anderen Operanden Sonst: Zeit = Grundausführungszeit + (Länge ∗ LM) Grundausführungszeit bei Operand = Z, T Grundausführungszeit bei allen anderen Operanden LM bei Operand = Z, T LM bei allen anderen Operanden Wenn Länge als Variable gespeichert, zur Grundausführungszeit addieren	 17, 24 5 19, 19 28 8,6, 16,5 0,9 29
RCV		80
RET		13
RETI		23
RI	Gesamt = Grundausführungszeit + (Länge ∗ LM) Grundausführungszeit LM bei integrierten Ausgängen LM bei Erweiterungsausgängen Wenn Länge als Variable gespeichert, zur Grundausführungszeit addieren	 18 22 32 30
RLB	Gesamt = Grundausführungszeit + (Länge + LM) Grundausführungszeit Längenmultiplikator (LM)	 42 0,6
RLD	Gesamt = Grundausführungszeit + (Länge ∗ LM) Grundausführungszeit Längenmultiplikator (LM))	 52 2,5
RLW	Gesamt = Grundausführungszeit + (Länge ∗ LM) Grundausführungszeit Längenmultiplikator (LM))	 49 1,7
ROUND		108 183 (max.)
RRB	Gesamt = Grundausführungszeit + (Länge ∗ LM) Grundausführungszeit Längenmultiplikator (LM))	 42 0,6
RRD	Gesamt = Grundausführungszeit + (Länge ∗ LM) Grundausführungszeit Längenmultiplikator (LM))	 52 2,5
RRW	Gesamt = Grundausführungszeit + (Länge ∗ LM) Grundausführungszeit Längenmultiplikator (LM))	 49 1,7
RTA	Zeit = Grundausführungszeit + (LM * N) Grundausführungszeit (für erste Ziffer im Ergebnis) Längenmultiplikator (LM) N ist die Anzahl der zusätzlichen Ziffern im Ergebnis	 1000 240

Operation		µs
RTS	Zeit = Grundausführungszeit + (LM * N) Grundausführungszeit (für erste Ziffer im Ergebnis) Längenmultiplikator (LM) N ist die Anzahl der zusätzlichen Ziffern im Ergebnis	 1000 240
S	Länge = 1 und angegeben als Konstante Sonst: Zeit = Grundausführungszeit + (Länge ∗ LM) Grundausführungszeit Längenmultiplikator (LM) Wenn Länge als Variable gespeichert, zur Grundausführungszeit addieren	5 27 0,9 29
SBR		0
SCAT	Zeit = Grundausführungszeit + (LM * N) Grundausführungszeit Längenmultiplikator (LM) N ist die Anzahl der angehängten Zeichen	 55 8,8
SCPY	Zeit = Grundausführungszeit + (LM * N) Grundausführungszeit Längenmultiplikator (LM) N ist die Anzahl der kopierten Zeichen	 43 8,8
SCRE		0,37
SCRT		17
SEG		30
SFND	Max. Zeit = Grundausführungszeit + (N1-N2) * ((LM1*N2)+LM2) Grundausführungszeit Längenmultiplikator 1 (LM1) Längenmultiplikator 2 (LM2) N1 ist die Länge der Quell-Zeichenkette N2 ist die Länge der Such-Zeichenkette	 79 11,5 17,8
SHRB	Gesamt = Grundausführungszeit + (Länge ∗ LM1) + ((Länge /8) * LM2) Grundausführungszeit (konstante Länge) Grundausführungszeit (variable Länge) Längenmultiplikator 1 (LM1) Längenmultiplikator 2 (LM2)	 76 84 1,6 4
SI	Gesamt = Grundausführungszeit + (Länge ∗ LM) Grundausführungszeit LM bei integriertem Ausgang LM bei Erweiterungsausgang Wenn Länge als Variable gespeichert, zur Grundausführungszeit addieren	 18 22 32 30

F

SIN		1525 1800 (max.)
SLB	Gesamt = Grundausführungszeit + (Länge ∗ LM) Grundausführungszeit Längenmultiplikator (LM))	 43 0,7
SLD	Gesamt = Grundausführungszeit + (Länge ∗ LM) Grundausführungszeit Längenmultiplikator (LM))	 53 2,6
Operation		**µs**
SLEN		46
SLW	Gesamt = Grundausführungszeit + (Länge ∗ LM) Grundausführungszeit Längenmultiplikator (LM))	 51 1,3
SPA		243
SQRT		725 830 (max.)
SRB	Gesamt = Grundausführungszeit + (Länge ∗ LM) Grundausführungszeit Längenmultiplikator (LM))	 43 0,7
SRD	Gesamt = Grundausführungszeit + (Länge ∗ LM) Grundausführungszeit Längenmultiplikator (LM))	 53 2,6
SRW	Gesamt = Grundausführungszeit + (Länge ∗ LM) Grundausführungszeit Längenmultiplikator (LM))	 51 1,3
SSCPY	Zeit = Grundausführungszeit + (LM * N) Grundausführungszeit Längenmultiplikator (LM) N ist die Anzahl der kopierten Zeichen	 82 8,8
STD	Zeit = Grundausführungszeit + (LM*N) Grundausführungszeit (für erstes Quellzeichen) Längenmultiplikator (LM) N ist die Anzahl der zusätzlichen Quellzeichen	 84 59
STI	Zeit = Grundausführungszeit + (LM*N) Grundausführungszeit (für erstes Quellzeichen) Längenmultiplikator (LM) N ist die Anzahl der zusätzlichen Quellzeichen	 84 59
STOP		16
STR	Zeit = Grundausführungszeit + (LM*N) Grundausführungszeit (für erstes Quellzeichen) Längenmultiplikator (LM) N ist die Anzahl der zusätzlichen Quellzeichen	 100 120
SWAP		32
TAN		1825 2100 (max.)
TODR		2400

Operation			**µs**
TODW			1600
TOF			64
TON			64
TONR			56
TRUNC			103 178 (max.)
U	Verwendet:	E SM, T, Z, V, S, A, M L	0,37 1,1 10,8
UB < =, =, >=, >, <, <>			35
UD < =, =, >=, >, <, <>			53
UENO			0,6
UI	Verwendet:	Integrierte Eingänge Erweiterungsausgänge	27 35
ULD			0,37
UN	Verwendet:	E SM, T, Z, V, S, A, M L	0,37 1,1 10,8
UNDB			37
UNDD			55
UNDW			48
UNI	Verwendet:	Integrierte Eingänge Erweiterungsausgänge	27 35
UR <=, =, >=, >, <, <>			54
US=, <>	Zeit = Grundausführungszeit + (LM * N) Grundausführungszeit Längenmultiplikator (LM) N ist die verglichene Zeichenzahl		 51 9,2
UW <=, =, >=, >, <, <>			45
WDR			16
XMT			78
XORB			37
XORD			55
XORW			48
ZR	Beim Wechsel des Zähleingangs Sonst		48 36
ZV	Beim Wechsel des Zähleingangs Sonst		53 35
ZVR	Beim Wechsel des Zähleingangs Sonst		64 45

F

S7-200 Kurzinformation

Damit Sie bestimmte Informationen leichter finden können, werden in diesem Abschnitt die folgenden Informationen zusammengefasst:

- Sondermerker
- Beschreibung der Interruptereignisse
- Zusammenfassung der Speicherbereiche und Funktionen der S7-200 CPU
- Schnelle Zähler HSC0, HSC1, HSC2, HSC3, HSC4, HSC5
- S7-200 Operationen

Tabelle G-1 Sondermerker

Sondermerker			
SM0.0	Immer ein	SM1.0	Ergebnis der Operation = 0
SM0.1	Erster Zyklus	SM1.1	Überlauf oder unzulässiger Wert
SM0.2	Remanente Daten verloren	SM1.2	Negatives Ergebnis
SM0.3	Anlauf	SM1.3	Division durch 0
SM0.4	30 s ein / 30 s aus	SM1.4	Tabelle voll
SM0.5	0,5 s ein / 0,5 s aus	SM1.5	Tabelle leer
SM0.6	1 Zyklus ein / 1 Zyklus aus	SM1.6	Fehler beim Umwandeln BCD - Binär
SM0.7	Schalter in RUN	SM1.7	Fehler beim Umwandeln ASCII - Hex

Tabelle G-2 Interruptereignisse nach Prioritäten

Nummer des Ereignisses	Beschreibung des Interrupts	Prioritätsklasse	Priorität in Gruppen
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
19	Interrupt PTO0 beendet	E/A-Interrupts: mittlere Prioritätsklasse	0
20	Interrupt PTO1 beendet		1
0	E0.0, steigende Flanke		2
2	E0.1, steigende Flanke		3
4	E0.2, steigende Flanke		4
6	E0.3, steigende Flanke		5
1	E0.0, fallende Flanke		6
3	E0.1, fallende Flanke		7
5	E0.2, fallende Flanke		8
7	E0.3, fallende Flanke		9
12	HSC0: CV=PV (aktueller Wert = voreingestellter Wert)*		10
27	HSC0 Richtungswechsel		11
28	HSC0: externes Rücksetzen		12
13	HSC1: CV=PV (aktueller Wert = voreingestellter Wert)*		13
14	HSC1: Richtungswechsel		14
15	HSC1: externes Rücksetzen		15
16	HSC2: CV=PV		16
17	HSC2 Richtungswechsel		17
18	HSC2: externes Rücksetzen		18
32	HSC3: CV = PV (aktueller Wert = voreingestellter Wert)*		19
29	HSC4: CV = PV (aktueller Wert = voreingestellter Wert)*		20
30	HSC4 Richtungswechsel		21
31	HSC4: externes Rücksetzen		22
33	HSC5: CV = PV (aktueller Wert = voreingestellter Wert)*		23
10	Zeitgesteuerter Interrupt 0	Zeitgesteuerte Interrupts: niedrigste Prioritätsklasse	0
11	Zeitgesteuerter Interrupt 1		1
21	Zeit T32: Interrupt CT=PT		2
22	Zeit T96: Interrupt CT=PT		3

Tabelle G-3 Zusammenfassung der Speicherbereiche und Funktionen der S7-200 CPU

Beschreibung	CPU 221	CPU 222	CPU 224	CPU 226	CPU 226XM
Größe Anwenderprogramm	4096 Bytes	4096 Bytes	8192 Bytes	8192 Bytes	16384 Bytes
Größe Anwenderdaten	2048 Bytes	2048 Bytes	5120 Bytes	5120 Bytes	10240 Bytes
Prozessabbild der Eingänge	E0.0 bis E15.7	E0.0 bis E15.7	E0.0 bis E15.7	E0.0 bis E15.7	E0.0 bis E15.7
Prozessabbild der Ausgänge	A0.0 bis A15.7	A0.0 bis A15.7	A0.0 bis A15.7	A0.0 bis A15.7	A0.0 bis A15.7
Analogeingänge (nur Lesen)	--	AEW0 bis AEW30	AEW0 bis AEW62	AEW0 bis AEW62	AEW0 bis AEW62
Analogausgänge (nur Schreiben)	--	AAW0 bis AAW30	AAW0 bis AAW62	AAW0 bis AAW62	AAW0 bis AAW62
Variablenspeicher (V)	VB0 bis VB2047	VB0 bis VB2047	VB0 bis VB5119	VB0 bis VB5119	VB0 bis VB10239
Lokaldatenspeicher (L)[1]	LB0 bis LB63	LB0 bis LB63	LB0 bis LB63	LB0 bis LB63	LB0 bis LB63
Merker (M)	M0.0 bis M31.7	M0.0 bis M31.7	M0.0 bis M31.7	M0.0 bis M31.7	M0.0 bis M31.7
Sondermerker (SM)	SM0.0 bis SM179.7	SM0.0 bis SM299.7	SM0.0 bis SM549.7	SM0.0 bis SM549.7	SM0.0 bis SM549.7
Schreibgeschützt	SM0.0 bis SM29.7	SM0.0 bis SM29.7	SM0.0 bis SM29.7	SM0.0 bis SM29.7	SM0.0 bis SM29.7
Zeiten	256 (T0 bis T255)	256 (T0 bis T255)	256 (T0 bis T255)	256 (T0 bis T255)	256 (T0 bis T255)
Speich. Einschaltverz. 1 ms	T0, T64	T0, T64	T0, T64	T0, T64	T0, T64
10 ms	T1 bis T4 und T65 bis T68	T1 bis T4 und T65 bis T68	T1 bis T4 und T65 bis T68	T1 bis T4 und T65 bis T68	T1 bis T4 und T65 bis T68
100 ms	T5 bis T31 und T69 bis T95	T5 bis T31 und T69 bis T95	T5 bis T31 und T69 bis T95	T5 bis T31 und T69 bis T95	T5 bis T31 und T69 bis T95
Ein-/Ausschaltverzög. 1 ms	T32, T96	T32, T96	T32, T96	T32, T96	T32, T96
10 ms	T33 bis T36 und T97 bis T100	T33 bis T36 und T97 bis T100	T33 bis T36 und T97 bis T100	T33 bis T36 und T97 bis T100	T33 bis T36 und T97 bis T100
100 ms	T37 bis T63 und T101 bis T255	T37 bis T63 und T101 bis T255	T37 bis T63 und T101 bis T255	T37 bis T63 und T101 bis T255	T37 bis T63 und T101 bis T255
Zähler	Z0 bis Z255	Z0 bis Z255	Z0 bis Z255	Z0 bis Z255	Z0 bis Z255
Schnelle Zähler	HC0, HC3, HC4 und HC5	HC0, HC3, HC4 und HC5	HC0 bis HC5	HC0 bis HC5	HC0 bis HC5
Ablaufsteuerungsrelais (S)	S0.0 bis S31.7	S0.0 bis S31.7	S0.0 bis S31.7	S0.0 bis S31.7	S0.0 bis S31.7
Akkumulatoren	AC0 bis AC3	AC0 bis AC3	AC0 bis AC3	AC0 bis AC3	AC0 bis AC3
Sprünge/Sprungmarken	0 bis 255	0 bis 255	0 bis 255	0 bis 255	0 bis 255
Aufrufe/Unterprogramme	0 bis 63	0 bis 63	0 bis 63	0 bis 63	0 bis 127
Interruptprogramme	0 bis 127	0 bis 127	0 bis 127	0 bis 127	0 bis 127
Steigende/fallende Flanken	256	256	256	256	256
PID-Regler	0 bis 7	0 bis 7	0 bis 7	0 bis 7	0 bis 7
Anschlüsse	Schnittstelle 0	Schnittstelle 0	Schnittstelle 0	Schnittstelle 0, Schnittstelle 1	Schnittstelle 0, Schnittstelle 1

[1] LB60 bis LB63 sind für STEP 7-Micro/WIN ab Version 3.0 reserviert.

