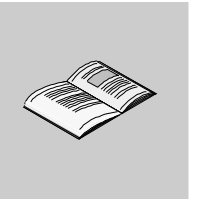
Speicherprogrammierbare Twido-Steuerungen
Software-Referenzhandbuch

TWD USE 10AE ger Version 3.2

a brand of Schneider Electric

Telemecanique

Inhaltsverzeichnis



Sicherheitshinweise 11

Über dieses Buch 15

Teil I Beschreibung der Twido Software 17

Auf einen Blick. 17

Kapitel 1 Einführung in die Twido Software 19

Auf einen Blick. 19
Einführung in TwidoSoft 20
Einführung in Twido Sprachen 21

Kapitel 2 Twido Sprachobjekte 25

Auf einen Blick. 25
Sprachobjektvalidierung 26
Bitobjekte. 27
Wortobjekte 30
Gleitkomma- und Doppelwortobjekte 33
Adressierung von Bitobjekten 37
Adressierung von Wortobjekten 38
Adressierung von Gleitkommaobjekten. 39
Adressierung von Doppelwortobjekten 40
Kennzeichnung der Eingänge/Ausgänge 41
Netzwerk-Adressierung. 43
Funktionsblockobjekte. 44
Strukturierte Objekte 46
Indexierte Objekte 50
Symbolisieren von Objekten 52

Kapitel 3 Anwenderspeicher 53

Auf einen Blick. 53
Struktur des Anwenderspeichers 54
Backup und Wiederherstellung ohne Sicherungs-Cartridge oder Speichererweiterung . . 57
Backup und Wiederherstellung mit einer 32KB-Sicherungs-Cartridge 59
Verwendung der 64 KB-Erweiterungs-Sicherungskassette 61

Kapitel 4 Betriebszustände der Steuerung 63
Auf einen Blick 63
Zyklische Abtastung 64
Periodische Abtastung 66
Prüfen der Zykluszeit 69
Betriebszustände 70
Umgehen mit Spannungsunterbrechungen und Spannungsrückkehr 72
Umgehen mit Warmstart 74
Umgehen mit Kaltstart 76
Initialisierung der Objekte 78

Kapitel 5 Verwaltung von Ereignistasks 79
Auf einen Blick 79
Überblick über Ereignistasks 80
Beschreibung der verschiedenen Ereignisquellen 81
Ereignisverwaltung 82

Teil II Sonderfunktionen 85
Auf einen Blick 85

Kapitel 6 Kommunikation 87
Auf einen Blick 87
Überblick über die verschiedenen Kommunikationstypen 89
Kommunikation zwischen TwidoSoft und der Steuerung 91
Kommunikation zwischen TwidoSoft und einem Modem 97
Kommunikation über die dezentrale Verbindung 109
ASCII-Kommunikation 122
Modbus-Kommunikation 134
Modbus-Standard-Requests 153
Transparent Ready Implementation Class (Twido Serial A05, Ethernet A15) . . 159
Ethernet TCP/IP-Kommunikations-Überblick 160
TCP/IP-Konfigurationsanleitung für die Ethernet-Kommunikation zwischen PC und Steuerung 162
Anschließen Ihrer Steuerung an das Netzwerk 168
IP-Adressierung 169
Zuweisen von IP-Adressen 170
TCP/IP-Setup 174
Registerkarte "IP-Adresskonfiguration" 176
Registerkarte "Markiertes IP" 179
Registerkarte "Timeout" 181
Registerkarte "Dezentrale Geräte" 183
Anzeigen der Ethernet-Konfiguration 185
Konfiguration der Ethernet-Verbindungen 187
Ethernet-Anzeige-LEDs 189
TCP Modbus-Nachrichtenübertragung 190

Kapitel 7 Integrierte Analogfunktionen 193
Auf einen Blick. 193
Analoger Potentiometer. 194
Analogkanal. 196

Kapitel 8 Verwalten von Analogmodulen 197
Auf einen Blick. 197
Übersicht über Analogmodule 198
Adressierung von analogen Ein- und Ausgängen 199
Konfiguration der analogen Ein- und Ausgänge 200
Statusinformationen zum Analogmodul. 207
Beispiel für die Verwendung von Analogmodulen. 208

Kapitel 9 Inbetriebnahme des AS-Interface V2-Busses 211
Auf einen Blick. 211
Beschreibung des AS-Interface V2-Busses 212
Allgemeine Funktionsbeschreibung 213
Prinzipien der softwaretechnischen Inbetriebnahme. 215
Beschreibung des Konfigurationsfensters des AS-Interface-Bus 216
Konfiguration des AS-Interface-Busses 218
Beschreibung des Debug-Fensters. 224
Ändern der Adresse eines Slaves 227
Aktualisierung der Konfiguration des AS-Interface-Busses im Online-Modus . 229
Automatische Adressierung eines AS-Interface V2-Slave 234
Einfügen eines Slave-Geräts in eine bestehende AS-Interface V2-Konfiguration. . 235
Automatischer Ersatz eines fehlerhaften AS-Interface V2-Slave 236
Adressierung der Ein-/Ausgänge, die den am AS-Interface V2-Bus angeschlossenen Slave-Geräten zugeordnet sind.. 237
Programmierung und Diagnose des AS-Interface V2-Busses 238
Betriebsart des AS-Interface V2-Bus-Schnittstellenmoduls 243

Kapitel 10 Installieren und Konfigurieren des CANopen-Feldbusses . . 245
Auf einen Blick. 245
10.1 CANopen-Feldbus - Übersicht 247
Auf einen Blick. 247
CANopen - Grundlagenwissen 248
Über CANopen 249
CANOpen Boot-Up 252
Process Data Object (PDO) -Übertragung 255
Datenzugriff per explizitem Austausch (SDO). 257
"Node Guarding" und "Life Guarding" 258
Verwaltung des internen Busses. 260
10.2 Implementieren des CANopen-Busses 261
Überblick 261
Überblick 262
Hardware-Setup 263

Konfigurationsverfahren 264
Deklaration des CANopen-Masters 266
CANopen-Netzwerk-Slave-Deklaration 267
CANopen-Objektzuweisung 271
Verbinden von CANopen-Objekten 275
Symbolisierung von CANopen-Objekten 277
Adressieren der PDOs des CANopen-Masters 279
Programmierung und Diagnose für den CANopen-Feldbus 280

Kapitel 11 Konfigurieren des TwidoPort-Ethernet-Gateways 287
Auf einen Blick 287
11.1 Normale Konfiguration und Anschluss von TwidoPort 289
Auf einen Blick 289
Normale Konfiguration mit TwidoSoft 290
BootP-Konfiguration 295
11.2 TwidoPort-Konfiguration mittels Telnet 296
Auf einen Blick 296
Einführung in die Telnet-Konfiguration 297
Telnet-Hauptmenü 298
IP/Ethernet-Einstellungen 299
Konfiguration der seriellen Parameter 301
Konfiguration des Gateways 302
Sicherheitskonfiguration 303
Ethernet-Statistiken 304
Serielle Statistiken 305
Speichern der Konfiguration 306
Wiederherstellen der Standardeinstellungen 307
Upgraden der TwidoPort-Firmware 308
Haben Sie Ihr Passwort und/Oder Ihre IP-Konfiguration vergessen? 310
11.3 Kommunikationsfunktionen 311
Auf einen Blick 311
Ethernet-Merkmale 312
Modbus/TCP-Kommunikationsprotokoll 313
Lokal unterstützte Modbus-Funktionscodes 314

Kapitel 12 Betrieb des Bedienterminals 315
Auf einen Blick 315
Bedienterminal 316
Steuerungskennung und Zustandsinformationen 319
Systemobjekte und Systemvariablen 321
Einstellungen des seriellen Ports 327
Echtzeituhr 328
Korrekturfaktor der Echtzeituhr 329

Teil III Beschreibung der Twido Sprachen 331
Auf einen Blick. 331

Kapitel 13 Kontaktplan 333
Auf einen Blick. 333
Einführung in Kontaktplan-Diagramme 334
Programmierprinzipien in Kontaktplänen 336
Blöcke in Kontaktplänen 338
Grafische Elemente des Kontaktplans 342
Besondere KOP-Anweisungen OPEN und SHORT 345
Tipps für die Programmierung. 346
KOP/AWL-Umkehrbarkeit 351
Empfehlungen für die Umkehrbarkeit zwischen KOP und AWL 352
Programmdokumentation 354

Kapitel 14 Anweisungsliste 357
Auf einen Blick. 357
Überblick über AWL-Programme 358
Funktionsweise von AWL-Anweisungen 360
AWL-Anweisungen 361
Verwenden von Klammern 364
Stapelanweisungen (MPS, MRD, MPP) 367

Kapitel 15 Grafcet 369
Auf einen Blick. 369
Beschreibung der Grafcet-Anweisungen. 370
Beschreibung des Grafcet-Programmaufbaus 374
Zu Grafcet-Schritten gehörige Aktionen 377

Teil IV Beschreibung von Anweisungen und Funktionen . . . 379
Auf einen Blick. 379

Kapitel 16 Basis-Anweisungen 381
Auf einen Blick. 381
16.1 Boolesche Verarbeitung 383
Auf einen Blick. 383
Boolesche Anweisungen. 384
Format für die Beschreibung von Booleschen Anweisungen 387
Ladeanweisungen (LD, LDN, LDR, LDF) 389
Zuweisungsbefehle (ST, STN, R, S) 391
Logische UND-Anweisungen (AND, ANDN, ANDR, ANDF) 393
Logische ODER-Anweisungen (OR, ORN, ORR, ORF) 395
exklusives OR, Anweisungen (XOR, XORN, XORR, XORF) 397
Anweisung NOT (N) 399

16.2 Basis-Funktionsblöcke . . . 401
Auf einen Blick . . . 401
Basis-Funktionsblöcke . . . 402
Prinzipien für die Programmierung von Standard-Funktionsblöcken . . . 404
Timer-Funktionsblock (%TMi) . . . 406
Timer-Typ TOF . . . 408
Timer-Typ TON . . . 409
Timer-Typ TP . . . 410
Programmieren und Konfigurieren von Timern . . . 411
Auf-/Abwärtszähler-Funktionsblock (%Ci) . . . 414
Programmierung und Konfiguration der Zähler . . . 418
Schiebebitregister-Funktionsblock (%SBRi) . . . 420
Schrittzähler-Funktionsblock (%SCi) . . . 422
16.3 Numerische Verarbeitung . . . 426
Auf einen Blick . . . 426
Einführung in numerische Anweisungen . . . 427
Zuweisungsanweisungen . . . 428
Vergleichsanweisungen . . . 433
Arithmetische Anweisungen für Ganzzahlen . . . 435
Logische Anweisungen . . . 439
Schiebeanweisungen . . . 440
Konvertierungsanweisungen . . . 442
Anweisungen zur Konvertierung zwischen einfachen Wörtern und Doppelwörtern . . . 444
16.4 Programmanweisungen . . . 445
Auf einen Blick . . . 445
END-Anweisungen . . . 446
Anweisung NOP . . . 448
Sprunganweisungen . . . 449
Unterprogramm-Anweisungen . . . 450

Kapitel 17 Erweiterte Anweisungen . . . 453
Auf einen Blick . . . 453
17.1 Erweiterte Funktionsblöcke . . . 455
Auf einen Blick . . . 455
Zu erweiterten Funktionsblöcken gehörige Bit- und Wortobjekte . . . 456
Prinzipien für die Programmierung von erweiterten Funktionsblöcken . . . 458
LIFO/FIFO-Register-Funktionsblock (%Ri) . . . 461
LIFO-Funktionsweise . . . 462
FIFO-Funktionsweise . . . 463
Programmierung und Konfiguration der Register . . . 464
Impulsbreitenmodulations-Funktionsblock (%PWM) . . . 466
Impulsgenerator-Funktionsblock (%PLS) . . . 469
Trommelsteuerungs-Funktionsblock (%DR) . . . 472
Funktionsweise des Trommelsteuerungs-Funktionsblocks %DRi . . . 474
Programmieren und Konfigurieren von Trommelsteuerungen . . . 476

Schneller Zähler-Funktionsblock (%FC) 478
Sehr schneller Zähler-Funktionsblock (%VFC) 481
Senden/Empfangen von Meldungen - die Austauschanweisung (EXCH). 494
Austauschsteuerungs-Funktionsblock (%MSGx) 495
17.2 Uhrfunktionen 499
Auf einen Blick. 499
Uhrfunktionen 500
Abarbeitungsblöcke. 501
Uhrzeit-/Datumsstempel 504
Einstellen von Datum und Uhrzeit. 506
17.3 Twido PID-Kurzanleitung. 509
Auf einen Blick. 509
Zweck des Dokuments 510
Schritt 1 - Konfigurierung der analogen Kanäle für die Regelung 512
Schritt 2 - Vorbedingungen für die PID-Konfigurierung 514
Schritt 3 - Konfiguration des PID. 516
Schritt 4 - Initialisierung der Inbetriebnahme der Regelung 523
Schritt 5 - Inbetriebnahme der Regelung AT + PID. 527
Schritt 6 - Debugging der Einstellungen 531
17.4 PID-Funktion 534
Auf einen Blick. 534
Allgemeine Beschreibung 535
Prinzip der Regelkreise 536
Methode zur Entwicklung einer Regelungsapplikation 537
Kompatibilität und Leistungsdaten 538
Ausführliche Beschreibung der Merkmale der PID-Funktion 539
Verfahren zum Zugreifen auf die Konfiguration des PID. 542
Registerkarte "Allgemein" des PID 544
Registerkarte "Eingang" des PID 547
Registerkarte "PID" des PID 549
Registerkarte AT des PID 551
Registerkarte "Ausgang" des PID 556
Verfahren zum Zugreifen auf des Debugging des PID 559
Registerkarte "Animation" des PID 561
Registerkarte "Trace" des PID 563
PID-Zustände und Fehlercodes 565
PID-Einstellung mittels Auto-Tuning (AT) 569
PID-Parameter-Anpassungsverfahren 577
Rolle und Einfluss der Paramater eines PID. 580
Anhang 1: Grundlegendes zur PID-Theorie 584
Anhang 2: Modell erster Ordnung mit Zeitverzögerung 586
17.5 Anweisungen zu Gleitkommas 588
Auf einen Blick. 588
Arithmetische Anweisungen für Gleitkommawerte 589
Trigonometrische Anweisungen 593

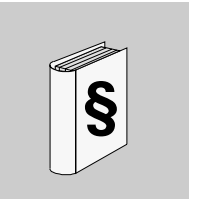
Konvertierungsanweisungen 595
Anweisungen zur Konvertierung von Ganzzahlen <-> Gleitpunktzahlen. 597
17.6 Anweisungen für Objekttabellen: 600
Auf einen Blick 600
Summierungsfunktion in Tabellen 601
Tabellenvergleichsfunktion 602
Suchfunktionen an Tabellen 604
Suchfunktionen nach Minimal- und Maximalwerten in Tabellen 606
Anzahl der Vorkommen eines Werts in einer Tabelle 607
Funktion Zirkularverschiebung in einer Tabelle 608
Sortierfunktion in Tabellen 610
Interpolationsfunktion in Gleitkommazahltabellen 612
Funktion zur Bildung des Mittelwertes der Werte einer Gleitkommawerttabelle . . 617

Kapitel 18 Systembits und Systemworte. 619
Auf einen Blick 619
Systembits (%S) 620
Systemwörter (%SW). 633

Glossar 647

Index 661

Sicherheitshinweise



Wichtige Informationen

HINWEIS

Lesen Sie diese Anleitung sorgfältig durch, und machen Sie sich mit den Geräten vertraut, bevor Sie mit der Installation, dem Betrieb oder der Wartung beginnen. Die folgenden Hinweise können an verschiedenen Stellen in dieser Dokumentation enthalten oder auf dem Gerät zu lesen sein. Die Hinweise warnen vor möglichen Gefahren oder machen auf Informationen aufmerksam, die Vorgänge erläutern bzw. vereinfachen.

Wenn dieses Symbol neben einem Sicherheitsschild für Gefahr oder Warnung erscheint, bedeutet das, dass eine elektrische Gefährdung besteht, die zu Körperverletzung führen kann, wenn die Anweisungen nicht befolgt werden.

Dies ist ein allgemeines Warnsymbol. Es macht Sie auf mögliche Verletzungsgefahren aufmerksam. Befolgen Sie alle Sicherheitshinweise bei diesem Symbol, um mögliche Gefahren für Gesundheit oder Leben zu vermeiden.

⚠ GEFAHR

GEFAHR macht auf eine unmittelbar gefährliche Situation aufmerksam, die bei Nichtbeachtung **unweigerlich** einen schweren oder tödlichen Unfall oder Beschädigungen an Geräten zur Folge hat.

⚠ WARNUNG

WARNUNG macht auf eine möglicherweise gefährliche Situation aufmerksam, die bei Nichtbeachtung **unter Umständen** einen schweren oder tödlichen Unfall oder Beschädigungen an Geräten zur Folge hat.

⚠ VORSICHT

VORSICHT macht auf eine möglicherweise gefährliche Situation aufmerksam, die bei Nichtbeachtung **unter Umständen** einen Unfall oder Beschädigungen an Geräten zur Folge hat.

BITTE BEACHTEN

Elektrische Geräte dürfen nur von Fachpersonal gewartet und instandgesetzt werden. Schneider Electric haftet nicht für Schäden, die aufgrund der Verwendung dieses Materials entstehen. Dieses Dokument ist nicht als Betriebsanleitung für nicht geschultes Personal vorgesehen. Einbau- und Installationsanleitungen finden Sie im Twido Hardware-Referenzhandbuch für TWD USE 10AE.

(c) 2002-2005 Schneider Electric Alle Rechte vorbehalten

Zusätzliche Sicherheitsinformationen

Die für den Betrieb oder die Anwendung dieses Produkts verantwortlichen Personen müssen sicherstellen, dass die erforderlichen Vorkehrungen getroffen werden, um die geltenden Gesetze, Leistungs- und Sicherheitsanforderungen, Vorschriften, Bestimmungen und Normen zu erfüllen.

Allgemeine Warnhinweise

⚠ GEFAHR

GEFAHR EINES ELEKTRISCHEN SCHLAGS, VERBRENNUNGEN ODER EXPLOSIONEN

Schalten Sie alle Stromversorgung aus, bevor Sie mit Arbeiten zur Installation, Demontage, Verkabelung, Wartung oder Überprüfung beginnen.

Die Nichtbeachtung dieser Anweisung wird den Tod, schwere Körperverletzung oder Materialschäden zur Folge haben!

⚠ WARNUNG

EXPLOSIONSGEFAHR

- Durch Ersetzen von Komponenten kann die Eignung für Class I, Div 2 beeinträchtigt werden.
- Entfernen Sie Anschlüsse nur dann von Geräten, wenn Sie zuvor die Stromversorgung abgeschaltet haben oder wenn bekannt ist, dass im betreffenden Bereich keine Gefahr besteht.

Die Nichtbeachtung dieser Anweisung kann den Tod, körperverletzung Oder materialschäden Zur folge Haben!

⚠ WARNUNG

EXPLOSIONSGEFAHR

- Schalten Sie die Stromversorgung aus, bevor Sie mit Arbeiten zur Installation, Demontage, Verkabelung oder Wartung beginnen.
- Dieses Produkt ist nicht für den Einsatz in sicherheitskritischen Maschinenfunktionen gedacht. Bei Gefahr für Personal oder Geräte sind geeignete Sicherheitssperren zu verwenden.
- Demontieren, reparieren oder verändern Sie die Module nicht.
- Diese Steuerung ist für den Einsatz in einem Gehäuse bestimmt.
- Installieren Sie die Module unter normalen Betriebsverhältnissen.
- Verwenden Sie die Sensor-Stromversorgung nur für an das Modul angeschlossene Sensoren.
- Verwenden Sie für Netzleitungen und Ausgangsschaltkreise eine Sicherung des Typs T gemäß IEC60127. Die Sicherung muss den Leiterspannungs- und Stromvorschriften entsprechen. Empfohlene Sicherung: Littelfuse®, Serie 218, 5 x 20 mm, träge.

Die Nichtbeachtung dieser Anweisung kann den Tod, körperverletzung Oder materialschäden Zur folge Haben!

Sichere Batterieentsorgung

Die kompakten Grundgeräte TWDLCA•40DRF verwenden eine optionale externe Lithium-Batterie für eine längerfristige Datensicherung. (Hinweis: Die Lithium-Batterie ist nicht im Lieferumfang der kompakten Grundgeräte enthalten; Sie müssen sie separat erwerben.)

⚠ WARNUNG

EXPLOSIONS- UND VERGIFTUNGSGEFAHR

- Verbrennen Sie keine Lithium-Batterien, da sie explodieren und giftige Substanzen freisetzen können.
- Berühren Sie keine beschädigten oder ausgelaufenen Lithium-Batterien.
- Verbrauchte Batterien müssen ordnungsgemäß entsorgt werden. Falsch entsorgte Batterien können Gesundheits- und Umweltschäden verursachen.
- In bestimmten Ländern ist die Entsorgung von Lithium-Batterien als Haushalts- oder Industrieabfall untersagt. Für die vorschriftsmäßige Entsorgung von Batterien sind in jedem Fall Sie selbst verantwortlich..

Die Nichtbeachtung dieser Anweisung kann den Tod, körperverletzung Oder materialschäden Zur folge Haben!

Warnung vor Polaritätsumkehr

Eine umgekehrte Polarität am Transistorausgang ist nicht zulässig.
Die Transistorausgänge der kompakten Grundgeräte TWDLCA•40DRF können bei einer Polaritätsumkehr beschädigt werden.

⚠ ACHTUNG

GEFAHR EINER BESCHÄDIGUNG BEI FALSCHER POLARITÄT AN DEN TRANSISTORAUSGÄNGE

- Vergewissern Sie sich, dass die Anschlüsse entsprechend den an den Transistorausgängen angegebenen Polaritätsmarkierungen vorgenommen werden.
- Eine falsche Polarität kann zu einer irreversiblen Beschädigung der Ausgangsschaltkreise führen.

Die Nichtbeachtung dieser Anweisung kann Körperverletzung oder Materialschäden zur Folge haben!

Über dieses Buch

Auf einen Blick

Ziel dieses Dokuments

Das Referenzhandbuch der Software für speicherprogrammierbare Twido-Steuerungen besteht aus folgenden Abschnitten:

- Beschreibung der Twido Programmiersoftware und Einführung in die Grundlagen zur Programmierung von Twido Steuerungen.
- Beschreibung der Kommunikationsarten, der Verwaltung der analogen E/A, der Installation des AS-Interface-Bus-Schnittstellenmoduls und des CANopen-Feldbus-Mastermoduls und anderer Sonderfunktionen.
- Beschreibung der Software-Sprachen für die Erstellung von Twido Programmen.
- Beschreibung der Anweisungen und Funktionen von Twido Steuerungen.

Gültigkeitsbereich

Die Informationen in diesem Handbuch gelten **nur** für speicherprogrammierbare Twido Steuerungen.

Produktbezogene Warnhinweise

Schneider Electric übernimmt keinerlei Verantwortung für Fehler, die in diesem Dokument vorhanden sein können. Kein Teil dieses Dokuments darf ohne die vorherige schriftliche Genehmigung durch den Herausgeber Schneider Electric in irgendeiner Weise oder mit irgendeinem Verfahren elektronischer oder mechanischer Art vervielfältigt werden.

Benutzerkommentar

Ihre Anmerkungen und Hinweise sind uns jederzeit willkommen. Senden Sie sie einfach an unsere E-mail-Adresse: TECHCOMM@modicon.com

Beschreibung der Twido Software

I

Auf einen Blick

Inhalt dieses Abschnitts

Dieser Teil bietet eine Einführung in die Software-Sprachen und die grundlegenden Informationen zum Erstellen von Steuerungsprogrammen für speicherprogrammierbare Twido Steuerungen.

Inhalt dieses Teils

Dieser Teil enthält die folgenden Kapitel:

Kapitel	Kapitelname	Seite
1	Einführung in die Twido Software	19
2	Twido Sprachobjekte	25
3	Anwenderspeicher	53
4	Betriebszustände der Steuerung	63
5	Verwaltung von Ereignistasks	79

Einführung in die Twido Software

1

Auf einen Blick

Inhalt dieses Kapitels

Dieses Kapitel bietet eine kurze Einführung in TwidoSoft, die Programmier- und Konfigurations-Software für Twido Steuerungen und in die Programmiersprachen AWL, KOP und Grafcet, mit denen Sie Steuerungsprogramme erstellen.

Inhalt dieses Kapitels

Dieses Kapitel enthält die folgenden Themen:

Thema	Seite
Einführung in TwidoSoft	20
Einführung in Twido Sprachen	21

Einführung in TwidoSoft

Einleitung

TwidoSoft ist eine graphische Entwicklungsumgebung zum Erstellen, Konfigurieren und Pflegen von Applikationen für programmierbare Twido-Steuerungen. In TwidoSoft können Sie mit verschiedenen Sprachtypen (Siehe *Twido Sprachen, S. 21*) Programme erstellen und anschließend die Applikation zur Ausführung in eine Steuerung übertragen.

TwidoSoft

TwidoSoft ist ein auf Windows basierendes 32-Bit-Programm für einen PC mit einem der Betriebssysteme Microsoft Windows 98 (Second Edition), Microsoft Windows 2000 Professional und Microsoft Windows XP.
Wesentliche Software-Merkmale von TwidoSoft:

- Standard Windows-Bedienoberfläche
- Programmieren und Konfigurieren von Twido-Steuerungen
- Steuerungskommunikation und -steuerung

Hinweis: Die Verbindung Steuerung-PC nutzt das TCP/IP-Protokoll. Dieses Protokoll muss auf dem PC installiert sein.

Voraussetzungen

Die für die Nutzung von TwidoSoft erforderliche minimale Konfiguration lautet:

- Pentium 300 MHz
- 128 MB RAM
- 40 MB freier Festplatten-Platz

Einführung in Twido Sprachen

Einleitung

Eine speicherprogrammierbare Steuerung liest Eingänge, schreibt in Ausgänge und löst die Logik entsprechend einem Steuerungsprogramm. Wenn Sie ein Steuerungsprogramm für eine Twido Steuerung erstellen möchten, schreiben Sie eine Reihe von Anweisungen in einer der Twido Programmiersprachen.

Twido Sprachen

Mit den folgenden Sprachen können Sie Twido Steuerungsprogramme erstellen:

- Anweisungsliste (AWL):
  Ein AWL-Programm besteht aus einer Reihe logischer Ausdrücke, die in einer Folge von Booleschen Anweisungen geschrieben sind.
- Kontaktplan (KOP):
  Ein Kontaktplan zeigt logische Ausdrücke auf graphische Art und Weise an.
- Grafcet:
  Die Programmiersprache Grafcet besteht aus einer Folge von Schritten und von Transitionen. Twido unterstützt Grafcet-AWL-Anweisungen, aber keine graphischen Grafcet-Anweisungen.

Sie können mit diesen Programmiersprachen Twido Steuerungsprogramme auf einem PC erstellen und bearbeiten.
Durch die AWL/KOP-Umkehrbarkeit können Sie KOP-Programme in AWL umwandeln und AWL-Programme in KOP umwandeln.

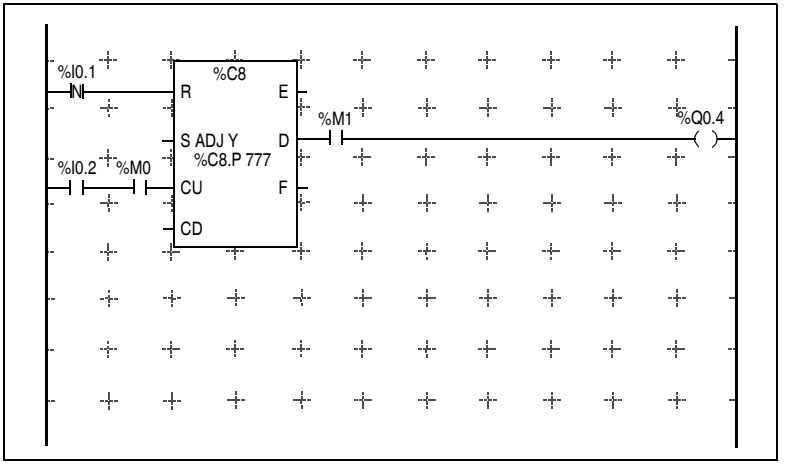
Anweisungsliste (AWL)

Ein in der Anweisungsliste (AWL) geschriebenes Programm besteht aus einer Folge von Anweisungen, die von der Steuerung der Reihe nach ausgeführt werden. Dieses Beispiel zeigt die Programmierung in AWL.

```
0    BLK     %C8
1    LDF     %I0.1
2    R
3    LD      %I0.2
4    AND     %M0
5    CU
6    OUT_BLK
7    LD      D
8    AND     %M1
9    ST      %Q0.4
10   END_BLK
```

Kontaktplan (KOP)

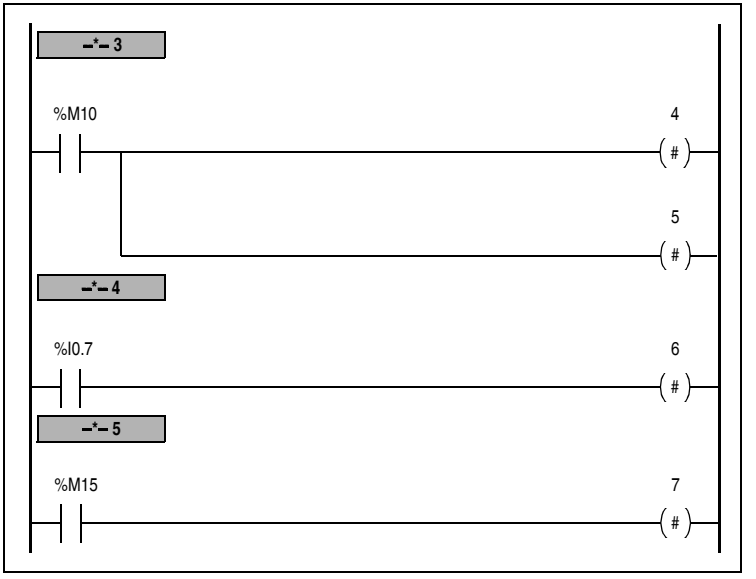
Kontaktplan-Diagramme ähneln Schaltplänen, in denen Relaisschaltkreise dargestellt werden. In diesen Diagrammen stellen graphische Elemente wie Spulen, Kontakte und Blöcke die Programmanweisungen dar. Im Folgenden sehen Sie ein Beispiel für einen Kontaktplan.



Grafcet

Beim analytischen Verfahren Grafcet werden Ablaufsteuerungssysteme in eine Reihe von Schritten unterteilt, denen Aktionen, Transitionen und Bedingungen zugewiesen sind. Die folgende Abbildung zeigt Beispiele für Grafcet-Anweisungen in AWL- und KOP-Programmen.

0	-*-	3
1	LD	%M10
2	#	4
3	#	5
4	-*-	4
5	LD	%I0.7
6	#	6
7	-*-	5
8	LD	%M15
9	#	7
10	...	



Twido Sprachobjekte

2

Auf einen Blick

Inhalt dieses Kapitels

In diesem Kapitel erfahren Sie Einzelheiten zu den Sprachobjekten für die Programmierung von Twido Steuerungen.

Inhalt dieses Kapitels

Dieses Kapitel enthält die folgenden Themen:

Thema	Seite
Sprachobjektvalidierung	26
Bitobjekte	27
Wortobjekte	30
Gleitkomma- und Doppelwortobjekte	33
Adressierung von Bitobjekten	37
Adressierung von Wortobjekten	38
Adressierung von Gleitkommaobjekten	39
Adressierung von Doppelwortobjekten	40
Kennzeichnung der Eingänge/Ausgänge	41
Netzwerk-Adressierung	43
Funktionsblockobjekte	44
Strukturierte Objekte	46
Indexierte Objekte	50
Symbolisieren von Objekten	52

Sprachobjektvalidierung

Einführung

Wort- und Bitobjekte sind gültig, wenn Ihnen Speicherplatz in der Steuerung zugeordnet wurde. Hierzu müssen Sie in der Applikation verwendet werden, bevor sie in die Steuerung geladen werden.

Beispiel

Der Bereich gültiger Objekte liegt zwischen Null und der Maximalreferenz für den jeweiligen Objekttyp. Ist z.B. die Maximalreferenz in Ihrer Applikation für Speicherworte %MW9, dann sind %MW0 bis %MW9 zugeordneter Speicher. %MW10 ist in diesem Beispiel nicht gültig und es kann weder intern noch extern darauf zugegriffen werden.

Bitobjekte

Einführung

Bitobjekte sind Softwarevariablen vom Typ Bit, die wie Operanden verwendet und durch boolesche Anweisungen getestet werden können. In der nachstehenden Tabelle Bitobjekte aufgelistet:

- E/A-Bits
- Interne Bits (Speicherbits)
- Systembits
- Schrittbits
- Aus Worten extrahierte Bits

Liste der Operandenbits

In der folgenden Tabelle sind alle wesentlichen Bitobjekte aufgeführt und beschrieben, die als Operanden in Booleschen Anweisungen verwendet werden.

Typ	Beschreibung	Adresse oder Wert	Maximale Anzahl	Schreibzugriff (1)
Sofortwerte	0 oder 1 (False oder True)	0 oder 1	-	-
Eingänge Ausgänge	Diese Bits sind die "logischen Abbilder" der elektrischen Zustände der E/A. Sie werden im Datenspeicher abgelegt und während des Zyklus durch die Programmlogik aktualisiert.	%Ix.y.z (2) %Qx.y.z (2)	Hinweis (4)	Nein Ja
AS-Interface Eingänge Ausgänge	Diese Bits sind die "logischen Abbilder" der elektrischen Zustände der E/A. Sie werden im Datenspeicher abgelegt und während des Zyklus durch die Programmlogik aktualisiert.	 %IAx.y.z %QAx.y.z	Hinweis (5)	 Nein Ja
Intern (Speicher)	Interne Bits sind interne Speicherbereiche, in denen Zwischenberechnungswerte abgelegt werden, wenn ein Programm in Bearbeitung ist. **Hinweis:** Freie E/A-Bits können nicht als interne Bits verwendet werden.	%Mi	128 TWDLC•A10DRF, TWDLC•A16DRF 256 alle anderen SPS	Ja
Systemworte	Die Systembits %S0 bis %S127 überwachen den korrekten Betrieb der Steuerung und den korrekten Ablauf des Anwendungsprogramms.	%Si	128	Entsprechend i
Funktionsblöcke	Die Funktionsblockbits entsprechen den Ausgängen der Funktionsblöcke. Diese Ausgänge sind entweder direkt angeschlossen oder werden als Objekt genutzt.	%TMi.Q, %Ci.P usw.	Hinweis (4)	Nein (3)
Reversible Funktionsblöcke	Funktionsblöcke, die mit den reversiblen Programmanweisungen BLK, OUT_BLK und END_BLK programmiert wurden.	E, D, F, Q, TH0, TH1	Hinweis (4)	Nein
Wortextrakte	Für einige Worte wird eins der 16 Bits als Operandenbit extrahiert.	Variable	Variable	Variable

Typ	Beschreibung	Adresse oder Wert	Maximale Anzahl	Schreibzugriff (1)
Grafcet-Schritte	Die Bits %X1 bis %Xi werden den Grafcet-Schritten zugewiesen. Das Schrittbit Xi befindet sich in Zustand 1, wenn der entsprechende Schritt aktiviert ist, und in Zustand 0, wenn der Schritt deaktiviert ist.	%X21	62 TWDLC•A10DRF, TWDLC•A16 DRF 96 TWDLC•A24DRF, TWDLCA•40DRF und modulare Steuerungen	Ja

Legende:

1. Geschrieben vom Programm oder mittels Animationstabellen-Editor.
2. Siehe E/A-Adressierung.
3. Ausnahme sind %SBRi.j und %SCi.j, diese Bits können gelesen und geschrieben werden.
4. Die Anzahl richtet sich nach der Art der Steuerung.
5. wobei x = Adresse des Erweiterungsmoduls (0 - 7); y = AS-Interface-Adresse (0A - 31B); z = Kanalnummer (0 - 3) (siehe *Adressierung der Ein-/Ausgänge, die den am AS-Interface V2-Bus angeschlossenen Slave-Geräten zugeordnet sind., S. 237*).

Wortobjekte

Einführung

Wortobjekte, die in Form von 16-Bit-Wörtern adressiert werden, werden im Datenspeicher abgelegt und können ganzzahlige Werte zwischen -32768 und 32767 enthalten (mit Ausnahme des schnellen Zähler-Funktionsblocks, der den Bereich von 0 bis 65535 hat).
Beispiele für Wortobjekte:

- Sofortwerte
- Interne Wörter (%MWi) (Speicherwörter)
- Konstante Wörter (%KWi)
- E/A-Austauschwörter (%IWi, %Qwi%)
- Analoge AS-Interface-E/A-Wörter (IWAi, %QWAi)
- Systemwörter (%SWi)
- Funktionsblöcke (Konfigurations- und/oder Laufzeitdaten)

Wortformate

Der Inhalt der Wörter oder Werte wird im Anwenderspeicher im 16-Bit-Binärcode gespeichert (Zweierkomplement), wobei folgende Konvention gilt:

	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0	Bitposition
	0	1	0	1	0	0	1	0	0	1	0	0	1	1	0	1	Bitzustand
|+	16384	8192	4096	2048	1024	512	256	128	64	32	16	8	4	2	1		Bitwert

Bei Binärnotierung mit Vorzeichen ist Bit 15 nach der Konvention dem Vorzeichen des codierten Werts zugeordnet:

- Bit 15 ist gleich 0: der Inhalt des Worts ist ein positiver Wert.
- Bit 15 ist gleich 1: der Inhalt des Worts ist ein negativer Wert (negative Werte werden in der Logik eines Zweierkomplements ausgedrückt).

Wörter und direkte Werte können in den folgenden Formaten eingegeben oder abgerufen werden:

- Dezimal
  min.: -32768, max.: 32767 (zum Beispiel 1579)
- Hexadezimal
  min.: 16#0000, max.: 16#FFFF (z.B. 16#A536)
  Alternative Syntax: #A536

Beschreibungen von Wortobjekten Die folgende Tabelle beschreibt die Wortobjekte.

Wörter	Beschreibung	Adresse oder Wert	Maximale Anzahl	Schreibzugriff (1)
Sofortwerte	Dies sind ganzzahlige Werte im gleichen Format wie die 16-Bit-Worte, wodurch diesen Worten Werte zugewiesen werden können.		-	Nein
	Basis 10	-32768 bis 32767		
	Basis 16	16#0000 bis 16#FFFF		
Intern (Speicher)	Dienen als "Arbeitswörter", die während des Betriebs Werte im Datenspeicher speichern. Die Wörter %MW0 bis %MW255 werden direkt vom Programm gelesen oder geschrieben.	%MWi	3000	Ja
Konstanten	Speichern von Konstanten oder alphanumerischen Meldungen. Ihr Inhalt kann nur mit TwidoSoft während der Konfiguration geschrieben oder geändert werden. Die konstanten Wörter %KW0 bis %KW63 sind vom Programm schreibgeschützt.	%KWi	256	Ja, nur mit TwidoSoft.
Systemworte	Diese 16-Bit-Wörter haben verschiedene Funktionen: ● Sie bieten Zugriff auf Daten, die direkt aus der Steuerung kommen, indem die Wörter %SWi gelesen werden. ● Sie führen Befehle in der Applikation aus (z.B. Einstellen von Abarbeitungsblöcken).	%SWi	128	Entsprechend i
Funktionsblöcke	Diese Wörter entsprechen den aktuellen Parametern oder Werten von Funktionsblöcken.	%TM2.P, %Ci.P usw.		Ja

Wörter	Beschreibung	Adresse oder Wert	Maximale Anzahl	Schreibzugriff (1)
Netzwerkaustauschwörter	Sie sind Steuerungen zugewiesen, die als dezentrale Verbindungen angeschlossen sind. Diese Wörter werden für die Kommunikation zwischen Steuerungen verwendet:			
	Netzwerkeingang	%INWi.j	4 je dezentrale Verbindung	Nein
	Netzwerkausgang	%QNWi.j	4 je dezentrale Verbindung	Ja
Analoge E/A-Wörter	Zugewiesen zu Analogein- und -ausgängen von AS-Interface-Slave-Modulen.			
	Analogeingänge	%IWAx.y.z	Hinweis (3)	Nein
	Analogausgänge	%QWAx.y.z	Hinweis (3)	Ja
Extrahierte Bits	Es ist möglich, eines von 16 Bits aus den folgenden Wörtern zu extrahieren:			
	Intern	%MWi:Xk	1500	Ja
	Systemworte	%SWi:Xk	128	Abhängig von i
	Konstanten	%KWi:Xk	64	Nein
	Eingang	%IWi.j:Xk	Hinweis (2)	Nein
	Ausgang	%QWi.j:Xk	Hinweis (2)	Ja
	AS-Interface-Slaveeingang	%IWAx.y.z:Xk	Hinweis (2)	Nein
	AS-Interface-Slaveausgang	%QWAx.y.z:Xk	Hinweis (2)	Ja
	Netzwerkeingang	%INWi.j:Xk	Hinweis (2)	Nein
	Netzwerkausgang	%QNWi.j:Xk	Hinweis (2)	Ja

Hinweis:
1. Geschrieben vom Programm oder mittels Animationstabellen-Editor.
2. Die Anzahl richtet sich nach der Konfiguration.
3. wobei x = Adresse des Erweiterungsmoduls (0 - 7); y = AS-Interface-Adresse (0A - 31B); z = Kanalnummer (0 - 3). (siehe *Adressierung der Ein-/Ausgänge, die den am AS-Interface V2-Bus angeschlossenen Slave-Geräten zugeordnet sind., S. 237*).

Gleitkomma- und Doppelwortobjekte

Einführung

Die Software TwidoSoft ermöglicht die Durchführung von Operationen an Gleitkomma- und Ganzzahl-Doppelwortobjekten.
Ein Gleitkommawert ist ein mathematisches Argument, das in seinem Ausdruck über eine Dezimale verfügt (Beispiele: 3.4E+38, 2.3 oder 1.0).
Ein Ganzzahl-Doppelwort besteht aus 4 Bytes, die im Datenspeicher abgelegt sind und die einen Ganzzahlwert zwischen -2147483648 und +2147483647 enthalten können.

Format und Wert des Gleitkomma-wertes

Das verwendete Gleitkommaformat ist das Format der Norm IEEE Std 734-1985 (entspricht IEC 559). Die Wortlänge ist 32 Bits, was der Anzahl Gleitkommata einfacher Genauigkeit entspricht.
Tabelle zur Beschreibung des Formats eines Gleitkommawerts:

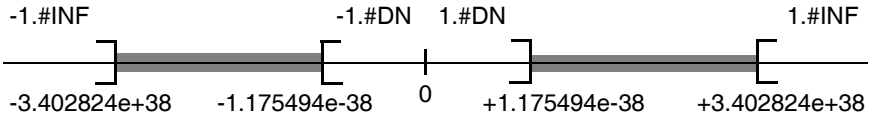
Bit 31	Bits {30 - 23}	Bits {22 - 0}
S	Exponent	Mantisse

Der Wert des oben aufgeführten Formats wird durch folgende Gleichung bestimmt:

32-Bit-Gleitkommawert = $(-1)^S * 2^{(\text{Exposant} - 127)} * 1.\text{Mantisse}$

Die Gleitkommawerte können mit oder ohne Exponent dargestellt werden, sie müssen immer einen Punkt enthalten (Gleitkomma).
Die Gleitkommawerte befinden sich zwischen -3.402824e+38 und -1.175494e-38 ; 1.175494e-38 und 3.402824e+38 (Im Schema abgeblendete Werte). Sie enthalten auch den Wert 0, geschrieben 0.0

.



Wenn ein Rechenergebnis :

- unter -3.402824e+38 ist, wird das Symbol -1.#INF (für unendlich) angezeigt
- über +3.402824e+38 ist, wird das Symbol 1.#INF (für unendlich) angezeigt
- zwischen -1.175494e-38 und 1.175494e-38 liegt, wird das Ergebnis auf 0.0 gerundet. Ein Wert zwischen diesen Grenzen kann nicht als Gleitkomma eingegeben werden.
- unendlich (zum Beispiel die Quadratwurzel einer negativen Zahl) ist, wird das Symbol 1.#NAN oder -1.#NAN angezeigt.

Die Darstellungsgenauigkeit beträgt 2-24. Um Gleitkommazahlen anzuzeigen, ist es nicht erforderlich, mehr als 6 Ziffern nach dem Dezimalkomma anzuzeigen.

Hinweis:
- Der Wert "1285" wird als ganzer Wert interpretiert, um als Gleitkommawert berücksichtigt zu werden, muss er geschrieben werden: "1285.0"

Grenzbereich der auf Gleitkommaobjekte angewandten arithmetischen Funktionen

Die folgende Tabelle beschreibt den Grenzbereich der auf Gleitkommaobjekte angewandten arithmetischen Funktionen

Arithmetische Funktion		**Grenzbereich und ungültige Operationen**	
Typ	**Syntax**	**#QNAN (ungültig)**	**#INF (unendlich)**
Quadratwurzel eines Operanden	SQRT(x)	x < 0	x > 1,7E38
Potenz einer Ganzzahl durch eine Realzahl EXPT(%MF,%MW)	EXPT(y, x) (wobei: x^y = %MW^%MF)	x < 0	y.ln(x) > 88
10er-Logarithmus	LOG(x)	x <= 0	x > 2,4E38
Natürlicher Logarithmus	LN(x)	x <= 0	x > 1,65E38
Natürliche Exponentialgröße	EXP(x)	x < 0	x > 88.0

Hardware-Kompatibilität

Die Operationen an Gleitkommawerten und Doppelwörtern werden nicht von allen Twido-Steuerungen unterstützt.
Die folgende Tabelle beschreibt die Hardware-Kompatibilität:

Twido-Steuerung	**Unterstützte Doppelwörter**	**Unterstützte Gleitkommawerte**
TWDLMDA40DUK	Ja	Ja
TWDLMDA40DTK	Ja	Ja
TWDLMDA20DUK	Ja	Nein
TWDLMDA20DTK	Ja	Nein
TWDLMDA20DRT	Ja	Ja
TWDLCA•40DRF	Ja	Ja
TWDLC•A24DRF	Ja	Nein
TWDLC•A16DRF	Ja	Nein
TWDLC•A10DRF	Nein	Nein

Gültigkeits-prüfung

Das Systembit %S18 wird auf 1 gesetzt, wenn das Ergebnis nicht in den gültigen Grenzen liegt.
Die Statuswortbits %SW17 zeigen die Ursache eines Fehlers bei einer Gleitkomma-operation an:
Unterschiedliche Bits des Worts %SW17:

%SW17:X0	ungültige Operation, das Ergebnis ist keine Zahl (1.#NAN oder -1.#NAN)
%SW17:X1	reserviert
%SW17:X2	Division durch 0, das Ergebnis ist unendlich (-1.#INF oder 1.#INF)
%SW17:X3	absoluter Wert des Ergebnisses höher als +3.402824e+38, das Ergebnis ist unendlich (-1.#INF oder 1.#INF)
%SW17:X4 bis X15	reserviert

Dieses Wort wird vom System auf 0 zurückgesetzt nach Kaltstart und vom Programm auf 0 zurückgesetzt für eine Wiederverwendung.

Beschreibung der Gleitkomma- und Doppelwort-objekte

Die folgende Tabelle beschreibt die Gleitkomma- und Doppelwortobjekte.

Objekttyp	**Beschreibung**	**Adresse**	**Maximale Anzahl**	**Schreibzugriff**	**indexierte Form**
Sofortwerte	Ganzzahlen oder Dezimalzahlen im gleichen Format wie die 32-Bit-Objekte.	-	[-]	Nein	-
interner Gleitkomma	Dienen als "Arbeitsobjekte", die während des Betriebs Werte im Datenspeicher speichern.	%MFi	1500	Ja	%MFi[index]
Internes Doppelwort		%MDi	1500	Ja	%MDi[index]
Gleitkommakonstante	speichern die Konstanten	%KFi	128	Ja, nur mit TwidoSoft	%KFi[index]
Doppelwortkonstante		%KDi	128	Ja, nur mit TwidoSoft	%KDi[index]

Möglichkeit der Überschneidung von Objekten

Die Wörter einfacher, doppelter und gleitender Länge werden im Innern des Datenraumes in der selben Speicherzone eingeordnet. So entsprechen das Gleitkommawort %MFi und das Doppelwort %MDi den Wörtern einfacher Länge %MWi und %MWi+1 (das Wort %MWi schließt die niederwertigen Bytes und das Wort %MWi+1 die höherwertigen Bytes des Wortes %MFi ein).
Die folgende Tabelle zeigt die Überschneidung der internen Gleitkommawerte und Doppelwörter.

Gleitkommawerte und Doppelwörter	ungerade Adresse	Interne Wörter
%MF0 / %MD0		%MW0
	%MF1 / %MD1	%MW1
%MF2 / %MD2		%MW2
	%MF3 / %MD3	%MW3
%MF4 / %MD4		%MW4
	...	%MW5
...		...
	%MFi / %MDi	%MWi
%MFi+1 / %MDi+1		%MWi+1

Die folgende Tabelle zeigt die Überschneidung der Gleitkomma- und Doppelwortkonstanten.

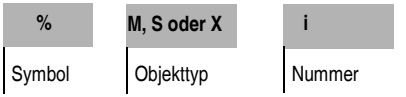
Gleitkommawerte und Doppelwörter	ungerade Adresse	Interne Wörter
%KF0 / %KD0		%KW0
	%KF1 / %KD1	%KW1
%KF2 / %KD2		%KW2
	%KF3 / %KD3	%KW3
%KF4 / %KD4		%KW4
	...	%KW5
...		...
	%kFi / %kDi	%KWi
%KFi+1 / %KDi+1		%KWi+1

Beispiel:
%MF0 entspricht %MW0 und %MW1. %KF543 entspricht %KW543 und %KW544.

Adressierung von Bitobjekten

Syntax

Die Adressierung von Schrittbitobjekten, konstanten Bitobjekten und Systembitobjekten muss mit der folgenden Syntax konform sein:



Beschreibung

Die folgende Tabelle beschreibt die Elemente im Format für die Adressierung.

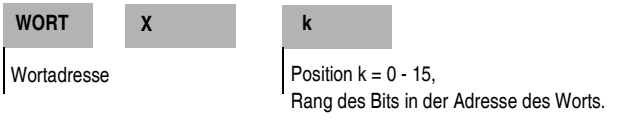
Gruppe	Element	Beschreibung
Symbol	%	Das Prozentzeichen steht immer vor einer Software-Variablen.
Objekttyp	M	Interne Bits speichern Zwischenwerte, während ein Programm läuft.
	S	Systembits bieten Status- und Steuerungsinformationen für die Steuerung.
	X	Schrittbits liefern den Status von Schrittaktivitäten.
Nummer	i	Der Wert für die maximale Nummer richtet sich nach der Anzahl der konfigurierten Objekte.

Beispiel für die Adressierung von Bitobjekten:

- %M25 = internes Bit Nummer 25
- %S20 = Systembit Nummer 20
- %X6 = Schrittbit Nummer 6

Aus Worten extrahierte Bitobjekte

Mit TwidoSoft können Sie eines von 16 Bits eines Worts extrahieren. Die Adresse des Worts wird entsprechend der folgenden Syntax durch den Rang des extrahierten Bits vervollständigt:



Beispiele:

- %MW5:X6 = Bit Nummer 6 des internen Worts %MW5
- %QW5.1:X10 = Bit Nummer 10 des Ausgangsworts %QW5.1

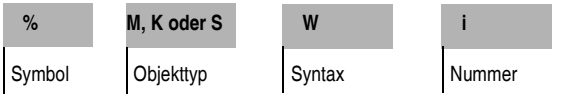
Adressierung von Wortobjekten

Einleitung

Die Adressierung von Wortobjekten, mit Ausnahme der Adressierung von Eingängen/Ausgängen (siehe *Kennzeichnung der Eingänge/Ausgänge, S. 41*) und Funktionsblöcken (siehe *Funktionsblockobjekte, S. 44*), richtet sich nach dem im folgenden beschriebenen Format.

Syntax

Die Adressierung von internen Worten, konstanten Worten und Systemworten muss mit der folgenden Syntax konform sein:



Beschreibung

Die folgende Tabelle beschreibt die Elemente im Format für die Adressierung.

Gruppe	Element	Beschreibung
Symbol	%	Das Prozentzeichen steht immer vor einer internen Adresse.
Objekttyp	M	Interne Worte speichern Zwischenwerte, während ein Programm läuft.
	K	Konstantenworte speichern konstante Werte oder alphanumerische Meldungen. Ihr Inhalt kann nur mit TwidoSoft geschrieben oder geändert werden.
	S	Systemworte bieten Status- und Steuerungsinformationen für die Steuerung.
Syntax	W	Wort (16 Bits).
Nummer	i	Der Wert für die maximale Nummer richtet sich nach der Anzahl der konfigurierten Objekte.

Beispiel für die Adressierung von Wortobjekten:

- %MW15 = internes Wort Nummer 15
- %KW26 = konstantes Wort Nummer 26
- %SW30 = Systemwort Nummer 30

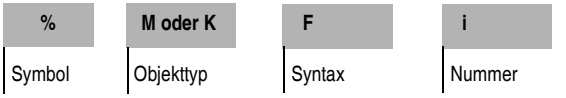
Adressierung von Gleitkommaobjekten

Einleitung

Die Adressierung von Gleitkommaobjekten muss mit der im Folgenden beschriebenen Syntax konform sein. Diese Syntax gilt nicht für die Adressierung von Eingängen/Ausgängen (siehe *Kennzeichnung der Eingänge/Ausgänge, S. 41*) und Funktionsblöcken (siehe *Funktionsblockobjekte, S. 44*).

Syntax

Die Adressierung von internen und konstanten Gleitkommaobjekten muss mit der folgenden Syntax konform sein:



Beschreibung

Die folgende Tabelle beschreibt die Elemente im Format für die Adressierung.

Gruppe	Element	Beschreibung
Symbol	%	Das Prozentzeichen steht immer vor einer internen Adresse.
Objekttyp	M	Interne Gleitkommaobjekte speichern Zwischenwerte, während ein Programm läuft.
	K	Konstante Gleitkommaobjekte speichern konstante Werte. Ihr Inhalt kann nur mit TwidoSoft geschrieben oder geändert werden.
Syntax	F	Objekt (32 Bit).
Nummer	i	Der Wert für die maximale Nummer richtet sich nach der Anzahl der konfigurierten Objekte.

Beispiele für die Adressierung von Gleitkommaobjekten:

- %MF15 = internes Gleitkommaobjekt Nummer 15
- %KF26 = konstantes Gleitkommaobjekt Nummer 26

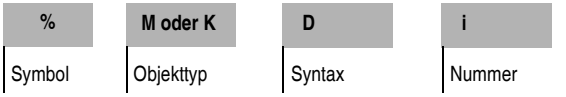
Adressierung von Doppelwortobjekten

Einleitung

Die Adressierung von Doppelwortobjekten muss mit der im Folgenden beschriebenen Syntax konform sein. Diese Syntax gilt nicht für die Adressierung von Eingängen/Ausgängen (siehe *Kennzeichnung der Eingänge/Ausgänge, S. 41*) und Funktionsblöcken (siehe *Funktionsblockobjekte, S. 44*).

Syntax

Die Adressierung von internen und konstanten Doppelworten muss mit der folgenden Syntax konform sein:



Beschreibung

Die folgende Tabelle beschreibt die Elemente im Format für die Adressierung.

Gruppe	Element	Beschreibung
Symbol	%	Das Prozentzeichen steht immer vor einer internen Adresse.
Objekttyp	M	Interne Doppelworte ermöglichen das Ablegen von Zwischenberechnungswerten während der Ausführung eines Programms.
	K	Konstante Doppelworte speichern konstante Werte oder alphanumerische Meldungen. Ihr Inhalt kann nur mit TwidoSoft geschrieben oder geändert werden.
Syntax	D	Doppelwort (32 Bit)-
Nummer	i	Der Wert für die maximale Nummer richtet sich nach der Anzahl der konfigurierten Objekte.

Beispiele für die Adressierung von Doppelwortobjekten:

- %MD15 = internes Doppelwortobjekt Nummer 15
- %KD26 = konstantes Doppelwortobjekt Nummer 26

Kennzeichnung der Eingänge/Ausgänge

Einleitung

Jeder Eingang/Ausgang (E/A) in einer Twido Konfiguration hat eine eindeutige Adresse: Beispielweise repräsentiert die Adresse "%I0.0.4" nur Eingang 4 einer Steuerung.

E/A-Adressen können folgender Hardware zugewiesen werden:

- Steuerung, die als dezentral verknüpfter Master konfiguriert ist
- Steuerung, die als dezentrale E/A konfiguriert ist
- E/A-Erweiterungsmodule

Das AS-Interface-Bus-Schnittstellenmodul TWDNOI10M3 und das CANopen-Feldbusmodul TWDNCO1M verfügen jeweils über ein eigenes, spezielles Adresssystem für die Adressierung der Eingänge/Ausgänge der an den Bus angeschlossenen Slave-Geräte:

- Für TWDNOI10M3 siehe *Adressierung der Ein-/Ausgänge, die den am AS-Interface V2-Bus angeschlossenen Slave-Geräten zugeordnet sind., S. 237*.
- Für TWDNCO1M siehe *Adressieren der PDOs des CANopen-Masters, S. 279*.

Mehrere Referenzen eines Ausgangs bzw. einer Spule

In einem Programm kann es mehrere Referenzen auf einen einzigen Ausgang bzw. eine Spule geben. Nur das Ergebnis der letzten gelösten Referenz wird in den Hardware-Ausgängen aktualisiert. %Q0.0.0 kann z.B. mehr als einmal in einem Programm verwendet werden, ohne dass es eine Warnung für mehrere Vorkommen gibt. Es ist daher wichtig, nur die Gleichung zu bestätigen, die den gewünschten Zustand des Ausgangs ergibt.

⚠ ACHTUNG

UNBEABSICHTIGTER BETRIEB

Ausgänge werden nicht auf doppelte Verwendung geprüft und es werden auch keine Warnungen ausgegeben. Überprüfen Sie die Verwendung von Ausgängen bzw. Spulen, bevor Sie sie in Ihrer Applikation ändern.

Die Nichtbeachtung dieser Anweisung kann Körperverletzung oder Materialschäden zur Folge haben!

Format

Adressieren Sie Eingänge und Ausgänge in folgendem Format.

%	I, Q	x	.	y	.	z
Symbol	**Objekttyp**	**Steuerungsposition**	**Punkt**	**E/A-Typ**	**Punkt**	**Kanalnummer**

Adressieren Sie Eingangs-/Ausgangs-Austauschworte in folgendem Format.

%	I, Q	W	x	.	y
Symbol	**Objekttyp**	**Format**	**Steuerungsposition**	**Punkt**	**E/A-Typ**

Beschreibung

Die folgende Tabelle beschreibt das Format für die E/A-Adressierung.

Gruppe	Element	Wert	Beschreibung
Symbol	%	-	Das Prozentzeichen steht immer vor einer internen Adresse.
Objekttyp	I	-	Eingang. Das "logische Abbild" des elektrischen Zustands eines Eingangs einer Steuerung oder eines E/A-Moduls.
	Q	-	Ausgang. Das "logische Abbild" des elektrischen Zustands eines Ausgangs einer Steuerung oder eines E/A-Moduls.
Steuerungsposition	x	0 1 - 7	Master-Steuerung (dezentral verknüpfter Master). Dezentrale Steuerung (dezentral verknüpfter Slave).
E/A-Typ	y	0 1 - 7	Basis-E/A (lokale E/A auf der Steuerung). Erweiterungs-E/A-Module.
Kanalnummer	z	0 - 31	E/A-Kanalnummer auf der Steuerung oder auf dem Erweiterungs-E/A-Modul. Die Anzahl der verfügbaren E/A-Punkte richtet sich nach der Art der Steuerung oder dem Erweiterungs-E/A-Modultyp.

Beispiele

Die folgende Tabelle zeigt einige Beispiele für die E/A-Adressierung.

E/A-Objekt	Beschreibung
%I0.0.5	Eingang Nr. 5 am Grundgerät SPS (lokale E/A).
%Q0.3.4	Ausgang Nr. 4 am E/A-Erweiterungsmodul der Adresse 3 für das Grundgerät SPS (Erweiterungs-E/A).
%I0.0.3	Eingang Nr. 3 am Grundgerät SPS.
%I3.0.1	Eingang Nr. 1 an der SPS mit dezentralen E/A der Adresse 3 der dezentralen Verbindung.
%I0.3.2	Eingang Nr. 2 am E/A-Erweiterungsmodul der Adresse 3 für das Grundgerät SPS.

Netzwerk-Adressierung

Einführung

Applikationsdaten werden zwischen Partnersteuerungen und der Master-Steuerung in einem Twido Netzwerk über dezentrale Verbindung mit den Netzwerkworten %INW und %QNW ausgetauscht. Ausführliche Informationen finden Sie unter *Kommunikation , S. 87*.

Format

Verwenden Sie für die Netzwerkadressierung das folgende Format.

%	IN,QN	W	x	.	j
Symbol	**Objekttyp**	**Format**	**Steuerungs-position**	**Punkt**	**Wort**

Beschreibung des Formats

Die folgende Tabelle beschreibt das Format für die Netzwerkadressierung.

Gruppe	Element	Wert	Beschreibung
Symbol	%	-	Das Prozentzeichen steht immer vor einer internen Adresse.
Objekttyp	IN	-	Netzwerkeingangswort. Datenübertragung vom Master zum Partner.
	QN	-	Netzwerkausgangswort. Datenübertragung vom Partner zum Master.
Format	W	-	16-Bit-Wort.
Steuerungsposition	x	0 1 - 7	Master-Steuerung (dezentral verknüpfter Master). Dezentrale Steuerung (dezentral verknüpfter Slave).
Wort	j	0 - 3	Jede Partnersteuerung verwendet zwischen einem und vier Worten für den Datenaustausch mit der Master-Steuerung.

Beispiele

Die folgende Tabelle zeigt einige Beispiele für die Netzwerkadressierung.

Netzwerkobjekt	Beschreibung
%INW3.1	Netzwerkwort Nummer 1 der dezentralen Steuerung Nummer 3.
%QNW0.3	Netzwerkwort Nummer 3 der Basissteuerung.

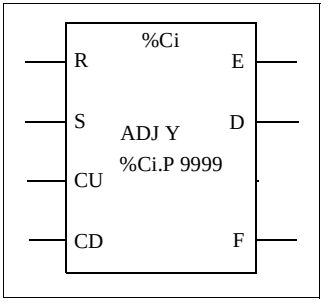
Funktionsblockobjekte

Einführung

Funktionsblöcke bieten Bitobjekte und bestimmte Worte, auf die das Programm zugreifen kann.

Beispiel für einen Funktionsblock

Die folgende Abbildung zeigt einen Zähler-Funktionsblock.



Auf-/Abwärtszähler-Funktionsblock

Bitobjekte

Bitobjekte entsprechen den Blockausgängen. Auf diese Bits können Boolesche Testanweisungen auf eine der folgenden Arten zugreifen:

- Direkt (z.B. LD E), wenn sie in der reversiblen Programmierung mit dem Block verdrahtet sind (siehe *Prinzipien für die Programmierung von Standard-Funktionsblöcken, S. 404*).
- Durch Angabe des Blocktyps (z.B. LD %Ci.E).

Auf Eingänge kann in Form von Anweisungen zugegriffen werden.

Wortobjekte

Wortobjekte entsprechen bestimmten Parametern und Werten:

- Blockkonfigurationsparameter: Einige Parameter sind für das Programm zugänglich (z.B. Vorauswahl von Parametern) und einige sind nicht für das Programm zugänglich (z.B. Zeitbasis).
- Aktuelle Werte: z.B. %Ci.V, der aktuelle Zählwert.

Wortobjekte

Doppelwortobjekte erhöhen die Rechenkapazität Ihrer Twido-Steuerung während der Ausführung von Systemfunktionen wie etwa schnelle Zähler (%FC), sehr schnelle Zähler (%VFC) und Impulsgeneratoren (%PLS).
Die Adressierung von 32-Bit-Doppelwortobjekten, die mit Funktionsblöcken verwendet werden, besteht darin, einfach das Zeichen "D" an die Syntax des Standard-Wortobjekts anzuhängen. Das folgende Beispiel zeigt, wie der aktuelle Wert eines schnellen Zählers im Standardformat und im Doppelwortformat adressiert wird:

- %FCi.V ist der aktuelle Wert des schnellen Zählers im Standardformat.
- %FCi.VD ist der aktuelle Wert des schnellen Zählers im Doppelwortformat.

Hinweis: Doppelwortobjekte werden nicht von allen Twido-Steuerungen unterstützt. Siehe *Hardware-Kompatibilität, S. 34*, um zu ermitteln, ob Ihre Twido-Steuerung Doppelwörter unterstützt.

Vom Programm zugängliche Objekte

In den folgenden entsprechenden Abschnitten finden Sie eine Liste mit Objekten, auf die das Programm zugreifen kann.

- Basis-Funktionsblöcke finden Sie unter *Basis-Funktionsblöcke, S. 402*.
- Erweiterte Funktionsblöcke finden Sie unter *Zu erweiterten Funktionsblöcken gehörige Bit- und Wortobjekte, S. 456*.

Strukturierte Objekte

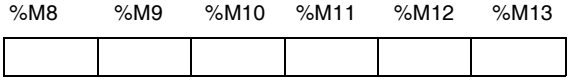
Einführung

Strukturierte Objekte sind Kombinationen aus benachbarten Objekten. Twido unterstützt die folgenden Arten von strukturierten Objekten:

- Bitzeichenketten
- Worttabellen
- Doppelworttabellen
- Gleitkommazahltabellen

Bitzeichenketten

Bitzeichenketten sind eine Reihe aus benachbarten Objektbits des gleichen Typs und mit einer definierten Länge (L).
Beispiel: Bitzeichenkette %M8:6



Hinweis: %M8:6 ist akzeptabel (8 ist ein Vielfaches von 8), während %M10:16 nicht akzeptabel ist (10 ist kein Vielfaches von 8).

Bitzeichenketten können mit der Zuweisungsanweisung verwendet werden (siehe *Zuweisungsanweisungen, S. 428*).

Verfügbare Arten von Bits

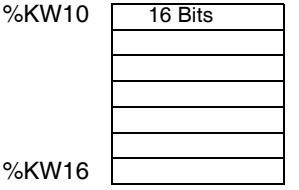
Die folgenden Arten von Bits sind für Bitzeichenketten verfügbar:

Typ	Adresse	Maximale Größe	Schreibzugriff
Digitale Eingangsbits	%I0.0:L oder %I1.0:L (1)	0<L<17	Nein
Digitale Ausgangsbits	%Q0.0:L oder %Q1.0:L (1)	0<L<17	Ja
Systembits	%Si:L wobei i = Vielfaches von 8	0<L<17 und i+L≤ 128	Abhängig von i
Grafcet-Schrittbits	%Xi:L wobei i = Vielfaches von 8	0<L<17 und i+L≤ 95 (2)	Ja (vom Programm)
Interne Bits	%Mi:L wobei i = Vielfaches von 8	0<L<17 und i+L≤ 256 (3)	Ja

Legende:
1. Nur die E/A-Bits 0 bis 16 können in der Bitzeichenfolge lesen. Bei Steuerungen mit 24 Eingängen und Modulen mit 32 E/A können Bits über 16 nicht in der Bitzeichenfolge gelesen werden.
2. Maximum von I i+L für TWWDLCAA10DRF und TWDLCAA16DRF ist 62
3. Maximum von I i+L für TWWDLCAA10DRF und TWDLCAA16DRF ist 128

Worttabellen

Worttabellen sind eine Reihe aus benachbarten Wörtern des gleichen Typs und mit einer definierten Länge (L).
Beispiel: Worttabelle %KW10:7



Worttabellen können mit der Zuweisungsanweisung verwendet werden (siehe *Zuweisungsanweisungen, S. 428*).

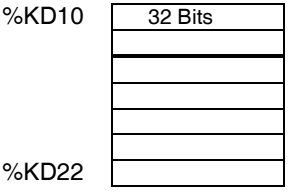
Verfügbare Arten von Wörtern

Für Worttabellen verfügbare Arten von Wörtern:

Typ	Adresse	Maximale Größe	Schreibzugriff
Interne Wörter	%MWi:L	0<L<256 und i+L< 3000	Ja
Konstante Wörter	%KWi:L	0<L<256 und i+L< 256	Nein
Systemwörter	%SWi:L	0<L und i+L<128	Abhängig von i

Doppelworttabellen

Doppelworttabellen sind eine Reihe aus benachbarten Wörtern des gleichen Typs und mit einer definierten Länge (L).
Beispiel: Doppelworttabelle %KD10:7)



Doppelworttabellen können mit der Zuweisungsanweisung verwendet werden (siehe *Zuweisungsanweisungen, S. 428*).

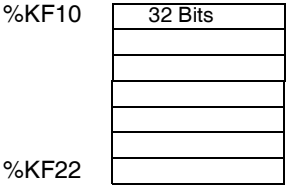
Verfügbare Arten von Doppelwörtern

Für Doppelworttabellen verfügbare Arten von Wörtern:

Typ	Adresse	Maximale Größe	Schreibzugriff
Interne Wörter	%MDi:L	0<L<256 und i+L< 3000	Ja
Konstante Wörter	%KDi:L	0<L und i+L< 256	Nein

Gleitkommazahltabellen

Gleitkommazahltabellen sind eine Reihe aus benachbarten Wörtern des gleichen Typs und mit einer definierten Länge (L).
Beispiel: Gleitkommazahltabelle %KF10:7



Gleitkommazahltabellen können mit der Zuweisungsanweisung verwendet werden (siehe Abschnitt in Erweiterte Anweisungen).

Verfügbare Gleitkommazahltypen

Für Gleitkommazahltabellen verfügbare Arten von Wörtern:

Typ	Adresse	Maximale Größe	Schreibzugriff
Interne Wörter	%MFi:L	0<L<256 und i+L< 3000	Ja
Konstante Wörter	%KFi:L	0<L und i+L<256	Nein

Indexierte Objekte

Einführung

Ein indexiertes Wort ist ein Einzel- oder Doppelwort oder ein Gleitkommawert mit einer indexierten Objektadresse. Es gibt zwei Arten von Objektadressierung:

- Direkte Adressierung
- Indexierte Adressierung

Direkte Adressierung

Eine direkte Adresse eines Objekts wird gesetzt und definiert, wenn ein Programm geschrieben wird.
Beispiel: %M26 ist ein internes Bit mit der direkten Adresse 26.

Indexierte Adressierung

Eine indexierte Adresse eines Objekts bietet eine Methode zum Ändern der Adresse eines Objekts, indem ein Index zur direkten Adresse eines Objekts hinzugefügt wird. Der Inhalt des Index wird zur direkten Adresse des Objekts hinzugefügt. Der Index wird durch ein internes Wort %Mwi definiert. Die Anzahl der "Indexwörter" ist unbegrenzt.
Beispiel: %MW108[%MW2] ist ein Wort mit einer Adresse, die aus der direkten Adresse 108 plus dem Inhalt von Wort %MW2 besteht.
Hat Wort %MW2 den Wert 12, ist ein Schreiben in %MW108[%MW2] äquivalent zum Schreiben in %MW120 (108 plus 12).

Verfügbare Objekte für die indexierte Adressierung

Die folgenden Arten von Objekten sind für die indexierte Adressierung verfügbar.

Typ	Adresse	Maximale Größe	Schreibzugriff
Interne Wörter	%MWi[MWj]	0≤ i+%MWj<3000	Ja
Konstante Wörter	%KWi[%MWj]	0≤ i+%MWj<256	Nein
Interne Doppelwörter	%MDi[MWj]	0≤ i+%MWj<2999	Ja
Konstante Doppelwörter	%KDi[%MWj]	0≤ i+%MWj<255	Nein
Interne Gleitkommawerte	%MFi[MWj]	0≤ i+%MWj<2999	Ja
Konstante Gleitkommawerte	%KFi[%MWj]	0≤ i+%MWj<255	Nein

Indexierte Objekte orte können mit der Zuweisungsanweisung (siehe *Zuweisungsanweisungen, S. 428* für Wörter und Doppelwörter) und in Vergleichsanweisungen (siehe *Vergleichsanweisungen, S. 433* für Wörter und Doppelwörter) verwendet werden. Diese Art der Adressierung ermöglicht es, dass Reihen von Objekten des gleichen Typs (z.B. interne Wörter und Konstanten) nacheinander abgetastet werden, indem der Inhalt des indexierten Objektes vom Programm geändert wird.

Systembit für Indexüberlauf %S20

Ein Überlauf des Index tritt auf, wenn die Adresse eines indexierten Objekts die Grenzwerte des Speicherbereichs überschreitet, der den gleichen Objekttyp enthält. Zusammenfassung:

- Die Objektadresse plus Inhalt des Index ist kleiner als 0.
- Die Objektadresse plus Inhalt des Index ist größer als das größte Wort, das in der Applikation direkt referenziert wird. Die maximale Anzahl ist 2999 (für Wörter %MWi) und 255 (für Wörter %KWi).

Tritt ein Indexüberlauf auf, wird das Systembit %S20 auf 1 gesetzt, und dem Objekt wird der Indexwert 0 zugewiesen.

Hinweis: Der Anwender ist dafür verantwortlich, auf Überlauf zu achten. Das Bit %S20 muss vom Anwenderprogramm auf eine mögliche Bearbeitung gelesen werden. Der Anwender muss bestätigen, dass es auf 0 zurückgesetzt ist.
%S20 (Ausgangszustand = 0):

- Bei Indexüberlauf: wird vom System auf 1 gesetzt.
- Quittierung bei Überlauf: wird vom Anwender nach Änderung des Index auf 0 gesetzt.

Symbolisieren von Objekten

Einführung

Symbole dienen dazu, Sprachobjekte in der Twido Software mit Namen oder anwenderdefinierter Mnemonik zu adressieren. Symbole ermöglichen einen schnellen Überblick und eine schnelle Analyse der Programmlogik und vereinfachen die Entwicklung und Testphase einer Applikation.

Beispiel

Beispiel: WASCH_ENDE ist ein Symbol zur Kennzeichnung eines Timer-Funktionsblocks, der das Ende eines Waschzyklus darstellt. Es ist einfacher, sich über den Zweck an einen Namen zu erinnern als zu versuchen, sich die Funktion einer Programmadresse wie %TM3 einzuprägen.

Richtlinien für die Definition von Symbolen

Beachten Sie folgende Richtlinien für die Definition von Symbolen:

- Maximal 32 Zeichen.
- Buchstaben (A-Z), Zahlen (0 -9) und Unterstriche (_).
- Das erste Zeichen muss ein alphabetisches Zeichen oder ein Zeichen mit Akzent sein. Das Prozentzeichen (%) dürfen Sie nicht verwenden.
- Verwenden Sie keine Leerzeichen oder Sonderzeichen.
- Groß- und Kleinschreibung spielt keine Rolle. Beispiel: Pumpe1 und PUMPE1 ist das gleiche Symbol und kann nur einmal in einer Applikation verwendet werden.

Bearbeiten von Symbolen

Symbole werden im Symbol-Editor definiert und Sprachobjekten zugewiesen. Symbole und deren Kommentare werden mit der Applikation auf der Festplatte des PC gespeichert, sie werden jedoch nicht in der Steuerung gespeichert. Deshalb können sie nicht mit der Applikation in die Steuerung übertragen werden.

Anwenderspeicher

3

Auf einen Blick

Inhalt dieses Kapitels

In diesem Kapitel wird der Aufbau und die Verwendung des Twido Anwenderspeichers beschrieben.

Inhalt dieses Kapitels

Dieses Kapitel enthält die folgenden Themen:

Thema	Seite
Struktur des Anwenderspeichers	54
Backup und Wiederherstellung ohne Sicherungs-Cartridge oder Speichererweiterung	57
Backup und Wiederherstellung mit einer 32KB-Sicherungs-Cartridge	59
Verwendung der 64 KB-Erweiterungs-Sicherungskassette	61

Struktur des Anwenderspeichers

Einführung

Der für Ihre Anwendung zugängliche Steuerungsspeicher ist in zwei verschiedene Sätze unterteilt:

- Bitwerte
- Wortwerte (vorzeichenbehaftete 16-Bit-Werte) und Doppelwortwerte (vorzeichenbehaftete 32-Bit-Werte)

Bitspeicher

Der Bitspeicher befindet sich im internen RAM-Speicher der Steuerung. Er enthält die Abbildung von 128 Bitobjekten.

Wortspeicher

Der Wortspeicher (16 Bits) unterstützt:

- **Dynamische Wörter**: Laufzeitspeicher (nur im RAM gespeichert)
- **Speicherwörter (%MW) und Speicherdoppelwörter (%MD)**: dynamische Systemdaten und Systemdaten
- **Programm**: Deskriptoren und ausführbare Codes für Tasks
- **Konfigurationsdaten**: konstante Wörter, Ausgangswerte und Eingangs-/Ausgangskonfiguration

Speichersicherungsarten

Im Folgenden werden die verschiedenen Speichersicherungsarten für Twido-Steuerungen aufgeführt:

- Random Access Memory.
  Interner flüchtiger Speicher: enthält dynamische Wörter, Speicherwörter, Programm- und Konfigurationsdaten.
- EEPROM
  Ein integrierter 32-KB-EEPROM-Speicher, der eine interne Programm- und Datensicherung bietet. Er schützt das Programm mehr als 30 Tage vor Beschädigung durch Batterieausfall oder Spannungsverlust. Er enthält Programm- und Konfigurationsdaten. Er speichert maximal 512 Speicherwörter. Das Programm wird hier nicht gesichert, wenn eine 64-KB-Speichererweiterungskassette verwendet wird und Twido für die Nutzung der 64-KB-Speichererweiterungskassette konfiguriert wurde.
- 32-KB-Sicherungskassette
  Eine optionale, externe Sicherungskassette, die für die Speicherung eines Programms und die Übertragung dieses Programms an andere Twido-Steuerungen verwendet wird. Die Sicherungskassette kann zur Aktualisierung des Programms im Steuerungs-RAM verwendet werden. Sie enthält das Programm und Konstanten, aber keine Speicherwörter.
- 64-KB-Speichererweiterungskassette
  Eine optionale, externe Sicherungskassette, die ein Programm mit bis zu 64 KB speichert. Sie muss so lange in der Steuerung eingesteckt bleiben, wie das Programm genutzt wird.

Sichern des Speichers

Ihr Steuerungsprogramm und Ihre Speicherwörter können in folgenden Speichern gesichert werden:

- RAM (bis zu 30 Tage bei voller Batterie)
- EEPROM-Speicher, (maximal 32 KB)

Das Programm wird automatisch vom EEPROM-Speicher in den RAM-Speicher übertragen, wenn das Programm im RAM verloren geht (oder wenn keine Batterie vorhanden ist).
Die Übertragung kann auch manuell in TwidoSoft durchgeführt werden.

Speicherkonfigurationen

In den folgenden Tabellen werden die mit kompakten und modularen Twido-Steuerungen möglichen Speicherkonfigurationsarten beschrieben.

Speichertyp	Kompakte Steuerungen				
	10DRF	**16DRF**	**24DRF**	**40DRF (32k)**	**40DRF** (64k)**
Interner RAM Mem. 1*	10 KB	10 KB	10 KB	10 KB	10 KB
Externer RAM Mem. 2*		16 KB	32 KB	32 KB	64 KB
Interner EEPROM	8 KB	16 KB	32 KB	32 KB	32 KB***
Externer EEPROM	32 KB	32 KB	32 KB	32 KB	64 KB
Maximale Programmgröße	8 KB	16 KB	32 KB	32 KB	64 KB
maximale externe Sicherung	8 KB	16 KB	32 KB	32 KB	64 KB

Speichertyp	Modulare Steuerungen		
	20DUK 20DTK	**20DRT 40DUK 40DTK (32 KB)**	**20DRT 40DUK 40DTK** (64 KB)**
Interner RAM Mem. 1*	10 KB	10 KB	10 KB
Externer RAM Mem. 2*	32 KB	32 KB	64 KB
Interner EEPROM	32 KB	32 KB	32 KB***
Externer EEPROM	32 KB	32 KB	64 KB
Maximale Programmgröße	32 KB	32 KB	64 KB
maximale externe Sicherung	32 KB	32 KB	64 KB

(*) Mem. 1 und Mem. 2 der Speicherbilanz
(**) In diesem Fall muss die 64 KB-Speicherkassette in Twido installiert bleiben und in der Konfiguration deklariert werden, wenn sie nicht angegeben ist.
(***) Reserviert für die Speicherung der ersten 512 %MW-Wörter oder die ersten 256 %MD-Doppelwörter.

Backup und Wiederherstellung ohne Sicherungs-Cartridge oder Speichererweiterung

Einleitung

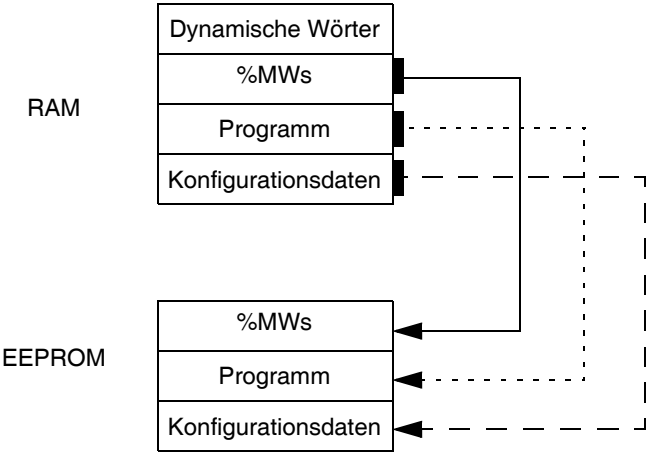
Nachfolgend sind die backup- und Wiederherstellungsfunktionen in modularen und kompakten Steuerungen ohne eine backup Sicherungs-Cartridge oder einen eingesteckten erweiterten Speicher beschrieben.

Auf einen Blick

Twido-Programme, Speicherworte und Konfigurationsdaten können mittels des internen EEPROM der Steuerung gesichert werden. Da die Speicherung eines Programms im internen EEPROM jegliche zuvor gesicherten Speicherworte löscht, muss zunächst das Programm gesichert werden, bevor die konfigurierten Speicherworte gespeichert werden können. Dynamische Daten können in Speicherworten gespeichert und dann im EEPROM gesichert werden. Wenn kein Programm im internen EEPROM gespeichert ist, können Sie keine Speicherworte im EEPROM sichern.

Speicherstruktur

Nachfolgend ist ein Diagramm der Speicherstruktur einer Steuerung abgebildet. Die Pfeile zeigen, welche Inhalte des RAM im EEPROM gesichert werden können:



Programm Backup

Nachfolgend ist das Verfahren zur Sicherung Ihres Programms im EEPROM aufgeführt:

Schritt	Aktion
1	Folgende Bedingungen müssen erfüllt sein: Es befindet sich ein gültiges Programm im RAM.
2	Rufen Sie in der Twido-Software das Menü unter 'Steuerung' auf, führen Sie einen Bildlauf bis Backup' aus, und klicken Sie darauf.

Wiederherstellung des Programms

Während der Einschaltung wird das Programm aus dem EEPROM im RAM wiederhergestellt, wenn folgende Bedingung erfüllt ist (vorausgesetzt, es befindet sich keine Cartridge oder Speichererweiterung in der Steuerung):

- Das RAM-Programm ist nicht gültig.

Gehen Sie folgendermaßen vor, um ein Programm manuell aus dem EEPROM wiederherzustellen:

- Rufen Sie in der Twido-Software das Menü unter 'Steuerung' auf, und klicken Sie auf 'Wiederherstellen'.

Daten (%MWs) Backup

Nachfolgend ist das Verfahren zur Sicherung Ihrer Daten (Speicherworte) im EEPROM aufgeführt:

Schritt	Aktion
1	Die folgenden Bedingungen müssen erfüllt sein, damit dieses Verfahren ausgeführt werden kann: Es befindet sich ein gültiges Programm im RAM (%SW96:X6=1). Dasselbe gültige Programm ist bereits im EEPROM gesichert. Die Speicherworte sind im Programm konfiguriert.
2	Setzen Sie %SW97 auf die Länge der zu sichernden Speicherworte. **Hinweis:** Die Länge darf die konfigurierte Speicherwortlänge nicht überschreiten und sie muss größer als 0, jedoch nicht größer als 512 sein.
3	Setzen von %SW96:X0 auf 1.

Datenwiederherstellung (%MWs)

Stellen Sie %MWs manuell durch Setzen des Systembits %S95 auf 1 wieder her. Die folgenden Bedingungen müssen erfüllt sein, damit dieses Verfahren ausgeführt werden kann:

- Es muss ein gültiges backup-Applikation im EEPROM vorhanden sein.
- Die Applikation im RAM stimmt mit der backup-Applikation im EEPROM überein.
- Der backup Speicherwörter sind gültig.

Backup und Wiederherstellung mit einer 32KB-Sicherungs-Cartridge

Einleitung

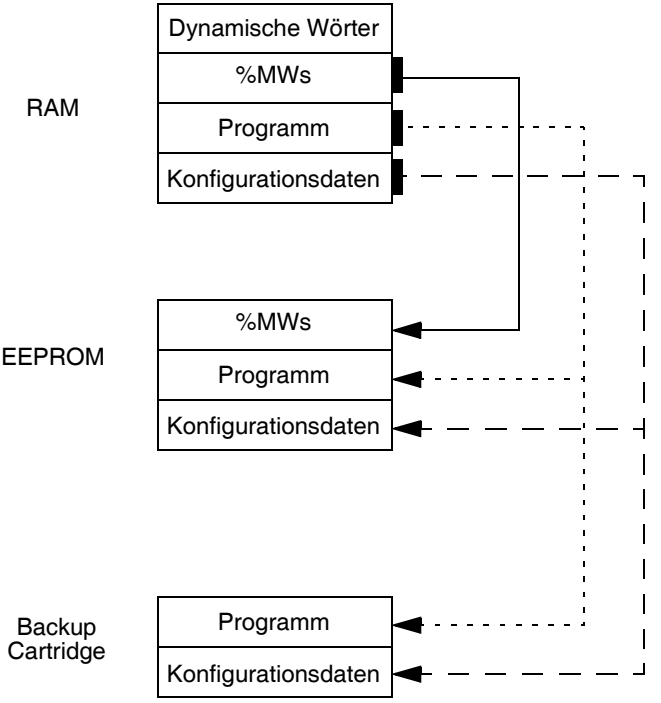
Nachfolgend sind die backup- und Wiederherstellungsfunktionen in modularen und kompakten Steuerungen unter Verwendung einer 32-KB-Sicherungs-Cartridge beschrieben.

Auf einen Blick

Die Sicherungs-Cartridge wird für die Speicherung eines Programms und die Übertragung dieses Programms an andere Twido-Steuerungen verwendet. Sie sollte aus einer Steuerung entfernt und beiseite gelegt werden, wenn das Programm installiert oder gesichert worden ist. Nur Programm- und Konfigurationsdaten können auf der Cartridge gespeichert werden (%MWs können nicht auf der 32KB-Sicherungs-Cartridge gespeichert werden). Dynamische Daten können in Speicherworten gespeichert und dann im EEPROM gesichert werden. Wenn die Programminstallation vollständig ist, gehen alle %MWs verloren, die vor der Installation im internen EEPROM gesichert worden sind.

Speicherstruktur

Nachfolgend ist ein Diagramm der Speicherstruktur einer Steuerung mit installierter Sicherungs-Cartridge abgebildet. Die Pfeile zeigen, welche RAM-Inhalte im EEPROM und in der Cartridge gesichert werden können:



Programm Backup

Nachfolgend ist das Verfahren zur Sicherung Ihres Programms in der Sicherungs-Cartridge aufgeführt:

Schritt	Aktion
1	Schalten Sie die Steuerung aus.
2	Stecken Sie die Sicherungs-Cartridge ein.
3	Schalten Sie die Steuerung ein.
4	Rufen Sie in der Twido-Software das Menü unter 'Steuerung' auf, führen Sie einen Bildlauf bis 'Backup' aus, und klicken Sie darauf.
5	Schalten Sie die Steuerung aus.
6	Nehmen Sie die Sicherungs-Cartridge aus der Steuerung heraus.

Wiederherstellung des Programms

Gehen Sie folgendermaßen vor, um ein in einer Sicherungs-Cartridge gespeichertes Programm in eine Steuerung zu laden:

Schritt	Aktion
1	Schalten Sie die Steuerung aus.
2	Stecken Sie die Sicherungs-Cartridge ein.
3	Schalten Sie die Steuerung ein. (Wenn Autostart konfiguriert ist, müssen Sie die Steuerung erneut aus- und wieder einschalten, um in den Modus 'Run' zu gelangen.)
4	Schalten Sie die Steuerung aus.
5	Nehmen Sie die Sicherungs-Cartridge aus der Steuerung heraus.

Daten (%MWs) Backup

Nachfolgend ist das Verfahren zur Sicherung Ihrer Daten (Speicherworte) im EEPROM aufgeführt:

Schritt	Aktion
1	Die folgenden Bedingungen müssen erfüllt sein, damit dieses Verfahren ausgeführt werden kann: Es befindet sich ein gültiges Programm im RAM. Dasselbe gültige Programm ist bereits im EEPROM gesichert. Die Speicherworte sind im Programm konfiguriert.
2	Setzen Sie %SW97 auf die Länge der zu sichernden Speicherworte. **Hinweis:** Die Länge darf die konfigurierte Speicherwortlänge nicht überschreiten und sie muss größer als 0, jedoch nicht größer als 512 sein.
3	Setzen von %SW96:X0 auf 1.

Datenwiederherstellung (%MWs)

Stellen Sie %MWs manuell durch Setzen des Systembits %S95 auf 1 wieder her.
Die folgenden Bedingungen müssen erfüllt sein, damit dieses Verfahren ausgeführt werden kann:

- Es muss eine gültige backup-Applikation im EEPROM vorhanden sein.
- Die Applikation im RAM stimmt mit der backup-Applikation im EEPROM überein.
- Die backup Speicherwörter sind gültig.

Verwendung der 64 KB-Erweiterungs-Sicherungskassette

Einleitung

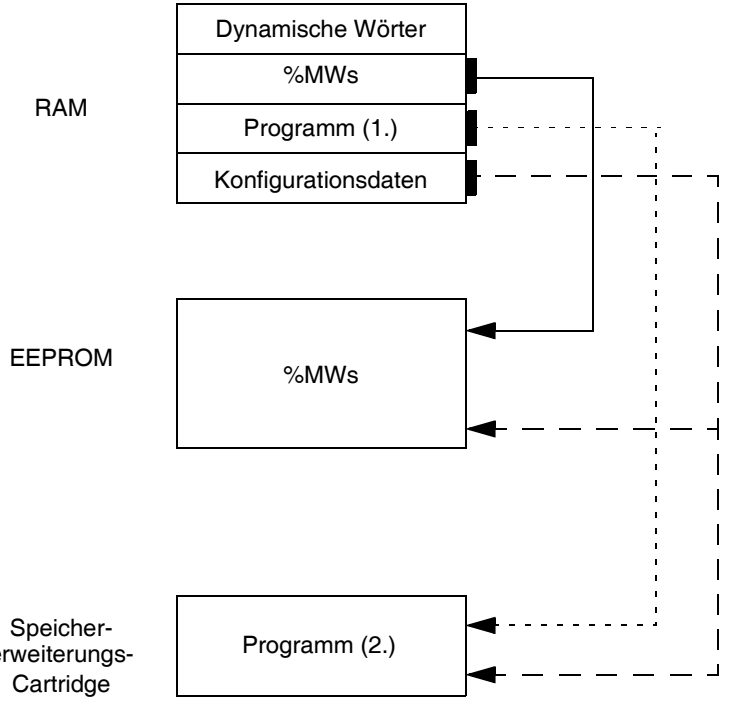
Nachfolgend ist die Nutzung der Speicherfunktionen in modularen Steuerungen unter Verwendung einer 64 KB-Erweiterungs-Sicherungskassette beschrieben.

Auf einen Blick

Die 64 KB-Erweiterungs-Sicherungskassette wird verwendet, um die Programmspeicherungskapazitäten Ihrer Twido-Steuerung von 32 KB auf 64 KB zu erhöhen. Sie muss so lange in der Steuerung eingesteckt bleiben, wie das erweiterte Programm genutzt wird. Wenn die Sicherungskassette entfernt wird, geht die Steuerung in den angehalten Status über. Speicherwörter werden weiterhin im EEPROM der Steuerung gesichert. Dynamische Daten können in Speicherworten gespeichert und dann im EEPROM gesichert werden. Die 64 KB-Erweiterungs-Sicherungskassette weist dasselbe Einschaltverhalten wie die 32 KB-Erweiterungs-Sicherungskassette auf.

Speicherstruktur

Nachfolgend ist ein Diagramm der Speicherstruktur einer Steuerung mit installierter Erweiterungs-Sicherungskassette abgebildet. Die Pfeile geben die ausgehend vom RAM im EEPROM und in der 64 KB-Erweiterungs-Sicherungskassette gesicherten Elemente an:



Software konfigurieren und Speichererweiterung installieren

Bevor Sie Ihr erweitertes Programm schreiben, müssen Sie die 64 KB-Erweiterungs-Sicherungskassette in Ihrer Steuerung installieren. Nachfolgend sind die vier zu befolgenden Schritte beschrieben:

Schritt	Aktion
1	Geben Sie in der Twido-Software unter dem Optionsmenü 'Hardware' 'TWDXCPMFK64' ein.
2	Schalten Sie die Steuerung aus.
3	Stecken Sie die 64 KB-Erweiterungs-Sicherungskassette ein.
4	Schalten Sie die Steuerung ein.

Speichern Sie Ihr Programm.

Wenn die 64 KB-Erweiterungs-Sicherungskassette installiert und Ihr Programm geschrieben ist:

- Rufen Sie in der Twido-Software das Menü unter 'Steuerung' auf, führen Sie einen Bildlauf bis Backup' aus, und klicken Sie darauf.

Daten (%MWs) Backup

Nachfolgend ist das Verfahren zur Sicherung Ihrer Daten (Speicherworte) im EEPROM aufgeführt:

Schritt	Aktion
1	Die folgenden Bedingungen müssen erfüllt sein, damit dieses Verfahren ausgeführt werden kann: Es muss ein gültiges Programm vorhanden sein. Die Speicherwörter sind im Programm konfiguriert.
2	Setzen Sie %SW97 auf die Länge der zu sichernden Speicherworte. **Hinweis:** Die Länge darf die konfigurierte Speicherwortlänge nicht überschreiten und sie muss größer als 0, jedoch nicht größer als 512 sein.
3	Setzen von %SW96:X0 auf 1.