Tabelle G-4 Schnelle Zähler HSC0, HSC3, HSC4 und HSC5

Zählerart	HSC0			HSC3	HSC4			HSC5
	E0.0	E0.1	E0.2	E0.1	E0.3	E0.4	E0.5	E0.4
0	Taktgeber			Taktgeber	Taktgeber			Taktgeber
1	Taktgeber		Rücksetzen		Taktgeber		Rücksetzen	
2								
3	Taktgeber	Richtung			Taktgeber	Richtung		
4	Taktgeber	Richtung	Rücksetzen		Taktgeber	Richtung	Rücksetzen	
5								
6	Taktgeber: vorwärts	Taktgeber: rückwärts			Taktgeber: vorwärts	Taktgeber: rückwärts		
7	Taktgeber: vorwärts	Taktgeber: rückwärts	Rücksetzen		Taktgeber: vorwärts	Taktgeber: rückwärts	Rücksetzen	
8								
9	Phase A	Phase B			Phase A	Phase B		
10	Phase A	Phase B	Rücksetzen		Phase A	Phase B	Rücksetzen	
11								

Tabelle G-5 Schnelle Zähler HSC1 und HSC2

Zählerart	HSC1				HSC2			
	E0.6	E0.7	E1.0	E1.1	E1.2	E1.3	E1.4	E1.5
0	Taktgeber				Taktgeber			
1	Taktgeber		Rücksetzen		Taktgeber		Rücksetzen	
2	Taktgeber		Rücksetzen	Starten	Taktgeber		Rücksetzen	Starten
3	Taktgeber	Richtung			Taktgeber	Richtung		
4	Taktgeber	Richtung	Rücksetzen		Taktgeber	Richtung	Rücksetzen	
5	Taktgeber	Richtung	Rücksetzen	Starten	Taktgeber	Richtung	Rücksetzen	Starten
6	Taktgeber: vorwärts	Taktgeber: rückwärts			Taktgeber: vorwärts	Taktgeber: rückwärts		
7	Taktgeber: vorwärts	Taktgeber: rückwärts	Rücksetzen		Taktgeber: vorwärts	Taktgeber: rückwärts	Rücksetzen	
8	Taktgeber: vorwärts	Taktgeber: rückwärts	Rücksetzen	Starten	Taktgeber: vorwärts	Taktgeber: rückwärts	Rücksetzen	Starten
9	Phase A	Phase B			Phase A	Phase B		
10	Phase A	Phase B	Rücksetzen		Phase A	Phase B	Rücksetzen	
11	Phase A	Phase B	Rücksetzen	Starten	Phase A	Phase B	Rücksetzen	Starten

Boolesche Operationen		
LD	Bit	Laden
LDI	Bit	Direkt Laden
LDN	Bit	Negiert Laden
LDNI	Bit	Direkt Negiert Laden
U	Bit	UND
UI	Bit	Direkt UND
UN	Bit	Negiert UND
UNI	Bit	Direkt Negiert UND
O	Bit	ODER
OI	Bit	Direkt ODER
ON	Bit	Negiert ODER
ONI	Bit	Direkt Negiert ODER
LDBx	IN1, IN2	Ergebnis von Bytevergleich laden IN1 (x:<, <=,=, >=, >, <>I) IN2
UBx	IN1, IN2	UND Ergebnis von Bytevergleich IN1 (x:<, <=,=, >=, >, <>) IN2
OBx	IN1, IN2	ODER Ergebnis von Bytevergleich IN1 (x:<, <=,=, >=, >, <>) IN2
LDWx	IN1, IN2	Ergebnis von Wortvergleich laden IN1 (x:<, <=,=, >=, >, <>) IN2
UWx	IN1, IN2	UND Ergebnis von Wortvergleich IN1 (x:<, <=,=, >=, >, <>) IN2
OWx	IN1, IN2	ODER Ergebnis von Wortvergleich IN1 (x:<, <=,=, >=, >, <>) IN2
LDDx	IN1, IN2	Ergebnis von Doppelwortvergleich laden IN1 (x:<, <=,=, >=, >, <>) IN2
UDx	IN1, IN2	UND Ergebnis von Doppelwortvergleich IN1 (x:<, <=,=, >=, >, <>) IN2
ODx	IN1, IN2	ODER Ergebnis von Doppelwortvergleich IN1 (x:<, <=,=, >=, >, <>) IN2
LDRx	IN1, IN2	Ergebnis von Realzahlenvergleich laden IN1 (x:<, <=,=, >=, >, <>) IN2
URx	IN1, IN2	UND Ergebnis von Realzahlenvergleich IN1 (x:<, <=,=, >=, >, <>) IN2
ORx	IN1, IN2	ODER Ergebnis von Realzahlenvergleich IN1 (x:<, <=,=, >=, >, <>) IN2

NOT		Stack negieren
EU		Erkennung steigende Flanke
ED		Erkennung fallende Flanke
=	Bit	Wert zuweisen
=I	Bit	Wert direkt zuweisen
S	Bit, N	Bitbereich setzen
R	Bit, N	Bitbereich rücksetzen
SI	Bit, N	Bitbereich direkt setzen
RI	Bit, N	Bitbereich direkt rücksetzen
LDSx	IN1, IN2	Ergebnis von Zeichenkettenvergl. laden IN1 (x: =, <>) IN2
USx	IN1, IN2	UND Ergebnis von Zeichenkettenvergleich IN1 (x: =, <>) IN2
OSx	IN1, IN2	ODER Ergebnis von Zeichenkettenvergl. IN1 (x: =, <>) IN2
ULD		Erste und zweite Stackebene durch UND verknüpfen
OLD		Erste und zweite Stackebene durch ODER verknüpfen
LPS		Obersten Stackwert duplizieren
LRD		Zweiten Stackwert kopieren
LPP		Obersten Stackwert aus Stack schieben
LDS	N	Stack laden
UENO		UND ENO

Arithmetische Operationen, Inkrementieren und Dekrementieren		
+I	IN, OUT	Ganze Zahlen (16 Bit) addieren, Ganze Zahlen (32 Bit) addieren oder Realzahlen addieren IN1+OUT=OUT
+D	IN, OUT	
+R	IN, OUT	
-I	IN, OUT	Ganze Zahlen (16 Bit) subtrahieren, Ganze Zahlen (32 Bit) subtrahieren oder Realzahlen subtrahieren OUT-IN1=OUT
-D	IN, OUT	
-R	IN, OUT	
MUL	IN, OUT	Ganze Zahlen (16 Bit) multiplizieren (16*16->32)
*I	IN, OUT	Ganze Zahlen (16 Bit) multiplizieren, Ganze Zahlen (32 Bit) multiplizieren oder Realzahlen multiplizieren IN1 * OUT = OUT
*D	IN, OUT	
*R	IN1, IN2	
DIV	IN, OUT	Ganze Zahlen (16 Bit) dividieren (16/16->32)
/I	IN, OUT	Ganze Zahlen (16 Bit) dividieren, Ganze Zahlen (32 Bit) dividieren oder Realzahlen dividieren OUT / IN1 = OUT
/D,	IN, OUT	
/R	IN, OUT	
SQRT	IN, OUT	Quadratwurzel einer Realzahl ziehen
LN	IN, OUT	Natürlicher Logarithmus
EXP	IN, OUT	Natürlicher Exponent
SIN	IN, OUT	Sinus
COS	IN, OUT	Cosinus
TAN	IN, OUT	Tangens
INCB	OUT	Byte, Wort oder Doppelwort um 1 erhöhen
INCW	OUT	
INCD	OUT	
DECB	OUT	Byte, Wort oder Doppelwort um 1 verringern
DECW	OUT	
DECD	OUT	
PID	TBL, LOOP	PID-Regler
Zeit- und Zähloperationen		
TON	Txxx, PT	Einschaltverzögerung
TOF	Txxx, PT	Ausschaltverzögerung
TONR	Txxx, PT	Speichernde Einschaltverzögerung
ZV	Zxxx, PV	Vorwärtszählen
ZR	Zxxx, PV	Rückwärtszählen
ZVR	Zxxx, PV	Vorwärts-/Rückwärtszählen
Echzeituhr-Operationen		
TODR	T	Echtzeituhr lesen
TODW	T	Echtzeituhr schreiben
Operationen für die Programmsteuerung		
END		Bearbeitung bedingt beenden
STOP		In STOP versetzen
WDR		Überwachungszeit rücksetzen (300 ms)
JMP	N	Zu Sprungmarke springen
LBL	N	Sprungmarke definieren
CALL	N [N1,...]	Unterprogramm aufrufen [N1, ... maximal 16 optionale Parameter]
CRET		Unterprogramm bedingt beenden
FOR	INDX,INIT,FINAL	Schleife mit FOR/NEXT
NEXT		
LSCR	N	Ablaufsteuerungsrelais laden, Transition Ablaufsteuerungsrelais, Ablaufsteuerungsrelais bedingt beenden und Ende Ablaufsteuerungsrelais
SCRT	N	
CSCRE		
SCRE		

Übertragungs-, Schiebe- und Rotieroperationen		
MOVB	IN, OUT	Byte, Wort, Doppelwort, Realzahl übertragen
MOVW	IN, OUT	
MOVD	IN, OUT	
MOVR	IN, OUT	
BIR	IN, OUT	Byte direkt lesen und übertragen
BIW	IN, OUT	Byte direkt schreiben und übertragen
BMB	IN, OUT, N	Bereich Byte, Wort, Doppelwort, Realzahl übertragen
BMW	IN, OUT, N	
BMD	IN, OUT, N	
SWAP	IN	Bytes im Wort tauschen
SHRB	DATA, S_BIT, N	Wert in Schieberegister schieben
SRB	OUT, N	Byte, Wort, Doppelwort rechts schieben
SRW	OUT, N	
SRD	OUT, N	
SLB	OUT, N	Byte, Wort, Doppelwort links schieben
SLW	OUT, N	
SLD	OUT, N	
RRB	OUT, N	Byte, Wort, Doppelwort rechts rotieren
RRW	OUT, N	
RRD	OUT, N	
RLB	OUT, N	Byte, Wort, Doppelwort links rotieren
RLW	OUT, N	
RLD	OUT, N	
Verknüpfungsoperationen		
UNDB	IN, OUT	Byte, Wort, Doppelwort durch UND verknüpfen
UNDW	IN, OUT	
UNDD	IN, OUT	
ORB	IN, OUT	Byte, Wort, Doppelwort durch ODER verknüpfen
ORW	IN, OUT	
ORD	IN, OUT	
XORB	IN, OUT	Byte, Wort, Doppelwort durch XOR verknüpfen
XORW	IN, OUT	
XORD	IN, OUT	
INVB	OUT	Byte, Wort, Doppelwort invertieren (Einerkomplement bilden)
INVW	OUT	
INVD	OUT	
Zeichenkettenoperationen		
SLEN	IN, OUT	Zeichenkettenlänge
SCAT	IN, OUT	Zeichenkette verketten
SCPY	IN, OUT	Zeichenkette kopieren
SSCPY	IN, INDX, N, OUT	Teilzeichenkette aus Zeichenkette kop.
CFND	IN1, IN2, OUT	Erstes Zeichen in Zeichenkette suchen
SFND	IN1, IN2, OUT	Zeichenkette in Zeichenkette suchen

G

Tabellen-, Such- und Umwandlungsoperationen		
ATT	DATA, TBL	Wert in Tabelle eintragen
LIFO	TBL, DATA	Wert aus Tabelle holen
FIFO	TBL, DATA	
FND=	TBL, PTN, INDX	Datenwert in Tabelle suchen, der dem Vergleich entspricht
FND<>	TBL, PTN, INDX	
FND<	TBL, PTN, INDX	
FND>	TBL, PTN, INDX	
FILL	IN, OUT, N	Speicher mit Bitmuster belegen
BCDI	OUT	BCD in ganze Zahl wandeln
IBCD	OUT	Ganze Zahl in BCD wandeln
BTI	IN, OUT	Byte in ganze Zahl wandeln
ITB	IN, OUT	Ganze Zahl in Byte wandeln
ITD	IN, OUT	Ganze Zahl (16 Bit) in ganze Zahl (32 Bit) wandeln
DTI	IN, OUT	Ganze Zahl (32 Bit) in ganze Zahl (16 Bit) wandeln
DTR	IN, OUT	Doppelwort in Realzahl wandeln
TRUNC	IN, OUT	Realzahl in ganze Zahl (32 Bit) wandeln
ROUND	IN, OUT	Realzahl in ganze Zahl (32 Bit) wandeln
ATH	IN, OUT, LEN	ASCII-Zeichenkette in Hexadezimalzahl wandeln
HTA	IN, OUT, LEN	Hexadezimalzahl in ASCII-Zeichenkette wandeln
ITA	IN, OUT, FMT	Ganze Zahl in ASCII-Zeichenkette wandeln
DTA	IN, OUT, FM	Ganze Zahl (32 Bit) in ASCII-Zeichenkette wandeln
RTA	IN, OUT, FM	Realzahl in ASCII-Zeichenkette wandeln
DECO	IN, OUT	Bit in Hexadezimalzahl wandeln
ENCO	IN, OUT	Hexadezimalzahl in Bit wandeln
SEG	IN, OUT	Muster für Sieben-Segment-Anzeige erzeugen
ITS	IN, FMT, OUT	Ganze Zahl in Zeichenkette wandeln
DTS	IN, FMT, OUT	Ganze Zahl (32 Bit) in Zeichenk. wandeln
RTS	IN, FMT, OUT	Realzahl in Zeichenkette wandeln
STI	STR, INDX, OUT	Teilzeichenkette in ganze Zahl wandeln
STD	STR, INDX, OUT	Teilzeichenk. in ganze Zahl (32 Bit) wandeln
STR	STR, INDX, OUT	Teilzeichenkette in Realzahl wandeln
Interruptoperationen		
CRETI		Interruptprogramm bedingt beenden
ENI		Alle Interruptereignisse freigeben
DISI		Alle Interruptereignisse sperren
ATCH	INT, EVNT	Interrupt zuordnen
DTCH	EVNT	Ereignis trennen
Kommunikationsoperationen		
XMT	TBL, PORT	Senden in frei programm. Kommunikation
RCV	TBL, PORT	Empfangen in frei programm. Komm.unik.
NETR	TBL, PORT	Aus Netz lesen
NETW	TBL, PORT	In Netz schreiben
GPA	ADDR,PORT	Schnittstellenadresse holen
SPA	ADDR,PORT	Schnittstellenadresse einstellen
Schnelle Zähler		
HDEF	HSC, MODE	Modus für schnellen Zähler definieren
HSC	N	Schnellen Zähler aktivieren
PLS	A	Impulsausgabe

G

Index

Symbols

&, 32
*, 32

A

AA. *Siehe* Analogausgang (AA)
Abbilden von Adressen in der S7-200, Modbus, 341
Ablaufsteuerungsrelais (S), 29
Ablaufsteuerungsrelais bedingt beenden, 174
Ablaufsteuerungsrelais laden, 174
Ablegen, S7-200 Programmdaten, 34
Abmessungen, CPU, 2
Abschließen, Netzwerkkabel, 222
Absolute Eingänge, 55
AC-Ausgänge und Relais, 20
ACCEL_TIME (Beschleunigungszeit), Positioniermodul EM 253, 257
Addieren, 141
Adresse
  Einstellen der entfernten Adresse, 211
  Modbus, 341
  Netz, 209
  Schnelle Zähler, 120
  Symbolisch, 58
  Zuordnen, 58
Adressierung
  Ablaufsteuerungsrelais (SCR), 29
  Akkumulatoren, 27
  Analogausgänge, 29
  Analogeingänge, 29
  Byte:Bit, 24
  Direkt, 24
  Erweiterungs-E/A, 31
  Indirekt (Pointer), 32
  Integrierte Ein- und Ausgänge, 31
  Lokaldatenspeicher, 28
  Merker, 25
  Prozessabbild der Ausgänge, 25
  Prozessabbild der Eingänge, 25
  S7-200 Speicher, 24
  Schnelle Zähler, 27
  Sondermerker (SM), 28
  Speicherbereiche, 25–28
  Variablenspeicher, 25
  Zähler, 26
  Zeiten, 26
Adressschalter, EM 277 PROFIBUS-DP, 386
AE. *Siehe* Analogeingang (AE)
Akkumulatoren, 27
Aktivierungspegel Eingang, Positioniermodul EM 253, 255
Aktualisieren, PWM-Wellenform, 128
Aktueller Wert
  Ändern in HSC, 124
  Einstellen für HSC, 119
Alarmfunktionen, PID-Regler, 152
Alle Interruptereignisse freigeben, 156
Alle Interruptereignisse sperren, 156
American Bureau of Shipping (ABS), 348
Analogausgang (AA), Adressierung, 29
Analogeingang (AE)
  Adressierung, 29
  Filter, 42
Analogmodule, 3
  Analogeingang EM 231, 366
  EM 231 RTD, 373
  EM 231 Thermoelement, 373
  EM 232 Analogausgabe, 370
  EM 235 Analogein-/Analogausgabe, 367
Analogpotentiometer
  Potentiometer, 45
  SMB28 und SMB29, 432
Ändern
  Aktueller Wert in HSC, 124
  Impulsdauer, 133
  Pointer, 32
  PTO-Impulszählwert, 136
  PTO-Zykluszeit, 135
  PTO-Zykluszeit und Impulszählwert, 136
  Voreingestellter Wert in HSC, 124
Anfangswerte, Zuordnen, 58
Anforderungen, Modbus-Slave-Protokoll, 340
Anlauf, Datenhaltung, 35–37
Anlegen
  Liste mit symbolischen Namen, 49
  Programm, 9
  Programme mit STEP 7-Micro/WIN, 51
Anschließen
  an die S7-200, 8
  Funkmodem an S7-200, 232
  MM3-Antrieb, 332
  MM4-Antrieb, 335
  Modem an S7-200, 231
  Multi-Master-Kabel, 7
  Netzgeräte, 235
Anschlussbelegung, Kommunikationsschnittstelle, 221
Anschlüsse
  CP 243-2, 412
  CPU 221 AC/DC/Relais, 354
  CPU 221 DC/DC/DC, 354
  CPU 222 AC/DC/Relais, 355
  CPU 222 DC/DC/DC, 355
  CPU 224 AC/DC/Relais, 355
  CPU 224 DC/DC/DC, 355
  CPU 226 AC/DC/Relais, 356
  CPU 226 DC/DC/DC, 356
  CPU 226SM AC/DC/Relais, 356
  CPU 226XM DC/DC/DC, 356
  EM 221 DI 8 x AC, 361
  EM 221 DI 8x24 VDC, 361
  EM 222 DO 8 x Relais, 361
  EM 222 DO 8x24 VDC, 361
  EM 223 4x24 VDC In/4x24 VDC Out, 360
  EM 223 DI 16/DO 16 x 24 V DC Relais, 362
  EM 223 DI 16/DO 16 x DC 24V, 362
  EM 223 DI 4/DO 4 x DC 24V/Relais, 360, 361
  EM 223 DI 8/ DO 8 x 24 V DC/Relais, 362
  EM 223 DI 8/DO 8 x 24 V DC, 362
  EM 231 AI 4 x 12 Bit, 364
  EM 231 RTD, 374
  EM 231 Thermoelement, 374
  EM 232 AQ 2 x 12 Bit, 364
  EM 235 AI 4/AQ 1 x 12 Bit, 364