Datenwiederherstellung (%MWs)

Stellen Sie %MWs manuell durch Setzen des Systembits %S95 auf 1 wieder her. Die folgenden Bedingungen müssen erfüllt sein, damit dieses Verfahren ausgeführt werden kann:

- Es muss ein gültiges Programm vorhanden sein.
- Der backup Speicherwörter sind gültig.

Betriebszustände der Steuerung

4

Auf einen Blick

Inhalt dieses Kapitels

In diesem Kapitel werden die Betriebszustände der Steuerung und die zyklische und periodische Programmausführung beschrieben. Außerdem werden Spannungsausfälle und Spannungsrückkehr beschrieben.

Inhalt dieses Kapitels

Dieses Kapitel enthält die folgenden Themen:

Thema	Seite
Zyklische Abtastung	64
Periodische Abtastung	66
Prüfen der Zykluszeit	69
Betriebszustände	70
Umgehen mit Spannungsunterbrechungen und Spannungsrückkehr	72
Umgehen mit Warmstart	74
Umgehen mit Kaltstart	76
Initialisierung der Objekte	78

Zyklische Abtastung

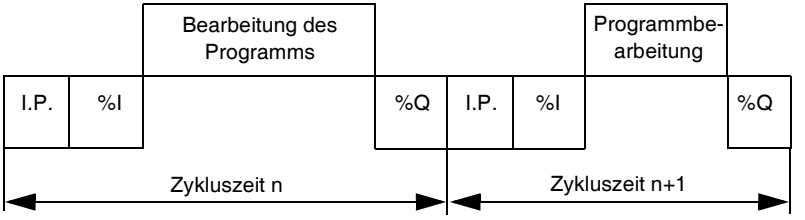
Einleitung

Die zyklische Abtastung ist die Aneinanderreihung der Steuerungszyklen. Nachdem die Aktualisierung der Ausgänge bewirkt wurde (dritte Phase des Task-Zyklus), führt das System eine bestimmte Anzahl eigener Tasks aus und löst sofort einen weiteren Task-Zyklus aus.

Hinweis: Die Dauer der Abtastung des Anwenderprogramms wird durch den Watchdog-Timer der Steuerung geregelt und darf 500 ms nicht überschreiten. Andernfalls tritt ein Fehler auf und schaltet die Steuerung sofort in den Halt-Modus. In diesem Modus werden die Ausgänge standardmäßig auf ihren Fehlerzustand gesetzt.

Funktionsweise

Die nachfolgende Abbildung zeigt die Ausführungsphasen der zyklischen Abtastung.



Beschreibung der Phasen eines Zyklus

In der folgenden Tabelle sind die Phasen eines Zyklus beschrieben.

Adresse	Phase	Beschreibung
I.P.	Interne Bearbeitung	Das System beobachtet die Steuerung implizit (Verwaltung von Systembits und Systemworten, Aktualisierung von aktuellen Timer-Werten, Aktualisierung von Statusleuchten, Erkennung von RUN/STOP-Schaltern usw.) und bearbeitet Anforderungen von TwidoSoft (Änderungen und Animation).
%I, %IW	Erfassung der Eingänge	Schreiben des Zustands der mit den digitalen und analogen Modulen verbundenen Eingänge in den Speicher.
-	Programmbearbeitung	Das vom Anwender geschriebene Anwendungsprogramm wird ausgeführt.
%Q, %QW	Aktualisierung der Ausgänge	Schreiben der Bits oder Worte der mit den digitalen und analogen Modulen verbundenen Ausgänge.

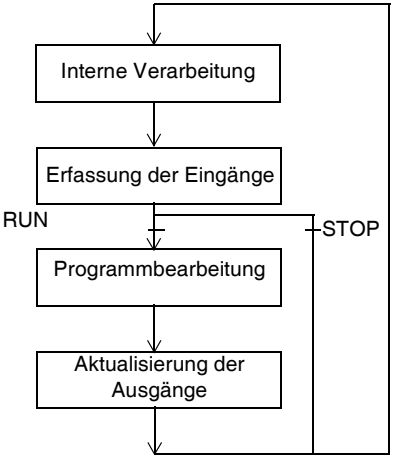
Betriebsart

Steuerung im RUN-Modus, der Prozessor führt folgende Operationen aus:
- Interne Verarbeitung
- Erfassung der Eingänge
- Bearbeiten des Anwendungsprogramms
- Aktualisierung der Ausgänge

Steuerung im STOP-Modus, der Prozessor führt folgende Operationen aus:
- Interne Verarbeitung
- Erfassung der Eingänge

Abbildung

Die folgende Abbildung zeigt die Betriebszyklen.



Prüfzyklus

Der Prüfzyklus wird vom Watchdog durchgeführt.

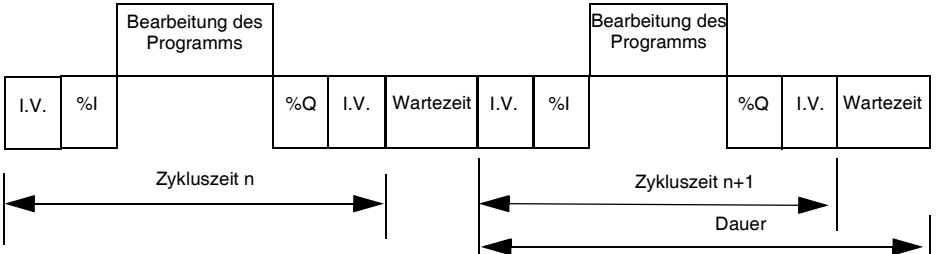
Periodische Abtastung

Einleitung

In dieser Betriebart werden die Erfassung der Eingänge, die Verarbeitung des Applikationsprogramms und die Aktualisierung der Ausgänge entsprechend einem Intervall (von 2 bis 150 ms), der während der Konfiguration festgelegt wurde, periodisch ausgeführt.
Zu Beginn der Steuerungsabtastung startet ein Timer, dessen Wert auf die während der Konfiguration festgelegte Periode initialisiert wird, die Abwärtszählfunktion. Das Abtasten der Steuerung muss vor dem Ende der Abwärtszählfunktion und vor dem Beginn einer neuen Abtastung enden.

Funktionsweise

Die nachfolgende Abbildung zeigt die Ausführungsphasen der periodischen Abtastung.



I.V: Interne Verarbeitung

Beschreibung der Betriebsphasen

Die folgende Tabelle beschreibt die Betriebsphasen.

Adresse	Phase	Beschreibung
I.P.	Interne Bearbeitung	Das System beobachtet die Steuerung implizit (Verwaltung von Systembits und Systemworten, Aktualisierung von aktuellen Timer-Werten, Aktualisierung von Statusleuchten, Erkennung von RUN/STOP-Schaltern usw.) und bearbeitet Anforderungen von TwidoSoft (Änderungen und Animation).
%I, %IW	Erfassung der Eingänge	Schreiben des Zustands der mit den digitalen und analogen Modulen verbundenen Eingänge in den Speicher.
-	Programmbearbeitung	Das vom Anwender geschriebene Anwendungsprogramm wird ausgeführt.
%Q, %QW	Aktualisierung der Ausgänge	Schreiben der Bits oder Worte der mit den digitalen und analogen Modulen verbundenen Ausgänge.

Betriebsart

Steuerung im RUN-Modus, der Prozessor führt folgende Operationen aus:

- Interne Verarbeitung
- Erfassung der Eingänge
- Bearbeiten des Anwendungsprogramms
- Aktualisierung der Ausgänge

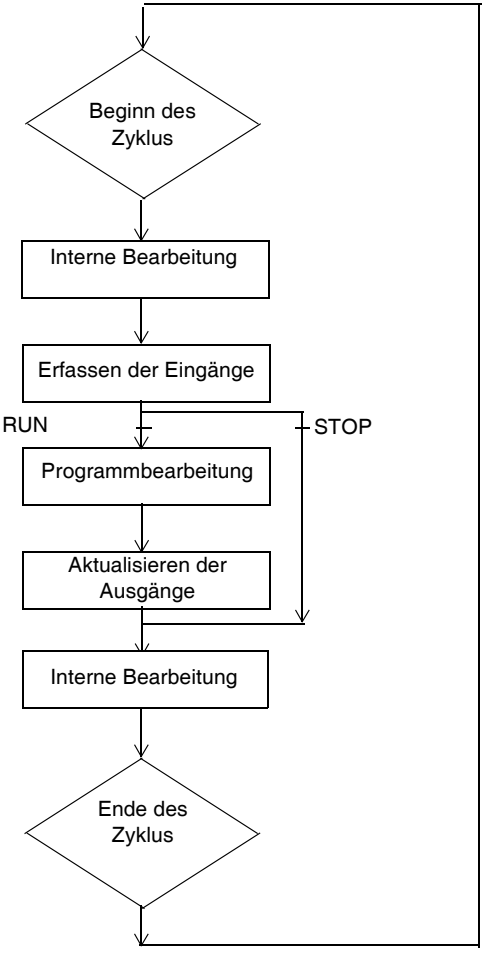
Ist der Zeitraum nicht abgelaufen, beendet der Prozessor den Betriebszyklus bis zum Ende des Zeitraums für die interne Bearbeitung. Wenn die Dauer des Betriebs die dem Zeitraum zugewiesene Dauer übersteigt, meldet die Steuerung einen Zeitraumüberlauf, indem sie das Systembit %S19 auf 1 setzt. Die Verarbeitung wird bis zum Ende der gesamten Ausführung ausgeführt. Die Zeitgrenze des Watchdogs darf jedoch nicht überschritten werden. Der folgende Zyklus ist eingeknüpft, nachdem die Ausgänge des Zyklus in der Bearbeitung implizit geschrieben werden.

Steuerung im STOP-Modus, der Prozessor führt folgende Operationen aus:

- Interne Verarbeitung
- Erfassung der Eingänge

Abbildung

Die folgende Abbildung zeigt die Betriebszyklen.



Prüfzyklus

Es werden zwei Prüfungen ausgeführt:

- Zeitraumüberlauf
- Watchdog

Prüfen der Zykluszeit

Allgemeines

Der Task-Zyklus wird durch einen Watchdog-Timer namens Tmax (maximale Dauer des Task-Zyklus) reguliert. Es können Applikationsfehler aufgezeigt werden (unendliche Schleifen usw.), und es kann eine maximale Dauer für die Aktualisierung der Ausgänge sichergestellt werden.

Software-Watchdog (periodischer oder zyklischer Betrieb)

Beim periodischen oder zyklischen Betrieb verursacht das Auslösen des Watchdog einen Software-Fehler. Die Applikation wird in den Modus HALT geschaltet, und das Systembit %S11 wird auf 1 gesetzt. Der Neustart des Tasks erfordert eine Verbindung zu TwidoSoft, damit die Fehlerursache analysiert werden kann, eine Applikation zum Korrigieren des Fehlers und ein Zurücksetzen des Programms auf RUN.

Hinweis: Im Betriebszustand HALT wird die Applikation sofort aufgrund eines Applikations-Softwarefehlers wie z.B. Zyklusüberlauf gestoppt. Die Daten speichern die aktuellen Werte, wodurch die Fehlerursache analysiert werden kann. Das Programm wird während der laufenden Anweisung beendet. Die Kommunikation mit der Steuerung ist möglich.

Prüfen bei periodischem Betrieb

Während des periodischen Betriebs ermöglicht eine zusätzliche Prüfung das Erkennen eines Zeitraumüberlaufs:

- **%S19** zeigt an, dass der Zeitraum überschritten wurde. Er wird gesetzt auf:
  - 1 vom System, wenn die Zykluszeit größer ist als der Task-Zeitraum,
  - 0 vom Anwender.
- **%SW0** enthält den Wert des Zeitraums (0-150 ms). Er ist:
  - während eines Kaltstarts initialisiert durch den während der Konfiguration gewählten Wert und
  - kann vom Anwender geändert werden.

Laufzeit des Master-Tasks

Die folgenden Systemworte ermöglichen das Abrufen von Informationen über die Zykluszeit der Steuerung:

- **%SW11** initialisiert die maximale Dauer für den Watchdog (10 bis 500 ms).
- **%SW30** enthält die Ausführungszeit des letzten Steuerungszyklus.
- **%SW31** enthält die Ausführungszeit des längsten Steuerungszyklus seit dem letzten Kaltstart.
- **%SW32** enthält die Ausführungszeit des kürzesten Steuerungszyklus seit dem letzten Kaltstart.

Hinweis: Sie können auch über den Konfigurationseditor auf diese verschiedenen Informationen zugreifen.

Betriebszustände

Einleitung

Twido Soft wird verwendet, um drei Gruppen von Betriebszuständen zu berücksichtigen:

- Prüfen
- Ausführen oder Produktion
- Stoppen

Starten über Grafcet

Sie können von Grafcet aus auf die verschiedenen Betriebszustände zugreifen oder, indem Sie Grafcet verwenden und eine der folgenden Methoden anwenden:

- Grafcet-Initialisierung
- Voreinstellen von Schritten
- Beibehalten einer Situation
- Einfrieren von Graphen

Durch vorläufige Bearbeitung und Einsatz von Systembits erzielen Sie eine effektive Betriebszustands-Verwaltung ohne das Anwenderprogramm zu verkomplizieren oder zu überlasten.

Grafcet-Systembits

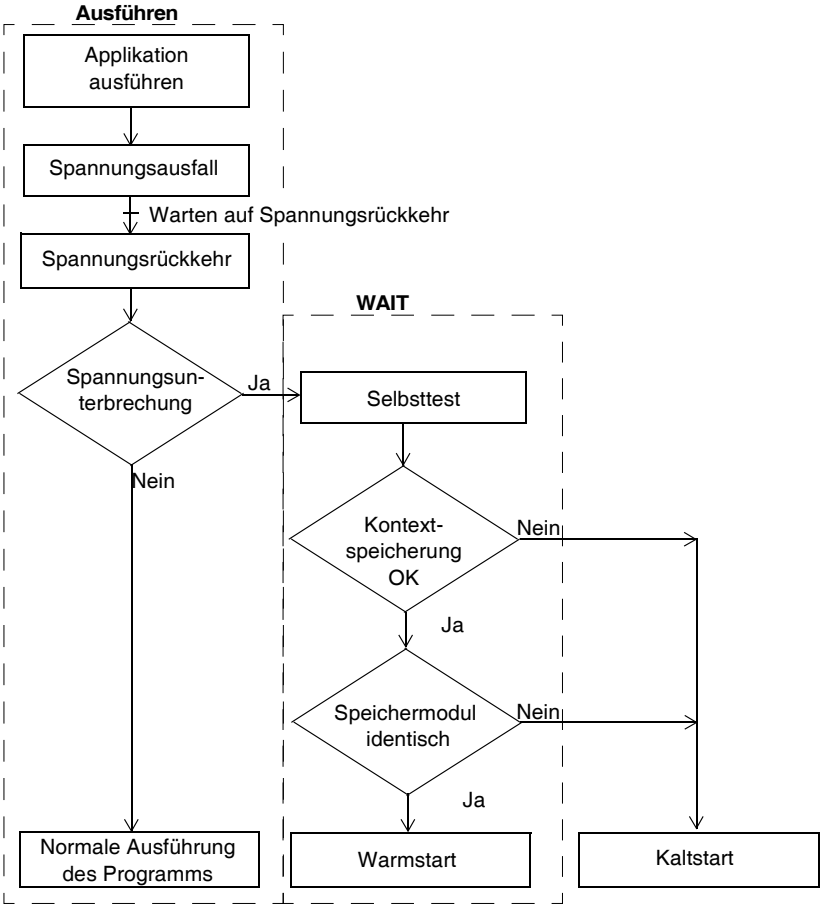
Der Einsatz der Bits %S21, %S22 und %S23 ist einzig für die vorläufige Bearbeitung reserviert. Diese Bits werden automatisch durch das System auf 0 zurückgesetzt und dürfen nur von der Anweisung Set **S** geschrieben werden.
Die folgende Tabelle zeigt die mit Grafcet verknüpften Systembits:

Bit	Funktion	Beschreibung
%S21	GRAFCET-Initialisierung	Normalerweise auf 0; das Bit wird durch Folgendes auf 1 gesetzt: ● Einen Kaltstart, **%S0=1**; ● Den Benutzer; nur im Programmabschnitt zur Vorverarbeitung mit Hilfe der Anweisung Set **S %S21** oder einer Spule Set**-(S)- %S21**. Folgen: ● Deaktivierung aller aktiven Schritte. ● Aktivierung aller Ausgangsschritte.
%S22	GRAFCET-RÜCKSETZEN	Normalerweise im Status 0; das Bit kann nur durch das Programm während der Vorverarbeitung auf 1 gesetzt werden. Folgen: ● Deaktivierung aller aktiven Schritte. ● Abtasten der Ablaufbearbeitung wird gestoppt.
%S23	Vorpositionierung und Einfrieren des GRAFCET	Normalerweise im Status 0; das Bit kann nur durch das Programm während der Vorverarbeitung auf 1 gesetzt werden. ● Vorpositionierung durch Setzen von %S22 auf 1. ● Die Schritte vorpositionieren, die von einer Reihe von Anweisungen S Xi aktiviert werden sollen. ● Vorpositionierung freigeben, indem %S23 auf 1 gesetzt wird. Einfrieren einer Situation: ● In der Ausgangssituation: durch Beibehalten von %S21 auf 1 durch das Programm. ● In einer "leeren" Situation: durch Beibehalten von %S22 auf 1 durch das Programm. ● In einer Situation, die durch das Beibehalten von %S23 auf 1 festgelegt ist.

Umgehen mit Spannungsunterbrechungen und Spannungsrückkehr

Abbildung

Die folgende Abbildung zeigt die verschiedenen vom System erkannten Arten eines Neustarts. Ist die Dauer der Unterbrechung kürzer als die Filterzeit der Spannungsversorgung (ca. 10 ms bei AC-Versorgung und 1 ms bei DC-Versorgung), wird diese Unterbrechung vom Programm nicht bemerkt, es läuft normal weiter.



Hinweis: Der Kontext wird in einem batteriegepufferten RAM gespeichert. Beim Anlauf prüft das System den Zustand der Batterie und den gespeicherten Kontext, um zu ermitteln, ob ein Warmstart durchgeführt werden kann.

Run/Stop-Eingangsbit und Auto-Run

Das Run/Stop-Eingangsbit hat eine höhere Priorität als die Option "Autostart in Ausführen", die im Dialogfeld "Abtastmodus" verfügbar ist. Ist das Run/Stop-Bit gesetzt, startet die Steuerung bei Spannungsrückkehr im Run-Modus.
Der Modus der Steuerung wird wie folgt ermittelt:

Run/Stop-Eingangsbit	Autostart in Ausführen	Resultierender Zustand
Null	Null	Stopp
Null	Eins	Stopp
Steigende Flanke	Nicht relevant	Run
Eins	Nicht relevant	Run
Nicht in Software konfiguriert	Null	Stopp
Nicht in Software konfiguriert	Eins	Run

Hinweis: Ist die Steuerung bei allen kompakten Steuerungen mit einer Softwareversion V1.0 bei Spannungsunterbrechung im Betriebszustand RUN und das Flag "Autostart in Ausführen" nicht im Dialogfeld "Abtastmodus" gesetzt, dann startet die Steuerung bei Spannungsrückkehr im Betriebszustand STOP neu. Andernfalls wird ein Neustart durchgeführt.

Hinweis: Funktioniert die Batterie bei allen modularen Steuerungen und bei allen kompakten Steuerungen mit der Softwareversion V1.11 bei Spannungsunterbrechung normal, startet die Steuerung in dem Betriebszustand neu, der zum Zeitpunkt der Spannungsunterbrechung eingestellt war. Das Flag "Autostart in Ausführen", das im Dialogfeld "Abtastmodus" eingestellt wird, hat bei Spannungsrückkehr keine Auswirkungen auf den Betriebszustand.

Funktionsweise

Die folgende Tabelle beschreibt die Bearbeitungsphasen bei Spannungsunterbrechungen.

Phase	Beschreibung
1	Bei einer Spannungsunterbrechung speichert das System den Applikationskontext und die Uhrzeit der Unterbrechung.
2	Es setzt alle Ausgänge in den Fehlerzustand (Zustand 0).
3	Bei Spannungsrückkehr wird der gespeicherte Kontext mit dem in Bearbeitung befindlichen Kontext verglichen. Dieser Vergleich ermöglicht die Ermittlung des auszuführenden Starttyps: ● Hat sich der Applikationskontext geändert (Verlust von Systemkontext oder neue Applikation), initialisiert die Steuerung die Applikation: Kaltstart (systematisch für kompakte Steuerungen). ● Ist der Applikationskontext der gleiche, startet die Steuerung ohne Dateninitialisierung neu: Warmstart.

Umgehen mit Warmstart

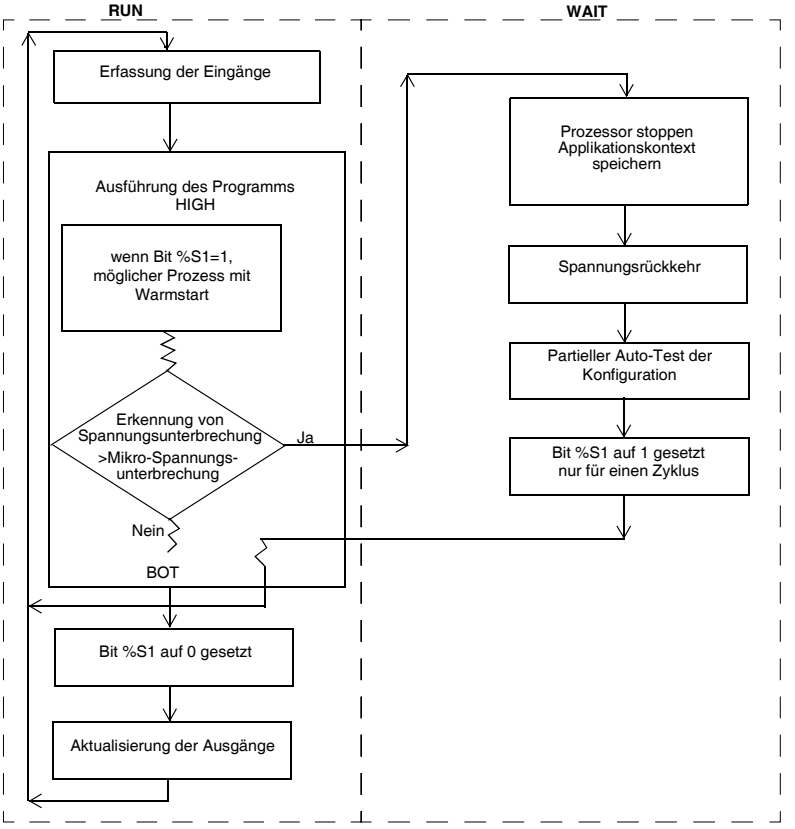
Ursache für einen Warmstart

Ein Warmstart kann verursacht werden durch:

- Spannungsrückkehr ohne Verlust des Applikationskontexts,
- Setzen des Systembits **%S1** auf 1 durch das Programm,
- Durch das Bedienterminal, wenn sich die Steuerung im Betriebszustand STOP befindet

Abbildung

Das untenstehende Schema beschreibt die Funktionsweise eines Warmstarts im Ausführungsmodus (RUN).



Neustart der Programmausführung

Die folgende Tabelle beschreibt die Neustartphasen zum Ausführen eines Programms nach einem Warmstart.

Phase	Beschreibung
1	Die Programmausführung nimmt die Bearbeitung an dem gleichen Element wieder auf, an dem sie sich vor der Spannungsunterbrechung befand, ohne die Ausgänge zu aktualisieren. **Hinweis:** Es wird nur das gleiche Element des Anwendercodes neu gestartet. Der Systemcode (z.B. die Aktualisierung der Ausgänge) wird nicht neu gestartet.
2	Am Ende des Startzyklus führt das System Folgendes aus: ● Aufheben der Reservierung der Applikation, falls sie reserviert ist (und Aufrufen einer STOP-Applikation im Falle von Debugging); ● Neuinitialisierung der Meldungen
3	Das System führt einen Warmstartzyklus durch, bei dem es Folgendes erledigt: ● Neustart des Tasks mit den Bits **%S1** (Anzeige des Warmstarts) und **%S13** (erster Zyklus im RUN-Modus) auf 1, ● Zurücksetzen der Bits**%S1** und **%S13** am Ende des ersten Zyklus des Tasks auf Status 0.

Bearbeitung eines Warmstarts

Tritt ein Warmstart auf und ein bestimmter Applikationsprozess ist erforderlich, muss das Bit **%S1** zu Beginn des Task-Zyklus getestet und das entsprechende Programm aufgerufen werden.

Ausgänge nach Spannungsausfall

Wenn ein Spannungsausfall erkannt wird, werden die Ausgänge in einen Fehlerzustand von (standardmäßig) 0 gesetzt.
Bei Spannungsrückkehr sind die Ausgänge solange im letzten Zustand, bis sie wieder vom Task aktualisiert werden.

Umgehen mit Kaltstart

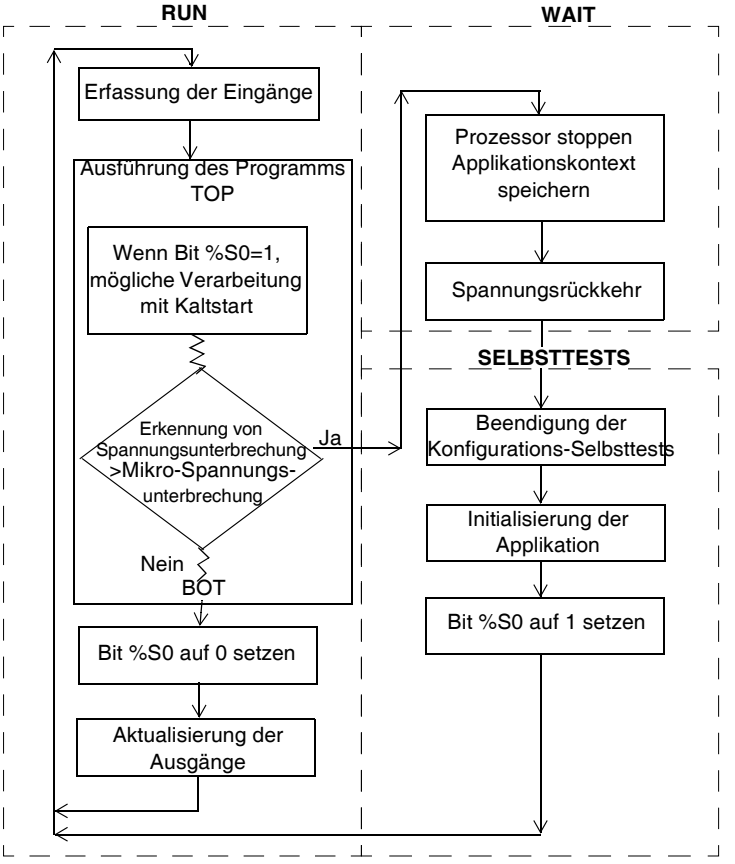
Ursache für einen Kaltstart

Ein Kaltstart kann hervorgerufen werden durch:

- Laden einer neuen Applikation in den RAM
- Spannungsrückkehr mit Verlust des Applikationskontexts
- Setzen des Bits **%S0** auf 1 durch das Programm,
- Durch das Bedienterminal, wenn sich die Steuerung im Betriebszustand STOP befindet

Abbildung

Die folgende Abbildung zeigt einen Kaltstart im Betriebszustand RUN.



Befehl

Die folgende Tabelle beschreibt die Neustartphasen zum Ausführen eines Programms nach einem Kaltstart.

Phase	Beschreibung
1	Beim Anlauf befindet sich die Steuerung im Betriebszustand RUN. Im Falle eines Neustarts nach einer Unterbrechung durch einen Fehler erzwingt das System einen Kaltstart. Die Programmausführung startet zu Beginn des Zyklus neu.
2	Das System führt Folgendes aus: ● Zurücksetzen der internen Bits und Worte und der E/A-Abbildungen auf 0, ● Initialisierung der Systembits und -worte, ● Initialisiert Funktionsblöcke mit Konfigurationsdaten
3	Beim ersten Neustartzyklus führt das System Folgendes aus: ● Neustart des Tasks mit den Bits **%S0** (Anzeige des Kaltstarts) und **%S13** (erster Zyklus im RUN-Modus) auf 1, ● Rücksetzen der Bits **%S0** und **%S13** am Ende dieses ersten Task-Zyklus auf 0 ● Setzen der Bits **%S31** und **%S38** (Ereignissteuerungsanzeigen) auf ihren Ausgangsstatus 1. ● Rücksetzen der Bits **%S39** (Anzeige für Ereignissteuerung) und des Worts **%SW48** (Anzahl der ausgeführten Ereignisse mit Ausnahme von periodischen Ereignisse).

Bearbeitung eines Kaltstarts

Wenn ein Kaltstart durchgeführt wird und eine bestimmte Verarbeitung einer Applikation erforderlich ist, so muss das Bit **%S0** (das sich im Zustand 1 befindet) während des ersten Task-Zyklus getestet werden.

Ausgänge nach Spannungs-ausfall

Wenn ein Spannungsausfall erkannt wird, werden die Ausgänge in einen Fehlerzustand von (standardmäßig) 0 gesetzt.
Bei erneuter Stromzufuhr befinden sich die Ausgänge solange auf 0, bis sie wieder durch den Task aktualisiert werden.

Initialisierung der Objekte

Einleitung

Die Steuerungen können von TwidoSoft initialisiert werden, indem die Systembits **%S0** (Kaltstart) und **%S1** (Warmstart) gesetzt werden.

Kaltstart-Initialisierung

Für eine Kaltstart-Initialisierung muss das Systembit **%S0** auf 1 gesetzt werden.

Initialisierung der Objekte (identisch mit dem Kaltstart) bei Einschaltung mittels %S0 und %S1

Für eine Initialisierung der Objekte bei Einschaltung müssen die Systembits **%S1** und **%S0** auf 1 gesetzt werden.

Das folgende Beispiel zeigt, wie Sie eine Warmstart-Initialisierung der Objekte mit Systembits programmieren.

LD %S1 Wenn %S1 = 1 (Warmstart), dann %S0 auf 1 setzen, um die Steuerung zu initialisieren.
ST %S0 Diese beiden Bits werden vom System am Ende des nächsten Zyklus auf 0 zurückgesetzt.
ST %S9 Diese bit für die Ausgänge initialsieren benutz

Hinweis: Setzen Sie %S0 nur einen Zyklus lang auf 1.

Verwaltung von Ereignistasks

5

Auf einen Blick

Auf einen Blick

In diesem Kapitel werden die Ereignistasks und ihre Ausführung in der Steuerung beschrieben.

Hinweis: Die Steuerung TWDLCAA10DRF verwaltet keine Ereignistasks.

Inhalt dieses Kapitels

Dieses Kapitel enthält die folgenden Themen:

Thema	Seite
Überblick über Ereignistasks	80
Beschreibung der verschiedenen Ereignisquellen	81
Ereignisverwaltung	82

Überblick über Ereignistasks

Einleitung

Im vorherigen Kapitel wurden die periodischen (Siehe *Periodische Abtastung, S. 66*) und zyklischen (Siehe *Zyklische Abtastung, S. 64*) Tasks beschrieben, bei denen die Objekte am Anfang und am Ende des Tasks aktualisiert werden. Ereignisquellen können Unterbrechungen dieses Tasks hervorrufen, während denen Tasks mit höherer Priorität (Ereignistasks) ausgeführt werden, um eine schnellere Aktualisierung des Objekts zu ermöglichen.
Ein Ereignistask:

- Ist ein Programmteil, der ausgeführt wird, wenn eine bestimmte Bedingung (Ereignisquelle) erfüllt ist,
- Besitzt eine höhere Priorität als das Hauptprogramm,
- Garantiert eine kurze Antwortzeit, wodurch die Antwortzeit des Systems verkürzt werden kann.

Beschreibung eines Ereignisses

Ein Ereignis besteht aus:

- Einer Ereignisquelle, die als software- oder hardwaretechnische Unterbrechungsbedingung definiert werden kann, die das Hauptprogramm unterbricht (Siehe *Beschreibung der verschiedenen Ereignisquellen, S. 81*),
- Einem Abschnitt, der eine eigenständige, mit einem Ereignis verknüpfte Programmeinheit ist,
- Einer Ereigniswarteschlange, die das Speichern der Ereignisliste bis zu deren Ausführung ermöglicht,
- Einer Priorität, welche die Ausführungsreihenfolge der Elemente angibt.

Beschreibung der verschiedenen Ereignisquellen

Beschreibung der verschiedenen Ereignisquellen

Eine Ereignisquelle muss durch die Software verwaltet werden, um zu gewährleisten, dass das Hauptprogramm ordnungsgemäß durch das Ereignis unterbrochen wird und um die mit dem Ereignis verknüpfte Programmier-Section aufzurufen. Die Applikations-Abtastzeit hat keinen Einfluss auf die Ausführung der Ereignisse.
Die folgenden neun Ereignisquellen sind erlaubt:

- Vier Bedingungen, die mit dem Grenzwert der Funktionsblöcke "Sehr schneller Zähler" verbunden sind (zwei Ereignisse pro Instanz von %VFC)
- Vier Bedingungen, die mit den physikalischen Eingängen eines Grundgeräts SPS verbunden sind.
- Eine periodische Bedingung.

Eine Ereignisquelle kann nur mit einem einzigen Ereignis verbunden sein und muss sofort von der Software TwidoSoft erkannt werden. Sofort nach der Erkennung führt die Software den mit dem Ereignis verbundenen Programmabschnitt aus: Jedes Ereignis ist einem Unterprogramm mit folgender Bezeichnung zugeordnet: **SRi:** definiert bei der Konfiguration der Ereignisquellen.

Ereignis an den physikalischen Eingängen eines Grundgeräts SPS

Die Eingänge %I0.2, %I0.3, %I0.4 und %I0.5 können als Ereignisquelle verwendet werden, wenn sie nicht gesperrt sind, und wenn die Ereignisse während der Konfiguration hierfür zugelassen wurden.
Die Ereignisverarbeitungen können durch die Eingänge 2 bis 5 eines Grundgeräts SPS (Position 0) bei steigender oder fallender Flanke ausgelöst werden.
Detaillierte Informationen über die Ereigniskonfiguration finden Sie in der Online-Hilfe "TwidoSoft Betriebshandbuch" im Abschnitt "Hardwarekonfiguration -> Eingangkonfiguration".

Ereignisse an den Ausgängen eines Funktionsblocks %VFC

Die Ausgänge TH0 und TH1 des Funktionsblocks %VFC sind Ereignisquellen. Die Ausgänge TH0 und TH1 gehen jeweils in folgenden Zustand über:

- Zustand 1, wenn der Wert über dem Grenzwert S0 und S1, liegt,
- Zustand 0, wenn der Wert unter dem Grenzwert S0 und S1. liegt.

Eine steigende oder fallende Flanke dieser Ausgänge kann eine Ereignisverarbeitung auslösen.
Detaillierte Informationen über die Ereigniskonfiguration finden Sie in der Online-Hilfe "TwidoSoft Betriebshandbuch" im Abschnitt "Softwarekonfiguration -> Sehr schnelle Zähler".

Periodisches Ereignis

Dieses Ereignis führt einen Programmabschnitt periodisch aus. Diese Task hat Vorrang vor der Mastertask.
Diese Ereignisquelle hat jedoch eine geringere Priorität als die anderen Ereignisquellen.
Die Periode dieser Task wird während der Konfiguration zwischen 5 und 255 ms festgelegt. Es kann nur ein periodisches Ereignis verwendet werden.
Detaillierte Informationen über die Ereigniskonfiguration finden Sie in der Online-Hilfe "TwidoSoft Betriebshandbuch" im Abschnitt "Konfiguration der Programmparameter -> Abtastmodus".

Ereignisverwaltung

Ereigniswarteschlange und Priorität

Die Ereignisse haben zwei mögliche Prioritäten: hoch oder niedrig. Es kann jedoch nur ein **einziger** Ereignistyp (also nur eine einzige Ereignisquelle) die Priorität "Hoch" haben. Die anderen Ereignisse haben daher die Priorität "Niedrig" und ihre Ausführungsreihenfolge ist abhängig von ihrer Erkennungsreihenfolge.
Zum Generieren der Ausführungsreichenfolge der Ereignistasks existieren zwei Ereigniswarteschlangen:

- Eine ermöglicht das Speichern von bis zu 16 Ereignissen (derselben Ereignisquelle) mit hoher Priorität,
- Die andere ermöglicht das Speichern von bis zu 16 Ereignissen (anderer Ereignisquellen) mit niedriger Priorität.

Diese Warteschlagen werden als FIFO (First In First Out) generiert: Das erste gespeicherte Ereignis wird zuerst ausgeführt. Die Warteschlangen können jedoch nur 16 Ereignisse speichern. Alle anderen Ereignisse gehen verloren.
Die Warteschlange mit niedriger Priorität wird erst dann ausgeführt, wenn die Warteschlange mit hoher Priorität leer ist.

Verwaltung von Ereigniswarteschlangen

Jedes Mal, wenn eine (mit einer Ereignisquelle verknüpfte) Unterbrechung auftritt, wird folgende Sequenz ausgeführt:

Schritt	Beschreibung
1	Verwaltung der Unterbrechung: ● Erkennung der physikalischen Unterbrechung, ● Ereignis wird in der entsprechenden Ereigniswarteschlange gespeichert, ● Prüfung, ob ein Ereignis derselben Priorität gerade ausgeführt wird (ist dies der Fall, bleibt das Ereignis in seiner Warteschlange).
2	Kontextspeicherung.
3	Ausführung des mit dem Ereignis verknüpften Programmabschnitts (Unterprogramm mit der Bezeichnung SRi:).
4	Aktualisierung der Ausgänge
5	Wiederherstellung des Kontexts

Bevor der Kontext wiederhergestellt wird, müssen alle Ereignisse der Warteschlange ausgeführt werden.

Ereignis-steuerung

Zur Steuerung der Ereignisse (Siehe *Systembits und Systemworte, S. 619*) werden Systembits und -worte verwendet.

- %S31: Ermöglicht das Ausführen oder Verzögern eines Ereignisses,
- %S38: Ermöglicht das Einreihen eines Ereignisses in die Ereigniswarteschlange,
- %S39: Ermöglicht zu sehen, ob Ereignisse verloren gegangen sind,
- %SW48: Enthält die Anzahl der Ereignisse, die seit dem letzten Kaltstart ausgeführt wurden (zählt alle Ereignisse mit Ausnahme von periodischen Ereignissen).

Der Wert von Bit %S39 und von Wort %SW48 wird bei einem Kaltstart oder nach dem Laden einer Applikation auf 0 zurückgesetzt und der Wert von %S31 und %S38 wird in den Ausgangszustand von 1 gesetzt. Bei einem Warmstart bleibt der Wert unverändert. In allen Fällen wird die Ereigniswarteschlange initialisiert.

Sonderfunktionen

II

Auf einen Blick

Inhalt dieses Abschnitts

Dieser Abschnitt beschreibt die verschiedenen Kommunikationstypen, die integrierten analogen Funktionen, die Verwaltung der analogen E/A-Module und die Installation des AS-Interface V2-Busses und des CANopen-Feldbusses für Twido-Steuerungen.

Inhalt dieses Teils

Dieser Teil enthält die folgenden Kapitel:

Kapitel	Kapitelname	Seite
6	Kommunikation	87
7	Integrierte Analogfunktionen	193
8	Verwalten von Analogmodulen	197
9	Inbetriebnahme des AS-Interface V2-Busses	211
10	Installieren und Konfigurieren des CANopen-Feldbusses	245
11	Konfigurieren des TwidoPort-Ethernet-Gateways	287
12	Betrieb des Bedienterminals	315

Kommunikation

6

Auf einen Blick

Inhalt dieses Kapitels

Dieses Kapitel bietet eine Übersicht über die Konfiguration, Programmierung und Verwaltung der Kommunikation bei Twido Steuerungen.

Inhalt dieses Kapitels

Dieses Kapitel enthält die folgenden Themen:

Thema	Seite
Überblick über die verschiedenen Kommunikationstypen	89
Kommunikation zwischen TwidoSoft und der Steuerung	91
Kommunikation zwischen TwidoSoft und einem Modem	97
Kommunikation über die dezentrale Verbindung	109
ASCII-Kommunikation	122
Modbus-Kommunikation	134
Modbus-Standard-Requests	153
Transparent Ready Implementation Class (Twido Serial A05, Ethernet A15)	159
Ethernet TCP/IP-Kommunikations-Überblick	160
TCP/IP-Konfigurationsanleitung für die Ethernet-Kommunikation zwischen PC und Steuerung	162
Anschließen Ihrer Steuerung an das Netzwerk	168
IP-Adressierung	169
Zuweisen von IP-Adressen	170
TCP/IP-Setup	174
Registerkarte "IP-Adresskonfiguration"	176
Registerkarte "Markiertes IP"	179
Registerkarte "Timeout"	181
Registerkarte "Dezentrale Geräte"	183
Anzeigen der Ethernet-Konfiguration	185
Konfiguration der Ethernet-Verbindungen	187
Ethernet-Anzeige-LEDs	189
TCP Modbus-Nachrichtenübertragung	190

Überblick über die verschiedenen Kommunikationstypen

Auf einen Blick

Twido verfügt über einen oder zwei serielle Kommunikationsports für die Kommunikation mit Steuerungen der dezentralen Peripherie, Partnersteuerungen und allgemeinen Geräten. Beide Ports, sofern verfügbar, können für alle Dienste eingesetzt werden, außer für die Kommunikation mit Twido Soft, die nur über den ersten Port aufgebaut werden kann. Bei Jeder Twido-Steuerung werden drei verschiedene Basis-Protokolle berücksichtigt: Dezentrale Verbindung, ASCII oder Modbus (Modbus-Master oder Modbus-Slave).
Außerdem bietet die kompakte Steuerung TWDLCAE40DRF einen RJ-45-Ethernet-Kommunikationsport. Er unterstützt das Modbus TCP/IP-Client/Server-Protokoll für die Kommunikation zwischen gleichrangigen Steuerungen über das Ethernet-Netzwerk.

Dezentrale Verbindung

Bei der dezentralen Verbindung handelt es sich um einen sehr schnellen Master/Slave-Bus, der zur Übertragung einer kleinen Datenmenge von der Master-Steuerung zu einer größtmöglichen Anzahl dezentraler Steuerungen (Slaves) dient. Applikations- oder E/A-Daten werden entsprechend der Konfiguration der dezentralen Steuerungen übertragen. Eine Kombination aus dezentralen Steuerungstypen ist möglich, wobei es sich bei einigen um dezentrale E/A-Steuerungen und bei anderen um SPS-Erweiterungen handeln kann.

ASCII

Das ASCII-Protokoll ist ein Halbduplex-Protokoll im einfachen Zeichenmodus, das zum Übertragen/Empfangen einer Zeichenkette zu oder von einem Peripheriegerät (Drucker oder Terminal) dient. Dieses Protokoll wird nur über die Anweisung "EXCH" verwendet.

Modbus

Das Modbus-Protokoll ist ein Master/Slave-Protokoll, bei dem nur ein einziger Master Antworten von Slaves anfordern und anhand der Anforderungen agieren darf. Der Master kann einzelne Slaves ansprechen oder er kann eine Broadcast-Meldung an alle Slaves ausgeben. Slaves senden eine Meldung (Antwort) auf Requests, die individuell an sie gerichtet worden sind, zurück. Es werden keine Antworten auf Broadcast-Requests vom Master zurückgesendet.
Modbus-Master Durch den Modbus-Master-Modus kann die Twido-Steuerung einen Modbus-Request an einen Slave schicken und auf die Antwort des Slaves warten. Der Modbus-Master-Modus wird nur über die Anweisung "EXCH" verwendet. Modbus-ASCII und Modbus-RTU werden beide im Modbus-Mastermodus unterstützt.
Modbus-Slave - Im Modbus-Slavemodus kann die Twido Steuerung auf Modbus-Anfragen eines Modbus-Master reagieren. Dies ist der voreingestellte Kommunikationsmodus, wenn keine andere Kommunikationsart eingerichtet ist. Die Twido Steuerung unterstützt die Standard-Modus-Daten- und -Steuerungsfunktionen sowie Service-Erweiterungen für den Objektzugriff. Modbus-ASCII und Modbus-RTU werden beide im Modbus-Slavemodus unterstützt.

Hinweis: 32 Geräte (ohne Repeater) mit Adressen zwischen 1 und 247 können in einem RS-485-Netzwerk installiert werden (1 Master und bis zu 31 Slaves).

Modbus TCP/IP

Hinweis: Modbus TCP/IP wird ausschließlich von kompakten Steuerungen der Serie TWDLCAE40DRF mit integrierter Ethernet-Netzwerkschnittstelle unterstützt.

Nachfolgend wird das Modbus-Applikationsprotokoll (MBAP) beschrieben.
Das Modbus-Applikationsprotokoll (MBAP) ist ein Schicht-7-Protokoll, das eine gleichrangige Kommunikation zwischen speicherprogrammierbaren Steuerungen (SPS) und anderen Knoten in einem LAN ermöglicht.
Die aktuelle Implementierung der Twido-Steuerung TWDLCAE40DRF überträgt das Modbus-Applikationsprotokoll über TCP/IP im Ethernet-Netzwerk. Bei den Modbus-Protokolltransaktionen handelt es sich üblicherweise um Request-Antwort-Nachrichtenpaare. Eine SPS kann sowohl Client als auch Server sein. Dies ist davon abhängig, ob sie nach Nachrichten fragt oder diese beantwortet.

Kommunikation zwischen TwidoSoft und der Steuerung

Auf einen Blick

Jede Twido-Steuerung verfügt an Port 1 über einen integrierten EIA RS-485-Anschluss. Dieser Anschluss verfügt über eine interne Stromversorgung. Der Port 1 muss für die Kommunikation mit der TwidoSoft-Programmiersoftware verwendet werden.
Kein optionales Speichermodul und kein Kommunikationsmodul dürfen diesen Port verwenden. Ein Modem darf jedoch diesen Port nutzen.
Es gibt verschiedene Möglichkeiten zur Verbindung des PC mit dem RS-485-Port 1 der Twido-Steuerung:

- per TSXPCX-Kabel
- per Telefonleitung: MODEM-Verbindung

Außerdem verfügt die kompakte Steuerung TWDLCAE40DRF über einen integrierten RJ-45-Ethernet-Netzwerkverbindungs-Port, der für die Kommunikation mit dem Ethernet-fähigen PC, auf dem die Programmiersoftware TwidoSoft ausgeführt wird, verwendet werden kann.
Es gibt zwei Möglichkeiten für den Ethernet-fähigen PC, mit dem RJ-45-Port der Twido-Steuerung TWDLCAE40DRF zu kommunizieren:

- Über eine direkte Kabelverbindung über ein UTP Cat5 RJ45-Ethernet-Überbrückungskabel (nicht empfohlen),
- über einen Anschluss an das Ethernet-Netzwerk über ein SFTP Cat5 RJ45-Ethernet-Kabel, das im Schneider Electric-Katalog verfügbar ist (Kabel-Teilenummer: 490NTW000••).

⚠ ACHTUNG

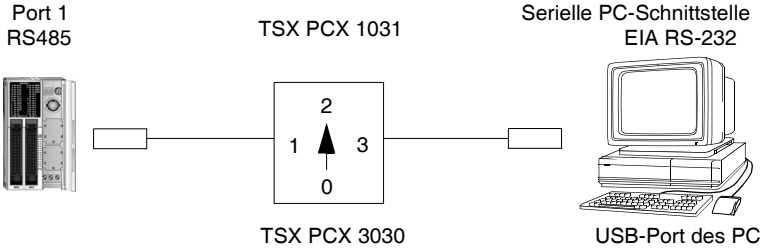
BESCHÄDIGUNG VON GERÄTEN

TwidoSoft erkennt das Trennen der Verbindung möglicherweise nicht, wenn Sie das TSX PCX 1031-, TSX PCX 3030- oder Ethernet-Kommunikationskabel physikalisch von einer Steuerung entfernen und schnell in eine andere Steuerung stecken. Um dies zu verhindern, müssen Sie die Verbindung mit TwidoSoft trennen, bevor Sie das Kabel umstecken.

Die Nichtbeachtung dieser Anweisung kann Körperverletzung oder Materialschäden zur Folge haben!

Anschluss des TSXPCX-Kabels

Der EIA RS-232C- oder USB-Port Ihres PC wird über das Multifunktions-Kommunikationskabel TSX PCX 1031 oder TSX PCX 3030 mit Port 1 der Steuerung verbunden. Dieses Kabel, das die Umwandlung der Signale zwischen EIA RS-232 und EIA RS-485 für das TSX PCX 1031 bzw. zwischen USB und EIA RS-485 für das TSX PCX 3030 gewährleistet, ist mit einem Drehschalter mit 4 Stellungen ausgestattet, an dem Sie verschiedene Betriebsarten einstellen können. Die vier Stellungen des Schalters sind bezeichnet mit "0 bis 3" und die entsprechende Einstellung für die Verbindung zwischen TwidoSoft und einer Twido Steuerung ist die Einstellung 2.
Diese Verbindung ist in der folgenden Abbildung dargestellt.



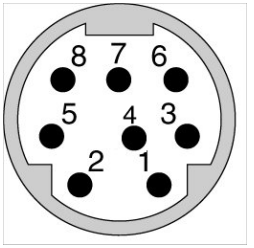
Hinweis: Bei diesem Kabel wird das DPT-Signal an Pin 5 nicht auf 0 V gesetzt. Dies zeigt der Steuerung, dass die aktuelle Verbindung eine TwidoSoft-Verbindung ist. Das Signal wird intern eingestellt, um der Firmware anzuzeigen, dass es sich bei der aktuellen Verbindung um eine TwidoSoft-Verbindung handelt.

Anschlussbelegung von Stecker und Buchse

Die folgende Abbildung zeigt die Anschlussbelegung eines 8-poligen miniDIN-Steckers und einer Klemmenleiste.

Mini DIN

TWD NAC232D, TWD NAC485D
TWD NOZ485D, TWD NOZ232D

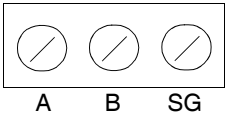


Pin	Base RS485	RS485 Option	RS232-C
1	D1 (A+)	D1 (A+)	RTS
2	D0 (B-)	D0 (B-)	DTR
3	NC	NC	TXD
4	/DE	NC	RXD
5	/DPT	NC	DSR
6	NC	NC	GND
7	0 V	0 V	GND
8	5 V	5 V	5 V

Bemerkung: Maximaler Gesamtverbrauch für 5 V-Modus (Pin 8): 180 mA

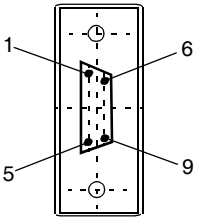
Klemmenleiste

TWD NAC485T
TWD NOZ485T



Pin	RS485
A	D1 (A+)
B	D0 (B-)
SG	0V

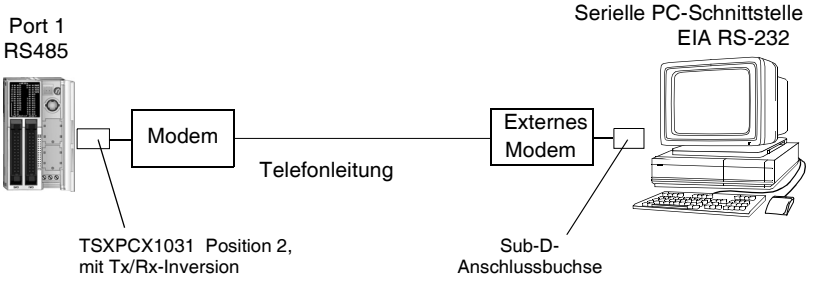
Die folgende Abbildung zeigt die Anschlussbelegung einer 9-poligen SubD-Buchse für das TSX PCX 1031.



Pin	RS232
1	DCD
2	RX
3	TX
4	DTR
5	SG
6	NC
7	RTS
8	CTS
9	NC

Verbindung per Telefonleitung

Eine Modem (Siehe *Kommunikation zwischen TwidoSoft und einem Modem, S. 97*)-Verbindung ermöglicht die Programmierung einer Steuerung sowie die Kommunikation mit einer Steuerung per Telefonleitung.
Das der Steuerung zugewiesene Modem ist ein **Empfangsmodem**, das an Port 1 der Steuerung angeschlossen wird. Das dem PC zugewiesene Modem kann ein internes bzw. externes, an den seriellen COM-Port angeschlossenes Modem sein. Diese Verbindung ist in der folgenden Abbildung dargestellt.



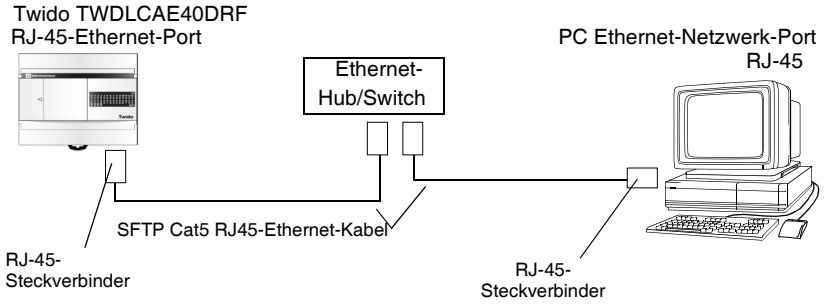
Hinweis: Es kann nur ein einziges Modem an den Port 1 der Steuerung angeschlossen werden.

Hinweis: Vorsicht. Vergessen Sie nicht, die mit dem Modem mitgelieferte Software zu installieren, da TwidoSoft nur die installierten Modems berücksichtigt.

Ethernet-Netzwerkverbindung

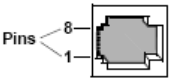
Hinweis: Obwohl eine direkte Kabelverbindung (mittels eines Ethernet-Überbrückungskabels) zwischen der Twido-Steuerung TWDLCAE40DRF und dem PC, auf dem die Programmiersoftware TwidoSoft ausgeführt wird, wird von dieser Möglichkeit abgeraten. Sie sollten statt dessen eine Verbindung über einen Ethernet-Netzwerk-Hub/Switch vorziehen.

Die folgende Abbildung zeigt eine Verbindung zwischen dem PC und einer Twido-Steuerung über einen Ethernet-Netzwerk-Hub/Switch.



Hinweis: Der PC, auf dem die TwidoSoft-Applikation ausgeführt wird, muss Ethernet-fähig sein.

Die Twido-Steuerung TWDLCAE40DRF ist mit einem RJ-45-Stecker für den Anschluss an das 100 BASE-TX Ethernet-Netzwerk mit automatischer Einstellung ausgestattet. Er unterstützt eine Netzwerk-Übertragungsrate sowohl von 100 Mbps als auch von 10 Mbps.
Die folgende Abbildung zeigt den RJ-45-Stecker der Twido-Steuerung.



Die acht Pins des RJ-45-Steckers sind vertikal angeordnet und von oben nach unten nummeriert. Die Anschlussbelegung des RJ-45-Steckers ist in der folgenden Tabelle beschrieben.

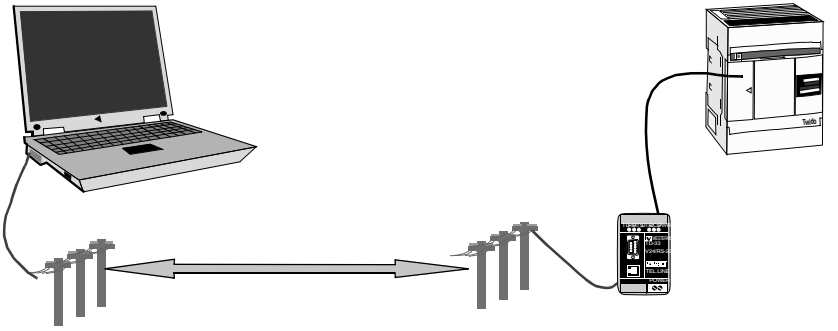
Anschlussbelegung	Funktion	Polarität
8	NC	
7	NC	
6	RxD	(-)
5	NC	
4	NC	
3	RxD	(+)
2	TxD	(-)
1	TxD	(+)

Hinweis:
- Für 10Base-T und 100Base-TX werden derselbe Stecker und dieselbe Anschlussbelegung verwendet.
- Bei Anschluss der Twido-Steuerung an ein 100Base-TX-Netzwerk sollten Sie mindestens ein Ethernet-Kabel der Kategorie 5 verwenden.

Kommunikation zwischen TwidoSoft und einem Modem

Allgemeines

Es ist möglich, einen PC, auf dem TwidoSoft ausgeführt wird, mit einer Twido-Steuerung zu verbinden, um Applikationen zu übertragen, Objekte zu animieren und um Befehle im Bedienermodus auszuführen. Es ist außerdem möglich, eine Twido-Steuerung an andere Geräte wie etwa eine andere Twido-Steuerung anzuschließen, um eine Kommunikation mit dem Applikationsprozess aufzubauen.

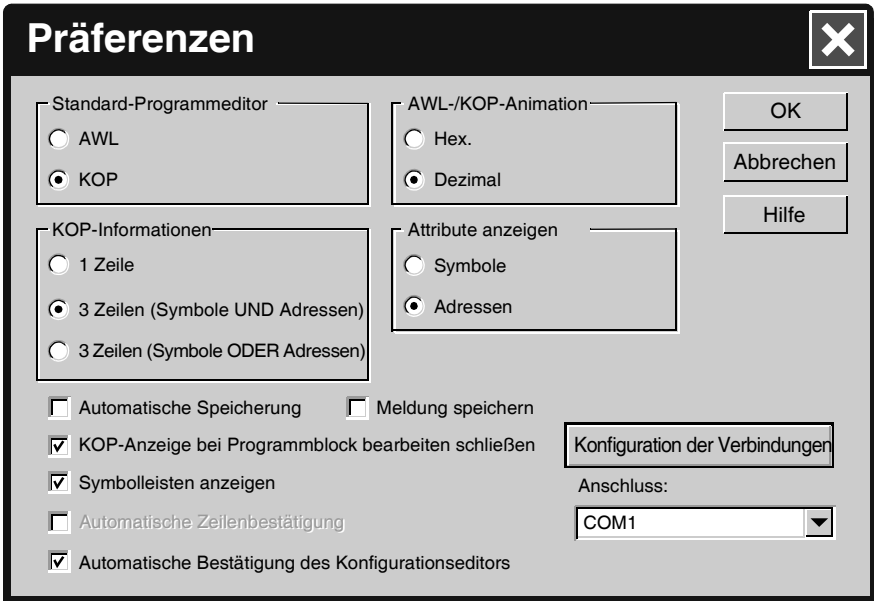


Installation des Modems

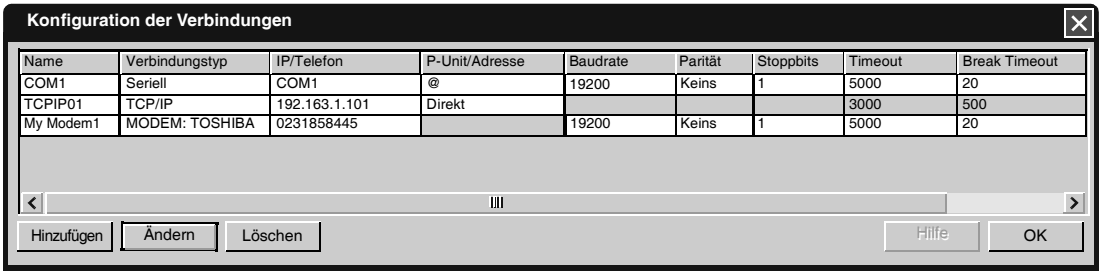
Alle Modems, die der Benutzer in Verbindung mit TwidoSoft verwenden möchte, müssen unter dem Betriebssystem Windows ausgehend von Ihrem PC installiert werden.
Anweisungen zur Installation Ihrer Modems unter Windows entnehmen Sie bitte der Windows-Dokumentation.
Diese Installation erfolgt unabhängig von TwidoSoft.

Herstellung der Verbindung

Die standardmäßige Kommunikationsverbindung zwischen TwidoSoft und der Twido-Steuerung erfolgt über einen seriellen Kommunikationsport mittels des Kabels TSX PCX 1031 und eines Kreuzadapters (siehe *Anhang 1, S. 106*).
Wenn ein Modem für die Verbindung zum PC verwendet wird, muss dies in der Software TwidoSoft angegeben werden.
Um eine Verbindung mit TwidoSoft auszuwählen, rufen Sie im Menü "Datei" die Option "Präferenzen" auf.



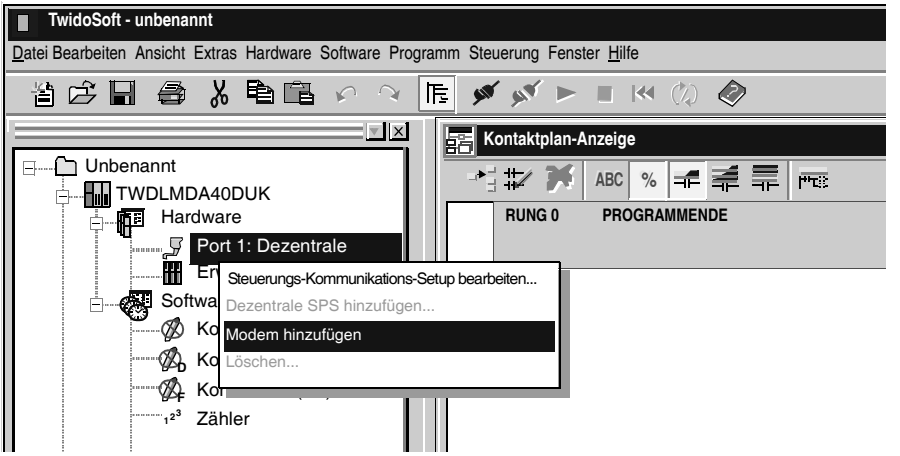
Das Dialogfeld ermöglicht die Auswahl einer Verbindung oder die Verwaltung der Verbindungen (Erstellung, Änderung etc.).
Um eine bereits bestehende Verbindung zu verwenden, wählen Sie eine Verbindung unter den im Pull-down-Menü aufgeführten Verbindungen aus.
Wenn Sie eine Verbindung hinzufügen, ändern oder löschen möchten, klicken Sie einmal auf "Konfiguration der Verbindungen". Daraufhin wird ein Fenster geöffnet, in dem die Liste der Verbindungen sowie deren Eigenschaften angezeigt werden.



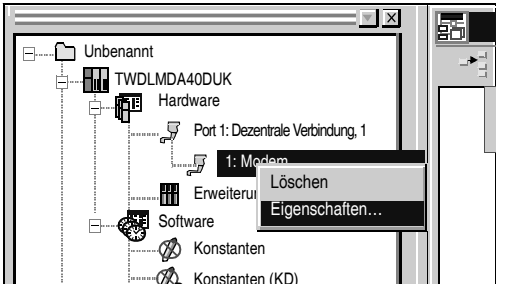
In diesem Fall sind 2 serielle Ports (COM1 und COM4) und eine Modemverbindung aufgeführt. Bei dem Modem handelt es sich um das Modell TOSHIBA V.90, das für die Anwahl der folgenden Rufnummer konfiguriert ist: 0231858445 (Inlandsgespräch).
Sie können den Namen jeder Verbindung ändern. Dieser Name dient der Verwaltung der Applikation. Die Änderung von COM1 oder COM4 ist jedoch unzulässig.
Auf diese Weise legen Sie die Verbindung fest bzw. wählen sie aus, die Sie für die Verbindung Ihres PC mit einem Modem verwenden möchten.
Dies ist jedoch nur ein Teil der Vorgänge, die von Ihnen für die Herstellung der globalen Verbindung zwischen dem Rechner und der Twido-Steuerung ausgeführt werden müssen.
Der nächste Schritt betrifft die Twido-Steuerung. Die dezentrale Twido-Steuerung muss mit einem Modem verbunden werden.
Alle Modems müssen für den Aufbau einer Verbindung initialisiert werden. Die Twido-Steuerung, die mindestens die Firmware-Version V2.0 enthalten muss, ist in der Lage, bei Einschaltung eine geeignete Zeichenkette an das Modem zu senden, wenn das Modem in der Applikation konfiguriert ist.

Konfiguration des Modems

Gehen Sie folgendermaßen vor, um ein Modem in einer Twido-Steuerung zu konfigurieren:



Nach der Konfiguration des Modems an Port 1 müssen die Eigenschaften festgelegt werden. Nach Anklicken mit der rechten Maustaste werden die Menüoptionen "Löschen" und "Eigenschaften" angezeigt. Die Option "Eigenschaften" ermöglicht die Auswahl eines bekannten Modems, die Erstellung eines neuen Modems oder die Änderung eines vorhandenen Modems.



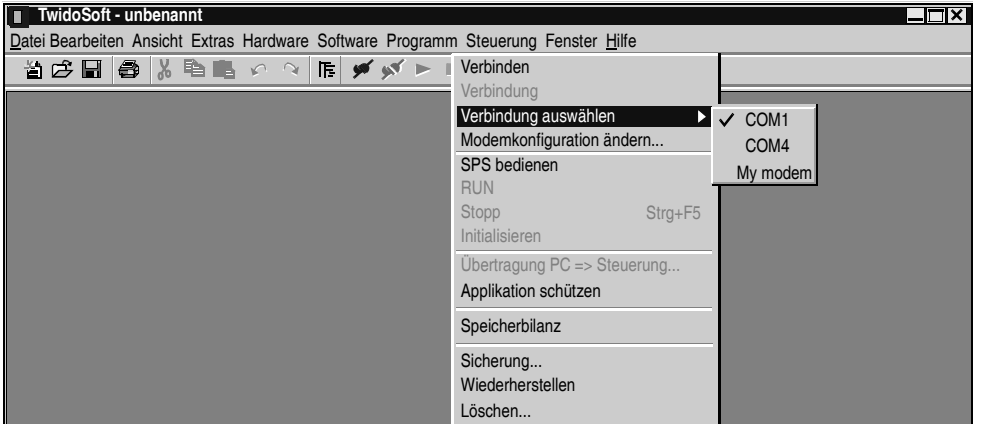
Hinweis: Die Verwaltung des Modems mittels der Twido-Steuerung erfolgt vollständig über Port 1. Das bedeutet, dass Sie ein Modem an den Kommunikationsport 2 anschließen können, aber dass in diesem Fall alle Betriebsarten und die Initialisierungssequenz des Modems manuell ausgeführt werden müssen und nicht auf gleiche Weise wie am Kommunikationsport 1 ausgeführt werden können.

Wählen Sie als nächstes "Eigenschaften" aus.

Nun können Sie ein zuvor definiertes Modem auswählen oder ein neues Modem erstellen, indem Sie auf die Schaltfläche "..." klicken.

Weisen Sie dem neuen Profil anschließend einen Namen zu, und geben Sie den Hayes-Initialisierungsbefehl wie in der Dokumentation des Modems beschrieben ein.
In der Abbildung oben steht "xxxxxx" für die Initialisierungssequenz, die Sie eingeben müssen, um das Modem für die geeignete Kommunikation vorzubereiten, d.h. Baudrate, Parität, Stoppbit, Empfangsmodus etc.
Die entsprechende Sequenz entnehmen Sie bitte der Dokumentation Ihres Modems. Die maximale Länge der Zeichenkette beträgt 127 Zeichen.
Übertragen Sie Ihre Applikation mittels einer "Punkt-zu-Punkt-Verbindung", wenn Ihre Applikation fertig gestellt ist oder wenn zumindest der Kommunikationsport 1 vollständig beschrieben ist.
Dann ist die Twido-Steuerung bereit, um mittels eines Modems mit einem PC verbunden zu werden, auf dem TwidoSoft ausgeführt wird.

Verbindungs-sequenz Bauen Sie nach der Vorbereitung von TwidoSoft und der Twido-Steuerung die Verbindung wie folgt auf:

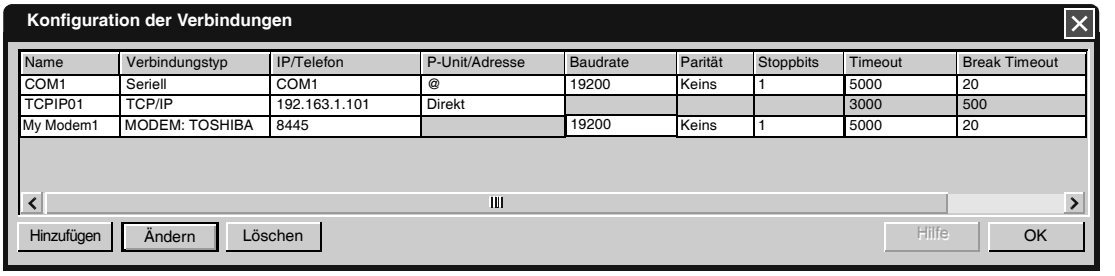
Schritt	Aktion
1	Schalten Sie die Twido-Steuerung und das Modem ein.
2	Schalten Sie Ihren Computer ein und starten Sie TwidoSoft.
3	Wählen Sie im Menü "Steuerung" die Option "Verbindung auswählen" und anschließend die Option "My modem" (oder den Namen, den Sie Ihrer Modemverbindung zugewiesen haben (siehe "Erstellen einer Verbindung")) aus. 
4	Bauen Sie eine Verbindung zu TwidoSoft auf.

Hinweis: Wenn Sie immer Ihre Modemverbindung verwenden möchten, wählen Sie im Menü "Datei", "Präferenzen" die Option "My modem" (oder den dieser Verbindung zugewiesenen Namen) aus. Auf diese Weise speichert TwidoSoft diese Einstellung.

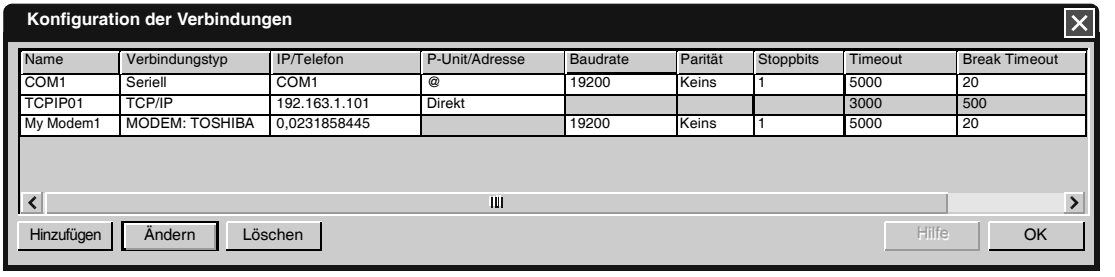
Betriebsarten Die Twido-Steuerung sendet die Initialisierungs-Zeichenkette an das angeschlossene, eingeschaltete Modem. Wenn ein Modem in der Twido-Applikation konfiguriert ist, sendet die Steuerung zunächst einen "FF"-Befehl, um zu ermitteln, ob das Modem angeschlossen ist. Wenn die Steuerung eine Antwort empfängt, wird die Initialisierungs-Zeichenkette an das Modem gesendet.

Interner, externer und internationaler Anruf

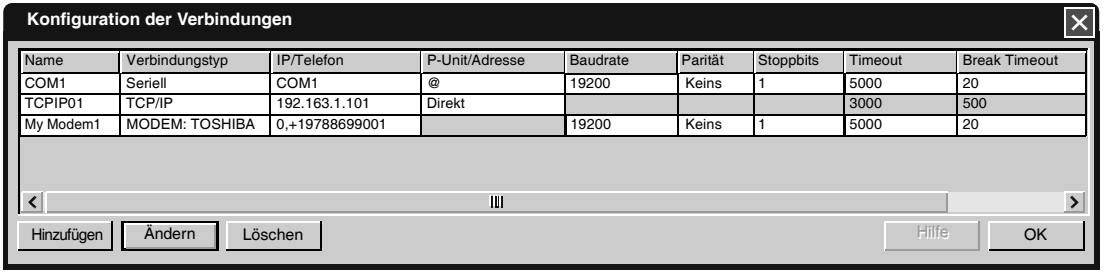
Wenn Sie mit einer Twido-Steuerung innerhalb Ihres Unternehmens kommunizieren, können Sie die Durchwahl der von Ihnen anzuwählenden Leitung verwenden. Bsp.: 8445



Wenn Sie einen internen Standard für die Wahl von Telefonnummern außerhalb Ihres Unternehmens verwenden (Amtsfreigabe) und hierzu eine "0" oder eine "9" vorwählen müssen, so verwenden Sie folgende Syntax: 0,0231858445 oder 9,0231858445

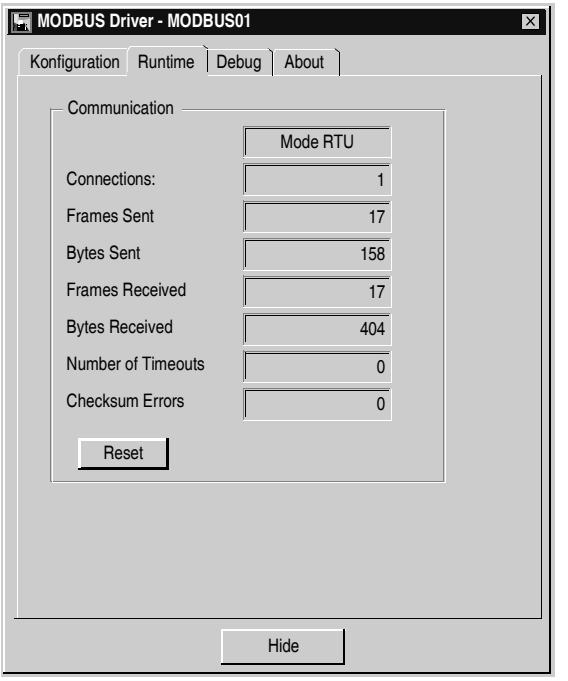


Für internationale Anrufe lautet die Syntax beispielsweise +19788699001. Bei Verwendung einer Vorwahl für die Amtsfreigabe lautet die Syntax: 0,+ 19788699001



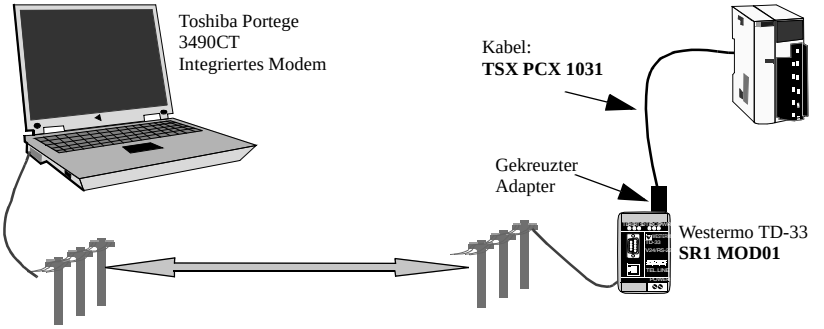
Häufig gestellte Fragen

Wenn die Verbindung bereits seit einigen Minuten aufgebaut ist, können bestimmte Kommunikationsfehler auftreten. In diesem Fall müssen Sie die Kommunikationsparameter anpassen.
TwidoSoft verwendet einen Modbus-Treiber für die Kommunikation über die seriellen Ports oder über interne Modems. Wenn die Verbindung aufgebaut ist, wird der Modbus-Treiber in der Symbolleiste angezeigt. Doppelklicken Sie auf das Symbol des Modbus-Treibers, um das Fenster zu öffnen. Nun können Sie auf die Parameter des Modbus-Treibers zugreifen. Die Registerkarte "Runtime" enthält Informationen über die mit der dezentralen Steuerung ausgetauschten Frames. Wenn die "Anzahl der Timeouts" zunimmt oder anders als 0 lautet, ändern Sie den Wert mittels der Option "Konfiguration der Verbindungen", die Sie unter TwidoSoft im Menü "Datei", "Präferenzen" aufrufen können. Klicken Sie auf das Feld "Timeout" und anschließend auf die Schaltfläche zur Änderung des Werts und geben Sie einen höheren Wert ein. Die Standardeinstellung ist 5000 Millisekunden.
Bauen Sie die Verbindung anschließend erneut auf. Passen Sie den Wert solange an, bis Ihre Verbindung stabil ist.



Beispiele

- **Beispiel 1**: TwidoSoft, verbunden mit einer TWD LMDA 20DRT (Windows 98 SE)
  - PC: Toshiba Portege 3490CT unter Windows 98
  - Modem (PC-intern): internes Toshiba V.90-Modem
  - Twido-Steuerung: TWD LMDA 20DRT, Version 2.0
  - Modem (mit der Twido-Steuerung verbunden): Typ Westermo TD-33 / V.90, Artikelnummer SR1 MOD01, aufgeführt im neuen Twido-Katalog (September 03) (siehe *Anhang 2:, S. 107*)
    (**Nur für Kunden in Nordamerika:** Der in Ihrer Region verfügbare Modemtyp ist TD-33/V.90 US),
  - Kabel: TSX PCX 1031, angeschlossen an den Twido-Kommunikationsport 1, und ein Adapter: 9-poliger Steckverbinder/9-polige Steckbuchse für die Rx/Tx-Kreuzung während der Verbindung zwischen dem Westermo-Modem und der Twido-Steuerung (siehe *Anhang 1, S. 106*). Sie können auch das Kabel TSX PCX 1130 verwenden (RS485/232-Konvertierung und Rx/Tx-Kreuzung).

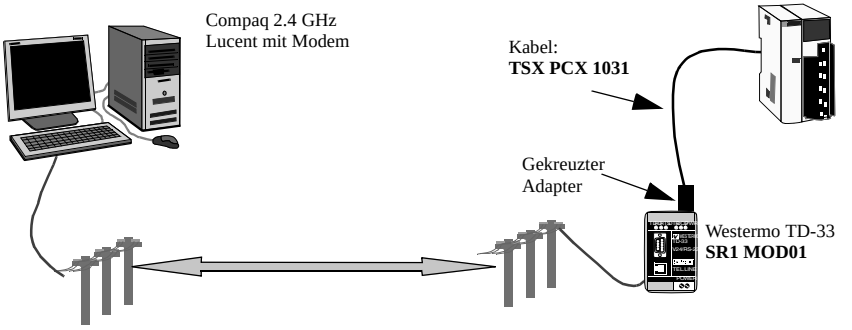


Für einen ersten Test werden 2 unternehmensinterne analoge Telefonleitungen verwendet, die nicht die vollständige Telefonnummer, sondern lediglich die Durchwahl verwenden (daher umfasst die Telefonnummer des internen Toshiba V.90-Modems nur 4 Ziffern).
Für diesen Test werden die Standard-Verbindungsparameter (TwidoSoft, Menü "Präferenzen", "Konfiguration der Verbindungen") verwendet. Diese lauten: Timeout = 5000 und Break Timeout = 20.

- **Beispiel 2**: TwidoSoft, verbunden mit einer TWD LMDA 20DRT (Windows XP Pro)
  - PC: Compaq Pentium 4, 2,4 GHz
  - Modem: Lucent Win Modem, PCI-Bus
  - Twido-Steuerung: TWD LMDA 20DRT, Version 2.0
  - Modem (mit der Twido-Steuerung verbunden): Typ WESTERMO TD-33 / V.90, Artikelnummer SR1 MOD01, aufgeführt im neuen Twido-Katalog (September 03) (siehe *Anhang 2:, S. 107*)

(**Nur für Kunden in Nordamerika:** Der in Ihrer Region verfügbare Modemtyp ist TD-33/V.90 US),

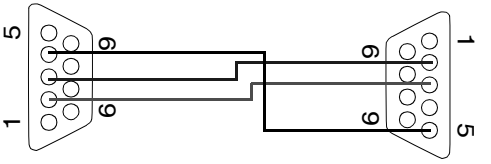
- Kabel: TSX PCX 1031, angeschlossen an den Twido-Kommunikationsport 1, und ein Adapter: 9-poliger Steckverbinder/9-polige Steckbuchse für die Rx/Tx-Kreuzung während der Verbindung zwischen dem Westermo-Modem und der Twido-Steuerung (siehe *Anhang 1, S. 106*). Sie können auch das Kabel TSX PCX 1130 verwenden (RS485/232-Konvertierung und Rx/Tx-Kreuzung).



Für den Test werden zwei unternehmensinterne analoge Telefonleitungen verwendet, die nicht die vollständige Telefonnummer, sondern lediglich die Durchwahl verwenden (daher umfasst die Telefonnummer des internen Toshiba V.90-Modems nur 4 Ziffern).
Für diesen Test werden die Standard-Verbindungsparameter (TwidoSoft, Menü "Präferenzen", "Konfiguration der Verbindungen") verwendet. Diese lauten: Timeout = 5000 und Break Timeout = 20.

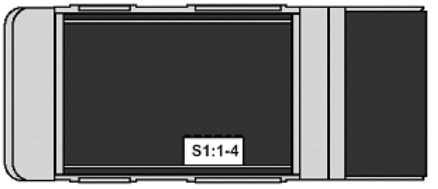
Anhang 1

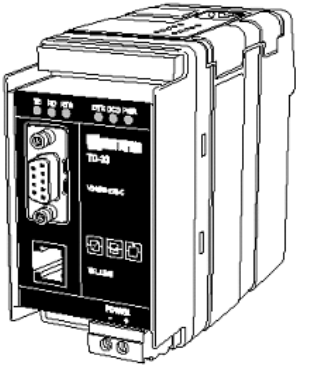
Gekreuzter Adapter für das Kabel TSX PCX 1031 und das Westermo-Modem TD-33:



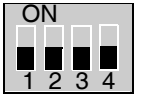
Anhang 2:

Westermo-Modem TD-33, Schneider-Artikelnummer **SR1 MOD01**(1). Dieses Modem verwaltet 4 DIP-Schalter, die sich alle in der Stellung **OFF** befinden müssen:





Werkseitige Einstellungen



Verwenden Sie die gespeicherte Konfiguration (Baudrate & Format etc.)
Deaktivieren Sie DTR Hotcall, Auto Band

Hinweis:

1. Bestimmte Produkte sind möglicherweise nicht in allen Regionen kompatibel und/oder verfügbar. Bitte erfragen Sie die Verfügbarkeit bei Ihrem Schneider-Händler.

Anhang 3:

Wavecom-Modem WMOD2B, Schneider-Artikelnummer **SR1 MOD02**(1) Dual-Band (900/1800 Hz):

Hinweis:
1. Bestimmte Produkte sind möglicherweise nicht in allen Regionen kompatibel und/oder verfügbar. Bitte erfragen Sie die Verfügbarkeit bei Ihrem Schneider-Händler.

Anhang 4:

Artikelnummern der in diesem Dokument verwendeten Produkte:

- Twido-Produkt: TWD LMDA 20DRT
- TwidoSoft-Software: TWD SPU 1002 V10M
- Kabel TSX PCX 1031
- Kabel TSX PCX 1130
- RTU-Modem: Westermo TD-33 / V90 **SR1 MOD01**(1),
- GSM-Modem: Wavecom WMOD2B **SR1 MOD02**(1).

Hinweis:
1. Bestimmte Produkte sind möglicherweise nicht in allen Regionen kompatibel und/oder verfügbar. Bitte erfragen Sie die Verfügbarkeit bei Ihrem Schneider-Händler.

Kommunikation über die dezentrale Verbindung

Einführung

Die dezentrale Verbindung ist ein Hochgeschwindigkeits-Master/Slave-Bus, der für die Übertragung geringer Datenmengen zwischen Master-Steuerung und maximal sieben dezentralen (Slave-)Steuerungen ausgelegt ist. Applikations- oder E/A-Daten werden entsprechend der Konfiguration der dezentralen Steuerungen übertragen. Eine Kombination aus dezentralen Steuerungstypen ist möglich, wobei es sich bei einigen um dezentrale E/A-Steuerungen und bei anderen um SPS-Erweiterungen handeln kann.

Hinweis: Die Master-Steuerung enthält Informationen bezüglich der Adresse einer dezentralen E/A. Die Steuerung erkennt jedoch nicht, welche bestimmte Steuerung sich an der Adresse befindet. Deshalb kann der Master nicht prüfen, ob alle dezentralen Eingänge und Ausgänge, die in der Benutzerapplikation verwendet werden, tatsächlich vorhanden sind. Achten Sie darauf, dass diese dezentralen Eingänge und Ausgänge tatsächlich vorhanden sind.

Hinweis: Der dezentrale E/A-Bus und das verwendete Protokoll sind proprietär, kein Drittgerät ist im Netzwerk zugelassen.

⚠ ACHTUNG

UNERWARTETER SYSTEMVORGANG

- Achten Sie darauf, dass es in einer dezentralen Verbindung nur einen Master gibt und dass jeder Slave über eine eindeutige Adresse verfügt. Bei Missachtung dieser Vorsichtsmaßnahme können Daten verändert oder unerwartete und mehrdeutige Ergebnisse hervorgerufen werden.
- Achten Sie darauf, dass alle Slaves über eindeutige Adressen verfügen. Slave-Adressen dürfen nicht doppelt vergeben werden. Bei Missachtung dieser Vorsichtsmaßnahme können Daten verändert oder unerwartete und mehrdeutige Ergebnisse hervorgerufen werden.

Die Nichtbeachtung dieser Anweisung kann Körperverletzung oder Materialschäden zur Folge haben!

Hinweis: Die dezentrale Verbindung benötigt einen EIA RS-485-Anschluss und kann nur an einer Kommunikationsschnittstelle gleichzeitig betrieben werden.

Hardwarekonfiguration

Die dezentrale Verbindung benötigt eine mindestens dreiadrige EIA RS-485-Schnittstelle. Die dezentrale Verbindung kann so konfiguriert werden, dass entweder die erste oder, sofern vorhanden, eine zweite optionale Schnittstelle verwendet werden kann.

Hinweis: Es darf nur eine Kommunikationsschnittstelle zur Zeit als dezentrale Verbindung konfiguriert sein.

Die folgende Tabelle führt die Geräte auf, die verwendet werden können.

Dezentral	Port	Kenndaten
TWDLC•A10/16/24DRF, TWDLCA•40DRF, TWDLMDA20/40DUK, TWDLMDA20/40DTK, TWDLMDA20DRT	1	Basissteuerung, ausgestattet mit einem dreiadrigen EIA RS-485-Port mit miniDIN-Stecker.
TWDNOZ485D	2	Kommunikationsmodul, ausgestattet mit einem dreiadrigen EIA RS-485-Port mit miniDIN-Steckverbinder. **Hinweis:** Dieses Modul ist nur für die modularen Steuerungen erhältlich. Wenn das Modul angeschlossen ist, kann die Steuerung kein Bedienterminal als Erweiterungsmodul haben.
TWDNOZ485T	2	Kommunikationsmodul, ausgestattet mit einem dreiadrigen EIA RS-485-Port mit Anschlussklemme. **Hinweis:** Dieses Modul ist nur für die modularen Steuerungen erhältlich. Wenn das Modul angeschlossen ist, kann die Steuerung kein Bedienterminal als Erweiterungsmodul haben.
TWDNAC485D	2	Kommunikations-Adapter, ausgestattet mit einem dreiadrigen EIA RS-485-Port mit miniDIN-Steckverbinder. **Hinweis:** Dieser Adapter ist nur für die kompakten Steuerungen mit 16, 24 und 40 E/A und für das Bedienterminal-Erweiterungsmodul verfügbar.
TWDNAC485T	2	Kommunikations-Adapter, ausgestattet mit einem dreiadrigen EIA RS-485-Port mit Anschlussklemme. **Hinweis:** Dieser Adapter ist nur für die kompakten Steuerungen mit 16, 24 und 40 E/A und für das Bedienterminal-Erweiterungsmodul verfügbar.
TWDXCPODM	2	Operator-Anzeige-Erweiterungsmodul, ausgestattet mit einem dreiadrigen EIA RS-485-Port mit miniDIN-Stecker oder einem dreiadrigen EIA RS-485-Port mit Anschlussklemme. **Hinweis:** Dieses Modul ist nur für die modularen Steuerungen erhältlich. Wenn das Modul angeschlossen ist, kann die Steuerung kein Kommunikationserweiterungsmodul haben.

Hinweis: Die Überprüfung des Vorhandenseins von Port 2 und dessen Konfiguration (RS232 oder RS 485) wird nur beim Einschalten oder bei der Reinitialisierung durch die Firmware der Steuerung durchgeführt.

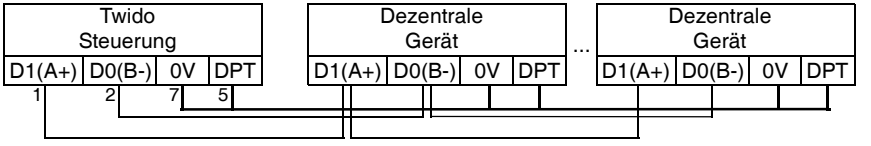
Kabelverbindung zu allen Geräten

Hinweis: Das DPT-Signal von Pin 5 muss an Pin 7 mit dem Nullleiter verbunden werden, damit die Verwendung der Kommunikation über die dezentrale Verbindung kenntlich gemacht wird. Ist dieses Signal nicht mit Erde verbunden, versucht die Twido-Steuerung entweder als Master oder als Slave standardmäßig die Kommunikation zu TwidoSoft aufzubauen.

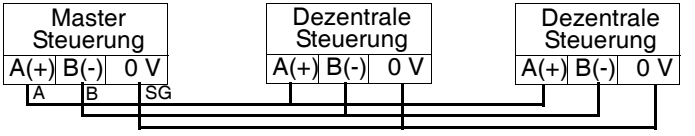
Hinweis: Die Verbindung DPT-0V ist nur erforderlich, wenn an Schnittstelle 1 eine Basissteuerung angeschlossen ist.

Die Kabelverbindungen zu allen Geräten sind im Folgenden dargestellt.

miniDIN-Anschluss



Klemmenleistenanschluss



Softwarekonfiguration

In der dezentralen Verbindung darf nur eine Master-Steuerung definiert sein. Außerdem muss jede dezentrale Steuerung über eine eindeutige Slave-Adresse verfügen. Mehrere Master oder Slaves, die identische Adressen verwenden, können bewirken, dass Übertragungen verändert werden oder Mehrdeutigkeiten entstehen.

⚠ ACHTUNG

UNERWARTETES GERÄTEVERHALTEN

Achten Sie darauf, dass es in einer dezentralen Verbindung nur einen Master gibt und dass jeder Slave über eine eindeutige Adresse verfügt. Bei Missachtung dieser Vorsichtsmaßnahme können Daten verändert oder unerwartete und mehrdeutige Ergebnisse hervorgerufen werden.

Die Nichtbeachtung dieser Anweisung kann Körperverletzung oder Materialschäden zur Folge haben!

Konfiguration der Master-Steuerung

Die Master-Steuerung wird mit TwidoSoft konfiguriert, damit sie ein dezentrales Verbindungsnetzwerk mit bis zu sieben dezentralen Steuerungen verwaltet. Diese sieben dezentralen Steuerungen können als dezentrale E/A oder als SPS-Erweiterungen konfiguriert werden. Die Adresse des Master wird mit TwidoSoft auf die Adresse 0 eingestellt.
Um eine Steuerung als Master zu konfigurieren, müssen Sie mithilfe von TwidoSoft den Port 1 oder den Port 2 als dezentrale Verbindung konfigurieren und die Adresse 0 (Master) auswählen.
Dann konfigurieren Sie die Slave-Steuerungen mittels des Fensters "Dezentrale SPS hinzufügen" entweder als dezentrale E/A oder als SPS-Erweiterung und legen ihre Adresse fest.

Konfiguration der dezentralen Steuerung

Die Konfiguration einer dezentralen Steuerung erfolgt mithilfe von TwidoSoft. Konfigurieren Sie den Port 1 oder den Port 2 als dezentrale Verbindung und weisen Sie ihm eine Adresse zwischen 1 und 7 zu.
Die folgende Tabelle bietet einen Überblick über die Unterschiede und Bedingungen der verschiedenen Konfigurationstypen für dezentrale Steuerungen.

Typ	Applikationsprogramm	Datenzugriff
Dezentrale E/A	Nein Nicht einmal eine einfache Anweisung "END" Der RUN-Modus ist vom Modus des Masters abhängig.	%I und %Q Es sind nur lokale E/A der dezentralen Steuerung zugänglich (und nicht die E/A-Erweiterung).
SPS-Erweiterung	Ja Der RUN-Modus ist nicht an den Modus des Masters gekoppelt.	%INW und %QNW Es können maximal 4 Eingangswörter und 4 Ausgangswörter zu und von jeder Erweiterung übertragen werden.

Zyklussynchronisierung der dezentralen Steuerung

Der Aktualisierungszyklus der dezentralen Verbindung ist nicht mit dem Zyklus der Master-Steuerung synchronisiert. Die Kommunikation mit der dezentralen Steuerung ist interruptgesteuert und läuft als Hintergrund-Task parallel zum Zyklus der Master-Steuerung ab. Am Ende des Zyklus werden die aktuellsten Werte in den für die nächste Programmausführung zu verwendenden Applikationsdaten gelesen. Diese Vorgehensweise ist für dezentrale E/A und SPS-Erweiterungen identisch. Jede Steuerung kann die allgemeine Verbindungsaktivität über Systembit %S111 prüfen. Um jedoch eine Synchronisation zu erreichen, muss eine Master- oder Erweiterungssteuerung das Systembit %S110 verwenden. Dieses Bit wird auf 1 gesetzt, wenn ein vollständiger Aktualisierungszyklus abgelaufen ist. Das Applikationsprogramm ist dafür verantwortlich, dieses Bit auf 0 zurückzusetzen.
Der Master kann die dezentrale Verbindung mittels des Systembits %S112 aktivieren oder deaktivieren. Die Steuerungen können die Konfiguration und die ordnungsgemäße Funktionsweise der dezentralen Verbindung mittels %S113 prüfen. Das DPT-Signal an Port 1 (gibt an, ob TwidoSoft angeschlossen ist) wird abgefragt und in %S100 gemeldet.
Diese Systembits sind noch einmal in der folgenden Tabelle zusammengefasst.

Systembit	Zustand	Anzeige
%S100	0	Master/Slave: DPT inaktiv (TwidoSoft-Kabel NICHT angeschlossen)
	1	Master/Slave: DPT aktiv (TwidoSoft-Kabel angeschlossen)
%S110	0	Master/Slave: wird durch die Applikation auf 0 gesetzt
	1	Master: Der gesamte Austausch über die dezentrale Verbindung ist beendet (nur dezentrale E/A). Slave: Der Austausch mit Master ist beendet.
%S111	0	Master: Der einzelne Datenaustausch über die dezentrale Verbindung ist beendet. Slave: Der einzelne Datenaustausch über die dezentrale Verbindung wurde erkannt.
	1	Master: Der einzelne Datenaustausch über die dezentrale Verbindung läuft. Slave: Der einzelne Datenaustausch über die dezentrale Verbindung wurde erkannt.
%S112	0	Master: Dezentrale Verbindung deaktiviert
	1	Master: Dezentrale Verbindung aktiviert
%S113	0	Master/Slave: Die Konfiguration/der Betrieb der dezentralen Verbindung ist OK.
	1	Master: Fehler in der Konfiguration/im Betrieb der dezentralen Verbindung Slave: Fehler im Betrieb der dezentralen Verbindung

Neustart der Master-Steuerung

Startet eine Master-Steuerung neu, tritt eines der folgenden Ereignisse auf:
- Ein Kaltstart (%S0 = 1) forciert eine Reinitialisierung der Kommunikation.
- Ein Warmstart (%S1 = 1) forciert eine Reinitialisierung der Kommunikation.
- Im Betriebszustand STOP kommuniziert der Master weiterhin mit den Slaves.