Anschlüsse (Fortsetzung)
Ethernet-Modul CP 243-1, 407
Internet-Modul CP 243-1 IT, 410
Kommunikationsschnittstelle, 221
Ansicht, Fehler, 56
Antriebe. *Siehe* MicroMaster-Antriebe
Anweisungsliste. *Siehe* AWL-Editor
Anwenderabbruch, 87
Anwenderbibliotheken, 60
Anwenderdefiniertes Protokoll, Frei programmierbare Kommunikation, 225
Anzahl an Doppelwörtern übertragen, 168
Anzahl an Wörtern übertragen, 168
Anzeige-Panels, TP070 Touch Panel, 4
Anzeigegeräte, TD 200 Textdisplay, 4
Anzeigen
Programmelemente, 51
Programmstatus, 246
Arithmetische Operationen
Addieren, 141
Dekrementieren, 145
Dividieren, 141
Ganze Zahlen (16 Bit) dividieren mit Divisionsrest (DIV), 143
Ganze Zahlen (16 Bit) in ganze Zahl (32 Bit) multiplizieren (MUL), 143
Inkrementieren, 145
Multiplizieren, 141
Subtrahieren, 141
AS-i-Assistent, 411
AS-Interface-Modul CP 243-2
Bestellnummer, 411, 441
Funktionen, 411
AS-Interface-Module, Bestellnummern für Handbücher, 442
ASCII-Umwandlungsoperationen, 97
ASCII-Zeichenkette in Hexadezimalzahl wandeln, 97
Assistenten
AS-i, 411
Erweiterter Modemassistent, 302
Ethernet, 407
Internet, 410
Modem, 228
Operation, HSC, 113
Operation, NETR, NETW, 76
PID, 146
Positioniersteuerung, 254
TD 200, 4
Asynchrone Aktualisierungen, Operation PWM, 128
Aufbauen, Netz, 220
Auflösung, Zeit, 197, 198
Aufrufen, STEP 7-Micro/WIN, 7
Aus Netz lesen, 76, 77
Fehlercodes, 77
Ausbau
CPU, 17
Erweiterungsmodul, 17
Klemmenblock, 17
Ausführen
Operationen, 23
Programm, 12
Ausführung, Steuerungslogik, 22
Ausführungsfehler, 57
Ausführungszeiten, AWL-Operationen, 445
Ausgänge, 22
CPU, 354
Digitalerweiterungsmodul, 360
Konfigurieren der Zustände, 40
Operationen ohne Ausgänge, 55
Positioniermodul EM 253, 252
Ausgänge und Relais, 20
Ausgangskonfigurationen, Bearbeiten der Voreinstellungen des Positioniermoduls, 255
Auswählen
Befehlssätze, 53
Programm-Editor, 51
Automatikbetrieb, PID-Regler, 152
AWL-Editor
Beschreibung, 51
Funktionen, 51
AWL-Operationen
Ausführungszeiten, 445
Kurzinformation, 455

B

Batteriemodul, 34, 413
Bestellnummer, 442
Baudraten
Einstellung, 209
Netz, 220
Optimal, 233
Schalterstellungen
Multi-Master-Kabel, 7
PPI-Multi-Master-Kabel, 211, 224, 226, 231, 232
BCD in ganze Zahl wandeln, 94
Bearbeiten, Kommunikationsaufträge, 23
Bearbeitung, Im Betriebszustand RUN, 244
Bearbeitung bedingt beenden, 169
Bearbeitung beenden, 169
Bedienerschnittstellen, Bestellnummer, 443
Bedingte Eingänge, 55
Befehlsbyte, Positioniermodul EM 253, 291
Befehlssätze
Auswählen, 53
IEC 1131-3, 53
SIMATIC, 53
Behebung, Fehler, 56
Behördliche Zulassungen, 348
Beispiele
Berechnen des Leistungsbedarfs, 419
Betriebsarten von schnellen Zählern, 115
Bytes im Wort tauschen, 184
Festpunktarithmetik, 142
Frei programmierbare Kommunikation, 225
Genormte Umwandlungsoperationen, 95
Gleitpunktarithmetik, 142
IEC Zähler, 110
IEC Zeiten, 201
Impulsdauermodulation (PWM), 133
Interruptoperationen, 162
Interruptprogramm für zeitgesteuerten Interrupt, 162
Interruptprogramme, 49
Invertieroperationen, 163
Kontakte, 68
Modbus-Slave-Protokoll, Programmierung, 346
Modemmodul EM 241, 311
ODER-Verknüpfung, 165

Beispiele (Fortsetzung)
Operation ASCII-Zeichenkette in Hexadezimalzahl wandeln, 100
Operation Bearbeitung bedingt beenden, 170
Operation Bit in Hexadezimalzahl wandeln, 106
Operation Bitmuster für Sieben-Segment-Anzeige erzeugen, 96
Operation Dekrementieren, 145
Operation DIV, 143
Operation Einzel-Segment-Impulsfolge, 137
Operation Ersten Wert aus Tabelle löschen, 191
Operation Ganze Zahl in ASCII-Zeichenkette wandeln, 100
Operation Hexadezimalzahl in Bit wandeln, 106
Operation In STOP gehen, 170
Operation Inkrementieren, 145
Operation Letzten Wert aus Tabelle löschen, 191
Operation Meldung aus Zwischenspeicher übertragen, 88
Operation Meldung in Zwischenspeicher empfangen, 88
Operation MUL, 143
Operation Multi-Segment-Impulsfolge, 139
Operation Programmschleife mit FOR/NEXT, 172
Operation Realzahl in ASCII-Zeichenkette wandeln, 100
Operation Rücksetzen, 69
Operation Rückwärtszählen, 109
Operation Schnellen Zähler aktivieren, 125
Operation Setzen, 69
Operation Speicher mit Bitmuster belegen, 192
Operation Überwachungszeit rücksetzen, 170
Operation Unterprogramm aufrufen, 204
Operation Unterprogramm beenden, 206
Operation Vorwärts-/Rückwärtszählen, 109
Operation Wert in Schieberegister schieben, 183
Operation Wert in Tabelle eintragen, 190
Operation Wert in Tabelle suchen, 195
Operation Wertebereich übertragen, 168
Operation Zeit als Ausschaltverzögerung starten, 199
Operation Zeit als Einschaltverzögerung starten, 198
Operation Zeit als speichernde Einschaltverzögerung starten, 200
Operation Zu Sprungmarke springen, 173
Operationen Aus Netz lesen/In Netz schreiben, 79
Operationen für das Ablaufsteuerungsrelais, 174
Operationen für Unterprogramme, 206
PID-Programm, 153
PID-Regler, 154
Positioniermodul, 276–280
Programm für das USS-Protokoll, 330
Rotieroperationen, 181
Schiebeoperationen, 181
SIMATIC Zähler, 109
SIMATIC Zeiten, 198, 199, 200
Stackoperationen, 73
Teilung von Ablaufketten, 177
Token-Ring-Netz, 234
Übertragungsoperationen, 184
UND-Verknüpfung, 165
Unterprogramm, 49
Vergleichsoperationen, 90
Weiterschaltbedingungen, 179
XOR-Verknüpfung, 165
Zusammenführung von Ablaufketten, 178
Beispielprogramm, 9
Beobachten, 12
Programmstatus, 246
Prozessvariablen, 59
Variablen in einer Statustabelle, 247
Berechnen
Leistungsbedarf, 419–421
Token-Umlaufzeit, 233
Bereiche, PID-Regler, 151
Bereiche Byte und ganze Zahl, 24
Bestellnummern, 441–884
Analoge Erweiterungsmodule, 373
Analogerweiterungsmodul, 363, 373
AS-Interface-Modul CP 243-2, 411
CPUs, 351
Digitales Erweiterungsmodul, 357
Eingangssimulatoren, 418
PPI-Multi-Master-Kabel, 414
Betriebsart Profil, Positioniermodul EM 253, 260
Betriebsarten
PID-Regler, 152
Schnelle Zähler, 114
Betriebsartenschalter, 37
Betriebszustand RUN, 12, 37
Bearbeiten des Programms, 244
Betriebszustand STOP, 12, 37
Betriebszustände, CPU
Ändern, 37
Frei programmierbare Kommunikation, 225
Statusbits, 428
Bewegungsprofil, Definieren für das Positioniermodul EM 253, 260
Bibliothek für das Modbus-Protokoll, 339
Bibliothek für das USS-Protokoll, Steuern von MicroMaster-Antrieben, 319
Bibliotheken, Operationen, 60
Bistabiler Funktionsbaustein: Vorrangig Rücksetzen, 74
Bistabiler Funktionsbaustein: Vorrangig Setzen, 74
Bit in Hexadezimalzahl wandeln, 106
Bitverknüpfungsoperationen
Bistabiler Funktionsbaustein: Vorrangig Rücksetzen, 74
Bistabiler Funktionsbaustein: Vorrangig Setzen/Bistabiler Funktionsbaustein: Vorrangig Rücksetzen, 74
Kontakte, 66
Spulen, 70
Stackoperationen, 72
Bitwert direkt rücksetzen, 69, 70
Bitwert direkt setzen, 69
Bitwert direkt zuweisen, 70
Bitwert durch EXKLUSIV ODER verknüpfen, 164
Bitwert durch ODER verknüpfen, 164
Boolesche Operationen
Bistabiler Funktionsbaustein: Vorrangig Setzen/Bistabiler Funktionsbaustein: Vorrangig Rücksetzen, 74
Kontakte, 66
Spulen, 70
Stack, 72
Bureau Veritas (BV), 348
Busverstärker
Bestellnummern, 442
Netz, 220
Byte direkt lesen und übertragen, 167
Byte direkt schreiben und übertragen, 167
Byte in ganze Zahl wandeln, 94

Byte links rotieren, 180
Byte links schieben, 180
Byte rechts rotieren, 180
Byte rechts schieben, 180
Byte übertragen, 166
Bytekonsistenz, PROFIBUS, 391
Bytes im Wort tauschen, 184
Bytevergleich, 90

C

Cosinus, 144
CP 243-2 AS-I Interface, Technische Daten, 411
CP 243-2 Kommunikationsprozessor
  Anschlüsse, 412
  Assistent, 411
  Betrieb, 412
  Funktionen, 412
CP-Karten
  Bestellnummern, 442
  Unterstützt von STEP 7 Micro/WIN, 224
  Wählen, 223
CP5411, 442
CP5511, 442
CP5611, 442
CPU
  Abmessungen, 2
  Anzahl Verbindungen, 212
  Ausführungsgeschwindigkeit, 2
  Backup, 2
  Digitale E/A, 2
  Echtzeituhr, 2
  Eingangssimulatoren, 418
  Erweiterungsmodule, 2
  Fehlerbehebung, 56
  Kennregister (SMB6), 430
  Kommunikationsschnittstellen, 2
  Leistungsbedarf, 15, 419
  Passwortschutz, 44
  Schraubengrößen für den Einbau, 16
  Speicher, 2
  Suchen im Netz, 211
  Zyklus, 22
CPU 224, DP-Programmierbeispiel, 396
CPUs
  Ausbau, 17
  Bestellnummern, 351, 441
  Einbau, 16
  Funktionen, 64, 483
  Leistungsdaten, 352
  Schaltpläne, 354–356
  Selbstdiagnose, 23
  Speicherbereiche, 64, 453, 483
  Technische Daten, 351
  Technische Daten der Ausgänge, 353
  Technische Daten der Eingänge, 352
  Unterstützung intelligenter Module, 311
CRC-Tabelle, Modbus-Slave-Protokoll, 340

D

Daten
  Empfangen, 82, 87
  Senden, 82
  Speichern und Zurückholen , 34
Datenbaustein, 50
Datenbaustein-Editor
  Zuordnen von Adressen, 58
  Zuordnen von Anfangswerten, 58
Datenhaltung, 34–37
  Anlauf, 35–37
  Batteriemodul (optional), 34
  Bereiche, 41
  EEPROM, 34–36
Datenkonsistenz, PROFIBUS, 391
Datentypen, Parameter für Unterprogramme, 204
Datenübertragungen, Modemmodul EM 241, 299
Datenwortformat der Ausgänge
  EM 232, 369
  EM 235, 369
Datenwortformat der Eingänge
  EM 231, 368
  EM 235, 368
Datum, Einstellung, 75
DC-Ausgänge und Relais, 20
DC-Transistor, Schutz, 20
Deaktivieren, Schnelle Zähler, 124
DECEL_TIME (Verzögerungszeit), Positioniermodul EM 253, 257
Dekrementieren, 145
Det Norske Veritas (DNV), 348
Diagnose, CPU-Selbsttest, 23
Diagnoseinformationen, Positioniermodul EM 253, 282
Dialogfeld 'Informationen', 56
Differentialanteil, PID-Algorithmus, 149
Digitale Ausgänge, Konfigurieren der Zustände, 40
Digitale Eingänge
  Filter, 41
  Impulsabgriff, 41
  Lesen, 39
Digitale Module, 3
Digitales Erweiterungsmodul, Adressierung, 31
Diode, Schutzbeschaltung, 20
DIP-Schalter
  RS-232/PPI-Multi-Master-Kabel, 416
  RTD, 380–381
  Thermoelement, 376
DIP-Schalterstellungen, PPI-Multi-Master-Kabel, 7
DIP-Schalterstellungen, PPI-Multi-Master-Kabel, 226
Direkter Kontakt, 66
Dividieren, 141
Doppelwort links rotieren, 180
Doppelwort links schieben, 180
Doppelwort rechts rotieren, 180
Doppelwort rechts schieben, 180
Doppelwort übertragen, 166
Doppelwortvergleich, 90
Drehmoment/Drehzahlen, Typisch für Motoren, 256

E

E/A, Lesen und Schreiben, 39
E/A-Adressierung, 31
E/A-Fehler, 57
E/A-Interrupts, 159
E/A-Status (SMB5), 430
Echtzeituhr (TOD), 75
Echtzeituhr lesen, 75
Echtzeituhr schreiben, 75