Neustart der Slave-Steuerung

Startet eine Slave-Steuerung neu, tritt eines der folgenden Ereignisse auf:

- Ein Kaltstart (%S0 = 1) forciert eine Reinitialisierung der Kommunikation.
- Ein Warmstart (%S1 = 1) forciert eine Reinitialisierung der Kommunikation.
- Im Modus STOP kommuniziert der Slave weiterhin mit dem Master. Wenn der Master einen Stopp-Status angibt:
  - Die dezentrale E/A wendet einen Stopp-Status an.
  - Die SPS-Erweiterung setzt den Betrieb in ihrem aktuellen Status fort.

Stopp der Master-Steuerung

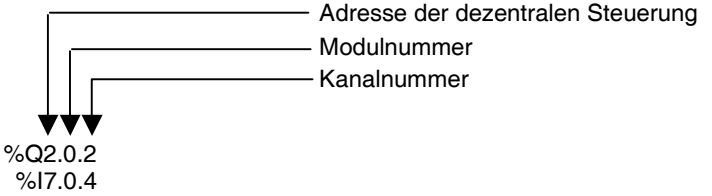
Geht die Master-Steuerung in den Betriebszustand STOP, setzen alle Slave-Geräte die Kommunikation mit dem Master fort. Zeigt der Master an, dass STOP gefordert ist, geht eine dezentrale E/A-Steuerung in den Modus STOP über, die SPS-Erweiterungen bleiben jedoch im jeweiligen Betriebszustand RUN oder STOP.

Zugriff auf Daten der dezentralen E/A

Die dezentrale Steuerung, die als dezentrale E/A konfiguriert ist, hat kein eigenes Applikationsprogramm. Die digitalen Eingänge und Ausgänge des Grundgeräts der dezentralen Steuerung sind eine einfache Erweiterung der Ein- und Ausgänge der Master-Steuerung. Die Applikation darf nur die vollständige Dreiziffernadressierung verwenden.

Hinweis: Die Modulnummer ist bei dezentralen E/A immer Null.

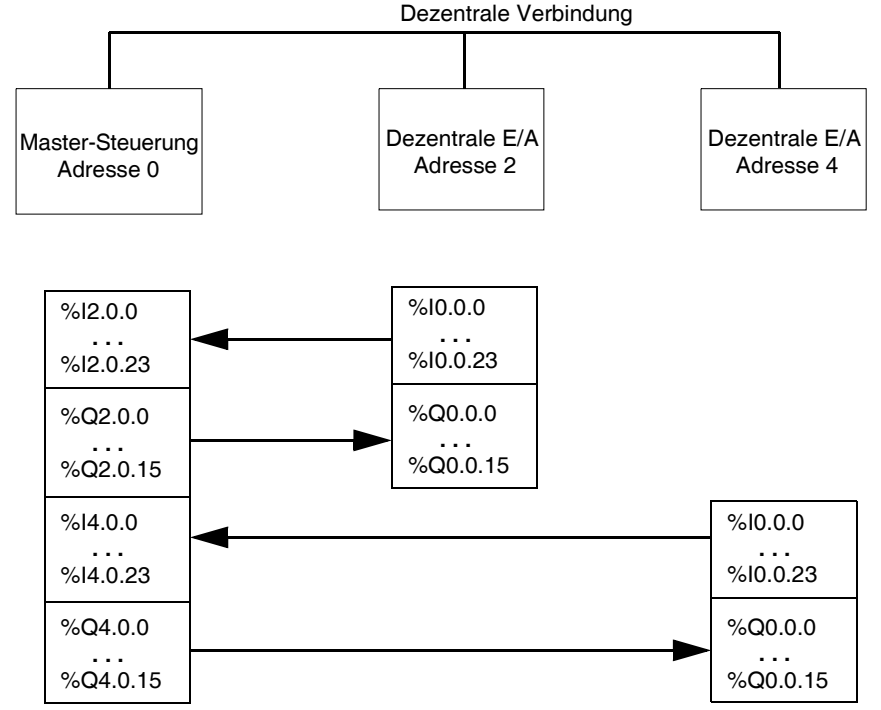
Darstellung



Um mit den dezentralen E/A zu kommunizieren, nutzt die Master-SPS die Standardbezeichnungen %I und %Q für die Ein- bzw. Ausgänge. Um beispielsweise auf das dritte Ausgangsbit der an Adresse 2 konfigurierten dezentralen E/A zuzugreifen, wird die Anweisung %Q2.0.2 verwendet. Um beispielsweise auf das fünfte Eingangsbit der an Adresse 7 konfigurierten dezentralen E/A zuzugreifen, wird die Anweisung %I7.0.4 verwendet.

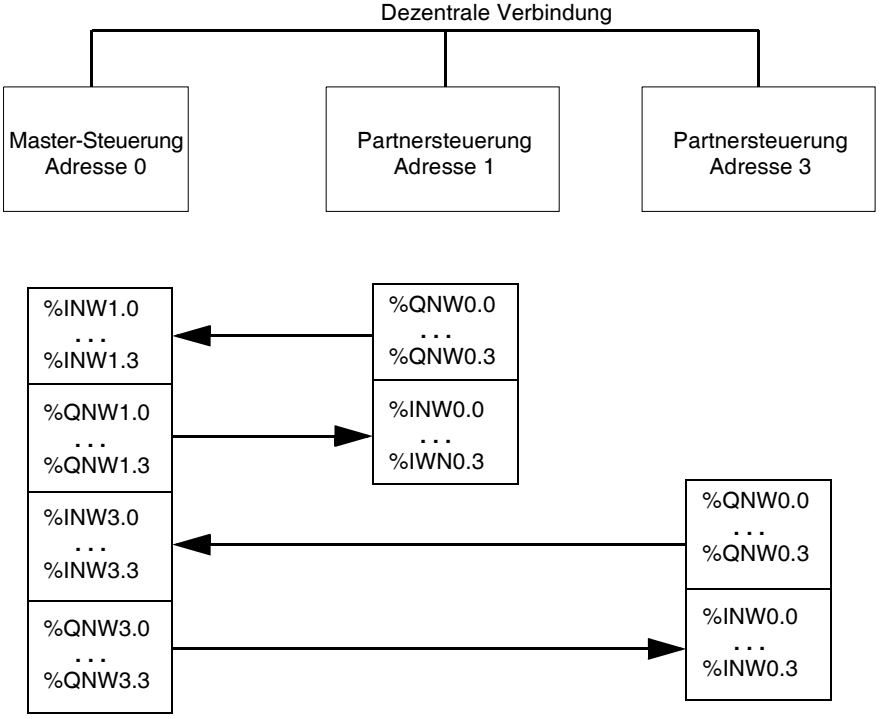
Hinweis: Der Master ist begrenzt auf den Zugriff auf die digitalen E/A, die Teil der dezentralen lokalen E/A sind. Es können keine analogen oder Erweiterungs-E/A übertragen werden, es sei denn, Sie verwenden die Partnerkommunikation.

Darstellung



Zugriff auf Daten der SPS-Erweiterung

Für die Kommunikation mit SPS-Erweiterungen nutzt der Master die Netzwerkwörter %INW und %QNW für den Datenaustausch. Auf jede Erweiterung des Netzwerks kann über ihre dezentrale Adresse "j" mithilfe der Wörter %INWj.k und %QNWj.k zugegriffen werden. Jede SPS-Erweiterung im Netzwerk nutzt die Wörter %INW0.0 bis %INW0.3 und %QNW0.0 bis %QNW0.3 für den Zugriff auf Daten im Master. Netzwerkwörter werden automatisch aktualisiert, wenn sich die Steuerungen im Betriebszustand RUN oder STOP befinden.
Das folgende Beispiel zeigt den Austausch eines Masters mit zwei konfigurierten SPS-Erweiterungen.



Es gibt keinen Austausch zwischen gleichrangigen Geräten in der dezentralen Verbindung. Es ist möglich, das Applikationsprogramm des Masters für die Verwaltung der Netzwerkwörter zu nutzen, um Informationen zwischen den dezentralen Steuerungen zu übertragen, wobei der Master als Gateway eingesetzt wird.

Statusinformationen

Neben den zuvor beschriebenen Systembits speichert der Master den Status (Vorhandensein und Konfiguration) von dezentralen Steuerungen. Dies geschieht in den Systemwörtern %SW111 und %SW113. Entweder die dezentrale Steuerung oder der Master kann während der Kommunikation über die dezentrale Verbindung im Systemwort %SW112 den Wert des letzten Fehlers ermitteln.

Systemwörter	Verwendung	
%SW111	Status der dezentralen Verbindung: zwei Bits für jede dezentrale Steuerung (nur Master)	
	x0-6	0 - dezentrale Steuerung 1-7 nicht vorhanden
		1 - dezentrale Steuerung 1-7 vorhanden
	x8-14	0 - dezentrale E/A an der dezentralen Steuerung 1-7 erkannt
		1 - SPS-Erweiterung an der dezentralen Steuerung 1-7 erkannt
%SW112	Fehlercode (Konfiguration/Betrieb) der dezentralen Verbindung	
		0 - Vorgänge erfolgreich abgeschlossen
		1 - Verzögerung abgelaufen (Slave)
		2 - Prüfsummenfehler erkannt (Slave)
		3 - Nichtübereinstimmung der Konfiguration (Slave)
%SW113	Konfiguration der dezentralen Verbindung: zwei Bits für jede dezentrale Steuerung (nur Master)	
	x0-6	0 - dezentrale Steuerung 1-7 nicht konfiguriert
		1- dezentrale Steuerung 1-7 konfiguriert
	x8-14	0 - dezentrale E/A als dezentrale Steuerung 1-7 konfiguriert
		1 - SPS-Erweiterung als dezentrale Steuerung 1-7 konfiguriert

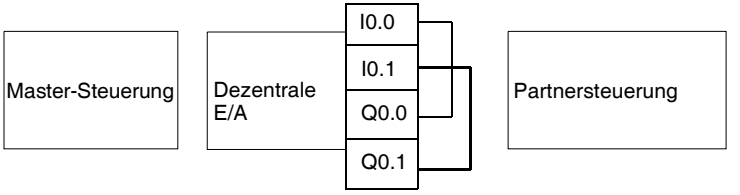
Beispiel für eine dezentrale Verbindung

Gehen Sie folgendermaßen zum Konfigurieren einer dezentralen Verbindung vor:

1. Die Hardware konfigurieren.
2. Verkabeln Sie die Steuerungen.
3. Schließen Sie das Kommunikationskabel am PC und an den Steuerungen an.
4. Konfigurieren Sie die Software.
5. Eine Applikation schreiben.

Die folgende Abbildung zeigt den Einsatz der dezentralen Verbindung mit dezentraler E/A und einer SPS-Erweiterung.

Schritt 1: Die Hardware konfigurieren:

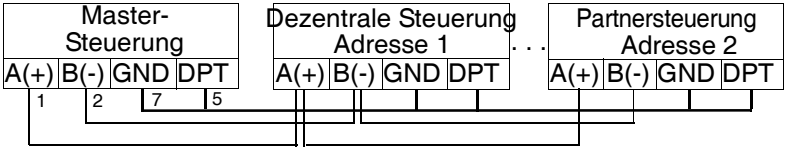


Die Hardwarekonfiguration besteht aus drei Basissteuerungen beliebigen Typs. Port 1 wird gemäß zwei Kommunikationsmodi genutzt. Ein Modus ermöglicht die Konfiguration und Übertragung des Applikationsprogramms mittels TwidoSoft. Der zweite Modus dient dem Netzwerk der dezentralen Verbindung. Sofern verfügbar, kann ein optionaler Port 2 an den Steuerungen genutzt werden, doch eine Steuerung unterstützt immer nur eine dezentrale Verbindung.

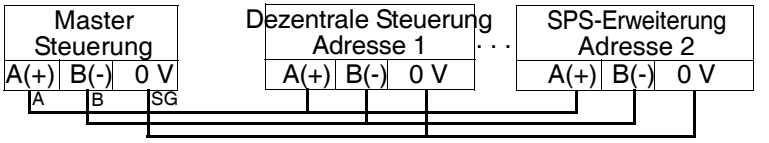
Hinweis: In diesem Beispiel sind die ersten beiden Eingänge an den dezentralen E/A fest mit den beiden ersten Ausgängen verdrahtet.

Schritt 2: Verkabelung der Steuerungen

miniDIN-Anschluss

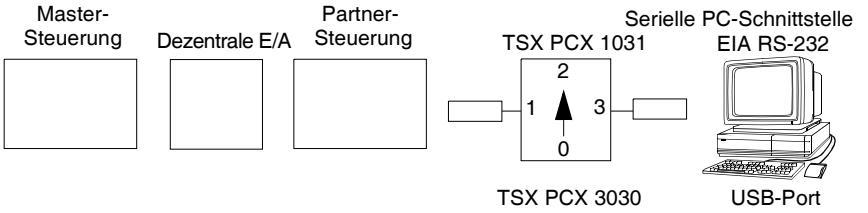


Klemmenleistenanschluss



Verbinden Sie die Signaldrähte A(+) und B(-) miteinander. Erden Sie an jeder Steuerung das DPT-Signal. Auch wenn es nicht erforderlich ist, das Signal zu erden, wenn die dezentrale Verbindung am Port 2 genutzt wird (optionale Sicherungskassette oder Kommunikationsmodul), ist es eine gute Maßnahme.

Schritt 3: Anschluss des Kommunikationskabels an den PC und an die Steuerungen



Die Multifunktions-Programmierkabel TSX PCX 1031 oder TSX PCX 3030 dienen der Kommunikation mit allen drei Basissteuerungen. Achten Sie darauf, dass am Kabel die Schalterstellung 2 eingestellt wurde. Für die Programmierung der einzelnen Steuerungen müssen Sie eine Punkt-zu-Punkt-Kommunikation zu jeder Steuerung herstellen. Gehen Sie folgendermaßen vor, um diese Kommunikationsverbindung herzustellen: Stellen Sie eine Verbindung zu Port 1 der ersten Steuerung her, übertragen Sie die Konfiguration und die Applikationsdaten und versetzen Sie die Steuerung in den Betriebszustand RUN. Wiederholen Sie diese Vorgehensweise für alle Steuerungen.

Hinweis: Das Kabel muss nach jeder Steuerungskonfiguration und Applikationsübertragung neu angeschlossen werden.

Schritt 4: Softwarekonfiguration:
Jede der drei Steuerungen nutzt TwidoSoft, um eine Konfiguration und bei Bedarf ein Applikationsprogramm zu erstellen.
Bearbeiten Sie für die Master-Steuerung das Kommunikations-Setup, um im Protokoll "Dezentrale Verbindung" und die Adresse "0" (Master) einzustellen.

Komm.-Setup der Steuerung
Typ: Dezentrale Verbindung
Adresse: 0 (Master)

Konfigurieren Sie die dezentrale Steuerung am Master, indem Sie an Adresse "1" eine "Dezentrale E/A" und an Adresse "2" eine "SPS-Erweiterung" hinzufügen.

Dezentrale Steuerungen hinzufügen

Steuerungsauslastung: Dezentrale E/A
Dezentrale Adresse: 1

Steuerungsauslastung: Partnersteuerung
Dezentrale Adresse: 2

Sie müssen für die Steuerung, die als dezentrale E/A eingerichtet ist, prüfen, dass im Kommunikations-Setup der Steuerung "Dezentrale Verbindung" und die Adresse "1" eingestellt sind.

Komm.-Setup der Steuerung
Typ: Dezentrale Verbindung
Adresse: 1

Sie müssen für die Steuerung, die als Erweiterung eingerichtet ist, prüfen, dass im Kommunikations-Setup der Steuerung "Dezentrale Verbindung" und die Adresse "2" eingestellt sind.

Komm.-Setup der Steuerung
Typ: Dezentrale Verbindung
Adresse: 2

Schritt 5: Schreiben von Applikationen:
Schreiben Sie für die Master-Steuerung das folgende Applikationsprogramm:

```
LD 1

[%MW0 := %MW0 +1]
[%QNW2.0 := %MW0]
[%MW1 := %INW2.0]

LD %I0.0
ST %Q1.00.0
LD %I1.0.0
ST %Q0.0

LD %I0.1
ST %Q1.0.1
LD %I1.0.1
ST %Q0.1
```

Schreiben Sie für die Steuerung, die als dezentrale E/A konfiguriert ist, kein Applikationsprogramm.

Schreiben Sie für die Steuerung, die als Erweiterung konfiguriert ist, die folgende Applikation:

```
LD 1
[%QNW0.0 := %INW0.0]
```

In diesem Beispiel inkrementiert die Master-Applikation ein internes Speicherwort und kommuniziert dieses über ein einzelnes Netzwerkwort an die SPS-Erweiterung. Die SPS-Erweiterung verwendet das vom Master empfangene Wort und meldet es zurück. Im Master empfängt ein anderes Speicherwort diese Übertragung und speichert sie.
Bei der Kommunikation mit der dezentralen E/A-Steuerung sendet der Master seine lokalen Eingänge an die Ausgänge der dezentralen E/A. Über die externe E/A-Verbindung der dezentralen E/A werden die Signale zurückgemeldet und vom Master abgerufen.

ASCII-Kommunikation

Einführung

ASCII-Protokolle bieten Twido-Steuerungen ein einfaches Halbduplex-Protokoll für den Zeichenmodus, um Daten mit einem einfachen Gerät zu senden und/oder zu empfangen. Dieses Protokoll wird von der Anweisung EXCHx unterstützt und vom Funktionsblock %MSGx gesteuert.
Beim ASCII-Protokoll sind drei Kommunikationsarten möglich:

- Nur Senden
- Senden/Empfangen
- Nur Empfangen

Die maximale Größe der gesendeten und/oder empfangenen Datenübertragungsblöcke beträgt bei der Anweisung EXCHx 256 Bytes.

Hardwarekonfiguration

Eine ASCII-Verbindung (siehe Systembits %S103 und %S104 (Siehe *Systembits (%S), S. 620*)) kann über den EIA RS-232- und den EIA RS-485-Port aufgebaut werden, und sie kann auf zwei Kommunikationsschnittstellen gleichzeitig laufen. Die folgende Tabelle führt die Geräte auf, die verwendet werden können.

Dezentral	Port	Kenndaten
TWDLC•A10/16/24DRF, TWDLCA•40DRF, TWDLMDA20/40DTK, TWDLMDA20DRT	1	Basissteuerung, ausgestattet mit einem dreiadrigen EIA RS-485-Port mit miniDIN-Stecker.
TWDNOZ232D	2	Kommunikationsmodul, ausgestattet mit einem dreiadrigen EIA RS-232-Port mit miniDIN-Steckverbinder. **Hinweis:** Dieses Modul ist nur für die modularen Steuerungen erhältlich. Wenn das Modul angeschlossen ist, kann die Steuerung kein Bedienterminal als Erweiterungsmodul haben.
TWDNOZ485D	2	Kommunikationsmodul, ausgestattet mit einem dreiadrigen EIA RS-485-Port mit miniDIN-Steckverbinder. **Hinweis:** Dieses Modul ist nur für die modularen Steuerungen erhältlich. Wenn das Modul angeschlossen ist, kann die Steuerung kein Bedienterminal als Erweiterungsmodul haben.
TWDNOZ485T	2	Kommunikationsmodul, ausgestattet mit einem dreiadrigen EIA RS-485-Port mit Anschlussklemme. **Hinweis:** Dieses Modul ist nur für die modularen Steuerungen erhältlich. Wenn das Modul angeschlossen ist, kann die Steuerung kein Bedienterminal als Erweiterungsmodul haben.
TWDNAC232D	2	Kommunikations-Adapter, ausgestattet mit einem dreiadrigen EIA RS-232-Port mit miniDIN-Steckverbinder. **Hinweis:** Dieser Adapter ist nur für die kompakten Steuerungen mit 16, 24 und 40 E/A und für das Bedienterminal-Erweiterungsmodul verfügbar.

Dezentral	Port	Kenndaten
TWDNAC485D	2	Kommunikations-Adapter, ausgestattet mit einem dreiadrigen EIA RS-485-Port mit miniDIN-Steckverbinder. **Hinweis:** Dieser Adapter ist nur für die kompakten Steuerungen mit 16, 24 und 40 E/A und für das Bedienterminal-Erweiterungsmodul verfügbar.
TWDNAC485T	2	Kommunikations-Adapter, ausgestattet mit einem dreiadrigen EIA RS-485-Port mit Anschlussklemme. **Hinweis:** Dieser Adapter ist nur für die kompakten Steuerungen mit 16, 24 und 40 E/A und für das Bedienterminal-Erweiterungsmodul verfügbar.
TWDXCPODM	2	Bedienterminal als Erweiterungsmodul, ausgestattet mit einem dreiadrigen EIA RS-232-Port mit miniDIN-Steckverbinder, einem dreiadrigen EIA RS-485-Port mit miniDIN-Steckverbinder und einem dreiadrigen EIA RS-485-Port mit einer Anschlussklemme. **Hinweis:** Dieses Modul ist nur für die modularen Steuerungen erhältlich. Wenn das Modul angeschlossen ist, kann die Steuerung kein Kommunikationserweiterungsmodul haben.

Hinweis: Die Überprüfung des Vorhandenseins von Port 2 und dessen Konfiguration (RS232 oder RS 485) wird nur beim Einschalten oder bei der Reinitialisierung durch die Firmware der Steuerung durchgeführt.

Nominale Verkabelung

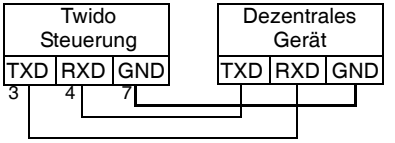
Im Folgenden sind die Anschlüsse eines nominalen Kabels für die Typen EIA RS-232 und EIA RS-485 abgebildet.

Hinweis: Wird an der Twido Steuerung Port 1 verwendet, muss das DPT-Signal von Pin 5 mit 0 V an Pin 7 verbunden werden. Dies bedeutet für die Twido Steuerung, dass die Kommunikation über Port 1 eine ASCII-Kommunikation ist und dass es sich nicht um das Protokoll für die Kommunikation mit der Software TwidoSoft handelt.

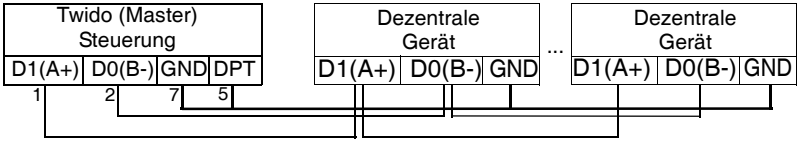
Die Kabelverbindungen zu allen Geräten sind im Folgenden dargestellt.

miniDIN-Anschluss

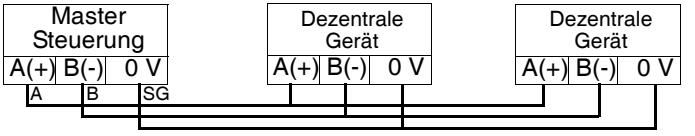
EIA RS-232-Kabel



EIA RS-485-Kabel



Klemmenleistenanschluss



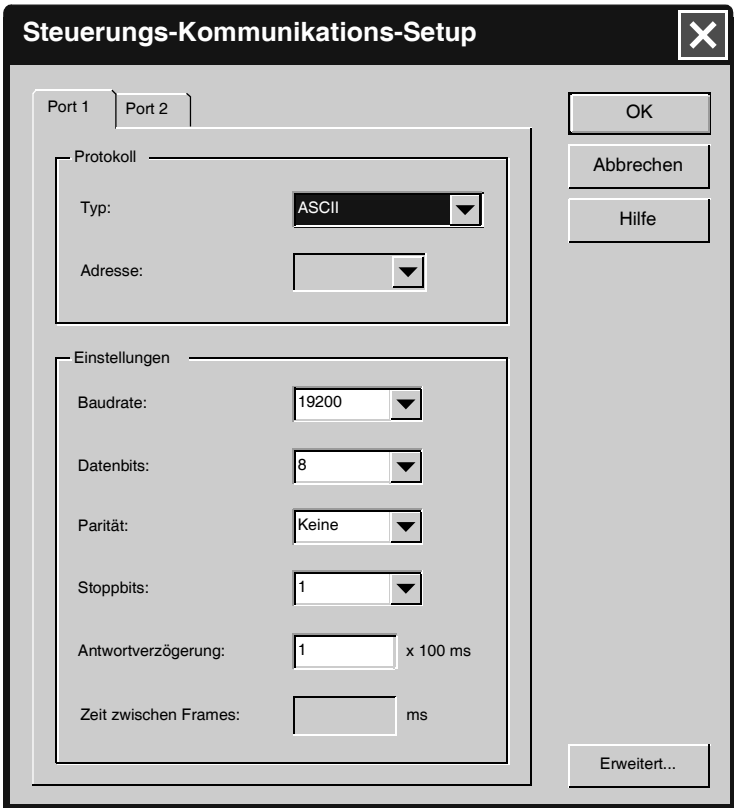
Softwarekonfiguration

Um die Steuerung so einzurichten, dass sie Zeichen mittels ASCII-Protokoll über eine serielle Verbindung sendet und empfängt, müssen Sie:

Schritt	Beschreibung
1	die serielle Schnittstelle mit TwidoSoft für ASCII konfigurieren
2	in Ihrer Applikation eine Sende-/Empfangstabelle erstellen, die von der Anweisung EXCHx genutzt wird.

Konfigurieren der Schnittstelle

Eine Twido-Steuerung kann die primäre Schnittstelle 1 oder eine alternativ konfigurierte Schnittstelle 2 für das ASCII-Protokoll verwenden. So konfigurieren Sie eine serielle Schnittstelle für ASCII:

Schritt	Aktion
1	Definieren Sie alle zusätzlichen Module oder Kommunikations-Adapter, die für die Basis konfiguriert sind.
2	Klicken Sie mit der rechten Maustaste auf den Port im Applikationsnavigator, und wählen Sie **Steuerungs-Kommunikations-Setup bearbeiten...** **Ergebnis:** Das folgende Fenster wird geöffnet. 
3	Wählen Sie einen Port des Typs ASCII seriell aus dem Listenfeld **Protokolltyp** aus.
4	Stellen Sie die zugehörigen Kommunikationsparameter ein.
5	Klicken Sie auf die Schaltfläche **Erweitert**, um die erweiterten Parameter festzulegen.

Konfiguration der Sende-/ Empfangstabelle für den ASCII-Modus

Die Maximalgröße der gesendeten und/oder empfangenen Rahmen lautet 256 Bytes. Die zur Anweisung EXCHx gehörende Worttabelle besteht aus den Sende- und Empfangs-Steuertabellen.

	Höherwertiges Byte	**Niederwertiges Byte**
Steuertabelle	Befehl	Länge (Senden/Empfangen)
	Reserviert (0)	Reserviert (0)
Sendetabelle	Byte 1 gesendet	Byte 2 gesendet
	...	...
		Byte n gesendet
	Byte n+1 gesendet	
Empfangstabelle	Byte 1 empfangen	Byte 2 empfangen
	...	...
		Byte p empfangen
	Byte p+1 empfangen	

Steuertabelle

Das **Längen**byte enthält die Länge der Sendetabelle in Bytes (max. 250), die von der Anzahl der empfangenen Zeichen am Ende des Empfangs überschrieben wird, sofern ein Empfang angefordert wird.
Das **Befehls**byte muss eines der folgenden Elemente enthalten:

- 0: Nur Senden
- 1: Senden/Empfangen
- 2: Nur Empfangen

Sende-/ Empfangstabellen

Im Modus "Nur Senden" werden die Steuer- und Sendetabellen vor der Ausführung der Anweisung EXCHx belegt; sie können vom Typ %KW oder %MW sein. Im Modus "Nur Senden" ist kein Platz erforderlich für den Empfang von Zeichen. Nachdem alle Bytes gesendet wurden, wird %MSGx.D auf 1 gesetzt und eine neue Anweisung EXCHx kann ausgeführt werden.
Im Modus "Senden/Empfangen" werden die Steuer- und Sendetabellen vor der Ausführung der Anweisung EXCHx belegt; sie müssen vom Typ %MW sein. Am Ende der Sendetabelle wird Platz für maximal 256 Empfangsbytes benötigt. Nachdem alle Bytes gesendet wurden, schaltet die Twido-Steuerung in den Empfangsmodus um und wartet auf den Empfang von Bytes.
Im Modus " Nur Empfangen" wird die Steuertabelle vor der Ausführung der Anweisung EXCHx gefüllt und muss vom Typ %MW sein. Am Ende der Steuertabelle wird Platz für maximal 256 Empfangsbytes benötigt. Die Twido-Steuerung muss sofort in den Empfangsmodus gehen und wartet auf den Empfang von Bytes.
Der Empfang endet, wenn verwendete Bytes zur Kennzeichnung des Datenübertragungsblockendes empfangen werden oder die Empfangstabelle voll ist. In diesem Fall wird ein Fehler (Überlauf Empfangstabelle) in den Wörtern %SW63 und %SW64 angezeigt. Ist ein Timeout ungleich Null eingerichtet, endet der Empfang nach Ablauf des Timeouts. Wurde ein Timeout gleich Null eingestellt, gibt es kein Empfangs-Timeout. Deshalb muss der Eingang %MSGx.R aktiviert werden, um den Empfang zu stoppen.

Meldungsaustausch

Die Sprache bietet für die Kommunikation zwei Funktionen:

- **Anweisung EXCHx:** zum Senden/Empfangen von Meldungen
- **Funktionsblock %MSGx:** zum Steuern des Meldungsaustauschs.

Die Twido-Steuerung nutzt das für den Port eingerichtete Protokoll für die Bearbeitung der Anweisung EXCHx.

Hinweis: Jeder Kommunikations-Port kann für ein unterschiedliches Protokoll oder alle Ports können für das gleiche Protokoll eingerichtet werden. Auf die Anweisung EXCHx oder den Funktionsblock %MSGx für jeden Kommunikations-Port wird über die angehängte Portnummer zugegriffen (1 oder 2).

Anweisung EXCHx

Mit der Anweisung EXCH kann die Twido-Steuerung Informationen von ASCII-Geräten empfangen und Informationen an ASCII-Geräte senden. Der Anwender definiert eine Tabelle mit Wörtern (%MWi:L oder %KWi:L), die Steuerungsinformationen und die Daten enthalten, die gesendet und/oder empfangen werden sollen (maximal 256 Bytes zum Senden und/oder Empfangen). Das Format der Worttabelle wurde oben beschrieben.
Ein Meldungsaustausch wird mit der Anweisung EXCHx durchgeführt:

Syntax: [EXCHx %MWi:L]
wobei gilt: x = Portnummer (1 oder 2)
L = Anzahl der Wörter der Steuerworttabelle, der Sendetabelle und der Empfangstabelle

Die Twido-Steuerung muss den Austausch einer ersten Anweisung EXCHx beenden, bevor eine zweite Anweisung gestartet werden kann. Der Funktionsblock %MSGx muss verwendet werden, wenn mehrere Meldungen gesendet werden sollen.
Die Bearbeitung der AWL-Anweisung EXCHx geschieht sofort und alle Übertragungen werden interruptgesteuert gestartet (auch der Empfang von Daten läuft interruptgesteuert ab), was als Hintergrundbearbeitung betrachtet wird.

Funktionsblock %MSGx

Der Einsatz des Funktionsblocks %MSGx ist optional. Mit diesem Funktionsblock kann der Datenaustausch verwaltet werden. Der Funktionsblock %MSGx dient drei Zwecken:

- **Kommunikationsfehlerprüfung**
  Die Fehlerprüfung bestätigt, dass der mit der Anweisung EXCHx programmierte Parameter L (Länge der Worttabelle) groß genug ist, um die Länge der zu sendenden Meldung aufzunehmen. Es wird ein Vergleich mit der im niederwertigsten Byte des ersten Worts der Worttabelle programmierten Länge durchgeführt.
- **Koordination mehrerer Meldungen**
  Um beim Senden von mehreren Meldungen die Koordination sicherzustellen, liefert der Funktionsblock %MSGx die Informationen, anhand derer festgestellt werden kann, wann die vorherige Meldung beendet ist.
- **Senden von Prioritätsmeldungen**
  Der Funktionsblock %MSGx lässt es zu, dass die aktuelle Meldungsübertragung gestoppt wird, damit eine dringende Meldung sofort gesendet werden kann.

Der Funktionsblock %MSGx hat einen zugehörigen Eingang und zwei zugehörige Ausgänge:

Eingang/Ausgang	Definition	Beschreibung
R	RESET-Eingang	Auf 1 gesetzt: initialisiert die Kommunikation neu oder setzt den Block zurück (%MSGx.E = 0 und %MSGx.D = 1).
%MSGx.D	Kommunikation beendet	0: Request in Bearbeitung. 1: Kommunikation beendet, wenn Ende der Übertragung, Endezeichen empfangen, Fehler oder Rücksetzen des Blocks.
%MSGx.E	Fehler	0: Meldungslänge OK und Verbindung OK. 1: wenn falscher Befehl, Tabelle falsch konfiguriert, falsches Zeichen empfangen (Geschwindigkeit, Parität usw.) oder Empfangstabelle voll.

Einschränkungen

Es ist wichtig, folgende Einschränkungen zu beachten:

- Die Verfügbarkeit und der Typ von Schnittstelle 2 (siehe %SW7) werden nur beim Anlauf oder beim Rücksetzen geprüft.
- Die Meldungsbearbeitung an Port 1 wird abgebrochen, wenn TwidoSoft verbunden wird.
- EXCHx oder %MSG kann nicht an einem Port bearbeitet werden, der als dezentrale Verbindung konfiguriert ist.
- EXCHx bricht aktive Modbus-Slave-Bearbeitung ab.
- Die Bearbeitung der Anweisungen EXCHx wird im Fehlerfall nicht wiederholt.
- Mit dem Rücksetzeingang kann die Empfangsbearbeitung der Anweisung EXCHx beendet werden.
- Anweisungen EXCHx können mit einem Timeout zum Beenden des Empfangs konfiguriert werden.
- Mehrere Meldungen werden über %MSGx.D gesteuert.

Fehler- und Betriebszustandsbedingungen

Tritt bei Verwendung einer Anweisung EXCHx ein Fehler auf, werden die Bits %MSG.D und %MSG.E auf 1 gesetzt und das Systemwort %SW63 enthält den Fehlercode für Port 1 und das Systemwort %SW64 enthält den Fehlercode für Port 2.

Systemwörter	Verwendung
%SW63	Fehlercode EXCH1: 0 - Vorgang war erfolgreich 1 - Anzahl zu sendender Bytes zu hoch (> 250) 2 - Sendetabelle zu klein 3 - Worttabelle zu klein 4 - Überlauf Empfangstabelle 5 - Timeout abgelaufen 6 - Sendefehler 7 - Falscher Befehl in Tabelle 8 - Ausgewählter Port nicht konfiguriert/verfügbar 9 - Empfangsfehler 10 - %KW kann nicht beim Empfang verwendet werden 11 - Sende-Offset größer als Sendetabelle 12 - Empfangs-Offset größer als Empfangstabelle 13 - Steuerung hat Bearbeitung von EXCH gestoppt
%SW64	Fehlercode EXCH2: Siehe %SW63.

Auswirkung des Neustarts der Steuerung auf die Kommunikation

Startet eine Steuerung neu, tritt eines der folgenden Ereignisse auf:

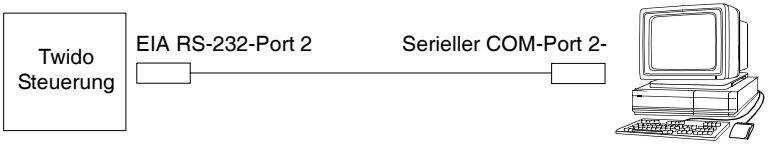
- Ein Kaltstart (%S0 = 1) forciert eine Reinitialisierung der Kommunikation.
- Ein Warmstart (%S1 = 1) forciert eine Reinitialisierung der Kommunikation.
- Im Betriebszustand STOP stoppt die Steuerung sämtliche ASCII-Kommunikation.

Beispiel für eine ASCII-Verbindung

Zum Konfigurieren einer ASCII-Verbindung müssen Sie:
1. Die Hardware konfigurieren.
2. Das ASCII-Kommunikationskabel anschließen.
3. Den Port konfigurieren.
4. Eine Applikation schreiben.
5. Den Animationstabellen-Editor initialisieren.

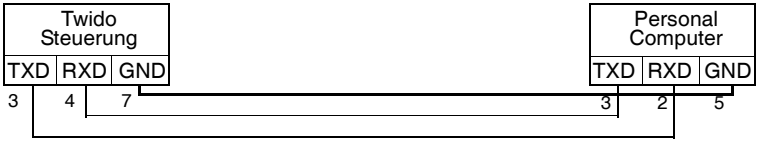
Die folgende Abbildung zeigt den Einsatz der ASCII-Kommunikation mit einem Terminal-Emulator auf einem PC.

Schritt 1: Hardwarekonfiguration:



Die Hardware-Konfiguration besteht aus zwei seriellen Verbindungen vom PC zu einer Twido Steuerung mit einer optionalen EIA RS-232 Schnittstelle 2. Bei einer modularen Steuerung handelt es sich bei der optionalen Schnittstelle 2 um eine TWDNOZ232D oder TWDNAC232D in TWDXCPODM. Bei der kompakten Steuerung ist die optionale Schnittstelle 2 eine TWDNAC232D.
Zum Konfigurieren der Steuerung schließen Sie das Kabel TSXPCX1031 (nicht gezeigt) an Schnittstelle 1 der Twido-Steuerung an. Dann schließen Sie das Kabel an den COM 1-Port des PC an. Achten Sie darauf, dass der Schalter sich in Stellung 2 befindet. Schließen Sie zum Schluss die Schnittstelle COM 2 des PC an die optionale EIA RS-232 Schnittstelle 2 der Twido Steuerung an. Der Verdrahtungsplan ist im nächsten Schritt dargestellt.

Schritt 2: Verdrahtungsplan für das ASCII-Kommunikationskabel (EIA RS-232):



Die minimale Anzahl der im ASCII-Kommunikationskabel verwendeten Drähte lautet 3. Kreuzen Sie die Sende- und Empfangssignale.

Hinweis: Auf der PC-Seite des Kabels sind möglicherweise zusätzliche Anschlüsse (z.B. Endgerät betriebsbereit und Betriebsbereitschaft) erforderlich, um die Handshake-Funktion zu ermöglichen. Für die Twido-Steuerung sind keine weiteren Anschlüsse erforderlich.

Schritt 3: Port-Konfiguration:

Hardware -> Option hinzufügen TWDNOZ232D	
Serieller Port 2	
Protokoll	ASCII
Adresse	
Baudrate	19200
Datenbits	8
Parität	Keine
Stoppbit	1
Antwort-Zeitüberschreitung (x100 ms)	100
Zeit zwischen Frames (ms)	
Startzeichen	
1. Endezeichen	65
2. Endezeichen	
Stopp bei Ruhe (ms)	
Stopp bei Anzahl empfangener Bytes	

Terminal-Emulator auf einem PC	
Port:	COM2
Baudrate:	19200
Daten:	8 Bit
Parität:	Keine
Stopp:	1 Bits
Flusskontrolle:	Keine

Verwenden Sie eine Terminal-Emulator-Applikation auf dem PC, um den COM-Port 2 zu konfigurieren und um sicherzustellen, dass keine Flusskontrolle eingestellt ist. Konfigurieren Sie die Schnittstelle der Steuerung in TwidoSoft. Zuerst wird die Hardware-Option eingerichtet. In diesem Beispiel wird die modulare Basissteuerung um TWDNOZ232D ergänzt.
Dann wird das SPS-Kommunikations-Setup mit den gleichen Parametereinstellungen wie der Terminal-Emulator auf dem PC initialisiert. In diesem Beispiel wird der Großbuchstabe "A" als "1. Endezeichen" gewählt, um den Zeichenempfang zu beenden. Für den Parameter "Antwort-Zeitüberschreitung" wird eine Zeit von 10 Sekunden gewählt. Nur einer dieser beiden Parameter wird verwendet, je nachdem, was zuerst eintritt.

Schritt 4: Schreiben einer Applikation:

```
LD 1
[%MW10 := 16#0104]
[%MW11 := 16#0000]
[%MW12 := 16#4F4B]
[%MW13 := 16#0A0D]
LD 1
AND %MSG2.D
[EXCH2 %MW10:8]
LD %MSG2.E
ST %Q0.0
END
```

Erstellen Sie mit TwidoSoft ein Anwendungsprogramm mit drei primären Teilen. Initialisieren Sie zunächst die Steuerungs- und Sendetabelle für die Verwendung mit der Anweisung EXCH. In diesem Beispiel wird ein Befehl eingerichtet, Daten sowohl zu senden als auch zu empfangen. Die zu sendende Datenmenge wird wie in der Applikation definiert auf 4 Bytes gesetzt, gefolgt vom verwendeten Endezeichen (in diesem Fall ist das erste Endezeichen ein "A"). Start- und Endezeichen werden nicht in der Animationstabelle angezeigt. In dieser Tabelle werden nur Datenzeichen angezeigt. Diese Zeichen werden jedoch automatisch gesendet oder beim Empfang überprüft (durch %SW63 und %SW64), wenn sie verwendet werden. Prüfen Sie dann das mit %MSG2 verknüpfte Kommunikationsstatusbit und führen die Anweisung EXCH2 nur aus, wenn die Schnittstelle bereit ist. Für die Anweisung EXCH2 ist ein Wert von 8 Wörtern angegeben. Es gibt 2 Steuerwörter (%MW10 und %MW11), 2 Wörter zum Senden von Informationen (%MW12 und %MW13) und 4 Wörter zum Empfangen von Daten (%MW14 bis %MW16).
Zum Schluss wird der Fehlerstatus von %MSG2 abgefragt und im ersten Ausgangsbit der lokalen Basissteuerungs-E/A gespeichert. Es könnte eine weitere Fehlerprüfung mit %SW64 durchgeführt werden, um eine höhere Genauigkeit zu erreichen.

Schritt 5: Initialisierung des Animationstabellen-Editors:

Aktuell	gespeichertes	Adressformat
1 %MW10	0104	Hexadezimal
2 %MW11	0000	Hexadezimal
3 %MW12	4F4B	Hexadezimal
4 %MW13	0A0D	Hexadezimal
5 %MW14	TW	ASCII
6 %MW15	ID	ASCII
7 %MW16	O	ASCII

Im letzten Schritt wird diese Applikationssteuerung geladen und ausgeführt. Initialisieren Sie einen Animationstabellen-Editor, um die Wörter %MW10 bis %MW16 zu animieren und anzuzeigen. Die Zeichen "O- K - CR - LF - A" können am Terminal Emulator so häufig angezeigt werden, wie die Antwortverzögerung des Blocks EXCH abgelaufen ist. Geben Sie am Terminal-Emulator "T - W - I - D - O - A" ein. Diese Information wird mit der Twido-Steuerung ausgetauscht und im Animationstabellen-Editor angezeigt.

Modbus-Kommunikation

Einführung

Das Modbus-Protokoll ist ein Master/Slave-Protokoll, bei dem nur ein einziger Master Antworten von Slaves anfordern und anhand des Requests agieren darf. Der Master kann einzelne Slaves ansprechen oder er kann eine Broadcast-Meldung an alle Slaves ausgeben. Slaves senden eine Meldung (Antwort) auf Requests, die individuell an sie gerichtet worden sind, zurück. Es werden keine Antworten auf Broadcast-Requests vom Master zurückgesendet.

⚠ ACHTUNG

UNERWARTETER SYSTEMVORGANG

- Achten Sie darauf, dass es in auf dem Bus nur einen Modbus-Master gibt und dass jeder Modbus-Slave über eine eindeutige Adresse verfügt. Bei Missachtung dieser Vorsichtsmaßnahme können Daten verändert oder unerwartete und mehrdeutige Ergebnisse hervorgerufen werden.
- Achten Sie darauf, dass alle Modbus-Slaves über eindeutige Adressen verfügen. Slave-Adressen dürfen nicht doppelt vergeben werden. Bei Missachtung dieser Vorsichtsmaßnahme können Daten verändert oder unerwartete und mehrdeutige Ergebnisse hervorgerufen werden.

Die Nichtbeachtung dieser Anweisung kann Körperverletzung oder Materialschäden zur Folge haben!

Hardwarekonfiguration

Eine Modbus-Verbindung kann über den EIA RS-232- und den EIA RS-485-Port aufgebaut werden, und sie kann auf zwei Kommunikations-Ports gleichzeitig laufen. Jedem dieser Ports kann mithilfe des Systembits %S101 und der Systemwörter %SW101 und %SW102 (Siehe *Systembits (%S), S. 620*) eine eigene Modbus-Adresse zugewiesen werden. . (Siehe auch *Systemwörter (%SW), S. 633*)
Die folgende Tabelle führt die Geräte auf, die verwendet werden können.

Dezentral	Port	Kenndaten
TWDLC•A10/16/24DRF, TWDLCA•40DRF, TWDLMDA20/40DTK, TWDLMDA20DRT	1	Basissteuerung, ausgestattet mit einem dreiadrigen EIA RS-485-Port mit miniDIN-Steckverbinder.
TWDNOZ232D	2	Kommunikationsmodul, ausgestattet mit einem dreiadrigen EIA RS-232-Port mit miniDIN-Steckverbinder. **Hinweis:** Dieses Modul ist nur für die modularen Steuerungen erhältlich. Wenn das Modul angeschlossen ist, kann die Steuerung kein Bedienterminal als Erweiterungsmodul haben.

Dezentral	Port	Kenndaten
TWDNOZ485D	2	Kommunikationsmodul, ausgestattet mit einem dreiadrigen EIA RS-485-Port mit miniDIN-Steckverbinder. **Hinweis:** Dieses Modul ist nur für die modularen Steuerungen erhältlich. Wenn das Modul angeschlossen ist, kann die Steuerung kein Bedienterminal als Erweiterungsmodul haben.
TWDNOZ485T	2	Kommunikationsmodul, ausgestattet mit einem dreiadrigen EIA RS-485-Port mit Anschlussklemme. **Hinweis:** Dieses Modul ist nur für die modularen Steuerungen erhältlich. Wenn das Modul angeschlossen ist, kann die Steuerung kein Bedienterminal als Erweiterungsmodul haben.
TWDNAC232D	2	Kommunikations-Adapter, ausgestattet mit einem dreiadrigen EIA RS-232-Port mit miniDIN-Steckverbinder. **Hinweis:** Dieser Adapter ist nur für die kompakten Steuerungen mit 16, 24 und 40 E/A und für das Bedienterminal-Erweiterungsmodul verfügbar.
TWDNAC485D	2	Kommunikations-Adapter, ausgestattet mit einem dreiadrigen EIA RS-485-Port mit miniDIN-Steckverbinder. **Hinweis:** Dieser Adapter ist nur für die kompakten Steuerungen mit 16, 24 und 40 E/A und für das Bedienterminal-Erweiterungsmodul verfügbar.
TWDNAC485T	2	Kommunikations-Adapter, ausgestattet mit einem dreiadrigen EIA RS-485-Port mit Anschlussklemme. **Hinweis:** Dieser Adapter ist nur für die kompakten Steuerungen mit 16, 24 und 40 E/A und für das Bedienterminal-Erweiterungsmodul verfügbar.
TWDXCPODM	2	Bedienterminal als Erweiterungsmodul, ausgestattet mit einem dreiadrigen EIA RS-232-Port mit miniDIN-Steckverbinder, einem dreiadrigen EIA RS-485-Port mit miniDIN-Steckverbinder und einem dreiadrigen EIA RS-485-Port mit einer Anschlussklemme. **Hinweis:** Dieses Modul ist nur für die modularen Steuerungen erhältlich. Wenn das Modul angeschlossen ist, kann die Steuerung kein Kommunikationserweiterungsmodul haben.

Hinweis: Die Überprüfung des Vorhandenseins von Port 2 und dessen Konfiguration (RS232 oder RS 485) wird beim Einschalten oder beim Zurücksetzen durch die Firmware der Steuerung durchgeführt.

Nominale Verkabelung

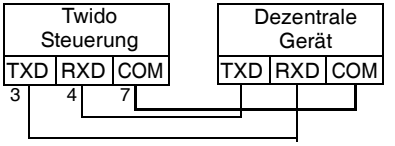
Im Folgenden sind die Anschlüsse eines nominalen Kabels für die Typen EIA RS-232 und EIA RS-485 abgebildet.

Hinweis: Wird an der Twido Steuerung Port 1 verwendet, muss das DPT-Signal von Pin 5 mit der gemeinsamen Rückleitung (COM) an Pin 7 verbunden werden. Dies bedeutet für die Twido Steuerung, dass die Kommunikation über Port 1 eine Modbus-Kommunikation ist und dass es sich nicht um das Protokoll für die Kommunikation mit der Software TwidoSoft handelt.

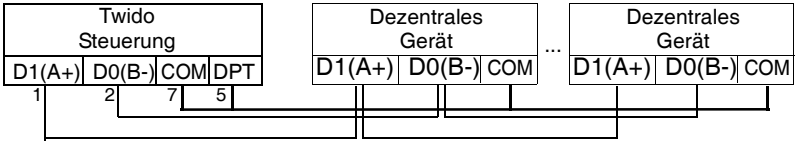
Die Kabelverbindungen zu allen Geräten sind im Folgenden dargestellt.

miniDIN-Anschluss

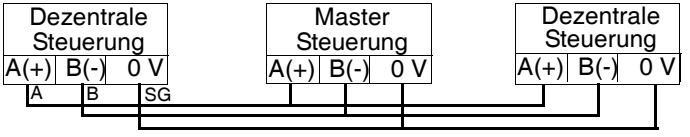
EIA RS-232-Kabel



EIA RS-485-Kabel



Klemmenleistenanschluss



EIA RS-485 Leitungspolarisierung an TWDLCA•40DRF-Steuerungen

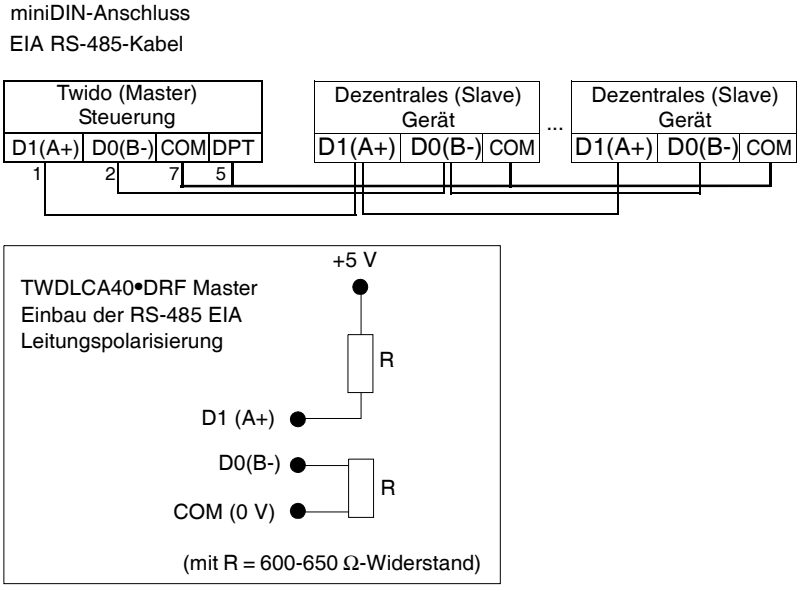
An TWDLCA•40DRF-Steuerungen wird keine interne Vorpolarisierung durchgeführt. Daher ist eine externe Leitungspolarisierung erforderlich, wenn die Modbus-Mastersteueurng TWDLCA•40DRF an das EIA-485 Modbus-Netzwerk angeschlossen wird.
(Wenn keine Datenaktivität auf dem symmetrischen EIA-485-Adernpaar vorhanden ist, dann werden die Leitungen nicht gesteuert und sind daher anfällig gegenüber externem Rauschen oder Interferenzen. Um zu gewährleisten, dass der Verstärker in einem konstanten Status bleibt, wenn kein Datensignal vorhanden ist, muss das Modbus-Mastergerät über eine externe Leitungspolarisierung an das Netzwerk angeschlossen werden.)

Hinweis: Die externe RS-485-Leitungspolarisierung muss nur an der Modbus Master-Steuerung durchgeführt werden; sie darf nicht an einem Slave-Gerät erfolgen.

Das System zur externen Leitungspolarisierung der TWDLCA•40DRF mini-DIN RS-485 EIA-Leitung muss folgende Komponenten umfassen:

- einen Pull-up-Widerstand zu einem 5 V auf D1(A+)-Schaltkreis,
- einen Pull-down-Widerstand zur gemeinsamen Rückleitung am D0(B-)-Schaltkreis.

Die folgende Abbildung zeigt die externe Leitungspolarisierung der TWDLCA•40DRF mini-DIN RS-485 EIA-Leitung.



Die externe Polarisierung kann auf zwei Arten durchgeführt werden:
- Externer Anschluss des vom Benutzer gelieferten Polarisierungssystems über ein mini-DIN-Kabel (siehe Pinbelegung für den Steckverbinder).
- Verwendung eines Polarisierungsabgriffs (konfiguriert für 2-adrige Polarisierung) und eines Polarisierungssystems (bald im Katalog verfügbar).

Softwarekonfiguration

Um die Steuerung so einzurichten, dass sie Zeichen mittels Modbus-Protokoll über eine serielle Verbindung sendet und empfängt, müssen Sie:

Schritt	Beschreibung
1	den seriellen Port mit TwidoSoft für Modbus konfigurieren,
2	in Ihrer Applikation eine Sende-/Empfangstabelle erstellen, die von der Anweisung EXCHx genutzt wird.

Konfigurieren des Ports

Eine Twido-Steuerung kann den primären Port 1 oder einen alternativ konfigurierten Port 2 für das Modbus-Protokoll verwenden. So konfigurieren Sie einen seriellen Port für Modbus:

Schritt	Aktion
1	Definieren Sie alle zusätzlichen Module oder Kommunikations-Adapter, die für die Basis konfiguriert sind.
2	Klicken Sie mit der rechten Maustaste auf den Port, und wählen Sie "SPS-Kommunikations-Setup bearbeiten..." und stellen Sie für den Typ des seriellen Ports "Modbus" ein.
3	Stellen Sie die zugehörigen Kommunikationsparameter ein.

Modbus-Master

Im Modbus-Mastermodus kann die Steuerung einen Modbus-Request an einen Slave senden und auf die Antwort warten. Der Modbus-Mastermodus wird nur über die Anweisung EXCHx unterstützt. Modbus-ASCII und Modbus-RTU werden beide im Modbus-Mastermodus unterstützt.
Die maximale Größe der gesendeten und/oder empfangenen Rahmen lautet 250 Bytes. Die zur Anweisung EXCHx gehörende Worttabelle besteht aus Steuer-, Sende- und Empfangstabellen.

	Höherwertiges Byte	**Niederwertiges Byte**
Steuertabelle	Befehl	Länge (Senden/Empfangen)
	Empfangsverzögerung	Sendeverzögerung
Sendetabelle	Byte 1 gesendet	Byte 2 gesendet
	...	...
	...	Byte n gesendet
	Byte n+1 gesendet	
Empfangstabelle	Byte 1 empfangen	Byte 2 empfangen
	...	...
	...	Byte p empfangen
	Byte p+1 empfangen	

Hinweis: Zusätzlich zu Requests an einzelne Slaves kann die Modbus-Mastersteuerung einen **Broadcast**-Request an alle Slaves senden. Das **Befehls**byte bei einem Broadcast-Request muss auf 00 gesetzt werden, während die **Slave-Adresse** auf 0 gesetzt werden muss.

Steuertabelle

Das **Längen**byte enthält die Länge der Sendetabelle (max. 250 Bytes), die von der Anzahl der empfangenen Zeichen am Ende des Empfangs überschrieben wird, sofern ein Empfang angefordert wird.
Dieser Parameter ist die Länge der Sendetabelle in Bytes. Ist der Sendeverzögerungsparameter gleich Null ist, so ist er gleich der Länge des Sende-Datenübertragungsblocks. Ist der Sendeverzögerungsparameter ungleich Null, wird ein Byte der Sendetabelle (wird vom Offset-Wert angegeben) nicht gesendet, und dieser Parameter ist gleich der Länge des Datenübertragungsblocks plus 1.
Das **Befehls**byte muss bei einer Modbus-RTU-Anforderung (mit Ausnahme von Broadcast) immer gleich 1 sein (Senden und Empfangen).
Das **Sendeverzögerungs**byte enthält den Rang (1 für das erste Byte, 2 für das zweite Byte usw.) in der Sendetabelle des Bytes, der beim Senden der Bytes ignoriert werden soll. Dies dient zum Berücksichtigen der Thematik der Byte-/Wortwerte im Modbus-Protokoll. Enthält dieses Byte beispielsweise den Wert 3, wird das dritte Byte ignoriert, wodurch das vierte Byte in der Tabelle das dritte zu sendende Byte ist.
Das **Empfangsverzögerungs**byte enthält den Rang (1 für das erste Byte, 2 für das zweite Byte usw.) in der Empfangstabelle, der beim Senden der Bytes hinzugefügt werden soll. Dies dient zum Berücksichtigen der Thematik der Byte-/Wortwerte im Modbus-Protokoll. Enthält dieses Byte beispielsweise den Wert 3, wird das dritte Byte in der Tabelle mit NULL gefüllt und das dritte tatsächlich empfangene Byte wird an der vierten Stelle in der Tabelle abgelegt.

Sende-/ Empfangsta-bellen

In beiden Modi (Modbus-ASCII oder Modbus-RTU) wird die Sendetabelle vor der Ausführung der Anweisung EXCHx mit den Inhalt des Requests gefüllt. Zur Ausführungszeit ermittelt die Steuerung die Sicherungsschicht und führt alle erforderlichen Umwandlungen aus, um das Senden und die Antwort zu bearbeiten. Start-, Ende- und Prüfzeichen werden nicht in den Sende-/Empfangstabellen gespeichert.
Nachdem alle Bytes gesendet wurden, schaltet die Steuerung in den Empfangsmodus um und wartet auf den Empfang von Bytes.
Der Empfang wird auf eine der folgenden Arten beendet:

- in einem Zeichen oder Datenübertragungsrahmen wurde eine Verzögerung erkannt,
- das Rahmenendzeichen wird im ASCII-Modus empfangen,
- die Empfangstabelle ist voll.

Die Einträge **Gesendetes Byte X** umfassen Modbus-Protokolldaten (RTU-Codierung), die gesendet werden sollen. Ist der Kommunikations-Port für Modbus-ASCII konfiguriert, werden die entsprechenden Datenübertragungsblockzeichen an die Übertragung angehängt. Das erste Byte enthält die Geräteadresse (spezifisch oder Broadcast), das zweite Byte enthält den Funktionscode, und die übrigen Zeichen enthalten die zum Funktionscode gehörigen Informationen.

Hinweis: Dies ist eine typische Applikation, die jedoch nicht alle Möglichkeiten aufzeigt. Es wird keine Validierung der gesendeten Daten durchgeführt.

Die Einträge **Empfangene Bytes X** umfassen die zu empfangenden Modbus-Protokolldaten (RTU-Codierung). Ist der Kommunikations-Port für Modbus-ASCII konfiguriert, werden die entsprechenden Datenübertragungsblockzeichen in der Antwort gelöscht. Das erste Byte enthält die Geräteadresse, das zweite Byte enthält den Funktionscode (oder Antwortcode) und die übrigen Zeichen enthalten die zum Funktionscode gehörigen Informationen.

Hinweis: Dies ist eine typische Applikation, die jedoch nicht alle Möglichkeiten aufzeigt. Es wird keine Validierung der empfangenen Zeichen durchgeführt, mit Ausnahme der Prüfsummen-Validierung.

Modbus-Slave

Im Modbus-Slavemodus kann die Steuerung auf Modbus-Standardrequests von einem Modbus-Master antworten.
Wenn das Kabel TSXPCX1031 an die Steuerung angeschlossen ist, wird an dem Port die Kommunikation mit Twido begonnen, wobei kurzzeitig der Kommunikationsmodus deaktiviert wird, der vor Anschluss des Kabels aktiviert war.
Das Modbus-Protokoll unterstützt zwei Formate der Sicherungsschicht: ASCII und RTU. Beide Formate werden durch die implementierte Bitübertragungsschicht definiert, wobei ASCII 7 Datenbits und RTU 8 Datenbits verwendet.
Im Modbus-ASCII-Modus wird jedes Byte der Meldung als zwei ASCII-Zeichen gesendet. Der Datenübertragungsblock bei Modbus-ASCII beginnt mit einem Startzeichen (':') und kann mit zwei Endezeichen (CR und LF) enden.
Standardmäßig ist das Zeichen für das Datenübertragungsblockende 0x0A (Zeilenvorschub). Der Anwender kann den Wert dieses Bytes während der Konfiguration ändern. Der Prüfwert für den Datenübertragungsblock bei Modbus-ASCII ist ein einfaches Zweierkomplement des Datenübertragungsblocks ohne die Start- und Endezeichen.
Im Modbus-RTU-Modus wird die Meldung vor dem Senden nicht neu formatiert, doch es wird ein anderer Prüfsummenberechnungsmodus verwendet, der als CRC angegeben ist.
Die Modbus-Sicherungsschicht hat folgende Einschränkungen:

- Adresse 1-247
- Bits: 128 Bits auf Anforderung
- Wörter: 125 Wörter mit 16 Bits auf Anforderung

Meldungsaustausch

Die Sprache bietet für die Kommunikation zwei Funktionen:

- **Anweisung EXCHx:** zum Senden/Empfangen von Meldungen
- **Funktionsblock %MSGx:** zum Steuern des Meldungsaustauschs.

Die Twido-Steuerung nutzt das für den Port eingerichtete Protokoll für die Bearbeitung der Anweisung EXCHx.

Hinweis: Jeder Kommunikations-Port kann für ein unterschiedliches Protokoll oder alle Ports können für das gleiche Protokoll eingerichtet werden. Auf die Anweisung EXCHx oder den Funktionsblock %MSGx für jeden Kommunikations-Port wird über die angehängte Portnummer zugegriffen (1 oder 2).

Anweisung EXCHx

Mit der Anweisung EXCH kann die Twido-Steuerung Informationen von Modbus-Geräten empfangen und Informationen an Modbus-Geräte senden. Der Anwender definiert eine Worttabelle (%MWi:L), die Steuerungsinformationen und die Daten enthalten, die gesendet und/oder empfangen werden sollen (maximal 250 Bytes zum Senden und/oder Empfangen). Das Format der Worttabelle wurde oben beschrieben.
Ein Meldungsaustausch wird mit der Anweisung EXCHx durchgeführt:

Syntax: [EXCHx %MWi:L]

wobei gilt: x = Portnummer (1 oder 2)

L = Anzahl der Wörter in der Steuerwort-, Sende- und Empfangstabelle

Die Twido-Steuerung muss den Austausch einer ersten Anweisung EXCHx beenden, bevor eine zweite Anweisung gestartet werden kann. Der Funktionsblock %MSGx muss verwendet werden, wenn mehrere Meldungen gesendet werden sollen.
Die Bearbeitung der AWL-Anweisung EXCHx geschieht sofort und alle Übertragungen werden interruptgesteuert gestartet (auch der Empfang von Daten läuft interruptgesteuert ab), was als Hintergrundbearbeitung betrachtet wird.

Funktionsblock %MSGx

Der Einsatz des Funktionsblocks %MSGx ist optional. Mit diesem Funktionsblock kann der Datenaustausch verwaltet werden. Der Funktionsblock %MSGx dient drei Zwecken:

- **Kommunikationsfehlerprüfung**
  Die Fehlerprüfung bestätigt, dass der mit der Anweisung EXCHx programmierte Parameter L (Länge der Worttabelle) groß genug ist, um die Länge der zu sendenden Meldung aufzunehmen. Es wird ein Vergleich mit der im niederwertigsten Byte des ersten Worts der Worttabelle programmierten Länge durchgeführt.
- **Koordination mehrerer Meldungen**
  Um beim Senden von mehreren Meldungen die Koordination sicherzustellen, liefert der Funktionsblock %MSGx die Informationen, anhand derer festgestellt werden kann, wann die vorherige Meldung beendet ist.
- **Senden von Prioritätsmeldungen**
  Der Funktionsblock %MSGx lässt es zu, dass die aktuelle Meldungsübertragung gestoppt wird, damit eine dringende Meldung sofort gesendet werden kann.

Der Funktionsblock %MSGx hat einen zugehörigen Eingang und zwei zugehörige Ausgänge:

Eingang/Ausgang	Definition	Beschreibung
R	RESET-Eingang	Auf 1 gesetzt: initialisiert die Kommunikation neu oder setzt den Block zurück (%MSGx.E = 0 und %MSGx.D = 1).
%MSGx.D	Kommunikation beendet	0: Request in Bearbeitung. 1: Kommunikation beendet, wenn Ende der Übertragung, Endezeichen empfangen, Fehler oder Rücksetzen des Blocks.
%MSGx.E	Fehler	0: Meldungslänge OK und Verbindung OK. 1: wenn falscher Befehl, Tabelle falsch konfiguriert, falsches Zeichen empfangen (Geschwindigkeit, Parität usw.) oder Empfangstabelle voll.

Einschränkungen

Es ist wichtig, folgende Einschränkungen zu beachten:

- Die Überprüfung des Vorhandenseins von Port 2 und dessen Konfiguration (RS232 oder RS 485) wird beim Einschalten oder beim Zurücksetzen durchgeführt.
- Die Meldungsbearbeitung an Port 1 wird abgebrochen, wenn TwidoSoft verbunden wird.
- EXCHx oder %MSG kann nicht an einem Port bearbeitet werden, der als dezentrale Verbindung konfiguriert ist.
- EXCHx bricht aktive Modbus-Slave-Bearbeitung ab.
- Die Bearbeitung der Anweisungen EXCHx wird im Fehlerfall nicht wiederholt.
- Mit dem Rücksetzeingang kann die Empfangsbearbeitung der Anweisung EXCHx beendet werden.
- Anweisungen EXCHx können mit einem Timeout zum Beenden des Empfangs konfiguriert werden.
- Mehrere Meldungen werden über %MSGx.D gesteuert.

Fehler- und Betriebszustandsbedingungen

Tritt bei Verwendung einer Anweisung EXCHx ein Fehler auf, werden die Bits %MSG.D und %MSG.E auf 1 gesetzt und das Systemwort %SW63 enthält den Fehlercode für Port 1 und das Systemwort %SW64 enthält den Fehlercode für Port 2.

Systemwörter	Verwendung
%SW63	Fehlercode EXCH1: 0 - Vorgang war erfolgreich 1 - Anzahl zu sendender Bytes zu hoch (> 250) 2 - Sendetabelle zu klein 3 - Worttabelle zu klein 4 - Überlauf Empfangstabelle 5 - Timeout abgelaufen 6 - Senden 7 - Falscher Befehl in Tabelle 8 - Ausgewählter Port nicht konfiguriert/verfügbar 9 - Empfangsfehler 10 - %KW kann nicht beim Empfang verwendet werden 11 - Sende-Offset größer als Sendetabelle 12 - Empfangs-Offset größer als Empfangstabelle 13 - Steuerung hat Bearbeitung von EXCH gestoppt
%SW64	Fehlercode EXCH2: Siehe %SW63.

Neustart der Master-Steuerung

Startet eine Master/Slave-Steuerung neu, tritt eines der folgenden Ereignisse auf:

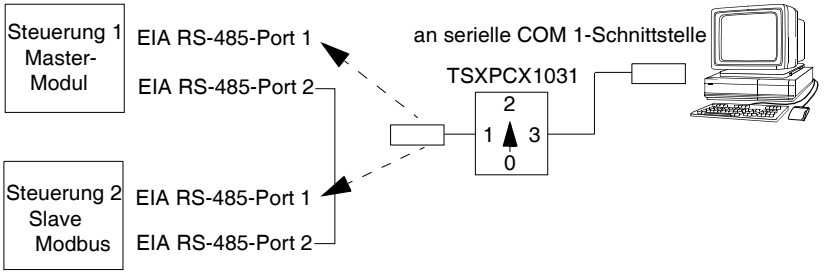
- Ein Kaltstart (%S0 = 1) forciert eine Reinitialisierung der Kommunikation.
- Ein Warmstart (%S1 = 1) forciert eine Reinitialisierung der Kommunikation.
- Im Betriebszustand STOP stoppt die Steuerung sämtliche Modbus-Kommunikation.

Beispiel 1 für eine Modbus-Verbindung

Zum Konfigurieren einer Modbus-Verbindung müssen Sie:
1. Die Hardware konfigurieren.
2. Das Modbus-Kommunikationskabel anschließen.
3. Den Port konfigurieren.
4. Eine Applikation schreiben.
5. Den Animationstabellen-Editor initialisieren.

Die folgenden Abbildungen zeigen die Verwendung des Modbus-Requests 3 zum Lesen der Ausgangswörter eines Slave. In diesem Beispiel werden zwei Twido-Steuerungen verwendet.

Schritt 1: Hardwarekonfiguration:



Die Hardwarekonfiguration besteht aus zwei Twido-Steuerungen. Die eine wird als Modbus-Master und die andere als Modbus-Slave konfiguriert.

Hinweis: In diesem Beispiel werden beide Steuerungen für die Verwendung von EIA RS-485 an Schnittstelle 1 und optional EIA RS-485 an Schnittstelle 2 konfiguriert. Bei einer modularen Steuerung kann die optionale Schnittstelle 2 entweder TWDNOZ485D oder TWDNOZ485T sein. Wenn Sie TWDXCPODM verwenden, kann die Schnittstelle eine TWDNAC485D oder eine TWDNAC485T sein. Bei einer kompakten Steuerung kann der optionale Port 2 entweder TWDNAC485D oder TWDNAC485T sein.

Zum Konfigurieren jeder Steuerung schließen Sie das Kabel TSXPCX1031 an Port 1 der Steuerung an.

Hinweis: Das Kabel TSXPCX1031 kann nur an jeweils eine Steuerung und nur am EIA RS-485-Port 1 angeschlossen werden.

Dann schließen Sie das Kabel an den COM 1-Port des PC an. Achten Sie darauf, dass sich das Kabel in der Schalterstellung 2 befindet. Laden und beobachten Sie die Applikation. Wiederholen Sie diese Vorgehensweise für die zweite Steuerung.

Schritt 2: Schließen Sie das Modbus-Kommunikationskabel an:

miniDIN-Anschluss

Klemmenleistenanschluss

Die Verdrahtung in diesem Beispiel zeigt eine einfache Punkt-zu-Punkt-Verbindung. Die drei Signale D1(A+), D0(B-) und COM(0 V) werden entsprechend der Abbildung verdrahtet.
Wenn Sie den Port 1 der Twido Steuerung verwenden, muss das DPT-Signal (Pin 5) mit der gemeinsamen Rückleitung (Pin 7) verbunden werden. Dieser Zustand des DPT-Signals legt fest, ob TwidoSoft angeschlossen ist. Ist das Signal an Erde angeschlossen, nutzt die Steuerung die in der Applikation eingestellte Port-Konfiguration, um die Art der Kommunikation zu ermitteln.

Schritt 3: Port-Konfiguration:

Hardware -> Option hinzufügen TWDNOZ485-

Hardware => Komm.-Setup Einstellung	
Serieller Port 2	
Protokoll	Modbus
Adresse	1
Baudrate	19200
Datenbits	8 (RTU)
Parität	Keine
Stoppbit	1
Antwort-Zeitüberschreitung (x100 ms)	10
Zeit zwischen Frames (ms)	10

Hardware -> Option hinzufügen TWDNOZ485-

Hardware => Komm.-Setup Einstellung	
Serieller Port 2	
Protokoll	Modbus
Adresse	2
Baudrate	19200
Datenbits	8 (RTU)
Parität	Keine
Stoppbit	1
Antwort-Zeitüberschreitung (x100 ms)	100
Zeit zwischen Frames (ms)	10

In beiden Applikationen, in der Master- und der Slave-Applikation, sind die optionalen EIA RS-485-Ports konfiguriert. Vergewissern Sie sich, dass die Kommunikationseinstellungen der Steuerung im Modbus-Protokoll geändert wurden und unterschiedliche Adressen konfiguriert sind.

In diesem Beispiel ist für den Master die Adresse 1 und für den Slave die Adresse 2 eingestellt. Die Anzahl der Bits ist auf 8 gesetzt, wodurch angezeigt wird, dass der Modbus-RTU-Modus verwendet wird. Wäre eine Anzahl von 7 Bits eingestellt, würde der Modbus-ASCII-Modus verwendet werden. Eine weitere Voreinstellung, die geändert wurde, ist die Erhöhung des Timeout-Werts auf 1 Sekunde.

Hinweis: Weil der Modbus-RTU-Modus eingestellt wurde, wird der Parameter "Datenübertragungsblockende" ignoriert.

Schritt 4: Schreiben der Applikation:

```
LD 1
[%MW0 := 16#0106]
[%MW1 := 16#0300]
[%MW2 := 16#0203]
[%MW3 := 16#0000]
[%MW4 := 16#0004]
LD 1
AND %MSG2.D
[EXCH2 %MW0:11]
LD %MSG2.E
ST %Q0.0
END
```

```
LD 1
[%MW0 := 16#6566]
[%MW1 := 16#6768]
[%MW2 := 16#6970]
[%MW3 := 16#7172]
END
```

In TwidoSoft wird ein Anwendungsprogramm für den Master und eines für den Slave geschrieben. Für den Slave werden einfach einige Speicherwörter an bekannte Werte geschrieben. Für den Master wird die Worttabelle der Anweisung EXCHx so initialisiert, dass 4 Wörter aus dem Slave an die Modbus-Adresse 2 mit Beginn an Position %MW0 gelesen werden.

Hinweis: Beachten Sie die Verwendung der Sendeverzögerung in %MW1 beim Modbus-Master. Der Offset mit dem Wert 3 ergänzt ein Byte (Wert = 0) an der dritten Stelle im Empfangsbereich der Tabelle. Dadurch werden die Wörter im Master ausgerichtet, so dass sie korrekt auf Wortgrenzen fallen. Ohne diesen Offset würde jedes Datenwort zwischen zwei Wörtern im Austauschblock aufgeteilt werden. Der Offset wird zur Vereinfachung eingesetzt.

Bevor die Anweisung EXCH2 ausgeführt wird, prüft die Applikation das mit %MSG2 verknüpfte Kommunikationsbit. Zum Schluss wird der Fehlerstatus von %MSG2 abgefragt und im ersten Ausgangsbit der lokalen Basissteuerungs-E/A gespeichert. Es könnte eine weitere Fehlerprüfung mit %SW64 durchgeführt werden, um eine höhere Genauigkeit zu erreichen.

Schritt 5: Initialisierung des Animationstabellen-Editors im Master:

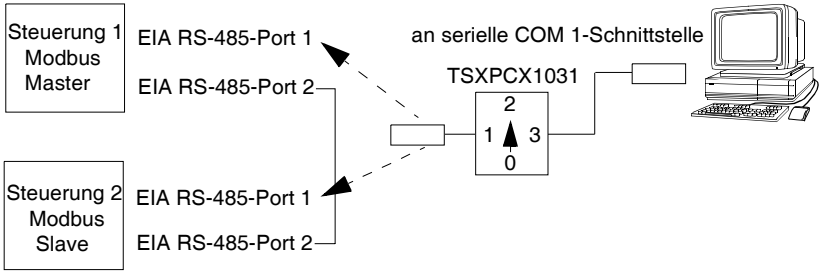
Aktuell	gespeichertes	Adressformat
1 %MW5	0203	0000 Hexadezimal
2 %MW6	0008	0000 Hexadezimal
3 %MW7	6566	0000 Hexadezimal
4 %MW8	6768	0000 Hexadezimal
5 %MW9	6970	0000 Hexadezimal
6 %MW10	7172	0000 Hexadezimal

Nach dem Ladevorgang und wenn die Steuerungen in den Betriebszustand RUN versetzt wurden, öffnen Sie eine Animationstabelle auf dem Master. Prüfen Sie im Antwortabschnitt in der Tabelle, dass der Antwortcode 3 ist und dass die richtige Anzahl an Bytes gelesen wurde. Beachten Sie in diesem Beispiel außerdem, dass die aus dem Slave gelesenen Wörter (mit Beginn an %MW7) korrekt mit den Wortgrenzen im Master ausgerichtet werden.

Beispiel 2 für eine Modbus-Verbindung

Die folgende Abbildung zeigt die Verwendung des Modbus-Requests 16 zum Schreiben der Ausgangswörter eines Slave. In diesem Beispiel werden zwei Twido-Steuerungen verwendet.

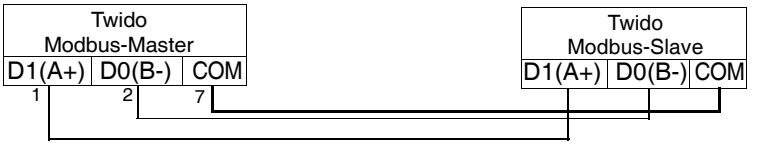
Schritt 1: Hardwarekonfiguration:



Die Hardwarekonfiguration ist die gleiche wie im vorigen Beispiel.

Schritt 2: Anschluss des Modbus-Kommunikationskabels (RS-485):

miniDIN-Anschluss



Klemmenleistenanschluss

Die Verkabelung für die Modbus-Kommunikation ist die gleiche wie im vorigen Beispiel.

Schritt 3: Port-Konfiguration:

Hardware -> Option hinzufügen TWDNOZ485-

Hardware => Komm.-Setup Einstellung	
Serieller Port 2	
Protokoll	Modbus
Adresse	1
Baudrate	19200
Datenbits	8 (RTU)
Parität	Keine
Stoppbit	1
Antwort-Zeitüberschreitung (x100 ms)	10
Zeit zwischen Frames (ms)	10

Hardware -> Option hinzufügen TWDNOZ485-

Hardware => Komm.-Setup Einstellung	
Serieller Port 2	
Protokoll	Modbus
Adresse	2
Baudrate	19200
Datenbits	8 (RTU)
Parität	Keine
Stoppbit	1
Antwort-Zeitüberschreitung (x100 ms)	100
Zeit zwischen Frames (ms)	10

Die Port-Konfigurationen sind die gleichen wie im vorigen Beispiel.

Schritt 4: Schreiben einer Applikation:

```
LD 1
[%MW0 := 16#010C]
[%MW1 := 16#0007]
[%MW2 := 16#0210]
[%MW3 := 16#0010]
[%MW4 := 16#0002]
[%MW5 := 16#0004]
[%MW6 := 16#6566]
[%MW7 := 16#6768]
LD 1
AND %MSG2.D
[EXCH2 %MW0:11]
LD %MSG2.E
ST %Q0.0
END
```

```
LD 1
[%MW18 := 16#FFFF]
END
```

In TwidoSoft wird ein Anwendungsprogramm für den Master und eines für den Slave erstellt. Schreiben Sie für den Slave ein einzelnes Speicherwort %MW18. Dadurch wird im Slave Speicherplatz für die Speicheradressen von %MW0 bis %MW18 geschaffen. Ohne diese Zuordnung würde der Modbus-Request versuchen, in Adressen zu schreiben, die im Slave nicht vorhanden sind.
Für den Master wird die Worttabelle der Anweisung EXCH2 so initialisiert, dass 4 Bytes in den Slave an der Modbus-Adresse 2 mit Beginn an Position %MW16 (10 hexadezimal) geschrieben werden.

Hinweis: Beachten Sie die Verwendung der Empfangsverzögerung in %MW1 der Applikation des Modbus-Masters. Der Offset 7 unterdrückt das oberste Byte im sechsten Wort (der Wert 00 hexadezimal in %MW5). Dadurch werden die Datenwerte in der Sendetabelle der Worttabelle so ausgerichtet, dass sie korrekt auf Wortgrenzen fallen.

Bevor die Anweisung EXCH2 ausgeführt wird, prüft die Applikation das mit %MSG2 verknüpfte Kommunikationsbit. Zum Schluss wird der Fehlerstatus von %MSG2 abgefragt und im ersten Ausgangsbit der lokalen Basissteuerungs-E/A gespeichert. Es könnte eine weitere Fehlerprüfung mit %SW64 durchgeführt werden, um eine höhere Genauigkeit zu erreichen.
Schritt 5: Initialisierung des Animationstabellen-Editors:

Erstellen Sie die folgende Animationstabelle im Master:

Aktuell	gespeichertes		Adressformat
1 %MW0	010C	0000	Hexadezimal
2 %MW1	0007	0000	Hexadezimal
3 %MW2	0210	0000	Hexadezimal
4 %MW3	0010	0000	Hexadezimal
5 %MW4	0002	0000	Hexadezimal
6 %MW5	0004	0000	Hexadezimal
7 %MW6	6566	0000	Hexadezimal
8 %MW7	6768	0000	Hexadezimal
9 %MW8	0210	0000	Hexadezimal
10 %MW9	0010	0000	Hexadezimal
11 %MW10	0004	0000	Hexadezimal

Erstellen Sie die folgende Animationstabelle im Slave:

Aktuell	gespeichertes		Adressformat
1 %MW16	6566	0000	Hexadezimal
2 %MW17	6768	0000	Hexadezimal

Nach dem Ladevorgang und wenn die Steuerungen in den Betriebszustand RUN versetzt wurden, öffnen Sie eine Animationstabelle in der Slave-Steuerung. Die beiden Werte in %MW16 und %MW17 werden in den Slave geschrieben. Im Master kann mit der Animationstabelle der Empfangsteil der Austauschdaten geprüft werden. Diese Daten zeigen die Slave-Adresse, den Antwortcode, das erste geschriebene Wort und die Anzahl der geschriebenen Wörter mit Beginn an %MW8 im Beispiel oben.

Modbus-Standard-Requests

Einleitung

Diese Requests werden verwendet, um Speicherwörter oder -bits zwischen dezentralen Geräten auszutauschen. Das Tabellenformat ist für den RTU- und den ASCII-Modus identisch.

Format	Artikelnummer
Bit	%Mi
Wort	%MWi

Modbus-Master: Lesen von N Bits

Diese Tabelle beschreibt die Requests 01 und 02.

	Tabelle Index	Höherwertiges Byte	Niederwertiges Byte
Steuertabelle	0	01 (Senden/Empfangen)	06 (Sendelänge) (*)
	1	03 (Empfangsverzögerung)	00 ´(Sendeverzögerung)
Sendetabelle	2	Slave@(1..247)	01 oder 02 (Request-Code)
	3	Adresse des ersten zu lesenden Bits	
	4	N_1 = Anzahl der zu lesenden Bits	
Empfangstabelle (nach Antwort)	5	Slave@(1..247)	01 oder 02 (Antwortcode)
	6	00 (infolge einer Empfangsverzögerung hinzugefügtes Byte)	N_2 = Anzahl der zu lesenden Datenbytes = [1+(N_1-1)/8], wobei [] den integralen Teil angibt
	7	Wert des 1. Bytes (Wert = 00 oder 01)	Wert des 2. Bytes (wenn N_1>1)
	8	Wert des 3. Bytes (wenn N_1>1)	
	...		
(N_2/2)+6 (wenn N_2 gerade ist) (N_2/2+1)+6 (wenn N_2 ungerade ist)		Wert des N_2ten Bytes (wenn N_1>1)	

(*) Dieses Byte empfängt nach der Antwort auch die Länge der gesendeten Zeichenkette

Modbus-Master: Lesen von N Wörtern

Diese Tabelle beschreibt die Requests 03 und 04.

	Tabelle Index	Höherwertiges Byte	Niederwertiges Byte
Steuertabelle	0	01 (Senden/Empfangen)	06 (Sendelänge) (*)
	1	03 (Empfangsverzögerung)	00 ´(Sendeverzögerung)
Sendetabelle	2	Slave@(1..247)	03 oder 04 (Request-Code)
	3	Adresse des ersten zu lesenden Wortes	
	4	N = Anzahl der zu lesenden Wörter	
Empfangstabelle (nach Antwort)	5	Slave@(1..247)	03 oder 04 (Antwortcode)
	6	00 (infolge einer Empfangsverzögerung hinzugefügtes Byte)	2*N (Anzahl gelesener Bytes)
	7	Erstes gelesenes Wort	
	8	Zweites gelesenes Wort (wenn N>1)	
	...		
	N+6	N-tes gelesenes Wort (wenn N>2)	

(*) Dieses Byte empfängt nach der Antwort auch die Länge der gesendeten Zeichenkette

Hinweis: Die Operation Empfangsverzögerung = 3 fügt ein Byte (Wert = 0) an der dritten Stelle der Empfangstabelle hinzu. Dies gewährleistet eine gute Anordnung der Anzahl der gelesenen Bytes und der Werte der gelesenen Wörter in dieser Tabelle.

Modbus-Master: Schreiben eines Bits

Diese Tabelle stellt Request 05 dar.

	Tabelle Index	Höherwertiges Byte	Niederwertiges Byte
Steuertabelle	0	01 (Senden/Empfangen)	06 (Sendelänge) (*)
	1	00 (Empfangsverzögerung)	00 ´(Sendeverzögerung)
Sendetabelle	2	Slave@(1..247)	05 (Request-Code)
	3	Adresse des zu schreibenden Bits	
	4	Zu schreibender Bitwert	
Empfangstabelle (nach Antwort)	5	Slave@(1..247)	05 (Antwortcode)
	6	Adresse des geschriebenen Bits	
	7	Geschriebener Wert	

(*) Dieses Byte empfängt nach der Antwort auch die Länge der gesendeten Zeichenkette

Hinweis:
- Dieser Request benötigt keinen Offset.
- Der Antwort-Datenübertragungsblock ist der gleiche wie der Request-Datenübertragungsblock hier (im Normalfall).
- Um einem Bit den Wert 1 zuzuweisen, muss das zugehörige Wort in der Sendetabelle den Wert FF00H aufweisen. Um einem Bit den Wert 0 zuzuweisen, muss das zugehörige Wort den Wert 0 aufweisen.

Modbus-Master: Schreiben eines Wortes

Diese Tabelle stellt Request 06 dar.

	Tabelle Index	Höherwertiges Byte	Niederwertiges Byte
Steuertabelle	0	01 (Senden/Empfangen)	06 (Sendelänge) (*)
	1	00 (Empfangsverzögerung)	00 ´(Sendeverzögerung)
Sendetabelle	2	Slave@(1..247)	06 (Request-Code)
	3	Adresse des zu schreibenden Wortes	
	4	Zu schreibender Wortwert	
Empfangstabelle (nach Antwort)	5	Slave@(1..247)	06 (Antwortcode)
	6	Adresse des geschriebenen Wortes	
	7	Geschriebener Wert	

(*) Dieses Byte empfängt nach der Antwort auch die Länge der gesendeten Zeichenkette

Hinweis:
- Dieser Request benötigt keinen Offset.
- Der Antwort-Datenübertragungsblock ist der gleiche wie der Request-Datenübertragungsblock hier (im Normalfall).

Modbus-Master: Schreiben von N Bits

Diese Tabelle stellt Request 15 dar.

	Tabelle Index	Höherwertiges Byte	Niederwertiges Byte
Steuertabelle	0	01 (Senden/Empfangen)	8 + Anzahl Bytes (Senden)
	1	00 (Empfangsverzögerung)	07 ´(Sendeverzögerung)
Sendetabelle	2	Slave@(1..247)	15 (Request-Code)
	3	Nummer des ersten zu schreibenden Bits	
	4	N_1 = Anzahl der zu schreibenden Bits	
	5	00 (Byte nicht gesendet, Offset-Auswirkung)	N_2 = Anzahl der zu schreibenden Datenbytes = [1+(N_1-1)/8], wobei [] den integralen Teil angibt
	6	Wert des 1. Bytes	Wert des 2. Bytes
	7	Wert des 3. Bytes	Wert des 4. Bytes
	...		
	(N_2/2)+5 (wenn N_2 gerade ist) (N_2/2+1)+5 (wenn N_2 ungerade ist)	Wert des N_2^{ten} Bytes	
Empfangstabelle (nach Antwort)		Slave@(1..247)	15 (Antwortcode)
		Adresse des 1. geschriebenen Bits	
		Adresse der geschriebenen Bits (= N_1)	

Hinweis:

- Die Sendeverzögerung = 7 unterdrückt das siebte Byte im gesendeten Datenübertragungsblock. Ermöglichen Sie eine gute Entsprechung der Wortwerte in der Sendetabelle.

Modbus-Master: Schreiben von N Wörtern

Diese Tabelle stellt Request 16 dar.

	Tabelle Index	Höherwertiges Byte	Niederwertiges Byte
Steuertabelle	0	01 (Senden/Empfangen)	8 + (2*N) (Sendelänge)
	1	00 (Empfangsverzögerung)	07 ´(Sendeverzögerung)
Sendetabelle	2	Slave@(1..247)	16 (Request-Code)
	3	Adresse des ersten zu schreibenden Wortes	
	4	N = Anzahl der zu schreibenden Wörter	
	5	00 (Byte nicht gesendet, Offset-Auswirkung)	2*N = Anzahl zu schreibender Bytes
	6	Erster zu schreibender Wortwert	
	7	Zweiter zu schreibender Wert	
	...		
	N+5	Zu schreibender Wert N	
Empfangstabelle (nach Antwort)	N+6	Slave@(1..247)	16 (Antwortcode)
	N+7	Adresse des ersten zu schreibenden Wortes	
	N+8	Adresse der geschriebenen Wörter (= N)	

Hinweis: Die Sendeverzögerung = 7 unterdrückt das fünfte MMSB-Byte im gesendeten Datenübertragungsblock. Ermöglichen Sie eine gute Entsprechung der Wortwerte in der Sendetabelle.

Transparent Ready Implementation Class (Twido Serial A05, Ethernet A15)

Überblick

Die folgenden Modbus-Funktionscodes werden sowohl vom seriellen Modbus als auch von TCP/IP Modbus unterstützt. Ausführliche Informationen über das Modbus-Protokoll finden Sie im Dokument *Modbus-Anwendungsprotokoll*, das unter http://www.modbus-ida.org verfügbar ist.

Von Twido unterstützte Modbus-Funktionscodes (MB FC)

Die folgende Tabelle beschreibt die sowohl vom seriellen Twido als auch von TCP/IP Modbus unterstützten Funktionscodes.

Unterstützte MB FC	Unterstützte Unterfunktionscodes	Funktion
1	—	Mehrere interne Bits lesen %M
2	—	Mehrere interne Bits lesen %M
3	—	Mehrere interne Register lesen %MW
4	—	Mehrere interne Register lesen %MW
5	—	Einzelnes internes Bit forcieren %M
6	—	Einzelne interne Register schreiben %M"
8	nur 00	Echo-Diagnose
15	—	Mehrere interne Bits schreiben %M
16	—	Mehrere interne Register schreiben %MW
23	—	Mehrere interne Register lesen/schreiben %MW
43	14	Geräte-ID lesen (regulärer Dienst)

Ethernet TCP/IP-Kommunikations-Überblick

Ethernet-Merkmale

Nachfolgend sind die Ethernet-fähigen Leistungsmerkmale des Twido-SPS-Grundgeräts TWDLCAE40DRF beschrieben.
Das SPS-Grundgerät TWDLCAE40DRF ist ein Ethernet-fähiges Gerät, das das Modbus Application Protocol (MBAP) über TCP/IP implementiert. Modbus TCP/IP bietet eine Kommunikation zwischen gleichrangigen Steuerungen über das Netzwerk in einer Client/Server-Topologie.

Frame-Format

Das Twido-SPS-Grundgerät TWDLCAE40DRF unterstützt nur das Ethernet II-Frame-Format. Es unterstützt nicht die IEEE802.3-Rahmung. Bitte beachten Sie, dass andere von Schneider Electric erhältliche SPS wie etwa die Premium- und Quantum-Serien sowohl das Ethernet II- als auch das IEEE802.3-Frame-Format unterstützen und die Frame-Formate wählbar sind. Wenn Sie also planen, Ihre Twido-Steuerung in Verbindung mit Premium- oder Quantum-SPSs zu verwenden, sollten Sie sie für die Verwendung des Ethernet II-Frame-Formats konfigurieren, um eine optimale Kompatibilität zu gewährleisten.

TCP-Verbindungen

Die kompakte Steuerung TWDLCAE40DRF ist ein Gerät mit 4 simultanen Kanälen, das über ein 100Base-TX Ethernet-Netzwerk kommunizieren kann. Es implementiert die automatische 100Base-TX-Einstellung und kann auch in einem 10Base-T-Netzwerk eingesetzt werden. Darüber hinaus ermöglicht es eine markierte IP-Verbindung wie im TwidoSoft-Applikationsprogramm konfiguriert (siehe *Registerkarte "Markiertes IP", S. 179* für ausführliche Informationen über Markiertes IP).
Es wird maximal 1 Server-Transaktion je TCP-Verbindung von der Twido-Steuerung unterstützt.

IP-Adresse

Die TWDLCAE40DRF-Steuerung führt BootP aus, eine IP Adresse von einem BootP Diener zu erhalten. Für mehr Flexibilität haben Sie noch die Fähigkeit, einen statischen IP Adresse durch TwidoSoft Programmierung Software anzugebe, sowie die Festlegung von Subnetz- und Gateway-IP-Adressen.
ööAußerdem wenn die TWDLCAE40DRF-Steuerung zu erhalten einen gültigen IP Adresse vom BootP Diener (versagt, oder wenn es entdeckt einen doppelten IP Adresse wenn Sie einen statischen IP Adresse) zuteilen, geht das Steuergerät in fallback Modus und benutzt die Standard IP Adresse. Jedem SPS-Grundgerät TWDLCAE40DRF ist eine physikalischen MAC-Adresse (IEEE Globale Adresse) die permanent in der kompakten Steuerung gespeichert zugeordnet. Die standard-mäßige IP-Adresse des Geräts ist von der eindeutigen physikalischen MAC-Adresse abgeleitet.

Hinweis: Wenn die Standard IP Adresse benutzt ist, ist BootP-Kundedienst deaktiviert.

Modbus TCP Client/Server

Eine TWDLCAE40DRF-Steuerung kann abhängig davon, ob sie einen Request an ein dezentrales Gerät richtet bzw. einen Request von einem dezentralen Gerät beantwortet, sowohl Modbus TCP/IP-Client als auch -Server sein. Der TCP-Nachrichtenübertragungsdienst wird über den TCP-Port 502 implementiert.

- Der Modbus-Server implementiert die den Schneider Transparent Ready Class Messaging TR A15 Standard.
- Der Modbus-Client wird über die Anweisung EXCH3 und die Funktion %MSG3 implementiert. Sie können mehrere EXCH3-Anweisungen programmieren, aber es kann jeweils nur eine EXCH3 aktiv sein. Die TCP-Verbindung wird von der kompakten Steuerung automatisch eingestellt, sobald die Anweisung %EXCH3 aktiv ist.
  Der Modbus-Client implementiert die den Schneider Transparent Ready Class Messaging TR A10 Standard.

TCP/IP-Konfigurationsanleitung für die Ethernet-Kommunikation zwischen PC und Steuerung

Umfang

Diese TCP/IP-Konfigurationsanleitung enthält Informationen zur Ethernet-Vernetzung und TCP/IP-Konfiguration, um die Kommunikation zwischen Ihrem PC, auf dem die TwidoSoft-Applikation ausgeführt wird, und der Twido-Steuerung über ein autonomes Ethernet-Netzwerk einzurichten.

Überprüfen der aktuellen IP-Einstellungen Ihres PC

Nachfolgend ist die Vorgehensweise zur Überprüfung der aktuellen IP-Einstellungen Ihres PC beschrieben. Diese Vorgehensweise ist für alle Versionen des Windows-Betriebssystems gültig.

Schritt	Aktion
1	Wählen Sie aus dem Windows **Startmenü** die Option **Ausführen** aus.
2	Geben Sie im Dialogfeld "Ausführen" in das Textfeld **Öffnen** den Befehl **"command"** ein. **Ergebnis:** Die Eingabeaufforderung **C:\WINDOWS\system32\command.com** wird angezeigt.
3	Geben Sie **"ipconfig"** in die Eingabeaufforderung ein.
4	Die Windows-**IP-Konfiguration** wird mit den folgenden Parametern angezeigt: IP-Adresse.....: Subnetzmaske.....: Standardgateway.....: **Hinweis:** Die oben aufgeführten IP-Einstellungen können nicht direkt an der Eingabeaufforderung geändert werden. Sie werden nur zu Referenzzwecken angezeigt. Lesen Sie den folgenden Abschnitt, wenn Sie die IP-Konfiguration Ihres PC ändern möchten.

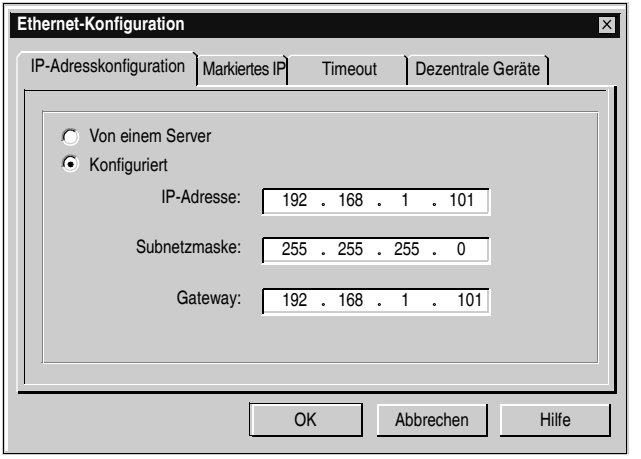
Konfigurieren der TCP/IP-Einstellungen Ihres PC

Die folgenden Informationen unterstützen Sie bei der Konfiguration der TCP/IP-Einstellungen Ihres PC, auf dem die TwidoSoft-Applikation zur Programmierung und Steuerung der Twido-Steuerung über das Netzwerk ausgeführt wird. Das nachfolgend beschriebene Verfahren kann auf einem mit dem Betriebssystem Windows XP ausgestatteten PC nachvollzogen werden, dient jedoch lediglich als Beispiel. (Die Vorgehensweise für andere Betriebssysteme finden Sie in den TCP/IP-Konfigurationsanleitungen, die im Benutzerhandbuch des auf Ihrem PC installierten Betriebssystems aufgeführt sind.)

Schritt	Aktion
Hinweis: Wenn Ihr PC bereits installiert ist und die Ethernet-Karte über das bestehende autonome Netzwerk konfiguriert ist, müssen Sie die IP-Adresseinstellungen nicht ändern (überspringen Sie in diesem Fall die Schritte 1-6 und fahren Sie mit dem folgenden Abschnitt fort). Befolgen Sie die Schritte 1-6 dieses Verfahrens nur, wenn Sie die TCP/IP-Einstellungen des PC ändern möchten.	
1	Wählen Sie **Systemsteuerung > Netzwerkverbindungen** aus dem Windows **Startmenü** aus.
2	Klicken Sie mit der rechten Maustaste auf die **LAN-Verbindung** (das autonome Netzwerk), in der Sie die Twido-Steuerung installieren möchten, und wählen Sie dann die Option **Eigenschaften** aus.
3	Wählen Sie **TCP/IP** aus der Liste der installierten Netzwerkkomponenten aus und klicken Sie auf **Eigenschaften**. **Hinweis:** Wenn sich TCP/IP nicht in der Liste der installierten Komponenten befindet, ziehen Sie das Benutzerhandbuch Ihres Betriebssystems zu Rate, um Informationen über die Installation der TCP/IP-Netzwerkkomponente zu erhalten.
4	Das Dialogfeld **Eigenschaften von Internetprotokoll (TCP/IP)** wird angezeigt. Es enthält die aktuellen TCP/IP-Einstellungen Ihres PC einschließlich der **IP-Adresse** und der **Subnetzmaske**. **Hinweis:** Verwenden Sie in einem autonomen Netzwerk nicht die Option **IP-Adresse automatisch beziehen**. Das runde Optionsfeld **Folgende IP-Adresse** verwenden muss ausgewählt werden, und die Felder "IP-Adresse" und "Subnetzmaske" müssen gültige IP-Einstellungen enthalten.
5	Geben Sie eine gültige **statische IP-Adresse** in Dezimalpunktschreibweise ein. Bei einem autonomen Netzwerk raten wir Ihnen, eine IP-Adresse der Netzwerkklasse C anzugeben (siehe *IP-Adressierung, S. 169*). `192.168.1.198` ist beispielsweise eine IP-Adresse der Klasse C. **Hinweis:** Die von Ihnen angegebene IP-Adresse muss mit der Netzwerk-ID des vorhandenen Netzwerks kompatibel sein. Wenn das vorhandene Netzwerk beispielsweise die IP-Adressen `192.168.1.xxx` unterstützt (wobei `192.168.1` die Netzwerk-ID und `xxx = 0-255` die Host-ID ist), dann können Sie `191.168.1.198` als eine gültige IP-Adresse für Ihren PC angeben. (Vergewissern Sie sich, dass die Host-ID `198` im Netzwerk eindeutig ist.)
6	Geben Sie eine gültige **Subnetzmaske** in Dezimalpunktschreibweise ein. Wenn keine Subnetzmasken in Ihrem Netzwerk der Klasse C verwendet wird, empfehlen wir die Eingabe einer Standard-Subnetzmaske für Netzwerke der Klasse C wie etwa `255.255.255.0`.

Konfigurieren der TCP/IP-Einstellungen Ihrer Twido-Steuerung

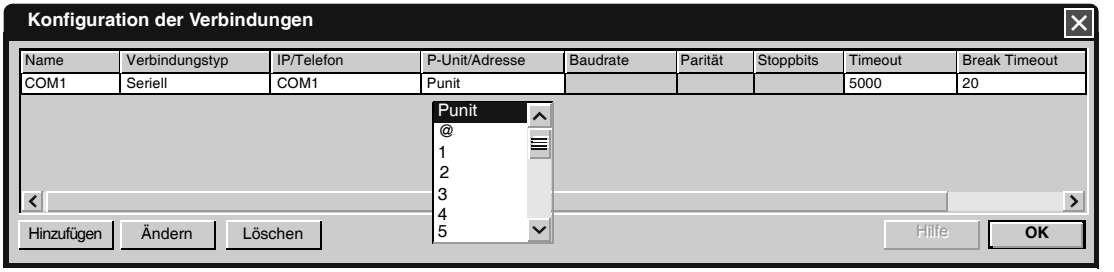
Wenn Sie die TCP/IP-Einstellungen Ihres PC, auf dem die TwidoSoft-Applikation ausgeführt wird, konfiguriert haben, müssen Sie die TCP/IP-Einstellungen der Twido-Steuerung konfigurieren, mit der TwidoSoft über das Netzwerk kommunizieren soll. Gehen Sie hierzu wie folgt vor:

Schritt	Aktion
1	Schließen Sie ein serielles Kabel (TSXPCX1031) zwischen dem PC, auf dem TwidoSoft ausgeführt wird, und dem RS-485-Port der Twido-Steuerung an.
2	Starten Sie das Applikationsprogramm **TwidoSoft** auf Ihrem PC.
3	Wählen Sie eine neue **Hardware** aus dem TwidoSoft-Applikationsnavigator aus, und wählen Sie dann die Steuerung **TWDLCAE40DRF** aus.
4	Wählen Sie **SPS > Verbindung auswählen** aus der TwidoSoft-Menüleiste und anschließend den Port **COM1** aus.
5	Doppelklicken Sie auf das Symbol **Ethernet-Port** im TwisoSoft-Applikationsnavigator (oder wählen Sie **Hardware > Ethernet** aus der Menüleiste aus), um das nachfolgend abgebildete Dialogfeld **Ethernet-Konfiguration** aufzurufen. 
6	Wählen Sie auf der Registerkarte **IP-Adresskonfiguration** das runde Optionsfeld **Konfiguriert** aus und beginnen Sie die Konfiguration der Felder "IP-Adresse", "Subnetzmaske" und "Gateway" wie in den Schritten 7-9 erläutert. **Hinweis:** Hier wird nur die Basiskonfiguration der Kommunikation zwischen PC und Steuerung über das Ethernet-Netzwerk beschrieben. Daher müssen Sie jetzt nicht die Registerkarten "Markierte IP", "Timeout" und "Dezentrale Geräte" konfigurieren.

Schritt	Aktion
7	Geben Sie eine gültige statische **IP-Adresse** für die Twido-Steuerung in Dezimalpunktschreibweise ein. Diese IP-Adresse muss mit der IP-Adresse des PC kompatibel sein, die Sie im vorigen Abschnitt konfiguriert haben. **Hinweis:** Die IP-Adressen der Twido-Steuerung und des PC müssen dieselbe Netzwerk-ID aufweisen. Die Host-ID der Twido-Steuerung muss jedoch von der Host-ID des PC abweichen und darf nur einmal im gesamten Netzwerk vergeben sein. Wenn beispielsweise die IP-Adresse der Klasse C des PC `192.168.1.198` lautet, dann ist eine gültige Adresse für die Twido-Steuerung `192.168.1.xxx` (wobei `192.168.1` die Netzwerk-ID und `xxx = 0-197, 199-255` die Host-ID ist).
8	Geben Sie eine gültige **Subnetzmaske** in Dezimalpunktschreibweise ein. Die Twido-Steuerung und der PC, auf dem die TwidoSoft-Applikation ausgeführt wird, müssen sich im selben Netzwerksegment befinden. Daher müssen Sie eine Subnetzmaske eingeben, die identisch mit der für den PC angegebenen Subnetzmaske ist. **Hinweis:** Wenn keine Subnetzmasken in Ihrem Netzwerk der Klasse C verwendet wird, empfehlen wir die Eingabe einer Standard-Subnetzmaske für Netzwerke der Klasse C wie etwa `255.255.255.0`.
9	Geben Sie eine gültige **Gateway**-Adresse in Dezimalpunktschreibweise ein. **Hinweis:** Wenn sich kein Gateway-Gerät in Ihrem autonomen Netzwerk befindet, geben Sie die IP-Adresse der Twido-Steuerung ein, die Sie zuvor in Schritt 6 in diesem Feld konfiguriert haben.
10	Klicken Sie auf **OK**, um die Ethernet-Konfigurationseinstellungen Ihrer Twido-Steuerung zu speichern.

Einrichten einer neuen TCP/IP-Verbindung in TwidoSoft

Als nächstes richten Sie eine neue TCP/IP-Verbindung in der TwidoSoft-Applikation ein. Die neue zweckbestimmte TCP/IP-Verbindung ermöglicht die Kommunikation zwischen dem PC, auf dem TwidoSoft ausgeführt wird, und der Twido-Steuerung über das Ethernet-Netzwerk.
Wählen Sie **Datei** → **Präferenzen** aus der TwidoSoft-Menüleiste aus, um das Dialogfeld **Konfiguration der Verbindungen** aufzurufen.



Schritt	Aktion
1	Klicken Sie im Dialogfeld **Konfiguration der Verbindungen** auf die Schaltfläche **Hinzufügen**. **Ergebnis:** Eine neue Verbindungszeile wird hinzugefügt. Die neue Zeile zeigt die vorgeschlagenen Standard-Verbindungseinstellungen an. Sie müssen diese Einstellungen ändern. **Hinweis:** Sie haben zwei Möglichkeiten, um einen neuen Wert in einem Feld zu konfigurieren: ● Wählen Sie das gewünschte Feld aus, und klicken Sie dann auf die Schaltfläche **Ändern**. ● Doppelklicken Sie auf das gewünschte Feld.
2	Geben Sie einen die neue Verbindung beschreibenden Namen in das Feld **Name** ein. Ein gültiger Name kann bis zu 32 alphanumerische Zeichen enthalten.
3	Klicken Sie auf das Feld **Verbindungstyp**, um die Dropdown-Liste anzuzeigen, die folgende Elemente enthält: TCP/IP, Seriell, Modem (falls vorhanden) und USB (falls vorhanden). Wählen Sie **TCP/IP**, da Sie eine neue Ethernet-Verbindung zwischen Ihrem PC und einer Ethernet-fähigen Twido-Steuerung einrichten.
4	Geben Sie im Feld **IP/Telefon** eine gültige **IP-Adresse** ein, bei der es sich um die IP-Information der Twido-Steuerung TWDLCAE40DRF handelt, zu der Sie eine Verbindung aufbauen möchten. **IP-Adresse:** Geben Sie die statische IP-Adresse ein, die Sie in einem vorigen Abschnitt für Ihre Twido-Steuerung angegeben haben.
5	Das Feld **Punit / Adresse** kann ausgefüllt werden, wenn "IP/Telefon" ausgewählt ist. Für eine Verbindung des Typs "TCP/IP" lautet der Standardwert **Direkt**. Für eine Verbindung des Typs "Seriell" lautet der Standardwert **Punit**. Wenn eine dieser beiden Optionen ausgewählt ist, sind die nächsten drei Felder (Baudrate, Parität und Stoppbits) deaktiviert. Wenn Sie die Steuerungsadresse nicht kennen, können Sie durch Auswahl von **@** die Adresse zu einem späteren Zeitpunkt auswählen, wenn das Programm geladen wurde. *(Vor dem Aufbau der ersten Verbindung wird ein Fenster zur Auswahl der Steuerung, an die Sie übertragen, mit einem Bereich von 1 - 247 angezeigt, wobei 1 die Standardadresse ist.)*

Schritt	Aktion
6	Verwenden Sie die Standardeinstellungen in den Feldern **Timeout** und **Break Timeout**, wenn Sie keine speziellen Timeout-Anforderungen haben. (Ausführliche Informationen finden Sie unter *Konfiguration der Ethernet-Verbindungen, S. 187*.)
7	Klicken Sie auf die Schaltfläche **OK**, um die neuen Verbindungseinstellungen zu speichern, und schließen Sie das Dialogfeld "Konfiguration der Verbindungen". **Ergebnis:**Die Namen aller neu hinzugefügten Verbindungen werden zum Dropdown-Listenfeld der Verbindungen im Dialogfeld **Datei** → **Präferenzen** oder zum Dialogfeld **SPS** → **Verbindung auswählen** hinzugefügt.

Anschließen Ihrer Steuerung an das Netzwerk

Überblick

Nachfolgend ist beschrieben, wie Ihre kompakte Steuerung TDWLCAE40DRF an Ihr Ethernet-Netzwerk angeschlossen wird.

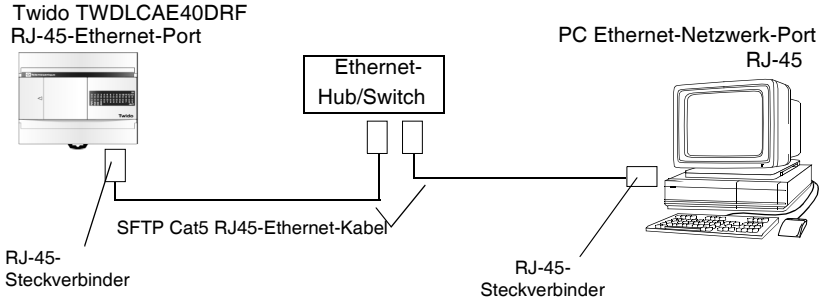
Ermittlung geeigneter IP-Adressen

Wenden Sie sich an Ihren Netzwerkadministrator, um zu ermitteln, ob Sie neue Geräte-IP-, Gateway- und Subnetzmaskenadressen konfigurieren müssen. Wenn der Administrator neue IP-Adressparameter zuordnet, müssen Sie diese Informationen manuell in die TwidoSoft-Applikation eingeben. Befolgen Sie die im Abschnitt *TCP/IP-Setup, S. 174* aufgeführten Anweisungen.

Ethernet-Netzwerkverbindung

Hinweis: Obwohl eine direkte Kabelverbindung (mittels eines Ethernet-Überbrückungskabels) zwischen der Twido-Steuerung TWDLCAE40DRF und dem PC, auf dem die Programmiersoftware TwidoSoft ausgeführt wird, unterstützt wird, wird von dieser Möglichkeit abgeraten. Sie sollten statt dessen eine Verbindung über einen Ethernet-Netzwerk-Hub/Switch vorziehen.

Die folgende Abbildung zeigt eine Twido-Netzwerkverbindung über einen Ethernet-Hub/Switch.



Die Twido-Steuerung TWDLCAE40DRF ist mit einem RJ-45-Stecker für den Anschluss an das 100BASE-TX Ethernet-Netzwerk mit automatischer Einstellung ausgestattet. Sie unterstützt eine Netzwerk-Übertragungsrate sowohl von 100 Mbps als auch von 10 Mbps.

Hinweis: Bei Anschluss der Twido-Steuerung an ein 100BASE-TX-Netzwerk sollten Sie mindestens ein Ethernet-Kabel der Kategorie 5 verwenden.

IP-Adressierung

Überblick

Dieser Abschnitt enthält Informationen über die IP-Adressnotierung sowie über das Subnetz- und Gateway-Konzept.

IP-Adresse

Eine IP-Adresse ist eine aus 32 Bits bestehende Einheit, die in Dezimalpunktschreibweise angegeben wird. Sie besteht aus vier Zahlengruppen im Bereich von 0 bis 255, die durch einen Punkt voneinander getrennt sind. 192.168.2.168 ist beispielsweise eine IP-Adresse in Dezimalpunktschreibweise (beachten Sie, dass es sich hierbei um eine lediglich als Beispiel angegebene, reservierte IP-Adresse handelt).
In normalen Netzwerken fallen IP-Adressen in drei Kategorien, die als Netzwerke der Klasse A, B und C bezeichnet werden. Die Klassen sind abhängig von der ersten Zahl der IP-Adresse (siehe folgende Tabelle).

Erste Dezimalgruppe	IP-Klasse
0-127	Klasse A
128-191	Klasse B
192-223	Klasse C

IP-Subnetzmaske

Eine IP-Adresse besteht aus zwei Teilen, der Netzwerk-ID und der Host-ID. Die Subnetzmaske wird verwendet, um den Netzwerkteil der IP-Adresse zu unterteilen, um künstlich Subnetzwerke mit einer größeren Anzahl an Host-IDs zu erstellen. Folglich wird die Subnetzerstellung verwendet, um mehrere physikalische Netzwerke zu logischen Netzwerken zu verbinden. Alle Geräte im selben Subnetz nutzen dieselbe Netzwerk-ID.
Alle Geräte im selben Subnetz nutzen dieselbe Netzwerk-ID.

Hinweis: Wenn Sie in einem großen Unternehmen arbeiten, ist es sehr wahrscheinlich, dass Ihr Unternehmensnetzwerk aus mehreren Subnetzen besteht. Wenden Sie sich an Ihren Netzwerkadministrator, um entsprechende Informationen zur Subnetzerstellung zu erhalten, wenn Sie Ihre neue Twido-Steuerung in einem bestehenden Netzwerk installieren.

Gateway-Adresse

Das Gateway (auch "Router" genannt) ist das Netzwerkgerät, das für Ihr Netzwerksegment einen Zugang zu anderen Netzwerksegmenten im globalen Netzwerk Ihres Unternehmens sowie einen Zugang zum Internet oder zu einem dezentralen Intranet bietet.
Die Gateway-Adresse nutzt dieselbe Dezimalpunktschreibweise wie die oben beschriebene IP-Adresse.

Hinweis: Wenden Sie sich an Ihren Netzwerkadministrator, um entsprechende Gateway-Informationen zu erhalten, wenn Sie Ihre neue Twido-Steuerung in einem bestehenden Netzwerk installieren.

Zuweisen von IP-Adressen

Überblick

Dieser Abschnitt beschreibt, wie Sie ermitteln können, welchen IP-Adresstyp Sie der Twido-Steuerung TWDLCAE40DRF zuweisen können, die Sie ihn Ihrem Netzwerk installieren möchten.

Installation in einem autonomen Netzwerk

Ihre Twido-Steuerung TWDLCAE40DRF ist für die Installation in einem autonomen Ethernet-Netzwerk konzipiert.

Hinweis: Ein Netzwerk wird als autonom bezeichnet, wenn es nicht mit dem Internet oder dem Intranet eines Unternehmens verbunden ist.

Erhalt einer Adresse über BootP

Von BootP gelieferte Adresse: Wenn Sie auf der Registerkarte **IP-Adresskonfiguration** die Option **Von einem Server** auswählen, versucht die Twido-Steuerung zunächst, eine IP-Adresse vom BootP-Server zu erhalten.
Der BootP-Prozess erwartet eine Antwort vom BootP-Server. Wenn nach der Übertragung des BootP-Requests keine gültige IP-Adresse empfangen wird, übernimmt Twido die Standard-IP-Konfiguration, die von einer MAC-Adresse abgeleitet wird (siehe *MAC-Adresse und Standard-IP-Adresse der Steuerung, S. 170* unten).

MAC-Adresse und Standard-IP-Adresse der Steuerung

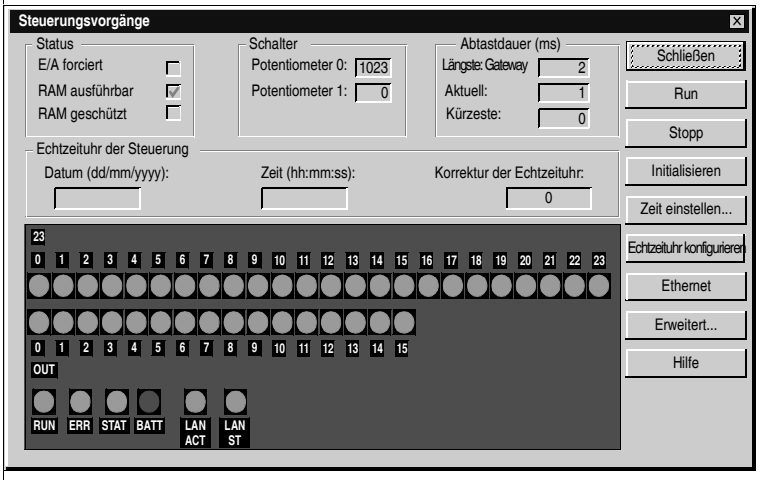
MAC-Adresse: Jede Twido-Steuerung TWDLCAE40DRF verfügt über ihre eigene MAC-Adresse, bei der es sich um eine weltweit einmalig vergebene 48-Bit-Adresse handelt, die jedem Ethernet-Gerät zugewiesen wird.
Standard-IP-Adresse: Die Standard-IP-Adresse der Ethernet-Schnittstelle der Twido-Steuerung wird von der eindeutigen MAC-Adresse abgeleitet.
Die in Dezimalformatschreibweise angegebene Standard-IP-Adresse ist wie folgt definiert:
`085.016.xxx.yyy`, wobei:

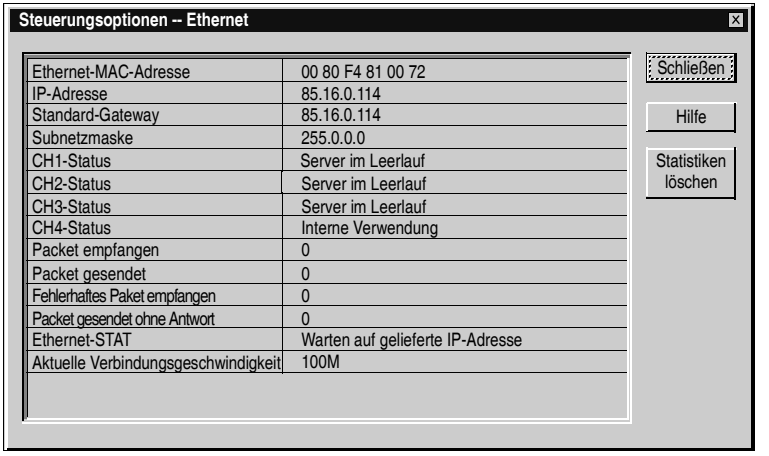
- `085.016.` ein festgelegter Header für alle von MAC-Adressen abgeleiteten IP-Adressen und
- `xxx` und `yyy` die beiden letzten Zahlen der MAC-Adresse des Geräts sind.

So lautet beispielsweise die von der MAC-Adresse `00.80.F4.81.01.11` abgeleitete IP-Adresse `085.016.001.17`.

Überprüfung der MAC-Adresse und der aktuellen IP-Adresse der Steuerung

Gehen Sie folgendermaßen vor, um die MAC-Adresse und die aktuelle IP-Adresse Ihrer Twido-Steuerung sowie die IP-Konfigurationseinstellungen (Subnetzmaske und Gateway-Adressen) und den Ethernet-Verbindungsstatus zu ermitteln:

Schritt	Aktion
1	Wählen Sie im TwidoSoft-Applikationsprogramm aus der Menüleiste die Option **SPS** aus.
2	Wählen Sie aus der Liste der Menüoptionen die Option **SPS überprüfen** aus. **Ergebnis:** Das Dialogfeld **Steuerungsvorgänge** wird angezeigt. Es zeigt die Twido LEDs auf einer softwaremäßigen Frontseite wie in folgender Abbildung dargestellt: 

Schritt	Aktion
3	Klicken Sie auf die Schaltfläche **Ethernet** im rechten Teil des Bildschirms, um die Verbindungsparameter anzuzeigen. **Ergebnis:** Die Tabelle **Steuerungsvorgänge - Ethernet** wird angezeigt. Sie enthält die MAC-Adresse, die aktuelle IP-Adresse sowie Informationen zum Subnetz und zum Gateway sowie Ethernet-Verbindungsinformationen wie in folgender Abbildung dargestellt. 
4	Beachten Sie, dass die eindeutige **MAC**-Adresse der Twido-Steuerung in der ersten Zeile der Ethernet-Tabelle angezeigt wird.
5	Die in dieser Tabelle angezeigte IP-Information ist abhängig von den Benutzereinstellungen auf der Registerkarte **IP-Konfiguration** im Dialogfeld **Ethernet-Konfiguration** (siehe *Registerkarte "IP-Adresskonfiguration", S. 176*): ● Wenn Sie die Option **Von einem Server** auf der Registerkarte "IP-Konfiguration" ausgewählt haben, zeigt die Tabelle oben die (von der MAC-Adresse abgeleitete) Standard-IP-Adresse der Twido-Steuerung sowie die Standard-Subnetzmaske und das Standard-Gateway an. Beachten Sie, dass die Standard-IP-Adresse im Fallback-Modus nur verwendet wird, wenn keine gültige, von BootP gelieferte IP-Adresse vom Server erhalten werden kann. Wenn ein Kanal als UDP für BootP verwendet wird, zeigt der Kanalstatus **Interne Verwendung.** ● Wenn Sie die Option **Konfiguriert** auf der Registerkarte "IP-Konfiguration" ausgewählt haben, zeigt die Tabelle oben die aktuelle IP-Adresse und die Subnetz- und Gateway-Einstellungen an, die Sie zuvor auf der Registerkarte "IP-Konfiguration" eingegeben haben. Hinweis: Die übrigen Felder enthalten Informationen über den aktuellen Status der Ethernet-Verbindung. Weitere Informationen finden Sie unter (Siehe TwidoSoft Betriebshandbuch - Online-Hilfe).

Private IP-Adressen

Wenn Sie über ein autonomes Netzwerk verfügen (nicht mit dem Internet verbunden), können Sie Ihrem Netzwerkknoten (der Twido-Steuerung) jede beliebige IP-Adresse zuweisen (vorausgesetzt, dass die IP-Adresse den IANA-Notationsregeln entspricht und sie nicht in Konflikt mit den IP-Adressen anderer, bereits an das Netzwerk angeschlossener Geräte steht).
Private IP-Adressen entsprechen der Anforderung nach einer beliebigen IP-Adressierung in einem autonomen Netzwerk. Bitte beachten Sie, dass Adressen innerhalb dieses privaten Adressbereichs nur innerhalb des Unternehmens eindeutig sind.
Die folgende Tabelle beschreibt den privaten IP-Adressbereich.

Netzwerk	Gültiger Bereich für private IP-Adressen
Klasse A	10.0.0.0 -> 10.255.255.255
Klasse B	172.16.0.0 -> 172.31.255.255
Klasse C	192.168.0.0 -> 192.168.255.255

Zuweisen einer IP-Adresse zu Ihrer Steuerung

Heutige Netzwerke sind in den seltensten Fällen vollständig vom Internet oder vom Rest des Ethernet-Netzwerks des Unternehmens getrennt. Weisen Sie daher, wenn Sie Ihr Twido-SPS-Grundgerät installieren und mit einem bestehenden Netzwerk verbinden, keine beliebigen IP-Adressen zu, ohne sich zuvor mit Ihrem Netzwerkadministrator zu beraten. Befolgen Sie die nachfolgend aufgeführten Regeln zur Zuweisung einer IP-Adresse zu Ihrer Steuerung.

Hinweis: Es ist eine gute Praxis, in autonomen Netzwerken IP-Adressen der Klasse C zu verwenden.

TCP/IP-Setup

Überblick

Nachfolgend ist ausführlich die Einrichtung der Ethernet TCPI/IP-Konfiguration für Ihre kompakte Twido-Steuerung TWDLCAE40DRF beschrieben.

Hinweis: Das TCP/IP-Setup kann nur ausgeführt werden, wenn das TwidoSoft-Applikaitonsprogramm im Offline-Modus befindet.

⚠ ACHTUNG

UNBEABSICHTIGTER BETRIEBSZUSTAND DES GERÄTS

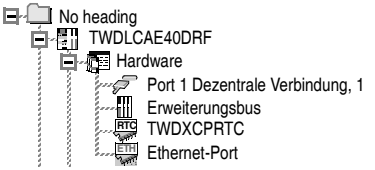
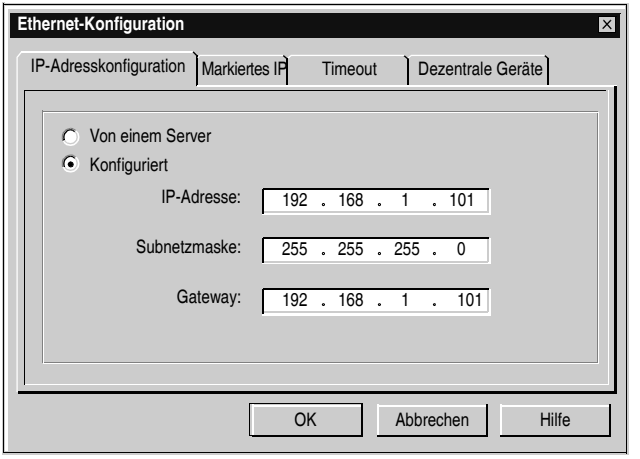
Das Vorhandensein von zwei Geräten mit derselben IP-Adresse kann zu einem unvorhersehbaren Betrieb Ihres Netzwerks führen.

- Stellen Sie sicher, dass dieses Gerät eine eindeutige IP-Adresse erhält.
- Lassen Sie sich Ihre IP-Adresse immer von Ihrem Systemadministrator zuweisen, um die Möglichkeit doppelter Adressen zu vermeiden.

Die Nichtbeachtung dieser Anweisung kann Körperverletzung oder Materialschäden zur Folge haben!

Aufrufen des Dialogfelds "Ethernet-Konfiguration"

Nachfolgend ist beschrieben, wie das Dialogfeld **Ethernet-Konfiguration** aufgerufen wird.

Schritt	Aktion
1	Öffnen Sie den **Applikationsnavigator** wie in folgender Abbildung gezeigt. **Ergebnis:**  **Hinweis:** Vergewissern Sie sich, dass ein Ethernet-fähiges Gerät wie etwa die Steuerung TWDLCAE40DRF als aktuelle Hardware ausgewählt ist, da ansonsten nicht die Hardwareoption "Ethernet-Port" angezeigt wird.
2	Doppelklicken Sie auf das Symbol **Ethernet-Port**, um das unten dargestellte Dialogfeld **Ethernet-Konfiguration** anzuzeigen. **Ergebnis:**  **Hinweis:** Es gibt zwei verschiedene Möglichkeiten, das Ethernet-Konfigurationsfenster aufzurufen: **1.** Klicken Sie mit der rechten Maustaste auf das Symbol **Ethernet-Port**, und wählen Sie die Option **Bearbeiten** aus dem Kontextmenü aus. **2.** Wählen Sie in der Menüleiste von TwidoSoft die Option **Hardware > Ethernet** aus.

TCP/IP-Setup

Die folgenden Abschnitte beschreiben ausführlich die Konfiguration der TCP/IP-Parameter der Twido-Steuerung TWDLCAE40DRF mittels der Registerkarten **IP-Adresse, Markiertes IP, Timeout** und **Dezentrale Geräte**.

Registerkarte "IP-Adresskonfiguration"

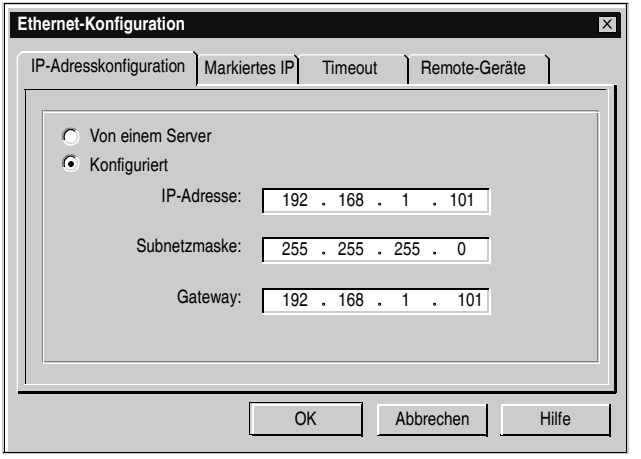
Überblick

Nachfolgend ist die Konfiguration der Registerkarte "IP-Adresskonfiguration" des Dialogfelds "Ethernet-Konfiguration" beschrieben.

Hinweis: Die IP-Adresse der Twido-Steuerung kann nur konfiguriert werden, wenn sich das TwidoSoft-Applikationsprogramm im Offline-Modus befindet.

Registerkarte "IP-Adresskonfiguration"

Die folgende Abbildung zeigt beispielhaft die Registerkarte "IP-Adresskonfiguration" mit den manuell durch den Benutzer konfigurierten IP-, Teilnetz- und Gateway-Adressen:



Konfigurieren der Registerkarte "IP-Adresskonfiguration"

Nachfolgend ist die Konfiguration der verschiedenen Felder der Registerkarte "IP-Adresskonfiguration" beschrieben.

Feld	Konfigurieren
Von einem Server	Aktivieren Sie diese Optionsschaltfläche, wenn Sie die IP-Adresse der Twido-Steuerung nicht manuell konfigurieren möchten (die Textfelder "IP-Adresse", "Subnetzmaske" und "Gateway" sind abgeblendet). Die Twido-Steuerung (BootP-Client) wird dann die automatisch vom Server zugewiesene IP-Adresse verwenden. Die Twido-Steuerung wird die Standard-IP-Adresse verwenden (Fallback-Zustand), wenn sie nach drei im Abstand von 200 ms durchgeführten Wiederholungen keine gültige gelieferte IP-Adresse erhalten kann. (Beachten Sie, dass die Twido-Steuerung periodisch in 15-Sek.-Intervallen Requests an den Server sendet, bis sie eine gültige IP-Adresse erhält.) Die Standard-IP-Adresse der Ethernet-Schnittstelle wird von ihrer MAC-Adresse abgeleitet. (Beachten Sie weiterhin, dass die Standard-IP-Adresse nicht automatisch geändert wird, wenn ein Kanal (mit Ausnahme des Kanals für interne Verwendung) der SPS aktiv ist.) **Hinweis:** Weitere Informationen über BootP und die MAC-Adresse finden Sie unter *Zuweisen von IP-Adressen, S. 170*.
Konfiguriert	Aktivieren Sie dieses runde Optionsfeld, um die IP-, Subnetz- und Gateway-Adressen manuell zu konfigurieren. **Hinweis:** Wenden Sie sich an Ihren Netzwerk- oder Systemadministrator, um gültige IP-Parameter für Ihr Netzwerk zu erhalten.
IP-Adresse	Geben Sie die statische IP-Adresse Ihrer Twido-Steuerung in Dezimalpunktschreibweise ein. **Vorsicht:** Um eine gute Kommunikation des Geräts zu gewährleisten, müssen die IP-Adressen des PC, auf dem die TwidoSoft-Applikation ausgeführt wird, und der Twido-Steuerung dieselbe Netzwerk-ID aufweisen. **Hinweis:** Um eine gute Kommunikation über das Netzwerk zu ermöglichen, muss jedes angeschlossene Gerät über eine eindeutige IP-Adresse verfügen. Wenn die Twido-Steuerung mit dem Netzwerk verbunden ist, führt sie eine Überprüfung auf doppelte IP-Adressen durch. Wenn eine doppelte IP-Adresse im Netzwerk erkannt wird, blinkt die LAN ST-LED der Twido-Steuerung regelmäßig 4 Mal auf. In diesem Fall müssen Sie eine neue eindeutige IP-Adresse in dieses Feld eingeben.
Subnetzmaske	Geben Sie die gültige, Ihrer Steuerung vom Netzwerkadministrator zugewiesene Subnetzmaske ein. Bitte beachten Sie, dass Sie dieses Feld nicht leer lassen können. Sie müssen einen Wert eingeben. Standardmäßig berechnet die TwidoSoft-Applikation eine Standard-Subnetzmaske auf der Grundlage der von Ihnen oben im Feld "IP-Adresse" angegebenen IP-Adresse und zeigt diese Subnetzmaske an. Die der Kategorie der Twido-Netzwerk-IP-Adresse entsprechenden standardmäßigen Maskenwerte entsprechen folgender Regel: Netzwerk der Klasse A -> Standard-Subnetzmaske: 255.0.0.0 Netzwerk der Klasse B -> Standard-Subnetzmaske: 255.255.0.0 Netzwerk der Klasse C -> Standard-Subnetzmaske: 255.255.255.0 **Vorsicht:** Um eine gute Kommunikation des Geräts zu gewährleisten, müssen die auf dem PC, auf dem die TwidoSoft-Applikation ausgeführt wird, konfigurierte Subnetzmaske und die Subnetzmaske der Twido-Steuerung übereinstimmen. **Hinweis:** Verwenden Sie die Standard-Subnetzmaske, es sei denn, Ihre Twido-Steuerung erfordert eine spezielle Subnetzmasken-Zuweisung.

Feld	Konfigurieren
Gateway	Geben Sie die IP-Adresse des Gateways an. Im LAN muss sich das Gateway in demselben Segment wie Ihre Twido-Steuerung befinden. Diese Information erhalten Sie üblicherweise von Ihrem Netzwerkadministrator. Bitte beachten Sie, dass kein Standardwert von der Applikation geliefert wird und dass Sie eine gültige Gateway-Adresse in dieses Feld eingeben müssen. **Hinweis:** Wenn sich kein Gateway-Gerät in Ihrem Netzwerk befindet, geben Sie einfach die IP-Adresse Ihrer Twido-Steuerung in das Feld "Gateway" ein.

Registerkarte "Markiertes IP"

Überblick

Nachfolgend ist die Konfiguration der Registerkarte "Markiertes IP" des Dialogfelds "Ethernet-Konfiguration" beschrieben.

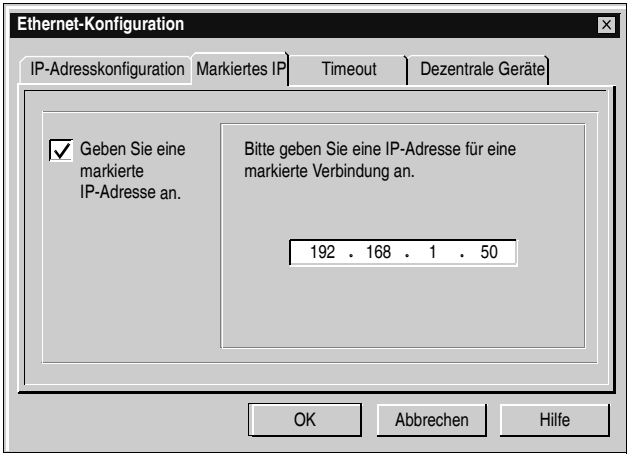
Hinweis: Die markierte IP kann nur konfiguriert werden, wenn sich das TwidoSoft-Applikationsprogramm im Offline-Modus befindet.

Definition der Funktion "Markiertes IP"

Diese Funktion ermöglicht Ihnen, einen der vier von Ihrer Twido-Steuerung unterstützten Ethernet TCP-Verbindungskanäle für einen bestimmten Client-Host zu reservieren, der als "Markiertes IP" bezeichnet wird.
"Markiertes IP" gewährleistet, dass ein TCP-Kanal reserviert und immer für die Kommunikation mit dem angegebenen dezentralen Gerät verfügbar ist, selbst wenn der Ruhe-Timeout deaktiviert ist (Ruhe-Timeout auf "0" gesetzt).

Registerkarte "Markiertes IP"

Die folgende Abbildung zeigt beispielhaft die Registerkarte "Markierte IP" mit einer vom Benutzer eingegebenen IP-Adresse.



Konfigurieren der Registerkarte "Markiertes IP"

Gehen Sie folgendermaßen vor, um die Registerkarte "Markiertes IP" zu konfigurieren:

Schritt	Aktion
1	Aktivieren Sie das Kontrollkästchen **Geben Sie eine Markiertes IP-Adresse an**, um die Funktion "Markiertes IP" zu aktivieren. Beachten Sie, dass die Funktion "Markiertes IP" standardmäßig deaktiviert ist. **Ergebnis:** Das IP-Adressfeld im rechten Bereich des Rahmens wird aktiv (siehe Abbildung oben).
2	Geben Sie die IP-Adresse des Client-Host, dessen IP Sie markieren möchten, in das angezeigte IP-Adressfeld ein. **Hinweis:** Es wird kein Standardwert in diesem Feld angezeigt. Sie müssen die IP-Adresse des markierten Geräts angeben. Deaktivieren Sie andernfalls das Kontrollkästchen "Geben Sie eine markierte IP-Adresse an", um diese Funktion zu deaktivieren.

Registerkarte "Timeout"

Überblick

Nachfolgend ist die Konfiguration der Registerkarte "Timeout" des Dialogfelds "Ethernet-Konfiguration" beschrieben.

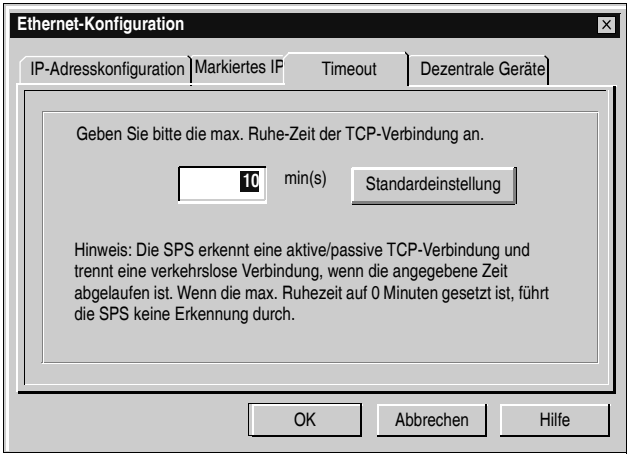
Hinweis: Der Timeout der Twido-Steuerung kann nur konfiguriert werden, wenn sich das TwidoSoft-Applikationsprogramm im Offline-Modus befindet.

Definition des Timeouts

Timeout wendet einen Ruhe-Timeout auf alle aktuellen Ethernet TCP-Verbindungen der Twido-Steuerung an. Der Ruhe-Timeout ist der Zeitraum, den jeder der vier Ethernet TCP-Verbindungskanäle ruhig bleiben kann, bevor die dezentrale Client-Host-Verbindung zu diesem Kanal getrennt wird.
Hinweis: Der Ruhe-Timer wird immer dann zurückgesetzt, wenn Daten über den überwachten Verbindungskanal gesendet werden.

Registerkarte "Timeout"

Die folgende Abbildung zeigt beispielhaft die Registerkarte "Timeout" mit dem 10-Minuten-Standardwert des Ruhe-Timers.



Konfigurieren der Registerkarte "Timeout"

Geben Sie zum Konfigurieren des Ruhe-Timers einfach direkt die Dauer in Minuten wie in der Abbildung oben dargestellt in das Textfeld **min(s)** ein.

Hinweis:

1. Die Standarddauer beträgt 10 Minuten. Klicken Sie, nachdem Sie einen Wert eingegeben haben, auf die Schaltfläche **Standard**, um die konfigurierte Dauer auf 10 Minuten **zurückzusetzen**.
2. Setzen Sie die Dauer auf **0**, um die Funktion "Timeout" zu **deaktivieren**. Die Twido-Steuerung führt in diesem Fall keine Ruhe-Überprüfungen mehr durch. Als Folge bleiben die TCP-Verbindungen unendlich bestehen.
3. Die maximal einstellbare Ruhe-Dauer beträgt 255 Minuten.

Registerkarte "Dezentrale Geräte"

Überblick

Nachfolgend ist die Konfiguration der Registerkarte "Dezentrale Geräte" des Dialogfelds "Ethernet-Konfiguration" beschrieben. Sie können diese Funktion verwenden, wenn Sie beabsichtigen, die Anweisung EXCH3 zu verwenden, damit die Twido-Steuerung als ein Modbus TCP/IP-Client fungiert.

Hinweis: Die Registerkarte "Dezentrale Geräte" der Twido-Steuerung kann nur konfiguriert werden, wenn sich das TwidoSoft-Applikationsprogramm im Offline-Modus befindet.

Was Sie wissen sollten

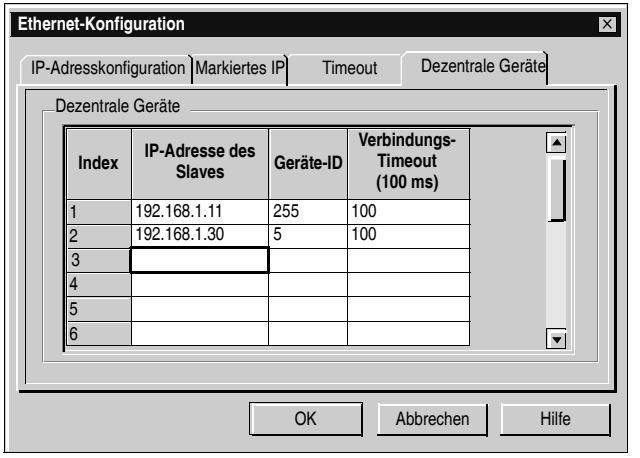
Sie müssen die Funktion "Dezentrale Geräte" an keiner anderen Steuerung konfigurieren als an der Steuerung, die die Modbus TCP/IP-Client- (früher Modbus-Master) Anweisung (EXCH3) verwenden soll.

Tabelle "Dezentrale Geräte"

Die Tabelle "Dezentrale Geräte" speichert Informationen über dezentrale Steuerungen (die als Modbus TCP/IP-Server fungieren) im Ethernet-Netzwerk, an die vom Modbus TCP/IP-Client mittels der Anweisung EXCH3 Requests gestellt werden können. Daher müssen Sie die Tabelle "Dezentrale Geräte" ordnungsgemäß konfigurieren, so dass die Modbus TCP/IP-Client-Steuerung Modbus TCP/IP-Server-Steuerungen über das Netzwerk abfragen können.

Registerkarte "Dezentrale Geräte"

Die folgende Abbildung zeigt beispielhaft die an einer als Modbus TCP/IP-Client fungierenden Twido-Steuerung konfigurierte Registerkarte "Dezentrale Geräte".



Konfigurieren der Registerkarte "Dezentrale Geräte"

Nachfolgend ist die Konfiguration der verschiedenen Felder der Registerkarte "Dezentrale Geräte" beschrieben.

Feld	Konfigurieren
Index	Dies ist ein schreibgeschütztes Feld, das den mit der Ethernet-Netzwerk-IP-Adresse des dezentralen Gerätes (im Slave-IP-Adressfeld angegebener Modbus TPC/IP-Server) verbundenen MBAP-Index enthält. Der MBAP-Index wird durch die Anweisung EXCH3 als eines der Argumente der Funktion aufgerufen, um zu ermitteln, welche in der Tabelle angegebene dezentrale Steuerung vom Modbus TCP/IP-Client abgefragt wird. **Hinweis:** Sie können bis zu 16 verschiedene, in dieser Tabelle von 1 bis 16 indexierte dezentrale Geräte angeben.
Slave-IP-Adresse	Geben Sie in dieses Feld die IP-Adresse der dezentralen Steuerung (Modbus TCP/IP-Server) ein. **Hinweis:** Sie müssen die Slave-IP-Adressen beginnend bei Index 1 und ununterbrochen in aufsteigender Indexnummer konfigurieren. So ist beispielsweise eine Konfiguration der Slave-IPs von Index 1 und dann 3 nicht zulässig. Sie müssen erst den Eintrag mit dem Index 2 konfigurieren, bevor Sie den Index 3 konfigurieren können.
Geräte-ID	Geben Sie in dieses Feld die Modbus-Geräte-ID (oder Protokolladresse) ein. Der zulässige Bereich für die Geräte-ID lautet 0 bis 255. Die Standardeinstellung ist 255. Eine Geräte-ID (eine andere als 255) ermöglicht die Kommunikation mit einem dezentralen Gerät über eine Modbus-Bridge oder ein Modbus-Gateway. Wenn das Zielgerät eine andere Twido-Steuerung oder ein älteres Modbus-Gerät ist, die bzw. das auf einem anderen Bus installiert ist - serielle Verbindungsadresse über ein Gateway -, dann müssen Sie die Geräte-ID des dezentralen Geräts entsprechend konfigurieren. Im Feld sollten Sie die Slave-IP als die Gateway- oder Bridge-IP-Adresse und die Geräte-ID als die serielle Modbus-Verbindungsadresse Ihres Zielgeräts setzen.
Verbindungs-Timeout (100 ms)	Geben Sie die Dauer in Einheiten zu 100 ms an, wie lange die Twido-Steuerung versuchen soll, eine TCP-Verbindung mit dem dezentralen Gerät aufzubauen. Wenn die Verbindung nach Ablauf dieser Dauer immer noch nicht hergestellt ist, stellt die Twido-Steuerung den Versuch eines Verbindungsaufbaus bis zum nächsten Verbindungs-Request durch eine EXCH3-Anweisung ein. Der zulässige Bereich für die Timeout-Einstellung beträgt 0 bis 65535 (was 0 bis 6553,5 s entspricht). Die Standardeinstellung ist 100.

Anzeigen der Ethernet-Konfiguration

Überblick

Sie können den TwidoSoft-**Konfigurationseditor** verwenden, um die aktuelle Ethernet-Konfiguration der Twido-Steuerung anzuzeigen.

Anzeigen der Ethernet-Konfiguration

Gehen Sie folgendermaßen vor, um die aktuellen Ethernet-Konfigurationseinstellungen mittels des Konfigurationseditors anzuzeigen:

Schritt	Aktion
1	Wählen Sie **Programm > Konfigurationseditor** aus der TwidoSoft-Menüleiste aus.
2	Klicken Sie auf den Shortcut mit der Bezeichnung **ETH** in der Konfigurationseditor-Taskleiste oder doppelklicken Sie auf den Shortcut **Ethernet-Port** im Applikationsnavigator.
3	Die Ethernet TCP/IP-Konfigurationsparameter werden wie in der Abbildung unten dargestellt in einer Tabelle angezeigt.

Ethernet-Konfiguration

IP-Adresskonfiguration		
IP-Adresse	192 . 168 . 1 . 101	
Subnetzmaske	255 . 255 . 255 . 0	
Gateway-Adresse	192 . 168 . 1 . 101	
Markiertes IP	192 . 168 . 1 . 50	
Dezentraler Server Slave-IP-Adresse	Geräte-ID	Verbindungs-Timeout
192 . 168 . 1 . 11	255	100
192 . 168 . 1 . 30	5	100
192 . 168 . 1 . 50	255	1500
192 . 168 . 1 . 16	255	1500
192 . 168 . 1 . 20	255	100

Schritt	Aktion
4	Zu diesem Zeitpunkt können Sie, wenn Sie Änderungen an den Ethernet TCP/IP-Konfigurationseinstellungen Ihrer Twido-Steuerung vorgenommen haben, immer noch entscheiden, ob Sie die Änderungen beibehalten oder verwerfen und die vorherige Konfiguration wiederherstellen möchten. Dieser Vorgang ist nachfolgend beschrieben. ● Wählen Sie **Extras > Änderungen übernehmen** aus der TwidoSoft-Menüleiste aus, um die an der TCP/IP-Ethernet-Konfiguration vorgenommenen Änderungen beizubehalten. ● Wählen Sie **Extras > Änderungen rückgängig machen** aus, um die vorgenommenen Änderungen zu verwerfen und die vorherigen TCP/IP-Ethernet-Konfigurationseinstellungen wiederherzustellen. ● Wählen Sie **Extras > Bearbeiten...** aus, um zum Dialogfeld "Ethernet-Konfiguration" zurückzukehren und die TCP/IP-Konfigurationseinstellungen zu bearbeiten. ● Wählen Sie aus der Menüleiste die Option **SPS > Übertragung PC => SPS...**, um die komplette SPS-Konfigurationsdatei in die Twido-Steuerung zu laden.

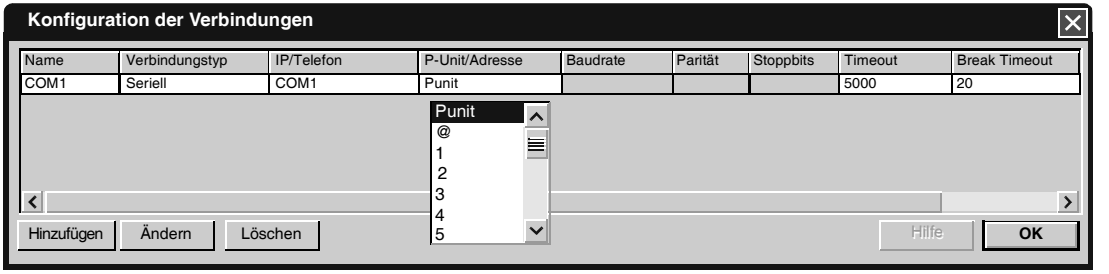
Konfiguration der Ethernet-Verbindungen

Überblick

Nachfolgend ist die Konfiguration und das Hinzufügen/Löschen/Auswählen einer Ethernet TPC/IP-Verbindung zwischen PC und Steuerung beschrieben.

Einrichten einer neuen TCP/IP-Verbindung

Befolgen Sie die nachfolgend aufgeführten Anweisungen, um eine Ethernet TCP/IP-Verbindung zwischen Ihrem PC, auf dem die TwidoSoft-Applikation ausgeführt wird, und einer in Ihrem Netzwerk installierten TWDLCAE40DRF-Steuerung zu konfigurieren.
Wählen Sie **Datei** → **Präferenzen** aus der TwidoSoft-Menüleiste aus, um das Dialogfeld **Konfiguration der Verbindungen** aufzurufen.



Schritt	Aktion
1	Klicken Sie im Dialogfeld **Konfiguration der Verbindungen** auf die Schaltfläche **Hinzufügen**. **Ergebnis:** Eine neue Verbindungszeile wird hinzugefügt. Die neue Zeile zeigt die vorgeschlagenen Standard-Verbindungseinstellungen an. Sie müssen diese Einstellungen ändern. **Hinweis:** Sie haben zwei Möglichkeiten, um einen neuen Wert in einem Feld zu konfigurieren: • Wählen Sie das gewünschte Feld aus, und klicken Sie dann auf die Schaltfläche **Ändern**. • Doppelklicken Sie auf das gewünschte Feld.
2	Geben Sie einen die neue Verbindung beschreibenden Namen in das Feld **Name** ein. Ein gültiger Name kann bis zu 32 alphanumerische Zeichen enthalten.
3	Klicken Sie auf das Feld **Verbindungstyp**, um die Dropdown-Liste anzuzeigen, die folgende Elemente enthält: TCP/IP, Seriell, Modem (falls vorhanden) und USB (falls vorhanden). Wählen Sie **TCP/IP**, da Sie eine neue Ethernet-Verbindung zwischen Ihrem PC und einer Ethernet-fähigen Twido-Steuerung einrichten.
4	Geben Sie im Feld **IP/Telefon** eine gültige **IP-Adresse** ein, bei der es sich um die IP-Information der Twido-Steuerung TWDLCAE40DRF handelt, zu der Sie eine Verbindung aufbauen möchten. **IP-Adresse:** Geben Sie die statische IP-Adresse ein, die Sie in einem vorigen Abschnitt für Ihre Twido-Steuerung angegeben haben.

Schritt	Aktion
5	Das Feld **Punit / Adresse** kann ausgefüllt werden, wenn "IP/Telefon" ausgewählt ist. Für eine Verbindung des Typs "TCP/IP" lautet der Standardwert **Direkt**. Für eine Verbindung des Typs "Seriell" lautet der Standardwert **Punit**. Wenn eine dieser beiden Optionen ausgewählt ist, sind die nächsten drei Felder (Baudrate, Parität und Stoppbits) deaktiviert. Wenn Sie die Steuerungsadresse nicht kennen, können Sie durch Auswahl von **@** die Adresse zu einem späteren Zeitpunkt auswählen, wenn das Programm geladen wurde. (Vor dem Aufbau der ersten Verbindung wird ein Fenster zur Auswahl der Steuerung, an die Sie übertragen, mit einem Bereich von 1 - 247 angezeigt, wobei 1 die Standardadresse ist.)
6	Geben Sie in das Feld **Timeout** einen Timeout-Wert in Millisekunden (ms) für den Aufbau einer Verbindung zur Twido-Steuerung ein. Wenn der PC bis zum Ablauf des Timeout keine Verbindung zur Steuerung aufbauen konnte, unternimmt die TwidoSoft-Applikation keinen weiteren Versuch eines Verbindungsaufbaus. Um einen neuen Versuch eines Verbindungsaufbaus zu starten, wählen Sie **SPS** → **Verbindung auswählen** aus der TwidoSoft-Menüleiste aus. **Hinweis:** Der standardmäßige Timeout-Wert ist **500** ms. Der maximale Timeout-Wert beträgt 65535 x 100 ms (6553,5 s).
7	Der **Break Timeout**-Wert ist die maximal zulässige Dauer zwischen einer Modbus TCP/IP-Request und dem Empfang des Antwort-Frames. Wenn der Break Timeout überschritten wird, ohne dass der angeforderte Antwort-Frame empfangen wurde, unterbricht die TwidoSoft-Applikation die Verbindung zwischen dem PC und der Steuerung. **Hinweis:** Der standardmäßige Break Timeout-Wert ist **20** ms. Sie müssen einen Wert ungleich Null konfigurieren.
8	Klicken Sie auf die Schaltfläche **OK**, um die neuen Verbindungseinstellungen zu speichern, und schließen Sie das Dialogfeld "Konfiguration der Verbindungen". **Ergebnis:** Die Namen aller neu hinzugefügten Verbindungen werden zum Dropdown-Listenfeld der Verbindungen im Dialogfeld **Datei** → **Präferenzen** oder zum Dialogfeld **SPS** → **Verbindung auswählen** hinzugefügt.

Ändern und Löschen einer TCP/IP-Verbindung

Nachfolgend ist beschrieben, wie vorhandene Ethernet TCP/IP-Verbindungen entfernt oder ihre Parameter geändert werden können.

- Um eine Verbindung aus dem Dialogfeld "Konfiguration der Verbindungen" zu löschen, wählen Sie einen Verbindungsnamen aus und klicken dann auf die Schaltfläche **Löschen**. Beachten Sie, dass nach dem Löschvorgang alle Verbindungsparameter permanent verloren sind.
- Um die Parameter einer vorhandenen Verbindung zu ändern, wählen Sie das gewünschte Feld aus und klicken dann auf die Schaltfläche **Ändern**. Dann können Sie einen neuen Wert in das ausgewählte Feld eingeben.

Ethernet-Anzeige-LEDs

Überblick

Im LED-Bereich an der Frontseite der Steuerung TWDLCAE40DRF und an der softwaremäßigen Frontseite, auf die über die Optionen **SPS > SPS überprüfen** in der TwidoSoft-Applikation zugegriffen werden kann, sind zwei Ethernet-Kommunikationsanzeige-LEDs vorhanden. Sie haben folgende Bezeichnungen:·

- LAN ACT
- LAN ST

Die Ethernet-LEDs bieten eine permanente Überwachung des Ethernet-Port-Verbindungsstatus und eine permanente Diagnose.

LED-Status

Die folgende Tabelle beschreibt den Status der Ethernet-Anzeige-LEDs **LAN ACT** und **LAN ST**.

LEDs	Zustand	Farbe	Beschreibung
LAN ACT	aus	-	Kein Ethernet-Signal am RJ-45-Port.
	permanent leuchtend	Grün	10BASE-TX-Verbindungs-Herschlagsignal zur Anzeige einer 10 Mbps-Verbindung.
	blinkend		Es werden Datenpakete über die 10BASE-TX-Verbindung gesendet oder empfangen.
	permanent leuchtend	Gelb	100BASE-TX-Verbindungs-Herzschlagsignal zur Anzeige einer 100 Mbps-Verbindung.
	blinkend		Es werden Datenpakete über die 100BASE-TX-Verbindung gesendet oder empfangen.
LAN ST	permanent leuchtend	Grün	Das SPS-Grundgerät ist eingeschaltet. Der Ethernet-Port ist bereit, um über das Netzwerk zu kommunizieren.
	Schnell blinkend		Ethernet-Initialisierung bei Einschaltung.
	2-maliges Blinken, lange aus		Keine gültige MAC-Adresse.
	3-maliges Blinken, lange aus		Eine beliebige der drei Ursachen: ● Kein Verbindungsherzschlag erkannt. ● Ethernet-Netzwerkkabel nicht richtig angeschlossen oder Kabel defekt. ● Netzwerkgerät (Hub/Switch) fehlerhaft oder nicht richtig konfiguriert.
	4-maliges Blinken, lange aus		Doppelte IP-Adresse im Netzwerk erkannt. (Versuchen Sie, Ihrer Twido-Steuerung eine andere IP-Adresse zuzuweisen, um diese Situation zu beheben.)
	6-maliges Blinken, lange aus		Verwendung einer gültigen, konvertierten Standard-IP-Adresse; FDR-sicherer Modus.
	9-maliges Blinken, lange aus		Ethernet-Hardwarefehler

TCP Modbus-Nachrichtenübertragung

Überblick

Sie können die TCP Modbus-Nachrichtenübertragung verwenden, um es dem Modbus TCP-Client (Master-Steuerung) zu ermöglichen, Ethernet-Meldungen vom Modbus TCP-Server (Slave-Steuerung) zu empfangen bzw. an sie zu senden. Da TCP Modbus ein Protokoll für die Kommunikation zwischen gleichrangigen Steuerungen ist, kann eine Ethernet-fähige Twido-Steuerung sowohl Client als auch Server sein. Dies ist davon abhängig, ob sie Nachrichten abfragt oder diese beantwortet.

Nachrichtenaustausch über das Ethernet-Netzwerk

Die Ethernet-Nachrichtenübertragung wird durch die Anweisung EXCH3 und den Funktionsblock %MSG3 verwaltet: Das Routing an einen Ethernet-Host oder über ein Gateway wird ebenfalls von EXCH3 unterstützt.

- **EXCH3-Anweisung:** zum Senden/Empfangen von Meldungen
- **Funktionsblock %MSG3:** zum Steuern des Meldungsaustauschs.

EXCH3-Anweisung

Mit der Anweisung EXCH3 kann die Twido-Steuerung Informationen vom Ethernet-Netzwerk empfangen bzw. Informationen an das Ethernet-Netzwerk senden. Der Anwender definiert eine Worttabelle (%MWi:L), die Steuerungsinformationen und die Daten enthalten, die gesendet und/oder empfangen werden sollen (maximal 128 Bytes zum Senden und/oder Empfangen). Das Format der Worttabelle wird im folgenden Abschnitt beschrieben.
Ein Meldungsaustausch wird mit der Anweisung EXCH3 durchgeführt:

Syntax: [EXCH3 %MWi:L]
wobei gilt: L = Anzahl der Wörter in der Steuerwort-, Sende- und Empfangstabelle

Die Twido-Steuerung muss den Austausch einer ersten Anweisung EXCH3 beenden, bevor eine zweite Anweisung gestartet werden kann. Der Funktionsblock %MSG3 muss verwendet werden, wenn mehrere Meldungen gesendet werden sollen.
Die Bearbeitung der AWL-Anweisung EXCH3 geschieht sofort und alle Übertragungen werden interruptgesteuert gestartet (auch der Empfang von Daten läuft interruptgesteuert ab), was als Hintergrundbearbeitung betrachtet wird.

Hinweis: Die Verwendung der Anweisung EXCH3 ist identisch mit EXCHx (wobei x = 1 oder 2 ist) der Modbus-Vorgängerversion. Die Anweisungssyntax ist ebenfalls identisch. Es gibt jedoch einen größeren Unterschied in der von Byte1 der Sende- und Empfangstabellen übertragenen Information. Während Byte1 der Modbus-Vorgängerversion die serielle Verbindungsadresse der Slave-Steuerung überträgt, enthält Byte1 von TCP Modbus die **Index**nummer der Modbus TCP-Client-Steuerung. Die Indexnummer wird in der Tabelle der dezentralen Geräte der TwidoSoft-Ethernet-Konfiguration angegeben und gespeichert (ausführliche Informationen finden Sie unter *Registerkarte "Dezentrale Geräte", S. 183*).

EXCH3-Worttabelle

Die maximale Größe der übertragenen und/oder empfangenen Frames beträgt 128 Bytes (beachten Sie, dass dieser Grenzwert nur für den TCP Modbus-Client gilt, während der TCP Modbus-Server die standardmäßige Modbus PDU-Länge von 256 Bytes unterstützt). Die zur Anweisung EXCH3 gehörende Worttabelle besteht aus Steuer-, Sende- und Empfangstabellen, wie nachfolgend beschrieben.

	Höherwertiges Byte	**Niederwertiges Byte**
Steuertabelle	Befehl	Länge (Senden/Empfangen)
	Empfangsverzögerung	Sendeverzögerung
Sendetabelle	Übertragenes Byte 1 (**Index** wie in der Tabelle der dezentralen Geräte des Dialogfelds "TwidoSoft-Ethernet-Konfiguration" angegeben).	Übertragenes Byte 2 als Modbus seriell
	...	Byte n gesendet
	Byte n+1 gesendet	
Empfangstabelle	Empfanges Byte 1 (**Index** wie in der Tabelle der dezentralen Geräte des Dialogfelds "TwidoSoft-Ethernet-Konfiguration" angegeben).	Empfangenes Byte 2 als Modbus seriell
	...	Byte p empfangen
	Byte p+1 empfangen	

Funktionsblock %MSG3

Die Verwendung der Funktion %MSG3 ist identisch mit der in der Modbus-Vorgängerversion verwendeten Funktion %MSGx. %MSG3 wird verwendet, um den Datenaustausch zu verwalten, und umfasst folgende Funktionen:

- Kommunikationsfehlerprüfung
- Koordination mehrerer Meldungen
- Senden von Prioritätsmeldungen

Der Funktionsblock %MSGx hat einen zugehörigen Eingang und zwei zugehörige Ausgänge:

Eingang/Ausgang	**Definition**	**Beschreibung**
R	RESET-Eingang	Auf 1 gesetzt: initialisiert die Kommunikation neu oder setzt den Block zurück (%MSGx.E = 0 und %MSGx.D = 1).
%MSGx.D	Kommunikation beendet	0: Request in Bearbeitung. 1: Kommunikation beendet, wenn Ende der Übertragung, Endezeichen empfangen, Fehler oder Rücksetzen des Blocks.
%MSGx.E	Fehler	0: Meldungslänge OK und Verbindung OK. 1: wenn falscher Befehl, Tabelle falsch konfiguriert, falsches Zeichen empfangen (Geschwindigkeit, Parität usw.) oder Empfangstabelle voll.

EXCH3-Fehlercode

Wenn im Zusammenhang mit der Anweisung EXCH3 ein Fehler auftritt:

- werden die Bits **%MSG3.D** und **%MSG3.E** auf **1** gesetzt, und
- der Ethernet-Kommunikations**fehlercode** wird im Systemwort **%SW65** gespeichert.

Die folgende Tabelle enthält die EXCH3-Fehlercodes:

EXCH3-Fehlercode (gespeichert im Systemwort %SW65)
Standard-Fehlercodes, gültig für alle EXCHx (x = 1, 2, 3): 0 - Vorgang war erfolgreich 1 - Anzahl zu sendender Bytes zu hoch (> 128) 2 - Sendetabelle zu klein 3 - Worttabelle zu klein 4 - Überlauf Empfangstabelle 5 - Timeout abgelaufen (beachten Sie, dass Fehlercode 5 ungültig ist und durch die nachfolgend beschriebenen, Ethernet-spezifischen Fehlercodes 109 und 122 ersetzt wurde) 6 - Senden 7 - Falscher Befehl in Tabelle 8 - Gewählte Schnittstelle nicht konfiguriert/verfügbar 9 - Empfangsfehler 10 - %KW kann nicht beim Empfang verwendet werden 11 - Sende-Offset größer als Sendetabelle 12 - Empfangs-Offset größer als Empfangstabelle 13 - Steuerung hat Bearbeitung von EXCH gestoppt
Ethernet-spezifische Fehlercodes für EXCH3: 101 - keine derartige IP-Adresse 102 - Die TCP-Verbindung ist unterbrochen 103 - Kein Socket verfügbar (alle Verbindungskanäle sind belegt) 104 - Netzwerk ausgefallen 105 - Keine Verbindung zum Netzwerk möglich 106 - Unterbrechung der Netzwerkverbindung beim Reset 107 - Verbindung durch Partnergerät getrennt 108 - Verbindung durch Partnergerät zurückgesetzt 109 - Verbindungs-Timeout abgelaufen 110 - Zurückweisung bei Verbindungsaufbau 111 - Host ausgefallen 120 - Unbekannter Index (das dezentrale Gerät ist in der Konfigurationstabelle nicht indexiert) 121 - schwer (MAC, Chip, doppelte IP)122 - Empfangs-Timeout nach dem Senden der Daten abgelaufen 123 - Ethernet-Initialisierung läuft

Integrierte Analogfunktionen

7

Auf einen Blick

Inhalt dieses Kapitels

In diesem Kapitel wird die Verwaltung des integrierten Analogkanals und der Analogpotentiometer beschrieben.

Inhalt dieses Kapitels

Dieses Kapitel enthält die folgenden Themen:

Thema	Seite
Analoger Potentiometer	194
Analogkanal	196

Analoger Potentiometer

Einführung

Twido-Steuerungen besitzen:

- Ein analoges Potentiometer an den Steuerungen TWDLC•A10DRF und TWDLC•A16DRF und an allen modularen Steuerungen (TWDLMDA20DTK, TWDLMDA20DUK, TWDLMDA20DRT, TWDLMDA40DTK und TWDLMDA40DUK),
- Zwei Potentiometer an den Steuerungen TWDLC•A24DRF und TWDLCA•40DRF.

Nicht reversible

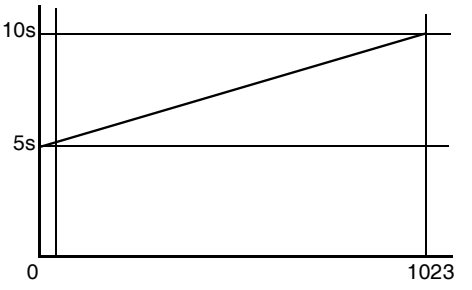
Die numerischen Werte, von 0 bis 1023 für das analoge Potentiometer 1 und von 0 bis 511 für das analoge Potentiometer 2, die den Analogwerten dieser Potentiometer entsprechen, sind in den folgenden beiden Eingangsworte enthalten:

- %IW0.0.0 für das analoge Potentiometer 1 (auf der linken Seite)
- %IW0.0.1 für das analoge Potentiometer 2 (auf der rechten Seite)

Diese Wörter können in arithmetischen Operationen verwendet werden. Sie können für jede Art von Anpassung genutzt werden, z.B. für die Voreinstellung einer Zeitverzögerung oder eines Zählers oder für die Anpassung der Frequenz eines Impulsgenerators oder der Vorwärmzeit einer Maschine.

Beispiel

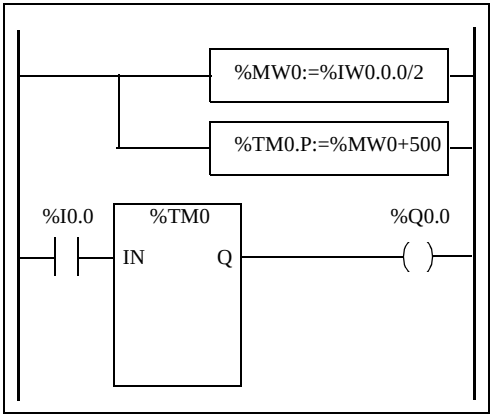
Anpassen der Dauer einer Zeitverzögerung von 5 auf 10 s mit dem analogen Potentiometer 1:

Für diese Anpassung wird praktisch der gesamte Einstellbereich des analogen Potentiometers 1 von 0 bis 1023 verwendet.	

Die folgenden Parameter werden während der Konfiguration des Zeitverzögerungsblocks %TM0 gewählt:

- Typ TON
- Zeitbasis (TB): 10 ms

Der Preset-Wert der Zeitverzögerung wird anhand des Einstellwerts des analogen Potentiometers mittels folgender Gleichung berechnet: %TM0.P := (%IW0.0.0/2)+500.
Code für das oben aufgeführte Beispiel:




```
LD      1
[%MW0:=%IW0.0.0/2]
[%TM0.P:=%MW0+500]
BLK     %TM0
LD      %I0.0
IN
OUT_BLK
LD      Q
ST      %Q0.0
END_BLK
.....................
```

Analogkanal

Einleitung

Alle modularen Steuerungen (TWDLMDA20DTK, TWDLMDA20DUK, TWDLMDA20DRT, TWDLMDA40DTK und TWDLMDA40DUK) haben einen integrierten Analogkanal. Die Eingangsspannung liegt zwischen 0 und 10 V und zwischen 0 und 511 für das digitalisierte Signal. Der analoge Kanal verwendet ein einfaches Mittelwert-Berechnungsschema, das auf acht Messungen angewendet wird.

Prinzip

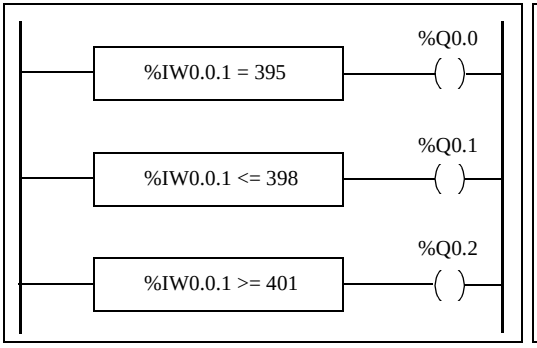
Ein Analog/Digital-Wandler nimmt eine Spannungsprobe aus dem Bereich zwischen 0 und 10 V und eine Digitalwertprobe aus dem Bereich zwischen 0 und 511. Dieser Wert wird im Systemwort %IW0.0.1 gespeichert. Der Wert ist zum gesamten Bereich linear, und jedes Inkrement beträgt 20 mV (10 V/512). Wenn das System den Wert 511 erkennt, wird der Kanal als gesättigt angesehen.

Programmier-beispiel

Regeln der Temperatur eines Ofens: Die Kochtemperatur ist auf 350°C eingestellt. Eine Änderung von +/- 2,5°C ruft eine Netzunterbrechung der Ausgänge %Q0.0 und %Q0.2 hervor. In diesem Beispiel werden praktisch alle möglichen Einstellbereiche des Analogkanals von 0 bis 511 verwendet. Im Folgenden sehen Sie die analogen Parameter der verschiedenen Temperaturen:

Temperatur (°C)	Spannung	Systemwort %IW0.0.1
0	0	0
347,5	7,72	395
350	7,77	398
352,5	7,83	401
450	10	511

Code für das obenstehende Beispiel:




```
LD      [%IW0.0.1 = 395]
ST      %Q0.0

LD      [%IW0.0.1 <= 398]
ST      %Q0.1

LD      [%IW0.0.1 >= 401]
ST      %Q0.2
```

Verwalten von Analogmodulen

8

Auf einen Blick

Inhalt dieses Kapitels

Dieses Kapitel gibt einen Überblick über die Verfahren zur Verwaltung der analogen Module der Twido-Steuerungen.

Inhalt dieses Kapitels

Dieses Kapitel enthält die folgenden Themen:

Thema	Seite
Übersicht über Analogmodule	198
Adressierung von analogen Ein- und Ausgängen	199
Konfiguration der analogen Ein- und Ausgänge	200
Statusinformationen zum Analogmodul	207
Beispiel für die Verwendung von Analogmodulen	208

Übersicht über Analogmodule

Einleitung

Neben dem analogen 10-Bit-Potentiometer und dem analogen 9-Bit-Kanal können auch die Twido-Steuerungen, die für die E/A-Erweiterung sorgen, mit den analogen E/A-Modulen kommunizieren.
Im Folgenden sehen Sie eine Liste der Analogmodule:

Name	Kanäle	Signalbereich	Codierung
TWDAMI2HT	2 Eingänge	0 - 10 Volt oder 4 - 20 mA	12 Bits
TWDAMO1HT	1 Ausgang	0 - 10 Volt oder 4 - 20 mA	12 Bits
TWDAMM3HT	2 Eingänge, 1 Ausgang	0 - 10 Volt oder 4 - 20 mA	12 Bits
TWDALM3LT	2 Eingänge, 1 Ausgang	0 - 10 V, Th- oder PT100- Eingänge, Ausgänge von 4 bis 20 mA	12 Bits
TWDAVO2HT	2 Ausgang	+/- 10 Volt	11 Bits + Vorzeichen
TWDAMI4LT	4 Eingänge	0 - 10 Volt, 0 - 20 mA, 3-adrige NI- oder PT-Sensoren	12 Bits
TWDAMI8HT	8 Eingänge	0 - 10 Volt oder 0 - 20 mA	10 Bits
TWDARI8HT	8 Eingänge	NTC- oder PTC-Sensoren	10 Bits

Betrieb von Analogmodulen

Die Eingangs- und Ausgangworte (%IW und %QW) werden zum Datenaustausch zwischen der Benutzerapplikation und den analogen Kanälen verwendet. Die Aktualisierung dieser Worte erfolgt synchronisiert mit der Steuerungsabtastung im RUN-Modus.

⚠ ACHTUNG

UNERWARTETER ANLAUF VON GERÄTEN

Wenn die Steuerung in den STOP-Modus geschaltet ist, befindet sich der Analogausgang in Fehlerposition. Im Falle eines Digitalausgangs ist der Standard-Sollwert gleich Null.

Die Nichtbeachtung dieser Anweisung kann Körperverletzung oder Materialschäden zur Folge haben!

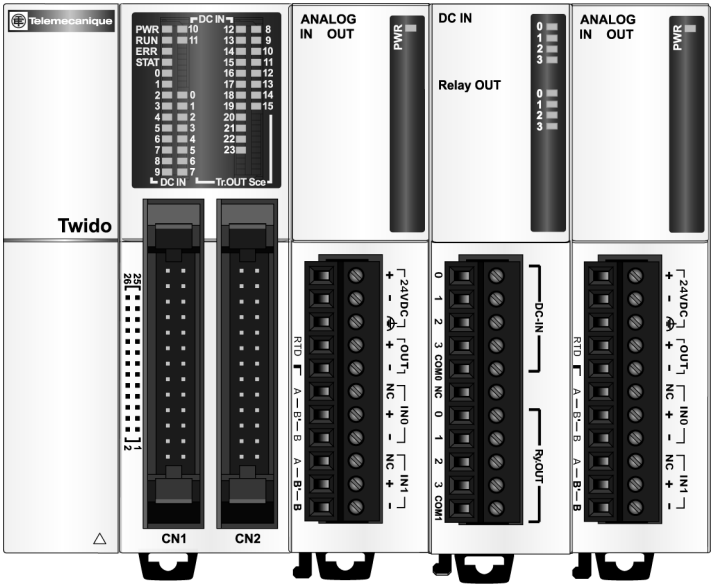
Adressierung von analogen Ein- und Ausgängen

Einleitung

Den analogen Kanälen werden abhängig von ihrem Platz am Erweiterungsbus Adressen zu gewiesen.

Beispiel für die Adressierung von analogen E/A

In diesem Beispiel verfügt TWDLMDA40DUK über ein integriertes 10-Bit-Potentiometer und einen integrierten 9-Bit-Analogkanal. Am Erweiterungsbus ist Folgendes konfiguriert: Das Analogmodul TWDAMM3HT, das digitale E/A-Relaismodul TWDDMM8DRT und ein zweites Analogmodul TWDAMM3HT.



Grundgerät SPS Modul 1 Modul 2 Modul 3

Die folgende Tabelle zeigt die Adressierung der einzelnen Ausgänge.

Beschreibung	Grundgerät SPS	Modul 1	Modul 2	Modul 3
Potentiometer 1	%IW0.0.0			
Integrierter Analogkanal	%IW0.0.1			
Analogeingangskanal 1		%IW0.1.0		%IW0.3.0
Analogeingangskanal 2		%IW0.1.1		%IW0.3.1
Analogausgangskanal 1		%QW0.1.0		%QW0.3.0
Digitaleingangskanäle			%I0.2.0 - %I0.2.3	
Digitalausgangskanäle			%Q0.2.0 -%Q0.2.3	

Konfiguration der analogen Ein- und Ausgänge

Einleitung

In diesem Abschnitt wird beschrieben, wie Sie die Eingänge und Ausgänge von Analogmodulen konfigurieren.

Konfigurieren von analogen E/A

Im Dialogfeld **Modul konfigurieren** verwalten Sie die Parameter der Analogmodule.
Sie können es über den Applikationsnavigator oder das Menü **Hardware** aufrufen.

Über den Applikationsnavigator	Über das Menü Hardware
1. Wählen Sie ein Modul aus.	1. Wählen Sie **Modul konfigurieren** aus.
2. Klicken Sie mit der rechten Maustaste auf **Konfigurieren**, um direkt das Dialogfeld **Modul konfigurieren - (Modulref. und -position)** aufzurufen.	2. Wählen Sie ein Modul im Dialogfeld **Modul konfigurieren - Modul auswählen** aus.
	3. Konfigurieren Sie die Parameter in dem sich öffnenden Dialogfeld **Modul konfigurieren - (Modulref. und -position)**.

Hinweis: Sie können die Parameter nur dann offline ändern, wenn Sie nicht mit einer Steuerung verbunden sind.

Titelleiste und Inhalte

In der Titelleiste werden die Modulreferenz und die Position des Moduls auf dem Erweiterungsbus angezeigt.
Im oberen Teil des Dialogfelds befindet sich der Bereich **Beschreibung**.
Eine Tabelle zeigt folgende Informationen an: **Adresse**, **Symbol**, **Typ**, **Bereich**, **Minimum**, **Maximum** und **Einheiten**

- Beim TWDAMI4LT und TWIDAMI8HT wird der Tabelle das Listenfeld **Eingangstyp** vorangestellt.
- Beim TWDAVO2HT und TWDAMI8HT ist die Spalte **Typ** durch die Spalte **Verwendet** mit Kontrollkästchen ersetzt.
- Beim TWDARI8HT wird jeder Kanal (0-7) einzeln auf einer Registerkarte konfiguriert, auf der Sie entweder das **Diagramm-** oder **Formel-**Konfigurationsverfahren auswählen können. Die Tabelle ist auf der Registerkarte **Zusammenf.** enthalten.

Beschreibung

Der Bereich **Beschreibung** enthält eine Kurzbeschreibung des Moduls.

Adresse

Jede Zeile der Tabelle stellt entweder einen Eingangs- oder einen Ausgangskanal des Moduls dar.
Die einzelnen Adressen sind in folgender Tabelle zusammengestellt, wobei "i" die Position des Moduls auf dem Erweiterungsbus angibt.

Modulname	Adresse
TWDALM3LT	2 Eingänge (%IWi.0, %IWi.1), 1 Ausgang (%QWi.0)
TWDAMM3HT	2 Eingänge (%IWi.0, %IWi.1), 1 Ausgang (%QWi.0)
TWDAMI2HT	2 Eingänge (%IWi.0, %IWi.1)
TWDAMO1HT	1 Ausgang (%QWi.0)
TWDAVO2HT	2 Ausgänge (%QWi.0, %QWi.1)
TWDAMI4LT	4 Eingänge (%IWi.0 bis %IWi.3)
TWDAMI8HT	8 Eingänge (%IWi.0 bis %IWi.7)
TWDARI8HT	8 Eingänge (%IWi.0 bis %IWi.7)

Symbol

Dies ist eine schreibgeschützte Anzeige des Symbols für diese Adresse (falls zugewiesen).

Eingangstyp und/oder Typ

Gibt den Modus eines Kanals an. Die Auswahl ist vom Kanal und vom Modultyp abhängig.
Sie können den Typ des Ausgangskanals der Module TWDAMO1HT, TWDAMM3HT und TWDALM3LT wie folgt konfigurieren:

Typ
Nicht verwendet
0 - 10 V
4 - 20 mA

Sie können die beiden Eingangskanaltypen der Module TWDAMI2HT und TWDAMM3HT wie folgt konfigurieren:

Typ
Nicht verwendet
0 - 10 V
4 - 20 mA

Sie können die beiden Eingangskanaltypen des Moduls TWDALM3LT wie folgt konfigurieren:

Typ
Nicht verwendet
Thermoelement K
Thermoelement J
Thermoelement T
PT 100

Für das Modul TWDAVO2HT muss kein Typ konfiguriert werden.
Sie können die vier Eingangskanaltypen des Moduls TWDAMI4LT wie folgt konfigurieren:

Eingangstyp	**Typ**
Spannung	Nicht verwendet 0 - 10 V
Strom	Nicht verwendet 0 - 20 mA
Temperatur	Nicht verwendet PT 100 PT 1000 NI 100 NI 1000

Sie können die acht Eingangskanaltypen des Moduls TWDAMI8HT wie folgt konfigurieren:

Eingangstyp
0 - 10 V
0 - 20 mA

Beim TWDARI8HT können Sie jeden Eingangskanal (0-7) individuell im Feld **Betrieb** im unteren Teil des Fensters konfigurieren. Wählen Sie direkt einen **Modus** und erforderlichenfalls einen **Bereich** aus. Sie können dann eine Zusammenfassung aller Informationen auf der Registerkarte "Zusammenf." anzeigen, wobei die Spalte **Typ** die folgenden Informationen anzeigt:

Typ
Nicht verwendet
NTC / CTN
PTC / CTP

⚠ ACHTUNG

GERÄTESCHADEN

Wenn Sie Ihren Eingang für eine Voltmessung verdrahtet haben und Sie TwidoSoft für eine Stromkonfiguration konfigurieren, kann dies zu einem irreparablen Schaden des Analogmoduls führen. Stellen Sie sicher, dass die Verdrahtung mit der TwidoSoft-Konfiguration übereinstimmt.

Die Nichtbeachtung dieser Anweisung kann Körperverletzung oder Materialschäden zur Folge haben!

Bereich Gibt den Wertebereich eines Kanals an. Die Auswahl ist vom speziellen Kanal- und Modultyp abhängig.
Wenn der **Typ** konfiguriert ist, können Sie den entsprechenden **Bereich** festlegen. Eine Tabelle zeigt die akzeptierten **Minimum-** und **Maximum**werte - entweder unveränderlich oder benutzerdefiniert - zusammen mit der **Einheit** (falls erforderlich).

Bereich (NTC-Sensoren)	**Minimum**	**Maximum**	**Einheiten**	**E/A-Analogmodule**
Normal	0	4095	Keine	TWDALM3LT TWDAMO1HT TWDAMM3HT TWDAMI2HT TWDAMI4LT
	-2048	2047		TWDAVO2HT
	0	1023		TWDAMI8HT TWDARI8HT
Benutzerdefiniert	Benutzerdefiniert mit einem Minimum von -32768	Benutzerdefiniert mit einem Maximum von 32767	Keine	Alle E/A-Analogmodule
Celsius	-1000	5000	0,1 °C	TWDALM3LT
	Dynamisch aktualisiert durch TwidoSoft gemäß Parametern wird von Verbraucher definiert.			TWDARI8HT
	-2000	6000		TWDAMI4LT (Pt Sensoren)
	-500	1500		TWDAMI4LT (Ni Sensoren)
Fahrenheit	-1480	9320	0,1 °F	TWDALM3LT
	Dynamisch aktualisiert durch TwidoSoft gemäß Parametern wird von Verbraucher definiert.			TWDARI8HT
	-3280	11120		TWDAMI4LT (Pt Sensoren)
	-580	3020		TWDAMI4LT (Ni Sensoren)

Bereich (NTC-Sensoren)	Minimum	Maximum	Einheiten	E/A-Analogmodule
Widerstand	100	10000	Ohm	TWDARI8HT
	74	199		TWDAMI4LT (Ni100)
	742	1987		TWDAMI4LT (Ni1000)
	18	314		TWDAMI4LT (Pt100)
	184	3138		TWDAMI4LT (Pt1000)

Diagramm- oder Formel-Verfahren

Beim TWDARI8HT wird jeder Kanal (0-7) einzeln auf einer Registerkarte konfiguriert. Aktivieren Sie das Kontrollkästchen **Verwendet**, und wählen Sie dann zwischen den Konfigurationsverfahren **Diagramm** und **Formel** aus.

- **Diagramm-Verfahren (graphisch)**
  (**R1**, **T1**) und (**R2**, **T2**) entsprechen den Gleitkomma-Koordinaten von zwei zur Kurve gehörenden Punkten.
  Die Werte **R1**(Standardwert 8700) und **R2** (Standardwert 200) werden in Ohm angegeben.
  Für die Werte **T1** (Standardwert 233,15) und **T2** (Standardwert 398,15) kann die Einheit im Listenfeld **Einheit** ausgewählt werden: **Kelvin** (Standardwert), **Celsius** oder **Fahrenheit**.
  Hinweis: Die Änderung der Temperatureinheit nach der Eingabe der Werte T1 und T2 führt nicht zur automatischen Neuberechnung der Werte T1 und T2 mit der neuen Einheit.
- **Formel-Verfahren**
  Wenn Sie die Parameter **Rref**, **Tref** und **B** kennen, können Sie dieses Verfahren verwenden, um die Sensoreigenschaften festzulegen.
  Rref (Standardwert 330) wird in Ohm angegeben.
  Der Standardwert von **B** lautet 3569 (min. 1, max. 32767).
  Für den Wert **Tref** (Standardwert 298,15) kann die Einheit im Listenfeld **Einheit** ausgewählt werden: **Kelvin** (Standardwert), **Celsius** oder **Fahrenheit**.

Die folgende Tabelle zeigt die entsprechenden min./max. **Tref**-Werte für die jeweiligen Einheiten.

Einheit	Min. Wert	Max. Wert
Kelvin	1	650
Celsius	-272	376
Fahrenheit	-457	710

Sowohl im Diagramm- als auch im Formelfenster können Sie Werte von einem anderen Kanal in den aktuell konfigurierten Kanal importieren.

1. Wählen Sie eine Kanalnummer aus dem Feld **Kanalnr.** aus.
2. Klicken Sie auf die Schaltfläche **Werte importieren**.

Mit diesen Fenstern können einige Fehler- und Warnmeldungen verbunden sein.

Hinweis: Wenn Sie die Eingabe von Werten beginnen und sich dann entschließen, von Diagramm zu Formel oder von Formel zu Diagramm zu wechseln, wird eine Warnmeldung angezeigt, die Sie darauf aufmerksam macht, dass die Standardwerte wiederhergestellt werden und die geänderten Werte verloren gehen.

Statusinformationen zum Analogmodul

Statustabelle Die folgende Tabelle enthält die Informationen, die Sie zum Überwachen des Status der analogen E/A-Module benötigen.

Systemwort	Funktion	Beschreibung
%SW80	E/A-Status Grundgerät	Bit [0] Kanäle im Normalbetrieb (für alle Kanäle) Bit [1] Modul wird initialisiert (oder Initialisierungsinformationen für alle Kanäle) Bit [2] Hardwarestörung (Ausfall der externen Spannungsversorgung, für alle Kanäle gemeinsam) Bit [3] Modulkonfigurationsfehler Bit [4] Datenkonvertierung Eingangskanal 0 läuft Bit [5] Datenkonvertierung Eingangskanal 1 läuft Bit [6] Eingang Thermoelementkanal 0 nicht konfiguriert Bit [7] Eingang Thermoelementkanal 1 nicht konfiguriert Bit [8] Frei Bit [9] Nicht verwendet Bit [10] Bereichsüberschreitung Datenkanal Analogeingang 0 Bit [11] Bereichsüberschreitung Datenkanal Analogeingang 1 Bit [12] Falsche Verdrahtung (Bereichsunterschreitung Datenkanal Analogeingang 0, Stromschleife offen) Bit [13] Falsche Verdrahtung (Bereichsunterschreitung Datenkanal Analogeingang 1, Stromschleife offen) Bit [14] Frei Bit [15] Ausgangskanal nicht verfügbar
%SW81	Status Erweiterungs-E/A-Modul 1: Gleiche Definitionen wie %SW80	
%SW82	Status Erweiterungs-E/A-Modul 2: Gleiche Definitionen wie %SW80	
%SW83	Status Erweiterungs-E/A-Modul 3: Gleiche Definitionen wie %SW80	
%SW84	Status Erweiterungs-E/A-Modul 4: Gleiche Definitionen wie %SW80	
%SW85	Status Erweiterungs-E/A-Modul 5: Gleiche Definitionen wie %SW80	
%SW86	Status Erweiterungs-E/A-Modul 6: Gleiche Definitionen wie %SW80	
%SW87	Status Erweiterungs-E/A-Modul 7: Gleiche Definitionen wie %SW80	

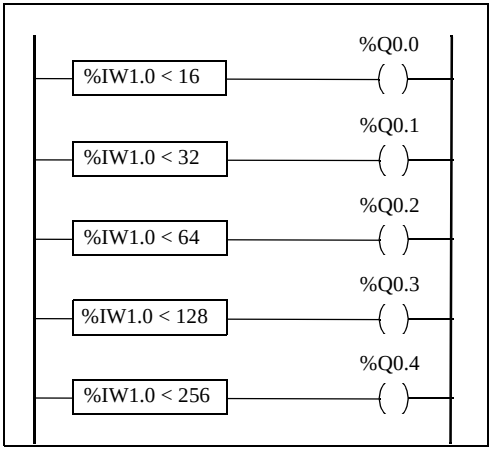
Beispiel für die Verwendung von Analogmodulen

Einleitung

Dieser Abschnitt liefert ein Beispiel für die Verwendung von Analogmodulen mit Twido.