Echtzeituhroperationen, 75
Editoren
  Anweisungsliste (AWL), 51
  Funktionsplan (FUP), 52
  Kontaktplan (KOP), 52
EEPROM, 34
  Fehlercodes, 424
  Speichern des Variablenspeichers (V), 38, 433
  Speichern von Merkern (M), 35
Einbau
  CPU, 16
  Elektrische Störungen, 14
  EM 231, 370
  EM 235, 370
  Erweiterungsmodul, 16
  Hochspannungsgeräte, 14
  Mindestabstand, 14
  Montagevoraussetzungen, 16
  Richtlinien, 14
  S7-200, 15
  Spannungsversorgung, 15
  Speichermodul, 36
  Steckleitung, 413
  Wärmeproduzierende Geräte, 14
Einerkomplement von Byte erzeugen, 163
Einerkomplement von ganzer Zahl (16 Bit) erzeugen, 163
Einerkomplement von ganzer Zahl (32 Bit) erzeugen, 163
Eingänge, 22, 23
  Analogfilter, 42
  Bedingte/absolute, 55
  CPU, 354
  Digitaler Filter, 41
  Digitalerweiterungsmodul, 358
  Kalibrierung, 365
  Positioniermodul EM 253, 252
  Schnelle Zähler, 114
  Starten und Rücksetzen (HSC), 117
Eingangskonfigurationen, Bearbeiten der Voreinstellungen des Positioniermoduls EM 253, 255
Eingangssimulatoren, 418
Eingeben, Operation, 10
Eingebettete Variablen, In Textnachrichten und SMS-Kurzmitteilungen, 299
Einrichten
  AWL-Statusanzeige, 247
  MM3-Antrieb, 333
  MM4-Antrieb, 337
  Passwort, 44
  Statusanzeige in FUP und KOP, 246
  Steuerbyte (HSC), 119
Einstellen
  Aktuellen und voreingestellten Wert für HSC, 119
  Betriebszustand der S7-200, 37
  DIP-Schalter des Thermoelements, 376
  Entfernte Adresse für die S7-200, 211
  Kommunikationsprotokoll, 212
  RTD DIP-Schalter, 380–381
Einstellung
  Baudrate, 209
  Datum, 75
  Uhrzeit, 75
Einzel-Segment-Betrieb
  Ändern der PTO-Zykluszeit, 135
  Ändern der PTO-Zykluszeit und des Impulszählwerts, 136
  Ändern des PTO-Impulszählwerts, 136
  Initialisieren von PTO, 135
Elektrische Lebensdauer, 349
Elektromagnetisch
  Störabstrahlung, 350
  Störfestigkeit, 350
  Verträglichkeit, S7-200, 350
EM 231 Analogeingabemodul
  Datenwortformat der Eingänge, 368
  Einbau, 370
  Genauigkeit und Wiederholbarkeit, 371
  Kalibrierung, 365
  Schaltbild der Eingänge, 368
  Technische Daten, 372
EM 231 RTD-Modul
  Anschlüsse, 374
  CPU-Kompatibilität, 374
  Einstellen der DIP-Schalter, 380–381
  Konfigurieren, 380
  Statusanzeigen, 382
  Technische Daten, 373
  Temperaturbereiche und Genauigkeit, 383–384
EM 231 Thermoelement-Modul
  Anschlüsse, 374
  CPU-Kompatibilität, 374
  Einstellen der DIP-Schalter, 376
  Grundlagen, 375
  Konfigurieren, 375
  Statusanzeigen, 377
  Technische Daten, 373
  Temperaturbereiche und Genauigkeit, 378–379
EM 232 Analogausgabemodul
  Datenwortformat der Ausgänge, 369
  Schaltbild der Ausgänge, 370
EM 235 Analogein-/Analogausgabemodul
  Datenwortformat der Ausgänge, 369
  Datenwortformat der Eingänge, 368
  Einbau, 370
  Genauigkeit und Wiederholbarkeit, 371
  Kalibrierung, 365
  Konfiguration, 367
  Schaltbild der Ausgänge, 370
  Schaltbild der Eingänge, 369
  Technische Daten, 372
EM 253 Steuer-Panel, 281–283
EM 277 PROFIBUS-DP-Modul
  Adressschalter, 386
  Als DP-Slave, 387
  Anzahl Verbindungen, 212
  CPU-Kompatibilität, 386
  DP-Protokoll, 387
  Im PROFIBUS-Netz, 388
  Konfigurationsdatei, 394–395
  Konfigurationsmöglichkeiten, 390
  Konfigurieren, 389–390
  LED-Statusanzeigen, 393
  Modus für Datenaustausch, 391
  Sondermerkerbytes, 392
  Status-LEDs, 386
  Technische Daten, 385
  Weitere Leistungsmerkmale, 393
Empfangen von Daten, 87
EN, 55, 63
Ende Ablaufsteuerungsrelais, 174
Ende Programmschleife mit NEXT, 171
Endezeichenerkennung, 86
ENO, 55, 63

Entfernte Adresse, Einstellen für die S7-200, 211
Entwerfen
  Sicherheitsstromkreise, 48
  System für eine Micro-SPS, 48
Erdung, 18, 19
Erste und zweite Stackebene durch ODER verknüpfen, 72
Erste und zweite Stackebene durch UND verknüpfen, 72
Erstellen
  Anwenderdefinierte Protokolle, 225
  Konfigurationspläne, 49
Ersten Wert aus Tabelle löschen, 191
Erstes Zeichen in Zeichenkette suchen, 188
Erweiterter Modemassistent, Modemmodul EM 241, 302
Erweiterungs-E/A, 31
  Busfehler (SMW98), 437
Erweiterungsmodule, 3
  Abmessungen, 16
  Adressierung der Ein- und Ausgänge, 31
  Analog
    Bestellnummern, 363, 373
    Schaltpläne, 364
    Technische Daten, 363
    Technische Daten der Ausgänge, 364
    Technische Daten der Eingänge, 363
  Ausbau, 17
  Bestellnummern, 441
  Digital
    Allgemeine technische Daten, 357
    Bestellnummern, 357
    Schaltpläne, 360–362
    Technische Daten der Ausgänge, 359
    Technische Daten der Eingänge, 358
  Einbau, 16, 413
  Kenn- und Fehlerregister, 431
  Leistungsbedarf, 15, 419
Ethernet, 212
  TCP/IP, 213
Ethernet-Modul, Assistent, 407
Ethernet-Modul CP 243-1
  Anschlüsse, 407
  Anzahl der Verbindungen, 213
  Assistent, 407
  Bestellnummer, 406, 441
  Funktionen, 407
  Kommunikationsprozessoren, 406
  Konfiguration, 407
  Technische Daten, 406
Ethernet-Netz, CP 243-1, 217
EXKLUSIV ODER-Verknüpfung, Beispiel, 165

F

Fallende Flanke, 66
Fehler
  Ansicht, 56
  Aus Netz lesen und In Netz schreiben, 77
  Ausführung des Modbus-Slave-Protokolls, 345
  Behebung, 56
  E/A, 57
  Konfiguration des Modemmoduls EM 241, 301
  Laufzeit, 57
  Leichte, 56
  Operationen für das Modemmodul EM 241, 309
  PID-Regler, 152
  Programmausführung, 57
  Programmübersetzung, 56
  Schwere, 57
  SMB1, Ausführungsfehler, 428
Fehlerbehebung
  Fehlercodes, 424
  Leichte Fehler, 56
  Richtlinien, 249
  S7-200 Hardware, 249
  Schwere Fehler, 57
Fehlercodes, 424
  Modulfehler des Positioniermoduls EM 253, 284
  Operationen für das Positioniermodul EM 253, 283
  Operationen für das USS-Protokoll, 331
  Positioniermodul EM 253, 283
  Programmierfehler zur Laufzeit, 425
  Schwere Fehler, 424
  Verletzung der Übersetzungsregeln, 426
Fehlercodes bei der Ausführung, Operationen für das USS-Protokoll, 331
Fehlercodes der Operationen, Positioniermodul EM 253, 283
Fehlercodes des Moduls, Positioniermodul EM 253, 284
Festpunktarithmetik, Beispiel, 142
Filter
  Analogeingänge, 42
  Digitale Eingänge, 41
Flankenoperationen, 66
Forcen von Werten, 247, 248
Format Byteadresse, 24
Format von CPU-Datenübertragungsnachrichten, Modemmodul EM 241, 318
Format von Telefonnummern für die Nachrichtenübermittlung, Modemmodul EM 241, 316
Format von Textnachrichten, Modemmodul EM 241, 317
Frei programmierbare Kommunikation, 429
  Aktivieren, 81
  Anwenderdefinierte Protokolle, 225
  Beispiel, 225
  Definition, 159
  Empfangene Zeichen (SMB2), 429
  Interruptgesteuert, 87
  Konfigurieren, PPI-Multi-Master-Kabel, 230
  Operationen Meldung aus Zwischenspeicher übertragen und Meldung in Zwischenspeicher empfangen, 81
  Paritätsfehler (SMB3), 429
  RS-232-Standard, 226
  SMB30 und SMB130, 432
  Wechseln von PPI, 82
Freigabeausgang (ENO), 55
Freigabeeingang (EN), 55
Funkmodem
  Mit PPI-Multi-Master-Kabel, 232
  PPI-Modus, 232
  PPI/frei programmierbare Kommunikation, 232
Funkruf, Modemmodul, 298
Funktionen
  CPUs, 64, 483
  Modbus, 342
  Modemmodul EM 241, 296
  Positioniermodul EM 253, 252
Funktionsleisten, 51
Funktionsplan. *Siehe* FUP-Editor

FUP-Editor
Beschreibung, 52
Funktionen, 52
Konventionen, 54

G

Ganze Zahl (16 Bit) in ganze Zahl (32 Bit) wandeln, 94
Ganze Zahl (32 Bit) in ASCII-Zeichenkette wandeln, 98
Ganze Zahl (32 Bit) in ganze Zahl (16 Bit) wandeln, 94
Ganze Zahl (32 Bit) in Realzahl wandeln, 94
Ganze Zahl (32 Bit) in Zeichenkette wandeln, 101, 104
Ganze Zahl in ASCII-Zeichenkette wandeln, 97
Ganze Zahl in BCD wandeln, 94
Ganze Zahl in Byte wandeln, 94
Ganze Zahl in Zeichenkette wandeln, 101, 104
Ganze Zahlen (16 Bit) dividieren mit Divisionsrest (DIV), 143
Ganze Zahlen (16 Bit) in ganze Zahl (32 Bit) multiplizieren (MUL), 143
Ganzzahlenvergleich, 90
GAP-Aktualisierungsfaktor, 233
Genormte Umwandlungsoperationen, 93
Gerätestammdaten (GSD), EM 277 PROFIBUS-DP, 394–395
Germanisher Lloyd (GL), 348
Gleitpunktarithmetik, Beispiel, 142
Gleitpunktwerte, 29, 150
Globale Variablentabelle, 58
GSD-Datei, EM 277 PROFIBUS-DP, 394–395

H

Handbetrieb, PID-Regler, 152
Handbücher, Bestellnummern, 442
Hardware, Fehlerbehebung, 249
Hexadezimalzahl in ASCII-Zeichenkette wandeln, 97
Hexadezimalzahl in Bit wandeln, 106
HMI-Geräte, 224
Hochspannungs-Isolationsprüfung, 350
Höchste Teilnehmeradresse (HSA), 233
HSC, 46
Operations-Assistent, 113
Hutschiene
Abmessungen, 16
Montage, 16
HyperTerminal, Konfigurieren des RS-232/PPI-Multi-Master-Kabels, 239–483

I

IEC 1131-3 Befehlssatz, 53
IEC Zähler
Beispiel, 110
Operation Vorwärts-/Rückwärtszähler, 110
Rückwärtszähler, 110
Vorwärtszähler, 110
IEC Zeiten, 201
Beispiel, 201
Impuls (TP), 201
Impulsabgriff, 41, 42
Impulsausgabe
Impulsausgabe (PLS), 126
Impulsdauermodulation (PWM), 126
Impulsfolge (PTO), 126
Schnelle Impulsausgänge, 46
SMB66-SMB85, 436
Impulsausgabe (PLS), 126
Impulsausgänge und Richtungsausgänge, Positioniermodul EM 253, 255
Impulsdauermodulation (PWM), 46, 126
Aktualisierungsmethoden, 128
Ändern der Impulsdauer, 133
Beispiel, 133
Beschreibung, 128
Initialisieren, 133
Konfigurieren der Sondermerker, 129
Positionier-Assistent, 126
Schrittmotorsteuerung, 131
Werte in der Profiltabelle, 131
Zykluszeit, 128
Impulsfolge (PTO), 46, 126
Ändern der Zykluszeit, 135
Ändern der Zykluszeit und des Impulszählwerts, 136
Ändern des Impulszählwerts, 136
Beschreibung, 127
Einzel-Segment-Pipeline, 127
Initialisieren des Multi-Segment-Betriebs, 136
Initialisieren für Einzel-Segment-Betrieb, 135
Konfigurieren der Sondermerker, 129
Multi-Segment-Pipeline, 127
Positionier-Assistent, 126
Schrittmotorsteuerung, 131
Werte in der Profiltabelle, 131
Zykluszeit, 127
In Netz schreiben, 76, 77
Fehlercodes, 77
In STOP gehen, 169
Indirekte Adressierung, 32
& und *, 32
Ändern eines Pointers, 32
Initialisieren
Modbus-Protokoll, 340
PTO für Einzel-Segment-Betrieb, 135
PTO für Multi-Segment-Betrieb, 136
PWM-Ausgabe, 133
Initialisierung, Schnelle Zähler, 121
Inkrementieren, 145
Inkrementieren eines Pointers, 32
Installation, STEP 7-Micro/WIN, 4
Integralanteil, PID-Algorithmus, 148
Integrierte Ein- und Ausgänge, 31
Intelligente Module, 3
Status (SMB200 bis SMB549), 439
Von CPUs unterstützt, 311
Internationale Normen, 348
Internet-Geräte, CP 243-1 IT, 217
Internet-Modul, Assistent, 410
Internet-Modul CP 243-1 IT
Anschlüsse, 410
Anzahl der Verbindungen, 213
Bestellnummer, 409, 441
Funktionen, 409
Internet-Assistent, 410
Kommunikationsprozessoren, 409
Konfiguration, 410
Technische Daten, 409
Interrupt trennen, 156
Interrupt zuordnen, 156

Interruptereignisse
Arten, 157
Kurzinformation, 452
Priorität, 161
Interruptgesteuert, 87
Interruptoperationen
Alle Interruptereignisse freigeben, 156
Alle Interruptereignisse sperren, 156
Beispiel, 162
Interrupt trennen, 156
Interrupt zuordnen, 156
Interruptprogramm bedingt beenden, 156
Interruptprogramm absolut beenden, 156
Interruptprogramm bedingt beenden, 156, 203
Interruptprogramm für zeitgesteuerten Interrupt, Beispiel, 162
Interruptprogramme, 23, 39
Aufrufen von Unterprogrammen, 159
Beispiel, 49
Beschreibung, 158
E/A, 159
Gemeinsames Nutzen von Daten mit dem Hauptprogramm, 158
Kommunikationsschnittstelle, 159
Priorität, 160
Richtlinien, 50, 158
Steigende/fallende Flanke, 159
Systemunterstützung, 158
Von der S7-200 unterstützte Arten, 159
Warteschlangen, 160
Zeitgesteuert, 159
Interrupts
Schnelle Zähler, 114
Zuweisen zu HSC, 120
Invertieroperationen, Beispiel, 163

J

JOG_INCREMENT, Positioniermodul EM 253, 257
JOG_SPEED, Positioniermodul EM 253, 257

K

Kabel
Abschließen, 222
Bestellnummern, 442
Einstellen, 208
Konfigurieren
Entferntes Modem, 229
Frei programmierbare Kommunikation, 230
Netzwerk, 220, 221
RS-232/PPI-Multi-Master, 414
Steckleitung für Erweiterungsmodule, 413
USB/PPI-Multi-Master, 414
Kalibrierung
Eingänge, 365
EM 231, 365
EM 235, 365
Kenn- und Fehlerregister des E/A-Moduls, 431
Klemmenblock
Ausbau, 17
Wiedereinbau, 17
Kommunikation, 208
Baudrate, 231, 232
Funkmodem, 232–233
Konflikte, 236
Mit MicroMaster-Antrieben, 320
Modbus-Slave-Protokoll, 340
Modem, 231–232
S7-200, 8
Schnittstelle, Einstellen, 208
Unterstützte Protokolle, 212
Kommunikation mit Antrieben, Berechnen des Zeitbedarfs, 321
Kommunikationsaufträge, Bearbeiten, 23
Kommunikationsbearbeitung, 40
Kommunikationseinstellungen, STEP 7 Micro/WIN, 8
Kommunikationskarten, Bestellnummern, 442
Kommunikationsoperationen
Aus Netz lesen, 76
In Netz schreiben, 76
Meldung aus Zwischenspeicher übertragen, 80
Meldung in Zwischenspeicher empfangen, 80
Schnittstellenadresse einstellen, 89
Schnittstellenadresse holen, 89
Kommunikationsprotokoll
Anwenderdefiniert, 225
Einstellen, 212
Mehrpunktfähige Schnittstelle (MPI), 213, 235
PROFIBUS, 213, 235
Punkt-zu-Punkt-Schnittstelle (PPI), 212, 235
TCP/IP, 213
Kommunikationsprozessoren
Siehe auch CP-Karten
CP 243-2 AS-Interface, 411
Ethernet-Modul CP 243-1, 406
Internet-Modul CP 243-1 IT, 409
Kommunikationsschnittstelle
Anschlussbelegung, 221
Frei programmierbare Kommunikation, 225
Interrupts, 159
Kompatibilität
EM 231 RTD, 374
EM 231 Thermoelement, 374
EM 277 PROFIBUS-DP, 386
Modemmodul EM 241, 399
Positioniermodul EM 253, 402
Komplexe PPI-Netze, 215
Konfiguration
Ausgangszustände, 40
EM 231, 366
EM 231 RTD, 380
EM 231 Thermoelement, 375
EM 235, 367
EM 277 PROFIBUS-DP, 389–390
Ethernet-Modul CP 243-1, 407
Internet-Modul CP 243-1 IT, 410
Positioniermodul EM 253, 282
Remanente Bereiche im Speicher, 41
Symboltabelle für Modbus, 341
Konfigurations-/Profiltabelle, Positioniermodul EM 253, 285
Konfigurationstabelle für das Modemmodul, 301
Konfigurationspläne, 49
Konfigurationstabelle, Modemmodul EM 241, 301, 314
Konfigurieren
Funktionsweise von PTO/PWM, 129
HyperTerminal, 239–483
Netz, 220
Konflikte, Vermeiden im Netz, 236