Beispiel: Analogeingang

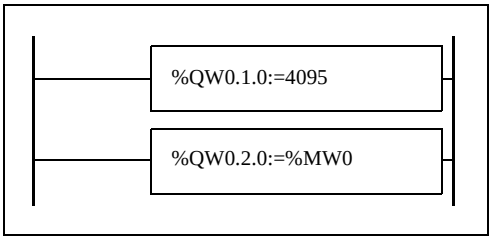
In diesem Beispiel wird das Analogeingangssignal mit fünf einzelnen Grenzwerten verglichen. Der Analogeingang wird verglichen und es wird in der Basissteuerung ein Bit gesetzt, wenn der Analogeingang kleiner als der Grenzwert ist.




```
LD [%IW1.0 < 16]
ST %Q0.0

LD [%IW1.0 < 32]
ST %Q0.1

LD [%IW1.0 < 64]
ST %Q0.2

LD [%IW1.0 < 128]
ST %Q0.3

LD [%IW1.0 < 256]
ST %Q0.4
```

Beispiel: Analogausgang

Im nachfolgenden Programm wird eine analoge Karte an den Steckplätzen 1 und 2 verwendet. Die in Steckplatz 1 verwendete Karte hat einen 10-Volt-Ausgang mit "normalem" Bereich:




```
LD   1
[%QW0.1.0:=4095
LD   1
[%QW0.2.0:=%MW0
```

- Beispiel für Ausgangswerte für %QW1.0=4095 (normaler Fall):

Die folgende Tabelle zeigt die Ausgangsspannungswerte gemäß dem %QW1.0 zugewiesenen maximalen Wert.

	Numerischer Wert	**Analoger Wert (Volt)**
Minimum	0	0
Maximum	4095	10
Wert 1	100	0,244
Wert 2	2460	6

- Beispiel der Ausgangswerte für einen individuell angepassten Bereich (Minimum =0, Maximum =1000):

Die folgende Tabelle zeigt die Ausgangsspannungswerte gemäß dem %QW1.0 zugewiesenen maximalen Wert.

	Numerischer Wert	**Analoger Wert (Volt)**
Minimum	0	0
Maximum	1000	10
Wert 1	100	1
Wert 2	600	6

Inbetriebnahme des AS-Interface V2-Busses

9

Auf einen Blick

Inhalt dieses Kapitels

Dieses Kapitel enthält Einzelheiten zur softwaretechnischen Inbetriebnahme des AS-Interface-Master-Moduls TWDNOI10M3 und seiner Slaves.

Inhalt dieses Kapitels

Dieses Kapitel enthält die folgenden Themen:

Thema	Seite
Beschreibung des AS-Interface V2-Busses	212
Allgemeine Funktionsbeschreibung	213
Prinzipien der softwaretechnischen Inbetriebnahme	215
Beschreibung des Konfigurationsfensters des AS-Interface-Bus	216
Konfiguration des AS-Interface-Busses	218
Beschreibung des Debug-Fensters	224
Ändern der Adresse eines Slaves	227
Aktualisierung der Konfiguration des AS-Interface-Busses im Online-Modus	229
Automatische Adressierung eines AS-Interface V2-Slave	234
Einfügen eines Slave-Geräts in eine bestehende AS-Interface V2-Konfiguration	235
Automatischer Ersatz eines fehlerhaften AS-Interface V2-Slave	236
Adressierung der Ein-/Ausgänge, die den am AS-Interface V2-Bus angeschlossenen Slave-Geräten zugeordnet sind.	237
Programmierung und Diagnose des AS-Interface V2-Busses	238
Betriebsart des AS-Interface V2-Bus-Schnittstellenmoduls	243

Beschreibung des AS-Interface V2-Busses

Einführung

Der AS-Interface-Bus (Actuator Sensor-Interface) ermöglicht die Verbindung von Gebern/Stellgliedern auf der niedrigsten Ebene der Automatisierung mittels eines einzigen Kabels.
Diese Geber/Stellglieder werden in der Dokumentation als **Slave-Geräte** definiert.

Die Inbetriebnahme der Applikationsfunktion AS-Interface erfordert das Definieren des physischen Kontextes der Applikation, in die sie integriert werden soll (Erweiterungsbus, Versorgung, Prozessor, Module, an den Bus angeschlossene AS-Interface-Slave-Geräte) und anschließend die Sicherstellung ihrer Software-Inbetriebnahme.

Dieser zweite Aspekt wird von den verschiedenen Editoren von TwidoSoft aus realisiert:
- entweder im Offline-Betrieb
- oder im Online-Modus.

AS-Interface V2-Bus

Das AS-Interface-Master-Modul **TWDNOI10M3** beinhaltet folgende Funktionen:
- M3 -Profil: Dieses Profil umfasst alle vom Standard AS-Interface V2 definierten Funktionen jedoch nicht die analogen Profile S7-4.
- einen AS-Interface-Kanal pro Modul.
- Automatische Adressierung für den Slave mit der Adresse 0
- Verwaltung von Profilen und Parametern.
- Verpolungsschutz an den Buseingängen.

Der AS-Interface-Bus unterstützt:
- bis zu 31 Slaves mit Standardadressierung und 62 Slaves mit erweiterter Adressierung,
- bis zu 248 Eingänge und 186 Ausgänge,
- bis zu 7 analoge Slaves (max. 4 E/A pro Slave),
- eine Zykluszeit von maximal 10 ms.

Es können maximal 2 AS-Interface-Master-Module an eine modulare Twido-Steuerung oder eine kompakte Steuerung der Serie TWDLC•A24DRF oder TWDLCA•40DRF angeschlossen werden.

Allgemeine Funktionsbeschreibung

Allgemeine Beschreibung

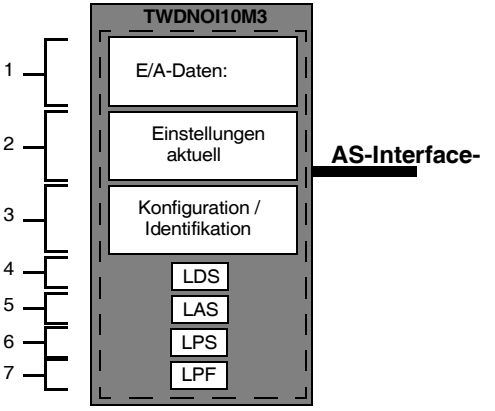
TwidoSoft ermöglicht dem Benutzer für die AS-Interface-Konfiguration:

- Manuelles Konfigurieren des Busses (Deklarieren der Slaves und Zuweisen der Adressen am Bus),
- Anpassen der Konfiguration an die am Bus vorhandenen Geräte,
- Berücksichtigen der Slave-Parameter,
- Steuern des Buszustandes.

Zu diesem Zweck werden alle Informationen vom oder für den AS-Interface-Master in Spezialobjekten (Worte oder Bits) gespeichert.

Struktur des AS-Interface-Masters

Das AS-Interface-Modul besitzt Datenfelder, mit denen Slavelisten und die Bilder der Ein-/Ausgangsdaten verwaltet werden können. Diese Informationen werden im flüchtigen Speicher gespeichert.
Die nachfolgende Abbildung zeigt die Architektur des Moduls **TWDNOI10M3**.



Legende:

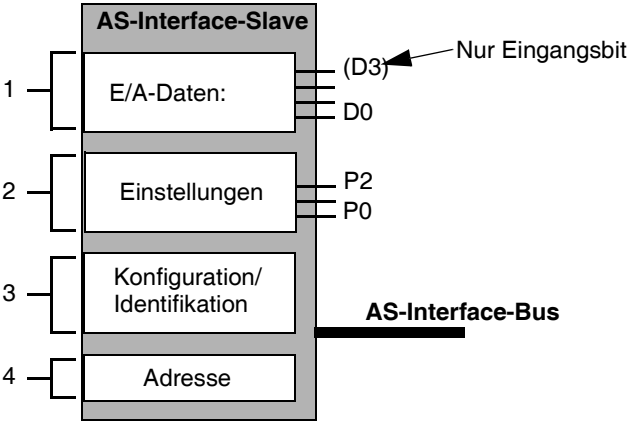
Adresse	Element	Beschreibung
1	E/A-Daten: (IDI, ODI)	Bilder der 248 Eingänge und der 186 Ausgänge des AS-Interface V2-Busses.
2	Aktuelle Parameter (PI, PP)	Bild der Parameter aller Slaves.
3	Konfiguration/Identifikation (CDI, PCD)	Dieses Feld enthält alle E/A-Identifikations-Codes aller erkannter Slaves.
4	LDS	Liste aller auf dem Bus gefundener Slaves.
5	LAS	Liste der auf dem Bus aktivierten Slaves.
6	LPS	Liste der für den Bus vorgesehenen und durch TwidoSoft konfigurierten Slaves.
7	LPF	Liste der Slaves mit einem dezentralen Fehler.

Struktur von Slavegeräten

Die Slaves mit Standardadressierung verfügen jeweils über:
- 4 Eingangs-/Ausgangsbits,
- 4 Parametrierungsbits.

Die Slaves mit erweiterter Adressierung verfügen jeweils über:
- 4 Eingangs-/Ausgangsbits (letztes Bit ausschließlich für den Eingang reserviert),
- 3 Parametrierungsbits.

Jeder Slave besitzt seine eigene Adresse sowie ein Profil und ein Unterprofil (Definition zum Austausch von Variablen).
Die nachfolgende Abbildung zeigt den Aufbau eines Slaves mit erweiterter Adressierung:



Legende:

Adresse	Element	Beschreibung
1	Ein-/ Ausgangsdaten	Die Eingangsdaten werden vom Slave gespeichert und dem AS-Interface-Master zur Verfügung gestellt. Die Ausgangsdaten werden vom Mastermodul aktualisiert.
2	Einstellungen	Mit den Parametern können die Betriebsarten im Sensor oder Stellglied gesteuert und umgeschaltet werden.
3	Konfiguration/ Identifikation	Dieses Feld enthält: ● Den Code, der der Konfiguration der Ein-/Ausgänge (E/A) entspricht, ● den Identifikationscode des Slaves (ID), ● die Unter-Identifikationscodes des Slaves (ID1 und ID2).
4	Adresse	Physische Adresse des Slaves.
Bemerkung: Die Funktions- und Adressparameter sowie die Konfigurations- und Identifikationsdaten werden in einem nichtflüchtigen Speicher gespeichert.		

Prinzipien der softwaretechnischen Inbetriebnahme

Auf einen Blick

Entsprechend des Ansatzes von TwidoSoft muss der Benutzer schrittweise vorgehen, um eine AS-Interface-Applikation zu erstellen.

Prinzip der Inbetriebnahme

Der Benutzer muss wissen, wie ein AS-Interface-Bus funktionell konfiguriert wird (Siehe *Einfügen eines Slave-Geräts in eine bestehende AS-Interface V2-Konfiguration, S. 235*). Die nachfolgende Tabelle beschreibt die verschiedenen Phasen der softwaretechnischen Inbetriebnahme des AS-Interface V2-Busses.

Modus	Phase	Beschreibung
Offline-Zustand	Deklaration des Moduls	Wahl der Position des AS-Interface-Master-Moduls TWDNOI10M3 am Erweiterungsbus.
	Konfiguration des Kanals des Moduls	Wahl der Master-Modi.
	Deklaration der Slave-Geräte	Wahl für jedes Gerät: • seiner Positionsnummer am Bus, • des Slave-Typs (Standardadressierung oder erweiterte Adressierung).
	Bestätigung der Konfigurationsparameter	Bestätigung auf Slave-Ebene
	Globale Bestätigung der Applikation	Bestätigung der Applikations-Ebene
Offline oder online	Symbolisierung (optional)	Symbolisierung der Variablen, die den Slave-Geräten zugeordnet sind
	Programmieren	Programmierung der AS-Interface V2-Funktion.
Online-Zustand	Transfer	Transfer der Applikation in die Steuerung
	Debug	Debug der Applikation mit Hilfe: • des Debug-Fensters, das die Anzeige der Slaves (Adresse, Parameter) und die Adressierung der Slaves an den gewünschten Adressen ermöglicht. • von Diagnosefenstern zum Identifizieren der Fehler.

Hinweis: Die Deklarierung und das Löschen des AS-Interface-Master-Moduls am Erweiterungsbus erfolgen wie für jedes andere Erweiterungsmodul. Wenn jedoch bereits zwei AS-Interface-Master-Module am Erweiterungsbus deklariert sind, erlaubt TwidoSoft kein Deklarieren eines weiteren Moduls.

Vorsichtsmaßnahmen vor Erstellung der Verbindung

Vor dem Verbinden (softwaretechnisch) des PCs mit der Steuerung und zum Vermeiden von Erkennungsproblemen:

- Vergewissern Sie sich, dass kein Slave an der Adresse 0 des Busses physisch vorhanden ist,
- Vergewissern Sie sich, dass keine 2 Slaves an derselben Adresse des Busses physisch vorhanden sind.

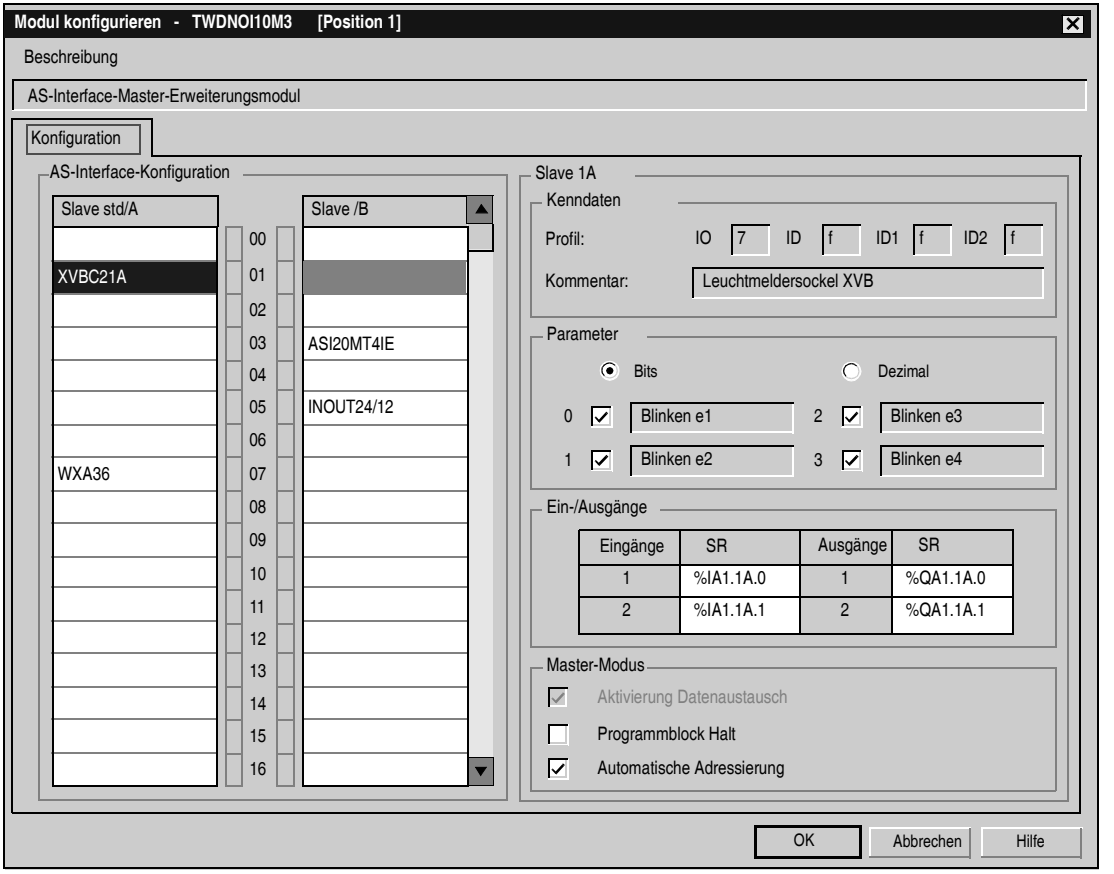
Beschreibung des Konfigurationsfensters des AS-Interface-Bus

Auf einen Blick

Das Konfigurationsfenster des AS-Interface-Mastermoduls ermöglicht den Zugriff auf die dem Modul und den Slave-Geräten zugeordneten Parameter.
Es ermöglicht die Anzeige und die Änderung der Parameter im Offline-Modus.

Abbildung im Offline-Modus

Abbildung des Konfigurationsfensters im Offline-Modus



Beschreibung des Fensters im Offline-Modus

Dieses Fenster enthält alle Informationen über den Bus in drei Informationsblöcken:

Blöcke	Beschreibung
AS-Interface-Konfiguration	Vom Benutzer gewünschtes Bus-Abbild: Anzeige der auf dem Bus geplanten (vorgesehenen) Slaves mit Standardadressierung und erweiterter Adressierung. Die vertikale Bildlaufleiste ist nach unten zu verschieben, um auf die folgenden Adressen zugreifen zu können. Die abgeblendeten Adressen entsprechen für die Konfiguration eines Slaves nicht verfügbaren Adressen. Wenn beispielsweise ein neuer Standard-Slave an Adresse 1A deklariert wird, wird die Adresse 1B automatisch abgeblendet.
Slave xxA/B	Konfiguration des ausgewählten Slave: ● Kenndaten: IO-Code, ID-Code, ID1- und ID2-Codes (Profile) und Kommentare über den Slave ● Parameter: Liste der Parameter (veränderbar) in binärer Form (4 Kontrollkästchen) oder dezimaler Form (1 Kontrollkästchen) (durch den Benutzer wählbar) ● Ein-/Ausgänge: Liste der verfügbaren E/A und ihren jeweiligen Adressen
Master-Modus	Aktivierung oder Deaktivierung dieser beiden verfügbaren Funktionen für dieses AS-Interface-Modul (wie beispielsweise die automatische Adressierung) möglich. "Programmblock Halt" ermöglicht Ihnen, den AS-Interface-Bus in den Offline-Modus zu forcieren. Der Modus "Automatische Adressierung" ist standardmäßig aktiviert. Hinweis: Die Funktion "Aktivierung des Datenaustauschs" ist derzeit noch nicht verfügbar.

Das Fenster umfasst außerdem 3 Schaltflächen:

Schaltflächen	Beschreibung
OK	Ermöglicht die Speicherung der im Konfigurationsfenster angezeigten AS-Interface-Bus-Konfiguration. Anschließend Rückkehr zum Hauptmenü. Die Konfiguration kann dann an die Twido-Steuerung übertragen werden.
Abbrechen	Rückkehr zum Hauptmenü ohne Berücksichtigung der vorgenommenen Änderungen.
Hilfe	Aufruf des Hilfemenüs.

Hinweis: Änderungen im Konfigurationsfenster sind nur im Offline-Modus möglich.

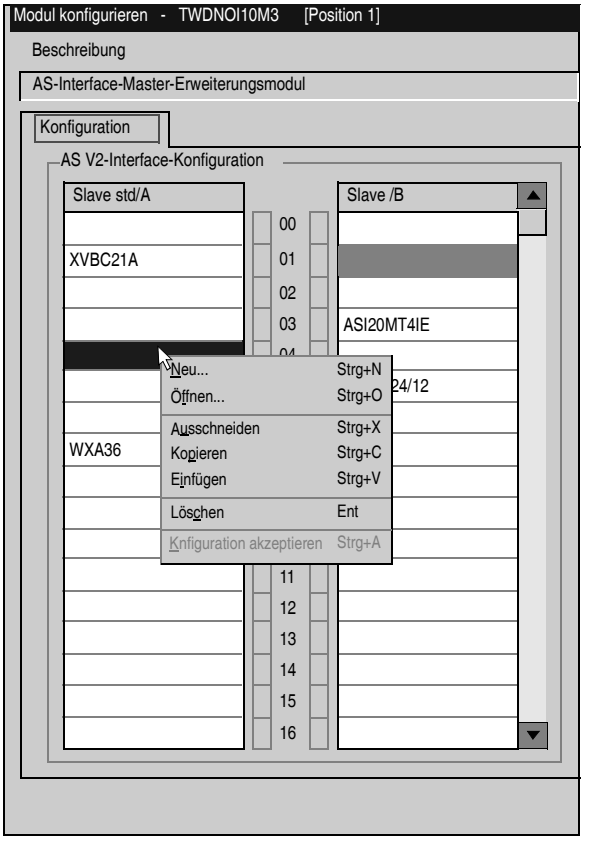
Konfiguration des AS-Interface-Busses

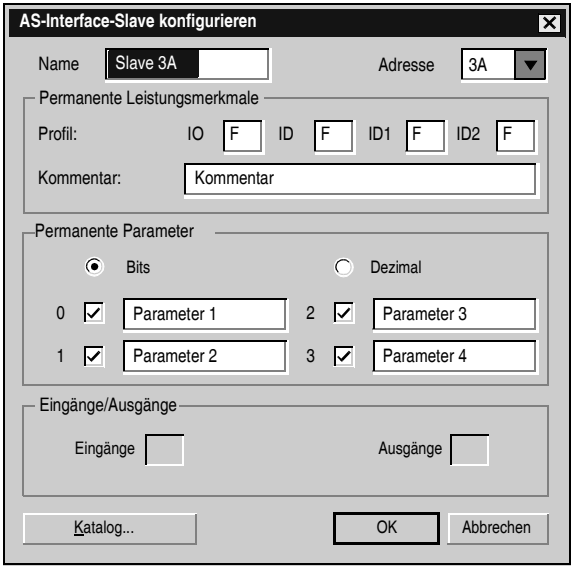
Einleitung

Die Konfiguration des AS-Interface-Busses erfolgt im Konfigurationsfenster im Offline-Modus.
Nachdem der AS-Interface-Master und die Master-Modi ausgewählt sind, werden bei der AS-Interface-Konfiguration die Slave-Geräte konfiguriert.

Verfahren zur Deklaration und Konfiguration eines Slave

Vorgehensweise zum Erstellen oder Ändern eines Slave am AS-Interface V2-Bus:

Schritt	Aktion
1	In der gewünschten Adresszelle (nicht abgeblendet) im Abbild des Busses: ● Doppelklicken Sie: Zugang zu Schritt 3 ODER ● Klicken Sie mit der rechten Maustaste: Ergebnis:  Hinweis: Es wird ein Kontextmenü angezeigt. Dieses ermöglich folgende Aktionen: ● Konfigurieren eines neuen Slave am Bus, ● Ändern der Konfiguration des gewünschten Slave, ● Kopieren (oder Strg+C), Ausschneiden (oder Strg+X) und Einfügen (oder Strg+V) eines Slave, ● Löschen eines Slave (oder Entf).

Schritt	Aktion
2	Wählen Sie im Kontextmenü: ● "Neu", um einen neuen Slave zu erstellen: Es erscheint ein Slave-Konfigurationsfenster. Das Feld "Adresse" zeigt die ausgewählte Adresse an, in den Feldern "Profil" steht standardmäßig ein F und alle anderen Felder im Fenster sind leer. ● "Öffnen" zum Erstellen eines neuen Slave oder zum Ändern der Konfiguration des gewählten Slave. Wenn es sich um einen neuen Slave handelt, erscheint ein neues Slave-Konfigurationsfenster. Das Feld "Adresse" zeigt die ausgewählte Adresse an, in den Feldern "Profil" steht standardmäßig ein F, und alle anderen Felder im Fenster sind leer. Im Falle einer Änderung zeigt das Slave-Konfigurationsfenster die vorher definierten Werte für den ausgewählten Slave an. Abbildung eines Konfigurationsfensters für einen neuen Slave: 
3	Geben Sie im angezeigten Slave-Konfigurationsfenster Folgendes ein oder ändern Sie folgende Angaben: ● Den Namen des neuen Profils (max. 13 Zeichen), ● Einen Kommentar (optional). oder klicken Sie auf die Schaltfläche "Katalog...", und wählen Sie einen vorkonfigurierten Slave der AS-Interface-Profilfamilie aus.

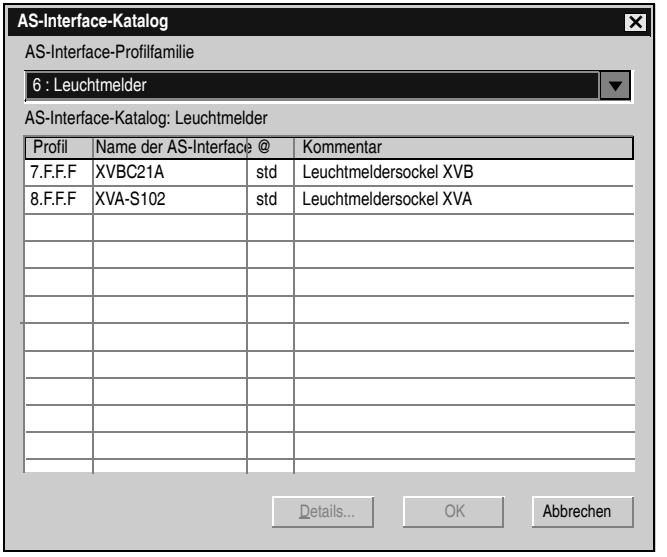
Schritt	Aktion
4	Geben Sie Folgendes ein: ● Den E/A-Code (entspricht der Eingangs-/Ausgangs-Konfiguration), ● Den ID-Code (Kennung) (plus ID1 und ID2 für einen Slave mit erweiterter Adressierung). Hinweis: Die Felder "Eingänge" und "Ausgänge" geben die Anzahl der Eingangs- und Ausgangskanäle an. Sie werden bei der Eingabe des E/A-Codes automatisch implementiert.
5	Legen Sie für jeden Parameter fest: ● Seine Berücksichtigung durch das System (markiertes Kontrollkästchen in der Ansicht "Bits" oder Dezimalwert zwischen 0 und 15 in der Ansicht "Dezimal"), ● Eine aussagekräftigere Bezeichnung als "Parameter X" (optional). Hinweis: Die ausgewählten Parameter sind die Abbildung der permanenten Parameter, die dem AS-Interface-Master zur Verfügung gestellt werden müssen.
6	Wenn Sie den Wert im Feld "Adresse" ändern müssen (im Bereich der am Bus verfügbaren Adressen), klicken Sie auf den nach oben oder nach unten weisenden Pfeil rechts neben der Adresse (d. h. Zugriff auf die gültigen Adressen) oder geben Sie die Adresse direkt über die Tastatur ein.
7	Bestätigen Sie die Slave-Konfiguration, indem Sie auf die Schaltfläche "OK" klicken. Es wird überprüft, ob: ● die E/A- und ID-Codes gültig sind, ● die Slave-Adresse entsprechend des ID-Codes gültig ist (im Falle der Eingabe über die Tastatur) (die Slaves in Bank B sind nur verfügbar, wenn der ID-Code gleich A ist). Im Falle eines Fehlers wird der Fehlertyp durch eine Meldung angezeigt (Beispiel: "Ungültige Adresse für den Slave"), und das Fenster wird erneut mit den ursprünglichen Werten (für das Profil oder die Adresse, je nach Art des Fehlers) angezeigt.

Hinweis: Die Anzahl der Deklarationen für analoge Slaves ist durch die Software auf 7 begrenzt.

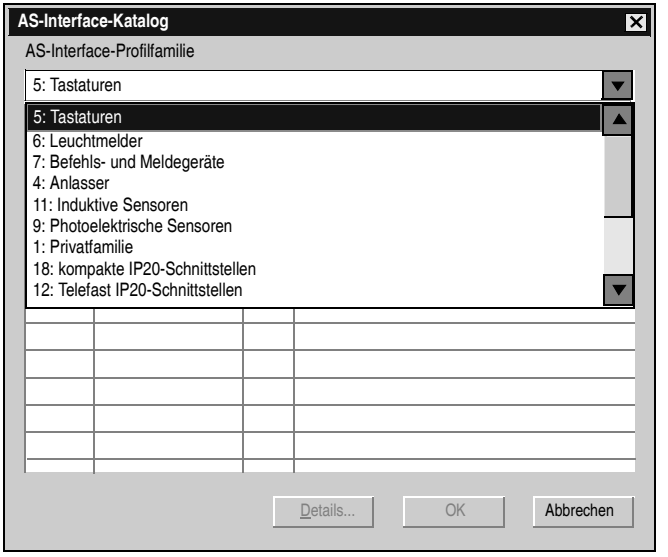
Hinweis: Über die AS-Interface-Produkte des Schneider-Katalogs Wenn Sie auf die Schaltfläche "Katalog" klicken, können Sie Slaves in der "Privaten Familie" erstellen und konfigurieren (andere als die des AS-Interface-Katalogs von Schneider).

AS-Interface-Katalog

Die Schaltfläche "Katalog" dienst der Konfiguration der Slaves auf dem Bus. Wenn Sie einen Slave der Schneider-Familie verwenden, so nutzen Sie diese Schaltfläche für eine sehr einfache und schnelle Konfiguration.
Wenn Sie auf die Schaltfläche "Katalog" des Fensters "AS-Interface-Slave konfigurieren" klicken, wird das folgende Fenster geöffnet:



Im Dropdown-Menü verfügen Sie über Zugriff auf alle Familien des AS-Interface-Katalogs von Schneider:



Wenn Sie Ihre Familie ausgewählt haben, wird die Liste der entsprechenden Slaves angezeigt. Klicken Sie auf den gewünschten Slave und bestätigen Sie Ihre Auswahl, indem Sie auf die Schaltfläche "OK" klicken.

Hinweis: Durch Anklicken der Schaltfläche "Details" können Sie die Eigenschaften eines Slaves anzeigen.

Hinweis: Sie können Slaves, die nicht Bestandteil des Schneider-Katalogs sind, hinzufügen und konfigurieren. Hierzu müssen Sie lediglich die private Familie auswählen und den neuen Slave konfigurieren.

Beschreibung des Debug-Fensters

Auf einen Blick

Wenn der PC mit der Steuerung **verbunden** ist (nach dem Laden der Applikation in die Steuerung), wird die Registerkarte "Debug" rechts neben der Registerkarte "Konfiguration" angezeigt. Sie ermöglicht den Zugriff auf das Debug-Fenster.
Das Debug-Fenster zeigt ein dynamisches Abbild des physikalischen Busses mit folgenden Informationen an:

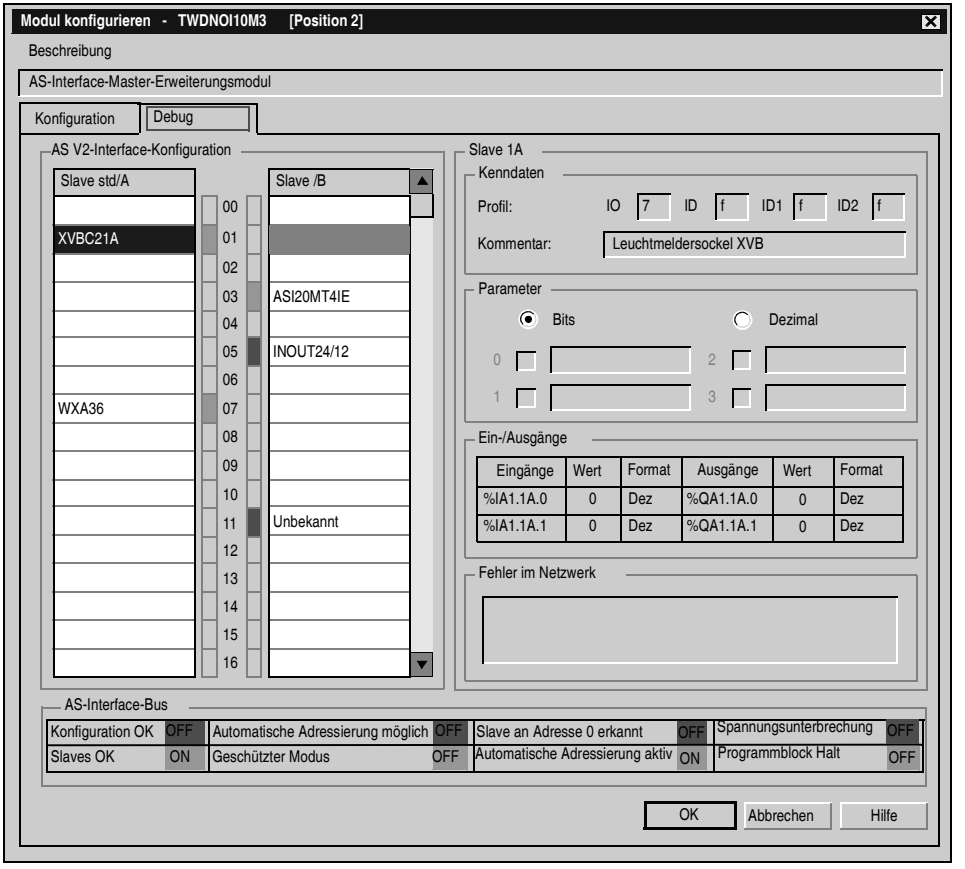
- die Liste der während der Konfiguration geplanten (eingegebenen) Slaves mit deren Namen sowie die Liste der erkannten Slaves (mit unbekanntem Namen, falls nicht geplant)
- der Status des AS-Interface-Kopplers und der Slave-Geräte
- das Abbild des Profils, der Parameter und der Werte der Ein-/Ausgänge der ausgewählten Slaves

Es ermöglicht dem Benutzer außerdem:

- die Durchführung einer Diagnose der fehlerhaften Slaves (Siehe *Anzeige der Slave-Zustände, S. 226*),
- die Änderung der Adresse eines Slaves im Online-Modus (Siehe *Ändern der Adresse eines Slaves, S. 227*),
- die Übertragung des Abbilds der Slaves an das Konfigurationsfenster (Siehe *Aktualisierung der Konfiguration des AS-Interface-Busses im Online-Modus, S. 229*),
- die Adressierung aller Slaves mit den gewünschten Adressen (während des ersten Debuggings).

Abbildung des Fensters "Debug"

Nachfolgend ist das Debug-Fenster (nur im Online-Modus) abgebildet.



Beschreibung des Debug-Fensters

Das Fenster "Debug" liefert dieselben Informationen wie das Konfigurationsfenster (Siehe *Beschreibung des Fensters im Offline-Modus, S. 217*).
Die Unterschiede sind in folgender Tabelle zusammengefasst.

Block	Beschreibung
AS-Interface V2-Konfiguration	Abbild des physikalischen Busses Dieses Abbild umfasst den Status der Slaves: ● Grüne LED: Der Slave an dieser Adresse ist aktiv. ● Rote LED: Der Slave an dieser Adresse ist fehlerhaft, und eine Meldung zeigt den Fehlertyp im Fenster "Fehler im Netzwerk".
Slave xxA/B	Abbild der Konfiguration des ausgewählten Slaves: ● Kenndaten: Abbild des erkannten Profils (abgeblendet, nicht änderbar) ● Parameter: Abbild der erkannten Parameter. Der Benutzer kann nur das Anzeigeformat der Parameter auswählen. ● Ein-/Ausgänge: Die Werte der erkannten Ein-/Ausgänge werden angezeigt, können jedoch nicht verändert werden.
Fehler im Netzwerk	Informiert über den Fehlertyp, wenn der ausgewählte Slave fehlerhaft ist.
AS-Interface-Bus	Aus einem impliziten "Read Status"-Befehl resultierende Informationen. ● Zeigt den Status des Busses an: So zeigt beispielsweise "Konfiguration OK = OFF" an, dass die vom Benutzer vorgesehene Konfiguration nicht der physikalischen Konfiguration des Busses entspricht. ● Zeigt die für das AS-Interface-Mastermodul zulässigen Funktionen an: So zeigt beispielsweise "Automatische Adressierung aktiv = ON" an, dass der Master-Modus Automatische Adressierung zulässig ist.

Anzeige der Slave-Zustände

Wenn die einer Adresse zugewiesene LED rot ist, ist der dieser Adresse zugehörige Slave fehlerhaft. Das Fenster "Fehler im Netzwerk" liefert dann die Diagnose des ausgewählten Slaves.
Fehlerbeschreibung:

- Das vom Benutzer während der Konfiguration an einer bestimmten Adresse vorgesehene Profil entspricht nicht dem realen, an dieser Adresse auf dem Bus erkannten Profil (Diagnose: "Profilfehler").
- Ein neuer Slave, der in der Konfiguration nicht vorgesehen ist, wurde auf dem Bus erkannt: Für diese Adresse wird eine rote LED angezeigt, und der Name des Slaves wird als "Unbekannt" angegeben (Diagnose: "Slave nicht projektiert").
- Peripheriefehler, wenn der erkannte Slave diese Funktion unterstützt (Diagnose: "Peripheriefehler").
- In der Konfiguration ist ein Profil vorgesehen, aber an dieser Adresse auf dem Bus wurde kein Slave erkannt (Diagnose: "Slave nicht erkannt").

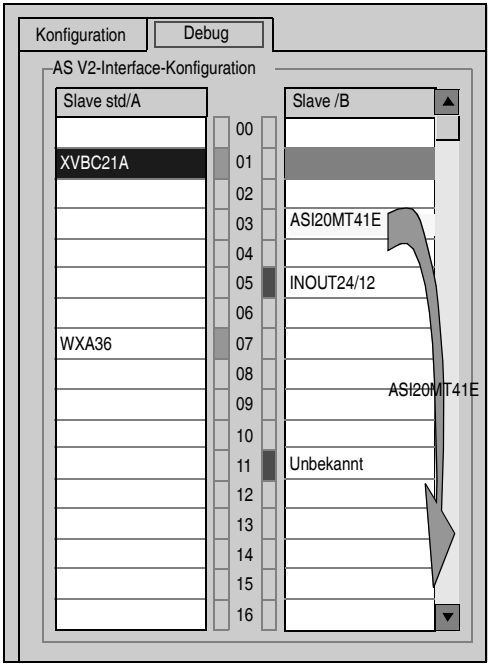
Ändern der Adresse eines Slaves

Auf einen Blick

Der Benutzer kann die Adresse eines Slaves im Online-Modus über das Debug-Fenster ändern.

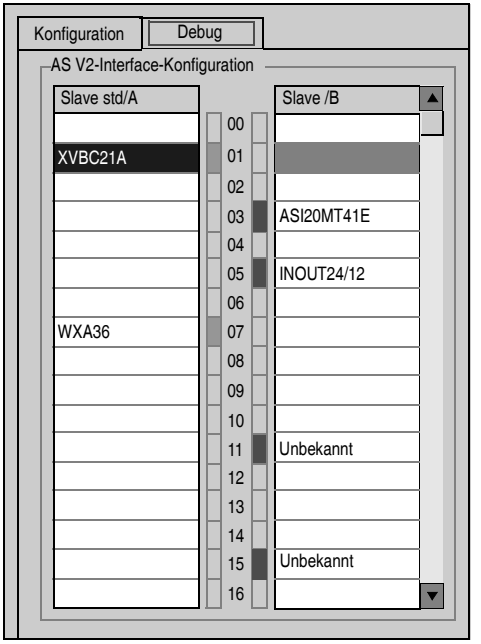
Ändern der Adresse eines Slaves

Die nachfolgende Tabelle beschreibt die Verfahrensweise zum Ändern der Adresse eines Slaves:

Schritt	Beschreibung
1	Öffnen Sie das Debug-Fenster.
2	Wählen Sie einen Slave im Bereich "AS-Interface V2-Konfiguration" aus.
3	Verschieben Sie den Slave mit Hilfe der Maus in die entsprechende Zelle an der gewünschten Adresse. Abbildung: Verschieben und Ablegen des Slaves 3B an die Adresse 15B. 

Schritt	Beschreibung

Ergebnis:
Es wird automatisch eine Prüfung aller Parameter des Slaves ausgeführt, um sicherzustellen, dass der Vorgang möglich ist.
Illustration des Ergebnisses:



Nach dem Vorgang zeigt die Diagnose des Slaves an der Adresse 3B "Slave nicht erkannt" an. Dies bedeutet, dass der für diese Adresse vorgesehene Slave nicht mehr vorhanden ist. Wenn Sie die Adresse 15B auswählen, finden Sie das Profil und die Parameter des verschobenen Slave. Der Name des Slaves bleibt, er wird jedoch nicht erkannt, da er an dieser Adresse nicht vorgesehen war.

Hinweis: Das Profil und die Parameter eines Slaves sind nicht mit seinem Namen verknüpft. Mehrere Slaves mit verschiedenen Namen können dieselben Profile und Parameter haben.

Aktualisierung der Konfiguration des AS-Interface-Busses im Online-Modus

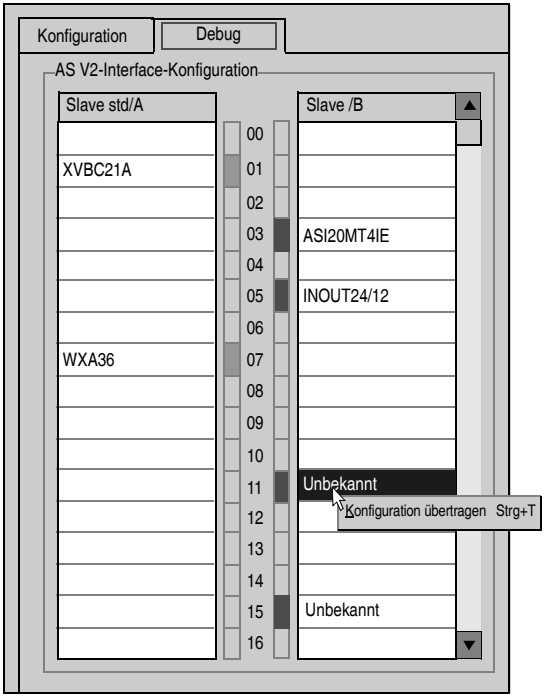
Auf einen Blick

Im Online-Modus ist keine Änderung im Konfigurationsfenster erlaubt; die physische Konfiguration und die Softwarekonfiguration können sich unterscheiden. Jeder Profil- oder Parameterunterschied eines Slave kann - unabhängig davon, ob der Slave in der Konfiguration vorgesehen ist oder nicht - im Konfigurationsfenster berücksichtigt werden. Es ist möglich, jede Änderung an das Konfigurationsfenster zu übertragen, bevor die neue Applikation an die Steuerung gesendet wird.
Für die Berücksichtigung der physischen Konfiguration gilt folgende Vorgehensweise:

Schritt	Bezeichnung
1	Übertragung der gewünschten Slave-Konfiguration an das Konfigurationsfenster.
2	Annahme der Konfiguration im Konfigurationsfenster.
3	Bestätigung der neuen Konfiguration.
4	Übertragung der Applikation in das Modul.

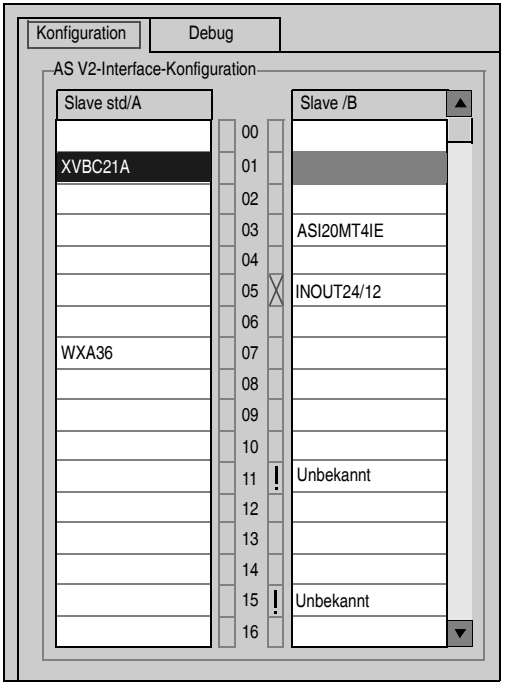
Übertragung der Abbildung eines Slave an die Konfiguration.

Wenn am Bus ein nicht in der Konfiguration vorgesehener Slave erkannt wird, erscheint im Debug-Fenster im Bereich "AS-Interface V2-Konfiguration" ein unbekannter Slave an der erkannten Adresse.
Die folgende Tabelle zeigt die Vorgehensweise zur Übertragung der Abbildung des unbekannten Slave an das Konfigurationsfenster:

Schritt	Bezeichnung
1	Öffnen Sie das Debug-Fenster.
2	Wählen Sie im Bereich "AS-Interface V2-Konfiguration" den gewünschten Slave aus.
3	Klicken Sie mit der rechten Maustaste und wählen Sie "Konfiguration übertragen". Abbildung:  Ergebnis: Die Abbildung des ausgewählten Slave (Abbildung des Profils und Parameter) wird an das Konfigurationsfenster übertragen.
4	Wiederholen Sie den Vorgang für jeden Slave, dessen Abbildung Sie an das Konfigurationsfenster übertragen wollen.

Rückkehr zum Konfigurations-fenster

Wenn Sie das Konfigurationsfenster erneut aufrufen, werden alle neuen (nicht vorgesehenen), übertragenen Slaves angezeigt.
Abbildung des Konfigurationsfensters nach der Übertragung aller Slaves:



Legende:

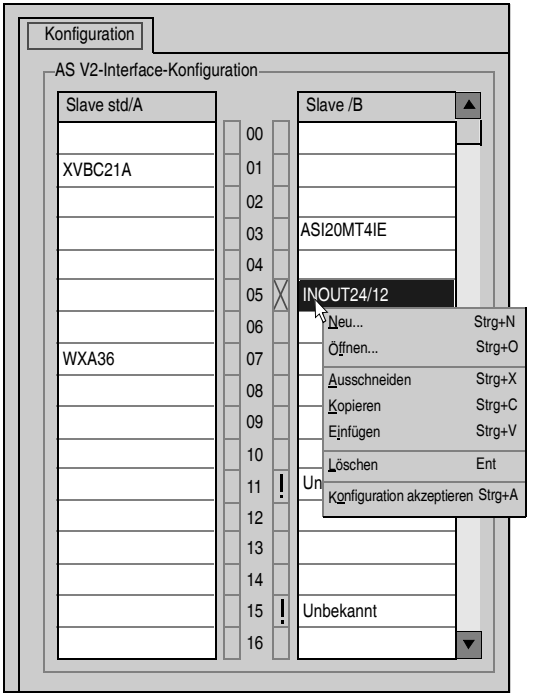
- Das Kreuz gibt an, dass es Unterschiede zwischen der Abbildung des Profils des übertragenen Slaves und dem ursprünglich im Konfigurationsfenster gewählten Profil gibt.
- Das Ausrufezeichen zeigt an, dass ein neues Profil in das Konfigurationsfenster aufgenommen wurde.

Erläuterung:
Das Konfigurationsfenster gibt immer die permanente Abbildung der gewünschten Konfiguration an (deshalb ist der Slave trotz der Adressänderung (Siehe *Ändern der Adresse eines Slaves, S. 227*) in 3B vorhanden), ergänzt durch die aktuelle Abbildung des Busses.
Die angezeigten Profile und Parameter der vorgesehenen Slaves entsprechen den vorgesehenen Profilen und Parametern. Die angezeigten Profile und Parameter der unbekannten Slaves entsprechen den Abbildungen der erkannten Slaves.

Verfahren zur Übertragung der endgültigen Applikation an das Modul

Bevor Sie eine neue Applikation an das Modul senden, können Sie für jeden Slave die erkannte Abbildung des Profils und der Parameter (die in das Konfigurationsfenster übertragen wurde) berücksichtigen oder die Konfiguration "von Hand" ändern (Siehe *Verfahren zur Deklaration und Konfiguration eines Slave, S. 219*). Die folgende Tabelle zeigt die Vorgehensweise zur Bestätigung und zur Übertragung der endgültigen Konfiguration an das Modul:

Schritt	Aktion
1	Trennen Sie den PC softwareseitig vom Modul. Hinweis: Wenn der PC mit dem Modul verbunden ist, sind keine Änderungen im Konfigurationsfenster möglich.
2	Klicken Sie mit der rechten Maustaste auf den gewünschten Slave.

Schritt	Aktion
3	2 Auswahlmöglichkeiten: ● Wählen Sie "Konfiguration akzeptieren", um das **erkannte** Profil des gewählten Slave zu akzeptieren. Abbildung:  Für jeden mit einem Kreuz markierten Slave erscheint eine Meldung, die angibt, dass dieser Vorgang das ursprüngliche (im Fenster angezeigte) Profil des Slave überschreibt. ● Wählen Sie die anderen Optionen des Kontextmenüs, um den gewählten Slave manuell zu konfigurieren.
4	Wiederholen Sie den Vorgang für jeden gewünschten Slave in der Konfiguration.
5	Klicken Sie auf die Schaltfläche "OK", um die Eingaben zu bestätigen und eine neue Applikation zu erstellen. Ergebnis: Automatische Rückkehr zum Hauptfenster.
6	Übertragen Sie die Applikation an das Modul.

Automatische Adressierung eines AS-Interface V2-Slave

Auf einen Blick

Jedem im Bus AS-Interface vorhandenen Slave ist (per Konfiguration) eine nur einmal vorhandene physische Adresse zuzuordnen. Diese muss das Abbild von derjenigen sein, die in TwidoSoft angemeldet wurde.

Die Software TwidoSoft bietet eine automatische Adressierungsfunktion der Slaves und vermeidet so die Verwendung eines AS-Interface-PG.
Die automatische Adressierung wird verwendet für:

- Das Ersetzen eines fehlerhaften Slave,
- Das Einsetzen eines neuen Slave.

Vorgehensweise

Die nachfolgende Tabelle zeigt die Vorgehensweise bei der Definition des Parameters **Autom.**.

Schritt	Aktion
1	Greifen Sie auf das Konfigurationsfenster des AS-Interface V2-Master-Moduls zu.
2	Klicken Sie auf das Kontrollkästchen **Automatische Adressierung**, das sich im Bereich **Master-Modus** befindet. **Ergebnis**: Das Tool **Autom. Adressierung** ist aktiv (Kontrollkästchen aktiviert) oder nicht aktiv (Kontrollkästchen deaktiviert). **Bemerkung**: Der Parameter **Autom.** wird standardmäßig im Konfigurationsfenster ausgewählt.

Einfügen eines Slave-Geräts in eine bestehende AS-Interface V2-Konfiguration

Auf einen Blick

Es ist möglich, ein Gerät in eine vorhandene AS-Interface V2-Konfiguration einzufügen, ohne den Taschenprogrammierer zu verwenden.
Dieser Arbeitsgang ist möglich, wenn:

- der Dienst **Automatische Adressierung** des Konfigurationsmodus aktiv (Siehe *Automatische Adressierung eines AS-Interface V2-Slave, S. 234*) ist,
- in der physischen Konfiguration nur ein Slave fehlt,
- der einzufügende Slave im Konfigurationsfenster vorgesehen ist,
- der Slave das von der Konfiguration erwartete Profil besitzt,
- der Slave die Adresse 0(A) besitzt.

Das AS-Interface V2-Modul weist dem Slave dann automatisch den in der Konfiguration vordefinierten Wert zu.

Vorgehensweise

Die nachfolgende Tabelle beschreibt die Vorgehensweise, die anzuwenden ist, damit die automatische Einfügung eines neuen Slave wirksam wird.

Schritt	Aktion
1	Fügen Sie den neuen Slave im Offline-Modus im Konfigurationsfenster hinzu.
2	Führen Sie im Online-Modus einen Konfigurationstransfer zur Steuerung durch.
3	Schließen Sie den neuen Slave mit der Adresse 0(A) physisch an den AS-Interface V2-Bus an.

Hinweis: Es ist möglich, eine Applikation zu ändern, indem die oben angegebene Bearbeitung so oft wie nötig durchgeführt wird.

Automatischer Ersatz eines fehlerhaften AS-Interface V2-Slave

Prinzip

Wenn ein Slave als fehlerhaft gemeldet wird, kann er automatisch durch einen Slave desselben Typs ersetzt werden.
Der Austausch erfolgt, ohne dass der AS-Interface V2-Bus gestoppt werden muss und ohne dass eine besondere Manipulation erforderlich ist, wenn das Tool **Automatische Adressierung** des Konfigurationsmodus aktiv ist (Siehe *Automatische Adressierung eines AS-Interface V2-Slave, S. 234*).

Es gibt zwei Möglichkeiten:

- Der als Ersatz vorgesehene Slave wird mit dem Taschenprogrammierer mit derselben Adresse programmiert und hat dasselbe Profil und Unterprofil wie der fehlerhafte Slave. Er wird somit automatisch in die Liste der erkannten Slaves (LDS) und in die Liste der aktiven Slaves (LAS) eingefügt.
- Der als Ersatz vorgesehene Slave ist noch nicht programmiert (Adresse 0, neuer Slave) und hat dasselbe Profil wie der fehlerhafte Slave. Er nimmt automatisch die Adresse des ersetzten Slave an und wird somit in die Liste mit den erkannten Slaves (LDS) und in die Liste mit den aktiven Slaves (LAS) eingefügt.

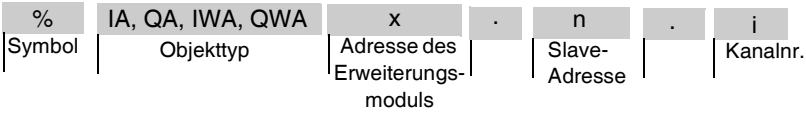
Adressierung der Ein-/Ausgänge, die den am AS-Interface V2-Bus angeschlossenen Slave-Geräten zugeordnet sind.

Auf einen Blick

Diese Seite beschreibt die Besonderheiten der Adressierung der digitalen oder analogen Ein-/Ausgänge der Slave-Geräte.
Um jegliche Verwechselung mit den dezentralen E/A zu vermeiden, wurden neue Symbole mit einer AS-Interface-Syntax eingeführt: %I**A** zum Beispiel.

Abbildung

Kurze Zusammenfassung des Adressierungsprinzips:



Spezifische Werte

Die nachfolgende Tabelle enthält die spezifischen Werte für die AS-Interface V2-Slaveobjekte.

Element	Werte	Kommentar
IA	-	Abbild des physikalischen Digitaleingangs des Slaves
QA	-	Abbild des physikalischen Digitalausgangs des Slaves
IWA	-	Abbild des physikalischen Analogeingangs des Slaves
QWA	-	Abbild des physikalischen Analogausgangs des Slaves
x	1 à 7	Adresse des AS-Interface-Moduls auf dem Erweiterungsbus
n	0A bis 31B	Die Position 0 ist nicht konfigurierbar.
i	0 bis 3	-

Beispiele

Die nachfolgende Tabelle beschreibt einige Beispiele für die Adressierung der E/A.

E/A-Objekt	Beschreibung
%IWA4.1A.0	Analogeingang 0 des Slaves 1A des AS-Interface-Moduls an Position 4 des Erweiterungsbusses
%QA2.5B.1	Digitalausgang 1 des Slaves 5B des AS-Interface-Moduls an Position 2 des Erweiterungsbusses
%IA1.12A.2	Digitaleingang 2 des Slaves 12A des AS-Interface-Moduls an Position 1 des Erweiterungsbusses

Impliziter Austausch

Die nachfolgend beschriebenen Objekte werden implizit ausgetauscht, d.h., sie werden automatisch bei jedem Steuerungszyklus ausgetauscht.

Programmierung und Diagnose des AS-Interface V2-Busses

Expliziter Austausch

Mit dem AS-Interface-Bus verknüpfte Objekte (Wörter und Bits) bieten Informationen (Bsp.: Funktion des Busses, Status der Slaves...) und zusätzliche Befehle für eine fortgeschrittene Programmierung der AS-Interface-Funktion. Diese Objekte werden explizit über den Erweiterungsbus zwischen der Twido-Steuerung und dem AS-Interface-Master ausgetauscht:

- nach Aufforderung durch das Anwenderprogramm mit Hilfe folgender Anweisung: ASI_CMD (siehe "Beschreibung der Anweisung "ASI_CMD" weiter unten)
- über das Debugging-Fenster oder die Animationstabelle

Spezifische reservierte Systemwörter

Die in der Twido-Steuerung für die AS-Interface-Mastermodule reservierten Systemwörter geben den Status des Netzwerks wieder: %SW73 ist für das erste AS-Interface-Erweiterungsmodul reserviert, %SW74 für das zweite. Nur die 5 ersten Bits dieses Wortes werden verwendet. Sie sind schreibgeschützt.
Die folgende Tabelle beschreibt die verwendeten Bits:

Systemwörter	Bit	Beschreibung
%SW73 und %SW74	0	Systemstatus (= 1 wenn Konfiguration OK, ansonsten 0)
	1	Datenaustausch (= 1 wenn Datenaustausch aktiv, 0 wenn im Modus Data Exchange Off (Siehe *Betriebsart des AS-Interface V2-Bus-Schnittstellenmoduls, S. 243*))
	2	System im Status STOP (= 1 wenn der Modus Offline (Siehe *Offline-Modus, S. 243*) aktiv ist, ansonsten 0)
	3	Anweisung ASI_CMD beendet (= 1 wenn beendet, 0 wenn nicht beendet)
	4	Fehler Anweisung ASI_CMD (= 1 bei einem Fehler in der Anweisung, ansonsten 0)

Anzeigebeispiel (für das erste AS-Interface-Erweiterungsmodul):
Vor der Nutzung einer ASI_CMD-Anweisung muss das Bit %SW73:X3 überprüft werden, um zu gewährleisten, dass keine Anweisung ausgeführt wird: Überprüfen, ob %SW73:X3 = 1.
Um zu ermitteln, ob die Anweisung anschließend fehlerfrei ausgeführt worden ist, muss überprüft werden, ob das Bit %SW73:X4 auf 0 gesetzt ist.

Beschreibung der Anweisung ASI_CMD

Über das Benutzerprogramm ermöglicht die Anweisung ASI_CMD dem Benutzer die Programmierung seines Netzwerks und den Erhalt einer Diagnose der Slaves. Die Parameter der Anweisung werden über interne Speicherwörter **%MW**x weitergeleitet.
Die Syntax der Anweisung lautet wie folgt:
ASI_CMDn **%MW**x**:**l
Legende:

Symbol	Beschreibung
n	Adresse des AS-Interface-Erweiterungsmoduls (1 bis 7)
x	Nummer des ersten, im Parameter (0 bis 254) weitergeleiteten internen Speicherwortes
l	Länge der Anweisung in Wörtern (2).

Verwendung der Anweisung ASI_CMD

Die folgende Tabelle beschreibt die Aktion der Anweisung ASI_CMD abhängig vom Wert der Parameter %MW(x) und, falls erforderlich, %MW(x+1). Bei Requests zur Diagnose der Slaves werden die Ergebnisse in %MW(x+1) ausgegeben.

%MWx	%MWx+1	Aktion
1	0	beendet den Offline-Modus
1	1	geht in den Offline-Modus über
2	0	unterbindet den Datenaustausch zwischen dem Master und seinen Slaves (geht in den Modus "Data Exchange Off" über)
2	1	gibt den Datenaustausch zwischen dem Master und seinen Slaves frei (verlässt den Modus "Data Exchange Off")
3	reserviert	-
4	Ergebnis	liest die Liste der aktiven Slaves (LAS-Tabelle) der Adressen 0A bis 15A (1 Bit pro Slave)
5	Ergebnis	liest die Liste der aktiven Slaves (LAS-Tabelle) der Adressen 16A bis 31A (1 Bit pro Slave)
6	Ergebnis	liest die Liste der aktiven Slaves (LAS-Tabelle) der Adressen 0B bis 15B (1 Bit pro Slave)
7	Ergebnis	liest die Liste der aktiven Slaves (LAS-Tabelle) der Adressen 16B bis 31B (1 Bit pro Slave)
8	Ergebnis	liest die Liste der erkannten Slaves (LDS-Tabelle) der Adressen 0A bis 15A (1 Bit pro Slave)
9	Ergebnis	liest die Liste der erkannten Slaves (LDS-Tabelle) der Adressen 16A bis 31A (1 Bit pro Slave)
10	Ergebnis	liest die Liste der erkannten Slaves (LDS-Tabelle) der Adressen 0B bis 15B (1 Bit pro Slave)
11	Ergebnis	liest die Liste der erkannten Slaves (LDS-Tabelle) der Adressen 16B bis 31B (1 Bit pro Slave)
12	Ergebnis	liest die Liste der Slave-Peripheriefehler (LPF-Tabelle) der Adressen 0A bis 15A (1 Bit pro Slave)
13	Ergebnis	liestdie Liste der Slave-Peripheriefehler (LPF-Tabelle) der Adressen 16A bis 31A (1 Bit pro Slave)
14	Ergebnis	liest die Liste der Slave-Peripheriefehler (LPF-Tabelle) der Adressen 0B bis 15B (1 Bit pro Slave)
15	Ergebnis	liest die Liste der Slave-Peripheriefehler (LPF-Tabelle) der Adressen 16B bis 31B (1 Bit pro Slave)
16	Ergebnis	liest den Busstatus Siehe Beschreibung des Ergebnisses im folgenden Abschnitt.

Hinweis: Der Busstatus wird bei jedem Steuerungszyklus aktualisiert. Das Ergebnis der Bus-Leseanweisung ASI_CMD ist jedoch erst im nächsten Steuerungszyklus verfügbar.

Beschreibung der Anweisung ASI_CMD zum Lesen des Busstatus

Beim Lesen des Busstatus mittels der Anweisung ASI_CMD (Wert des Parameters %MWx gleich 16) lautet das Ergebnis im Wort %MWx+1 wie folgt:

%MWx+1		**Bezeichnung (1=OK, 0=NOK)**
niederwertig	Bit 0	Konfiguration OK
	Bit 1	LDS.0 (Slave an Adresse 0 vorhanden)
	Bit 2	Automatische Adressierung aktiv
	Bit 3	Automatische Adressierung verfügbar
	Bit 4	Konfigurationsmodus aktiv
	Bit 5	Normaler Betrieb aktiv
	Bit 6	APF (Versorgungsfehler)
	Bit 7	Offline bereit
höherwertig	Bit 0	Peripheriefehler
	Bit 1	Datenaustausch aktiv
	Bit 2	Offline-Modus
	Bit 3	Normaler Modus (1)
	Bit 4	Fehler während der Kommunikation mit dem AS-Interface-Master
	Bit 5	Anweisung ASI_CMD läuft
	Bit 6	Anweisung ASI_CMD fehlerhaft

Beschreibung der Anweisung ASI_CMD zum Lesen des Slave-Status

Bei einer Slave-Diagnose mittels der Anweisung ASI_CMD (Wert %MWx zwischen 4 und 15) wird der Status der Slaves in den Bits (1=OK) des Wortes %MWx+1 ausgegeben. Die folgende Tabelle beschreibt das Ergebnis abhängig vom Wert des Wortes %MWx:

%MWx	**%MWx+1**															
Wert	Höchstwertiges Byte								Niederwertigstes Byte							
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
4, 8, 12	15A	14A	13A	12A	11A	10A	9A	8A	7A	6A	5A	4A	3A	2A	1A	0A
5, 9, 13	31A	30A	29A	28A	27A	26A	25A	24A	23A	22A	21A	20A	19A	18A	17A	16A
6, 10, 14	15B	14B	13B	12B	11B	10B	9B	8B	7B	6B	5B	4B	3B	2B	1B	0B
7, 11, 15	31B	30B	29B	28B	27B	26B	25B	24B	23B	22B	21B	20B	19B	18B	17B	16B

Um zu lesen, ob der Slave 20B aktiv ist, muss die Anweisung ASI_CMD mittels des internen Wortes %MWx mit dem Wert 7 ausgeführt werden. Das Ergebnis wird im internen Wort %MWx+1 zurückgegeben. Der Status des Slaves 20B wird durch den Wert von Bit 4 des niederwertigen Bytes angegeben: Wenn Bit 4 gleich 1 ist, ist der Slave 20B aktiv.

Programmier-beispiel der Anweisung ASI_CMD

Für die Forcierung des Übergangs des AS-Interface-Masters (an Position 1 des Erweiterungsbusses) in den Offline-Modus:

```
LD 1
[%MW0 := 16#0001]
[%MW1 := 16#0001]
LD %SW73:X3          //Wenn keine ASI_CMD läuft, wird folgendermaßen fortgesetzt
[ASI_CMD1 %MW0:2]     //zur Forcierung des Übergangs in den Offline-Modus
```

Zum Lesen der Tabelle der aktiven Slaves der Adressen 0A bis 15A

```
LD 1
[%MW0 := 16#0004]
[%MW1 := 16#0000          //optional]
LD %SW73:X3   //Wenn keine ASI_CMD läuft, wird folgendermaßen fortgesetzt
[ASI_CMD1 %MW0:2]     //Zum Lesen der LAS-Tabelle der Adressen 0A bis 15A
```

Betriebsart des AS-Interface V2-Bus-Schnittstellenmoduls

Auf einen Blick

Das AS-Interface V2-Bus-Schnittstellenmodul TWDNOI10M3 ermöglicht drei Betriebsarten, die jeweils besonderen Anforderungen entsprechen. Diese Modi sind:

- der geschützte Modus,
- der Offline-Modus,
- der Data-Echange-Off-Modus.

Die Verwendung der Anweisung ASI_CMD (Siehe *Beschreibung der Anweisung ASI_CMD, S. 239*) in einem Benutzerprogramm ermöglicht den Aufruf bzw. das Verlassen dieser Modi.

Geschützter Modus

Der geschützte Betriebmodus ist der Modus, der im allgemeinen für eine in Ausführung begriffene Applikation benutzt wird. Dazu muss das AS-Interface V2-Modul in TwidoSoft konfiguriert sein. Es:

- überprüft ständig, ob die Liste mit den erkannten Slaves mit der Liste mit den vorgesehenen Slaves übereinstimmt.
- überwacht die Versorgung.

In diesem Modus wird ein Slave nur aktiviert, wenn er in der Konfiguration angemeldet und erkannt wurde.

Beim Einschalten oder während der Konfiguration forciert die Twido-Steuerung das AS-Interface-Modul in den geschützten Modus.

Offline-Modus

Beim Wechsel in den Offline-Modus setzt das Modul zuerst alle vorhandenen Slaves auf Null zurück und stoppt den Austausch mit dem Bus. Im Offline-Modus werden die Ausgänge auf Null forciert.

Neben der Nutzung der Schaltfläche PB2 am AS-Interface-Modul TWDNOI10M3 kann der Offline-Modus auch per Software mittels der Anweisung ASI_CMD (Siehe *Programmierbeispiel der Anweisung ASI_CMD, S. 242*) aufgerufen werden. Mittels derselben Anweisung kann der Modus auch beendet werden, um zum geschützten Modus zurückzukehren.

Data-Echange-Off-Modus

Beim Wechsel in den Modus Data Exchange Off wird der Austausch innerhalb des Busses fortgesetzt, aber die Daten werden nicht mehr aktualisiert.

Dieser Modus kann nur mittels der Anweisung ASI_CMD (Siehe *Verwendung der Anweisung ASI_CMD, S. 240*) aufgerufen werden.

Installieren und Konfigurieren des CANopen-Feldbusses

10

Auf einen Blick

Inhalt dieses Kapitels

Dieses Kapitel beschreibt die Installation und Konfiguration des CANopen-Mastermoduls TWDNCO1M und seiner Slaves auf dem CANopen-Feldbus.

Inhalt dieses Kapitels

Dieses Kapitel enthält die folgenden Abschnitte:

Abschnitt	Thema	Seite
10.1	CANopen-Feldbus - Übersicht	247
10.2	Implementieren des CANopen-Busses	261

10.1 CANopen-Feldbus - Übersicht

Auf einen Blick

Inhalt dieses Abschnitts

Dieser Abschnitt enthält allgemeine Informationen über die CANopen-Feldbustechnologie und eine Einführung in die CAN-spezifische Terminologie, die im weiteren Verlauf dieses Kapitels verwendet wird.

Inhalt dieses Abschnitts

Dieser Abschnitt enthält die folgenden Themen:

Thema	Seite
CANopen - Grundlagenwissen	248
Über CANopen	249
CANOpen Boot-Up	252
Process Data Object (PDO) -Übertragung	255
Datenzugriff per explizitem Austausch (SDO)	257
"Node Guarding" und "Life Guarding"	258
Verwaltung des internen Busses	260

CANopen - Grundlagenwissen

Einleitung

Die folgenden Erklärungen der technischen Begriffe und Abkürzungen sind nützlich für das Verständnis der Grundlagen der CANopen-Netzwerkkommunikation.

EDS-Datei

EDS (Electronic Data Sheet, dt.: elektronisches Datenblatt)
Eine EDS-Datei beschreibt die Kommunikationseigenschaften eines Geräts im CAN-Netzwerk (Baudraten, Übertragungsarten, E/A-Angebot etc.). Sie wird vom Gerätehersteller geliefert. Sie wird im Configuration Tool verwendet, um einen Knoten (wie etwa einen Treiber in einem Betriebssystem) zu konfigurieren.

PDO

PDO (Process Data Object, dt.: Prozessdatenobjekt)
CANopen-Frame, der E/A-Daten enthält.
Wie unterscheiden zwischen:

- Transmit-PDOs (TPDOs mit Daten, die von einem Knoten stammen) und
- Receive PDOs (RPDOs mit Daten, die von einem Knoten verarbeitet werden).

Die Übertragungsrichtung wird immer aus der Sicht eines Knotens angegeben. Ein PDO enthält nicht unbedingt das gesamte Datenabbild eines Knotens (gilt für TPDO und RPDO). Normalerweise werden analoge Eingangsdaten und digitale Eingangsdaten auf verschiedene TPDOs aufgeteilt. Das gleiche gilt für die Ausgangsdaten.

SDO

SDO (Service Data Object, dt.: Dienstdatenobjekt)
CANopen-Frames, die Parameter enthalten.
SDOs werden üblicherweise verwendet, um Parameter von Geräten zu lesen bzw. um Parameter in Geräte zu schreiben, während die Applikation ausgeführt wird.

COB-ID

COB-ID (Communication Object Identifier, dt.: Kommunikationsobjektkennung)
Jeder CANopen-Frame beginnt mit einer COB-ID, die als die Kennung im CAN-Frame fungiert. Während der Konfigurationsphase empfängt jeder Knoten die COB-ID(s) für den/die Frame(s), für den bzw. die er der Producer oder Consumer ist.

Über CANopen

Einleitung

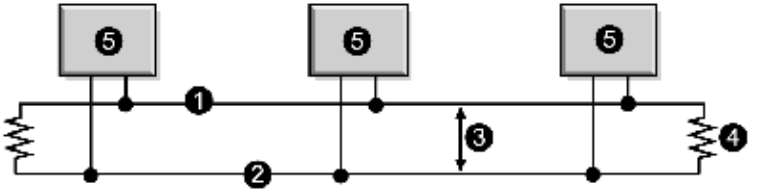
CANopen ist ein Standard-Feldbusprotokoll für industrielle Steuerungssysteme. Es ist besonders gut geeignet für Echtzeit-SPS, da es eine effektive, kostengünstige Lösung für integrierte und transportable industrielle Anwendungen bietet.

Das CANopen-Protokoll

Das CANopen-Protokoll wurde als ein Teil von CAL entwickelt. Durch die Definition von Profilen kann es noch spezieller an die Nutzung mit standardmäßigen industriellen Komponenten angepasst werden. CANopen ist ein CiA- (CAN in Automation-) Standard, der sehr schnell nach seiner Verfügbarkeit auf dem Markt angenommen wurde. In Europa ist CANopen jetzt als der Industriestandard für industrielle Systeme anerkannt, die auf einem CAN-Design basieren.

Physikalische Schicht

CAN verwendet eine differential gesteuerte zweiadrige Busleitung (gemeinsame Rückleitung). Ein CAN-Signal ist die Differenz zwischen den Spannungspegeln der CAN-high- und CAN-low-Drähte (siehe Abbildung unten).
Die folgende Abbildung zeigt die Komponenten der physikalischen Schicht eines zweiadrigen CAN-Busses.



1 CAN-high-Draht

2 CAN-low-Draht

3 Potenzialunterschied zwischen den CAN-high/CAN-low-Signalen

4 120Ω Widerstandsbuchse

5 Knoten

Die Busdrähte können parallel, verdrillt oder abgeschirmt in Übereinstimmung mit den Anforderungen an die elektromagnetische Verträglichkeit verlegt werden. Eine Einzelleitungsstruktur minimiert die Rückstrahlung.

CANopen-Profile

Das Kommunikationsprofil

Die CANopen-Profilfamilie basiert auf einem "Kommunikationsprofil", das die Hauptkommunikationsmechanismen und deren Beschreibung festlegt (DS301).

Das Geräteprofil

Die wichtigsten in der Industrieautomation verwendeten Gerätetypen werden in den "Geräteprofilen" beschrieben. Sie definieren auch die Funktionsweise der Geräte. Beispiele für die beschriebenen Standardgeräte sind:

- digitale und analoge Ein-/Ausgangsmodule (DS401),
- Motoren (DS402),
- Steuerungsgeräte (DSP403),
- Steuerungen mit geschlossenem Regelkreis (DSP404),
- SPS (DS405),
- Geber (DS406).

Gerätekonfiguration über den CAN-Bus

Die Möglichkeit der Gerätekonfiguration über den CAN-Bus ist eines der Grundprinzipien der von den Herstellern geforderten Autonomie (für jede Profilfamilie).

Allgemeine Spezifikationen für CANopen-Profile

CANopen ist eine Reihe von Profilen für CAN-Systeme mit den folgenden Merkmalen:

- Offenes Bussystem
- Echtzeit-Datenaustausch ohne Überlastung des Protokolls
- Modulares Design mit der Möglichkeit zur Erweiterung/Verkleinerung
- Interoperabilität und Austauschbarkeit von Geräten
- Unterstützung durch eine Vielzahl internationaler Hersteller
- Standardisierte Netzwerkkonfiguration
- Zugriff auf alle Geräteparameter
- Synchronisation und Zirkulation zyklischer Prozessdaten und/oder ereignisgesteuerter Daten (Möglichkeit kurzer Systemantwortzeiten)

CANopen-Produktzertifizierung

Alle Hersteller, die CANopen-zertifizierte Produkte auf dem Markt anbieten, sind Mitglieder der CiA-Gruppe. Als ein aktives Mitglied der CiA-Gruppe entwickelt Schneider Electric Industries SAS seine Produkte in Übereinstimmung mit den standardisierten Empfehlungen dieses Verbands.

CAN-Standards

Die CANopen-Spezifikationen werden von der CiA-Gruppe definiert und können (mit einigen Einschränkungen) auf der Internetseite der Gruppe unter http://www.can-cia.com eingesehen werden. Die Quellcodes für Master- und Slave-Geräte sind bei verschiedenen Anbietern erhältlich.

Hinweis: Ausführliche Informationen über die CANopen-Standardspezifikationen und -mechanismen finden Sie auf der CiA-Homepage (http://www.can-cia.de/).

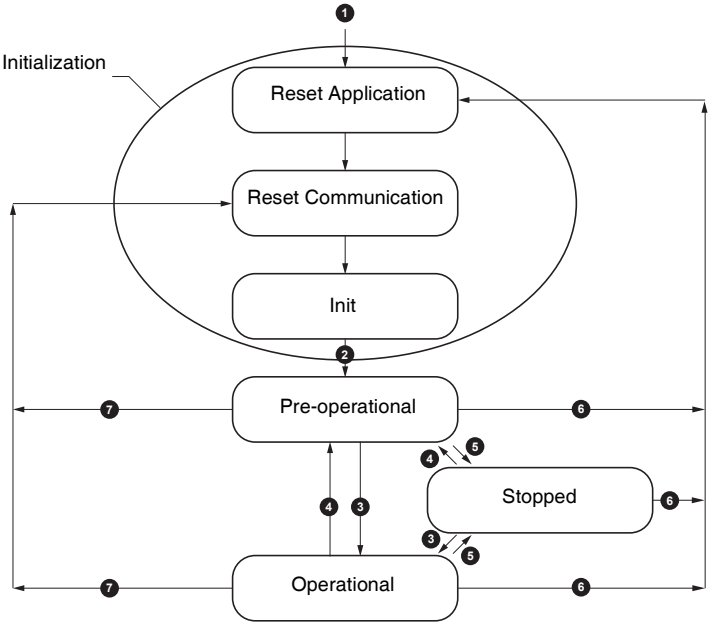
Kommunikation in einem CANopen-Netzwerk

Das Kommunikationsprofil basiert auf CAL-Diensten und -Protokollen.
Es bietet dem Benutzer einen Zugriff auf zwei Austauscharten: SDO und PDO.
Beim Einschalten geht das Gerät in eine Initialisierungsphase und anschließend in einen Anlaufstatus über. In dieser Phase ist nur die SDO-Kommunikation zulässig. Nach dem Empfang eines Anlaufbefehls schaltet das Gerät in den Betriebsstatus um. Dann kann der PDO-Austausch genutzt werden. Die SDO-Kommunikation ist weiterhin möglich.

CANOpen Boot-Up

Boot-up-Verfahren

Die minimale Gerätekonfiguration legt ein abgekürztes Boot-Verfahren fest. Dieses Verfahren ist in der folgenden Abbildung dargestellt.



Legende

Nummer	Beschreibung
1	Module power up
2	Nach der Initialisierung geht das Modul automatisch in den Zustand PRE-OPERATIONAL über.
3	NMT-Dienstanzeige: START REMOTE NODE
4	NMT-Dienstanzeige: PRE-OPERATIONAL
5	NMT-Dienstanzeige: STOP REMOTE NODE
6	NMT-Dienstanzeige: RESET NODE
7	NMT-Dienstanzeige: RESET COMMUNICATION

Aktive CANopen-Objekte abhängig von der Ablaufsteuer-einheit

Die Kreuze in der nachfolgenden Tabelle geben an, welche CANopen-Objekte bei welchen Zuständen der Ablaufsteuereinheit aktiv sind.

	Initialization	Pre-operational	Operational	Stopped
PDO-Objekt			X	
SDO-Objekt		X	X	
Emergency		X	X	
Boot-Up	X		X	
NMT		X	X	X

Reset Application

Das Gerät geht in den Zustand "Reset Application" über:
- nach dem Einschalten des Geräts,
- oder bei Verwendung des "Reset Node"-Netzwerkmanagement- (NMT) Diensts.

In diesem Zustand ist das Geräteprofil initialisiert, und alle Geräteprofildaten sind auf die Standardwerte zurückgesetzt. Wenn die Initialisierung abgeschlossen ist, geht das Gerät automatisch in den Zustand "Reset Communication" über.

Reset Communication

Das Gerät geht in den Zustand "Reset Communication" über:
- nach dem Zustand "Reset Application",
- oder bei Verwendung des "Reset Communication"-Netzwerkmanagement- (NMT) Diensts.

In diesem Zustand werden alle Parameter (Standardwert, abhängig von der Gerätekonfiguration) der unterstützten Kommunikationsobjekte (Objekte wie etwa der Gerätetyp, der heartbeat etc., die zur Geräteidentifikation gehören: 1000H - 1FFFH) im Objektverzeichnis gespeichert. Das Gerät geht dann automatisch in den Zustand "Init" über.

Init

Das Gerät geht in den Modus "Init" über, nachdem es im Zustand "Reset Communication" war.
Dieser Zustand ermöglicht Ihnen:
- die Definition der erforderlichen Kommunikationsobjekte (SDO, PDO, Emergency),
- die Installation der entsprechenden CAL-Dienste,
- die Konfiguration des CAN-Controller.

Die Initialisierung des Geräts ist abgeschlossen und das Gerät geht automatisch in den Zustand"Pre-Operational" über.

Hinweis: Das CANopen-Mastermodul TWDNCO1M unterstützt nicht den Modus SYNC.

Pre-Operational

Das Gerät geht in den Zustand "Pre-Operational" über:
- nach dem Zustand "Init",
- bei Empfang der NMT-Anzeige "Enter Pre-Operational", wenn es sich im Zustand "Operational" befunden hat.

Wenn sich das Gerät in diesem Zustand befindet, kann seine Konfiguration geändert werden. Es können jedoch nur SDOs verwendet werden, um gerätespezifische Daten zu lesen oder zu schreiben.
Wenn die Konfiguration abgeschlossen ist, geht das Gerät bei Empfang der entsprechenden Anzeige in einen der folgenden Zuständen über:
- "Stopped" bei Empfang der NMT-Anzeige "STOP REMOTE NODE",
- "Operational" bei Empfang der NMT-Anzeige "START REMOTE NODE".

Stopped

Das Gerät geht in den Zustand "Stopped" bei Empfang der Anzeige "Node stop" (NMT-Dienst) über, wenn es sich im Zustand "Pre-Operational" oder "Operational" befunden hat.
In diesem Zustand kann das Gerät nicht konfiguriert werden. Es ist kein Dienst zum lesen oder schreiben von gerätespezifischen Daten (SDO) verfügbar. Nur die Slave-Überwachungsfunktion ("Node guarding") bleibt aktiv.

Operational

Das Gerät geht in den Zustand "Operational" über, wenn es sich im Status "Pre-Operational" bei Empfang der Anzeige "Start Remote Node" befunden hat.
Wenn das CANopen-Netzwerk mittels des NMT-Dienstes "Node start" im Zustand "Operational" gestartet wird, können alle Gerätefunktionen verwendet werden. Die Kommunikation kann PDOs oder SDOs verwenden.

Hinweis: Änderungen an der Konfiguration im Modus "Operational" können unerwartete Auswirkungen haben und sollten daher im Modus "Pre-Operational" vorgenommen werden.

Process Data Object (PDO) -Übertragung

Definition eines PDO

PDOs sind Objekte, die der Kommunikationsschnittstelle Prozessdaten liefern und den Austausch dieser Prozessdaten in Echtzeit ermöglichen. Das PDO-Set eines CANopen-Geräts beschreibt den impliziten Austausch zwischen diesem Gerät und einen Kommunikationspartnern im Netzwerk.
Der Austausch von PDOs ist zulässig, wenn sich das Gerät im Modus "Operational" befindet.

PDO-Typen

Es gibt zwei Arten von PDOs:

- vom Gerät gesendete PDOs (häufig als Transmit-PDO oder Tx-PDO oder TPDO bezeichnet),
- vom Gerät empfangene PDOs (häufig alsReceive-PDO oder Rx-PDO oder RPDO bezeichnet).

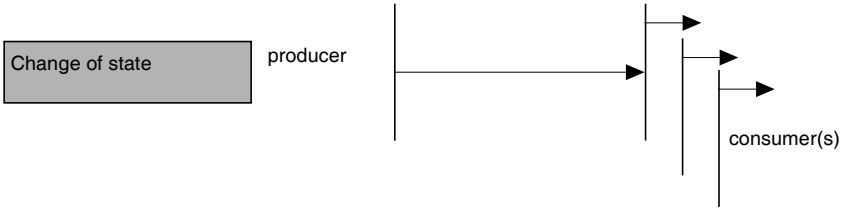
PDO Producers and Consumers

PDOs basieren auf dem "Producer / Consumer"-Modell. Das Gerät, das einen PDO sendet, wird als "producer" bezeichnet, während das empfangende Gerät als "consumer" bezeichnet wird.
Das Schreiben eines Ausgangs an das Mastermodul TWDNCO1M führt also zum Senden eines mit dem Master verbundenen TPDO, das den Wert des zu aktualisierenden Ausgangs enthält. In diesem Fall ist der Master der PDO "producer" (während der Slave der PDO "consumer" ist.
Dagegen wird ein Eingang durch das Senden eines RPDO vom Mastermodul aktualisiert. In diesem Fall ist das Mastermodul der "consumer".

PDO-Übertragungsmodus

Neben den zu übertragenden Daten kann die Art des Austauschs für jedes PDO konfiguriert werden.
PDOs können vom Mastermodul TWDNCO1M im folgenden Übertragungsmodus ausgetauscht werden:

Modusnummer	Modustyp	Modusname
254 oder 255	asynchron	Change of state



Change of state (Modi 254 und 255)

"Change of state" entspricht der Änderung eines Eingangswerts (Ereignissteuerung). Die Daten werden unmittelbar nach der Änderung auf dem Bus gesendet. Die Ereignissteuerung ermöglicht eine optimale Nutzung der Busbandbreite, da nur die Änderung und nicht das gesamte Prozessabbild übertragen wird. Dies ermöglicht sehr kurze Antwortzeiten, da es, wenn sich ein Eingangswert ändert, nicht erforderlich ist, auf den nächsten Request vom Master zu warten.
Bei Auswahl der "change of state" PDO-Übertragung sollten Sie jedoch berücksichtigen, dass es wahrscheinlich ist, dass eine Menge von Ereignissen gleichzeitig auftreten können, was durch das Warten auf die Übertragung eines PDO mit geringerer Priorität auf dem Bus zu Verzögerungen führen kann. Sie sollten außerdem eine Situation vermeiden, in der die kontinuierliche Änderung eines Eingangs mit einem PDO mit hoher Priorität den Bus blockiert (dies wird als "babbling idiot" bezeichnet).

Hinweis: Generell sollten Sie die PDO-Übertragung mit analogen Eingangsmodulen nur dann nutzen, wenn der Delta-Modus (Objekt 6426H) oder die Sperrzeit (Objekte 1800H bis 1804H, Unterindex 3) so konfiguriert sind, dass eine Überlastung des Busses vermieden wird.

Datenzugriff per explizitem Austausch (SDO)

Was ist ein SDO?

Service Data Objects (SDO) ermöglichen den Zugriff auf die Daten eines Geräts mittels expliziter Requests.
Der SDO-Dienst ist verfügbar, wenn sich das Gerät im Zustand "Operational" oder "Pre-Operational" befindet.

SDO-Typen

Es gibt zwei Arten von SDOs:

- Lese-SDOs (Download SDO),
- Schreib-SDOs (Upload SDO).

Client/Server-Modell

Das SDO-Protokoll basiert auf einem "Client/Server"-Modell.

Download SDO

Der Client sendet einen Request, der das zu lesende Objekt angibt.
Der Server sendet die im Objekt enthaltenen Daten zurück.

Upload SDO

Der Client sendet einen Request, der das Objekt angibt, in das geschrieben werden soll. Außerdem gibt es den gewünschten Wert an.
Nach der Aktualisierung des Objekts sendet der Server eine Bestätigungsmeldung zurück.

Nicht verarbeitetes SDO

In beiden Fällen sendet der Server, wenn ein SDO nicht verarbeitet werden konnte, einen Fehlercode (abort code).

"Node Guarding" und "Life Guarding"

Definition der Life-Time

Der Parameter "Life time" wird wie folgt berechnet:
Life Time = Guard Time x Life Time Factor
Das Objekt 100CH enthält den Parameter ''Guard Time'', ausgedrückt in Millisekunden. Das Objekt 100DH enthält den Parameter ''Life Time Factor".

Aktivierung der Überwachung

Wenn einer dieser beiden Parameter auf 0 (Standardkonfiguration) gesetzt ist, führt das Modul keine Überwachung durch (kein "Life Guarding").
Um die Überwachung zeitgesteuert zu aktivieren, müssen Sie mindestens den Wert 1 in das Objekt 100DH eingeben und im Objekt 100CH eine Zeit in ms festlegen.

Garantie eines zuverlässigen Betriebs

Um einen zuverlässigen Betrieb zu garantieren, empfiehlt es sich, einen "Life time factor" von 2 einzugeben.
Falls nicht, schaltet das Modul, wenn eine Verzögerung auftritt (zum Beispiel aufgrund der höchsten Priorität oder der internen Verarbeitung im "Node Guarding"-Master), in den Zustand "Pre-Operational", ohne einen Fehler zu generieren.

Bedeutung der Überwachung

Diese zwei Überwachungsmechanismen sind besonders wichtig für das CANopen-System, da die Geräte üblicherweise nicht im ereignisgesteuerten Modus betrieben werden.

Slave-Überwachung

Die Überwachung erfolgt auf die folgende Art und Weise:

Phase	Beschreibung
1	Der Master setzt "Remote Frames" (remote transmit requests) an den "Guarding COB-IDs" der zu überwachenden Slaves.
2	Die betroffenen Slaves antworten durch das Senden der Meldung "Guarding". Sie enthält den "Status Code" des Slaves und das "Toggle Bit", das nach jeder Meldung geändert werden muss.
3	Der Master vergleicht die "Status"- und "Toggle Bit"-Informationen: Wenn Sie sich nicht in dem vom NMT-Master erwarteten Zustand befinden oder wenn keine Antwort empfangen wird, geht der Master davon aus, dass ein Fehler am Slave aufgetreten ist.

Master-Überwachung

Wenn der Master "Guarding"-Meldungen auf einer streng zyklischen Basis fordert, dann kann der Slave einen Master-Defekt erkennen.
Wenn der Slave innerhalb des festgelegten "Life Time" Intervalls keinen Request vom Master empfängt (Guarding-Fehler), geht er davon aus, dass ein Fehler am Master aufgetreten ist ("Watchdog"-Funktion).
In diesem Fall gehen die entsprechenden Ausgänge in den Fehlerzustand, und der Slave schaltet in den Modus "Pre-Operational" zurück.

Hinweis: Der "Remote"-Request vom Master erhält eine Antwort, selbst wenn keine Werte in die Objekte "Guard Time" und "Life Time Factor" eingegeben sind. Die Zeitüberwachung ist nur aktiviert, wenn die Werte in den beiden Objekten größer als 0 sind. Die typischen Werte für den Parameter "Guard Time" liegen zwischen 250 ms und 2 Sekunden.

"Guarding"-Protokoll

Der Wert des "Toggle Bit" (t), der in der ersten "Guarding"-Meldung gesendet wird, ist 0. Dann ändert sich das Bit ("toggles") in jeder weiteren "Guarding"-Meldung, wodurch es möglich ist, anzugeben, wenn eine Meldung verloren gegangen ist.
Der Buskopf gibt seinen Netzwerkzustand in den sieben verbleibenden Bits an:

Netzwerkzustand	Antwort
Stopped	0x04 oder 0x84
Pre-operational	0x7F oder 0xFF
Operational	0x05 oder 0x85

Verwaltung des internen Busses

Schalten des internen Busses in den Zustand "Stop"	Der interne Bus schaltet automatisch vom Zustand ''Stop'' in den Zustand ''Run'', wenn das Kommunikationsmodul vom Zustand "Pre-operational" in den Zustand "Operational" schaltet. Wenn der interne Bus in den Zustand ''Stop'' schaltet, werden alle Ausgänge des Erweiterungsmoduls auf 0 gesetzt. Die Ausgänge des Kommunikationsmoduls werden in ihrem aktuellen Zustand gehalten.
Konfiguration der Erweiterungs-module	Der interne Bus wird verwendet, um die Konfiguration der Parameter der digitalen und analogen Erweiterungsmodule zu aktualisieren. Die Parameter werden an das Kommunikationsmodul gesendet, wenn sich der Bus im Zustand ''Stop'' befindet. Diese neuen Konfigurationsparameter werden bestätigt, wenn der Bus in den Zustand ''Run'' wechselt.

10.2 Implementieren des CANopen-Busses

Überblick

Einleitung

Dieser Abschnitt beschreibt die Implementierung des CANopen-Feldbusses im Twido-SPS-System mittels des CANopen-Mastermoduls TWDNCO1M.

Inhalt dieses Abschnitts

Dieser Abschnitt enthält die folgenden Themen:

Thema	Seite
Überblick	262
Hardware-Setup	263
Konfigurationsverfahren	264
Deklaration des CANopen-Masters	266
CANopen-Netzwerk-Slave-Deklaration	267
CANopen-Objektzuweisung	271
Verbinden von CANopen-Objekten	275
Symbolisierung von CANopen-Objekten	277
Adressieren der PDOs des CANopen-Masters	279
Programmierung und Diagnose für den CANopen-Feldbus	280

Überblick

Hardware- und Softwareanforderungen

Die folgende Hardware- und Softwareanforderungen müssen für die Implementierung eines CANopen-Busses in Ihrem Twido-SPS-System erfüllt sein:

Hardware	Anforderungen
Kompaktes oder modulares Twido-SPS-Grundgerät	**Kompaktes Grundgerät:** • TWDLC•24DRF • TWDLCA•40DRF **Modulares Grundgerät:** • TWDLMDA20••• • TWDLMDA40•••
CANopen-Master	1 CANopen-Mastermodul: TWDNCO1M
CANopen-Slave-Geräte	max. 16 CANopen-Slaves
CANopen-Steckverbinder und -Kabel	
Programmierkabel für die Twido-SPS	

Software	Anforderungen
Twido-SPS-Konfigurationssoftware	TwidoSoft V3.0 oder höher

CANopen-Implementierungsverfahren

Die nachfolgend aufgeführten Verfahren geleiten Sie durch die Installation, Konfiguration und den Betrieb Ihres CANopen-Netzwerks:

Schritt	Beschreibung
1	Hardware-Setup
2	Konfigurationsverfahren
3	Deklaration des CANopen-Masters
4	CANopen-Netzwerk-Slave-Deklaration
5	CANopen-Objektzuweisung
6	Verbinden von CANopen-Objekten
7	Symbolisierung von CANopen-Objekten
8	CANopen-Netzwerkdiagnose
Die folgenden Unterabschnitte enthalten eine detaillierte Beschreibung jedes Schritts dieser Verfahren.	

Hardware-Setup

Installieren des Master-Moduls TWDNCO1M

Installieren Sie das Master-Modul TWDNCO1M in einem Twido-SPS-System (DIN-Schienen- oder Schaltpultmontage) und schließen Sie es an den internen Bus der Twido-SPS an (ausführliche Informationen finden Sie unter "TwidoHW - Installieren eines Erweiterungsmoduls"). Führen Sie die nachfolgend aufgeführten Schritte aus.