Konstante Werte, 30
Kontakte, 66
  Beispiel, 68
Kontaktplan. *Siehe* KOP-Editor
Konventionen
  Programm-Editoren, 54
  S7-200 Programmierung, 55
KOP-Editor
  Beschreibung, 52
  Funktionen, 52
  Konventionen, 54
Kopieren, Programm in Speichermodul, 36
Kurzinformation, 451
Kurzmitteilungsdienst, Modemmodul EM 241, 298

L

Laden
  Neuen aktuellen Wert in HSC, 124
  Neuen voreingestellten Wert in HSC, 124
Laden aus CPU, Programm, 34
Laden in CPU, Programm, 12, 34
Lage des Arbeitsbereichs, Positioniermodul EM 253, 264
Landeskennzahlen, Vom EM 241 unterstützt, 296
Laufzeitfehler, 57
LEDs, Modemmodul EM 241, 302
Leerlauferkennung, 84
Leichte Fehler, 56
Leistungsbedarf, 15
  Beispiel, 420
  Berechnen, 419, 421
  CPU, 419
  Erweiterungsmodul, 419
  Tabelle zum Berechnen, 421
Leistungsdaten, CPUs, 352
Leistungsfähigkeit, Optimieren im Netz, 233
Lesen von Daten aus den Eingängen, 22, 23
Lesezeichen, 244
Letzten Wert aus Tabelle löschen, 191
Lloyds Register of Shipping (LRS), 348
Logik, Steuerung, 22
Logische Verbindungen
  MPI, 213
  PPI, 212
Lokaldatenspeicher (L), 28
Lokale Variablentabelle, 51, 59

M

Master-Geräte, 209
MAX_SPEED, Positioniermodul EM 253, 256
Meldung aus Zwischenspeicher übertragen, 80
  Frei programmierbare Kommunikation, 81
  Senden von Daten, 82
Meldung in Zwischenspeicher empfangen, 80
  Anwenderabbruch, 87
  Empfangen von Daten, 82
  Endebedingungen, 84
  Endezeichenerkennung, 86
  Frei programmierbare Kommunikation, 81
  Leerlauferkennung, 84
  Meldungs-Timer, 86
  Paritätsfehler, 87
  Pauseerkennung, 85
  Startbedingungen, 84
  Startzeichenerkennung, 84
  Timer für die Zeit zwischen den Zeichen, 86
  Zählwert für die Höchstzeichenzahl, 87
Meldungen, Token-Ring-Netz, 234
Meldungs-Timer, 86
Messart, Positioniermodul EM 253, 254
Micro-SPS, Entwerfen, 48
Micro/WIN. *Siehe* STEP 7-Micro/WIN
MicroMaster-Antrieb
  Anschließen, 332
  Kommunikation, 320
  Lesen und Schreiben, 327, 328
  Steuern, 319
Mindestabstand, 14
MM3-Antrieb
  Anschließen, 332
  Einrichten, 333
MM4-Antrieb
  Anschließen, 335
  Einrichten, 337
Modbus-RTU-Protokoll, 342
  Abbilden von Adressen, 298
  Modemmodul EM 241, 297
  Vom Modemmodul unterstützte Funktionen, 297
Modbus-Slave-Protokoll
  Abbilden von Adressen in der S7-200, 341
  Adressen, 341
  Ausführungszeit, 340
  Beispiel für die Programmierung, 346
  CRC-Tabelle, 340
  Fehlercodes für die Ausführung, 345
  Initialisierung, 340
  Konfigurieren der Symboltabelle, 341
  MBUS_INIT, 343
  MBUS_SLAVE, 345
  Operationen, 342
  Sondermerker, 340
  Unterstützte Funktionen, 342
  Verwendete Ressourcen, 340
Modem
  Entfernt, PPI-Multi-Master-Kabel, 229
  Mit PPI-Multi-Master-Kabel, 231
Modem-Assistent, 228
Modemmodul, 398
  Beispiel, 311
  Datenübertragungen, 299
  Erweiterter Modemassistent, 302
  Fehler der Operationen, 309
  Format von CPU-Datenübertragungsnachrichten, 318
  Format von Textnachrichten, 317
  Funkruf, 298
  Internationale Schnittstelle zu Telefonleitungen, 296
  Kurzmitteilungsdienst, 298
  Leistungsmerkmale, 296
  Nummernfunkruf, 298
  Operation MODx_CTRL, 307
  Operation MODx_MSG, 308
  Operation MODx_XFR, 307
  Operationen, 306
  Passwortschutz, 299
  RJ11-Stecker, 296
  Sicherheitsrückruf, 300
  SMS-Nachrichtenübermittlung, 298
  Status-LEDs, 302
  Telefonnummer für die Nachrichtenübermittlung, 316

Modemmodul, 398 (Fortsetzung)
Textfunkruf, 298
Modemmodul EM 241
Beispiel, 311
CPU-Kompatibilität, 399
Datenübertragungen, 299
Einbau, 399
Erweiterter Modemassistent, 302
Fehler der Operationen, 309
Format von CPU-Datenübertragungsnachrichten, 318
Funkruf, 298
Internationale Schnittstelle zu Telefonleitungen, 296
Konfigurationstabelle, 301, 314
Kurzmitteilungsdienst (SMS), 298
Leistungsmerkmale, 296
Modbus-Adressen, 298
Modbus-RTU-Protokoll, 297
Nummernfunkruf, 298
Operation MOD_XFR, 307
Operation MODx_CTRL, 307
Operation MODx_MSG, 308
Operationen, 306
Passwortschutz, 299
RJ11-Stecker, 296
Schaltbild, 398
Schnittstelle zu STEP 7-Micro/WIN, 297
Sicherheitsrückruf, 300
Sondermerker, 312
Status-LEDs, 302
Technische Daten, 398
Telefonnummer für die Nachrichtenübermittlung, 316
Textfunkruf, 298
Unterstützte Länder, 296
Modus für Datenaustausch, EM 277 PROFIBUS-DP, 391
Modus für schnellen Zähler definieren (HDEF), 112
Montage
Abmessungen, 16
Mindestabstand, 16
Schalttafel, 16
Standard-Hutschiene, 16
Motoren, Kurve Drehmoment/Drehzahl, typisch, 256
MPI-Netz
Mehr als 187,5 kBaud, 215
Weniger als 187,5 kBaud, 215
MPI-Protokoll, 213, 235
Multi-Master-Kabel, 7
Multi-Master-PPI-Netz, 214
Multi-Segment-Betrieb, Initialisieren für PTO, 136
Multiplizieren, 141

N

Nationale Normen, 348
Natürlicher Exponent, 144
Natürlicher Logarithmus, 144
Navigationsleiste, 51
Negative Polarität, Positioniermodul EM 253, 255
NETR, NETW, Operations-Assistent, 76
Netz
Abschließen des Kabels, 222
Adressen, 209
Aufbauen, 220
Baudrate, 220
Beispielkonfigurationen, 214, 215, 216
Berechnen der Abstände, 220
Busverstärker, 220
CP-Karte, 223
Einrichten der Kommunikation, 208–431
Funkmodem, 232
GAP-Aktualisierungsfaktor, 233
Geräteadressen, 212
HMI-Geräte, 224
Höchste Teilnehmeradresse (HSA), 233
Kabel abschließen, 222
Kommunikationsschnittstelle, 221
Komplex, 236
Komplexe PPI-Netze, 215
Komponenten, 220–223
Master-Geräte, 209
Modem, 231
MPI, mehr als 187,5 kBaud, 215
MPI, weniger als 187,5 kBaud, 215
Multi-Master-PPI, 214
Netzwerkkabel, 220, 221
Optimieren der Leistungsfähigkeit, 233
Potentialtrennung, 220
PPI-Multi-Master-Kabel, 223
PROFIBUS, 209
PROFIBUS-DP, 216
Richtlinien für die Konfiguration, 220
Single-Master-PPI, 214
Slave-Geräte, 209
Suchen nach der CPU, 211
Token-Umlaufzeit, 233
Übertragungsgeschwindigkeit, 220
Nippon Kaiji Kyokai (NK), 348
Normalisieren, Prozessvariablen, 150
Normen, National und international, 348
NOT, 66
Null-Modem-Adapter, 231
Nulloperation, 70
Nullspannungsfester Programmspeicher, 38
Numerische Funktionen
Cosinus, 144
Natürlicher Exponent, 144
Natürlicher Logarithmus, 144
Quadratwurzel einer Realzahl ziehen, 144
Sinus, 144
Tangens, 144
Nummernfunkruf, Modemmodul EM 241, 298

O

Obersten Stackwert aus Stack schieben, 72
Obersten Stackwert duplizieren, 72
ODER-Verknüpfung, Beispiel, 165
Öffnen, Programm-Editor, 9
OP3, OP7, OP17, Bestellnummern, 443
Operandenbereiche, 65
Operation, Eingeben, 10
Operation Anzahl an Bytes übertragen, 168
Operation ASCII-Zeichenkette in Hexadezimalzahl wandeln, Beispiel, 100
Operation Aus Netz lesen, Beispiel, 79
Operation Bearbeitung bedingt beenden, Beispiel, 170
Operation Bit in Hexadezimalzahl wandeln, Beispiel, 106
Operation Bitmuster für Sieben-Segment-Anzeige erzeugen, 96
Beispiel, 96
Operation Bytes im Wort tauschen, Beispiel, 184

Operation Dekrementieren, Beispiel, 145
Operation Ende Programmschleife mit NEXT, Beispiel, 172
Operation Ersten Wert aus Tabelle löschen, Beispiel, 191
Operation Ganze Zahl in ASCII-Zeichenkette wandeln, Beispiel, 100
Operation Ganze Zahlen (16 Bit) dividieren mit Divisionsrest (DIV), Beispiel, 143
Operation Ganze Zahlen (16 Bit) in ganze Zahl (32 Bit) multiplizieren (MUL), Beispiel, 143
Operation Hexadezimalzahl in Bit wandeln, Beispiel, 106
Operation Impulsfolge (PTO), Beispiele, 137, 139
Operation In Netz schreiben, Beispiel, 79
Operation In STOP gehen, Beispiel, 170
Operation Inkrementieren, Beispiel, 145
Operation Letzten Wert aus Tabelle löschen, Beispiel, 191
Operation MBUS_INIT, 343
Operation MBUS_SLAVE, 345
Operation Meldung aus Zwischenspeicher übertragen, Beispiel, 88
Operation Meldung in Zwischenspeicher empfangen Beispiel, 88
SMB86 bis SMB94, SMB186 bis SMB194, 436
Operation MODx_CTRL, Modemmodul EM 241, 307
Operation MODx_MSG, Modemmodul EM 241, 308
Operation MODx_XFR, Modemmodul EM 241, 307
Operation Programmschleife mit FOR, Beispiel, 172
Operation Realzahl in ASCII-Zeichenkette wandeln, Beispiel, 100
Operation Rücksetzen, Beispiel, 69
Operation Rückwärtszählen, Beispiel, 109
Operation Schnellen Zähler aktivieren (HSC), Beispiel, 125
Operation Setzen, Beispiel, 69
Operation Speicher mit Bitmuster belegen, Beispiel, 192
Operation Überwachungszeit rücksetzen, Beispiel, 170
Operation Unterprogramm aufrufen, Beispiel, 204
Operation Unterprogramm beenden, Beispiel, 206
Operation USS4_DRV_CTRL, 324
Operation USS4_INIT, 323
Operation USS4_RPM_x, 327, 328
Operation USS4_WPM_x, 327, 328
Operation Vorwärts-/Rückwärtszählen, Beispiel, 109
Operation Wert in Schieberegister schieben, Beispiel, 183
Operation Wert in Tabelle eintragen, Beispiel, 190
Operation Wert in Tabelle suchen, Beispiel, 195
Operation Wertebereich übertragen, Beispiel, 168
Operation Zeit als Ausschaltverzögerung starten (TOF), Beispiel, 199
Operation Zeit als Einschaltverzögerung starten (TON), Beispiel, 198
Operation Zeit als speichernde Einschaltverzögerung starten (TONR), Beispiel, 200
Operation Zu Sprungmarke springen, Beispiel, 173
Operationen
Ablaufsteuerungsrelais bedingt beenden, 174
Ablaufsteuerungsrelais laden, 174
Addieren, 141
Alle Interruptereignisse freigeben, 156
Alle Interruptereignisse sperren, 156
Anzahl an Bytes übertragen, 168
Anzahl an Doppelwörtern übertragen, 168
Anzahl an Wörtern übertragen, 168
ASCII-Zeichenkette in Hexadezimalzahl wandeln, 97
Aus Netz lesen, 76
Ausführen, 23
BCD in ganze Zahl wandeln, 94
Bearbeitung bedingt beenden, 169
Bearbeitung beenden, 169
Bistabiler Funktionsbaustein: Vorrangig Rücksetzen, 74
Bistabiler Funktionsbaustein: Vorrangig Setzen, 74
Bit in Hexadezimalzahl wandeln, 106
Bitmuster für Sieben-Segment-Anzeige erzeugen, 96
Bitverknüpfung, 66
Bitwert direkt rücksetzen, 70
Bitwert direkt setzen, 70
Bitwert direkt zuweisen, 70
Byte direkt lesen und übertragen, 167
Byte direkt schreiben und übertragen, 167
Byte in ganze Zahl wandeln, 94
Byte links rotieren, 180
Byte links schieben, 180
Byte rechts rotieren, 180
Byte rechts schieben, 180
Byte übertragen, 166
Bytes im Wort tauschen, 184
Cosinus, 144
Dekrementieren, 145
Direkter Kontakt, 66
Dividieren, 141
Doppelwort links rotieren, 180
Doppelwort links schieben, 180
Doppelwort rechts rotieren, 180
Doppelwort rechts schieben, 180
Doppelwort übertragen, 166
Echtzeituhr, 75
Echtzeituhr lesen, 75
Echtzeituhr schreiben, 75
Einerkomplement von Byte erzeugen, 163
Einerkomplement von ganzer Zahl (16 Bit) erzeugen, 163
Einerkomplement von ganzer Zahl (32 Bit) erzeugen, 163
Ende Ablaufsteuerungsrelais, 174
Ende Programmschleife mit NEXT, 171
Erste und zweite Stackebene durch ODER verknüpfen, 72
Erste und zweite Stackebene durch UND verknüpfen, 72
Erstellen von Positioniersteuerung, 294
Ersten Wert aus Tabelle löschen, 191
Erstes Zeichen in Zeichenkette suchen, 188
EXKLUSIV ODER, 164
Fallende Flanke, 66
Ganze Zahl (16 Bit) in ganze Zahl (32 Bit) wandeln, 94
Ganze Zahl (32 Bit) in ASCII-Zeichenkette wandeln, 98
Ganze Zahl (32 Bit) in ganze Zahl (16 Bit) wandeln, 94
Ganze Zahl (32 Bit) in Realzahl wandeln, 94
Ganze Zahl (32 Bit) in Zeichenkette wandeln, 101, 104
Ganze Zahl in ASCII-Zeichenkette wandeln, 97
Ganze Zahl in BCD wandeln, 94
Ganze Zahl in Byte wandeln, 94
Ganze Zahl in Zeichenkette wandeln, 101, 104
Ganze Zahlen (16 Bit) dividieren mit Divisionsrest (DIV), 143
Ganze Zahlen (16 Bit) in ganze Zahl (32 Bit) multiplizieren (MUL), 143
Hexadezimalzahl in ASCII-Zeichenkette wandeln, 97
Hexadezimalzahl in Bit wandeln, 106