Schritt	Aktion	Beschreibung
1	Vorbereitung der Installation	Anweisungen zu den folgenden Themen finden Sie im *Twido-SPS-Hardware-Referenzhandbuch (TWD USE 10AE)*: ● korrekte Montagepositionen für Twido-Module, ● Hinzufügen und Entfernen von Twido-Komponenten zu bzw. von einer DIN-Schiene, ● direkte Montage auf einer Schaltpultoberfläche, ● Mindestabstände für Module in einem Schaltpult.
2	Montage des Moduls TWDNCO1M	Installieren Sie das Master-Modul TWDNCO1M auf einer DIN-Schiene oder in einem Schaltpult. Ausführliche Informationen finden Sie unter "TwidoHW - Installieren eines Erweiterungsmoduls".
3	Modulanschluss an den Bus der Twido-SPS	Schließen Sie das CANopen-Master-Modul an den internen Bus der Twido-SPS an (ausführliche Informationen finden Sie unter "TwidoHW - Installieren eines Erweiterungsmoduls").
4	CANopen-Verdrahtung und Anschlüsse	Befolgen Sie die unter "CANopen-Verdrahtung und -Anschlüsse" aufgeführten Verdrahtungs- und Anschlussvorschriften, um die Stromversorgung und Signalleitungen des CAN-Busses anzuschließen.

Konfigurationsverfahren

Überblick

Die CANopen-Konfiguration wird mittels des CANopen Configuration Tools durchgeführt, das in TwidoSoft ab V3.0 verfügbar ist.

Hinweis:
1. Die CANopen-Netzwerk-, Master- und Slave-Konfiguration sowie die Konfiguration der Kommunikationsparameter wird nur im Offline-Modus durchgeführt.
2. Im Online-Modus ist keine Änderung an der CANopen-Konfiguration zulässig.
3. Im Online-Modus können nur bestimmte Parameter wie beispielsweise die PDO-Adressierungsparameter %IWC und %QWC PDO angepasst werden.

Konfigurations-verfahren

Die nachfolgende Tabelle beschreibt die verschiedenen Phasen der softwaretechnischen Inbetriebnahme des CANopen-Busses.

Modus	Phase	Beschreibung
Offline-Zustand	Deklaration des Moduls TWDNCO1M	Wählen Sie eine verfügbare Steckplatznummer für die Installation des Mastermoduls TWDNCO1M am Twido-Erweiterungsbus aus.
	Konfiguration des CANopen-Netzwerks	Konfigurieren Sie das CANopen-Netzwerk durch: ● den Import der EDS-Dateien von allen Slaves in den Netzwerkkatalog, ● das Hinzufügen von Slaves aus dem Katalog zum CANopen-Netzwerk.
	PDO-Zuweisung	Führen Sie die Zuweisung der TPDO- und RPDO-Objekte jedes im Netzwerk deklarierten Slaves durch.
	PDO-Verbindung	Verbinden Sie jedes Slave-PDO mit dem entsprechenden PDO des Mastermoduls.
Offline oder online	Symbolisierung (optional)	Symbolisierung der Variablen, die den Slave-Geräten zugeordnet sind
	Programmieren	Programmieren der CANopen-Funktion.
Online-Zustand	Übertragen	Übertragung der Applikation in die Steuerung.
	Debug	Debuggen der Applikation mithilfe: ● des Debug-Fensters, das die Anzeige der Slaves (Adresse, Parameter) und die Adressierung der Slaves an den gewünschten Adressen ermöglicht. ● von Diagnosefenstern zum Identifizieren der Fehler.

Hinweis: Die Deklaration und das Löschen des CANopen-Mastermoduls TWDNCO1M auf dem Erweiterungsbus erfolgen wie für jedes andere Erweiterungsmodul. Es ist jedoch nur ein CANopen-Mastermodul auf dem Twido-Erweiterungsbus zulässig. Das TwidoSoft-Benutzerschnittstellenprogramm lässt kein Hinzufügen eines weiteren CANopen-Moduls zu.

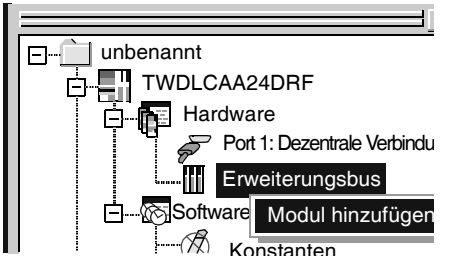
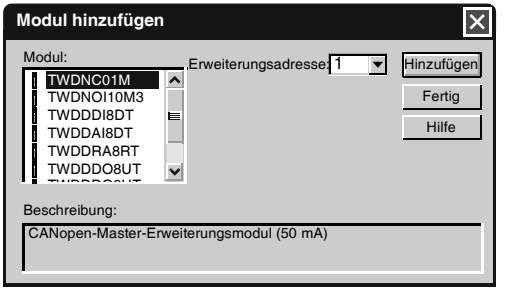
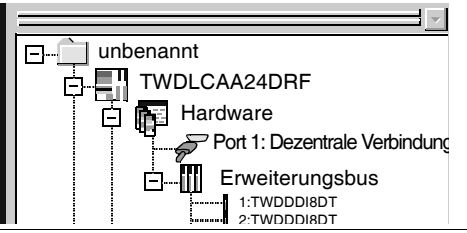
Vorsichtsmaßnahmen vor Erstellung der Verbindung

Vor dem Verbinden (softwaretechnisch) des PCs mit der Steuerung und zum Vermeiden von Erkennungsproblemen:

- Vergewissern Sie sich, dass kein Slave an der Adresse 127 des Busses physikalisch vorhanden ist (127 ist eine reservierte, werkseitig festgelegte Adresse, die dem Mastermodul TWDNCO1M zugewiesen ist).
- Vergewissern Sie sich, dass keine Slaves mit doppelter Adresse auf dem CANopen-Bus installiert sind.

Deklaration des CANopen-Masters

Verfahren Die nachfolgende Tabelle beschreibt die verschiedenen Schritte der Deklaration des CANopen-Masters.

Schritt	Aktion	Kommentar
1	Klicken Sie im TwidoSoft-Applikationsnavigator mit der rechten Maustaste auf **Erweiterungsbus** → **Modul hinzufügen...**	
2	Wenn das Dialogfeld **Modul hinzufügen** angezeigt wird: ● Wählen Sie **TWDNCO1M.** ● Klicken Sie auf **Hinzufügen.** ● Jetzt können Sie beliebige andere Erweiterungsmodule (bis zu 7) hinzufügen, die Sie in Ihr Twido-System integrieren möchten. **Hinweis:** Es ist nur ein CANopen-Master-Modul TWDNCO1M zulässig. ● Klicken Sie auf **Fertig.**	Es werden nur die Steuerungen TWDC•A24DRF, TWDCA•40DRF, TWDLMDA20••• und TWDLMDA40••• unterstützt. 
3	Es wird eine Erweiterungsbusstruktur angezeigt, die diesem Beispiel ähnelt. **Hinweis:** Ein TWDNCO1M-Master kann in jeden beliebigen Erweiterungssteckplatz von 1 bis 7 am Twido-Bus eingesteckt werden.	

CANopen-Netzwerk-Slave-Deklaration

Überblick

Die CANopen-Netzwerk-Slave-Deklaration ist ein drei Schritte umfassendes Verfahren:

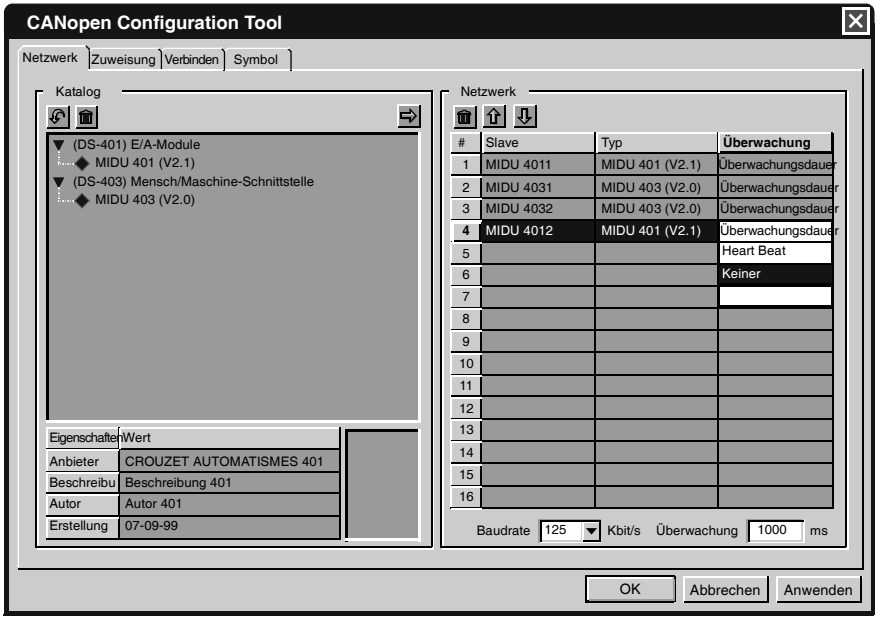
1. 1. Importieren der EDS-Dateien der CANopen-Slaves in den Katalog des Twido CANopen-Konfigurationseditors
2. 2. Erstellen des CANopen-Netzwerks durch Hinzufügen von bis zu 16 Slaves aus dem Katalog zum Netzwerk
3. 3. Konfiguration der Netzwerkmanagementparameter (Netzwerkgeschwindigkeit und Parameter des Fehlerüberwachungsprotokolls)

CANopen-Konfigurations-editor

Klicken Sie im TwidoSoft-Applikationsnavigator mit der rechten Maustaste auf den Namen des Master-Moduls, und wählen Sie **Hardware** → **Erweiterungsbus** → **TWDNCO1M** → **Konfigurieren** aus.
Ergebnis: Das CANopen Configuration Tool wird wie im folgenden Abschnitt dargestellt angezeigt.

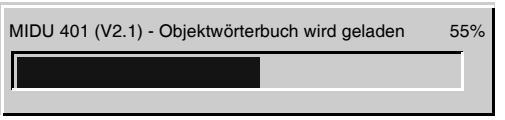
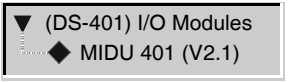
Dialogfeld "Netzwerk"

Die CANopen-Netzwerkkonfiguration und -Slave-Deklaration wird über das unten abgebildete Dialogfeld **Netzwerk** des TwidosSoft CANopen-Konfigurationseditors durchgeführt.



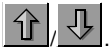
Importieren der Slave-Profile

Die nachfolgende Tabelle beschreibt die Vorgehensweise für den Import der CANopen-Slave-Profile (.EDS-Dateien) in den Katalog des CANopen Configuration Tools.

Schritt	Aktion
1	Klicken Sie im Dialogfeld "Netzwerk" im Bereich **Katalog** auf das Symbol **Importieren** . **Ergebnis:** Das Dialogfeld "Öffnen" des Betriebssystems wird angezeigt.
2	Suchen Sie den Pfad des Verzeichnisses, in dem die **EDS**-Dateien der CANopen-Slaves, die Sie zum Katalog hinzufügen möchten, gespeichert sind. **Ergebnis:** Die Namen der verfügbaren EDS-Dateien werden im Dialogfeld "Öffnen" angezeigt.
3	Wählen Sie eine EDS-Datei **("Dateiname".EDS)** aus der Liste aus und klicken Sie auf **Öffnen.** **Ergebnis:** Das CANopen Configuration Tool lädt das Objektwörterbuch für das ausgewählte Gerät. **Hinweis:** Dieser Vorgang kann abhängig von der Größe der ausgewählten EDS-Datei einige Minuten dauern. Ein Statusbalken zeigt den Fortschritt des Ladevorgangs an (siehe Beispiel unten). 
4	Warten Sie, bis der Ladevorgang abgeschlossen ist, und wiederholen Sie anschließend die Schritte 2 bis 3 für jedes neue Slave-Profil, das Sie zum Katalog hinzufügen möchten. **Hinweis:** Sie müssen diesen Vorgang nur einmal durchführen, damit alle im geladenen Katalog aufgeführten Geräteprofile und Objektwörterbücher von TwidoSoft gespeichert werden.
5	Gehen Sie folgendermaßen vor, um die Geräteeigenschaften eines CANopen-Slaves anzuzeigen: **1.** Doppelklicken Sie auf den im Katalog aufgeführten Gerätetyp. **Beispiel:** . **Ergebnis:**  . **2.** Klicken Sie einmal auf das Slave-Profil (z.B.: MIDU 401 V2.1). **Ergebnis:** Die Geräteeigenschaften des ausgewählten CANopen-Slaves werden in der unteren Hälfte des Katalogbereichs angezeigt. Sie umfassen: • den Herstellernamen (z.B.: Crouzet Automatismes 401), • das Slave-Profil (z.B. Beschreibung 401), • den Namen des Autors (z.B.: Autor 401), • das Erstellungsdatum dieses Profils (z.B.: 07-09-99).
6	Um ein Slave-Profil aus dem Katalog zu löschen, wählen Sie den Gerätenamen im Fenster "Katalog" aus und klicken Sie auf das Symbol **Löschen** . **Hinweis:** Sie können im CANopen-Netzwerkkatalog mehr Geräteprofile speichern, als Sie tatsächlich für Ihre aktuelle CANopen-Buskonfiguration benötigen. Profile, die bereits in den Katalog geladen wurden, können für zukünftige Zwecke dienen.
7	Klicken Sie auf die Schaltfläche **Übernehmen**, um die Änderungen am Katalog zu bestätigen und um die Slave-Profile im TwidoSoft-Projekt zu speichern.

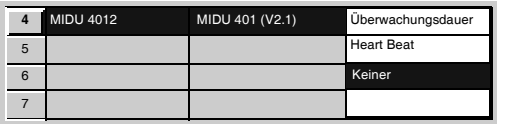
Errichten des CANopen-Netzwerks

Die nachfolgende Tabelle beschreibt die Vorgehensweise zum Deklarieren von Slaves im Twido CANopen-Netzwerk. (Beachten Sie, dass Sie nur Slaves deklarieren können, deren EDS-Profile zuvor zum Katalog hinzugefügt wurden oder bereits im Katalog gespeichert sind.)

Schritt	Aktion
1	Wählen Sie im Dialogfeld "Netzwerk" im Bereich **Katalog** das Slave-Profil aus der Liste der verfügbaren, bereits im Katalog gespeicherten Geräte aus. **Ergebnis:** Das Symbol **Hinzufügen** wird in der oberen rechten Ecke des Katalograhmens angezeigt.
2	Klicken Sie einmal auf das Symbol **Hinzufügen** . **Ergebnis:** Der Slave wird zur Tabelle der Netzwerk-Slaves hinzugefügt. **Hinweise:** ● In einem Twido CANopen-Netzwerk können maximal 16 Slaves deklariert werden. ● Der neu deklarierte Slave erhält die Knotenadresse mit dem kleinsten verfügbaren Index. (Wenn beispielsweise Slaves an den Knotenadressen 1, 2 und 4 deklariert sind, erhält der neu hinzugefügte Slave standardmäßig die verfügbare Knotenadresse 3.)
3	Sie können einem Slave jede verfügbare Knotenadresse (1 bis 16) zuweisen. Um einen Slave zur gewünschten Knotenadresse zu verschieben, verwenden Sie die Pfeilsymbole **Nach oben/unten verschieben**  .
4	Wiederholen Sie die Schritte 1 bis 3 für jeden neuen Slave, den Sie im CANopen-Netzwerk deklarieren möchten.
5	Um einen Slave aus dem Netzwerk zu löschen, wählen Sie den Gerätenamen in der Slave-Tabelle aus und klicken Sie auf das Symbol **Löschen** .
6	Klicken Sie auf die Schaltfläche **Übernehmen**, um die Änderungen zu bestätigen und um die Netzwerkkonfiguration im TwidoSoft-Projekt zu speichern.

Konfigurieren der Netzwerkmanage-mentparameter

Nachfolgend ist das Verfahren zum Konfigurieren der Netzwerkmanagementparameter wie etwa der Baudrate (Netzwerkgeschwindigkeit), der life-time und des Fehlerüberprüfungsprotokolls beschrieben.

Schritt	Aktion
1	Wählen Sie im Dialogfeld "Netzwerk" die Option Baudrate (Netzwerkgeschwindigkeit) aus der Dropdown-Liste aus: **10, 20, 50, 100, 125 (Standardwert), 250, 500, 800 oder 1000 Kbit/s.** **Hinweis:** Vergewissern Sie sich, dass jeder im Netzwerk deklarierte Slave einzeln konfiguriert wird, so dass seine eigene Baudrate genau identisch mit der oben festgelegten Netzwerkgeschwindigkeit ist, da andernfalls die CANopen-Netzwerkkommunikation nicht ordnungsgemäß funktioniert.
2	Konfigurieren Sie die Life-time-Dauer. Dieser Parameter legt die Kommunikationszykluszeit fest, die im Überwachungsfeld jedes Slaves implementiert wird (wie in Schritt 3 unten erklärt). **Hinweis:** Geben Sie nicht den Wert 0 in dieses Feld ein.
3	Klicken Sie einmal in das Feld **Überwachung**, um die Optionen des Fehlerüberprüfungsprotokolls jedes in der Slave-Tabelle des Netzwerks aufgeführten Slaves zu konfigurieren. **Ergebnis:** Die vom ausgewählten Gerät unterstützten verfügbaren Überwachungsoptionen werden in einem Listenfeld angezeigt (siehe folgendes Beispiel). 
4	Wählen Sie das Fehlerüberprüfungsprotokoll aus, das Sie für die Verwaltung der Kommunikation zwischen dem TWDNCO1M-Mastermodul und dem ausgewählten Slave verwenden möchten. • Guard Time • Heartbeat • Keins
5	Wenn die Überwachungsoption in der Slave-Tabelle des Netzwerks auf **Keins** gesetzt ist, nehmen die Ausgänge bei einer Unterbrechung der Verbindung (*) zwischen diesem Slave und dem TWDNCO1M-Mastermodul nicht ihre Fehlerwerte an. (*) Diese Unterbrechung kann hervorgerufen werden durch: • Abziehen des Erweiterungsbuskabels, das das CANopen-Mastermodul TWDNCO1M mit der Twido-Grundsteuerung verbindet, • Trennung dieses CANopen-Slaves vom Twido CANopen-Bus, • fehlerhaftes Buskabel, • ein Befehl zum Rücksetzen von TwidoSoft (Online → Firmware/Reset), • ein Befehl zum Laden einer TwidoSoft-Konfiguration (Online → Download), • ein Befehl zum Laden der Firmware in das TWDNCO1M-Mastermodul, ausgelöst über TwidoSoft (Online → Firmware Download).
6	Klicken Sie auf die Schaltfläche **Übernehmen**, um die Änderungen zu bestätigen und um die Netzwerkkonfiguration im TwidoSoft-Projekt zu speichern.

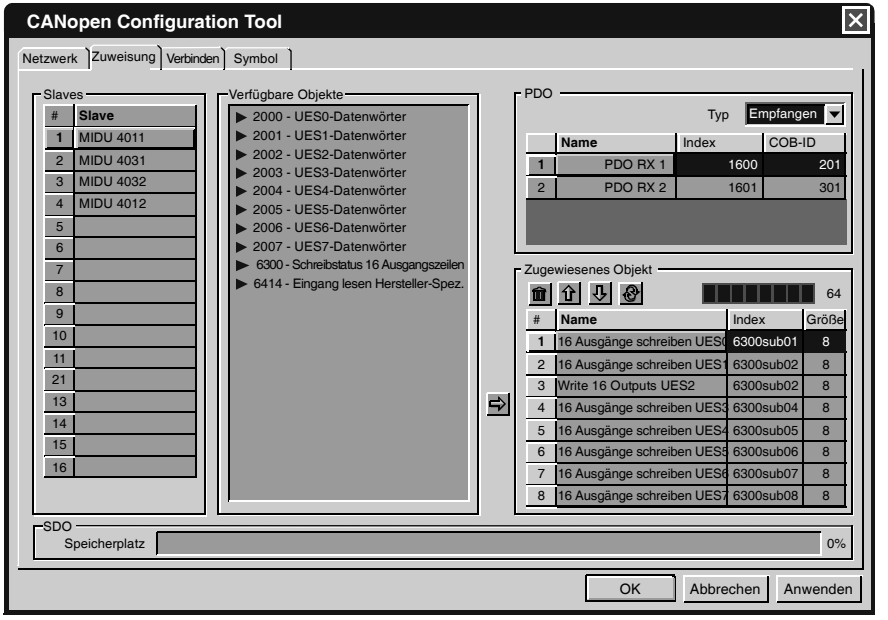
CANopen-Objektzuweisung

Überblick

Das Dialogfeld **Zuweisung** des CANopen Configuration Tools ermöglicht Ihnen die Konfiguration der PDOs jedes im Netzwerk deklarierten Slaves.

Dialogfeld "Zuweisung"

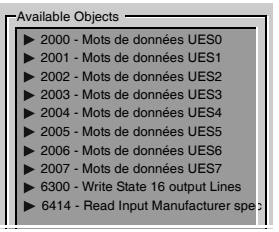
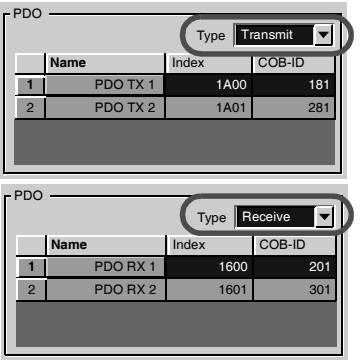
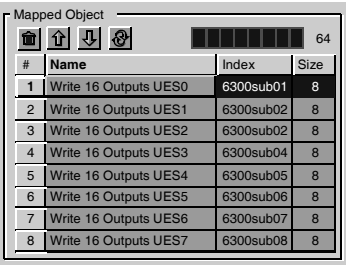
Klicken Sie im TwidoSoft-Applikationsnavigator mit der rechten Maustaste auf den Namen des Mastermoduls, und wählen Sie **Hardware** → **Erweiterungsbus** → **TWDNCO1M** → **Konfigurieren** aus. Klicken Sie dann auf die Registerkarte **Zuweisung** des CANopen Configuration Tools.
Ergebnis: Das CANopen Configuration Tool wird wie in der folgenden Abbildung dargestellt angezeigt.



Objektzuweisung

Nachfolgend ist das Verfahren zur Konfiguration der TPDOs und RPDOs jedes Slaves mittels des Dialogfelds "Zuweisung" beschrieben.

Schritt	Aktion
1	Klicken Sie im Bereich **Slaves** einmal auf den Gerätenamen, um den Slave auszuwählen, dessen PDOs Sie konfigurieren möchten.
2	**Beispiel:** Das DS-401 E/A-Modul mit der Bezeichnung MIDU 4011. Beachten Sie, dass die Slavenamen und Knotenadressen in diesem Bereich genau so angezeigt werden, wie im vorherigen Schritt der Netzwerkkonfiguration angegeben (siehe *CANopen-Netzwerk-Slave-Deklaration, S. 267*.)

Schritt	Aktion
3	**Ergebnis:** **1.** Alle vom ausgewählten Slave unterstützten CANopen-Objekte werden im Bereich **Verfügbare Objekte** angezeigt (siehe Beispiel unten).  **2.** Der Bereich **PDO** zeigt standardmäßig die vordefinierten Transmit-PDOs (PDO TX) für den ausgewählten Slave. Sie können die Umschaltliste **Typ** verwenden, um auch die vordefinierten Receive-PDOs (PDO RX) anzuzeigen. In diesem Beispiel unterstützt das E/A-Modul MIDU 4011 DS-401 zwei Transmit-PDOs (PDO TX) und zwei Receive-PDOs (PDO RX) (siehe Abbildung).  **3.** Außerdem wird im Bereich **Zugewiesene Objekte** die vordefinierte Zuweisung jedes ausgewählten PDO angezeigt. : 

Schritt	Aktion
4	Sie können die PDO-Zuweisung mittels des Bereichs **Zugewiesene Objekte** ändern. Ein RPDO oder TPDO ist ein 64-Byte-Objekt, das bis zu acht 8-Byte-Wortobjekte oder vier 16-Byte-Wortobjekte oder jede Kombination dieser beiden Wortobjekttypen enthält, wobei die 64-Byte-Obergrenze des PDO jedoch nicht überschritten werden darf. Um die PDO-Zuweisung anzupassen, fahren Sie mit Schritt 5 folgende unten fort und befolgen Sie die Anweisungen.
5	Wählen Sie für den gewünschten Slave (siehe Schritt 2) im Bereich **PDO** das PDO aus, dessen Zuweisung Sie ändern möchten. **Beispiel:** Wählen Sie das erste Transmit-PDO (PDO TX 1). **Ergebnis:** Die vordefinierte PDO-Zuweisung (oder die aktuell angepasste Zuweisung) wird im Bereich "Zugewiesene Objekte" angezeigt.
6	Um ein nicht verwendetes Wortobjekt aus der PDO-Zuweisungsstruktur zu löschen, wählen Sie das Wortobjekt (mit Index 1 bis 8 versehen) aus, und klicken Sie auf das Symbol **Löschen** .
7	Wählen Sie im Bereich **Verfügbare Objekte** das Wortobjekt in der Objektfamilie aus, das Sie zuweisen möchten, und klicken Sie auf das Symbol **Hinzufügen** , um das Wortobjekt zur Struktur **Zugewiesene Objekte** hinzuzufügen. **Hinweis:** Um die standardmäßige Zuweisungsstruktur für das ausgewählte PDO wiederherzustellen, klicken Sie auf das Symbol **Standard** .
8	Um die Adresse eines Wortobjekts in der zugewiesenen PDO-Struktur zu ändern, verwenden Sie die Pfeiltasten **Nach oben/unten verschieben** .
9	Klicken Sie auf die Schaltfläche **Übernehmen**, um die Änderungen an der zugewiesenen PDO-Struktur zu bestätigen und um die PDO-Zuweisung im TwidoSoft-Projekt zu speichern.
10	Wiederholen Sie die Schritte 5 bis 9 für jede PDO-Zuweisung, die Sie konfigurieren möchten.
11	**Hinweise zur Speicherbilanz:** ● **PDO-Speicherbilanz:** Die Verwendung des PDO-Speichers kann mittels der Speicherstatusleiste in der oberen rechten Ecke des Bereichs "Zugewiesene Objekte" überwacht werden. . ● **Zusätzliche SDO-Speicherverwendung:** Vordefinierte PDOs und Wortobjekte belegen keinen zusätzlichen SDO-Speicher. Das Löschen und Hinzufügen von Wortobjekten zur PDO-Zuweisungsstruktur erfordert jedoch die Verwendung von zusätzlichem Systemspeicher. Die aktuelle Verwendung des SDO-Speichers wird in der Statusleiste unten im Dialogfeld "Zuweisung" angezeigt. .

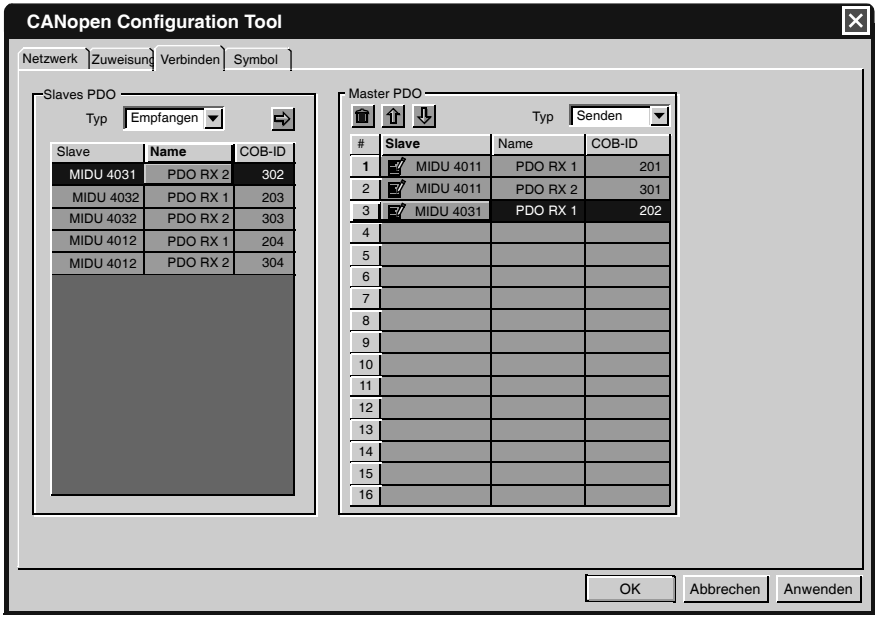
Verbinden von CANopen-Objekten

Überblick

Das Dialogfeld **Verbinden** des CANopen Configuration Tool wird verwendet, um die physikalische Verbindung zwischen den ausgewählten PDOs der Slaves und den PDOs des CANopen-Mastermoduls TWDNCO1M festzulegen.

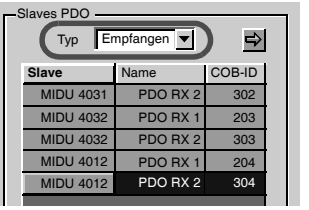
Dialogfeld "Verbinden"

Klicken Sie im TwidoSoft-Applikationsnavigator mit der rechten Maustaste auf den Namen des Mastermoduls, und wählen Sie **Hardware** → **Erweiterungsbus** → **TWDNCO1M** → **Konfigurieren** aus. Klicken Sie dann auf die Registerkarte **Verbinden** des CANopen Configuration Tools.
Ergebnis: Das CANopen Configuration Tool wird wie in der folgenden Abbildung dargestellt angezeigt.



Verbinden von Objekten

Nachfolgend ist beschrieben, wie Sie das Dialogfeld "Verbinden" verwenden, um die physikalische Verbindung zwischen den PDOs des Slaves und des Mastermoduls zu definieren.

Schritt	Aktion
1	Wählen Sie im Bereich **Slave PDOs** den PDO-**Typ aus:** Receive oder Transmit. **Ergebnis:** Alle PDO-Slaves des ausgewählten Typs werden im Bereich "Slave PDOs" angezeigt (siehe Beispiel unten).  **Hinweis:** Die Auswahl von Receive oder Transmit im Bereich "Slave PDOs" schaltet automatisch die PDOs des Masters in den entgegengesetzten Typ um: Transmit bzw. Receive.
2	Wählen Sie im Bereich **Slave PDOs** das PDO aus, das Sie mit dem CANopen-Master TWDNCO1M verbinden möchten, und klicken Sie auf das Symbol **Hinzufügen** , um das PDO zur Verbindungsliste der **Master PDOs** hinzuzufügen. **Hinweis:** Der Master TWDNCO1M unterstützt maximal 16 TPDO-Verbindungen und 16 RPDO-Verbindungen.
3	Um den Adressindex einer PDO-Verbindung im Bereich "Master PDOs" zu ändern, verwenden Sie die Pfeiltasten **Nach oben/unten verschieben** .
4	Um eine nicht verwendete PDO-Verbindung im Bereich "Master PDOs" zu löschen, wählen Sie das gewünschte PDO (mit Index 1 bis 16) aus, und klicken Sie dann auf das Symbol **Löschen** .
5	Klicken Sie auf die Schaltfläche **Übernehmen**, um die Änderungen an der zugewiesenen PDO-Struktur zu bestätigen und um die PDO-Zuweisung im TwidoSoft-Projekt zu speichern.
6	Wiederholen Sie die Schritte 1 bis 5 für jedes Slave PDO, das Sie mit dem CANopen-Master verbinden möchten.

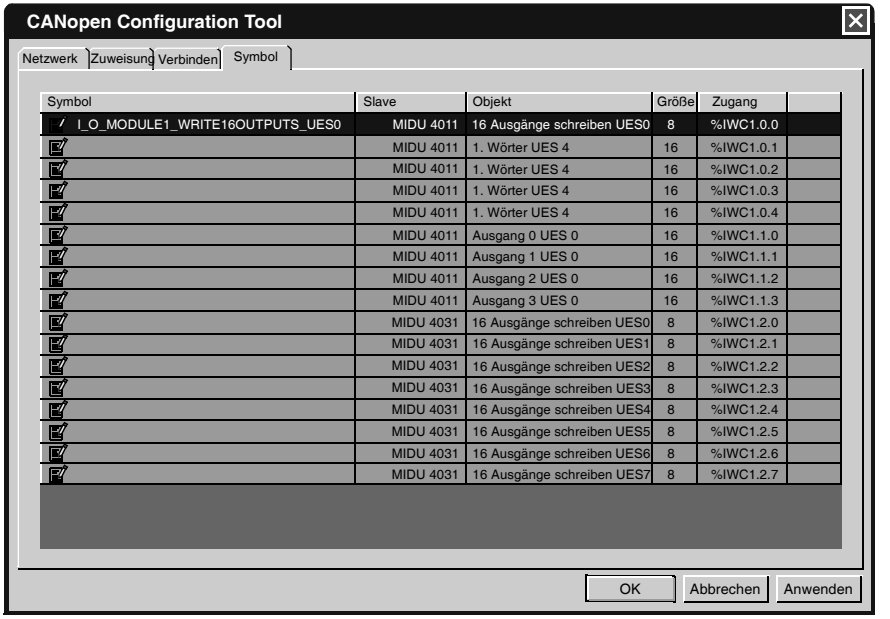
Symbolisierung von CANopen-Objekten

Überblick

Das Dialogfeld **Symbol** ermöglicht Ihnen die Definition einer Symbolisierung der mit dem CANopen-Master verbundenen Symbole.

Dialogfeld "Symbol"

Klicken Sie im TwidoSoft-Applikationsnavigator mit der rechten Maustaste auf den Namen des Mastermoduls, und wählen Sie **Hardware** → **Erweiterungsbus** → **TWDNCO1M** → **Konfigurieren** aus. Klicken Sie dann auf die Registerkarte **Symbol** des CANopen Configuration Tools.
Ergebnis: Das CANopen Configuration Tool wird wie in der folgenden Abbildung dargestellt angezeigt.



Symbol	Slave	Objekt	Größe	Zugang
I_O_MODULE1_WRITE16OUTPUTS_UES0	MIDU 4011	16 Ausgänge schreiben UES0	8	%IWC1.0.0
	MIDU 4011	1. Wörter UES 4	16	%IWC1.0.1
	MIDU 4011	1. Wörter UES 4	16	%IWC1.0.2
	MIDU 4011	1. Wörter UES 4	16	%IWC1.0.3
	MIDU 4011	1. Wörter UES 4	16	%IWC1.0.4
	MIDU 4011	Ausgang 0 UES 0	16	%IWC1.1.0
	MIDU 4011	Ausgang 1 UES 0	16	%IWC1.1.1
	MIDU 4011	Ausgang 2 UES 0	16	%IWC1.1.2
	MIDU 4011	Ausgang 3 UES 0	16	%IWC1.1.3
	MIDU 4031	16 Ausgänge schreiben UES0	8	%IWC1.2.0
	MIDU 4031	16 Ausgänge schreiben UES1	8	%IWC1.2.1
	MIDU 4031	16 Ausgänge schreiben UES2	8	%IWC1.2.2
	MIDU 4031	16 Ausgänge schreiben UES3	8	%IWC1.2.3
	MIDU 4031	16 Ausgänge schreiben UES4	8	%IWC1.2.4
	MIDU 4031	16 Ausgänge schreiben UES5	8	%IWC1.2.5
	MIDU 4031	16 Ausgänge schreiben UES6	8	%IWC1.2.6
	MIDU 4031	16 Ausgänge schreiben UES7	8	%IWC1.2.7

Symbolisierung von Objekten

Nachfolgend ist das Verfahren zur Definition von Symbolen für die CANopen-Objektvariablen mittels des Dialogfelds "Symbol" beschrieben.

Schritt	Aktion
1	Doppelklicken Sie im Feld **Symbol** auf das Symbolbearbeitungssymbol in der Zeile, deren Variable Sie symbolisieren möchten. **Ergebnis:** Das Textfeld "Symbol" wird aktiviert, und der Cursor wird rechtsbündig im Textfeld positioniert.
2	Geben Sie einen beschreibenden Namen ein. Ein gültiges Symbol kann bis zu 32 Zeichen umfassen: Es sind nur die Buchstaben A-Z, die Ziffern 0-9 und der Unterstrich "_" zulässig ("/", "%", Leerzeichen oder jegliche andere Sonderzeichen sind unzulässig). **Hinweis:** Ausführliche Informationen über die Bearbeitung von Symbolen finden Sie unter *Symbolisieren von Objekten, S. 52*.
3	Klicken Sie auf die Schaltfläche **Anwenden**, um die Änderungen an der Symboltabelle zu bestätigen und im TwidoSoft-Projekt zu speichern.
4	Wiederholen Sie die Schritte 1 bis 3 für jede Variable, die Sie symbolisieren möchten.

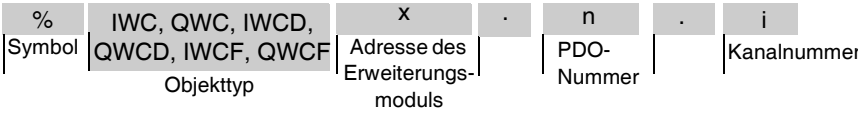
Adressieren der PDOs des CANopen-Masters

Auf einen Blick

Dieser Abschnitt beschreibt die Adressierung der PDO-Eingänge und PDO-Ausgänge des CANopen-Masters.
Um Verwechselungen mit dezentralen E/A zu vermeiden, wurde für die Syntax der CANopen-Objekte eine neue Bezeichnung eingeführt: %I**WC** zum Beispiel.

Abbildung

Kurze Zusammenfassung des Adressierungsprinzips:



Spezifische Werte

Die nachfolgende Tabelle enthält die spezifischen Werte für die CANopen-Slaveobjekte.

Teil	Werte	Kommentar
IWC	-	Abbild des physikalischen PDO-Eingangs
QWC	-	Abbild des physikalischen PDO-Ausgangs
IWCD	-	Gleiche Verwendung wie IWC, jedoch im Doppelwortformat.
QWCD	-	Gleiche Verwendung wie QWC, jedoch im Doppelwortformat.
IWCF	-	Gleiche Verwendung wie IWC, jedoch im Gleitkommaformat.
QWCF	-	Gleiche Verwendung wie QWC, jedoch im Gleitkommaformat.
x	1 bis 7	Adresse des CANopen-Mastermoduls TWDNCO1M auf dem Twido-Erweiterungsbus
n	0 bis 15	PDO-Nummer (gemäß dem PDO-Index)
i	0 bis 7	Kanalnummer (gemäß dem PDO-Subindex)

Beispiel

Die nachfolgende Tabelle zeigt ein Beispiel für die PDO-Adressierung.

E/A-Objekt	Beschreibung
%IWC4.1.0	PDO-Nummer 1, Subindex 0, Eingang des CANopen-Moduls an Adresse 4 auf dem Twido-Erweiterungsbus.

Impliziter Austausch

Die nachfolgend beschriebenen Objekte werden implizit ausgetauscht, d.h., sie werden automatisch bei jedem Steuerungszyklus ausgetauscht.

Programmierung und Diagnose für den CANopen-Feldbus

Expliziter Austausch

Mit dem CANopen-Feldbus verknüpfte Objekte (Wörter und Bits) bieten Informationen (Bsp.: Funktion des Busses, Status der Slaves etc.) und zusätzliche Befehle für eine fortgeschrittene Programmierung der CANopen-Funktion.
Diese Objekte werden explizit über den Erweiterungsbus zwischen der Twido-Steuerung und dem CANopen-Mastermodul ausgetauscht:

- nach Aufforderung durch das Anwenderprogramm mithilfe folgender Anweisung: CAN_CMD (siehe "Beschreibung der Anweisung "CAN_CMD" weiter unten)
- über das Debugging-Fenster oder die Animationstabelle

Spezifische, für den CANopen-Master reservierte Systemwörter

Die in der Twido-Steuerung für das CANopen-Mastermodul reservierten Systemwörter geben den Status des Netzwerks wieder: %SW8x (x=1-7) ist reserviert für das an der Erweiterungsadresse x auf dem Twido-Bus installierte CANopen-Mastermodul. Nur die 7 ersten Bits dieses Wortes werden verwendet. Sie sind schreibgeschützt.
Die folgende Tabelle beschreibt die verwendeten Bits:

Systemwörter	Bit	Beschreibung
%SW8x (x=1-7)	0	Konfigurationsstatus des CANopen-Masters (= 1 wenn Konfiguration OK, ansonsten 0)
	1	Betriebsart des CANopen-Masters (= 1 Datenaustausch aktiviert, ansonsten 0)
	2	System im Status STOP (= 1 wenn der Offline-Modus aktiviert ist, ansonsten 0)
	3	CAN_CMD-Anweisung vollständig (= 1, wenn die Anweisung vollständig ist, ansonsten 0, wenn der Befehl ausgeführt wird)
	4	CAN_CMD-Anweisungsfehler (= 1 bei einem Fehler in der Anweisung, ansonsten 0)
	5	Initialisierungsfehler (= 1)
	6	Verlust der Meldung, Stromversorgungsfehler (= 1)

Nutzungsbeispiel (für das an der Erweiterungsadresse 1 auf dem Twido-Bus installierte CANopen-Mastermodul):
Vor der Nutzung einer CAN_CMD-Anweisung muss das Bit %SW81:X3 überprüft werden, um zu gewährleisten, dass keine Anweisung ausgeführt wird: Überprüfen, ob %SW81:X3 = 1.
Um zu ermitteln, ob die Anweisung anschließend fehlerfrei ausgeführt worden ist, muss überprüft werden, ob das Bit %SW81:X4 auf 0 gesetzt ist.

Spezifische, für den CANopen-Slave reservierte Systemwörter

%SW20 bis %SW27 sind reservierte Systemwörter, die Ihnen die Ermittlung des aktuellen Zustands der 16 CANopen-Slaves mit den Knotenadressen von 1 bis 16 ermöglichen. Der Inhalt dieser Speicherwörter ist schreibgeschützt.
Die folgende Tabelle beschreibt die Systemwörter %SW20 bis %SW27:

Systemwörter	Knotenadresse (Slave-Nummer)		Wortinhalt/Beschreibung
	Bit [0-7]	Bit [8-15]	
%SW20	1	2	Wenn %SW2x den folgenden Wert annimmt: ● = 1 => Ein unerwartetes Modul war im Netzwerk vorhanden. Es hat sich selbst als "nicht fehlerfrei" gemeldet, bevor es aus dem Netzwerk entfernt wurde. ● = 2 => Knotenstatus: in Betrieb (das Modul befindet sich im Status "operational"): - "fehlerfrei". ● = 3 => Knotenstatus: in Betrieb (das Modul befindet sich im Status "operational"): - "nicht fehlerfrei". ● = 4 => Knotenstatus: "Pre-operational" (das Modul befindet sich im Status "pre-operational"): - nur erwartete Module (die, die in der Konfigurationstabelle als erwartet deklariert sind); - das Modul kann in den Status "operational" gesetzt werden; - "fehlerfrei". ● = 5 => Knotenstatus: "Pre-operational" (das Modul befindet sich im Status "pre-operational"): - nur erwartete Module (die, die in der Konfigurationstabelle als erwartet deklariert sind); - das Modul kann in den Status "operational" gesetzt werden; - "nicht fehlerfrei".
%SW21	3	4	
%SW22	5	6	
%SW23	7	8	
%SW24	9	10	
%SW25	11	12	
%SW26	13	14	
%SW27	15	16	

Systemwörter	Knotenadresse (Slave-Nummer)		Wortinhalt/Beschreibung
	Bit [0-7]	Bit [8-15]	
			• = 6 => Knotenstatus: "Pre-operational" (das Modul befindet sich im Status "pre-operational"): - nur erwartete Module (die, die in der Konfigurationstabelle als erwartet deklariert sind); - das Modul ist vorhanden, aber sein aktueller Zustand ermöglicht es nicht, es in den Status "operational" zu setzen; - "fehlerfrei". • = 7 => Knotenstatus: "Pre-operational" (das Modul befindet sich im Status "pre-operational"): - nur erwartete Module (die, die in der Konfigurationstabelle als erwartet deklariert sind); - das Modul ist vorhanden, aber sein aktueller Zustand ermöglicht es nicht, es in den Status "operational" zu setzen; - "nicht fehlerfrei". • = 8 => Falsches Modul (es wurde ein Modul mit unterschiedlichen Geräteidentifikationsdaten erkannt): - "fehlerfrei". • = 9 => Falsches Modul (es wurde ein Modul mit unterschiedlichen Geräteidentifikationsdaten erkannt): - "nicht fehlerfrei". • = 10 => Slave-Konfigurationsfehler (das Modul hat den SDO-Schreib-Request der SDO-Befehlstabelle mit einer Fehlerbestätigung beantwortet oder nicht die Regeln des SDO-Protokolls befolgt): - "fehlerfrei". • = 11 => Slave-Konfigurationsfehler: - "nicht fehlerfrei". • = 12 => Fehlendes Modul/Fehlerüberprüfungs-Timeout/SDO-Timeout (ein Modul, das konfiguriert wurde, ist nicht verfügbar, wurde während des Betriebs entfernt oder antwortet nicht auf den SDO-Zugriff): - "fehlerfrei".
			• = 13 => Fehlendes Modul/Fehlerüberprüfungs-Timeout/SDO-Timeout (ein Modul, das konfiguriert wurde, ist nicht verfügbar, wurde während des Betriebs entfernt oder antwortet nicht auf den SDO-Zugriff): - "nicht fehlerfrei". • = 14 => Unerwartetes Modul (es wurde ein Module erkannt, das sich nicht in der Konfigurationstabelle befindet): - "fehlerfrei". • = 15 => Unerwartetes Modul (es wurde ein Module erkannt, das sich nicht in der Konfigurationstabelle befindet): - "nicht fehlerfrei".

Beschreibung der Anweisung CAN_CMD

Über das Benutzerprogramm ermöglicht die Anweisung CAN_CMD dem Benutzer die Programmierung seines Netzwerks und den Erhalt einer Diagnose der Slaves. Die Parameter der Anweisung werden über interne Speicherwörter **%MW**x weitergeleitet.
Die Syntax der Anweisung lautet wie folgt:
CAN_CMDn **%MW**x**:**l
Legende:

Symbol	Beschreibung
n	Erweiterungsadresse des CANopen-Mastermoduls auf dem Twido-Bus (1 bis 7)
x	Nummer des ersten, im Parameter (0 bis 254) weitergeleiteten internen Speicherwortes
l	Länge der Anweisung in Wörtern (2).

Verwendung der Anweisung CAN_CMD

Die Anweisung CAN_CMD ermöglicht Ihnen die Programmierung und Verwaltung des CANopen-Netzwerks und die Durchführung von Diagnosen der individuellen Slaves. Die Befehlsparameter werden über die Speicherwörter %MWx weitergeleitet.
Die folgende Tabelle beschreibt die Aktion der Anweisung CAN_CMD abhängig vom Wert der Parameter %MW(x) bis %MW(x+5), falls erforderlich.

<table>
<tr><th rowspan="2">%MWx</th><th colspan="2">%MWx+1</th><th colspan="2">%MWx+2</th><th colspan="2">%MWx+3</th><th colspan="2">%MWx+4</th><th colspan="2">%MWx+5</th><th rowspan="2">Aktion</th></tr>
<tr><th>Bit [0-7]</th><th>Bit [8-15]</th><th>Bit [0-7]</th><th>Bit [8-15]</th><th>Bit [0-7]</th><th>Bit [8-15]</th><th>Bit [0-7]</th><th>Bit [8-15]</th><th>Bit [0-7]</th><th>Bit [8-15]</th></tr>
<tr><td>1</td><td colspan="2">0</td><td colspan="8" rowspan="4">—</td><td>Umschalten vom Initialisierungsmodus in den Modus "Pre-operational".</td></tr>
<tr><td>1</td><td colspan="2">1</td><td>Reset der CANopen-Knoten</td></tr>
<tr><td>2</td><td colspan="2">0</td><td>Umschalten vom Modus "Operational" in den Modus "Pre-operational".</td></tr>
<tr><td>2</td><td colspan="2">1</td><td>Umschalten in den Modus "Operational".</td></tr>
<tr><td rowspan="6">3 oder 4</td><td colspan="2" rowspan="2"></td><td colspan="8" rowspan="2"></td><td>3 => Start des Befehls "SDO lesen".
4 => Start des Befehls "SDO schreiben".</td></tr>
<tr><td>Knoten = 1-16 => Knotenadresse</td></tr>
<tr><td colspan="2">Index</td><td colspan="8"></td><td>PDO-Objektindex.</td></tr>
<tr><td colspan="4"></td><td>Len</td><td>Sub</td><td colspan="4"></td><td>Sub = 0-255 => Objekt-Subindex
Len = Datenlänge in Bytes</td></tr>
<tr><td colspan="6"></td><td colspan="2">Daten 1</td><td colspan="2"></td><td>Nutzinformation gemäß dem Wert des Längenfelds (Len)</td></tr>
<tr><td colspan="8"></td><td colspan="2">Daten 2</td><td>Nutzinformation gemäß dem Wert des Längenfelds (Len)</td></tr>
</table>

Hinweis: Der Busstatus wird bei jedem Steuerungszyklus aktualisiert. Das Ergebnis der Bus-Leseanweisung CAN_CMD ist jedoch erst im nächsten Steuerungszyklus verfügbar.

Programmier-beispiel der Anweisung CAN_CMD

Beispiel 1:
Um den CANopen-Master (an der Adresse 1 auf dem Twido-Erweiterungsbus) zum Umschalten in den Modus "Initialization" zu forcieren.

```
LD 1
[%MW0 := 16#0001]
[%MW1 := 16#0001]
LD %SW81:X3          // Wenn keine CAN_CMD-Anweisung ausgeführt wird, wird folgendermaßen fortgesetzt
[CAN_CMD1 %MW0:2]    // Um den CANopen-Master zum Umschalten in den Modus "Initialization" zu forcieren
LD %SW81:X4          // (optional) Um vor dem Senden einer neuen Anweisung zu ermitteln, ob die Anweisung CAN_CMD erfolgreich ausgeführt wurde.
```

Beispiel 2:
Um die folgende Variable zu lesen: SDO_Slave:1_index:24576_sub-index:1_length:4

```
LD 1
[%MW6 := %MW4]       // Das Ergebnis des letzten SDO-Befehls speichern
[%MW7 := %MW5]       // Das Ergebnis des letzten SDO-Befehls speichern

LD %SW81:X3  // Wenn keine CAN_CMD-Anweisung ausgeführt wird, wird folgendermaßen fortgesetzt
[%MW0 := 16#0003]
[%MW1 := 16#0001]    // SDO an Adressknoten 1 lesen
[%MW2 := 16#6000]    // Zugriff auf Indexnummer 24576
[%MW3 := 16#0104]    // Zugriff auf Subindexnummer 1 und Längenwert 4
[CAN_CMD1 %MW0:6] // SDO-Befehl starten
```

Beispiel 3:
Um die folgende Variable zu schreiben: SDO_Slave:1_index:24576_sub-index:1_length:4

```
LD 1
[%MW0 := 16#0004]
[%MW1 := 16#0001]    // SDO an Adressknoten 1 schreiben
[%MW2 := 16#6000]    // Zugriff auf Indexnummer 24576
[%MW3 := 16#0104]    // Zugriff auf Subindexnummer 1 und Längenwert 4
[%MW4 := 16#1234]    // Datenwert 1
[%MW5 := 16#1234]    // Datenwert 2
LD %SW81:X3  // Wenn keine CAN_CMD-Anweisung ausgeführt wird, wird folgendermaßen fortgesetzt
[CAN_CMD1 %MW0:6] // SDO-Befehl starten
```

Konfigurieren des TwidoPort-Ethernet-Gateways

11

Auf einen Blick

Inhalt dieses Kapitels

Dieses Kapitel beschreibt die Softwarekonfiguration des ConneXium TwidoPort-Ethernet-Gateway-Moduls.

Inhalt dieses Kapitels

Dieses Kapitel enthält die folgenden Abschnitte:

Abschnitt	Thema	Seite
11.1	Normale Konfiguration und Anschluss von TwidoPort	289
11.2	TwidoPort-Konfiguration mittels Telnet	296
11.3	Kommunikationsfunktionen	311

11.1 Normale Konfiguration und Anschluss von TwidoPort

Auf einen Blick

Inhalt dieses Abschnitts

Dieser Abschnitt enthält Informationen über die Durchführung einer normalen Konfiguration des ConneXium TwidoPort-Moduls mittels des TwidoSoft-Applikationsprogramns, über den Anschluss des Moduls und über die BootP-Konfiguration.

Inhalt dieses Abschnitts

Dieser Abschnitt enthält die folgenden Themen:

Thema	Seite
Normale Konfiguration mit TwidoSoft	290
BootP-Konfiguration	295

Normale Konfiguration mit TwidoSoft

Vorwort

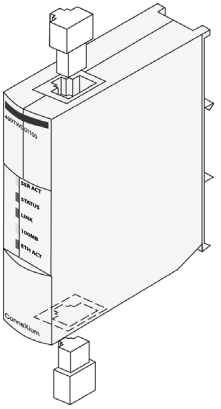
Wenn Sie über TwidoSoft (v. 3.0 oder höher) verfügen, dann konfigurieren Sie TwidoPort gemäß den nachfolgend aufgeführten Anweisungen.

> **Hinweis: Plug & Play-Funktion**
> Wenn TwidoPort mit TwidoSoft konfiguriert wird, wird die IP-Konfiguration von TwidoPort in der Twido-Steuerung gespeichert. Daher kann das Wartungspersonal TwidoPorts ohne zusätzliche Konfiguration austauschen.
> Um die Plug & Play-Funktion zu nutzen, verwenden Sie TwidoSoft ab Version 3.0, und aktualisieren Sie auf die Twido Firmware auf die Version 3.0 oder höher.
> Verwenden Sie Telnet, um TwidoPort manuell mit älteren TwidoSoft-Versionen zu konfigurieren.

Installieren des TwidoPort-Moduls 499TWD01100

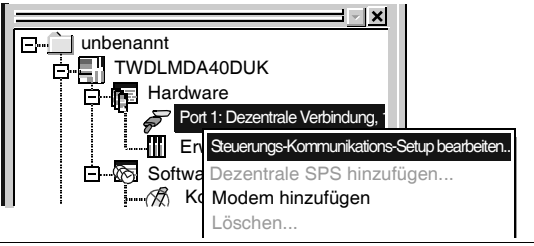
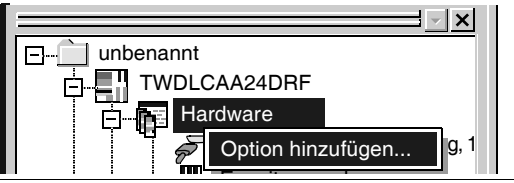
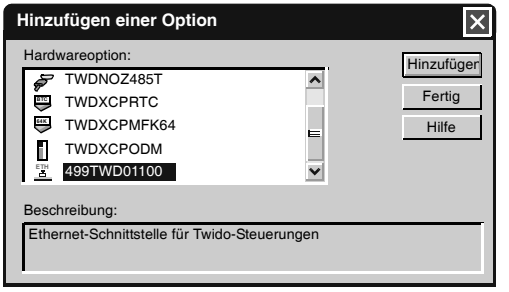
Um TwidoPort in einem Twido-Steuerungssystem (DIN-Schienen- oder Schaltpultmontage) zu installieren, schließen Sie es an den internen Bus der Twido-Steuerung an. Gehen Sie hierzu wie folgt vor:

Schritt	Beschreibung	Aktion
1	Vorbereitung der Installation	Anweisungen zu den folgenden Themen finden Sie im *Twido-SPS-Hardware-Referenzhandbuch (TWD USE 10AE)*: ● korrekte Montagepositionen für Twido-Module, ● Hinzufügen und Entfernen von Twido-Komponenten zu bzw. von einer DIN-Schiene, ● direkte Montage auf einer Schaltpultoberfläche, ● Mindestabstände für Module in einem Schaltpult.
2	Montage des TwidoPort-Moduls 499TWD01100	Installieren Sie das Modul auf einer DIN-Schiene oder in einem Schaltpult. Ausführliche Informationen finden Sie unter (Siehe TwdoHW Hardware-Referenzhandbuch).
3	Anschluss an die Schutzerde (Protective Earth, PE)	Schließen Sie einen geerdeten Draht an die M3-Schraubenklemme unten am TwidoPort-Modul an.

Schritt	Beschreibung	Aktion
4	Serielle und Ethernet-Verbindungen **Oberer Stecker: von Twido (seriell)**  **Unterer Stecker: von Ethernet, entweder ein durchgehendes Kabel oder ein Kreuzkabel**	Schließen Sie das modulare Steckerende des (mitgelieferten) TwidoPort-zu-Twido-Kabels an den seriellen Port des TwidoPort-Moduls an, und schließen Sie das andere Ende an den seriellen RS-485-Port der Twido-SPS an. Schließen Sie den RJ-45-Stecker von einem standardmäßigen Ethernet-Netzwerkkabel (nicht mitgeliefert) am Ethernet-Port des TwidoPort-Moduls an.

Deklarieren des TwidoPort-Moduls 499TWD01100

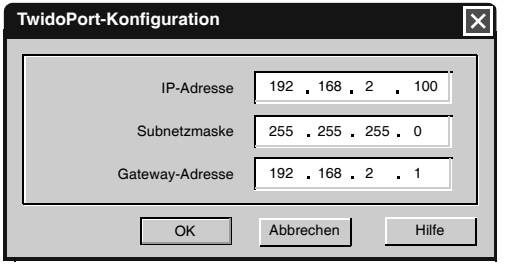
Die Tabelle unten zeigt die verschiedenen Schritte der Deklaration des TwidoPort-Moduls 499TWD01100.

Schritt	Aktion	Kommentar
1	Konfigurieren Sie bei Verwendung von TwidoSoft (ab Version 3.0) die Kommunikationsoptionen der Twido-Steuerung durch einen Rechtsklick auf **Port 1: xxxxxx , 1 → Steuerungs-Kommunikations-Setup bearbeiten...** im TwidoSoft-Applikationsnavigator (siehe Hinweis 1).	
2	Setzen Sie im Dialogfeld **SPS-Kommunikations-Setup** das Kommunikationsprotokoll auf **Modbus.**	Beachten Sie, dass das TwidoPort-Modul 499TWD01100 nicht zur Twido-Hardware hinzugefügt werden kann, wenn das Kommunikationsprotokoll nicht auf "Modbus" gesetzt ist.
3	Konfigurieren Sie die Modbus-Kommunikationsparameter.	Der RS-485-Modbus-Port der Twido-Steuerung muss auf **9600, 19200 oder 38400 Baud** gesetzt sein, um die Autobaud-Funktion des TwidoPort-Moduls zu unterstützten (siehe Hinweise 1 und 2).
4	Klicken Sie im TwidoSoft-Applikationsnavigator mit der rechten Maustaste auf **Hardware →** **Option hinzufügen...**.	
5	Wenn das Dialogfeld **Option hinzufügen** angezeigt wird: • Wählen Sie **499TWD01100** aus. • Klicken Sie auf **Hinzufügen.** • Jetzt können Sie jedes beliebige andere, von Ihrer Twido-Steuerung unterstützte optionale Modul hinzufügen. **Hinweis:** Es ist nur ein TwidoPort-Modul 499TWD01100 zulässig. • Klicken Sie auf **Fertig.**	Alle Twido-Steuerungen werden unterstützt (mit Ausnahme des TWDLCAE40DRF mit integriertem Ethernet). 
Hinweis 1	**Jeder RS-485-Modbus-Port am Twido-Modul kann verwendet werden.**	
Hinweis 2	**Wählen Sie für die schnellste Autobaudraten-Initialeinstellung 19200-8-N-1 mit der Twido Modbus-Adresse 1.**	

Konfigurieren des TwidoPort-Moduls 499TWD01100

Hinweis: Die Ethernet-Parameter des TwidoPort-Moduls können nur konfiguriert werden, wenn sich das TwidoSoft-Applikationsprogramm im Offline-Modus befindet.

Gehen Sie folgendermaßen vor, um die Ethernet-Parameter des TwidoPort-Moduls zu konfigurieren:

Schritt	Aktion	Kommentar
Vorwort	Weitere Informationen über die IP-Parameter (IP-Adresse, Subnetzmaske und Gateway-Adresse) finden Sie unter *IP-Adressierung, S. 169* und *Private IP-Adressen, S. 173*.	
1	Klicken Sie mit der rechten Maustaste auf das TwidoPort-Symbol im Applikationsnavigator, um die IP-Parameter des TwidoPort-Moduls zu konfigurieren.	**Ergebnis:** Das Dialogfeld **Ethernet-Konfiguration** wird angezeigt (siehe Beispiel unten). 
2	Geben Sie die statische **IP-Adresse** Ihres TwidoPort-Moduls in Dezimalpunktschreibweise ein (siehe *Hinweise 1 und 2*).	**Vorsicht:** Um die ordnungsgemäße Kommunikation des Geräts zu gewährleisten, müssen die IP-Adressen des PC, auf dem die TwidoSoft-Applikation ausgeführt wird, und das TwidoPort-Modul dieselbe Netzwerk-ID aufweisen.
Hinweis 1	**Wenden Sie sich an Ihren Netzwerk- oder Systemadministrator, um gültige IP-Parameter für Ihr Netzwerk zu erhalten.**	
Hinweis 2	**Um eine gute Kommunikation über das Netzwerk zu ermöglichen, muss jedes angeschlossene Gerät über eine eindeutige IP-Adresse verfügen. Wenn das TwidoPort-Modul mit dem Netzwerk verbunden ist, führt es eine Überprüfung auf doppelte IP-Adressen durch. Wenn eine doppelte IP-Adresse im Netzwerk erkannt wird, blinkt die STATUS-LED regelmäßig 4 Mal auf. In diesem Fall müssen Sie eine neue eindeutige IP-Adresse in dieses Feld eingeben.**	
Hinweis 3	**Verwenden Sie die Standard-Subnetzmaske, es sei denn, Ihr TwidoPort-Modul erfordert eine spezielle Subnetzmasken-Zuweisung.**	
Hinweis 4	**Wenn sich kein Gateway-Gerät in Ihrem Netzwerk befindet, geben Sie einfach die IP-Adresse Ihres TwidoPort-Moduls in das Feld "Gateway-Adresse" ein.**	

Schritt	Aktion	Kommentar
3	Geben Sie die gültige, Ihrem TwidoPort-Modul vom Netzwerkadministrator zugewiesene **Subnetzmaske** ein. Bitte beachten Sie, dass Sie dieses Feld nicht leer lassen können. Sie müssen einen Wert eingeben (siehe *Hinweise 1 und 3*).	**Vorsicht:** Um die ordnungsgemäße Kommunikation des Geräts zu gewährleisten, müssen die auf dem PC, auf dem die TwidoSoft-Applikation ausgeführt wird, konfigurierte Subnetzmaske und die Subnetzmaske des TwidoPort-Moduls übereinstimmen. Standardmäßig berechnet die TwidoSoft-Applikation eine Standard-Subnetzmaske auf der Grundlage der von Ihnen oben im Feld "IP-Adresse" angegebenen IP-Adresse und zeigt diese Subnetzmaske an. Die der Kategorie der TwidoPort-Netzwerk-IP-Adresse entsprechenden standardmäßigen Maskenwerte entsprechen folgender Regel: Netzwerk der Klasse A -> Standard-Subnetzmaske: 255.0.0.0 Netzwerk der Klasse B -> Standard-Subnetzmaske: 255.255.0.0 Netzwerk der Klasse C -> Standard-Subnetzmaske: 255.255.255.0
4	Geben Sie die IP-Adresse des **Gateways** an (siehe *Hinweise 1 und 4*).	Im LAN muss sich das Gateway in demselben Segment wie Ihr TwidoPort-Modul befinden. Diese Information erhalten Sie üblicherweise von Ihrem Netzwerkadministrator. Bitte beachten Sie, dass kein Standardwert von der Applikation geliefert wird und dass Sie eine gültige Gateway-Adresse in dieses Feld eingeben müssen.
5	Bestätigen Sie die Konfiguration und laden Sie sie in die Twido-Steuerung.	
Hinweis 1	**Wenden Sie sich an Ihren Netzwerk- oder Systemadministrator, um gültige IP-Parameter für Ihr Netzwerk zu erhalten.**	
Hinweis 2	**Um eine gute Kommunikation über das Netzwerk zu ermöglichen, muss jedes angeschlossene Gerät über eine eindeutige IP-Adresse verfügen. Wenn das TwidoPort-Modul mit dem Netzwerk verbunden ist, führt es eine Überprüfung auf doppelte IP-Adressen durch. Wenn eine doppelte IP-Adresse im Netzwerk erkannt wird, blinkt die STATUS-LED regelmäßig 4 Mal auf. In diesem Fall müssen Sie eine neue eindeutige IP-Adresse in dieses Feld eingeben.**	
Hinweis 3	**Verwenden Sie die Standard-Subnetzmaske, es sei denn, Ihr TwidoPort-Modul erfordert eine spezielle Subnetzmasken-Zuweisung.**	
Hinweis 4	**Wenn sich kein Gateway-Gerät in Ihrem Netzwerk befindet, geben Sie einfach die IP-Adresse Ihres TwidoPort-Moduls in das Feld "Gateway-Adresse" ein.**	

BootP-Konfiguration

BootP-Verfahren

TwidoPort erwartet innerhalb von zwei Minuten nach der Übertragung seines BootP-Requests eine Antwort vom BootP-Server. Erhält TwidoPort innerhalb dieser Zeit keine Antwort, so nimmt das Modul die Standard-IP-Konfiguration an, die von einer MAC-Adresse der folgenden Struktur abgeleitet wird:

85	16	MAC[4]	MAC[5]

MAC-Adresse

Die MAC-Adresse hat die folgende Struktur:
`MAC[0] MAC[1] MAC[2] MAC[3] MAC[4] MAC[5].`

Wenn beispielsweise die MAC-Adresse `0080F4012C71` lautet, dann lautet die Standard-IP-Adresse `85.16.44.113`.

11.2 TwidoPort-Konfiguration mittels Telnet

Auf einen Blick

Inhalt dieses Abschnitts

Dieser Abschnitt beschreibt die Konfiguration des ConneXium TwidoPort-Moduls mittels einer Tenet-Sitzung.

Inhalt dieses Abschnitts

Dieser Abschnitt enthält die folgenden Themen:

Thema	Seite
Einführung in die Telnet-Konfiguration	297
Telnet-Hauptmenü	298
IP/Ethernet-Einstellungen	299
Konfiguration der seriellen Parameter	301
Konfiguration des Gateways	302
Sicherheitskonfiguration	303
Ethernet-Statistiken	304
Serielle Statistiken	305
Speichern der Konfiguration	306
Wiederherstellen der Standardeinstellungen	307
Upgraden der TwidoPort-Firmware	308
Haben Sie Ihr Passwort und/Oder Ihre IP-Konfiguration vergessen?	310

Einführung in die Telnet-Konfiguration

Überblick über die Telnet-Konfiguration

Konfigurieren Sie TwidoPort mittels einer Telnet-Sitzung (unter Verwendung eines VT100-kompatiblen Telnet-Clients), wenn keine spezifische Twido-Konfiguration gefunden werden konnte oder wenn der BootP-Request nicht innerhalb von zwei Minuten beantwortet wird (was zur Implementierung der Standard-IP-Adresse führt).

Vorbereitung der Telnet-Konfiguration

Hinweis: Telenet-Anforderungen von TwidoPort
Vergewissern Sie sich für die Konfiguration von TwidoPort mittels Telnet, dass:
- TwidoPort über seine serielle Verbindung (über eine Twido-Steuerung) stromversorgt ist.
- Telnets **`lokales Echo`** auf **`aus`** gesetzt ist.

Um Telnet zu verwenden, fügen Sie TwidoPorts Standard-IP-Adresse (oder TwidoPorts konfigurierte IP-Adresse) mittels folgendem Befehl zur Routing-Tabelle des PC hinzu:

```
C:\> route add 85.0.0.0 mask 255.0.0.0 local_IP_address_of_PC
```

Beispiel:
Wenn die IP-Adresse des PC `192.168.10.30` lautet und die Standard-IP-Adresse (oder die konfigurierte IP-Adresse) von TwidoPort `85.16.44.113` ist, dann lautet der vollständige Befehl:

```
C:\> route add 85.0.0.0 mask 255.0.0.0 192.168.10.30
```

Telnet-Hauptmenü

Starten des Telnet-Hauptmenüs

Wenn Sie eine Telnet-Sitzung starten (z.B. durch Eingabe von `telnet 85.16.44.113` in der Eingabeaufforderung oder mittels Windows™ Hyperterminal™), dann wird das Telnet-Hauptmenü angezeigt, wenn Sie die **`Eingabetaste`** drücken.

```
        Telemecanique 499 TWD 01 100 Configuration and Diagnostics
                (c) 2004 Schneider Automation Inc

1) IP/Ethernet Settings
                    IP Source: DEFAULT
                IP Address: 85.16.44.113
            Default Gateway: 85.16.44.113
                    Netmask: 0.0.0.0
        Ethernet Frame Type: ETHERNETII

2) Serial Configuration
        Baud Rate: 19200
        Data Bits: 8
           Parity:  NONE
        Stop Bits: 1
         Protocol:  RTU

3) Gateway Configuration
          Slave Address Source: UNIT_ID
              Gateway Mode: SLAVE
                MB Broadcasts: ENABLED

4) Security Configuration

5) Ethernet Statistics

6) Serial Statistics

Commands: D)efault settings, S)ave, F)irmware Upgrade, Q)uit without save
     Select Command or Parameter(1..6) to change:
```

IP/Ethernet-Einstellungen

Konfigurieren der IP/Ethernet-Einstellungen

Gehen Sie folgendermaßen vor, um die IP/Ethernet-Einstellungen zu ändern:

Schritt	Aktion	Kommentar
1	Starten Sie eine Telnet-Sitzung.	Gehen Sie wie oben beschrieben vor, um das Telnet-Hauptmenü (Siehe *Telnet-Hauptmenü, S. 298*) aufzurufen.
2	Wählen Sie **1**, um die IP-Quelle in **GESPEICHERT** zu ändern, und drücken Sie anschließend die ***Eingabetaste.***	GESPEICHERT kann bereits die IP-Quelle sein.
3	Legen Sie die gewünschten IP-Parameter manuell fest (siehe *TwidoPort-Ethernet-Einstellungen* nach dieser Tabelle).	Zu den sonstigen Parametern gehören: • IP-Adresse • Standard-Gateway • Netzmaske • Ethernet-Frame-Typ
4	Wählen Sie **R** aus, und drücken Sie die ***Eingabetaste.***	Das Telnet-Hauptmenü wird angezeigt. (Eventuell müssen Sie erneut die ***Eingabetaste*** drücken, um den Bildschirm zu aktualisieren.)

IP Source

Die Option **IP-Quelle** bestimmt, wo die IP-Konfiguration abgefragt wird:

- GESPEICHERT - vom lokalen Flash-Speicher.
- GELIEFERT - vom BootP-Server.
- TWIDO - von der Twido-Steuerung.

Die Standard-IP-Adresse (STANDARD) wird von der MAC-Adresse abgeleitet. (Per Definition kann die Standardadresse nicht ausgewählt werden.)

Hinweis: Eine gültige IP-Konfiguration in der Twido-Steuerung hat Vorrang vor der Auswahl des Benutzers.

Beispiel für die Ethernet-Einstellung

Die folgende Abbildung zeigt ein Beispiel für die TwidoPort-Ethernet-Einstellungen.

```
Telemecanique 499 TWD 01 100 Configuration and Diagnost
        (c) 2004 Schneider Automation Inc

IP/Ethernet Settings
-----------------------------------------------------

        1)IP Source: DEFAULT
        2)IP Address: 85.16.44.113
        3)Default Gateway: 85.16.44.113
        4)Netmask: 0.0.0.0
        5)Ethernet Frame Type: ETHERNET2
-----------------------------------------------------
Commands: R)eturn to Main Menu
Select Command or Parameter(1..N) to change:
```

Konfiguration der seriellen Parameter

Vorwort

Hinweis: Unter normalen Bedingungen ist es nicht erforderlich, die seriellen Parameter von TwidoPort zu konfigurieren, da das Modul einen Autobaud-Algorithmus unterstützt, der eine serielle Konfiguration vermeidet.

Konfigurieren der seriellen Parameter

Gehen Sie folgendermaßen vor, um die seriellen Parameter von TwidoPort zu konfigurieren:

Schritt	Aktion	Kommentar
1	Starten Sie eine Telnet-Sitzung.	Gehen Sie wie oben beschrieben vor, um das Telnet-Hauptmenü (Siehe *Telnet-Hauptmenü, S. 298*) aufzurufen.
2	Wählen Sie **2** aus, um die seriellen Einstellungen zu ändern.	Siehe hierzu folgende Abbildung.
3	Überprüfen Sie die Einstellungen oder setzen Sie sie zurück.	Zu den sonstigen Parametern gehören: • Baudrate • Datenbits • Parität • Stoppbits • Protokoll
4	Wählen Sie **R** aus, und drücken Sie die ***Eingabetaste.***	Das Telnet-Hauptmenü wird angezeigt. (Eventuell müssen Sie erneut die ***Eingabetaste*** drücken, um den Bildschirm zu aktualisieren.)

Beispiel für die seriellen Einstellungen

Die folgende Abbildung zeigt ein Beispiel für die seriellen Einstellungen von TwidoPort.

```
Telemecanique 499 TWD 01 100 Configuration and Diagnostics
        (c) 2004 Schneider Automation Inc

 Serial Configuration
        1) Baud Rate:  19200
        2) Data Bits:  8
        3)    Parity:  NONE
        4) Stop Bits:  1
            Protocol:  RTU

Commands: R)eturn to Main Menu
Select Command or Parameter(1..N) to change:
```

Konfiguration des Gateways

Vorwort

Hinweis: Normalerweise ist es nicht erforderlich, die Gateway-Parameter von TwidoPort zu konfigurieren.

Konfigurieren der Gateway-Parameter

Gehen Sie folgendermaßen vor, um das Gateway von TwidoPort zu konfigurieren:

Schritt	Aktion	Kommentar	
1	Starten Sie eine Telnet-Sitzung.	Gehen Sie wie oben beschrieben vor, um das Telnet-Hauptmenü (Siehe *Telnet-Hauptmenü, S. 298*) aufzurufen.	
2	Wählen Sie `3` aus, um die Gateway-Parameter zu ändern.	Siehe hierzu folgende Abbildung.	
3	Die folgenden Gateway-Parameter sind verfügbar:		
	(1) Quelle der Slave-Adresse	`FEST:`	Wenn die Quelle der Slave-Adresse **`FEST`** ist, setzen Sie die Adresse auf den Wert der Modubus-Adresse der Twido-Steuerung. Die gültigen Adressen liegen im Bereich von 1 bis 247.
		`UNIT_ID`	Die Geräte-ID des Modbus/TCP-Frames wird verwendet.
	(2) Gateway-Modus	`SLAVE`	Einzige Option für diese Version.
	(3) MB-Broadcasts	`DEAKTIVIERT`	Es werden keine Broadcast-Meldungen über den seriellen Port des TwidoPort-Moduls gesendet.
		`AKTIVIERT`	Es werden Broadcast-Meldungen vom seriellen Port des TwidoPort-Moduls aus gesendet *(siehe Hinweis unten).*
4	Wählen Sie `R` aus, und drücken Sie die ***`Eingabetaste.`***	Das Telnet-Hauptmenü wird angezeigt. (Eventuell müssen Sie erneut die ***`Eingabetaste`*** drücken, um den Bildschirm zu aktualisieren.)	
Hinweis	**Twido unterstützt keine Modbus-Broadcast-Meldungen.**		

Beispiel für die Gateway-Einstellungen

Die folgende Abbildung zeigt ein Beispiel für die Gateway-Einstellungen von TwidoPort.

```
Telemecanique 499 TWD 01 100 Configuration and Diagnostics
        (c) 2004 Schneider Automation Inc

Gateway Configuration
        1) Slave Address Source: UNIT_ID
        2) Slave Address: 20
        3) Gateway Mode: SLAVE
        4) MB Broadcasts: ENABLED

Commands: R)eturn to Main Menu
Select Command or Parameter(1..4) to change: _
```

Sicherheitskonfiguration

Konfigurieren der Sicherheitseinstellungen

Gehen Sie folgendermaßen vor, um das Standard-Passwort zu ändern:

Schritt	Aktion	Kommentar
1	Starten Sie eine Telnet-Sitzung.	Gehen Sie wie oben beschrieben vor, um das Telnet-Hauptmenü (Siehe *Telnet-Hauptmenü, S. 298*) aufzurufen.
2	Wählen Sie **4** aus, und drücken Sie die ***Eingabetaste.***	Das Fenster "Sicherheitskonfiguration" wird angezeigt.
3	Wählen Sie **C** aus, und drücken Sie die ***Eingabetaste.***	
4	Geben Sie das alte Passwort ein.	Befugte Benutzer kennen das alte Passwort (die Standardeinstellung ist **USERUSER**).
5	Geben Sie ein neues Passwort ein.	Geben Sie das neue Passwort erneut ein *(siehe Hinweis unten).*
6	Geben Sie das neue Passwort erneut ein.	Beachten Sie die Hinweise für zulässige Passwörter.
7	Wählen Sie **R** aus, und drücken Sie die ***Eingabetaste.***	Das Telnet-Hauptmenü wird angezeigt. (Eventuell müssen Sie erneut die ***Eingabetaste*** drücken, um den Bildschirm zu aktualisieren.)
Hinweis	**Hinweise zu den Passwörtern:** • **minimale Länge: 4 Zeichen** • **maximale Länge: 10 Zeichen** • **zulässige Zeichen: 0 bis 9, a bis z, A bis Z (keine Leerzeichen)**	

Ethernet-Statistiken

Anzeigen der Ethernet-Statistiken

Gehen Sie folgendermaßen vor, um die Ethernet-Statistiken von TwidoPort anzuzeigen:

Schritt	Aktion	Kommentar
1	Starten Sie eine Telnet-Sitzung.	Gehen Sie wie oben beschrieben vor, um das Telnet-Hauptmenü (Siehe *Telnet-Hauptmenü, S. 298*) aufzurufen.
2	Wählen Sie `5` aus, um den Bildschirm **`Ethernet Module Statistics`** anzuzeigen.	Siehe Abbildung nach dieser Tabelle.
3	Drücken Sie die **`Eingabetaste`**, um den Bildschirm zu aktualisieren.	
4	Drücken Sie `C`, um die Statistiken zu löschen, und drücken Sie dann die ***`Eingabetaste`***.	Alle Zähler werden auf 0 zurückgesetzt.
5	Wählen Sie `R` aus, und drücken Sie die ***`Eingabetaste.`***	Das Telnet-Hauptmenü wird angezeigt. (Eventuell müssen Sie erneut die ***`Eingabetaste`*** drücken, um den Bildschirm zu aktualisieren.)

Der Bildschirm Ethernet Module Statistics

TwidoPort-Bildschirm **`Ethernet Module Statistics`**:

```
          Telemecanique 499 TWD 01 100 Configuration and Diagnostics
                  (c) 2004 Schneider Automation Inc
                  ETHERNET MODULE STATISTICS
-------------------------------------------------------------------------
Status: 0x9103                          IP Address: 192.168.1.141
System Log Entry: No                    Mac Address: 0:80:f4:0:4c:18
Transmit Speed: 100BASE-T               Subnet Mask: 255.255.0.0
Full/Half Duplex: Half Duplex           Gateway Address: 192.168.1.1
-------------------------------------------------------------------------
Transmit Statistics     Receive Statistics      Functioning Errors

Transmits: 63           Receives: 532            Missed Packets: 0
Transmit Retries: 0     Framing Errors: 0        Collision Errors: 0
Lost Carrier: 0         Overflow Errors: 0       Transmit Timeouts: 0
Late Collision: 0       CRC Errors: 0            Memory Errors: 0
Tx Buffer Errors: 0     Rx Buffer Errors: 0      Net Interface Restarts: 0
SILO Underflow: 0
-------------------------------------------------------------------------
Broadcast Packets Received: 37          Multicast Packets Received: 7
-------------------------------------------------------------------------
Commands: [Enter] to Refresh, C)lear Statistics, R)eturn to Main Menu
```

Serielle Statistiken

Anzeigen der seriellen Statistiken

Gehen Sie folgendermaßen vor, um die seriellen Statistiken von TwidoPort anzuzeigen:

Schritt	Aktion	Kommentar
1	Starten Sie eine Telnet-Sitzung.	Gehen Sie wie oben beschrieben vor, um das Telnet-Hauptmenü (Siehe *Telnet-Hauptmenü, S. 298*) aufzurufen.
2	Wählen Sie **6** aus, um den Bildschirm **Serial Statistics** anzuzeigen, und drücken Sie dann die **Eingabetaste**.	Siehe Abbildung nach dieser Tabelle. Die seriellen Statistiken werden aktualisiert.
3	Drücken Sie **C**, um die Statistiken zu löschen, und drücken Sie dann die ***Eingabetaste***.	Alle Zähler werden auf 0 zurückgesetzt.
4	Wählen Sie **R** aus, und drücken Sie die ***Eingabetaste.***	Das Telnet-Hauptmenü wird angezeigt. (Eventuell müssen Sie erneut die ***Eingabetaste*** drücken, um den Bildschirm zu aktualisieren.)

Der Bildschirm Serial Statistics

TwidoPort-Bildschirm **Serial Statistics**:

```
                Telemecanique 499 TWD 01 100 Configuration and Diagnostics
                        (c) 2004 Schneider Automation Inc

------------------------ SERIAL STATISTICS --------------------------------

Serial Bus Statistics
        Bus Message Count: 8284
        Bus Comm. Error Count: 0
Modbus Slave Statistics
        Slave Message Count: 4142
        Slave Exception Error Count: 3187
        Slave No Response Count: 0

---------------------------------------------------------------------------

Commands: [Enter] to Refresh, C)lear Statistics, R)eturn to Main Menu
```

Speichern der Konfiguration

Speichern der Konfiguration

Um die Änderungen an der Konfiguration zu speichern, geben Sie das Konfigurationspasswort ein:

Schritt	Aktion	Kommentar
1	Starten Sie eine Telnet-Sitzung.	Gehen Sie wie oben beschrieben vor, um das Telnet-Hauptmenü (Siehe *Telnet-Hauptmenü, S. 298*) aufzurufen.
2	Wählen Sie `S` aus, und drücken Sie die ***`Eingabetaste.`***	
3	Geben Sie das Konfigurationspasswort ein.	Das Standardpasswort ist **`USERUSER`** *(siehe Hinweis unten).*
Hinweis	**Ausführliche Informationen über die Einrichtung eines individuellen Sicherheitspassworts finden Sie unter *Sicherheitskonfiguration, S. 303*.**	

Das Bestätigungs-fenster Save Configuration

Das Bestätigungsfenster `Save Configuration` von TwidoPort:

```
                Telemecanique 499 TWD 01 100 Configuration and Diagnostics
                        (c) 2004 Schneider Automation Inc
                        SAVE CONFIGURATION
--------------------------------------------------------------------------------

        Configuration successfully stored to Twido.
        Reboot your module for the new Configuration to be in effect.

        Rebooting in 5 Seconds. You will lose your telnet connection.

Connection to host lost.
```

Wiederherstellen der Standardeinstellungen

Wiederherstellen der Standardeinstellungen

Gehen Sie folgendermaßen vor, um die Standardeinstellungen von TwidoPort wiederherzustellen:

Schritt	Aktion	Kommentar
1	Starten Sie eine Telnet-Sitzung.	Gehen Sie wie oben beschrieben vor, um das Telnet-Hauptmenü (Siehe *Telnet-Hauptmenü, S. 298*) aufzurufen.
2	Wählen Sie **D** aus, um das Fenster **Default Configuration** anzuzeigen.	Siehe Abbildung nach dieser Tabelle.
3	Drücken Sie die **Eingabetaste.**	Drücken Sie die **Eingabetaste.** ist erforderlich, um das Hauptmenü anzuzeigen.
4	Speichern Sie die Standardkonfiguration.	Siehe Speichern der Konfiguration (Siehe *Speichern der Konfiguration, S. 306*) oben.

Das Fenster Default Configuration

Das Fenster **Default Configuration** von TwidoPort:

```
          Telemecanique 499 TWD 01 100 Configuration and Diagnostics
                  (c) 2004 Schneider Automation Inc
                  DEFAULT CONFIGURATION
-------------------------------------------------------------------------------

IP Address: 192.168.2.102
Gateway Address: 192.168.2.102
Subnet Mask: 255.255.0.0
Frame Type: Ethernet II

Serial Mode: 19200-8-N-1

Gateway Mode: Modbus/RTU Slave Attached
              Broadcasts Disabled, Slave Address Source=Unit ID

Configuration Password: USERUSER

You must (S)ave the configuration to make it active.

Returning to Main Menu in 2 Seconds, Hit Enter to refresh._
```

Upgraden der TwidoPort-Firmware

Vorwort

Hinweis:
1. Beschaffen Sie sich eine neuere Version der TwidoPort-Firmware, bevor Sie versuchen, die Firmware gemäß den nachfolgend aufgeführten Anweisungen zu aktualisieren.
2. Stoppen Sie den Prozess, bevor Sie die Firmware upgraden.
3. Die Modbus-Kommunikation ist während des Firmware-Upgrades nicht verfügbar.

Upgraden der Firmware

Gehen Sie folgendermaßen vor, um die aktuelle Firmware von TwidoPort auf die neueste Firmware-Version zu aktualisieren, die Sie erhalten haben:

Schritt	Aktion	Kommentar
1	Starten Sie eine Telnet-Sitzung.	Gehen Sie wie oben beschrieben vor, um das Telnet-Hauptmenü (Siehe *Telnet-Hauptmenü, S. 298*) aufzurufen.
2	Wählen Sie **F**, um das Firmware-Upgrade zu starten.	Fünf Sekunden nach der Auswahl von **F** (Firmware-Upgrade) wird TwidoPort zurückgesetzt, und die Telnet-Verbindung wird unterbrochen.
3	Geben Sie in die Eingabeaufforderung Folgendes ein: **ftp** und die IP-Adresse von TwidoPort.	Beispiel: **ftp 85.16.44.113**
4	Geben Sie **ftptwd**	an der Login-Namenseingabe ein.
5	Geben Sie Folgendes ein: **cd fw**	Hierdurch gelangt der Benutzer zum Verzeichnis **fw**.
6	Geben Sie Folgendes ein: **put App.out** . *(siehe Hinweise 1 und 2).*	Eine Meldung zeigt an, dass der ftp erfolgreich war *(siehe Hinweis 3).*
Hinweis 1	**Bei Dateinamen wird zwischen Groß- und Kleinschreibung unterschieden.**	
Hinweis 2	**Vergewissern Sie sich, dass App.out das aktuelle Arbeitsverzeichnis des ftp-Client ist.**	
Hinweis 3	**Eine Meldung zeigt an, dass TwidoPort 5 Sekunden nach einem erfolgreichen ftp automatisch neu gestartet wird.**	

Firmware-Upgrade wird ausgeführt

Die folgende Abbildung zeigt einen typischen Bildschirm, der während des **Firmware-Upgrades** angezeigt wird.

```
        Telemecanique 499 TWD 01 100 Configuration and Diagnostics
        ---------------------------------------------------

        FIRMWARE UPGRADE IN-PROGRESS...
        Module will reboot in 5 Seconds.
        After Reboot, Connect via FTP to download new Firmware.

        FTP Instructions:
                1) Connect via FTP:  ftp  192.168.2.160
                2) Change to /fw directory:  ftp>cd fw
                3) Download new fw:  ftp>put App.out

        After the FTP download is complete, the module will reboot automatically
.

        Rebooting now.  Goodbye.


Connection to host lost.
```

Kernel-Modus

Wenn keine gültige Firmware vorhanden ist, geht TwidoPort in den **Kernel**-Modus über. Wen Sie versuchen, Telnet zu verwenden, um eine Verbindung zu TwidoPort aufzubauen, während sich das Modul in diesem Modus befindet, wird Folgendes angezeigt:

```
Telemecanique 499 TWD 01 100
-----------------------------------------------
Kernel Version 90.02d

Download valid Exec,App.out, to leave kernel mode.

   To exit type 'quit' 'QUIT' or control D
```

Haben Sie Ihr Passwort und/Oder Ihre IP-Konfiguration vergessen?

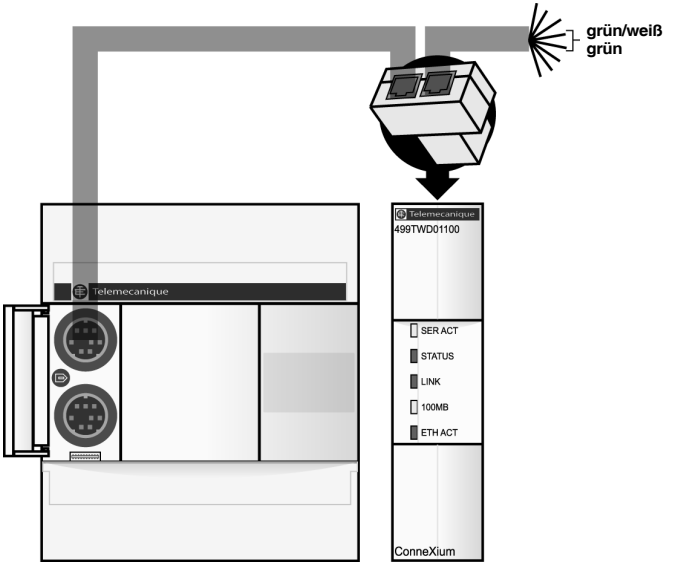
Verbinden im Backup-Modus

Gehen Sie folgendermaßen vor, um im Backup-Modus eine Verbindung zu TwidoPort aufzunehmen:

Schritt	Aktion	Kommentar
1	Verbinden Sie Pin 3 bis Pin 6 (Erde) des seriellen Steckverbinders.	Verwenden Sie den T-förmigen RJ-45-Steckverbinder 170 XTS 04 100 von Schneider *(siehe folgende Abbildung).*
2	Bauen Sie eine ftp-Verbindung zu TwidoPort auf *(siehe Hinweis).*	TwidoPort verwendet die folgende Standard-IP-Konfiguration: • IP-Adresse: 192.168.2.102 • Subnetzmaske: 255.255.0.0 • Gateway-Adresse: 192.168.2.102 • Frame-Typ: Ethernet II
3	Laden Sie die Datei **fw/Conf.dat.**	Fragen Sie die IP-Konfiguration und das Passwort aus der Datei **Conf.dat** ab.
4	Öffnen Sie die Datei **Conf.dat** in einem Texteditor.	
Hinweis	**Hierzu ist kein Passwort erforderlich.**	

FTP-Verbindung

Die folgende Abbildung zeigt, wie Sie eine ftp-Verbindung im Backup-Modus zum TwidoPort-Modul aufbauen können.



11.3 Kommunikationsfunktionen

Auf einen Blick

Inhalt dieses Abschnitts

Dieser Abschnitt beschreibt die vom ConneXium TwidoPort-Ethernet-Gateway unterstützten Kommunikationsfunktionen.

Inhalt dieses Abschnitts

Dieser Abschnitt enthält die folgenden Themen:

Thema	Seite
Ethernet-Merkmale	312
Modbus/TCP-Kommunikationsprotokoll	313
Lokal unterstützte Modbus-Funktionscodes	314

Ethernet-Merkmale

Einleitung

Das ConneXium TwidoPort-Modul erweitert die Twido-Produktreihe von Telemecanique um die Ethernet-Vernetzungsfähigkeit. Es ist das Gateway zwischen einem einfachen Twido Modbus/RTU- (RS-485) Gerät und der physikalischen Schicht von Modbus/TCP-Netzwerken im Slave-Modus. TwidoPort erfordert keine separate Stromversorgung, da es über seinen seriellen Port von der Twido-Steuerung gespeist wird. Dieses Gateway-Modul unterstützt nur den Slave-Modus.

Ethernet-Merkmale

TwidoPort unterstützt die folgenden Ethernet-Funktionen:

- **Automatische Einstellung**
  TwidoPort unterstützt die automatische 10/100TX-Einstellung. Es kommuniziert nur im Halbduplex-Modus.
- **Auto-MDI/MDI-X**
  TwidoPort unterstützt die automatische Umschaltung der Sende- und Empfangs-Drahtpaare, um die Kommunikation mit dem Endgerät aufzubauen (Auto-MDI/MDI-X). TwidoPort verbindet daher auf transparente Art und Weise Infrastrukturen oder Endgeräte mittels durchgehender Kabel oder Kreuzkabel.

Modbus/TCP-Kommunikationsprotokoll

Über Modbus

Das Modbus-Protokoll ist ein Master/Slave-Protokoll, bei dem ein Master Antworten von Slaves abfragen oder Aktionen auf der Grundlage ihrer Requests ausführen kann. Der Master kann individuelle Slaves adressieren oder eine Broadcast-Nachricht an alle Slaves senden. Slaves senden eine Meldung (Antwort) auf Requests, die individuell an sie gerichtet worden sind, zurück. Es werden keine Antworten auf Broadcast-Requests vom Master zurückgesendet.

Über die Modbus/TCP-Kommunikation

TwidoPort unterstützt bis zu 8 simultane Modbus/TCP-Verbindungen. Der Versuch, mehr als 8 Verbindungen zu verwenden, führt zu einer Verschlechterung der Leistung, da TwidoPort die Verbindung mit der längsten Ruhezeit schließt, um einen neuen Verbindungs-Request zu bearbeiten.

Funktionsweise

Modbus/TCP-Clients können über TwidoPort, eine Bridge zwischen Twido-Geräten (Modbus/RTU über eine serielle RS-485-Verbindung) und Modbus/TCP über Ethernet-Netzwerke mit Twido kommunizieren.

Hinweis: Bei der Implementierung von TwidoPort in einem Netzwerk muss das Systemdesign die inhärente begrenzte Bandbreite von seriellen Verbindungen berücksichtigen. Gehen Sie von einer Spitzenleistung von etwa 40 Modbus-Transaktionen pro Sekunde aus. Die Abfrage von mehreren Registern in einem einzigen Request ist effizienter als einen einzelnen Request für jedes Register zu senden.

Sie können das Lesen oder Schreiben von Requests von der Twido-Steuerung nicht über TwidoPort auslösen.

Lokal unterstützte Modbus-Funktionscodes

Funktionscodeliste

TwidoPort beantwortet die folgenden lokal unterstützten Modbus-Funktionscodes nur, wenn die Geräte-ID auf 254 gesetzt ist. (Lokal unterstützte Funktionscodes sind solche, die direkt von TwidoPort und nicht durch die Twido-Steuerung beantwortet werden.)

Modus-Funktioncode	Unterfunktionscode	OPCODE	Beschreibung
8	0	Nicht zutreffend	Query-Daten zurücksenden
8	10	Nicht zutreffend	Zähler zurücksetzen
8	11	Nicht zutreffend	Anzahl der Busmeldungen zurücksenden
8	12	Nicht zutreffend	Anzahl der Buskommunikationsfehler zurücksenden
8	13	Nicht zutreffend	Anzahl der Busausnahmefehler zurücksenden
8	14	Nicht zutreffend	Anzahl der Slave-Meldungen zurücksenden
8	15	Nicht zutreffend	Anzahl der vom Slave unbeantworteten Meldungen zurücksenden
8	21	3	Ethernet-Statistiken abfragen
8	21	4	Ethernet-Statistiken löschen
43	14	Nicht zutreffend	Geräte-ID lesen (siehe *Hinweise 1*)
Hinweis 1	**TwidoPort unterstützt nur die Basisobjekt-IDs des Funktionscodes "Geräte-ID lesen" mit Fluss- und individuellem Zugang.**		

Hinweis: Ausführliche Informationen über die Meldungsformate und Zugangsklassen finden Sie in der Modbus-Spezifikation unter *www.modbus.org*.