Impuls (TP), 201
Impulsausgabe (PLS), 126
Impulsdauermodulation (PWM), 126
Impulsfolge (PTO), 126
In Netz schreiben, 76
In STOP gehen, 169
Inkrementieren, 145
Interrupt, 156–162
Interrupt trennen, 156
Interrupt zuordnen, 156
Interruptprogramm absolut beenden, 156
Interruptprogramm bedingt beenden, 156, 203
Letzten Wert aus Tabelle löschen, 191
MBUS_INIT, 343
MBUS_SLAVE, 345
Meldung aus Zwischenspeicher übertragen, 80
Meldung in Zwischenspeicher empfangen, 80
Modbus-Slave-Protokoll, 342
Modemmodul EM 241, 306
Modus für schnellen Zähler definieren (HDEF), 112
MODx_CTRL, 307
MODx_MSG, 308
MODx_XFR , 307
Multiplizieren, 141
Natürlicher Exponent, 144
Natürlicher Logarithmus, 144
NOT, 66
Nulloperation, 70
Obersten Stackwert aus Stack schieben, 72
Obersten Stackwert duplizieren, 72
ODER, 164
Ohne Ausgänge, 55
PID-Regler, 146
Positioniermodul EM 253, 265
POSx_CFG, 275
POSx_CLR, 274
POSx_CTRL , 266
POSx_DIS, 274
POSx_GOTO, 268
POSx_LDOFF, 271
POSx_LDPOS, 272
POSx_MAN, 267
POSx_RSEEK , 270
POSx_RUN, 269
POSx_SRATE, 273
Programmschleife mit FOR, 171
Quadratwurzel einer Realzahl ziehen, 144
Realzahl in ASCII-Zeichenkette wandeln, 99
Realzahl in ganze Zahl (32 Bit) wandeln, 95
Realzahl in Zeichenkette wandeln, 101, 104
Realzahl übertragen, 166
Regler (PID), 146
Rücksetzen, 70
Rückwärtszählen, 107
Rückwärtszähler, 110
Schnelle Impulsausgabe (PLS), 126
Schnellen Zähler aktivieren (HSC), 112
Schnittstellenadresse einstellen, 89
Schnittstellenadresse holen, 89
Setzen, 70
Sinus, 144
Speicher mit Bitmuster belegen, 192
Sprungmarke definieren, 173
Stack laden, 72
Standardkontakt, 66
Steigende Flanke, 66
Subtrahieren, 141
Tabelle, 191–196
Tangens, 144
Teilzeichenkette aus Zeichenkette kopieren, 187
Teilzeichenkette in ganze Zahl (32 Bit) wandeln, 101, 104
Teilzeichenkette in ganze Zahl wandeln, 101, 104
Teilzeichenkette in Realzahl wandeln, 101, 104
Transition Ablaufsteuerungsrelais, 174
Überwachungszeit rücksetzen, 169
UENO, 72
UND-Verknüpfung, 164
Unterprogramm aufrufen, 203
Unterprogramm beenden, 203
USS-Protokoll, 322
Vergleichsoperationen, 55
Vorwärts-/Rückwärtszählen, 107
Vorwärts-/Rückwärtszähler, 110
Vorwärtszählen, 107
Vorwärtszähler, 110
Wert in Schieberegister schieben, 182
Wert in Tabelle eintragen, 190
Wert in Tabelle suchen, 193
Wort links rotieren, 180
Wort links schieben, 180
Wort rechts rotieren, 180
Wort rechts schieben, 180
Wort übertragen, 166
Zahl runden, 95
Zeichenkette in Zeichenkette suchen, 188
Zeichenkette kopieren, 185
Zeichenkette verketten, 185
Zeichenkettenlänge, 185
Zeit als Ausschaltverzögerung starten (TOF), 196, 201
Zeit als Einschaltverzögerung starten (TON), 196, 201
Zeit als speichernde Einschaltverzögerung starten (TONR), 196
Zu Sprungmarke springen, 173
Zuweisen, 70
Zweiten Stackwert kopieren, 72
Operationen für das Ablaufsteuerungsrelais
Ablaufsteuerungsrelais bedingt beenden, 174
Ablaufsteuerungsrelais laden, 174
Beispiel, 174
Einschränkungen, 174
Ende Ablaufsteuerungsrelais, 174
Teilung von Ablaufketten, 177
Transition Ablaufsteuerungsrelais, 174
Zusammenführung von Ablaufketten, 177
Operationen für das USS-Protokoll
Beispielprogramm, 330
Fehlercodes bei der Ausführung, 331
Richtlinien für die Verwendung, 322
USS4_DRV_CTRL, 324
USS4_INIT, 323
USS4_RPM_x und USS4_WPM_x, 327, 328
Operationen für Unterprogramme
Beispiel, 206
Interruptprogramm bedingt beenden, 203
Operationen, Kurzinformation, 455
Operations-Assistent
HSC, 113
NETR, NETW, 76
Operationsbaum, 9, 51
Operationsbibliotheken, 60

Operator-Stationen, Definieren, 48
Optimieren, Leistungsfähigkeit im Netz, 233
Optionen für die RP-Suchmodi, 261–265
  Positioniermodul EM 253, 261–265

P

Parameter
  In Unterprogrammen, 204
  Typen für Unterprogramme, 204
Paritätsfehler
  SMB3, 429
  SMB30 und SMB130, 87
Passwort
  CPU-Funktionen, 44
  Einrichten, 44
  Löschen, 45
  Verloren, 45
  Zugriffsschutz, 45
Passwortschutz, Modemmodul EM 241, 299
Pauseerkennung, 85
Physikalische Eingänge , Reaktion des Positioniermoduls EM 253, 256
PID-Regler, 146
  Alarmfunktionen, 152
  Assistent, 146
  Automatikbetrieb, 152
  Beispiel, 154
  Bereiche, 151
  Beschreibung, 147
  Betriebsarten, 152
  Differentialanteil, 149
  Fehlerbedingungen, 152
  Handbetrieb, 152
  Integralanteil, 148
  Normalisieren der Prozessvariablen, 150
  Programmbeispiel, 153
  Proportionalanteil, 148
  Reglerarten, 149
  Rückwärtsverhalten, 151
  Tabelle für den Regelkreis, 153
  Umwandeln der Eingangswerte, 150
  Umwandeln der Stellgröße in eine skalierte ganze Zahl, 151
  Variablen, 151
  Vorwärtsverhalten, 151
Pipeline, PTO-Impulse, 127
Pointer, Indirekte Adressierung, 32
Positionier-Assistent, 126, 254
Positionierbefehle, Positioniermodul EM 253, 292
Positioniermodul
  ACCEL_TIME, 257
  Aktivierungspegel Eingang, 255
  Anzeigen und Steuern des Betriebs, 281
  Beispielprogramm für die Steuerung, 294
  Beseitigen von Spiel, 264
  Betriebsart Profil, 260
  Bewegungsprofil, Definieren, 260
  DECEL_TIME, 257
  Diagnoseinformationen, 282
  Eingänge und Ausgänge, 252
  Eingangs- und Ausgangskonfigurationen, 255
  EM 253 Steuer-Panel, 281–283
  Erstellen von Operationen, 294
  Fehlercodes, 283
  Fehlercodes der Operationen, 283
  Fehlercodes des Moduls, 284
  Funktionen, 252
  Impulsausgänge und Richtungsausgänge, 255
  Konfiguration, 282
  Konfigurations-/Profiltabelle, 285
  Konfigurieren, 254
  MAX_SPEED, 256
  Messart, Einstellen, 254
  Negative Polarität, 255
  Operationen, 265
  Positionier-Assistent, 254
  Positionierbefehle , 292
  Positive Polarität, 255
  POSx_CFG, 275
  POSx_CLR, 274
  POSx_CTRL, 266
  POSx_DIS, 274
  POSx_GOTO, 268
  POSx_LDOFF, 271
  POSx_LDPOS, 272
  POSx_MAN, 267
  POSx_RSEEK, 270
  POSx_RUN, 269
  POSx_SRATE, 273
  Profile, 260
  Profilschritte, 261
  Programmierung, 253
  Referenzpunkt , 258
  Referenzpunktsuche, 258
  Richtlinien für Operationen, 265
  RP-Suchmodi, 261–265
  RP-Suchsequenz, 259
  Ruckausgleichszeit, 258
  Sondermerker, 289
  SS_SPEED, 256
  Tippbetriebsparameter, 257
  Zeiten Eingabefilter, 255
Positioniermodul EM 253
  ACCEL_TIME, 257
  Aktivierungspegel Eingang, 255
  Anzeigen und Steuern des Betriebs, 281
  Befehlsbyte, 291
  Beispiele, 276–280
  Beispielprogramm zum Steuern des Moduls, 294
  Beseitigen von Spiel, 264
  Betriebsart Profil, 260
  Bewegungsprofil, Definieren, 260
  CPU-Kompatibilität, 402
  DECEL_TIME, 257
  Diagnoseinformationen, 282
  Eingänge und Ausgänge, 252
  EM 253 Steuer-Panel, 281–283
  Erstellen von Operationen, 294
  Fehlercodes, 283
  Fehlercodes der Operationen, 283
  Fehlercodes des Moduls, 284
  Funktionen, 252
  Impulsausgänge und Richtungsausgänge, 255
  Konfiguration, 282
  Konfigurations-/Profiltabelle, 285
  Konfigurieren, 254
  Maximale Drehzahl und Start-/Stoppdrehzahl, 256
  Messart, Einstellen, 254
  Negative Polarität, 255
  Operationen, 265

Positioniermodul EM 253 (Fortsetzung)
Positionier-Assistent, 254
Positionierbefehle, 292
Positive Polarität, 255
POSx_CFG, 275
POSx_CLR, 274
POSx_CTRL, 266
POSx_DIS, 274
POSx_GOTO, 268
POSx_LDOFF, 271
POSx_LDPOS, 272
POSx_MAN , 267
POSx_RSEEK, 270
POSx_RUN, 269
POSx_SRATE, 273
Profile, 260
Profilschritte, 261
Programmierung, 253
Reaktion auf physikalische Eingänge, 256
Referenzpunkt (RP), 258
Referenzpunktsuche (RP-Suche), 258
Richtlinien für Operationen, 265
RP-Suchmodi, 261–265
RP-Suchsequenz, 259
Ruckausgleichszeit, 258
Schaltplan, 404–405
Sondermerker, 289
SS_SPEED, 256
Tippbetriebsparameter, 257
Zeiten Eingabefilter, 255
Positive Polarität, Positioniermodul EM 253, 255
POSx_CFG, 275
POSx_CLR, 274
POSx_CTRL, 266
POSx_DIS, 274
POSx_GOTO, 268
POSx_LDOFF, 271
POSx_LDPOS, 272
POSx_MAN, 267
POSx_RSEEK, 270
POSx_RUN, 269
POSx_SRATE, 273
Potentialtrennung
Netz, 220
Verdrahtungsrichtlinien, 18
Potientiometer, Analogpotentiometer, 45
PPI-Kommunikation, Wechseln in die frei programmierbare Kommunikation, 82
PPI-Modus, Funkmodem, 232
PPI-Multi-Master-Kabel, 4
Einstellen, 208
Frei programmierbare Kommunikation, 226
Konfigurieren, 229
Konfiguring für frei programmierbare Kommunikation, 230
Mit Funkmodem, 232
Mit Modem, 231
RS-232-Standard, 226
Schalterstellungen für Baudrate, 231
Funkmodem, 232
Wählen, 223
PPI-Protokoll, 212, 235
Komplexes Netz, 215
Netz mit einem Master, 214
Netz mit mehreren Mastern, 214
PPI/frei programmierbare Kommunikation, Funkmodem, 232
Priorität
Interruptereignisse, 161
Interruptprogramme, 160
PROFIBUS, Master- und Slave-Geräte, 209
PROFIBUS-DP
Beispielprogramm, 396
Datenkonsistenz, 391
Modul (EM 277), 388
Standard-Kommunikation, 387
PROFIBUS-DP-Netz
Anschlussbelegung, 221
Busverstärker, 220
S7-315-2 und EM 277, 216
STEP 7-Micro/WIN und HMI-Gerät, 216
Technische Daten des Kabels, 220
PROFIBUS-Protokoll, 213, 235
Profil der Bewegungsbeschreibung, Positioniermodul EM 253, 260
Profildefinitionstabelle PTO0, PTO1 (SMB166 bis SMB185), 438
Profilschritte, Positioniermodul EM 253, 261
Profiltabelle
Positioniermodul EM 253, 285
PTO/PWM-Generatoren, 131
Programm
Analogeingänge, 22
Anlegen, 9
Anlegen mit STEP 7-Micro/WIN, 51
Ausführen, 12
Ausführungsfehler, 57
Bearbeitung im Betriebszustand RUN, 244
Beobachten, 12
Beobachten des Status, 246
Bestimmte Anzahl Zyklen ausführen, 248
Gemeinsames Nutzen von Daten in Interruptprogrammen, 158
Grundlegende Komponenten, 49
Kopieren in Speichermodul, 36
Laden aus CPU, 34
Laden in CPU, 12, 34
PID-Beispiel, 153
Speichern, 11
Speicherung, 34–38
Statustabelle, 59
Struktur, 49
Testfunktionen, 244
Übersetzungsfehler, 56
Unterprogramme, 50
Zurückholen vom Speichermodul, 37
Programm, Beispiel, Steuern des Positioniermoduls, 294
Programm-Editor, Auswählen, 51
Programm-Editoren, 51
Anweisungsliste (AWL), 51
Funktionsplan (FUP), 51
Kontaktplan (KOP), 51
Konventionen, 54
Öffnen, 9
Programmierkabel, 4
Programmiersoftware, Bestellnummern, 442
Programmierung, schnelle Zähler, 113
Programmierung , Positioniermodul EM 253, 253
Programmierung zur Laufzeit, Fehlercodes, 425

Programmschleife mit FOR/NEXT
Ende Programmschleife mit NEXT, 171
Programmschleife mit FOR, 171
Programmsteuerungsoperationen
Ablaufsteuerungsrelais (SCR), 174
Bearbeitung bedingt beenden, 169
Grundlegende Programmsteuerung, 169
In STOP gehen, 169
Programmschleife mit FOR/NEXT, 171
Sprungoperationen, 173
Überwachungszeit rücksetzen, 169
Proportionalanteil, PID-Algorithmus, 148
Protokoll, Kommunikation
Anwenderdefiniert, 225
Einstellen, 212
Protokolle
PROFIBUS-DP, 387
Unterstützt von STEP 7-Micro/WIN, 224
Prozessabbild, 39
Prozessabbild der Ausgänge, 22
Prozessabbild der Ausgänge (A), 25
Prozessabbild der Eingänge, 23
Prozessabbild der Eingänge (E), 25
Prozessvariablen
Normalisieren, 150
Umwandeln, 150
PTO/PWM-Funktionen, Register (WMB66 bis SMB85), 436
Pufferkonsistenz, PROFIBUS, 391
Punkt-zu-Punkt-Kommunikation, 215–216

Q

Quadratwurzel einer Realzahl ziehen, 144

R

Realzahl in ASCII-Zeichenkette wandeln, 99
Realzahl in ganze Zahl (32 Bit) wandeln, 95
Realzahl in Zeichenkette wandeln, 101, 104
Realzahl übertragen, 166
Realzahlenvergleich, 90
Realzahlenwerte, 24, 29
Referenzpunkt (RP), 258
RP-Suche
RP_APPR_DIR, 258
RP_FAST, 258
RP_SEEK_DIR, 258
RP_SLOW, 258
Suchrichtung, 258
RP-Suchsequenz, Arten, 259
RP_OFFSET, 259
Regeln, PID-Regler, 146, 148
Register HSC0, HSC1, HSC2 (SMB36 bis SMB65), 433
Register HSC3, HSC4, HSC5 (SMB131 bis SMB165), 437
Regler
Anpassen der Integralsumme, 151
Auswählen, 149
Bereiche/Variablen, 151
Betriebsarten, 152
Fehlerbedingungen, 152
Operationen, 146–157
Programmbeispiel, 153–155
Umwandeln der Eingaben, 150
Umwandeln der Stellgröße, 151
Vorwärts-/Rückwärtsverhalten, 151
Reihenfolge, Interruptereignisse, 161
Relais, 20
Lebensdauer, 349
Remanente Merker, 35
Remanenter Speicher, 41
Richtlinien
Ändern eines Pointers für die indirekte Adressierung, 32
Einbau, 14
Entwerfen eines Systems für eine Micro-SPS, 48
Erdung und Stromkreise, 18
Erdung und Verdrahtung, 19
Interruptprogramme, 50, 158
Netzkonfiguration, 220
Operationen für das Positioniermodul EM 253, 265
Potentialtrennung, 18
Schutzbeschaltungen, 20
Umgebungen mit starken Schwingungen, 16
Unterprogramme, 50
Verdrahtung, 18
Vertikaler Einbau, 16
Richtlinien für AC-Aufbau, 19
Richtlinien für DC-Aufbau, 19
RJ11-Stecker, Modemmodul EM 241, 296, 399
Rotieroperationen, 180
Arten, 180
Beispiel, 181
RS-232-Standard
Frei programmierbare Kommunikation, 226
PPI-Multi-Master-Kabel, 226
RS-232/PPI-Multi-Master-Kabel
DIP-Schalterstellungen, 416
Konfigurieren im HyperTerminal, 239–483
Schnittstelleneinstellungen, 219
Technische Daten, 414
RS-485-Standard, 221
RTD-Modul (EM 231), 380
Ruckausgleichszeit, Positioniermodul EM 253, 258
Rückruf, Modemmodul EM 241, 300
Rücksetzen, 69, 70
Schnelle Zähler, 117
Rückwärtsverhalten im Regelkreis, 151
Rückwärtszählen, 107
Rückwärtszähler, 110