Betrieb des Bedienterminals

12

Auf einen Blick

Inhalt dieses Kapitels

In diesem Kapitel erfahren Sie Einzelheiten zur Verwendung des optionalen Twido Bedienterminals.

Inhalt dieses Kapitels

Dieses Kapitel enthält die folgenden Themen:

Thema	Seite
Bedienterminal	316
Steuerungskennung und Zustandsinformationen	319
Systemobjekte und Systemvariablen	321
Einstellungen des seriellen Ports	327
Echtzeituhr	328
Korrekturfaktor der Echtzeituhr	329

Bedienterminal

Einführung

Das Bedienterminal ist eine Twido Option, die das Anzeigen und Steuern von Applikationsdaten sowie von einigen Steuerungsfunktionen wie Betriebszustand und Echtzeituhr (RTC) ermöglicht. Diese Option ist als Cartridge (TWDXCPODC) für Kompaktsteuerungen und als Erweiterungsmodul (TWDXCPODM) für modulare Steuerungen erhältlich. Das Bedienterminal verfügt über zwei Betriebszustände:

- Anzeigemodus: nur Datenanzeige.
- Bearbeitungsmodus: Änderung von Daten möglich.

Hinweis: Das Bedienterminal wird in bestimmten Abständen vom Zyklus der Steuerung aktualisiert. Dies kann zu Verwirrung bei der Auswertung zweckbestimmter Ausgänge für die Impulse %PLS oder %PWM führen. Zum Zeitpunkt der Abtastung der Ausgänge ist deren Wert immer Null, und dieser Wert wird angezeigt.

Anzeigen und Funktionen

Das Bedienterminal bietet die folgenden Anzeigen mit den zugehörigen Funktionen, die Sie in jeder Anzeige ausführen können.

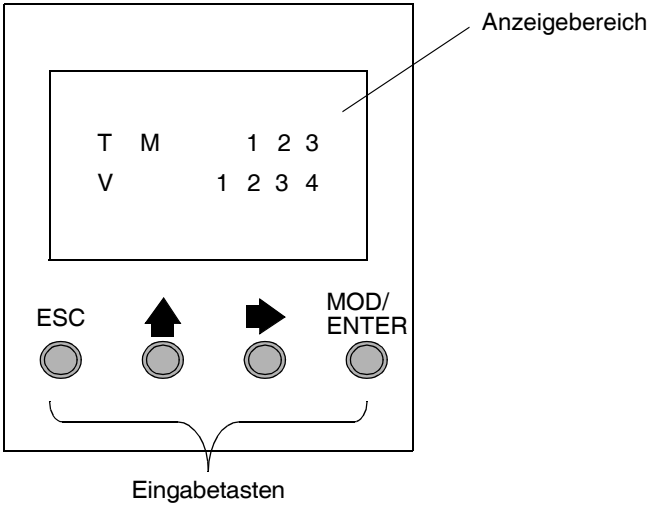
- Steuerungskennung und Zustandsinformationen: Bedienterminal
  Anzeigen der Firmware-Version und des Steuerungszustands. Ändern des Steuerungszustands mit den Befehlen "Run", "Initial" und "Stop".
- Systemobjekte und Systemvariablen: Datenanzeige
  Auswählen von Applikationsdaten nach der Adresse: %I, %Q und alle anderen Software-Objekte auf der Basissteuerung. Beobachten und Ändern des Werts eines gewählten Software-Datenobjekts.
- Einstellungen serieller Port: Kommunikationsanzeige
  Anzeigen und Ändern von Einstellungen für den Kommunikationsport.
- Echtzeituhr: Anzeige Uhrzeit/Datum
  Anzeigen und Konfigurieren des aktuellen Datums und der aktuellen Uhrzeit (wenn die Echtzeituhr installiert ist).
- Echtzeitkorrektur: RTC-Faktor
  Anzeigen und Ändern des Echzeituhr-Korrekturwerts für die optionale Echtzeituhr.

Hinweis:

1. Die TWDLCA•40DRF-Serie der kompakten Steuerungen sind mit einer Echtzeituhr ausgestattet.
2. An allen anderen Steuerungen sind die Echtzeituhr und Echtzeitkorrektur nur verfügbar, wenn die optionale Echtzeituhr-Cartridge (RTC) (TWDXCPRTC) installiert ist.

Abbildung

Im Folgenden sehen Sie eine Darstellung des Bedienterminals, der aus dem Anzeigebereich (hier im normalen Modus) und vier Drucktastern als Eingabetasten besteht.



Anzeigebereich

Das Bedienterminal verfügt über eine LCD-Anzeige, die zwei Zeilen mit Zeichen anzeigen kann:

- Die erste Zeile der Anzeige hat drei 13-Segment-Zeichen und vier 7-Segment-Zeichen.
- Die zweite Zeile hat ein 13-Segment-Zeichen, ein 3-Segment-Zeichen (für ein Plus-/Minuszeichen) und fünf 7-Segment-Zeichen.

Hinweis: Im normalen Modus zeigt die erste Zeile einen Objektnamen und die zweite Zeile dessen Wert an. Im Datenmodus zeigt die erste Zeile den Wert %SW68 und die zweite Zeile den Wert %SW68 an.

Eingabetasten

Die Funktionen der vier Eingabetasten richten sich nach dem Zustand des Bedienterminals:

Taste	Im Anzeigemodus	Im Bearbeitungsmodus
ESC		Änderungen verwerfen und zur vorherigen Anzeige zurückkehren.
▲		Zum nächsten Wert des Objekts gehen, das bearbeitet wird.
▶	Zur nächsten Anzeige weiterschalten.	Zum nächsten zu bearbeitenden Objekttyp gehen.
MOD/ ENTER	In den Bearbeitungsmodus wechseln.	Änderungen annehmen und zur vorherigen Anzeige zurückkehren.

Wählen der Anzeigen und Navigieren in den Anzeigen

Die erste Anzeige bzw. die erste Maske des Bedienterminals zeigt die Steuerungskennung und die Zustandsinformationen. Drücken Sie die Taste ▶, um durch die einzelnen Anzeigen zu blättern. Die Masken für die Echtzeituhr und den Korrekturfaktor der Echtzeituhr werden nicht angezeigt, wenn die optionale Echtzeituhr-Cartridge (TWDXCPRTC) nicht auf der Steuerung erkannt wird.
Als Kurztaste drücken Sie ESC, um zur ersten Anzeige zurückzukehren. In den meisten Anzeigen können Sie ESC drücken, wenn Sie zur Anzeige der Steuerungskennung und der Zustandsinformationen zurückkehren möchten. Wenn Sie allerdings ein Systemobjekt bzw. eine Systemvariable bearbeiten und nicht bei der ersten Eingabe (%I0.0.0) sind, kehren Sie mit ESC in die erste Anzeige des Systemobjekts zurück.
Wenn Sie einen Objektwert ändern möchten, können Sie noch einmal MOD/ENTER drücken, anstatt die Taste ▶ zu drücken, um zur Ziffer des ersten Werts zu gelangen.

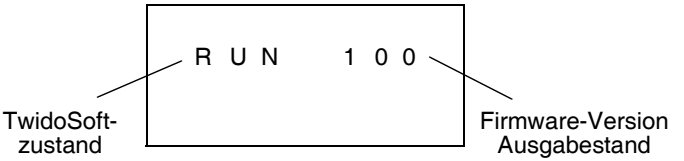
Steuerungskennung und Zustandsinformationen

Einleitung

Die erste Anzeige bzw. die erste Maske des optionalen Twido Bedienterminals zeigt die Steuerungskennung und die Zustandsinformationen.

Im Beispiel zur Bezugnahme

Der Ausgabestand der Firmware-Version wird in der oberen rechten Ecke des Anzeigebereichs angezeigt und der Steuerungszustand wird in der oberen linken Ecke des Anzeigebereichs angezeigt. Dies wird im Folgenden dargestellt:



Steuerungszustände

Es gibt folgende Steuerungszustände:

- **NCF: Nicht konfiguriert**
  Die Steuerung befindet sich im Zustand "NCF", bis eine Applikation geladen wird. Bis ein Applikationsprogramm geladen ist, ist kein anderer Zustand zulässig. Sie können die E/A testen, indem Sie Systembit S8 ändern (siehe *Systembits (%S), S. 620*).
- **STP: Angehalten**
  Wenn eine Applikation in der Steuerung vorhanden ist, wechselt der Zustand in "STP", also in den Zustand "Angehalten". In diesem Zustand läuft die Applikation nicht. Die Eingänge werden aktualisiert und die Datenwerte bleiben auf dem letzten Wert. Die Ausgänge werden in diesem Zustand nicht aktualisiert.
- **INI: Initialisierungszustand**
  Sie können die Steuerung nur aus dem Zustand STP in den Zustand INI, also in den Ausgangszustand, versetzen. Die Applikation läuft nicht. Die Eingänge der Steuerung werden aktualisiert und die Datenwerte werden auf die Ausgangswerte gesetzt. Die Ausgänge werden in diesem Zustand nicht aktualisiert.
- **RUN: Läuft**
  Im Zustand "RUN" läuft die Applikation. Die Eingänge der Steuerung werden aktualisiert und die Datenwerte werden entsprechend der Applikation gesetzt. Dies ist der einzige Zustand, in dem die Ausgänge aktualisiert werden.
- **HLT: Unterbrochen (Applikationsfehler)**
  Ist die Steuerung in den Zustand ERR, also in den Fehlerzustand gegangen, wird die Applikation unterbrochen. Die Eingänge werden aktualisiert und die Datenwerte bleiben auf dem letzten Wert. In diesem Zustand werden die Ausgänge nicht aktualisiert. In diesem Zustand wird der Fehlercode im unteren rechten Teil des Bedienterminals als vorzeichenloser Dezimalwert angezeigt.
- **NEX: Nicht ausführbar**
  Eine Online-Änderung wurde an die Benutzerlogik gesendet. Folge: Die Applikation ist nicht mehr ausführbar. Sie kann erst wieder in ihren Zustand zurückkehren, wenn alle Ursachen des Zustands Non Exec beseitigt sind.

Anzeigen und Ändern der Steuerungszustände

Mit dem Bedienterminal können Sie aus dem Zustand STP in den Zustand INI wechseln, Sie können aus dem Zustand STP in RUN und aus dem Zustand RUN in STP wechseln. Sie wechseln den Zustand der Steuerung folgendermaßen:

Schritt	Aktion
1	Drücken Sie die Taste ➡, bis das Fenster zur Anzeige des Zustands der Steuerung erscheint (oder drücken Sie die Taste ESC). Der aktuelle Betriebszustand der Steuerung wird in der oberen linken Ecke des Anzeigebereichs angezeigt.
2	Drücken Sie die Taste MOD/ENTER, um in den Bearbeitungsmodus zu wechseln.
3	Drücken Sie die Taste ⬆, um einen Steuerungszustand zu wählen.
4	Drücken Sie die Taste MOD/ENTER, um den geänderten Wert zu bestätigen, oder drücken Sie ESC, um alle im Bearbeitungsmodus vorgenommenen Änderungen zu verwerfen.

Systemobjekte und Systemvariablen

Einführung

Das optionale Bedienterminal bietet diese Funktionen zum Beobachten und Ändern von Applikationsdaten:

- Auswählen der Applikationsdaten nach Adresse (z.B. %I oder %Q)
- Beobachten des Werts eines gewählten Software-Objekts oder einer Software-Variablen
- Ändern des Werts des aktuell angezeigten Datenobjekts (einschließlich Forcieren von Eingängen und Ausgängen)

Systemobjekte und Systemvariablen

Die folgende Tabelle führt die Systemobjekte und Systemvariablen in der Zugriffsreihenfolge auf, die über das Bedienterminal angezeigt und geändert werden können.

Objekt	Variable/Attribut	Beschreibung	Zugriff
Eingang	%Ix.y.z	Wert	Lesen/Forcieren
Ausgang	%Qx.y.z	Wert	Lesen/Schreiben/Forcieren
Timer	%TMX.V %TMX.P %TMX.Q	Aktueller Wert Voreingestellter Wert Fertig	Lesen/Schreiben Lesen/Schreiben Lesen
Zähler	%Cx.V %Cx.P %Cx.D %Cx.E %Cx.F	Aktueller Wert Voreingestellter Wert Fertig Leer Voll	Lesen/Schreiben Lesen/Schreiben Lesen Lesen Lesen
Speicherbit	%Mx	Wert	Lesen/Schreiben
Wortspeicher	%MWx	Wert	Lesen/Schreiben
Konstantes Wort	%KWx	Wert	Lesen
Systembit	%Sx	Wert	Lesen/Schreiben
Systemwort	%SWx	Wert	Lesen/Schreiben
Analogeingang	%IWx.y.z	Wert	Lesen
Analogausgang	%QWx.y.z	Wert	Lesen/Schreiben
Schneller Zähler	%FCx.V %FCx.VD* %FCx.P %FCx.PD* %FCx.D	Aktueller Wert Aktueller Wert Voreingestellter Wert Voreingestellter Wert Fertig	Lesen Lesen Lesen/Schreiben Lesen/Schreiben Lesen

Objekt	Variable/Attribut	Beschreibung	Zugriff
Sehr schneller Zähler	%VFCx.V	Aktueller Wert	Lesen
	%VFCx.VD*	Aktueller Wert	Lesen
	%VFCx.P	Voreingestellter Wert	Lesen/Schreiben
	%VFCx.PD*	Voreingestellter Wert	Lesen/Schreiben
	%VFCx.U	Zählrichtung	Lesen
	%VFCx.C	Wert erfassen	Lesen
	%VFCx.CD*	Wert erfassen	Lesen
	%VFCx.S0	Schwellenwert 0	Lesen/Schreiben
	%VFCx.S0D*	Grenzwert 0	Lesen/Schreiben
	%VFCx.S1	Schwellenwert 1	Lesen/Schreiben
	%VFCx.S1D*	Schwellenwert 1	Lesen/Schreiben
	%VFCx.F	Überlauf-	Lesen
	%VFCx.T	Zeitbasis	Lesen/Schreiben
	%VFCx.R	Reflexausgang freigeben	Lesen/Schreiben
	%VFCx.S	Reflexeingang freigeben	Lesen/Schreiben
Eingangsnetzwerkwort	%INWx.z	Wert	Lesen
Ausgangsnetzwerkwort	%QNWx.z	Wert	Lesen/Schreiben
Grafcet	%Xx	Schrittbit	Lesen
Impulsgenerator	%PLS.N	Anzahl Impulse	Lesen/Schreiben
	%PLS.ND*	Anzahl Impulse	Lesen/Schreiben
	%PLS.P	Voreingestellter Wert	Lesen/Schreiben
	%PLS.D	Fertig	Lesen
	%PLS.Q	Stromausgang	Lesen
Impulsbreitenmodulator	%PWM.R	Verhältnis	Lesen/Schreiben
	%PWM.P	Voreingestellter Wert	Lesen/Schreiben
Trommelsteuerung	%DRx.S	Aktuelle Schrittnummer voll	Lesen
	%DRx.F		Lesen
Schrittzähler	%SCx.n	Schrittzählerbit	Lesen/Schreiben
Register	%Rx.I	Eingang	Lesen/Schreiben
	%Rx.O	Ausgang	Lesen/Schreiben
	%Rx.E	Leer	Lesen
	%Rx.F	Voll	Lesen
Schiebebitregister	%SBR.x.yy	Registerbit	Lesen/Schreiben
Meldung	%MSGx.D	Fertig	Lesen
	%MSGx.E	Fehler	Lesen
AS-Interface-Slaveeingang	%IAx.y.z	Wert	Lesen/Forcieren
Analoger AS-Interface-Slaveeingang	%IWAx.y.z	Wert	Lesen
AS-Interface-Slaveausgang	%QAx.y.z	Wert	Lesen/Schreiben/Forcieren
Analoger AS-Interface-Slaveausgang	%QWAx.y.z	Wert	Lesen/Schreiben
PDO-Eingang des CANopen-Slaves	%IWCx.y.z	Wert des Einzelworts	Lesen
PDO-Ausgang des CANopen-Slaves	%QWCx.y.z	Wert des Einzelworts	Lesen/Schreiben

Hinweise:

1. (*) Bedeutet eine 32-Bit-Doppelwortvariable. Die Doppelwortoption ist an allen Steuerungen mit Ausnahme der Twido-Steuerungen TWDLC•A10DRF verfügbar.
2. Variablen werden nicht angezeigt, wenn sie nicht in der Applikation verwendet werden, weil Twido die dynamische Speicherzuordnung nutzt.
3. Ist der Wert von %MW größer als +32767 oder kleiner als -32768, blinkt das Bedienterminal weiter.
4. Ist der Wert von %SW größer als 65535, blinkt das Bedienterminal weiter, jedoch nicht für %SW0 und %SW11. Wird ein Wert eingegeben, der den Grenzwert überschreitet, wird wieder der konfigurierte Wert verwendet.
5. Wird für %PLS.P ein Wert eingegeben, der den Grenzwert überschreitet, ist der geschriebene Wert der Sättigungswert.

Anzeigen und Ändern von Objekten und Variablen

Auf alle Arten von Systemobjekten wird auf folgende Weise zugegriffen: Beginn am Eingangsobjekt (%I), der Reihe nach weiter bis zum Meldungsobjekt (%MSG) und schließlich Rücksprung zum Eingangsobjekt (%I).
Anzeigen eines Systemobjekts:

Schritt	Aktion
1	Drücken Sie die Taste ➡, bis die Maske "Datenanzeige" angezeigt wird. Das Eingangsobjekt ("I") wird in der oberen linken Ecke des Anzeigebereichs angezeigt. Das Zeichen " I " (oder der Name des zuvor als Daten angezeigten Objekts) blinkt nicht.
2	Drücken Sie die Taste MOD/ENTER, um in den Bearbeitungsmodus zu wechseln. Das Zeichen "I" des Eingangsobjekts (oder der Name des zuvor als Daten angezeigten Objekts) beginnt zu blinken.
3	Drücken Sie die Taste ⬆, um der Reihe nach durch die Liste der Objekte zu blättern.
4	Drücken Sie die Taste ➡, um der Reihe nach durch das Feld eines Objekttyps zu blättern und drücken Sie die Taste ⬆, um den Wert des Felds zu inkrementieren. Mit den Tasten ➡ und ⬆ können Sie navigieren und alle Felder des angezeigten Objekts ändern.
5	Wiederholen Sie die Schritte 3 und 4, bis Sie die Bearbeitung beendet haben.
6	Drücken Sie die Taste MOD/ENTER, um die bearbeiteten Werte zu akzeptieren Hinweis: Der Name und die Adresse des Objekts müssen validiert werden, bevor Sie Änderungen akzeptieren. Das heißt, sie müssen in der Konfiguration der Steuerung vor Verwendung des Bedienterminals vorhanden sein. Drücken Sie ESC, um im Bearbeitungsmodus vorgenommene Änderungen zu verwerfen.

Datenwerte und Anzeigeformate

Im allgemeinen wird der Datenwert eines Objekts oder einen Variablen als Ganzzahl mit oder ohne Vorzeichen in der unteren rechten Ecke des Anzeigebereichs angezeigt. Außerdem unterdrücken alle Felder führende Nullen in den angezeigten Werten. Die Adresse der Objekte wird in dem Bedienterminal in einem der folgenden sieben Formate angezeigt:

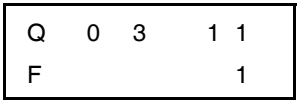
- E/A-Format
- Ein-/Ausgangs-Format der AS-Interface-Slaves
- E/A-Format der CANopen-Slaves
- Funktionsblockformat
- Einfaches Format
- Netzwerk-E/A-Format
- Schrittzählerformat
- Schiebebitregisterformat

Eingangs-/ Ausgangsformat

Die Eingangs-/Ausgangsobjekte (%I, %Q, %IW und %QW) haben dreiteilige Adressen (z.B. %IX.Y.Z) und werden wie folgt angezeigt:

- Objekttyp und Steuerungsadresse oben links
- Adresse der Erweiterung oben in der Mitte
- E/A-Kanal oben rechts

Bei einem einfachen Eingang (%I) oder Ausgang (%Q) enthält der untere linke Teil der Anzeige ein Zeichen, nämlich "U" für ein nicht forciertes Bit und "F" für ein forciertes Bit. Der Forcierungswert wird unten rechts auf dem Bildschirm angezeigt. Das Ausgangsobjekt %Q0.3.11 wird wie folgt auf dem Anzeigebereich angezeigt:

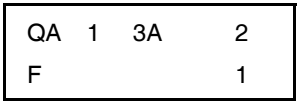


Ein-/Ausgangs-Format der AS-Interface-Slaves

Die Eingangs-/Ausgangsobjekte der AS-Interface-Slaves (%IA, %QA, %IWA und %QWA) haben vierteilige Adressen (z.B. %IAx.y.z) und werden wie folgt angezeigt:

- Objekttyp oben links
- Adresse des AS-Interface-Masters auf dem Erweiterungsbus oben Mitte links
- Adresse des Slaves auf dem AS-Interface-Bus oben Mitte rechts
- E/A-Kanal des Slaves oben rechts

Bei einem einfachen Eingang (%IA) oder Ausgang (%QA) enthält der untere linke Teil der Anzeige ein Zeichen, nämlich "U" für ein nicht forciertes Bit und "F" für ein forciertes Bit. Der Forcierungswert wird unten rechts auf dem Bildschirm angezeigt. Das Ausgangsobjekt 3A.2 wird wie folgt auf dem Anzeigebereich angezeigt:

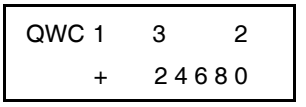


E/A-Format der CANopen-Slaves

Die PDO-E/A-Objekte der CANopen-Slaves (%IWC und %QWC) haben vierteilige Adressen (z.B. %IWCx.y.z) und werden wie folgt angezeigt:

- Objekttyp oben links
- Adresse des CANopen-Masters auf dem Erweiterungsbus oben Mitte links
- Adresse des Slaves auf dem CANopen-Bus oben Mitte rechts
- PDO-E/A-Kanal des Slaves oben rechts
- Wert des Objekts mit Vorzeichen im unteren Teil

Im folgenden Beispiel enthält das PDO-Ausgangsobjekt %QWC1.3.2 den vorzeichenbehafteten Wert +24680:

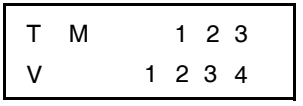


Funktionsblock-format

Die Funktionsblöcke (%TM, %C, %FC, %VFC, %PLS, %PWM, %DR, %R und %MSGj) haben zweiteilige Adressen, die eine Objektnummer und einen Variablen- bzw. Attributnamen enthalten. Sie werden wie folgt angezeigt:

- Funktionsblockname oben links
- Funktionsblocknummer (oder Instanz) oben rechts
- Die Variable bzw. das Attribut unten links
- Wert des Attributs unten rechts

Im folgenden Beispiel wird der aktuelle Wert für Timer Nummer 123 auf 1.234 gesetzt.

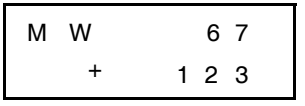


Einfaches Format

Ein einfaches Format wird für die Objekte %M, %MW, %KW, %MD, %KD, %MF, %KF, %S, %SW und %X wie folgt verwendet:

- Objektnummer oben rechts
- Wert des Objekts mit Vorzeichen im unteren Teil

Im folgenden Beispiel enthält das Speicherwort 67 den Wert +123.

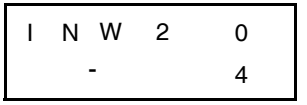


Netzwerkeingangs-/-ausgangsformat

Die Netzwerkeingangs-/-ausgangsobjekte (%INW und %QNW) werden in der Anzeige wie folgt angezeigt:

- Objektname oben links
- Steuerungsadresse oben in der Mitte
- Objektnummer oben rechts
- Wert des Objekts mit Vorzeichen im unteren Teil

Im folgenden Beispiel wird das erste Eingangsnetzwerkwort der dezentralen Steuerung an der dezentralen Adresse #2 auf den Wert -4 gesetzt.

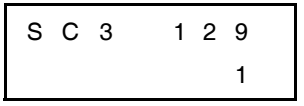


Schrittzählerformat

Das Format für den Schrittzähler (%SC) zeigt die Objektnummer und das Schrittzählerbit wie folgt an:

- Objektname und -nummer oben links
- Nummer des Schrittzählerbits oben rechts
- Wert des Schrittzählerbits im unteren Teil der Anzeige

Im folgenden Beispiel wird Bit-Nr. 129 der Schrittzählernummer 3 auf 1 gesetzt.

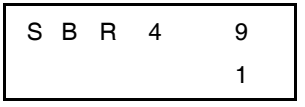


Schiebebitregisterformat

Das Schiebebitregister (%SBR) wird wie folgt auf dem Anzeigebereich angezeigt:

- Objektname und -nummer oben links
- Schiebebitnummer oben rechts
- Schiebebitwert unten rechts

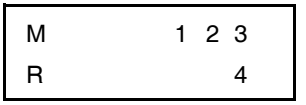
Das folgende Beispiel zeigt die Anzeige von Schiebebitregister Nummer 4.



Einstellungen des seriellen Ports

Einleitung

Mit dem Bedienterminal können Sie die Protokolleinstellungen anzeigen und die Adressen aller seriellen Ports, die mit TwidoSoft konfiguriert wurden, ändern. Die maximale Anzahl serieller Ports ist 2. Im folgenden Beispiel ist der erste Port als Modbus-Protokoll mit der Adresse 123 konfiguriert. Der zweite serielle Port ist als dezentrale Verbindung mit der Adresse 4 konfiguriert.



Anzeigen und Ändern der Einstellungen der seriellen Ports

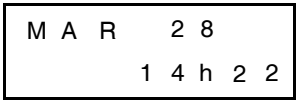
Twido Steuerungen unterstützen bis zu zwei serielle Ports. Anzeigen der Einstellungen der seriellen Ports auf dem Bedienterminal:

Schritt	Aktion
1	Drücken Sie die Taste ▶, bis die Maske "Kommunikation" angezeigt wird. Der einzelne Buchstabe der Protokolleinstellung des ersten seriellen Ports ("M", "R" oder "A") wird in der oberen linken Ecke des Bedienterminals angezeigt.
2	Drücken Sie die Taste MOD/ENTER, um in den Bearbeitungsmodus zu wechseln.
3	Drücken Sie die Taste ▶, bis Sie in dem Feld sind, das Sie ändern möchten.
4	Drücken Sie die Taste ▲, um den Wert in dem Feld zu erhöhen.
5	Fahren Sie mit den Schritten 3 und 4 fort, bis Sie die Adresseinstellungen beendet haben.
6	Drücken Sie die Taste MOD/ENTER, um die geänderten Werte anzunehmen, oder drücken Sie ESC, um alle im Bearbeitungsmodus vorgenommenen Änderungen zu verwerfen.
Hinweis: Die Adresse ist Teil der Konfigurationsdaten in der Steuerung. Wenn Sie die Adresse im Bedienterminal ändern, bedeutet dies, dass Sie die Verbindung nicht mehr wie gewohnt über TwidoSoft herstellen können. TwidoSoft benötigt einen Ladevorgang, um wieder auf gleichem Stand zu sein.	

Echtzeituhr

Einführung

Sie können das Datum und die Uhrzeit mit dem Bedienterminal ändern, wenn die optionale Echtzeituhr-Cartridge (TWDXCPRTC) in der Twido Steuerung installiert ist. Der Monat wird auf der oberen linken Seite des Bedienterminals angezeigt. Bis eine gültige Uhrzeit eingegeben wird, zeigt das Feld für den Monat den Wert "RTC" an. Der Wochentag wird in der oberen rechten Ecke der Anzeige angezeigt. Die Uhrzeit wird im Militärformat angezeigt. Die Stunden und Minuten werden in der unteren rechten Ecke der Anzeige angezeigt und sind durch den Buchstaben "h" getrennt. Das folgende Beispiel zeigt die Echtzeituhr mit dem Datum 28. März, 2:22 PM.



Hinweis:
1. Die TWDLCA•40DRF-Serie der kompakten Steuerungen sind mit einer Echtzeituhr ausgestattet.
2. An allen anderen Steuerungen sind die Echtzeituhr und Echtzeitkorrektur nur verfügbar, wenn die optionale Echtzeituhr-Cartridge (RTC) (TWDXCPRTC) installiert ist.

Anzeigen und Ändern der Echtzeituhr

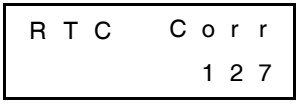
Anzeigen und Ändern der Echtzeituhr:

Schritt	Aktion
1	Drücken Sie die Taste ➡, bis die Maske "Zeit-/Datum" angezeigt wird. Der Monat ("JAN", "FEB") wird in der oberen linken Ecke des Anzeigebereichs angezeigt. Der Wert "RTC" wird in der oberen linken Ecke angezeigt, wenn kein Monat initialisiert wurde.
2	Drücken Sie die Taste MOD/ENTER, um in den Bearbeitungsmodus zu wechseln.
3	Drücken Sie die Taste ➡, bis Sie in dem Feld sind, das Sie ändern möchten.
4	Drücken Sie die Taste ⬆, um den Wert in dem Feld zu erhöhen.
5	Fahren Sie mit Schritt 3 und 4 fort, bis der Wert für die Uhrzeit vollständig eingestellt ist.
6	Drücken Sie die Taste MOD/ENTER, um die geänderten Werte anzunehmen, oder drücken Sie ESC, um alle im Bearbeitungsmodus vorgenommenen Änderungen zu verwerfen.

Korrekturfaktor der Echtzeituhr

Einleitung

Sie können den Korrekturfaktor für die Echtzeituhr mit der Bedieneranzeige anzeigen und ändern. Jedes optionale Echtzeituhr-Modul (RTC) hat einen RTC-Korrekturfaktor, mit dem Ungenauigkeiten im Kristall des Echtzeituhr-Moduls ausgeglichen werden. Der Korrekturfaktor ist eine vorzeichenlose dreistellige Ganzzahl zwischen 0 und 127. Dieser Wert wird in der unteren rechten Ecke der Anzeige angezeigt.
Im Folgenden sehen Sie ein Beispiel für einen Korrekturfaktor von 127.



Anzeigen und Ändern des Korrekturfaktors der Echtzeituhr

Anzeigen und Ändern des Korrekturfaktors der Echtzeituhr:

Schritt	Aktion
1	Drücken Sie die Taste ➡, bis die Maske "Echtzeituhr-Korrekturfaktor" angezeigt wird. "RTC Korr" wird in der oberen Zeile der Bedieneranzeige angezeigt.
2	Drücken Sie die Taste MOD/ENTER, um in den Bearbeitungsmodus zu wechseln.
3	Drücken Sie die Taste ➡, bis Sie in dem Feld sind, das Sie ändern möchten.
4	Drücken Sie die Taste ⬆, um den Wert in dem Feld zu erhöhen.
5	Fahren Sie mit Schritt 3 und 4 fort, bis der Wert für den Korrekturfaktor der Echtzeituhr vollständig eingestellt ist.
6	Drücken Sie die Taste MOD/ENTER, um die geänderten Werte anzunehmen, oder drücken Sie ESC, um alle im Bearbeitungsmodus vorgenommenen Änderungen zu verwerfen.

Beschreibung der Twido Sprachen

III

Auf einen Blick

Inhalt dieses Abschnitts

Dieser Teil bietet Anweisungen für den Einsatz der Programmiersprachen KOP, AWL und Grafcet, mit denen Sie Steuerungsprogramme für speicherprogrammierbare Twido Steuerungen erstellen können.

Inhalt dieses Teils

Dieser Teil enthält die folgenden Kapitel:

Kapitel	Kapitelname	Seite
13	Kontaktplan	333
14	Anweisungsliste	357
15	Grafcet	369

Kontaktplan

13

Auf einen Blick

Inhalt dieses Kapitels

Dieses Kapitel beschreibt die Programmierung mit Hilfe der Programmiersprache KOP.

Inhalt dieses Kapitels

Dieses Kapitel enthält die folgenden Themen:

Thema	Seite
Einführung in Kontaktplan-Diagramme	334
Programmierprinzipien in Kontaktplänen	336
Blöcke in Kontaktplänen	338
Grafische Elemente des Kontaktplans	342
Besondere KOP-Anweisungen OPEN und SHORT	345
Tipps für die Programmierung	346
KOP/AWL-Umkehrbarkeit	351
Empfehlungen für die Umkehrbarkeit zwischen KOP und AWL	352
Programmdokumentation	354

Einführung in Kontaktplan-Diagramme

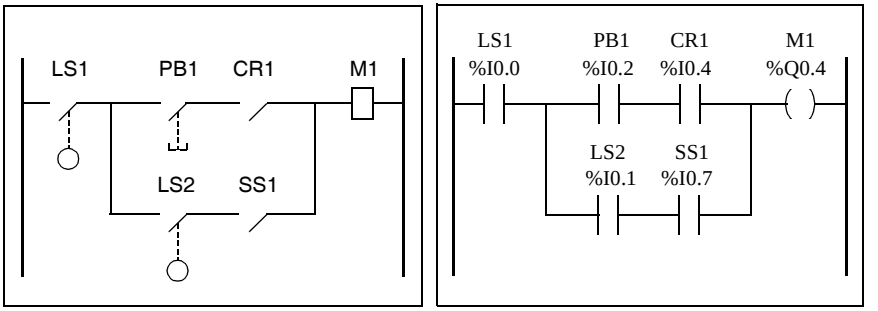
Einleitung

Kontaktplan-Diagramme ähneln Schaltplänen, in denen Relaisschaltkreise dargestellt werden. Die Hauptunterschiede zwischen den beiden sind die folgenden Funktionen der Kontaktplan-Programmierung, die Schaltkreise nicht bieten:

- Alle Eingänge werden in Form von Kontakten dargestellt (⊣⊢).
- Alle Ausgänge werden in Form von Spulen dargestellt (⊣○⊢).
- Der graphische KOP-Anweisungssatz enthält numerische Operationen.

KOP-Entsprechungen zu Relaisschalt-kreisen

Die folgende Abbildung zeigt einen vereinfachten Verdrahtungsplan eines Relaisschaltkreises und das entsprechende Kontaktplan-Diagramm.



Relaisschaltkreis

Kontaktplan

Beachten Sie, dass in der Abbildung oben alle Eingänge, denen im Relaisschaltkreis ein Schalter zugewiesen ist, im Kontaktplan-Diagramm als Kontakte dargestellt werden. Die Ausgangsspule M1 im Relaisschaltkreis wird im Kontaktplan-Diagramm durch das Symbol einer Spule dargestellt. Die Adressen oberhalb der Kontakte und Spulen im Kontaktplan-Diagramm verweisen auf die Positionen der externen Eingangs- und Ausgangsanschlüsse der Steuerung.

KOP-Programmblöcke

Ein in der Sprache KOP geschriebenes Programm besteht aus Programmblöcken, bei denen es sich um mehrere graphische Anweisungen handelt, die zwischen zwei vertikalen Potenzialschienen angeordnet sind. Die Programmblöcke werden von der Steuerung der Reihe nach ausgeführt.

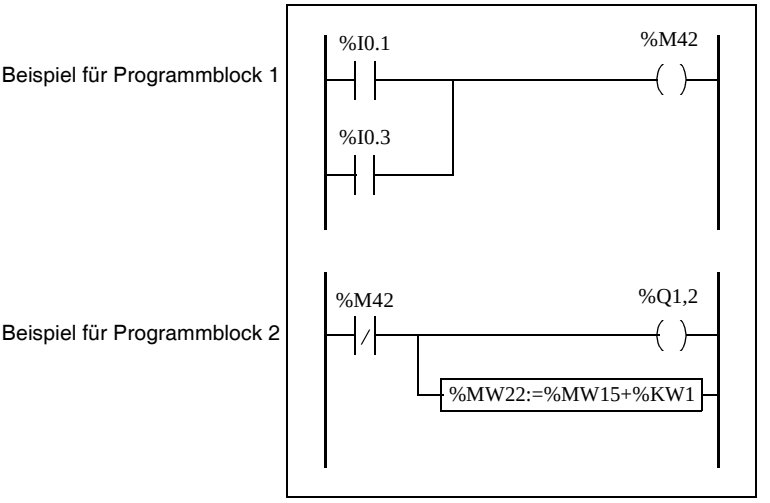
Die graphischen Anweisungen stellen die folgenden Funktionen dar:

- Eingänge und Ausgänge der Steuerung (Drucktaster, Sensoren, Relais, Anzeigeleuchten, ...)
- Funktionen der Steuerung (Timer, Zähler, ...)
- Arithmetische und logische Verknüpfungen (Addition, Division, UND, Exklusives ODER, ...)
- Vergleichsoperatoren und andere numerische Operationen (A<B, A=B, Schieben, Rotieren, ...)
- Interne Variablen in der Steuerung (Bits, Wörter, ...)

Diese graphischen Anweisungen werden mit vertikalen und horizontalen Verbindungen angeordnet, die schließlich zu einem Ausgang oder zu mehreren Ausgängen und/oder Aktionen führen. Ein Programmblock kann nicht mehrere Gruppen von verknüpften Anweisungen unterstützen.

Beispiel für KOP-Programmblöcke

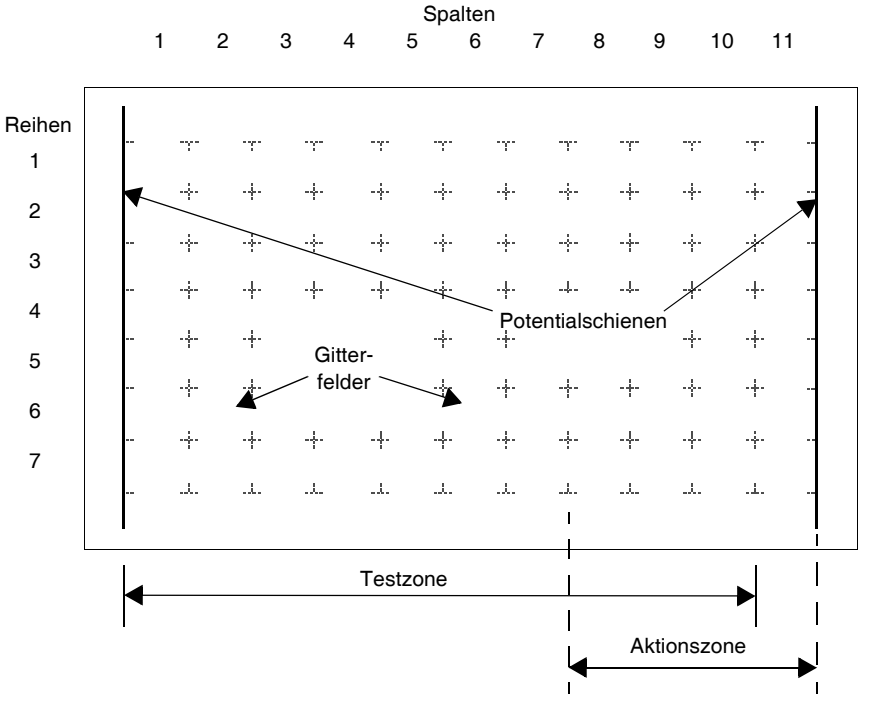
Das folgende Diagramm zeigt ein Beispiel für ein KOP-Programm, das aus zwei Programmblöcken besteht.



Programmierprinzipien in Kontaktplänen

Programmier-gitter

Jeder KOP-Strompfad besteht aus einem Gitter mit sieben Reihen und elf Spalten, das in zwei Zonen aufgeteilt ist. Dies ist in der folgenden Abbildung dargestellt.



Zonen im Gitter

Das Programmiergitter in KOP ist in zwei Zonen unterteilt:

- Testzone
  Enthält die Bedingungen, die getestet werden, damit Aktionen ausgeführt werden. Besteht aus den Spalten 1 bis 10 und enthält Kontakte, Funktionsblöcke und Vergleichsbausteine.
- Aktionszone
  Enthält den Ausgang bzw. den Befehl, der entsprechend den Ergebnissen der Bedingungstests in der Testzone ausgeführt wird. Besteht aus den Spalten 8 bis 11 und enthält Spulen und Operationsbausteine.

Eingeben von Anweisungen im Gitter

Ein KOP-Strompfad besteht aus einem Programmiergitter mit sieben Reihen und elf Spalten. Der Strompfad beginnt im ersten Feld oben links im Gitter. Bei der Programmierung werden Anweisungen in die Felder im Gitter eingegeben. Testanweisungen, Vergleiche und Funktionen werden in die Felder in der Testzone eingegeben. Diese sind linksbündig. Die Testlogik wird fortgesetzt in der Aktionszone, in der Spulen, numerische Verknüpfungen und Anweisungen für die Steuerung des Programmflusses eingegeben werden. Diese sind rechtsbündig. Der Strompfad wird im Gitter von oben nach unten und von links nach rechts gelöst bzw. ausgeführt (Tests werden durchgeführt und Ausgänge zugewiesen).

Header des Strompfads

Zusätzlich zum Strompfad erscheint direkt oberhalb von diesem ein Header für den Strompfad. In diesem Strompfad-Header können Sie den logischen Zweck des Strompfads dokumentieren. Der Header des Strompfads kann folgende Informationen enthalten:

- Strompfadnummer
- Markierungen (%Li)
- Unterprogrammdeklarationen (SRi:)
- Strompfadtitel
- Strompfadkommentare

Weitere Informationen zur Verwendung des Strompfad-Headers zur Dokumentation Ihres Programms finden Sie unter *Programmdokumentation, S. 354*.

Blöcke in Kontaktplänen

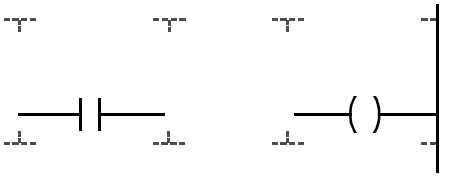
Einführung

Kontaktpläne bestehen aus Blöcken, die den Programmfluss und folgende Funktionen darstellen:

- Kontakte
- Spulen
- Programmflussanweisungen
- Funktionsblöcke
- Vergleichsbausteine
- Operationsbausteine

Kontakte, Spulen und Programmfluss

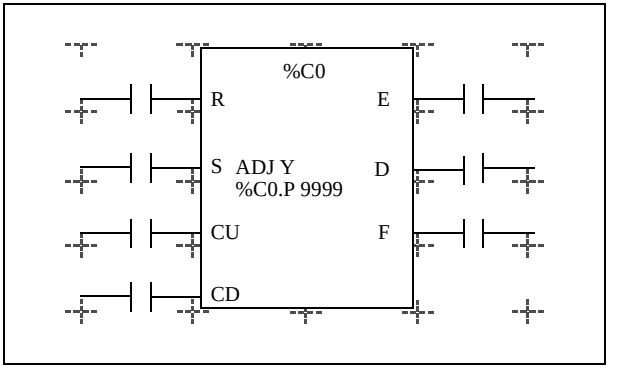
Kontakte, Spulen und Anweisungen für den Programmfluss (Sprung und Aufruf) belegen ein einzelnes Feld im Gitter für die KOP-Programmierung. Funktionsblöcke, Vergleichsbausteine und Operationsbausteine belegen mehrere Felder. Im Folgenden sehen Sie Beispiele für Kontakte und Spulen.



Kontakt Spule

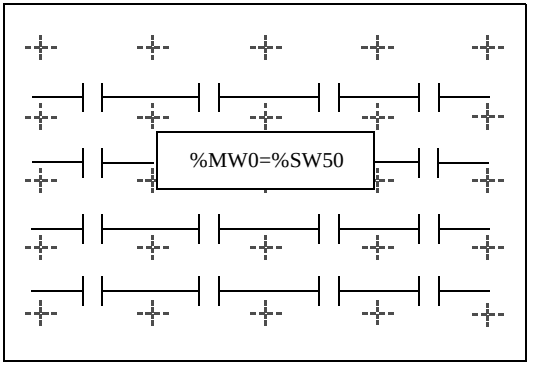
Funktionsblöcke

Funktionsblöcke werden in der Testzone des Programmiergitters angeordnet. Der Block muss sich in der ersten Reihe befinden. Es dürfen sich keine KOP-Anweisungen oder Fortsetzungszeilen über oder unter dem Funktionsblock befinden. KOP-Testanweisungen führen zur Eingangsseite eines Funktionsblocks und Testanweisungen und/oder Aktionsanweisungen gehen von der Ausgangsseite des Funktionsblocks ab.
Funktionsblöcke sind vertikal ausgerichtet und belegen zwei Spalten und vier Reihen im Programmiergitter.
Im Folgenden sehen Sie ein Beispiel für einen Zähler-Funktionsblock.



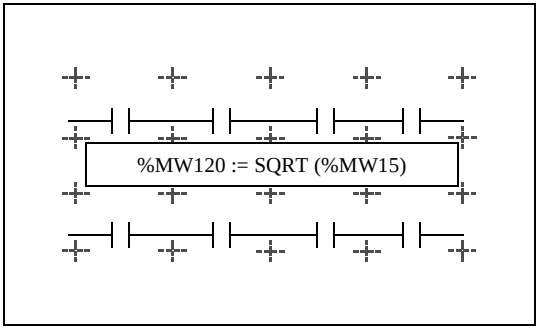
Vergleichsbausteine

Vergleichsbausteine werden in der Testzone des Programmiergitters angeordnet. Der Block kann in einer beliebigen Reihe und in einer beliebigen Spalte in der Testzone angeordnet werden, solange sich die gesamte Länge der Anweisung in der Testzone befindet.
Vergleichsbausteine sind horizontal ausgerichtet und belegen zwei Spalten und eine Reihe im Programmiergitter.
Im Folgenden sehen Sie ein Beispiel für einen Vergleichsbaustein.



Operationsbausteine

Operationsbausteine werden in der Aktionszone des Programmiergitters angeordnet. Der Baustein darf sich in einer beliebigen Reihe in der Aktionszone befinden. Die Anweisung ist rechtsbündig, d.h. Sie beginnt rechts und endet in der letzten Spalte.
Operationsbausteine sind horizontal ausgerichtet und belegen vier Spalten und eine Reihe im Programmiergitter.
Im Folgenden sehen Sie ein Beispiel für einen Operationsbaustein.



Grafische Elemente des Kontaktplans

Einführung

Anweisungen in Kontaktplänen bestehen aus grafischen Elementen.

Kontakte

Kontakte werden in der Testzone programmiert und belegen ein Feld (eine Reihe in der Höhe und eine Spalte in der Breite).

Name	Grafisches Element	Anweisung	Funktion
Arbeitskontakt		LD	Der Kontakt ist stromleitend, wenn sich das Befehlsbitobjekt im Zustand 1 befindet.
Ruhekontakt		LDN	Der Kontakt ist stromleitend, wenn sich das Befehlsbitobjekt im Zustand 0 befindet.
Kontakt zur Erkennung einer steigenden Flanke		LDR	Steigende Flanke: Erkennt den Übergang des Befehlsbitobjekts von 0 auf 1.
Kontakt zur Erkennung einer fallenden Flanke		LDF	Fallende Flanke: Erkennt den Übergang des Befehlsbitobjekts von 1 auf 0.

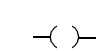
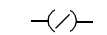
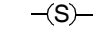
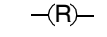
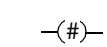
Verknüpfungselemente

Die grafischen Verknüpfungselemente dienen zur Verknüpfung der grafischen Test- und Aktionselemente.

Name	Grafisches Element	Funktion
Horizontale Verbindung		Verknüpft die grafischen Test- und Aktionselemente zwischen den zwei Potentialschienen in Reihe.
Vertikale Verbindung		Verbindet die graphischen Test- und Aktionselemente parallel.

Spulen

Spulen werden in der Aktionszone programmiert und belegen ein Feld (eine Reihe in der Höhe und eine Spalte in der Breite).

Name	Grafisches Element	Anweisung	Funktion
Direkte Spule		ST	Das zugehörige Bitobjekt nimmt den Wert des Testzonenergebnisses an.
Inverse Spule		STN	Das zugewiesene Bitobjekt nimmt den invertierten Wert des Ergebnisses des Testbereichs an.
Spule setzen		S	Das zugewiesene Bitobjekt wird auf 1 gesetzt, wenn das Ergebnis des Testbereichs 1 ist.
Spule zurücksetzen		R	Das zugewiesene Bitobjekt wird auf 0 gesetzt, wenn das Ergebnis des Testbereichs 1 ist.
Sprung oder Unterprogrammaufruf	->>%Li ->>%SRi	JMP SR	Verbindung zu einer markierten Anweisung, vor oder nach dieser Anweisung.
Weiterschaltbedingungsspule			Programmiersprache Grafcet. Wird verwendet, wenn die Programmierung der mit den Transitionen verknüpften Transitionsbedingungen eine Permutation mit dem folgenden Schritt hervorruft.
Rückkehr aus einem Unterprogramm	<RET>	RET	Wird am Ende von Unterprogrammen angeordnet und dient der Rückkehr ins Hauptprogramm.
Programm stoppen	<END>	END	Definiert das Ende des Programms.

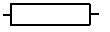
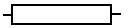
Funktionsblock

Die grafischen Elemente von Funktionsblöcken werden in der Testzone programmiert und benötigen Felder mit vier Reihen und zwei Spalten (mit Ausnahme sehr schneller Zähler, die fünf Reihen und zwei Spalten benötigen).

Name	Grafisches Element	Funktion
Timer, Zähler, Register usw.		Jeder Funktionsblock verwendet die Eingänge und Ausgänge, die eine Verbindung mit den anderen grafischen Elementen erlauben. Hinweis: Ausgänge von Funktionsblöcken können nicht miteinander verknüpft werden (kurze vertikale Verbindungen).

Operations- und Vergleichsbausteine

Vergleichsbausteine werden in der Testzone programmiert. Operationsbausteine werden in der Aktionszone programmiert.

Name	Grafisches Element	Funktion
Vergleichsbaustein		Vergleicht zwei Operanden und der Ausgang geht auf 1, wenn das Ergebnis geprüft ist. Größe: eine Reihe und zwei Spalten
Operationsbaustein		Führt arithmetische und logische Befehle aus. Größe: eine Reihe und vier Spalten

Besondere KOP-Anweisungen OPEN und SHORT

Einführung

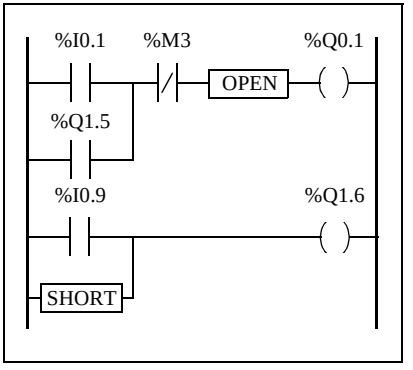
Die Anweisungen OPEN und SHORT bieten eine bequeme Methode zum Testen und Fehlerbeheben von KOP-Programmen. Diese besonderen Anweisungen ändern die Logik eines Strompfads, indem der Stromfluss in einem Strompfad entweder kurzgeschlossen oder geöffnet wird (siehe folgende Tabelle).

Anweisung	Beschreibung	AWL-Anweisung
OPEN	Unterbricht den Stromfluss in einem KOP-Strompfad unabhängig von den Ergebnissen der letzten logischen Verknüpfung.	`AND 0`
SHORT	Ermöglicht den Stromfluss im Strompfad unabhängig von den Ergebnissen der letzten logischen Verknüpfung.	`OR 1`

In der AWL-Programmierung werden die Anweisungen OPEN und SHORT mit den Anweisungen ODER und UND und den direkten Werten von jeweils 0 und 1 erstellt.

Beispiele

Im Folgenden sehen Sie Beispiele für die Verwendung der Anweisungen OPEN und SHORT.



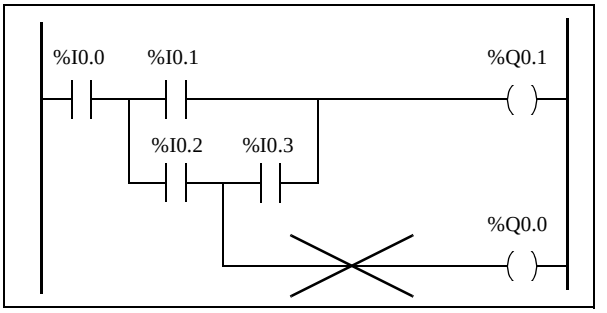

```
LD      %I0.1
OR      %Q1.5
ANDN    %M3
AND     0
ST      %Q0.1
LD      %I0.9
OR      1
ST      %Q1.6
```

Tipps für die Programmierung

Programm-sprünge	Setzen Sie Programmsprünge mit Vorsicht ein, um lange Schleifen zu vermeiden, die die Zykluszeit erhöhen können. Vermeiden Sie Sprünge zu Anweisungen, die sich vor der Sprunganweisung befinden. (Eine vorgeschaltete Anweisung erscheint vor einem Sprung in einem Programm. Im Gegensatz dazu erscheint eine nachgeschaltete Anweisung nach einem Sprung in einem Programm.)
Programmierung von Ausgängen	Die Ausgangbits dürfen genau wie die internen Bits nur einmal im Programm geändert werden. Bei Ausgangsbits wird nur der letzte abgetastete Wert für die Aktualisierung der Ausgänge berücksichtigt.
Direkt verdrahtete Not-Aus-Sensoren	Sensoren, die als Not-Aus-Einrichtungen genutzt werden, dürfen nicht von der Steuerung verarbeitet werden. Sie müssen direkt an die entsprechenden Ausgänge angeschlossen werden.
Spannungs-rückkehr	Erstellen Sie eine Bedingung für die Wiederkehr der Spannungsversorgung bei manueller Operation. Ein automatischer Neustart kann eine ungewünschte Funktion der Installation auslösen (verwenden Sie die Systembits %S0, %S1 und %S9).
Verwaltung der Uhr und der Abarbeitungs-blöcke	Der Zustand des Systembits %S51 muss überprüft werden. Dieses Bit zeigt einen eventuellen Fehler der Echtzeituhr an.
Syntax- und Fehlerprüfung	Nach der Eingabe eines Programms überprüft TwidoSoft die Syntax der Anweisungen und Operanden und deren Zuordnung.

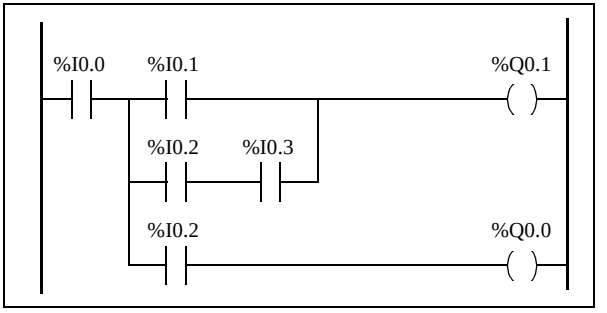
Zusätzliche Hinweise zur Verwendung von Klammern

Die Zuweisungsoperationen dürfen nicht in Klammern gesetzt werden:



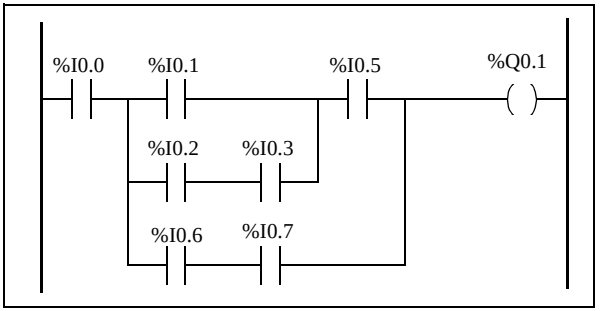

```
LD      %I0.0
AND     %I0.1
OR(     %I0.2
ST      %Q0.0
AND     %I0.3
)
ST      %Q0.1
```

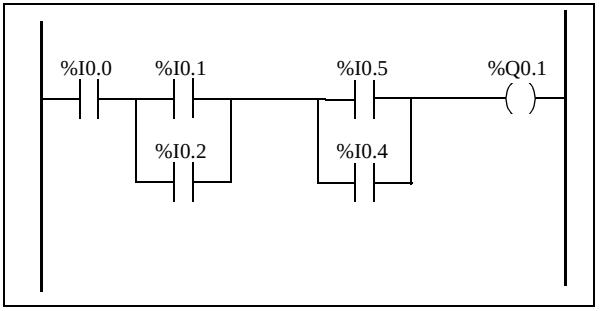
Um die entsprechende Funktion auszuführen, müssen die folgenden Gleichungen programmiert werden:



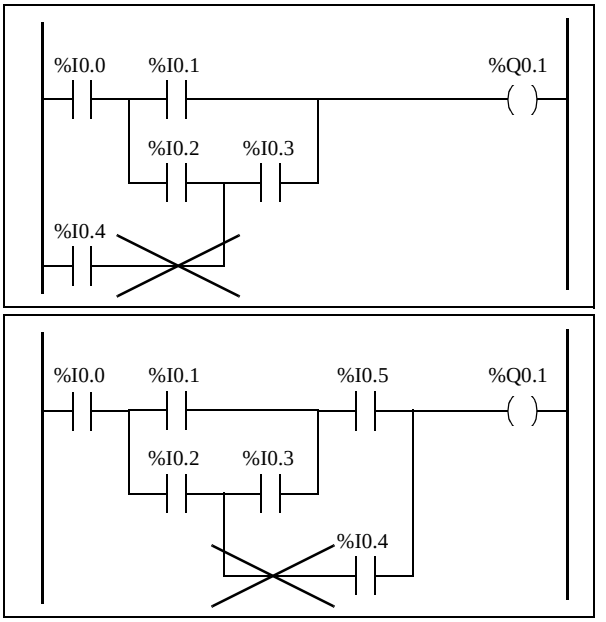

```
LD      %I0.0
MPS
AND(    %I0.1
OR(     %I0.2
AND     %I0.3
)
)
ST      %Q0.1
MPP
AND     %I0.2
ST      %Q0.0
```

Wenn mehrere Parallelschaltungen von Kontakten vorgenommen werden, müssen sie entweder ineinander verschachtelt oder vollkommen voneinander getrennt werden:

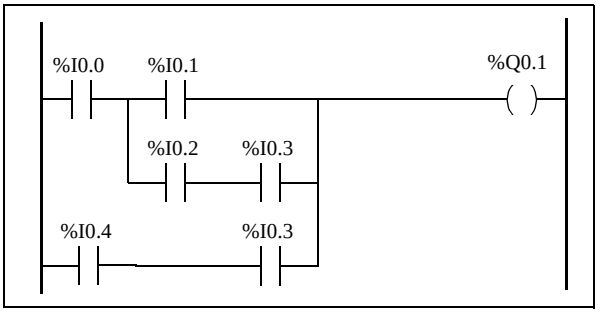




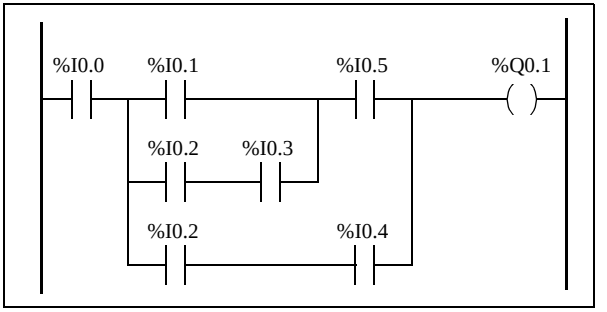
Folgende Schemata können nicht programmiert werden:



Um äquivalente Schemata auszuführen, modifizieren Sie die Schemata wie folgt:




```
LD      %I0.0
AND(    %I0.1
OR(     %I0.2
AND     %I0.3
)
)
OR(     %I0.4
AND     %I0.3
)
ST      %Q0.1
```




```
LD      %I0.0
AND(    %I0.1
OR(     %I0.2
AND     %I0.3
)
AND     %I0.5
OR(     %I0.2
AND     %I0.4
)
)
ST      %Q0.1
```

KOP/AWL-Umkehrbarkeit

Einleitung

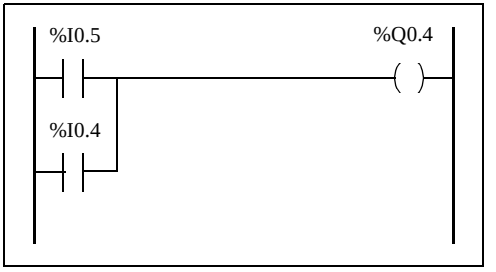
Die Umkehrbarkeit von Programmen ist ein Leistungsmerkmal der TwidoSoft Programmiersoftware, welches es Ihnen ermöglicht, Anwendungsprogramme von KOP in AWL und von AWL wieder in KOP umzuwandeln.
In TwidoSoft stellen Sie als Standardanzeige von Programmen das AWL- oder das KOP-Format ein (in den benutzerspezifischen Einstellungen). Sie können dann zwischen der AWL- und der KOP-Ansicht umschalten.

Umkehrbarkeit

Wesentlich für das Verständnis der Umkehrbarkeit von Programmen ist das Verständnis der Beziehung zwischen einem KOP-Programmblock und der zugehörigen Folge von AWL-Anweisungen:

- **KOP-Programmblock**: Gruppe von KOP-Anweisungen, die einen logischen Ausdruck bilden.
- **AWL-Sequenz**: Gruppe von Anweisungen eines AWL-Programms, die KOP-Anweisungen entsprechen und sich auf denselben logischen Ausdruck beziehen.

Die folgende Abbildung zeigt einen herkömmlichen KOP-Programmblock und die entsprechende Programmlogik in einer Folge von AWL-Anweisungen.




```
LD    %I0.5
OR    %I0.4
ST    %Q0.4
```

Ein Anwendungsprogramm wird intern in Form einer Anweisungsliste gespeichert, obwohl es in AWL oder KOP geschrieben wurde. TwidoSoft verwendet Ähnlichkeiten in der Programmstruktur der beiden Sprachen und die interne Abbildungsliste des Programms, um es in Form einer Anweisungsliste (Basisform) oder grafisch in Form eines Kontaktplans anhand der vom Benutzer gewählten Einstellungen anzuzeigen.

Sicherstellen der Umkehrbarkeit

Jedes Programm, das in KOP-Sprache geschrieben wurde, kann in AWL-Sprache konvertiert werden. Umgekehrt können jedoch einige Logiken der AWL-Sprache nicht in KOP-Sprache konvertiert werden. Um die vollständige Umkehrbarkeit zwischen den Sprachen AWL und KOP sicherzustellen, ist es wichtig, die Richtlinien im Abschnitt *Empfehlungen für die Umkehrbarkeit zwischen KOP und AWL, S. 352* zu beachten.

Empfehlungen für die Umkehrbarkeit zwischen KOP und AWL

Für die Umkehrbarkeit erforderliche Anweisungen

Die Struktur eines umkehrbaren Funktionsblocks in AWL erfordert die Verwendung folgender Anweisungen:

- **BLK** markiert den Beginn des Blocks und definiert den Beginn des Programmblocks und den Beginn des Eingangsteils des Block.
- **OUT_BLK** markiert den Beginn des Ausgangsteils des Blocks.
- **END_BLK** markiert das Ende des Blocks und des Programmblocks.

Die Verwendung von reversiblen Funktionsblockanweisungen ist für ein ordnungsgemäß funktionierendes AWL-Programm nicht obligatorisch. Einige Anweisungen können in AWL programmiert werden, die nicht reversibel sind. Vollständige Informationen über die nicht reversible AWL-Programmierung von Funktionsblöcken finden Sie im Hilfethema *Prinzipien für die Programmierung von Standard-Funktionsblöcken, S. 404*.

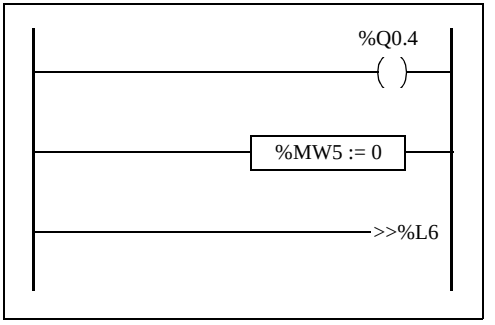
Zu vermeidende Anweisungen ohne KOP-Entsprechungen

Vermeiden Sie die Verwendung bestimmter AWL-Anweisungen und bestimmter Kombinationen von Anweisungen und Operanden, für die es in KOP keine Entsprechung gibt. Für die Anweisung N beispielsweise (invertiert den Wert im Booleschen Akkumulator) gibt es keine entsprechende KOP-Anweisung.
Die folgende Tabelle enthält alle Anweisungen für die AWL-Programmierung, die sich nicht in KOP umkehren lassen:

AWL-Anweisung	Operand	Beschreibung
JMPCN	%Li	Negation bedingter Sprung
N	keine	Negation (Not)
ENDCN	keine	Negation bedingtes Ende

Nicht bedingte Programmblöcke

Die Programmierung von nicht bedingten Programmblöcken erfordert ebenfalls die Anwendung der folgenden Empfehlungen für die AWL-Programmierung, damit die AWL/KOP-Umkehrbarkeit gegeben ist. Die nicht bedingten Programmblöcke unterliegen keinen Tests oder Bedingungen. Die Ausgänge oder Anweisungen sind immer aktiviert bzw. werden immer ausgeführt.
Die folgende Abbildung zeigt Beispiele für nicht bedingte Programmblöcke und die entsprechende AWL-Sequenz.




```
LD      1
ST      %Q0.4
LD      1
[%MW5 := 0]
JMP     %L6
```

Sie werden bemerkt haben, dass jede der obenstehenden nicht bedingten AWL-Sequenzen mit einer Ladeanweisung gefolgt von einer 1 beginnt, mit Ausnahme der JMP-Anweisung. Diese Kombination schreibt den Wert des booleschen Akkumulators auf 1 und setzt dadurch bei jeder Programmabtastung die Spule (Speicheranweisung) auf 1 und %MW5 auf 0. Eine Ausnahme ist die Anweisung für einen unbedingten Listensprung (JMP %L6), die unabhängig vom Wert des Akkumulators ausgeführt wird und kein Schreiben des Akkumulators auf 1 erfordert.

KOP/AWL-Programmblock

Wird ein AWL-Programm, das nicht komplett reversibel ist, in KOP umgekehrt, werden die reversiblen Teile in der KOP-Ansicht dargestellt und die nicht reversiblen Teile werden in KOP/AWL-Strompfaden dargestellt.
Ein KOP-Programmblock in AWL funktioniert genauso wie ein kleiner AWL-Editor. Er ermöglicht dem Anwender, die unumkehrbaren Teile eines KOP-Programms anzuzeigen und zu ändern.

Programmdokumentation

Dokumentieren Ihres Programms

Sie können Ihr Programm durch Eingeben von Kommentaren in den AWL- und KOP-Editoren dokumentieren:

- Im AWL-Editor dokumentieren Sie Ihr Programm mit Zeilenkommentaren. Diese Kommentare können auf der gleichen Zeile wie die Programmieranweisungen oder auf eigenen Zeilen erscheinen.
- Im KOP-Editor haben Sie durch Programmblock-Kopfzeilen die Möglichkeit, Ihr Programm zu dokumentieren. Diese Kopfzeilen befinden sich direkt unter dem Programmblock.

Die Programmiersoftware TwidoSoft nutzt diese Kommentare für die Umkehrbarkeit. Wenn ein Programm von AWL in KOP umgewandelt wird, nutzt TwidoSoft einige der AWL-Kommentare, um einen Programmblock-Header zu erstellen. Die Kommentare, die zwischen AWL-Sequenzen eingefügt wurden, werden in Programmblock-Header umgewandelt.

Beispiel für AWL-Zeilenkommentare

Im Folgenden sehen Sie ein Beispiel für ein AWL-Programm mit Zeilenkommentaren.

```
---- ( * KOPFZEILENTITEL VON PROGRAMMBLOCK 0 * )
---- ( * ERSTER KOPFZEILENKOMMENTAR VON PROGRAMMBLOCK 0 * )
---- ( * ZWEITER KOPFZEILENKOMMENTAR VON PROGRAMMBLOCK 0 * )
     0   LD  % I0. 0  ( * ZEILENKOMMENTAR *)
     1   OR  %I0. 1  ( * KOMMENTARZEILE WIRD BEI DER KONVERTIERUNG
    IN KOP IGNORIERT*)
     2   ANDM  %M10
     3   ST        M101
---- ( * KOPFZEILE VON PROGRAMMBLOCK 1 * )
---- ( * DIESER PROGRAMMBLOCK ENTHÄLT EINE MARKIERUNG * )
---- ( * ZWEITER KOPFZEILENKOMMENTAR VON PROGRAMMBLOCK 1 * )
---- ( * DRITTER KOPFZEILENKOMMENTAR VON PROGRAMMBLOCK 1 * )
---- ( * VIERTER KOPFZEILENKOMMENTAR VON PROGRAMMBLOCK 1 * )
     4   % L5:
     5   LD  %M101
     6   [ %MW20 := %KW2 * 16 ]
---- ( * DIESER PROGRAMMBLOCK ENTHÄLT NUR EINEN KOPFZEILENTITEL * )
     7   LD  %Q0. 5
     8   OR  %I0.  3
     9   ORR  I0.  13
    10   ST  %Q0.5
```

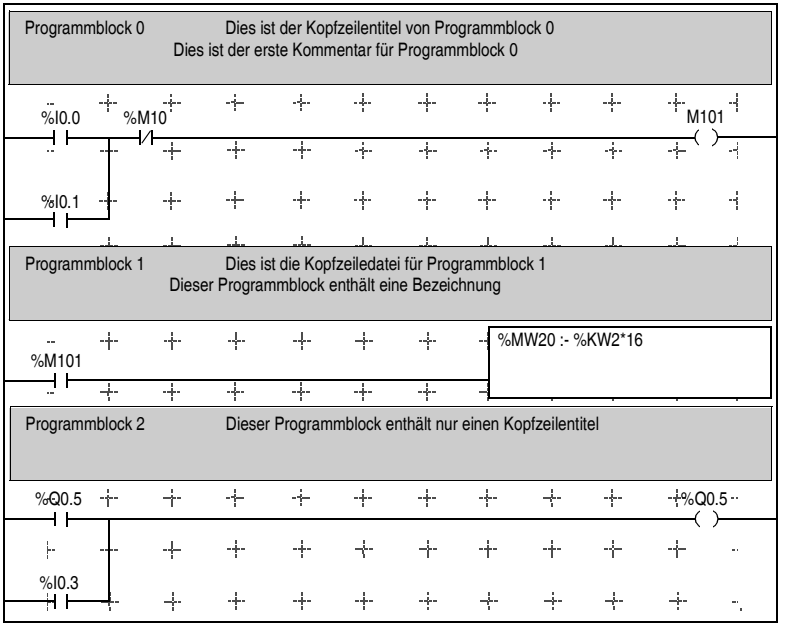
Umwandeln von AWL-Kommentaren in KOP

Wenn AWL-Anweisungen in KOP umgewandelt werden, werden die AWL-Zeilenkommentare entsprechend den folgenden Regeln im KOP-Editor angezeigt:

- Der erste Kommentar, der sich einzeln auf einer Zeile befindet, wird dem Programmblock-Header zugewiesen.
- Alle Kommentare nach dem ersten werden zum Körper des Programmblocks.
- Sind die Zeilen im Header belegt, werden die übrigen Zeilenkommentare zwischen AWL-Sequenzen ignoriert, genauso wie alle Kommentare, die sich auf AWL-Zeilen befinden, die auch AWL-Anweisungen enthalten.

Beispiel für Kommentare im Programmblock-Header

Im Folgenden sehen Sie ein Beispiel für ein KOP-Programm mit Kommentaren im Programmblock-Header.



Umwandeln von KOP-Kommentaren in AWL

Wenn ein Kontaktplan in AWL-Anweisungen umgewandelt wird, werden die Kommentare im Programmblock-Header entsprechend den folgenden Regeln im AWL-Editor angezeigt:

- Alle Kommentare im Programmblock-Header werden zwischen den zugehörigen AWL-Sequenzen eingefügt.
- Alle Bezeichnungen (%Li:) oder Unterprogrammdeklarationen (SRi:) werden in der Zeile nach der Kopfzeile und direkt vor der AWL-Sequenz platziert.
- Wurde die Anweisungsliste in einen Kontaktplan umgewandelt, werden alle Kommentare, die ignoriert wurden, wieder im AWL-Editor angezeigt.

Anweisungsliste

14

Auf einen Blick

Inhalt dieses Kapitels

Dieses Kapitel beschreibt die Programmierung mit Hilfe der Programmiersprache AWL.

Inhalt dieses Kapitels

Dieses Kapitel enthält die folgenden Themen:

Thema	Seite
Überblick über AWL-Programme	358
Funktionsweise von AWL-Anweisungen	360
AWL-Anweisungen	361
Verwenden von Klammern	364
Stapelanweisungen (MPS, MRD, MPP)	367

Überblick über AWL-Programme

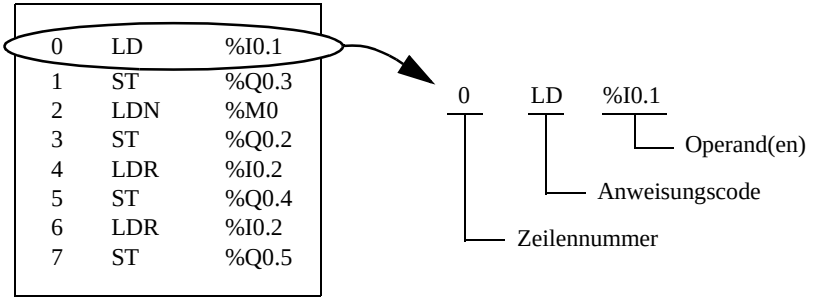
Einleitung

Ein Programm in der Sprache Steuerungsanweisungsliste (AWL) besteht aus einer Reihe von Anweisungen, die von der Steuerung nacheinander ausgeführt werden. Jede Listenanweisung besteht aus einer einzelnen Programmzeile aus drei Elementen:

- Zeilennummer
- Anweisungscode
- Operand(en)

Beispiel eines AWL-Programms

Hier sehen Sie ein Beispiel eines AWL-Programms.



Zeilennummer

Zeilennummern werden automatisch erzeugt, wenn Sie eine Anweisung eingeben. Leere Zeilen und Kommentarzeilen haben keine Nummern.

Anweisungscode

Anweisungscode ist ein Symbol für einen Operator, der die mit dem (den) Operanden durchzuführende Operation bestimmt. Typische Operatoren führen boolesche und numerische Operationen durch.
Im obigen Beispielprogramm ist LD die Abkürzung für den Anweisungscode einer LOAD-Anweisung. Die LOAD-Anweisung lädt den Wert des Operanden %I0.1 in ein internes Register, den Akkumulator.
Es gibt zwei grundlegende Anweisungstypen:

- Testanweisungen
  Diese Anweisungen bereiten vor oder testen auf die notwendigen Bedingungen, um eine Aktion durchzuführen. Beispiele sind LOAD (LD) und AND.
- Aktionsanweisungen
  Diese Anweisungen führen Aktionen auf der Grundlage der vorbereiteten Bedingungen aus. Beispiele sind Zuweisungsanweisungen wie STORE (ST) und RESET (R).

Operand

Ein Operand ist eine Zahl, Adresse oder ein Symbol, deren oder dessen Wert von einem Programm in einer Anweisung bearbeitet werden kann. Im obigen Beispiel ist der Operand %I0.1 eine Adresse, der der Wert eines Eingangs der Steuerung zugewiesen ist. Eine Anweisung kann null bis drei Operanden haben, je nach Art des Anweisungscodes.
Operanden können Folgendes sein:

- Steuerungs-Eingänge und -Ausgänge wie Sensoren, Tastschalter und Relais
- Vordefinierte Systemfunktionen wie Timer und Zähler
- Arithmetische, logische, numerische und Vergleichs-Operationen
- Steuerungsinterne Variablen wie Bits und Worte

Funktionsweise von AWL-Anweisungen

Einleitung

AWL-Anweisungen haben nur einen expliziten Operanden, der andere Operand ist impliziert. Der implizierte Operand ist der Wert im booleschen Akkumulator. Bei der Anweisung LD %I0.1 beispielweise ist %I0.1 der explizite Operand. Ein implizierter Operand wird im Akkumulator gespeichert und wird vom Wert von %I0.1 überschrieben.

Funktionsweise

Eine AWL-Anweisung führt eine bestimmte Funktion mit dem Inhalt des Akkumulators und dem expliziten Operanden aus. Der Inhalt des Akkumulators wird durch das Ergebnis ersetzt. Beispiel: Die Funktion AND %I1.2 führt die logische UND-Verknüpfung mit dem Inhalt des Akkumulators und dem Eingang 1.2 aus und ersetzt den Inhalt des Akkumulators mit dem Ergebnis.
Alle Booleschen Anweisungen verwenden zwei Operanden. Ausnahme sind die Anweisungen Laden, Speichern und NOT. Der Wert der beiden Operanden kann entweder True oder False sein. Die Programmausführung der Anweisungen erzeugt einen einzigen Wert, entweder True oder False. Ladeanweisungen platzieren den Wert des Operanden im Akkumulator, während Speicheranweisungen den Wert im Akkumulator in den Operanden übertragen. Die Anweisung NOT hat keine expliziten Operanden und invertiert einfach den Zustand des Akkumulators.

Unterstützte AWL-Anweisungen

Die folgende Tabelle zeigt einige Anweisungen in AWL:

Art der Anweisung	Im Beispiel zur Bezugnahme	Funktion
Bitanweisung	LD %M10	Liest das interne Bit %M10
Blockanweisung	IN %TM0	Startet den Timer %TM0
Wortanweisung	[%MW10 := %MW50+100]	Additionsverknüpfung
Programmanweisung	SR5	Ruft Unterprogramm #5 auf
Grafcet-Anweisung	-*-8	Schritt #8

AWL-Anweisungen

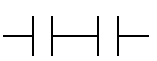
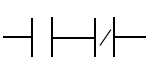
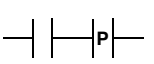
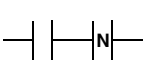
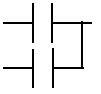
Einleitung

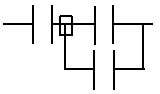
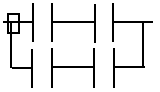
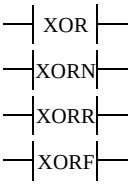
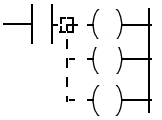
Die Programmiersprache AWL enthält folgende Anweisungstypen:
- Testanweisungen
- Aktionsanweisungen
- Funktionsblockanweisungen

In diesem Abschnitt werden die Twido Anweisungen für die AWL-Programmierung aufgeführt und beschrieben.

Testanweisungen

Die folgende Tabelle beschreibt die Testanweisungen in der Sprache AWL.

Name	Entsprechendes grafisches Element	Funktion
LD		Das boolesche Ergebnis entspricht dem Zustand des Operanden.
LDN		Das boolesche Ergebnis ist der umgekehrte Zustand des Operanden.
LDR		Das boolesche Ergebnis wechselt auf 1, wenn erkannt wird, dass der Operand (steigende Flanke) von 0 auf 1 wechselt.
LDF		Das boolesche Ergebnis wechselt auf 1, wenn erkannt wird, dass der Operand (fallende Flanke) von 1 auf 0 wechselt.
AND		Das boolesche Ergebnis ist gleich der UND-Verknüpfung zwischen dem booleschen Ergebnis der vorherigen Anweisung und dem Zustand des Operanden.
ANDN		Das boolesche Ergebnis ist gleich der UND-Verknüpfung zwischen dem booleschen Ergebnis der vorherigen Anweisung und dem umgekehrten Zustand des Operanden.
ANDR		Das boolesche Ergebnis ist gleich der UND-Verknüpfung zwischen dem booleschen Ergebnis der vorherigen Anweisung und der Erkennung der steigenden Flanke des Operanden (1 = steigende Flanke).
ANDF		Das boolesche Ergebnis ist gleich der UND-Verknüpfung zwischen dem booleschen Ergebnis der vorherigen Anweisung und der Erkennung der fallenden Flanke des Operanden (1 = fallende Flanke).
OR		Das boolesche Ergebnis ist gleich der ODER-Verknüpfung zwischen dem booleschen Ergebnis der vorherigen Anweisung und dem Zustand des Operanden.

Name	Entsprechendes grafisches Element	Funktion
AND(		Logisches UND (8 Klammerebenen)
OR(		Logisches ODER (8 Klammerebenen)
XOR, XORN, XORR, XORF		Exklusives ODER
MPS MRD MPP		Umschalten zu den Spulen.
N	-	Negation (NOT)

Aktionsanweisungen

Die folgende Tabelle beschreibt die Aktionsanweisungen in der Sprache AWL.

Name	Entsprechendes grafisches Element	Funktion
ST	—( )—	Der zugehörige Operand nimmt den Wert des Testzonenergebnisses an.
STN	—(/)—	Der zugehörige Operand nimmt den umgekehrten Wert des Testzonenergebnisses an.
S	—(S)—	Der zugehörige Operand wird auf 1 gesetzt, wenn das Ergebnis der Testzone 1 ist.
R	—(R)—	Der zugehörige Operand wird auf 0 gesetzt, wenn das Ergebnis der Testzone 1 ist.
JMP	->>%Li	Unbedingte Verbindung zu einer markierten Sequenz, vor oder nach dieser Anweisung.
SRn	->>%SRi	Verbindung am Anfang eines Unterprogramms.
RET	<RET>	Rückkehr aus einem Unterprogramm.
END	<END>	Ende des Programms.
ENDC	<ENDC>	Ende des bedingten Programms bei einem booleschen Ergebnis von 1.
ENDCN	<ENDCN>	Ende des bedingten Programms bei einem booleschen Ergebnis von 0.

Funktionsblockanweisungen

Die folgende Tabelle beschreibt die Funktionsblockanweisungen in der Sprache AWL.

Name	Entsprechendes grafisches Element	Funktion
Timer, Zähler, Register usw.		Für jeden Funktionsblock gibt es Anweisungen zum Steuern des Blocks. Eine strukturierte Form dient zur direkten Festverdrahtung der Eingänge und Ausgänge des Blocks. **Hinweis:** Ausgänge von Funktionsblöcken können nicht miteinander verknüpft werden (kurze vertikale Verbindungen).

Verwenden von Klammern

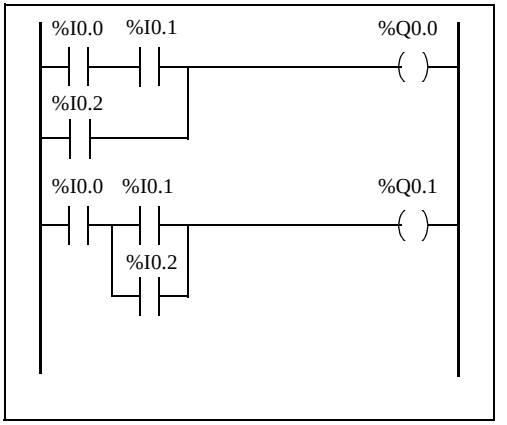
Einleitung

Mit Klammern können Sie bei logischen UND- und ODER-Anweisungen parallele Verzweigungen in Kontaktplänen kenntlich machen. Die öffnenden und schließenden Klammern werden wie folgt den Anweisungen zugewiesen:

- Das Öffnen einer Klammer wird der Anweisung UND oder ODER zugewiesen.
- Das Schließen der Klammer ist eine Anweisung, die für jede öffnende Klammer erforderlich ist.

Beispiel mit einer UND-Anweisung

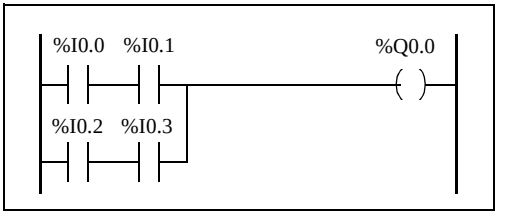
Die folgenden Diagramme zeigen die Verwendung von Klammern in einer UND-Anweisung: AND(...).




```
LD      %I0.0
AND     %I0.1
OR      %I0.2
ST      %Q0.0

LD      %I0.0
AND(    %I0.1
OR      %I0.2
)
ST      %Q0.1
```

Beispiel mit einer ODER-Anweisung

Die folgenden Diagramme zeigen die Verwendung von Klammern in einer ODER-Anweisung: OR(...).




```
LD      %I0.0
AND     %I0.1
OR(     %I0.2
AND     %I0.3
)
ST      %Q0.0
```

Modifikatoren

Die folgende Tabelle führt Modifikatoren auf, die den Klammern zugewiesen werden können.

Modifikator	Funktion	Im Beispiel zur Bezugnahme
N	Negation	AND(N oder OR(N
F	Fallende Flanke	AND(F oder OR(F
R	Steigende Flanke	AND(R oder OR(R
[	Vergleich	Siehe *Vergleichsanweisungen, S. 433*

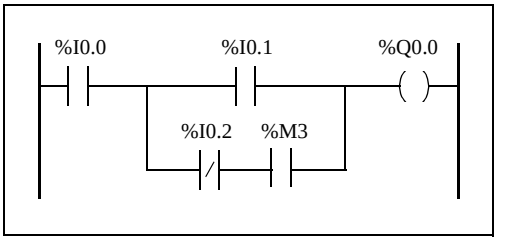
Verschachteln von Klammern

Sie können Klammern auf bis zu acht Ebenen verschachteln.
Beachten Sie die folgenden Richtlinien, wenn Sie Klammern verschachteln:

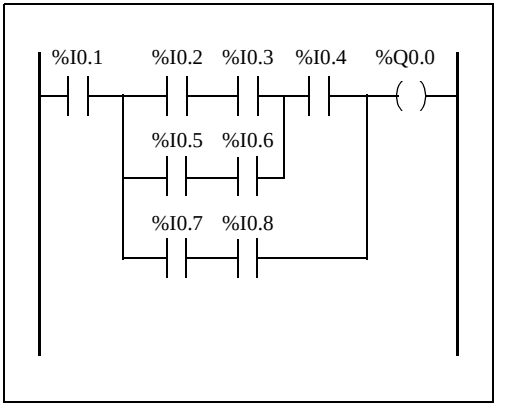
- Jede geöffnete Klammer muss auch eine entsprechende geschlossene Klammer besitzen.
- Markierungen (%Li:), Unterprogramme (SRi:), Sprunganweisungen (JMP) und Funktionsblockanweisungen dürfen nicht in Ausdrücken zwischen Klammern stehen.
- Die Speicheranweisungen ST, STN, S und R dürfen nicht zwischen Klammern programmiert werden.
- Die Stapelanweisungen MPS, MRD und MPP dürfen nicht zwischen Klammern verwendet werden.

Beispiele für die Verschachtelung von Klammern

Die folgenden Diagramme zeigen Beispiele für die Verschachtelung von Klammern.




```
LD      %I0.0
AND(    %I0.1
OR(N    %I0.2
AND     %M3
)
)
ST      %Q0.0
```




```
LD      %I0.1
AND(    %I0.2
AND     %I0.3
OR(     %I0.5
AND     %I0.6
)
AND     %I0.4
OR(     %I0.7
AND     %I0.8
)
)
ST      %Q0.0
```

Stapelanweisungen (MPS, MRD, MPP)

Einführung

Die Stapelanweisungen verarbeiten das Routing zu Spulen. Die Anweisungen MPS, MRD und MPP nutzen einen temporären Speicherbereich, der Stapel genannt wird und bis zu acht Boolesche Ausdrücke speichern kann.

Hinweis: Diese Anweisungen dürfen nicht in Ausdrücken zwischen Klammern verwendet werden.

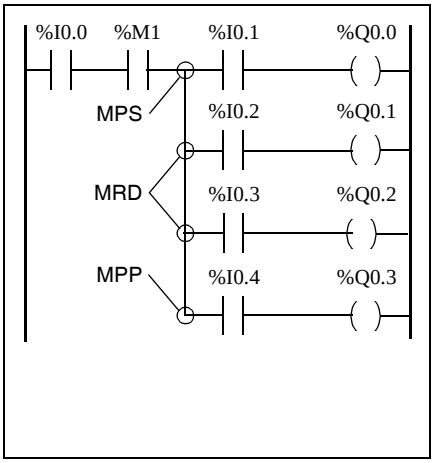
Funktionsweise von Stapelanweisungen

Die folgende Tabelle beschreibt die Funktionsweise der drei Stapelanweisungen.

Anweisung	Beschreibung	Funktion
MPS	Einlesen in den Stapelspeicher	Speichert das Ergebnis der letzten logischen Anweisung (Inhalt des Akkumulators) an oberster Stelle im Stapel (Einlesen) und schiebt die anderen Werte im Stapel nach unten.
MRD	Obersten Wert des Stapelspeichers lesen	Liest den obersten Wert des Stapels in den Akkumulator.
MPP	Informationen aus dem Stapelspeicher entnehmen	Kopiert den obersten Wert des Stapels in den Akkumulator (Auslesen) und schiebt die anderen Wert im Stapel nach oben.

Beispiele für Stapelanweisungen

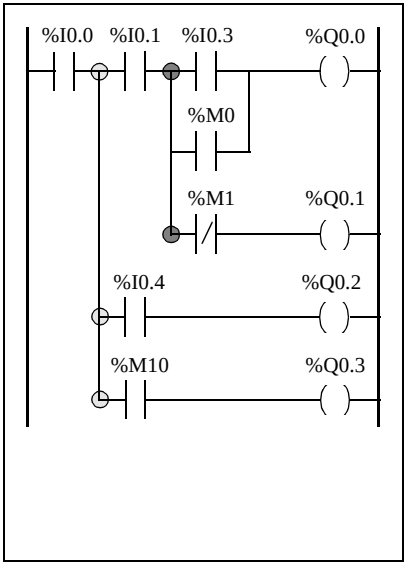
Die folgenden Diagramme sind Beispiele für Stapelanweisungen.




```
LD      %I0.0
AND     %M1
MPS
AND     %I0.1
ST      %Q0.0
MRD
AND     %I0.2
ST      %Q0.1
MRD
AND     %I0.3
ST      %Q0.2
MPP
AND     %I0.4
ST      %Q0.3
```

Beispiele für Stapelanweisungen

Die folgenden Diagramme zeigen die Funktionsweise von Stapelanweisungen.




```
LD      %I0.0
MPS
AND     %I0.1
MPS
AND(    %I0.3
OR      %M0
)
ST      %Q0.0
MPP
ANDN    %M1
ST      %Q0.1
MRD
AND     %I0.4
ST      %Q0.2
MPP
AND     %M10
ST      %Q0.3
```

Grafcet

15

Auf einen Blick

Inhalt dieses Kapitels

Dieses Kapitel beschreibt die Programmierung mit Hilfe der Programmiersprache Grafcet.

Inhalt dieses Kapitels

Dieses Kapitel enthält die folgenden Themen:

Thema	Seite
Beschreibung der Grafcet-Anweisungen	370
Beschreibung des Grafcet-Programmaufbaus	374
Zu Grafcet-Schritten gehörige Aktionen	377

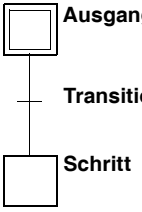
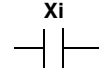
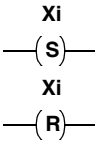
Beschreibung der Grafcet-Anweisungen

Einleitung

Grafcet-Anweisungen in TwidoSoft bieten eine einfache Methode zum Übersetzen einer Steuerungssequenz (Grafcet-Graph).
Die maximale Anzahl an Grafcet-Schritten richtet sich nach dem Typ der Twido-Steuerung. Die Anzahl der Schritte, die zu einem beliebigen Zeitpunkt aktiv sein dürfen, ist nur durch die Gesamtzahl der Schritte begrenzt.
Für TWDLCAA10DRF und TWDLCAA16DRF stehen die Schritte 1 bis 62 zur Verfügung. Die Schritte 0 und 63 sind reserviert für die Vor- und Nachbearbeitung.
Für alle anderen Steuerungen stehen die Schritte 1 bis 95 zur Verfügung.

Grafcet-Anweisungen

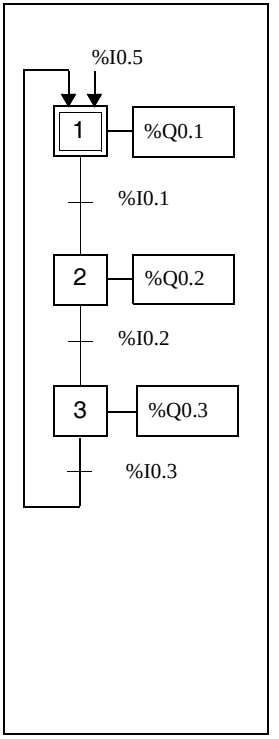
Die folgende Tabelle führt alle Anweisungen und Objekte auf, die zum Programmieren eines Grafcet-Graphen erforderlich sind:

Graphische Darstellung (1)	Transkription in TwidoSoft-Sprache	Funktion
Abbildung:  	=*= i	Ausgangsschritt starten (2)
	# i	Aktuellen Schritt deaktivieren und Schritt i aktivieren
	-*- i	Schritt i starten und zugehörige Transition validieren (2)
	#	Aktuellen Schritt deaktivieren, ohne andere Schritte zu aktivieren
	#Di	Schritt i und aktuellen Schritt deaktivieren
	=*= POST	Nachbearbeitung starten und Ablaufbearbeitung beenden
	%Xi	Zu Schritt i gehöriges Bit, kann getestet und geschrieben werden (die maximale Anzahl der Schritte richtet sich nach der Steuerung)
	LD %Xi, LDN %Xi AND %Xi, ANDN %Xi, OR %Xi, ORN %Xi XOR %Xi, XORN %Xi	Aktivität von Schritt i testen
 	S %Xi	Schritt i aktivieren
 	R %Xi	Schritt i deaktivieren

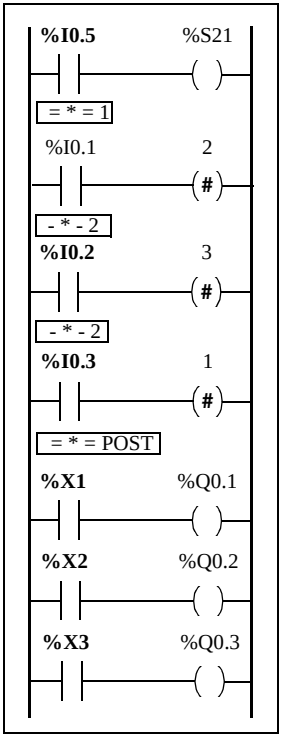
(1) Die graphische Darstellung wird nicht berücksichtigt.
(2) Der erste geschriebene Schritt =*=i oder -*-i kennzeichnet den Beginn der Ablaufbearbeitung und damit das Ende der Vorbearbeitung.