S

S7-200
Ablaufsteuerungsrelais (S), 29
Abmessungen, 2
Adressierung, 24
Akkumulatoren, 27
Als Slave-Gerät, 387
Als Slave-Geräte, 209
Analogausgänge (AA), 29
Analogeingänge (AE), 29
Anschließen über STEP 7-Micro/WIN, 8
Ausführung der Steuerungslogik, 22
Baudraten, 209, 210–212
Beispiel für eine Netzkonfiguration, 217
Beispiele für Netzkonfigurationen, 214, 215, 216
Betriebszustand RUN, 12, 37

S7-200 (Fortsetzung)
Betriebszustand STOP, 12, 37
CPUs, 2
Einbau, 15
Einbaurichtlinien, 14
Elektromagnetische Verträglichkeit, 350
Erdungsrichtlinien, 19
Erweiterungsmodule, 3
Fehlerbehebung, 56, 249
Fehlercodes, 424
Funkmodem, 232
Hardware-Fehlerbehebung, 249
Impulsabgriff, 42
Interruptprogramme, 158
Kommunikationsbearbeitung, 40
Konstante Werte, 30
Laden aus CPU, 34
Laden in CPU, 34
Lesen und Schreiben von Daten, 22
Lokaldatenspeicher (L), 28
Modem, 231
Netzadresse, 210–212
Passwortschutz, 44
Programmierkonventionen, 55
Prozessabbild, 39
Prozessabbild der Ausgänge (A), 25
Prozessabbild der Eingänge (E), 25
Reaktion auf schwere Fehler, 424
Remanenter Speicher, 41
RS-232, 226
Schaltpläne, 354–356
Schnelle Zähler, 27
Sonderfunktionen, 39
Sondermerker (SM), 28
Spannungsversorgung, 6
Speicher, 24
Speicherbereich A, 25
Speicherbereich der Merker (M), 25
Speicherbereich E, 25
Speicherbereich L, 28
Speicherbereich M, 25
Speicherbereich S, 29
Speicherbereich SM, 28
Speicherbereich T, 26
Speicherbereich Z, 26
Speicherbereiche, 64, 483
Speichermodul, 36
Speichern von Daten, 34
Statustabelle, 247
Systemkomponenten, 2
Technische Daten, 350
Umgebungsbedingungen, 350
Unterstützte Interruptprogramme, 159
Variablenspeicher, 25
Variablenspeicher (V), 25
Verdrahtungsrichtlinien, 19
Zähler (Z), 26
Zeiten (T), 26
Zugreifen auf Daten, 24
Zyklus, 22, 39
S7-200 Systemhandbuch, Bestellnummern, 442
S7-300, Beispiele für Netzkonfigurationen, 215
S7-400, Beispiele für Netzkonfigurationen, 215
Schachtelung, Unterprogramme, 203
Schaltbild, Modemmodul EM 241, 398
Schaltbild der Ausgänge
EM 232, 370
EM 235, 370
Schaltbild der Eingänge
EM 231, 368
EM 235, 369
Schaltpläne
Analogerweiterungsmodule, 364
CPUs, 354–356
Digitalerweiterungsmodule, 360–362
Eingänge und Ausgänge der CPU, 354
Positioniermodul EM 253, 404–405
Schalttafelmontage, 16
Schiebeoperationen
Arten, 180
Beispiel, 181
Schnelle Impulsausgabe
Ändern der Impulsdauer, 133
Funktionsweise, 126
Schnelle Impulsausgänge, 46
Schnelle Zähler, 46
Adressierung, 120
Ändern des aktuellen Werts, 124
Ändern des voreingestellten Werts, 124
Beschreibung, 113
Betriebsarten, 115
Deaktivieren, 124
Definieren von Zählerarten und Eingängen, 114
Einrichten des Steuerbyte, 119
Einstellen des aktiven Zustands, 118
Einstellen des aktuellen und des voreingestellten Werts, 119
Funktionsweise der Rücksetz- und Starteingänge, 117
Impulsdiagramme, 115–117
Initialisierungsreihenfolge, 121
Interrupts, 114
Programmierung, 113
SMB36-SMB65, 433
Speicherbereich, Adressierung, 27
Statusbyte, 120
Steuerbyte, 113
Zählarten, 454
Zählrichtungswechsel, 124
Zuweisen von Interrupts, 120
Schnellen Zähler aktivieren (HSC), 112
Schnittstelle
Einstellen der Kommunikation, 208
Einstellungen, PPI-Multi-Master-Kabel, 219
Schnittstelle zu Telefonleitungen, international, Modemmodul EM 241, 296
Schnittstellenadresse einstellen, 89
Schnittstellenadresse holen, 89
Schrittmotorsteuerung, PTO/PWM-Generatoren, 131
Schutzbeschaltungen, 20
Schutzkreis, 349
Schwere Fehler, 57
Ansehen, 424
Setzen, 69, 70
Sicherheitsrückruf, Modemmodul EM 241, 300
Sicherheitsstromkreise, Entwerfen, 48
Sieben-Segment-Anzeige, 96
Signalfluss, Parameter für Unterprogramm, 204
SIMATIC Befehlssatz, 53

SIMATIC Zähler
Beispiele, 109
Rückwärtszählen, 107
Vorwärts-/Rückwärtszählen, 107
Vorwärtszählen, 107
SIMATIC Zeiten, 196
Beispiel, 198, 199, 200
Simulatoren, Bestellnummern, 443
Simulatoren, Eingang, 418
Single-Master-PPI-Netz, 214
Sinus, 144
Skalierte ganze Zahl, Umwandeln der Stellgröße, 151
Slave-Gerät
EM 277 PROFIBUS-DP, 387
S7-200, 387
Slave-Geräte, 209
SM0.2: Remanente Daten verloren, 35
SMB0: Statusbits, 428
SMB1: Statusbits, 428
SMB130: Steuerungsregister der frei programmierbaren Kommunikation, 432
SMB131 bis SMB165: Register HSC3, HSC4, HSC5, 437
SMB166 bis SMB185: Profildefinitionstabelle PTO0, PTO1, 438
SMB186 bis SMB194: Steuerung des Meldungsempfangs, 436
SMB2: Empfangene Zeichen in der frei programmierbaren Kommunikation, 429
SMB200 bis SMB549: Status intelligentes Modul, 439
SMB28, SMB29 Analogpotentiometer, 45, 432
SMB3: Paritätsfehler in der frei programmierbaren Kommunikation, 429
SMB30 und SMB130: Steuerungsregister der frei programmierbaren Kommunikation, 432
SMB31 und SMW32: Schreibsteuerung EEPROM, 433
SMB34 und SMB35: Register für zeitgesteuerte Interrupts, 433
SMB36 bis SMB65: Register HSC0, HSC1, HSC2, 433
SMB4: Überlauf der Warteschlange, 429
SMB5: E/A-Status, 430
SMB6: Kennregister der CPU, 430
SMB66 bis SMB85: PTO/PWM-Register, 436
SMB7: Reserviert, 430
SMB8 bis SMB21: Kenn- und Fehlerregister des E/A-Moduls, 431
SMB86 bis SMB94, SMB186 bis SMB194: Steuerung des Meldungsempfangs, 436
SMS, Modemmodul, 298
SMW22 bis SMW26: Zykluszeiten, 432
SMW98: Busfehler Erweiterungs-E/A, 437
Software-Test, 243
Sondermerker, 428–436
Kurzinformation, 451
Modbus-Slave-Protokoll, 340
Modemmodul EM 241, 312
Positioniermodul EM 253, 289
PTO/PWM-Funktion, 129
Sondermerker (SM), 28
Analogpotentiometer, 45
Sondermerkerbytes, EM 277 PROFIBUS-DP, 392
Spannungsverlust, Remanenter Speicher, 35, 41
Spannungsversorgung, 6, 15
Speicher
CPU, 2
Remanent, 41
Zugreifen, 24
Zurückholen aus dem EEPROM, 35
Speicher mit Bitmuster belegen, 192
Speicherbereich A, 25
Speicherbereich der Merker (M), 25
Speichern bei Spannungsverlust, 35
Speicherbereich E, 25
Speicherbereich L, 28
Speicherbereich M, 25
Speicherbereich S, 29
Speicherbereich SM, 28
Speicherbereich T, 26
Speicherbereich Z, 26
Speicherbereiche
CPUs, 64, 483
Kurzinformation, 453
Operandenbereiche, 65
Urlöschen, 45
Speicherfunktionen
Bytes im Wort tauschen, 184
Rotieroperationen, 180
Schiebeoperationen, 180
Übertragungsoperationen, 166
Wert in Schieberegister schieben, 182
Wertebereich übertragen, 168
Speichermodul, 34, 36, 413
Bestellnummer, 442
Fehlercodes, 424
Kopieren des Programms, 36
Stecken, 36
Zurückholen des Programms, 37
Speichern
Merker (M) im EEPROM, 35
Programm, 11
S7-200 Programmdaten, 34
Variablenspeicher (V) im EEPROM, 38
Wert im EEPROM, 433
Spiel
Ausgleich, 259
Wählen des Arbeitsbereichs , 264
Sprungmarke definieren, 173
Sprungoperationen
Sprungmarke definieren, 173
Zu Sprungmarke springen, 173
Spulen
Bitwert direkt rücksetzen, 70
Bitwert direkt setzen, 70
Bitwert direkt zuweisen, 70
Nulloperation, 70
Rücksetzen, 70
Setzen, 70
Zuweisen, 70
SS_SPEED, Positioniermodul EM 253, 256
Stack laden, 72
Stackoperationen
Beispiel, 73
Erste und zweite Stackebene durch ODER verknüpfen, 72
Erste und zweite Stackebene durch UND verknüpfen, 72
Obersten Stackwert aus Stack schieben, 72
Obersten Stackwert duplizieren, 72
Stack laden, 72
UENO, 72
Zweiten Stackwert kopieren, 72
Standard-Hutschiene, 15
Standard-Hutschienenmontage, 16

Standardkontakt, 66
Starten, Schnelle Zähler, 117
Startzeichenerkennung, 84
Status
  Anzeigen in AWL, 247
  Anzeigen in KOP und FUP, 246
  Ausführung, 246
  Beobachten des Programms, 246
  Zyklusende, 246
Status-LEDs
  EM 231 RTD, 382
  EM 231 Thermoelement, 377
  EM 277 PROFIBUS-DP, 386, 393
  Positioniermodul EM 253, 402
Statusbyte, Schnelle Zähler, 120
Statustabelle, 59
  Beobachten von Werten, 247
  Forcen von Werten, 248
Steckleitung, 413
Steckmodule
  Bestellnummern, 442
  Speicher, 413
Steckverbinder, Bestellnummern, 442
Steigende Flanke, 66
Stellgröße, Umwandeln in skalierte ganze Zahl, 151
STEP 7-Micro/WIN
  Als Master-Gerät, 209
  Anlegen von Programmen, 51
  Anschließen an die S7-200, 8
  Aufrufen, 7
  Baudrate, 210–212
  Befehlssätze
    Auswählen, 53
    IEC 1131-3, 53
    SIMATIC, 53
  Beispiele für Netzkonfigurationen, 214–217
  Bestellnummern, 442
  CP-Karte, 224
  Installation, 4
  Kommunikationseinstellungen, 8
  Netzadresse, 210–212
  Öffnen, 51
  PPI-Multi-Master-Kabel, 224
  Programm-Editoren, 51
  Programmierpaket, 3
  Schnittstelle zum EM 241, 297
  Testwerkzeuge, 243
  Voraussetzungen an den PC, 3
  Voraussetzungen an die Ausrüstung, 3
STEP 7-Micro/WIN 32 Operationsbibliotheken, 442
Steuerbyte, Einrichten für HSC, 119
Steuerungslogik, 22
Steuerungsregister der frei programmierbaren Kommunikation (SMB30 und SMB130), 432
Störunterdrückung, Eingangsfilter, 41
Struktur, Programm, 49
Subtrahieren, 141
Suchparameter, Positioniermodul EM 253, 258
Symbolische Adressierung, 58
Symbolische Namen, Anlegen einer Liste, 49
Symboltabelle, 58
  Adressierung, 58
  Konfigurieren für Modbus, 341
Synchrone Aktualisierungen, Operation PWM, 128
Systemdatenbaustein, 50
Systementwurf, Micro-SPS, 48
Systemunterstützung, Für Interruptprogramme, 158

T

Tabelle der Querverweise, 244
Tabelle für den Regelkreis, 153
Tabellenoperationen
  Ersten Wert aus Tabelle löschen, 191
  Letzten Wert aus Tabelle löschen, 191
  Speicher mit Bitmuster belegen, 192
  Wert in Tabelle eintragen, 190
  Wert in Tabelle suchen, 193
Tangens, 144
TC/IP, Kommunikationsprotokoll, 213
TCP/IP , Protokoll, 213
TD 200, Assistent, 4
TD 200 Textdisplay, 4
  Bestellnummer für Handbuch, 442
Technische Daten
  Analogerweiterungsmodul, 363
  AS-Interface-Modul CP 243-2, 411
  CPUs, 351
  Digitales Erweiterungsmodul, 357
  EM 231 RTD, 373
  EM 231 Thermoelement, 373
  EM 241 Modem, 398
  EM 277 PROFIBUS-DP, 385
  Ethernet-Modul CP 243-1, 406
  Internet-Modul CP 243-1 IT, 409
  RS-232/PPI-Multi-Master-Kabel, 414
  System für eine Micro-SPS, 48
Technische Daten der Ausgänge
  Analogerweiterungsmodul, 364
  CPUs, 353
  Digitalerweiterungsmodul, 359
Technische Daten der Eingänge
  Analogerweiterungsmodul, 363
  CPUs, 352
  Digitalerweiterungsmodul, 358
Teilung von Ablaufketten
  Beispiel, 177
  Operationen für das Ablaufsteuerungsrelais, 177
Teilzeichenkette aus Zeichenkette kopieren, 187
Teilzeichenkette in ganze Zahl (32 Bit) wandeln, 101, 104
Teilzeichenkette in ganze Zahl wandeln, 101, 104
Teilzeichenkette in Realzahl wandeln, 101, 104
Teleservice, 297
Temperaturbereiche
  EM 231 RTD, 383–384
  EM 231 Thermoelement, 378–379
Terminal, Konfigurieren des RS-232/PPI-Multi-Master-Kabels, 239–483
Testen
  Bearbeitung im Betriebszustand RUN, 244
  Forcen von Werten, 248
  Funktionen, 244
  Mehrere Zyklen, 248
Textdisplay TD 200, Bestellnummer, 443
Textfunkruf, Modemmodul EM 241, 298
Thermoelement-Modul (EM 231)
  Grundlagen, 375
  Konfigurieren, 375
  Statusanzeigen, 377
  Temperaturbereiche, 378–379
Timer für die Zeit zwischen den Zeichen, 86

Tippbetriebsparameter
Positioniermodul EM 253, 257
Tippbetrieb, 257
TOD-Uhr, 75
Token-Ring-Netz, Beispiel, 234
Token-Umlaufzeit, 233
Vergleich, 235
Touch Panel TP070, Bestellnummer, 443
TP-Designer für TP070, Version 1.0, 442
TP070 Touch Panel, 4
Bestellnummer für Handbuch, 442
Transition Ablaufsteuerungsrelais, 174

U

Überlauf der Warteschlange (SMB4), 429
Übersetzungsfehler, 56
Übertragungsgeschwindigkeit, Netz, 220
Übertragungsoperationen, Beispiel, 184
Überwachungszeit rücksetzen, 169
UENO, 72
Uhr
Modul, 413
Statusbits, 428
Uhroperationen
Echtzeituhr lesen, 75
Echtzeituhr schreiben, 75
Uhrzeit, Einstellung, 75
Umgebungsbedingungen
Betrieb, 350
Transport und Lagerung, 350
Umwandeln
Prozessvariablen, 150
Stellgröße in skalierte ganze Zahl, 151
Umwandlungsoperationen
ASCII, 97
ASCII-Zeichenkette in Hexadezimalzahl wandeln, 97
Beispiel, 95
Bit in Hexadezimalzahl wandeln, 106
Bitmuster für Sieben-Segment-Anzeige erzeugen, 96
Genormte, 93
Hexadezimalzahl in ASCII-Zeichenkette wandeln, 97
Hexadezimalzahl in Bit wandeln, 106
Realzahl in ganze Zahl (32 Bit) wandeln, 95
Zahl runden, 95
Zeichenkette, 101, 104
UND-Verknüpfung, 164
Beispiel, 165
Unterprogramm, Unterprogramm aufrufen, 203
Unterprogramm aufrufen, 203
Unterprogramm beenden, 203
Unterprogramme
Aufrufen von Interruptprogrammen, 159
Beispiel, 49
Datentypen, 204
Mit Parametern, 204
Parametertypen, 204
Richtlinien, 50
Schachtelung, 203
Signalflussparameter, 204
USB/PPI-Multi-Master-Kabel
Bestellnummer, 414
Einstellen, 208
Schnittstelleneinstellungen, 219
USS-Protokoll, Anforderungen, 320

V

Variablen
Beobachten, 59
Beobachten in einer Statustabelle, 247
PID-Regler, 151
Symbolische Adressierung, 58
Variablenspeicher, 25
Speichern im EEPROM, 38
Zuordnen von Adressen, 58
Zurückholen aus dem EEPROM, 35
Variablenspeicher (V), 25
Verdrahtung, 18, 19
Vergleich, Token-Umlaufzeiten, 235
Vergleichsoperationen, 55
Beispiel, 90
Bytevergleich, 90
Doppelwortvergleich, 90
Ganzzahlenvergleich, 90
Realzahlenvergleich, 90
Zeichenkettenvergleich, 92
Verknüpfungsoperationen
Invertieren, 163
UND, ODER, EXKLUSIV ODER, 164
Verletzung der Übersetzungsregeln, 426
Verlorenes Passwort, 45
Vermeiden, Konflikte im Netz, 236
Verwalten, Komplexe Kommunikation, 236
Voraussetzungen an die Ausrüstung, 3
Voreingestellter Wert
Ändern in HSC, 124
Einstellen für HSC, 119
Vorgehensweise, Bei verlorenem Passwort, 45
Vorwärts-/Rückwärtszählen, 107
Vorwärts-/Rückwärtszähler, 110
Vorwärtsverhalten im Regelkreis, 151
Vorwärtszählen, 107
Vorwärtszähler, 110

W

Wählen
CP-Karte, 223
PPI-Multi-Master-Kabel, 223
Warteschlange für E/A-Interrupts, 160
Warteschlange für Kommunikationsinterrupts, 160
Warteschlange für zeitgesteuerte Interrupts, 160
Warteschlangen, Interruptprogramme, 160
Wechseln, Zählrichtung bei schnellen Zählern, 124
Weiterschaltbedingungen, Beispiel, 179
Wert in Schieberegister schieben, 182
Wert in Tabelle eintragen, 190
Wert in Tabelle suchen, 193
Wiederanlauf, Nach schwerem Fehler, 57
Wiedereinbau, Klemmenblock, 17
Wort links rotieren, 180
Wort rechts rotieren, 180
Wort rechts schieben, 180
Wort übertragen, 166
Wortkonsistenz, PROFIBUS, 391

Z

Zahl runden, 95
Zahlen, Darstellung, 24, 29, 30
Zähler
  IEC
    Rückwärtszähler, 110
    Vorwärts-/Rückwärtszähler, 110
    Vorwärtszähler, 110
  Modus für schnellen Zähler definieren (HDEF), 112
  Schnelle Zähler, 46
  Schnellen Zähler aktivieren (HSC), 112
  SIMATIC
    Rückwärtszählen, 107
    Vorwärts-/Rückwärtszählen, 107
    Vorwärtszählen, 107
Zähler (Z), 26
Zählrichtung, Ändern bei HSC, 124
Zählwert für die Höchstzeichenzahl, 87
Zeichenkette in Zeichenkette suchen, 188
Zeichenkette kopieren, 185
Zeichenkette verketten, 185
Zeichenkettenlänge, 185
Zeichenkettenoperationen
  Erstes Zeichen in Zeichenkette suchen, 188
  Teilzeichenkette aus Zeichenkette kopieren, 187
  Zeichenkette in Zeichenkette suchen, 188
  Zeichenkette kopieren, 185
  Zeichenkette verketten, 185
  Zeichenkettenlänge, 185
Zeichenkettenvergleich, 92
Zeichnungen, Konfigurationspläne, 49
Zeit als Ausschaltverzögerung starten (TOF), 196, 201
Zeit als Einschaltverzögerung starten (TON), 196, 201
Zeit als speichernde Einschaltverzögerung starten (TONR), 196
Zeitauflösung, 197, 198
Zeiten
  IEC
    Impuls (TP), 201
    Zeit als Ausschaltverzögerung starten (TOF), 201
    Zeit als Einschaltverzögerung starten (TON), 201
  Interrupts, 159
  SIMATIC
    Zeit als Ausschaltverzögerung starten (TOF), 196
    Zeit als Einschaltverzögerung starten (TON), 196
    Zeit als speichernde Einschaltverzögerung starten (TONR), 196
Zeiten (T), 26
Zeiten Eingabefilter, Positioniermodul EM 253, 255
Zeitgesteuerte Interrupts, 159
  Intervallregister (SMB34, SMB35), 433
Ziehen, Speichermodul, 36
Zu Sprungmarke springen, 173
Zugreifen
  Direkte Adressierung, 24
  S7-200 Daten, 24, 32
  Statustabelle, 59
Zugriff im Wortformat, 24
Zugriffsbeschränkung, 44
Zugriffsschutz, Passwort, 44
Zulassungen der Behörden für Seewesen, 348
Zuordnen
  Adressen, 58
  Anfangswerte, 58
  Netzadressen, 209
Zurückholen
  Daten aus dem EEPROM, 35
  Programm vom Speichermodul, 37
Zusammenführung von Ablaufketten
  Beispiel, 178
  Operationen für das Ablaufsteuerungsrelais, 177
Zuweisen, 70
  Interrupts zu HSC, 120
Zweiten Stackwert kopieren, 72
Zyklus, 22
  Bestimmte Anzahl, 248
  Zeiten, 198
Zykluszeit (Funktion PTO), 126
Zykluszeiten: SMW22 bis SMW26, 432

To

SIEMENS ENERGY & AUTOMATION INC
ATTN: TECHNICAL COMMUNICATIONS
ONE INTERNET PLAZA
PO BOX 4991
JOHNSON CITY TN USA 37602-4991

From

Name: _

Job Title: _

Company Name: _

Street: _

City and State: _

Country: _

Telephone: _

Please check any industry that applies to you:

- ❐ Automotive
- ❐ Chemical
- ❐ Electrical Machinery
- ❐ Food
- ❐ Instrument and Control
- ❐ Non-electrical Machinery
- ❐ Petrochemical
- ❐ Pharmaceutical
- ❐ Plastic
- ❐ Pulp and Paper
- ❐ Textiles
- ❐ Transportation
- ❐ Other ____________________________

Remarks Form

Your comments and recommendations will help us to improve the quality and usefulness of our publications. Please take the first available opportunity to fill out this questionnaire and return it to Siemens.

Please give each of the following questions your own personal mark within a range from 1 (very good) to 5 (very poor).

1. Do the contents meet your requirements? ☐
2. Is the information you need easy to find? ☐
3. Is the text easy to understand? ☐
4. Does the level of technical detail meet your requirements? ☐
5. Please rate the quality of the graphics and tables. ☐

Additional comments:

Speicherbereiche und Funktionen der S7-200

Beschreibung	CPU 221	CPU 222	CPU 224	CPU 226	CPU 226XM
Größe Anwenderprogramm	4096 Bytes	4096 Bytes	8192 Bytes	8192 Bytes	16384 Bytes
Größe Anwenderdaten	2048 Bytes	2048 Bytes	5120 Bytes	5120 Bytes	10240 Bytes
Prozessabbild der Eingänge	E0.0 bis E15.7	E0.0 bis E15.7	E0.0 bis E15.7	E0.0 bis E15.7	E0.0 bis E15.7
Prozessabbild der Ausgänge	A0.0 bis A15.7	A0.0 bis A15.7	A0.0 bis A15.7	A0.0 bis A15.7	A0.0 bis A15.7
Analogeingänge (nur Lesen)	--	AEW0 bis AEW30	AEW0 bis AEW62	AEW0 bis AEW62	AEW0 bis AEW62
Analogausgänge (nur Schreiben)	--	AAW0 bis AAW30	AAW0 bis AAW62	AAW0 bis AAW62	AAW0 bis AAW62
Variablenspeicher (V)	VB0 bis VB2047	VB0 bis VB2047	VB0 bis VB5119	VB0 bis VB5119	VB0 bis VB10239
Lokaldatenspeicher (L)[1]	LB0 bis LB63	LB0 bis LB63	LB0 bis LB63	LB0 bis LB63	LB0 bis LB63
Merker (M)	M0.0 bis M31.7	M0.0 bis M31.7	M0.0 bis M31.7	M0.0 bis M31.7	M0.0 bis M31.7
Sondermerker (SM)	SM0.0 bis SM179.7	SM0.0 bis SM299.7	SM0.0 bis SM549.7	SM0.0 bis SM549.7	SM0.0 bis SM549.7
Schreibgeschützt	SM0.0 bis SM29.7	SM0.0 bis SM29.7	SM0.0 bis SM29.7	SM0.0 bis SM29.7	SM0.0 bis SM29.7
Zeiten	256 (T0 bis T255)	256 (T0 bis T255)	256 (T0 bis T255)	256 (T0 bis T255)	256 (T0 bis T255)
Speich. Einschaltverz. 1 ms	T0, T64	T0, T64	T0, T64	T0, T64	T0, T64
10 ms	T1 bis T4 und T65 bis T68	T1 bis T4 und T65 bis T68	T1 bis T4 und T65 bis T68	T1 bis T4 und T65 bis T68	T1 bis T4 und T65 bis T68
100 ms	T5 bis T31 und T69 bis T95	T5 bis T31 und T69 bis T95	T5 bis T31 und T69 bis T95	T5 bis T31 und T69 bis T95	T5 bis T31 und T69 bis T95
Ein-/Ausschaltverzög. 1 ms	T32, T96	T32, T96	T32, T96	T32, T96	T32, T96
10 ms	T33 bis T36 und T97 bis T100	T33 bis T36 und T97 bis T100	T33 bis T36 und T97 bis T100	T33 bis T36 und T97 bis T100	T33 bis T36 und T97 bis T100
100 ms	T37 bis T63 und T101 bis T255	T37 bis T63 und T101 bis T255	T37 bis T63 und T101 bis T255	T37 bis T63 und T101 bis T255	T37 bis T63 und T101 bis T255
Zähler	Z0 bis Z255	Z0 bis Z255	Z0 bis Z255	Z0 bis Z255	Z0 bis Z255
Schnelle Zähler	HC0, HC3, HC4 und HC5	HC0, HC3, HC4 und HC5	HC0 bis HC5	HC0 bis HC5	HC0 bis HC5
Ablaufsteuerungsrelais (S)	S0.0 bis S31.7	S0.0 bis S31.7	S0.0 bis S31.7	S0.0 bis S31.7	S0.0 bis S31.7
Akkumulatoren	AC0 bis AC3	AC0 bis AC3	AC0 bis AC3	AC0 bis AC3	AC0 bis AC3
Sprünge/Sprungmarken	0 bis 255	0 bis 255	0 bis 255	0 bis 255	0 bis 255
Aufrufe/Unterprogramme	0 bis 63	0 bis 63	0 bis 63	0 bis 63	0 bis 127
Interruptprogramme	0 bis 127	0 bis 127	0 bis 127	0 bis 127	0 bis 127
Steigende/fallende Flanken	256	256	256	256	256
PID-Regler	0 bis 7	0 bis 7	0 bis 7	0 bis 7	0 bis 7
Schnittstelle	Schnittstelle 0	Schnittstelle 0	Schnittstelle 0	Schnittstelle 0, Schnittstelle 1	Schnittstelle 0, Schnittstelle 1

[1] LB60 bis LB63 sind für STEP 7-Micro/WIN ab Version 3.0 reserviert.

AWL	Seite
=	69
+D	141
-D	141
* D	141
/ D	141
+I	141
-I	141
=I	69
* I	141
/ I	141
+R	141
-R	141
*R	141
/R	141
ATCH	156
ATH	97
ATT	190
BCDI	93
BIR	167
BIW	167
BMB	168
BMD	168
BMW	168
BTI	93
CALL	204
CFND	188
COS	144
CRET	204
CRETI	156
CSCRE	174
DECB	145
DECD	145
DECO	106
DECW	145
DISI	156
DIV	143
DTA	97
DTCH	156
DTI	93
DTR	93
DTS	101
ED	66
EIN	66
ENCO	106
END	169
ENI	156
EU	66
EXP	144

AWL	Seite
FIFO	191
FILL	193
FND<	194
FND<>	194
FND=	194
FND>	194
FOR	171
GPA	89
HDEF	112
HSC	112
HTA	97
IBCD	93
INCB	145
INCD	145
INCW	145
INVB	163
INVD	163
INVW	163
ITA	97
ITB	93
ITD	93
ITS	101
JMP	173
LBL	173
LD	66
LDB <=	90
LDB =	90
LDB >=	90
LDB >	90
LDB <	90
LDB <>	90
LDD >=	90
LDD <	90
LDD <=	90
LDD =	90
LDD >	90
LDD <>	90
LDI	66
LDN	66
LDNI	66
LDR =	90
LDR <	90
LDR <=	90
LDR >	90
LDR >=	90
LDR <>	90
LDS	71
LDS=	92

AWL	Seite
LDS<>	92
LDW <=	90
LDW <	90
LDW =	90
LDW >	90
LDW >=	90
LDW <>	90
LIFO	191
LN	144
LPP	71
LPS	71
LRD	71
LSCR	174
MOVB	166
MOVD	166
MOVR	166
MOVW	166
MUL	143
NETR	75
NETW	75
NEXT	171
NOT	66
O	66
OB =	90
OB >=	90
OB >	90
OB <	90
OB <=	90
OB <>	90
OD <	90
OD <=	90
OD =	90
OD >	90
OD >=	90
OD <>	90
OI	66
OLD	71
ONI	66
OR =	90
OR <	90
OR <=	90
OR >	90
OR >=	90
OR <>	90
ORB	164
ORD	164
ORW	164
OS=	92

AWL	Seite
OS<>	92
OW <	90
OW <=	90
OW =	90
OW >	90
OW >=	90
OW <>	90
PID	146
PLS	126
R	69
RCV	80
RI	69
RLB	180
RLD	180
RLW	180
ROUND	93
RRB	180
RRD	180
RRW	180
RTA	97
RTS	101
S	69
SCAT	185
SCPY	185
SCRE	174
SCRT	174
SEG	93
SFND	188
SHRB	182
SI	69
SIN	144
SLB	180
SLD	180
SLEN	185
SLW	180
SPA	89
SQRT	144
SRB	180
SRD	180
SRW	180
SSCPY	187
STD	104
STI	104
STOP	169
STR	104
SWAP	184
TAN	144
TODR	74

AWL	Seite
TODW	74
TOF	197
TON	197
TONR	197
TRUNC	93
U	66
UB <=	90
UB =	90
UB >	90
UB <	90
UB >=	90
UB <>	90
UD <	90
UD <=	90
UD =	90
UD >	90
UD >=	90
UD <>	90
UENO	71
UI	66
ULD	71
UN	66
UNDB	164
UNDD	164
UNDW	164
UNI	66
UR =	90
UR <	90
UR <=	90
UR >	90
UR >=	90
UR <>	90
US=	92
US<>	92
UW <	90
UW <=	90
UW =	90
UW >	90
UW >=	90
UW <>	90
WDR	169
XMT	80
XORB	164
XORD	164
XORW	164
ZR	107
ZV	107
ZVR	107