Grafcet-Beispiele

Linearer Ablauf:



Nicht unterstützt

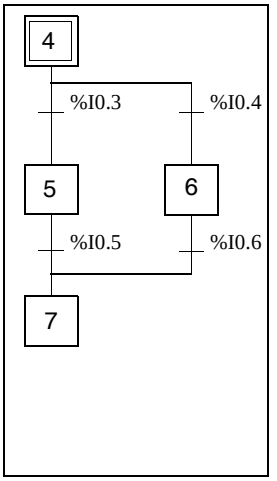


Twido-KOP-Programm

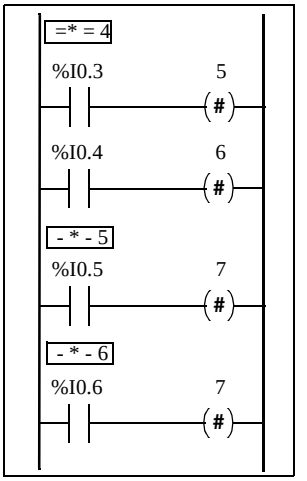
```
LD      %I0.5
ST      %S21
=*=     1
LD      %I0.1
 #      2
-*-     2
LD      %I0.2
 #      3
-*-     3
LD      %I0.3
 #      1
=*=   POST
LD      %X1
ST      %Q0.1
LD      %X2
ST      %Q0.2
LD      %X3
ST      %Q0.3
```

Twido-AWL-Programm

Alternativer Ablauf:



Nicht unterstützt

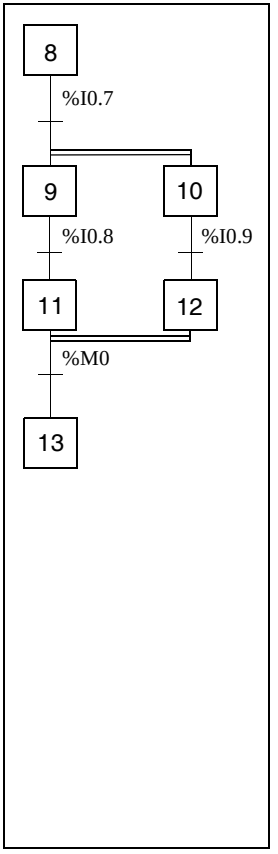


Twido-
KOP-Programm

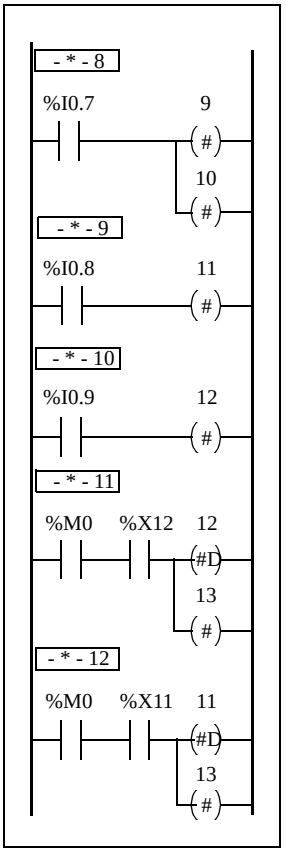
```
=*=     4
 LD     %I0.3
 #      5
 LD     %I0.4
 #      6

 -*-    5
 LD     %I0.5
 #      7

-*-     6
 LD     %I0.6
 #      7
```

Twido-
AWL-Programm

Gleichzeitiger Ablauf:



Nicht unterstützt



Twido-
KOP-Programm

```
 -*-      8
LD        %I0.7
#         9
#         10

-*-       9
LD        %I0.8
#         11

-*-       10
LD        %I0.9
#         12

-*-       11
LD        %M0
AND       %X12
#D        12
#         13

-*-       12
LD        %M0
AND       %X11
#D        11
#         13
```

Twido-
AWL-Programm

Hinweis: Damit ein Grafcet-Graph betriebsbereit ist, muss mindestens ein aktiver Schritt mit der Anweisung =*=i (Ausgangsschritt) deklariert werden oder der Graph muss während der Vorbearbeitung mit dem Systembit %S23 und der Anweisung S %Xi vorpositioniert werden.

Beschreibung des Grafcet-Programmaufbaus

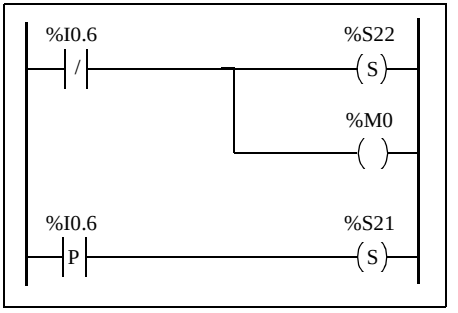
Einleitung

Ein TwidoSoft Grafcet-Programm läuft in drei Phasen ab:
- Vorbearbeitung
- Ablaufbearbeitung
- Nachbearbeitung

Vorbearbeitung

Während der Vorbearbeitung werden folgende Elemente generiert:
- Wiederkehr der Spannungsversorgung;
- Standardeinstellungen;
- Änderungen des Betriebszustands;
- Vorpositionierung von Grafcet-Schritten;
- Eingangslogik

Die steigende Flanke des Eingangs %I0.6 setzt das Bit %S21 auf 1. Dies bewirkt, dass die aktiven Schritte deaktiviert und die ursprünglichen Schritte aktiviert werden.




```
000     LDN     %I0.6
001     S       %S22
002     ST      %M0
003     LDR     %I0.6
004     S       %S21
```

Die Vorbearbeitung beginnt mit der ersten Zeile des Programms und endet mit dem ersten Auftreten einer Anweisung "= * =" oder "- * -".
Drei Systembits dienen der Grafcet-Steuerung: %S21, %S22 und %S23. Jedes Systembit wird während der allgemeinen Vorbearbeitung durch die Applikation auf 1 gesetzt (falls notwendig). Die zugehörige Funktion wird vom System am Ende der Vorbearbeitung ausgeführt, und das Systembit wird dann vom System auf 0 zurückgesetzt.

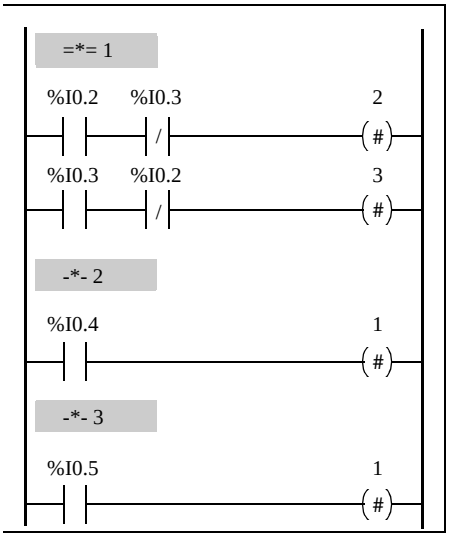
Systembit	Name	Beschreibung
%S21	Initialisierung des Grafcet	Alle aktiven Schritte werden deaktiviert und die Ausgangsschritte werden aktiviert.
%S22	Reinitialisierung des Grafcet	Alle Schritte werden deaktiviert.
%S23	Vorpositionierung des Grafcet	Dieses Bit muss auf 1 gesetzt werden, wenn die Objekte %Xi während der Vorbearbeitung explizit durch die Applikation geschrieben werden. Wenn dieses Bit während der Vorbearbeitung ohne explizite Änderung des Objekts %Xi auf 1 gehalten wird, wird das Grafcet starr (Aktualisierungen werden nicht berücksichtigt).

Ablaufbearbeitung

Die Ablaufbearbeitung wird im Graphen ausgeführt (Anweisungen, die den Graphen repräsentieren):

- Schritte
- Zu Schritten zugehörige Aktionen
- Transitionen
- Weiterschaltbedingungen

Beispiel:




```
005     =*=     1
006     LD      %I0.2
007     ANDN    %I0.3
008     #       2
009     LD      %I0.3
010     ANDN    %I0.2
011     #       3
012     -*-     2
013     LD      %I0.4
014     #       1
015     -*-     3
016     LD      %I0.5
017     #       1
```

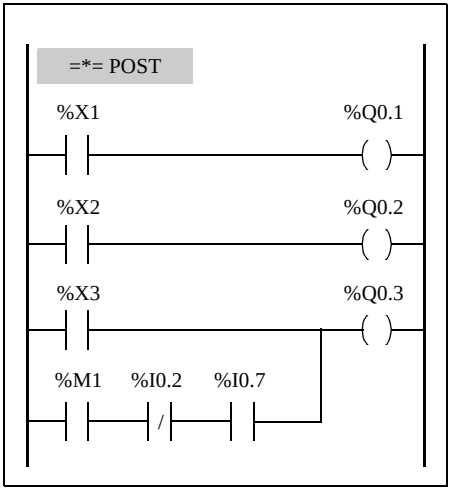
Die Ablaufbearbeitung endet mit der Ausführung der Anweisung "= * = POST" oder mit dem Ende des Programms.

Nachbearbeitung

Während der Nachbearbeitung werden folgende Elemente generiert:

- Befehle zur Ablaufbearbeitung für die Regelung der Ausgänge;
- Sicherheitsverriegelungen für die Ausgänge

Beispiel:




```
018     =*=     POST
019     LD      %X1
020     ST      %Q0.1
021     LD      %X2
022     ST      %Q0.2
023     LD      %X3
024     OR(     %M1
025     ANDN    %I0.2
026     AND     %I0.7
027     )
028     ST      %Q0.3
```

Zu Grafcet-Schritten gehörige Aktionen

Einführung

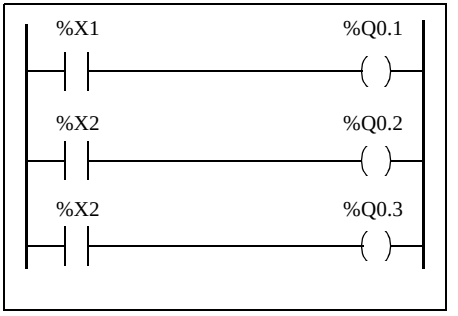
Ein TwidoSoft Grafcet-Programm bietet zwei Arten, um die zu Schritten gehörigen Aktionen zu programmieren:

- Im Abschnitt der Nachbearbeitung
- In AWL-Anweisungen oder KOP-Strompfaden der Schritte selbst

Zuweisen von Aktionen bei der Nachbearbeitung

Gibt es Sicherheitseinschränkungen oder Einschränkungen im Betriebszustand RUN, ist es besser, die Aktionen im Abschnitt der Nachbearbeitung einer Grafcet-Anwendung zu programmieren. Mit den AWL-Anweisungen zum Setzen und Rücksetzen oder durch Aktivieren von Spulen in einem KOP-Programm können Sie Grafcet-Schritte (%Xi) aktivieren.

Beispiel:

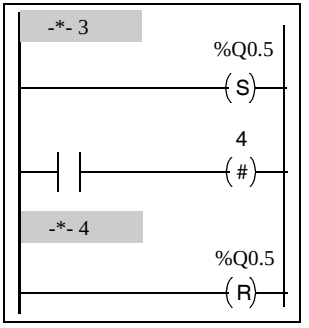



```
018     =*=     POST
019     LD      %X1
020     ST      %Q0.1
021     LD      %X2
022     ST      %Q0.2
023     LD      %X3
024     ST      %Q0.3
```

Zuweisen von Aktionen in einer Anwendung

Sie können die Aktionen, die zu Schritten gehören, in AWL-Anweisungen oder KOP-Strompfaden programmieren. In diesem Fall wird die AWL-Anweisung bzw. der KOP-Strompfad nur dann abgetastet, wenn der Schritt aktiv ist. Dies ist die effizienteste und am besten zu lesende und zu pflegende Art der Verwendung von Grafcet.

Beispiel:




```
020   -*-    3
021   LD     1
022   S      %Q0.5
023   LD     %M10
024   #      4
025   -*-    4
026   LD     1
027   R      %Q0.5
028   ...
029   ...
```

Beschreibung von Anweisungen und Funktionen

IV

Auf einen Blick

Inhalt dieses Abschnitts

Dieser Abschnitt enthält detaillierte Beschreibungen der Basis-Anweisungen und der erweiterten Anweisungen und die Systembits und -worte der Twido-Sprachen.

Inhalt dieses Teils

Dieser Teil enthält die folgenden Kapitel:

Kapitel	Kapitelname	Seite
16	Basis-Anweisungen	381
17	Erweiterte Anweisungen	453
18	Systembits und Systemworte	619

Basis-Anweisungen

16

Auf einen Blick

Inhalt dieses Kapitels

Dieses Kapitel enthält Detailinformationen über die Anweisungen und Funktionsblöcke, die zum Erstellen von Basis-Regelprogrammen für Twido -Steuerungen verwendet werden.

Inhalt dieses Kapitels

Dieses Kapitel enthält die folgenden Abschnitte:

Abschnitt	Thema	Seite
16.1	Boolesche Verarbeitung	383
16.2	Basis-Funktionsblöcke	401
16.3	Numerische Verarbeitung	426
16.4	Programmanweisungen	445

16.1 Boolesche Verarbeitung

Auf einen Blick

Inhalt des Abschnitts

Dieser Abschnitt enthält eine Einführung in die boolesche Verarbeitung. Sie basiert auf Beschreibungen und Programmierrichtlinien für boolesche Anweisungen.

Inhalt dieses Abschnitts

Dieser Abschnitt enthält die folgenden Themen:

Thema	Seite
Boolesche Anweisungen	384
Format für die Beschreibung von Booleschen Anweisungen	387
Ladeanweisungen (LD, LDN, LDR, LDF)	389
Zuweisungsbefehle (ST, STN, R, S)	391
Logische UND-Anweisungen (AND, ANDN, ANDR, ANDF)	393
Logische ODER-Anweisungen (OR, ORN, ORR, ORF)	395
exklusives OR, Anweisungen (XOR, XORN, XORR, XORF)	397
Anweisung NOT (N)	399

Boolesche Anweisungen

Einleitung

Die booleschen Anweisungen ähneln den graphischen Elementen der Programmiersprache KOP. Diese Anweisungen sind in folgender Tabelle beschrieben.

Element	Anweisung	Im Beispiel zur Bezugnahme	Beschreibung
Testelemente	Die Ladeanweisung (LD) entspricht einem Öffnerkontakt.	LD %I0.0	Der Kontakt ist geschlossen, wenn sich das Bit %I0.0 im Zustand 1 befindet.
Aktionselemente	Die Speicheranweisung (ST) entspricht einer Spule.	ST %Q0.0	Das zugehörige Bitobjekt nimmt einen logischen Wert des Bitakkumulators an (Ergebnis der vorherigen Logik).

Das Boolesche Ergebnis der Testelemente wird wie in den folgenden Anweisungen gezeigt auf die Aktionselemente angewendet.

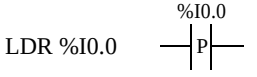
```
LD    %I0.0
AND   %I0.1
ST    %Q0.0
```

Testen von Steuerungseingängen

Mit Booleschen Testanweisungen können steigende oder fallende Flanken an den Eingängen der Steuerung erkannt werden. Eine Flanke wird erkannt, wenn sich der Zustand eines Eingangs zwischen dem "Zyklus n-1" und dem aktuellen "Zyklus n" geändert hat. Die Flanke bleibt während des aktuellen Zyklus erkannt.

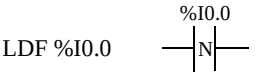
Erkennung steigende Flanke

Die Anweisung LDR (Steigende Flanke laden) entspricht dem Kontakt zur Erkennung einer steigenden Flanke. Die steigende Flanke erkennt den Zustandswechsel des Eingangswertes von 0 auf 1.
Ein Kontakt zur Erkennung einer steigenden Flanke erkennt eine steigende Flanke wie im folgenden Diagramm dargestellt.



Erkennung einer fallenden Flanke

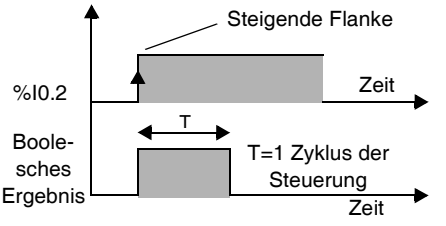
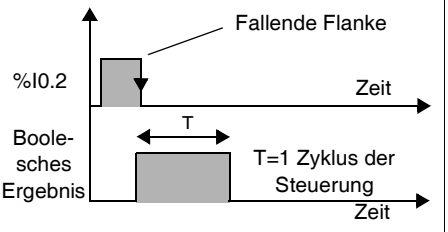
Die Anweisung LDF (Fallende Flanke laden) entspricht dem Kontakt zur Erkennung einer fallenden Flanke. Die fallende Flanke erkennt einen Zustandswechsel des steuernden Eingangswertes von 1 auf 0.
Ein Kontakt zur Erkennung einer fallenden Flanke erkennt eine fallende Flanke wie im folgenden Diagramm dargestellt.



N: Kontakt zur Erkennung fallender Flanken

Flankener-kennung

Die folgende Tabelle fasst die Anweisungen zur Flankenerkennung sowie deren Chronogramme zusammen.

Flanke	Testanweisung	Kontaktplan	Chronogramm
Steigende Flanke	LDR %I0.0	%I0.0 P	
Fallende Flanke	LDF %I0.0	%I0.0 N	

Hinweis: Es ist fortan möglich, die Flankenanweisungen auf die internen Bits %Mi anzuwenden.

Format für die Beschreibung von Booleschen Anweisungen

Einführung

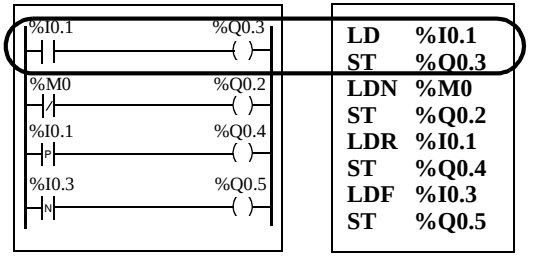
Jede Boolesche Anweisung in diesem Abschnitt wird mit Hilfe der folgenden Informationen beschrieben:

- Kurzbeschreibung
- Beispiel für die Anweisung und entsprechender Kontaktplan
- Liste der zugelassenen Operanden
- Zeitdiagramm

Die folgenden Erläuterungen bieten weitere Einzelheiten dazu, wie die Booleschen Anweisungen in diesem Abschnitt beschrieben werden.

Beispiele

Die folgende Abbildung zeigt, wie zu jeder Anweisung Beispiele geboten werden.



KOP-Entsprechungen AWL-Anweisungen

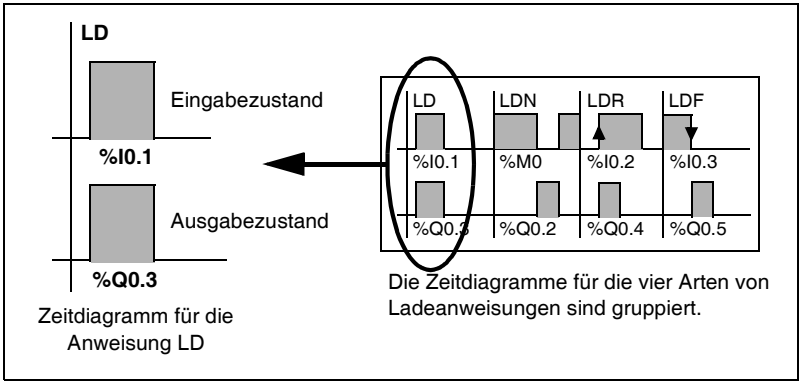
Zulässige Operanden

Die folgende Tabelle definiert die Arten der zulässigen Operanden für Boolesche Anweisungen.

Operand	Beschreibung
0/1	Direkter Wert 0 oder 1
%I	Steuerungseingang %Ii.j
%Q	Steuerungsausgang %Qi.j
%M	Internes Bit %Mi
%S	Systembit %Si
%X	Schrittbit %Xi
%BLK.x	Funktionsblockbit (z.B. %TMi.Q)
%•:Xk	Wortbit (z.B. %MWi:Xk)
[	Vergleichsausdruck (z.B. [%MWi<1000])

Zeitdiagramme

Die folgende Abbildung zeigt, wie zu jeder Anweisung Zeitdiagramme dargestellt werden.



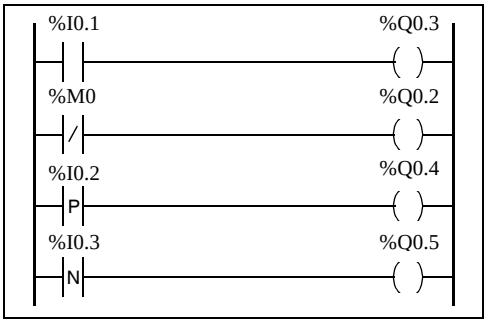
Ladeanweisungen (LD, LDN, LDR, LDF)

Einleitung

Die Ladeanweisungen LD, LDN, LDR und LDF entsprechen den Kontakten Öffnen, Schließen, steigende Flanke und fallende Flanke (LDR und LDF werden nur mit den Eingängen und internen Wörtern der Steuerung und den Eingängen der AS-Interface- und PDO CANopen-Slaves verwendet).

Beispiele

Die folgenden Diagramme sind Beispiele für Ladeanweisungen.



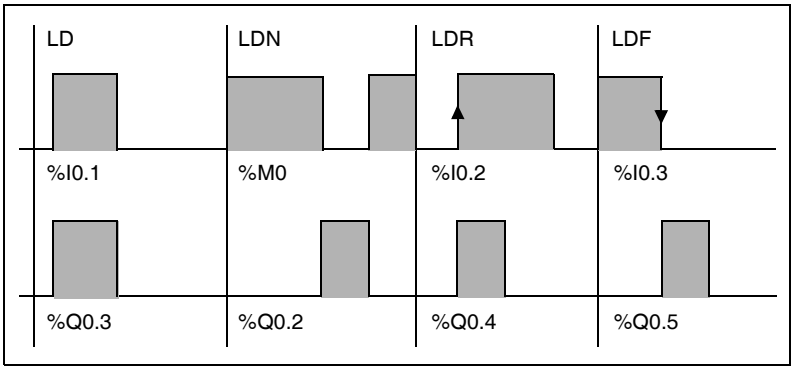

```
LD    %I0.1
ST    %Q0.3
LDN   %M0
ST    %Q0.2
LDR   %I0.2
ST    %Q0.4
LDF   %I0.3
ST    %Q0.5
```

Zulässige Operanden

Die folgende Tabelle führt die Arten von Ladeanweisungen mit den KOP-Entsprechungen und den zulässigen Operanden auf.

AWL-Anweisung	KOP-Entsprechung	Zulässige Operanden
LD		0/1, %I, %IA, %IWCx.y.z:Xk, %Q, %QA, %M, %S, %X, %BLK.x, %•:Xk,[
LDN		0/1, %I, %IA, %IWCx.y.z:Xk, %Q, %QA, %M, %S, %X, %BLK.x, %•:Xk,[
LDR		%I, %IA, %M
LDF		%I, %IA, %M

Chronogramm Das folgende Diagramm zeigt das Chronogramm der Ladeanweisungen.



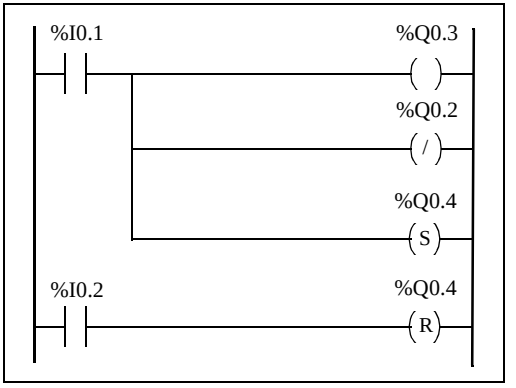
Zuweisungsbefehle (ST, STN, R, S)

Einleitung

Die Zuweisungsbefehle ST, STN, S und R entsprechen jeweils den direkten und inversen Spulen sowie den Spulen zum Setzen und Rücksetzen.

Beispiele

Die folgenden Diagramme sind Beispiele für Zuweisungsbefehle.



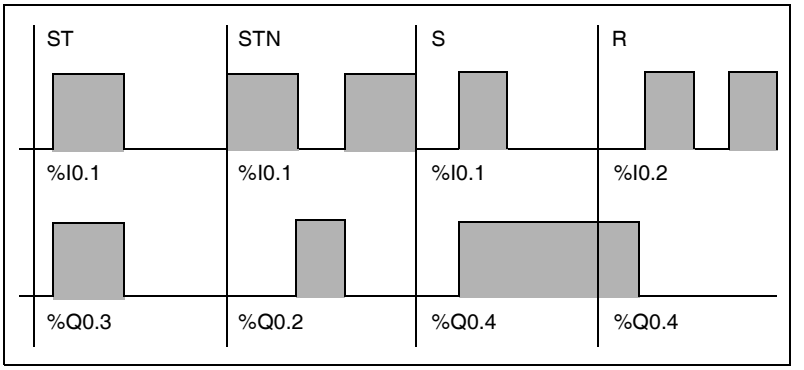

```
LD      %I0.1
ST      %Q0.3

STN     %Q0.2
S       %Q0.4

LD      %I0.2
R       %Q0.4
```

Zulässige Operanden

Die folgende Tabelle führt die Arten von Zuweisungsbefehlen mit den KOP-Entsprechungen und den zulässigen Operanden auf.

AWL-Anweisung	KOP-Entsprechung	Zulässige Operanden
ST	( )	%Q,%QA,%M,%S,%BLK.x,%•:Xk
STN	(/)	%Q,%QA%M,%S,%BLK.x,%•:Xk
S	(S)	%Q,%QA,%M,%S,%X,%BLK.x,%•:Xk
R	(R)	%Q,%QA,%M,%S,%X,%BLK.x,%•:Xk

Chronogramm Das folgende Diagramm zeigt das Chronogramm der Zuweisungsbefehle.



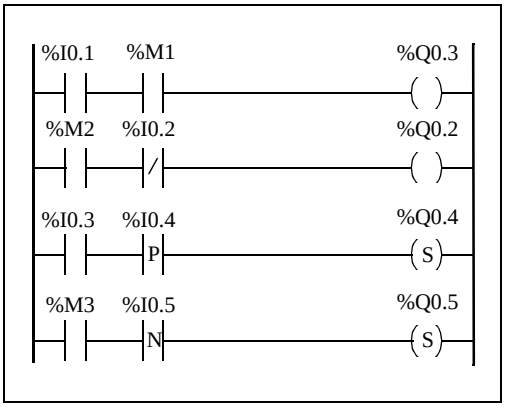
Logische UND-Anweisungen (AND, ANDN, ANDR, ANDF)

Einleitung

Die UND-Anweisungen führen eine logische UND-Verknüpfung zwischen dem Operanden (bzw. dem Inversen oder der steigenden oder der fallenden Flanke) und dem Booleschen Ergebnis der vorherigen Anweisung aus.

Beispiele

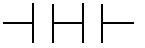
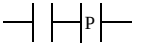
Die folgenden Diagramme sind Beispiele für logische UND-Anweisungen.



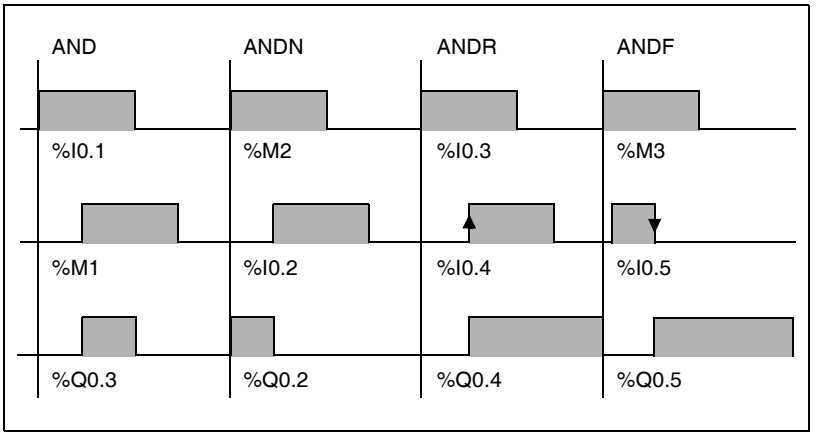

```
LD      %I0.1
AND     %M1
ST      %Q0.3
LD      %M2
ANDN    %I0.2
ST      %Q0.2
LD      %I0.3
ANDR    %I0.4
S       %Q0.4
LD      %M3
ANDF    %I0.5
S       %Q0.5
```

Zulässige Operanden

Die folgende Tabelle führt die Arten von UND-Anweisungen mit den KOP-Entsprechungen und den zulässigen Operanden auf.

AWL-Anweisung	KOP-Entsprechung	Zulässige Operanden
AND		0/1, %I, %IA, %Q, %QA, %M, %S, %X, %BLK.x, %•:Xk, [
ANDN		0/1, %I, %IA, %Q, %QA, %M, %S, %X, %BLK.x, %•:Xk, [
ANDR		%I, %IA, %M
ANDF		%I, %IA, %M

Chronogramm Das folgende Diagramm zeigt das Chronogramm für die UND-Anweisungen.



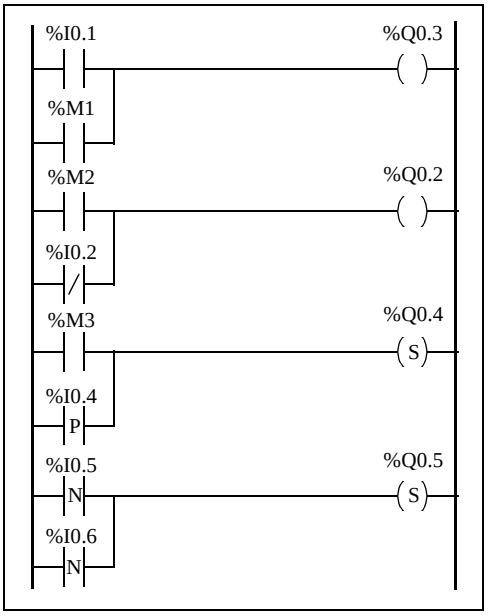
Logische ODER-Anweisungen (OR, ORN, ORR, ORF)

Einleitung

Die ODER-Anweisungen führen eine logische ODER-Verknüpfung zwischen dem Operanden (bzw. dem Inversen oder der steigenden oder der fallenden Flanke) und dem Booleschen Ergebnis der vorherigen Anweisung aus.

Beispiele

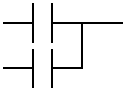
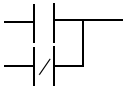
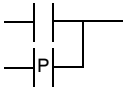
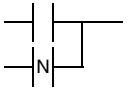
Die folgenden Diagramme sind Beispiele für logische ODER-Anweisungen.




```
LD      %I0.1
OR      %M1
ST      %Q0.3

LD      %M2
ORN     %I0.2
ST      %Q0.2

LD      %M3
ORR     %I0.4
S       %Q0.4

LDF     %I0.5
ORF     %I0.6
S       %Q0.5
```

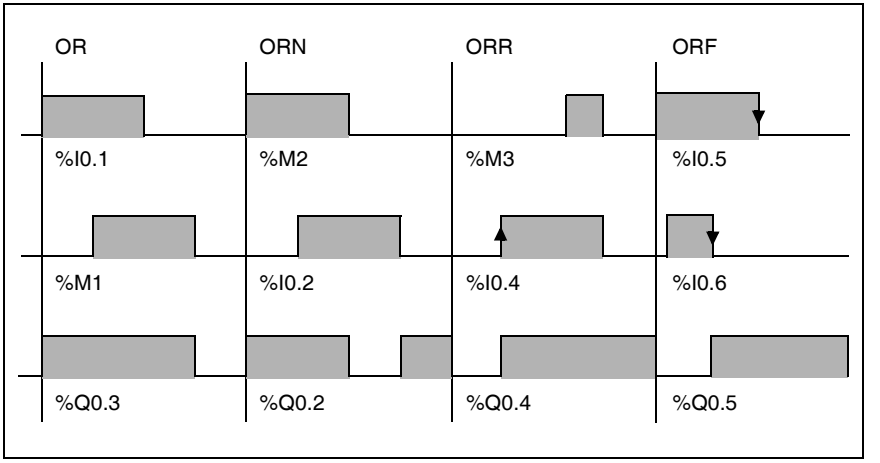
Zulässige Operanden

Die folgende Tabelle führt die Arten von ODER-Anweisungen mit den KOP-Entsprechungen und den zulässigen Operanden auf.

AWL-Anweisung	KOP-Entsprechung	Zulässige Operanden
OR		0/1, %I, %IA, %Q, %QA, %M, %S, %X, %BLK.x, %•:Xk
ORN		0/1, %I, %IA, %Q, %QA, %M, %S, %X, %BLK.x, %•:Xk
ORR		%I, %IA, %M
ORF		%I, %IA, %M

Chronogramm

Das folgende Diagramm zeigt das Chronogramm für die ODER-Anweisungen.



exklusives OR, Anweisungen (XOR, XORN, XORR, XORF)

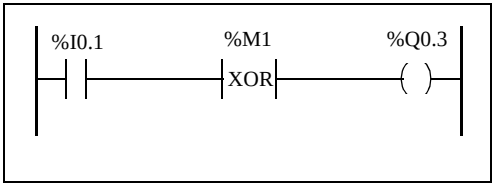
Einleitung

Die XOR-Anweisungen führen eine exklusive ODER-Verknüpfung zwischen dem Operanden (bzw. seines inversen Wertes oder seiner steigenden oder fallenden Flanke) und dem booleschen Ergebnis der vorherigen Anweisung aus.

Beispiele

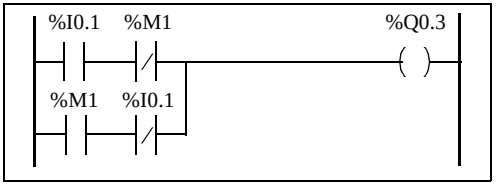
Das folgende Beispiel veranschaulicht die Verwendung der XOR-Anweisungen.

Schema mit XOR-Anweisungen :




```
LD    %I0.1
XOR   %M1
ST    %Q0.3
```

Schema ohne XOR-Anweisungen :



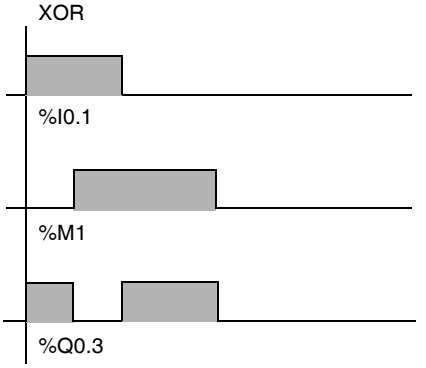

```
LD    %I0.1
ANDN  %M1
OR(   %M1
ANDN  %I0.1
)
ST    %Q0.3
```

Zulässige Operanden

Die folgende Tabelle führt die Arten von XOR-Anweisungen sowie die zulässigen Operanden auf.

AWL-Anweisung	Zulässige Operanden
XOR	%I, %IA, %Q, %QA, %M, %S, %X, %BLK.x, %•:Xk
XORN	%I, %IA, %Q, %QA, %M, %S, %X, %BLK.x, %•:Xk
XORR	%I, %IA, %M
XORF	%I, %IA, %M

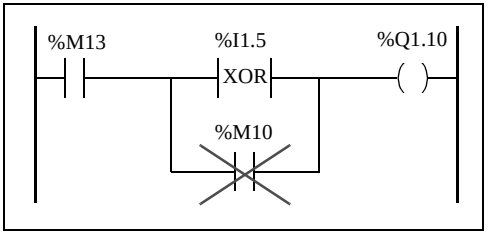
Chronogramm Das folgende Diagramm zeigt den Timeout für die XOR-Anweisungen.



Sonderfälle Beachten Sie die folgenden Hinweise bezüglich der Verwendung von XOR-Anweisungen in KOP-Programmen:

- Fügen Sie keinen XOR-Kontakt an der ersten Stelle eines Programmblocks ein.
- Fügen Sie keine XOR-Kontakte parallel zu anderen KOP-Elementen ein (siehe folgendes Beispiel).

Im folgenden Beispiel sehen Sie, dass durch Einfügen eines Elements parallel zum XOR-Kontakt ein Validierungsfehler auftritt.



Anweisung NOT (N)

Einführung

Die Anweisung NOT (N) negiert das Boolesche Ergebnis der vorherigen Anweisung.

Beispiel

Im Folgenden sehen Sie ein Beispiel für die Anweisung NOT.

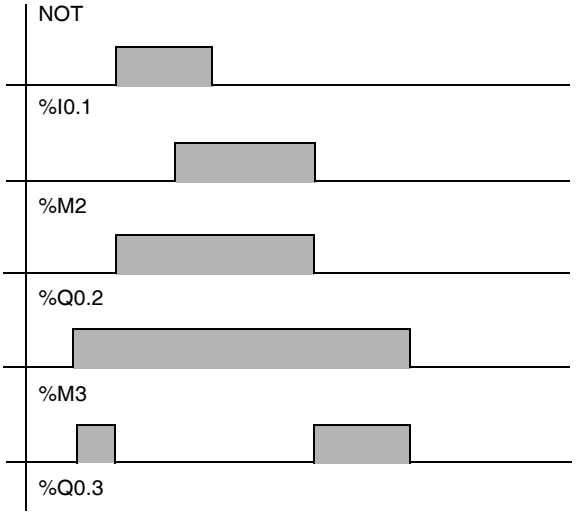
```
LD      %I0.1
OR      %M2
ST      %Q0.2
N
AND     %M3
ST      %Q0.3
```

Hinweis: Die Anweisung NOT ist nicht reversibel.

Zulässige Operanden

Nicht anwendbar.

Zeitdiagramm Das folgende Diagramm zeigt die Zeitmessung für die Anweisung NOT.



16.2 Basis-Funktionsblöcke

Auf einen Blick

Inhalt des Abschnitts

Dieser Abschnitt enthält eine Übersicht der Beschreibungen und Programmiertipps für Basis-Funktionsblöcke.

Inhalt dieses Abschnitts

Dieser Abschnitt enthält die folgenden Themen:

Thema	Seite
Basis-Funktionsblöcke	402
Prinzipien für die Programmierung von Standard-Funktionsblöcken	404
Timer-Funktionsblock (%TMi)	406
Timer-Typ TOF	408
Timer-Typ TON	409
Timer-Typ TP	410
Programmieren und Konfigurieren von Timern	411
Auf-/Abwärtszähler-Funktionsblock (%Ci)	414
Programmierung und Konfiguration der Zähler	418
Schiebebitregister-Funktionsblock (%SBRi)	420
Schrittzähler-Funktionsblock (%SCi)	422

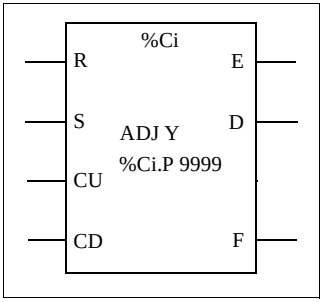
Basis-Funktionsblöcke

Einleitung

Funktionsblöcke sind die Quellen für Bitobjekte und bestimmte Wörter, die von Programmen verwendet werden. Basis-Funktionsblöcke bieten einfache Funktionen wie Timer oder Auf-/Abwärtszähler.

Beispiel für einen Funktionsblock

Im Folgenden sehen Sie eine Abbildung eines Auf-/Abwärtszähler-Funktionsblocks.



Auf-/Abwärtszähler-Funktionsblock

Bitobjekte

Bitobjekte entsprechen den Blockausgängen. Auf diese Bits können Boolesche Testanweisungen auf eine der folgenden Arten zugreifen:

- Direkt (z.B. LD E), wenn sie in der reversiblen Programmierung mit dem Block verdrahtet sind (siehe *Prinzipien für die Programmierung von Standard-Funktionsblöcken, S. 404*).
- Durch Angabe des Blocktyps (z.B. LD %Ci.E).

Auf Eingänge kann in Form von Anweisungen zugegriffen werden.

Wortobjekte

Wortobjekte entsprechen bestimmten Parametern und Werten:

- **Blockkonfigurationsparameter:** Einige Parameter sind für das Programm zugänglich (z.B. Vorauswahl von Parametern) und einige sind nicht für das Programm zugänglich (z.B. Zeitbasis).
- **Aktuelle Werte:** z.B. %Ci.V, der aktuelle Zählwert.

Zugängliche Bit- und Wortobjekte

Die folgende Tabelle beschreibt die Bit- und Wortobjekte von Basis-Funktionsblöcken, auf die das Programm zugreifen kann.

Basis-Funktionsblock	Symbol	Bereich (i)	Objekttypen	Beschreibung	Adresse	Schreibzugriff
Timer	%TMi	0 - 127	Wort	Aktueller Wert	%TMi.V	Nein
				Voreingestellter Wert	%TMi.P	Ja
			Bit	Timer-Ausgang	%TMi.Q	Nein
Auf-/Abwärtszähler	%Ci	0 - 127	Wort	Aktueller Wert	%Ci.V	Nein
				Voreingestellter Wert	%Ci.P	Ja
			Bit	Unterlauf-Ausgang (leer)	%Ci.E	Nein
				Preset-Ausgang erreicht	%Ci.D	Nein
				Überlauf-Ausgang (voll)	%Ci.F	Nein

Prinzipien für die Programmierung von Standard-Funktionsblöcken

Einleitung

Gehen Sie auf eine der folgenden Arten vor, um Standard-Funktionsblöcke zu programmieren:

- Funktionsblock-Anweisungen (z.B. `BLK %TM2`): Diese reversible Technik zum Programmieren eines Kontaktplans ermöglicht, dass Befehle für den Block an einer einzigen Stelle im Programm ausgeführt werden.
- Spezifische Anweisungen (z.B. `CU %Ci`): Diese nicht reversible Technik ermöglicht, dass Befehle für die Eingänge des Blocks an mehreren Stellen im Programm ausgeführt werden (z.B. `Zeile 100 CU %C1`, `Zeile 174 CD %C1`, `Zeile 209 LD %C1.D`).

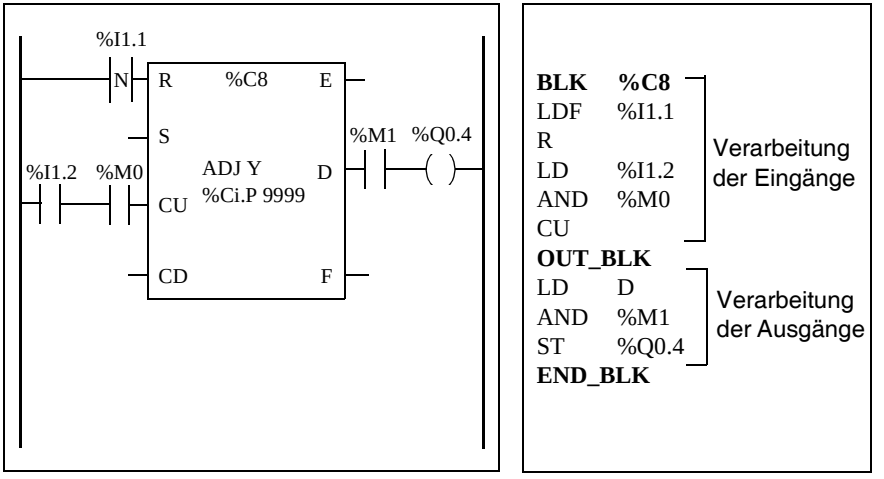
Reversible Programmierung

Verwenden Sie die Anweisungen BLK, OUT_BLK und END_BLK für die reversible Programmierung:

- **BLK:** Zeigt den Beginn des Blocks an.
- **OUT_BLK:** Verdrahtet die Ausgänge des Blocks direkt.
- **END_BLK:** Zeigt das Ende des Blocks an.

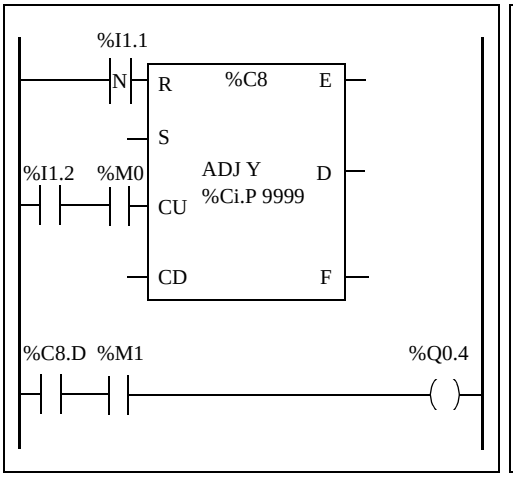
Beispiel mit Ausgangsverdrahtung

Das folgende Beispiel zeigt die reversible Programmierung eines Zähler-Funktionsblocks mit verdrahteten Ausgängen.



Beispiel ohne Ausgangsverdrahtung

Dieses Beispiel zeigt die reversible Programmierung eines Zähler-Funktionsblocks ohne verdrahtete Ausgänge.




```
BLK     %C8
LDF     %I1.1
R
LD      %I1.2
AND     %M0
CU
END_BLK
LD      %C8.D
AND     %M1
ST      %Q0.4
```

Verarbeitung der Eingänge

Verarbeitung der Ausgänge

Hinweis: Nur Test- und Eingangsanweisungen für den relevanten Block können zwischen den Anweisungen BLK und OUT_BLK (oder zwischen BLK und END_BLK, wenn OUT_BLK nicht programmiert ist) angeordnet werden.

Timer-Funktionsblock (%TMi)

Einleitung

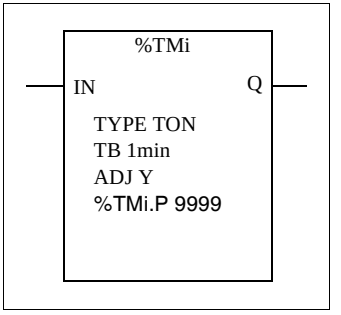
Es gibt drei Arten von Timer-Funktionsblöcken:

- TON (Einschaltverzögerungszeit): Mit dieser Art von Timer können Sie Einschaltverzögerungen verwalten.
- TOF (Ausschaltverzögerungszeit): Mit dieser Art von Timer können Sie Ausschaltverzögerungen verwalten.
- TP (Timer – Impuls): Mit diesem Timer erstellen Sie einen Impuls mit präziser Länge.

Die Verzögerungen oder Impulsdauern sind programmierbar und können mit TwidoSoft geändert werden.

Abbildung

Im Folgenden sehen Sie eine Abbildung des Timer-Funktionsblocks.



Timer-Funktionsblock

Einstellungen Der Timer-Funktionsblock verfügt über die folgenden Parameter:

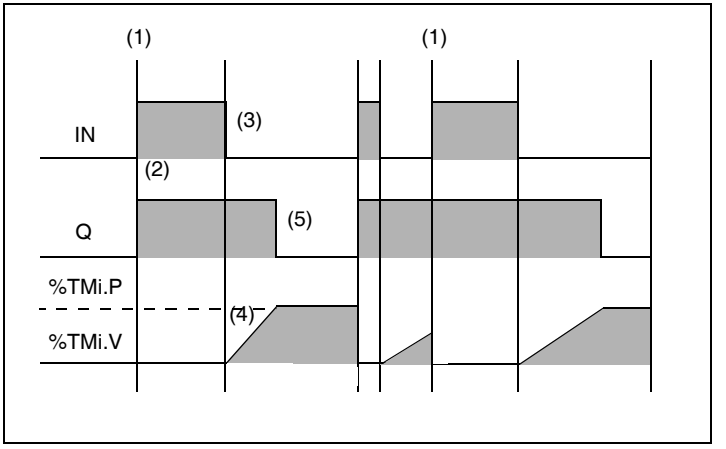
Parameter	Beschriftung	Wert
Timer-Nummer	%TMi	0 bis 63: TWDLCAA10DRF und TWDLCAA16DRF 0 bis 127 bei allen anderen Steuerungen.
Typ	TON	• Einschaltverzögerung (standardmäßig)
	TOF	• Ausschaltverzögerung
	TP	• Impuls (Monoflop)
Zeitbasis	TB	1 min (standardmäßig), 1 s, 100 ms, 10 ms, 1 ms
Aktueller Wert	%TMi.V	Wort, das von 0 bis %TMi.P inkrementiert wird, wenn der Timer läuft. Das Wort kann gelesen und getestet werden, es kann jedoch nicht vom Programm geschrieben werden. %TMi.V kann mit dem Animationstabellen-Editor gerändert werden.
Voreingestellter Wert	%TMi.P	0 - 9999. Wort, das gelesen, getestet und vom Programm geschrieben werden kann. Der Standardwert ist 9999. Die generierte Periode oder die Verzögerung entspricht %TMi.P x TB.
Animationstabelleneditor	Y/N	Y: Ja, der voreingestellte Wert %TMi.P kann mit dem Animationstabellen-Editor geändert werden. N: Nein, der voreingestellte Wert %TMi.P kann nicht geändert werden.
Freigabeeingang (oder Anweisungseingang)	IN	Startet den Timer bei einer steigenden Flanke (Typen TON oder TP) bzw. bei einer fallenden Flanke (Typ TOF).
Timer-Ausgang	Q	Das zugehörige Bit %TMi.Q wird auf 1 gesetzt, je nach der ausgeführten Funktion: TON, TOF oder TP.

Hinweis: Je größer der voreingestellte Wert, desto höher ist die Genauigkeit des Timers.

Timer-Typ TOF

Einführung Mit dem Timer-Typ TOF (Ausschaltverzögerungszeit) steuern Sie Ausschaltverzögerungen. Diese Verzögerung ist mit TwidoSoft programmierbar.

Zeitdiagramm Das folgende Zeitdiagramm zeigt die Funktionsweise des TOF-Timers.



Funktionsweise Die folgende Tabelle beschreibt die Funktionsweise des TOF-Timers.

Phase	Beschreibung
1	Der aktuelle Wert %TMi.V wird bei einer steigenden Flanke an Eingang IN auf 0 gesetzt, auch wenn der Timer läuft.
2	Das Ausgangsbit %TMi.Q wird bei auf 1 gesetzt, wenn am Eingang N eine steigende Flanke erkannt wird.
3	Der Timer wird bei einer fallenden Flanke an Eingang IN gestartet.
4	Der aktuelle Wert %TMi.V wird bei jedem Impuls der Zeitbasis TB in Inkrementen von einer Einheit auf %TMi.P erhöht.
5	Das Ausgangsbit %TMi.Q wird auf 0 zurückgesetzt, wenn der aktuelle Wert %TMi.P erreicht.

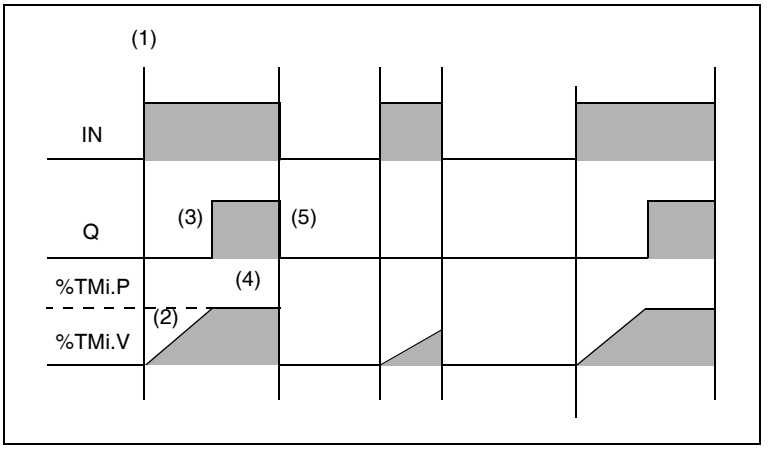
Timer-Typ TON

Einführung

Mit dem Timer-Typ TON (Einschaltverzögerungszeit) steuern Sie Einschaltverzögerungen. Diese Verzögerung ist mit TwidoSoft programmierbar.

Zeitdiagramm

Das folgende Zeitdiagramm zeigt die Funktionsweise des TON-Timers.



Funktionsweise

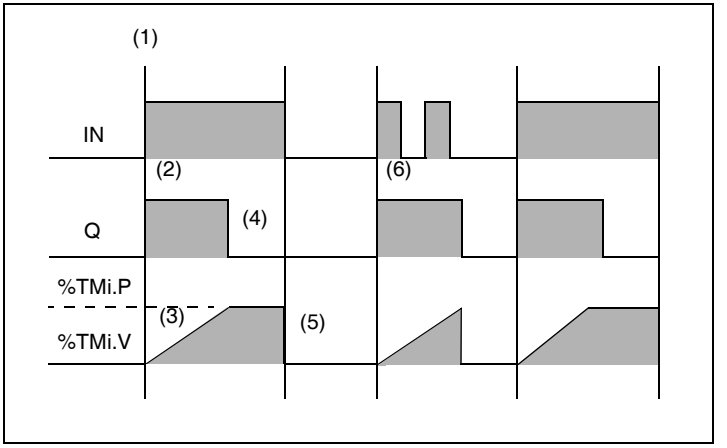
Die folgende Tabelle beschreibt die Funktionsweise des TON-Timers.

Phase	Beschreibung
1	Der Timer wird bei einer steigenden Flanke an Eingang IN gestartet.
2	Der aktuelle Wert %TMi.V wird bei jedem Impuls der Zeitbasis TB in Inkrementen von 0 auf %TMi.P erhöht.
3	Das Ausgangsbit %TMi.Q wird auf 1 gesetzt, wenn der aktuelle Wert %TMi.P erreicht hat.
4	Das Ausgangsbit %TMi.Q bleibt auf 1, solange der Eingang IN auf 1 ist.
5	Wird am Eingang IN eine fallende Flanke erkannt, wird der Timer gestoppt, auch wenn der Timer noch nicht %TMi.P erreicht hat, und %TMi.V wird auf 0 gesetzt.

Timer-Typ TP

Einleitung Mit dem Timer-Typ TP (Timer - Impuls) erzeugen Sie Impulse mit präziser Länge. Diese Verzögerung ist mit TwidoSoft programmierbar.

Zeitdiagramm Das folgende Zeitdiagramm zeigt die Funktionsweise des TP-Timers.



Funktionsweise Die folgende Tabelle beschreibt die Funktionsweise des TP-Timers.

Phase	Beschreibung
1	Der Timer wird bei einer steigenden Flanke an Eingang IN gestartet. Der aktuelle Wert %TMi.V wird auf 0 gesetzt, wenn der Timer noch nicht gestartet ist.
2	Das Ausgangsbit %TMi.Q wird auf 1 gesetzt, wenn der Timer startet.
3	Der aktuelle Wert %TMi.V des Timers wird bei jedem Impuls der Zeitbasis TB in Inkrementen von einer Einheit von 0 auf %TMi.P erhöht.
4	Das Ausgangsbit %TMi.Q wird auf 0 gesetzt, wenn der aktuelle Wert %TMi.P erreicht hat.
5	Der aktuelle Wert %TMi.V wird auf 0 gesetzt, wenn %TMi.V gleich %TMi.P ist und Eingang IN auf 0 zurückgeht.
6	Der Timer kann nicht auf Null zurückgesetzt werden. Wenn %TMi.V gleich %TMi.P ist und der Eingang IN auf 0 gesetzt wird, wird %TMi.V auf 0 gesetzt.

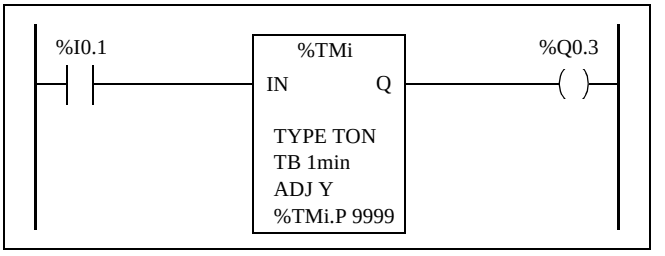
Programmieren und Konfigurieren von Timern

Einführung

Timer-Funktionsblöcke (%TMi) werden immer gleich programmiert, ganz gleich, wie sie eingesetzt werden sollen. Die Timer-Funktion (TON, TOF oder TP) wird während der Konfiguration gewählt.

Beispiele

Das folgende Bild zeigt einen Timer-Funktionsblock mit Beispielen für reversible und nicht reversible Programmierung.



Reversible Programmierung

```
BLK     %TM1
LD      %I0.1
IN
OUT_BLK
LD      Q
ST      %Q0.3
END_BLK
```

Nicht reversible Programmierung

```
LD      %I0.1
IN      %TM1
LD      %TM1.Q
ST      %Q0.3
```

Konfiguration

Die folgenden Parameter müssen während der Konfiguration eingegeben werden:

- Timer-Typ: TON, TOF oder TP
- Zeitbasis (TB): 1 min, 1 s, 100 ms, 10 ms oder 1 ms
- Vorwahlwert (%TMi.P): 0 bis 9999
- Einstellung: Geprüft oder Nicht geprüft

Sonderfälle Die folgende Tabelle enthält eine Liste mit Sonderfällen für die Programmierung des Timer-Funktionsblocks.

Sonderfall	Beschreibung
Auswirkung eines Kaltstarts (%S0=1)	Setzt den aktuellen Wert auf 0. Setzt den Ausgang %TMi.Q auf 0. Der voreingestellte Wert wird während der Konfiguration auf den definierten Wert gesetzt.
Auswirkung eines Warmstarts (%S1=1)	Hat keine Auswirkung auf den aktuellen und den voreingestellten Wert des Timers. Der aktuelle Wert wird während eines Spannungsausfalls nicht verändert.
Auswirkungen eines Steuerungsstopps	Beim Stoppen der Steuerung wird der aktuelle Wert nicht eingefroren.
Auswirkungen eines Programmsprungs	Durch einen Sprung über den Timer-Block wird der Timer nicht eingefroren. Der Timer wird weiter inkrementiert, bis der voreingestellte Wert (%TMi.P) erreicht ist. Zu diesem Zeitpunkt wechselt das Bit "Fertig" (%TMi.Q), das dem Ausgang Q des Timer-Blocks zugewiesen ist, den Zustand. Der zugehörige Ausgang jedoch, der direkt mit dem Blockausgang verdrahtet ist, wird von der Steuerung nicht aktiviert und nicht abgetastet.
Testen von Bit %TMi.Q (Bit "Fertig")	Bit %TMi.Q sollte nur einmal im Programm getestet werden.
Auswirkung einer Änderung der Voreinstellung %TMi.P	Wenn der voreingestellte Wert durch eine Anweisung oder durch Anpassen des Werts geändert wird, wirkt sich dies erst bei der nächsten Aktivierung des Timers aus.

Timer mit einer Zeitbasis von 1 ms

Die Zeitbasis von 1 ms ist nur bei den ersten fünf Timern verfügbar. Die vier Systemworte %SW76, %SW77, %SW78 und SW79 können als "Sanduhr" verwendet werden. Diese vier Worte werden jede Millisekunde vom System einzeln dekrementiert, **sofern sie einen positiven Wert besitzen.**
Mehrfach-Zeitmessung kann durch sukzessives Laden eines dieser Systemworte oder durch Testen des Zwischenberechnungswerts erreicht werden. Ist der Wert eines dieser vier Worte kleiner als 0, wird er nicht geändert. Ein Timer kann "eingefroren" werden, indem das entsprechende Bit 15 auf 1 gesetzt wird. Das Einfrieren wird rückgängig gemacht durch Rücksetzen des Werts auf 0.

Programmier-beispiel

Im Folgenden sehen Sie ein Beispiel für einen Timer-Funktionsblock.

```
LDR     %I0.1          (Starten des Timers bei einer steigenden Flanke an
%I0.1)
[%SW76:=XXXX]          (XXXX = erforderlicher Wert)
LD      %I0.2          (optionales Freeze-Management, Eingang I0.2 wird
eingefroren)
ST      %SW76:X15
LD      [%SW76=0]      (Testen Timer-Ende)
ST      %M0
.....
```

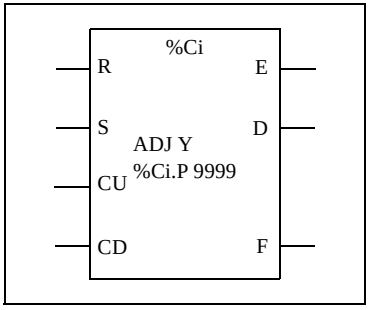
Auf-/Abwärtszähler-Funktionsblock (%Ci)

Einleitung

Der Zähler-Funktionsblock (%Ci) bietet Aufwärts- und Abwärtszählen von Ereignissen. Diese beiden Funktionen können gleichzeitig ausgeführt werden.

Abbildung

Im Folgenden sehen Sie eine Abbildung des Auf-/Abwärtszähler-Funktionsblocks.



Auf-/Abwärtszähler-Funktionsblock

Einstellungen Der Zähler-Funktionsblock hat die folgenden Parameter:

Parameter	Beschriftung	Wert
Zählernummer	%Ci	0 à 127
Aktueller Wert	%Ci.V	Das Wort wird entsprechend den Eingängen (oder Anweisungen) CU und CD inkrementiert oder dekrementiert. Das Wort kann gelesen und getestet werden, es kann jedoch nicht vom Programm geschrieben werden. Zum Ändern von %Ci.V verwenden Sie den Daten-Editor.
Voreingestellter Wert	%Ci.P	0 ≤ %Ci.P ≤ 9999. Das Wort kann gelesen, getestet und geschrieben werden (Standardwert: 9999).
Bearbeitung mit dem Animationstabellen-Editor	ADJ	• Y: Ja, der voreingestellte Wert kann mit dem Animationstabellen-Editor geändert werden. • N: Nein, der voreingestellte Wert kann nicht mit dem Animationstabellen-Editor geändert werden.
RESET-Eingang (oder Anweisung)	R	Im Zustand 1: %Ci.V = 0.
RESET-Eingang (oder Anweisung)	S	Im Zustand 1: %Ci.V = %Ci.P.
Aufwärtszähleingang (oder Anweisung)	CU	Inkrementiert %Ci.V bei einer steigenden Flanke.
Abwärtszähleingang (oder Anweisung)	CD	Dekrementiert %Ci.V bei einer steigenden Flanke.
Abwärtszählungs-Überlaufausgang	E (Leer)	Das zugehörige Bit %Ci.E=1, wenn der Abwärtszähler %Ci.V von 0 auf 9999 wechselt (wird auf 1 gesetzt, wenn %Ci.V den Wert 9999 erreicht, und wird auf 0 zurückgesetzt, wenn der Zähler weiter abwärts zählt).
Preset-Ausgang erreicht	D (Fertig)	Das zugehörige Bit %Ci.D=1, wenn %Ci.V=%Ci.P.
Aufwärtszählungs-Überlaufausgang	F (Voll)	Das zugehörige Bit %Ci.F=1, wenn %Ci.V von 9999 auf 0 wechselt (wird auf 1 gesetzt, wenn %Ci.V den Wert 0 erreicht, und wird auf 0 zurückgesetzt, wenn der Zähler weiter aufwärts zählt).

Funktionsweise Die folgende Tabelle beschreibt die wesentlichen Phasen des Auf-/Abwärtszählers.

Funktionsweise	Aktion	Ergebnis
Aufwärtszählen	Eine steigende Flanke tritt am Aufwärtszähleingang CU auf (oder die Anweisung CU wird aktiviert).	Der aktuelle Wert %Ci.V wird um eine Einheit inkrementiert.
	Der aktuelle Wert %Ci.V ist gleich dem voreingestellten Wert %Ci.P.	Das Ausgangsbit "Voreinstellung erreicht" %Ci.D wird auf 1 gesetzt.
	Der aktuelle Wert %Ci.V wechselt von 9999 auf 0.	Das Ausgangsbit %Ci.F (Aufwärtszählungs-Überlauf) wird auf 1 gesetzt.
	Wenn der Zähler weiter aufwärts zählt.	Das Ausgangsbit %Ci.F (Aufwärtszählungs-Überlauf) wird auf 0 zurückgesetzt.
Abwärtszählen	Eine steigende Flanke tritt am Abwärtszähleingang CD auf (oder die Anweisung CD wird aktiviert).	Der aktuelle Wert %Ci.V wird um eine Einheit dekrementiert.
	Der aktuelle Wert %Ci.V wechselt von 0 auf 9999.	Das Ausgangsbit %Ci.E (Abwärtszählungs-Überlauf) wird auf 1 gesetzt.
	Wenn der Zähler weiter abwärts zählt.	Das Ausgangsbit %Ci.F (Abwärtszählungs-Überlauf) wird auf 0 zurückgesetzt.
Auf-/ Abwärtszählen	Wenn Sie die Aufwärts- und die Abwärtszählfunktion gleichzeitig verwenden möchten (bzw. wenn Sie beiden Anweisungen CD und CU gleichzeitig aktivieren möchten), müssen die beiden entsprechenden Eingänge CU und CD simultan gesteuert werden. Diese beiden Eingänge werden anschließend abgetastet. Sind beide im Zustand 1, wird der aktuelle Wert nicht verändert.	
Rücksetzen	Setzen des Eingangs R auf 1 (oder die Anweisung R wird aktiviert).	Forciert das Zurücksetzen des Werts %Ci.V auf Null. Die Ausgänge %Ci.E, %Ci.D und %Ci.F werden auf 0 gesetzt. Der auf Null zurückgesetzte Eingang hat Vorrang.
Preset	Wenn der Eingang S auf 1 gesetzt wird (oder wenn die Anweisung S aktiviert wird) und der Rücksetzeingang den Status 0 aufweist (oder wenn die Anweisung R aktiviert wird).	Der aktuelle Wert %Ci.V übernimmt den Wert %Ci.P und der Ausgang %Ci.D wird auf 1 gesetzt.

Sonderfälle Die folgende Tabelle enthält eine Liste der Sonderfällen für die Funktionsweise und die Konfiguration der Zähler.

Sonderfall	Beschreibung
Auswirkung eines Kaltstarts (%S0=1)	• Der aktuelle Wert %Ci.V wird auf 0 gesetzt. • Die Ausgangsbits %Ci.E, %Ci.D und %Ci.F werden auf 0 gesetzt. • Der voreingestellte Wert wird mit dem während der Konfiguration definierten Wert initialisiert.
Auswirkung eines Warmstarts (%S1=1) bei Steuerungsstopp (STOP)	Hat keine Auswirkungen auf den aktuellen Wert des Zählers (%Ci.V).
Auswirkung einer Änderung der Voreinstellung %Ci.P	Wird der voreingestellte Wert durch eine Anweisung oder durch Anpassung geändert, wirkt sich dies aus, wenn der Block von der Applikation verarbeitet wird (einer der Eingänge wird aktiviert).

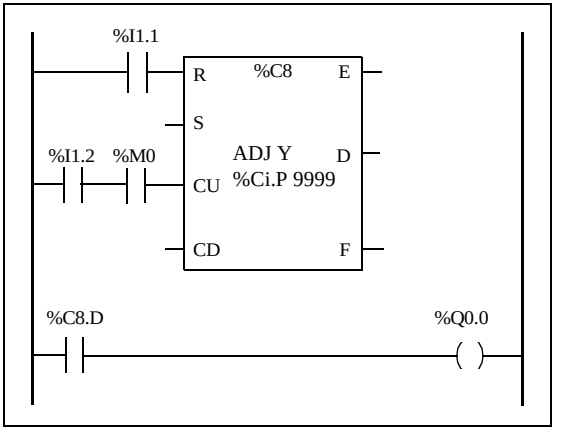
Programmierung und Konfiguration der Zähler

Einleitung

Das folgende Beispiel beschreibt einen Zähler, der das Zählen von maximal 5.000 Artikeln ermöglicht. Jeder Impuls am Eingang %I1.2 (wenn das interne Bit %M0 auf 1 gesetzt wird) inkrementiert den Wert des Zählers %C8 um eine Einheit, bis der Vorwahlendwert erreicht ist (Bit %C8.D=1). Der Zähler wird vom Eingang %I1.1 auf Null zurückgesetzt.

Programmier-beispiel

Die folgende Abbildung zeigt einen Zähler-Funktionsblock mit Beispielen für reversible und nicht reversible Programmierung.



Kontaktplan

```
BLK     %C8
LD      %I1.1
R
LD      %I1.2
AND     %M0
CU
END_BLK
LD      %C8.D
```

Reversible Programmierung

```
LD      %I1.1
R       %C8
LD      %I1.2
AND     %M0
CU      %C8
LD      %C8.D
ST      %Q0.0
```

Nicht reversible Programmierung

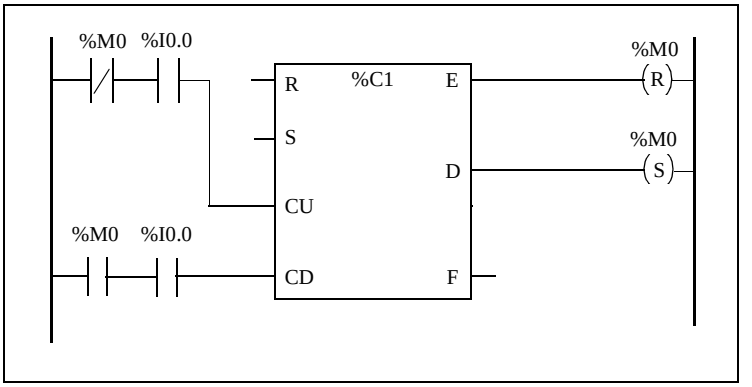
Konfiguration

Die folgenden Parameter müssen während der Konfiguration eingegeben werden:

- Vorwahlwert (%Ci.P): in diesem Beispiel auf 5.000 festgelegt
- Einstellung: Ja

Beispiel eines Auf-/ Abwärtszählers

Die folgende Abbildung zeigt einen Auf-/Abwärtszähler-Funktionsblock.



Kontaktplan

In diesem Beispiel wird, wenn man %C1.P 4 nimmt, der aktuelle Wert des Zählers %C1.V von 0 bis 3 inkrementiert und anschließend von 3 bis 0 dekrementiert. Solange %I0.0=1 pendelt %C1.V zwischen 0 und 3.

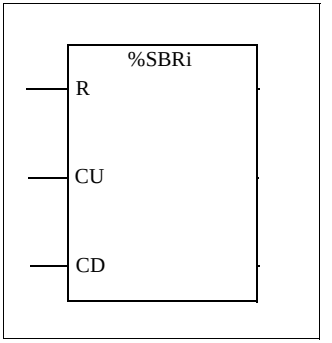
Schiebebitregister-Funktionsblock (%SBRi)

Einführung

Der Schiebebitregister-Funktionsblock (%SBRi) schiebt binäre Datenbits (0 oder 1) nach links oder nach rechts.

Abbildung

Im Folgenden sehen Sie ein Beispiel für einen Schieberegister-Funktionsblock.



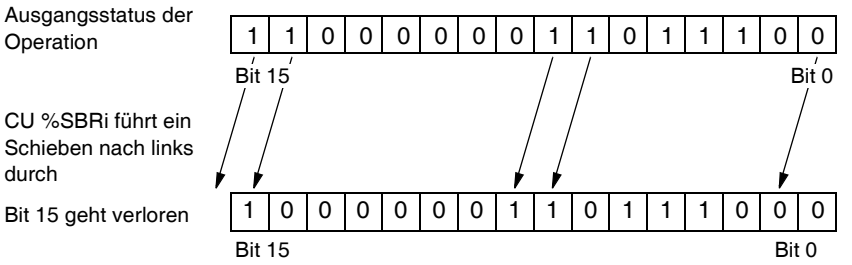
Einstellungen

Der Schiebebitregister-Funktionsblock hat die folgenden Parameter:

Parameter	Markierung	Wert
Registernummer	%SBRi	0 bis 7
Registerbit	%SBRi.j	Die Bits 0 bis 15 (j = 0 bis 15) des Schieberegisters können mit einer Testanweisung getestet und mit einer Zuweisungsanweisung geschrieben werden.
RESET-Eingang (oder Anweisung)	R	Wenn der Funktionsparameter R "1" ist, werden hierdurch die Registerbits 0 bis 15 %SBRi.j auf "0" gesetzt.
Eingang für Schieben nach links (bzw. Anweisung)	CU	Bei einer steigenden Flanke wird ein Registerbit nach links geschoben.
Eingang für Schieben nach rechts (bzw. Anweisung)	CD	Bei einer steigenden Flanke wird ein Registerbit nach rechts geschoben.

Befehl

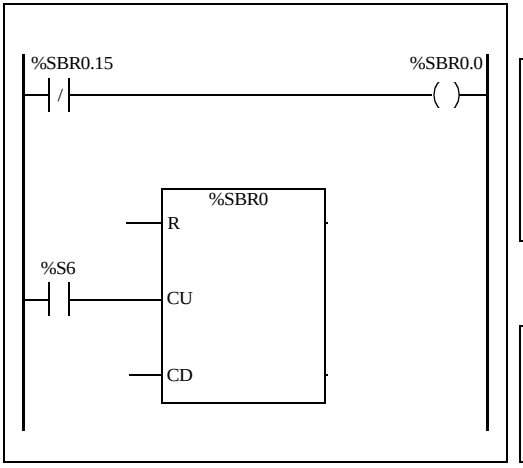
Die folgende Darstellung zeigt das Bitmuster vor und nach einer Schiebeoperation.



Dies gilt auch für die Anforderung mit der Anweisung CD, ein Bit nach rechts zu schieben (Bit 15 bis Bit 0). Bit 0 geht verloren.
Ist ein 16-Bitregister nicht angebracht, können mit Hilfe des Programms auch mehrere Register in Kaskade geschaltet werden.

Nicht reversible

Im folgenden Beispiel wird in jeder Sekunde ein Bit nach links geschoben, während Bit 0 den entgegengesetzten Zustand von Bit 15 einnimmt.



Nicht reversible Programmierung

```
LDN   %SBR0.15
ST    %SBR0.0
BLK   %SBR0
LD    %S6
CU
```

Nicht reversible Programmierung

```
LDN   %SBR0.15
ST    %SBR0.0
LD    %S6
CU    %SBR0
```

Sonderfälle

Die folgende Tabelle enthält eine Liste mit Sonderfällen für die Programmierung des Schiebebitregister-Funktionsblocks.

Sonderfall	Beschreibung
Auswirkung eines Kaltstarts (%S0=1)	Alle Bits des Registerworts werden auf 0 gesetzt.
Auswirkung eines Warmstarts (%S1=1)	Hat keine Auswirkungen auf die Bits im Registerwort.

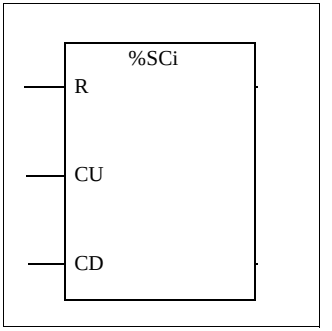
Schrittzähler-Funktionsblock (%SCi)

Einführung

Ein Schrittzähler-Funktionsblock (%SCi) liefert eine Reihe von Schritten, denen Aktionen zugewiesen werden können. Das Weiterschalten von einem Schritt zum nächsten richtet sich nach externen oder internen Ereignissen. Immer wenn ein Schritt aktiv ist, wird das zugehörige Bit auf 1 gesetzt. Es kann immer nur ein Schritt des Zählers zur Zeit aktiv sein.

Abbildung

Im Folgenden sehen Sie ein Beispiel für einen Schrittzähler-Funktionsblock.

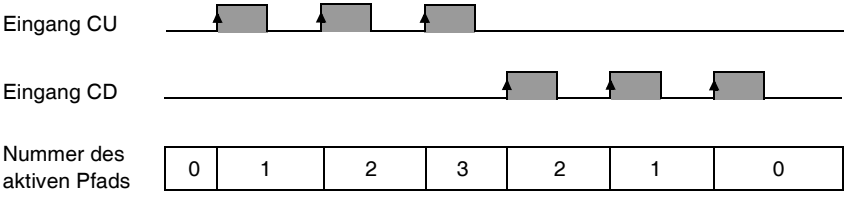


Einstellungen

Der Schrittzähler-Funktionsblock hat die folgenden Parameter:

Parameter	Markierung	Wert
Schrittzählernummer	%SCi	0 - 7
Schrittzählerbit	%SCi.J	Die Schrittzählerbits 0 bis 255 (j = 0 bis 255) können mit einem logischen Ladebefehl getestet und mit einer Zuweisungsanweisung geschrieben werden.
RESET-Eingang (oder Anweisung)	R	Wenn der Funktionsparameter R auf "1" gesetzt ist, wird der Schrittzähler zurückgesetzt.
Inkrementiereingang (bzw. Anweisung)	CU	Bei einer steigenden Flanke wird der Schrittzähler um einen Schritt inkrementiert.
Dekrementiereingang (bzw. Anweisung)	CD	Bei einer steigenden Flanke wird der Schrittzähler um einen Schritt dekrementiert.

Chronogramm Das folgende Diagramm zeigt die Funktionsweise des Schrittzähler-Funktionsblocks.

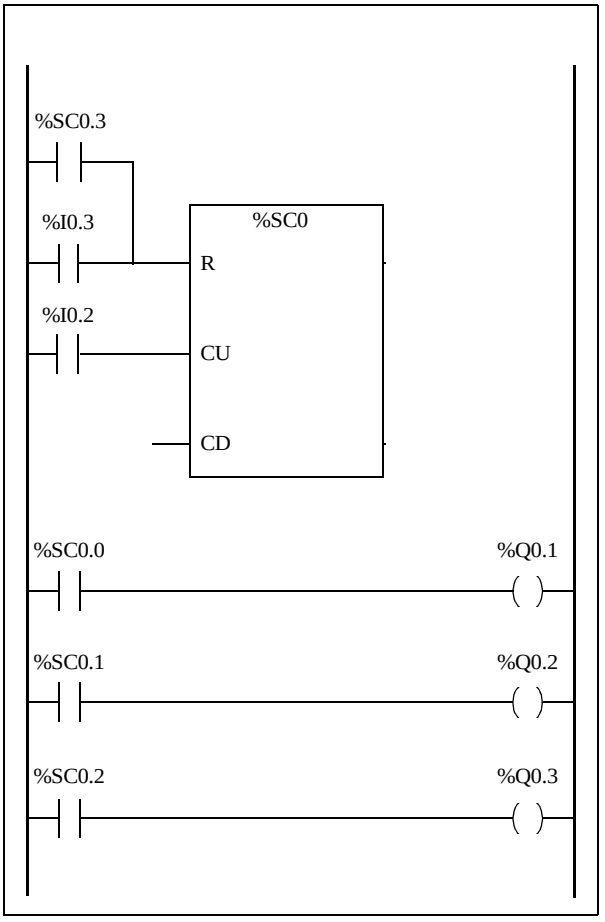


Nicht reversible

Im Folgenden sehen Sie ein Beispiel für einen Schrittzähler-Funktionsblock.

- Schrittzähler 0 wird vom Eingang %I0.2 inkrementiert.
- Schrittzähler 0 wird vom Eingang %I0.3 oder bei Erreichen von Schritt 3 auf 0 zurückgesetzt.
- Schritt 0 steuert Ausgang %Q0.1, Schritt 1 steuert Ausgang %Q0.2 und Schritt 2 steuert Ausgang %Q0.3.

Die folgende Abbildung zeigt die reversible und die nicht reversible Programmierung für dieses Beispiel.



Nicht reversible Programmierung

```
BLK    %SC0
LD     %SC0.3
OR     %I0.3
R
LD     %I0.2
CU
END_BLK
LD     %SC0.0
ST     %Q0.1
LD     %SC0.1
ST     %Q0.2
LD     %SC0.2
ST     %Q0.3
```

Nicht reversible Programmierung

```
LD     %SC0.3
OR     %I0.3
R      %SC0
LD     %I0.2
CU     %SC0
LD     %SC0.0
ST     %Q0.1
LD     %SC0.1
ST     %Q0.2
LD     %SC0.2
ST     %Q0.3
```

Sonderfall

Die folgende Tabelle enthält eine Liste mit Sonderfällen für die Funktionsweise des Schrittzähler-Funktionsblocks.

Sonderfall	Beschreibung
Auswirkung eines Kaltstarts (%S0=1)	Initialisiert den Schrittzähler.
Auswirkung eines Warmstarts (%S1=1)	Hat keine Auswirkungen auf den Schrittzähler.

16.3 Numerische Verarbeitung

Auf einen Blick

Inhalt des Abschnitts

Dieser Abschnitt gibt eine Einführung in die numerische Verarbeitung, die auf Beschreibungen und Programmierrichtlinien basiert.

Inhalt dieses Abschnitts

Dieser Abschnitt enthält die folgenden Themen:

Thema	Seite
Einführung in numerische Anweisungen	427
Zuweisungsanweisungen	428
Vergleichsanweisungen	433
Arithmetische Anweisungen für Ganzzahlen	435
Logische Anweisungen	439
Schiebeanweisungen	440
Konvertierungsanweisungen	442
Anweisungen zur Konvertierung zwischen einfachen Wörtern und Doppelwörtern	444

Einführung in numerische Anweisungen

Auf einen Blick

Numerische Anweisungen gelten im Allgemeinen für 16-Bit-Worte (siehe *Wortobjekte, S. 30*) und für 32-Bit-Doppelworte (Siehe *Gleitkomma- und Doppelwortobjekte, S. 33*). Sie werden in eckigen Klammern geschrieben. War das Ergebnis der vorherigen logischen Verknüpfung wahr (Boolescher Akkumulator = 1), wird die numerische Anweisung ausgeführt. War das Ergebnis der vorherigen logischen Verknüpfung falsch (Boolescher Akkumulator = 0), wird die numerische Anweisung nicht ausgeführt und der Operand wird nicht verändert.

Zuweisungsanweisungen

Einleitung

Mit Zuweisungsanweisungen laden Sie Operand Op2 in Operand Op1.

Zuweisung

Syntax von Zuweisungsanweisungen.

[Op1:=Op2]	<=>	Op2 -> Op1

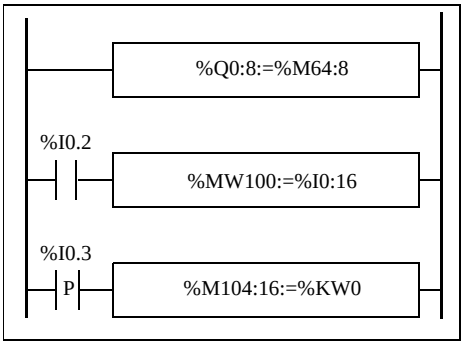
Zuweisungsbefehle können ausgeführt werden für:
- Bitzeichenketten
- Wörter
- Doppelwörter
- Gleitkommawerte
- Worttabellen
- Doppelworttabellen
- Gleitkommazahltabellen

Zuweisung von Bitzeichenketten

Für die folgenden Bitzeichenketten können die Befehle ausgeführt werden (siehe *Strukturierte Objekte, S. 46*):
- Bitzeichenkette -> Bitzeichenkette (Beispiel 1)
- Bitzeichenkette -> Wort (Beispiel 2) oder Doppelwort (indexiert)
- Wort oder Doppelwort -> Bitzeichenkette (Beispiel 3)
- Direkter Wert -> Bitzeichenkette

Beispiele

Beispiele für Zuweisungen von Bitzeichenketten.




```
LD      1
[%Q0:8:=%M64:8]          (Ex. 1)

LD      %I0.2
[%MW100:=%I0:16]         (Ex. 2)

LDR     %I0.3
[%M104:16:=%KW0]         (Ex. 3)
```

Regeln für die Verwendung:

- Für Zuweisungen Bitzeichenkette -> Wort: Die Bits der Zeichenkette werden von rechts zum Wort übertragen (das erste Bit in der Zeichenkette zu Bit 0 im Wort) und die Bits des Wortes, die nicht an der Übertragung (Länge≤16) beteiligt sind, werden auf 0 gesetzt.
- Für Zuweisungen Wort -> Bitzeichenkette: Die Wortbits werden von rechts übertragen (Wortbit 0 zum ersten Bit in der Zeichenkette).

Zuweisungen von Bitzeichenketten

Syntax bei Zuweisungen von Bitzeichenketten.

Operator	Syntax	Operand 1 (Op1)	Operand 2 (Op2)
:=	[Op1: = Op2] Operand 1 (Op1) nimmt den Wert von Operand 2 (Op2) an	%MWi,%QWi, %QWAi,%SWi %MWi[%MWi], %MDi, %MDi[%MWi] **%Mi:L, %Qi:L, %Si:L, %Xi:L**	Direkter Wert, %MWi, %KWi, %IW,%IWAi, %INWi, %QWi, %QWAi %QNWi, %SWi, %BLK.x, %MWi[%MWi], %KWi[%MWi], %MDi[%MWi], %KDi[%MWi], **%Mi:L,%Qi:L, %Si:L, %Xi:L, %Ii:L**

Hinweis: Die Abkürzung %BLK.x (z.B. %C0.P) beschreibt ein Funktionsblockwort.

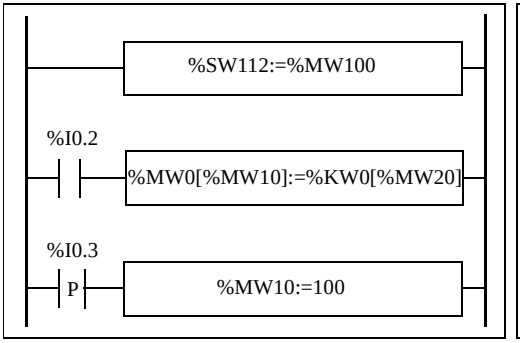
Zuweisung von Wörtern

Zuweisungsbefehle können für die folgenden Wörter und Doppelwörter ausgeführt werden:

- Wort (indexiert) -> Wort (Beispiel 2) (indexiert oder nicht)
- Doppelwort (indexiert) -> Doppelwort (indexiert oder nicht)
- Direkter Ganzzahlwert -> Wort (Beispiel 3) oder Doppelwort (indexiert oder nicht)
- Bitzeichenkette -> Wort oder Doppelwort
- Gleitkommawert (indexiert oder nicht)-> Gleitkommawert (indexiert oder nicht)
- Wort oder Doppelwort -> Bitzeichenkette
- Direkter Gleitkommawert -> Gleitkommawert (indexiert oder nicht)

Beispiele

Beispiele für Wortzuweisungen.




```
LD      1
[%SW112:=%MW100]        (Ex. 1)

LD      %I0.2
[%MW0[%MW10]:=          (Ex. 2)
%KW0[%MW20]]

LDR     %I0.3           (Ex. 3)
[%MW10:=100]
```

Syntax

Syntax von Wortzuweisungen.

Operator	Syntax
:=	[Op1: = Op2] Operand 1 (Op1) nimmt den Wert von Operand 2 (Op2) an

In der folgenden Tabelle sind die Operanden beschrieben.

Typ	Operand 1 (Op1)	Operand 2 (Op2)
Wort, Doppelwort, Bitzeichenkette	**%BLK.x, %MWi, %QWi, %QWAi, %SWi %MWi[MWi**, %MDi, %MDi[%MWj]], %Mi:L, %Qi:L, %Si:L, %Xi:L	**Direkter Wert, %MWi, %KWi, %IW, %IWAi, %QWi, %QWAi, %SWi, %MWi[MWi], %KWi[MWi], %MDi, %MDi[%MWj], %KDi, %KDi[MWj]** , %INW, %Mi:L, %Qi:L, %QNW, %Si:L, %Xi:L, %Ii:L
Gleitkomma	**%MFi, %MFi[%MWj]**	**Direkter Gleitkommawert, %MFi, %MFi[%MWj], %KFi, %KFi[%MWj]**

Hinweis: Die Abkürzung %BLK.x (z.B. R3.I) beschreibt ein Funktionsblockwort. Bei den Bitzeichenketten %Mi:L, %Si:L und %Xi:L muss die Basisadresse der ersten Bitzeichenkette ein Vielfaches von 8 sein (0, 8, 16, ..., 96, ...).

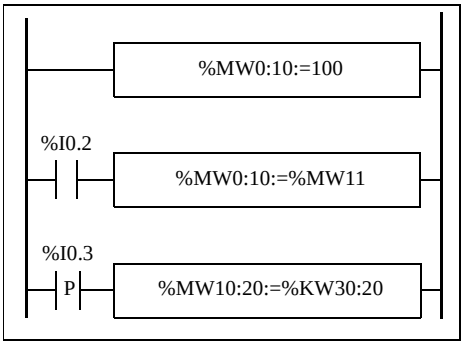
Zuweisung der Wort-, Doppelwort- oder Gleitkommawert-tabellen

Für die folgenden Objekttabellen können Zuweisungsbefehle ausgeführt werden (siehe *Worttabellen, S. 47*):

- Direkter Ganzzahlwert -> Worttabelle (Beispiel 1) oder Doppelworttabelle
- Wort -> Worttabelle (Beispiel 2)
- Worttabelle -> Worttabelle (Beispiel 3)
  Die Tabellenlänge (L) beider Tabellen muss gleich sein.
- Doppelwort -> Doppelworttabelle
- Doppelworttabelle -> Doppelworttabelle
  Die Tabellenlänge (L) beider Tabellen muss gleich sein.
- Direkter Gleitkommawert -> Gleitkommawerttabelle
- Gleitkommawert -> Gleitkommawerttabelle
- Gleitkommawerttabelle -> Gleitkommawerttabelle
  Die Tabellenlänge (L) beider Tabellen muss gleich sein.

Beispiele

Beispiele für Worttabellenzuweisungen:




```
LD      1
[%MW0:10:=100]                (Ex. 1)

LD      %I0.2
[%MW0:10:=%MW11]              (Ex. 2)

LDR     %I0.3
[%MW10:20:=%KW30:20]          (Ex. 3)
```

Syntax

Syntax der Zuweisung der Wort-, Doppelwort- und Gleitkommawerttabellen

Operator	**Syntax**
:=	[Op1: = Op2] Operand 1 (Op1) nimmt den Wert von Operand 2 (Op2) an

In der folgenden Tabelle sind die Operanden beschrieben.

Typ	**Operand 1 (Op1)**	**Operand 2 (Op2)**
Worttabelle	**%MWi:L, %SWi:L**	**%MWi:L, %SWi:L**, Direkter Ganzzahlwert, %MWi, %KWi, %IW, %QW, %IWA, %QWA, %SWi, %BLK.x
Doppelworttabelle	**%MDi:L**	**Direkter Ganzzahlwert, %MDi, %KDi,%MDi:L, %KDi:L**
Gleitkommazahltabelle	**%MFi:L]**	**Direkter Gleitkommawert, %MFi, %KFi, %MFi:L, %KFi:L**

Hinweis: Die Abkürzung %BLK.x (z.B. R3.I) beschreibt ein Funktionsblockwort.

Vergleichsanweisungen

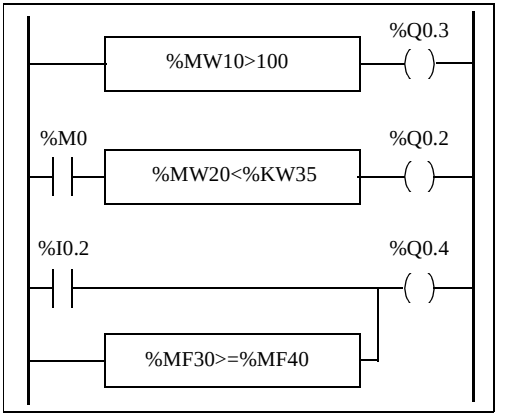
Einleitung

Vergleichsanweisungen dienen zum Vergleichen von zwei Operanden.
Die folgende Tabelle führt die Arten von Vergleichsanweisungen auf.

Anweisung	Funktion
>	Prüfen, ob Operand 1 größer ist als Operand 2
>=	Prüfen, ob Operand 1 größer als oder gleich Operand 2 ist
<	Prüfen, ob Operand 1 kleiner ist als Operand 2
<=	Prüfen, ob Operand 1 kleiner als oder gleich Operand 2 ist
=	Prüfen, ob Operand 1 gleich Operand 2 ist
<>	Prüfen, ob Operand 1 anders als Operand 2 ist

Struktur

Der Vergleich erfolgt zwischen den eckigen Klammern, die den Anweisungen LD, AND und OR folgen. Das Ergebnis ist 1, wenn das geforderte Vergleichsergebnis wahr ist.
Beispiele für Vergleichsanweisungen.




```
LD    [%MW10 > 100]
ST    %Q0.3

LD    %M0
AND   [%MW20 < %KW35]
ST    %Q0.2

LD    %I0.2
OR    [%MF30>=%MF40]
ST    %Q0.4
```

Syntax

Syntax von Vergleichsanweisungen:

Operator	Syntax
>, >=, <, <=, =, <>	LD [Op1 Operator Op2] AND [Op1 Operator Op2] OR [Op1 Operator Op2]

Operanden:

Typ	Operand 1 (Op1)	Operand 2 (Op2)
Wörter	%MWi, %KWi, %INWi, %IW, %IWAi, %QNWi, %QWi, %QWAi, %QNWi, %SWi, %BLK.x	Direkter Wert, %MWi, %KWi, %INWi, %IW, %IWAi, %QNWi, %QW, %QWAi, %SWi, %BLK.x, %MWi [%MWi], %KWi [%MWi]
Doppelwörter	%MDi, %KDi	Direkter Wert, %MDi, %KDi, %MDi [%MWi], %KD [%MWi]
Gleitkommawerte	%MFi, %KFi	Direkter Gleitkommawert, %MFi, %KFi, %MFi [%MWi], %KFi [%MWi]

Hinweis: Vergleichsanweisungen können in Klammern verwendet werden.

Ein Beispiel für die Verwendung einer Vergleichsanweisung in Klammern.

```
LD      %M0
AND(    [%MF20 > 10.0]
OR      %I0.0
)
ST      %Q0.1
```

Arithmetische Anweisungen für Ganzzahlen

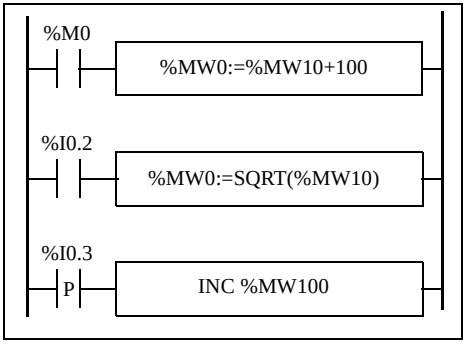
Einführung

Arithmetische Anweisungen führen arithmetische Verknüpfungen zwischen zwei ganzzahligen Operanden oder mit einem ganzzahligen Operanden aus.
Die folgende Tabelle führt die Arten von arithmetischen Anweisungen auf.

Anweisung	Funktion
+	Zwei Operanden addieren
-	Zwei Operanden subtrahieren
*	Zwei Operanden multiplizieren
/	Zwei Operanden dividieren
REM	Divisionsrest der Division von zwei Operanden
SQRT	Quadratwurzel eines Operanden
INC	Einen Operanden inkrementieren
DEC	Einen Operanden dekrementieren
ABS	Absolutwert eines Operanden

Struktur

Arithmetische Anweisungen werden wie folgt ausgeführt:




```
LD      %M0
[%MW0:=%MW10 + 100]

LD      %I0.2
[%MW0:=SQRT(%MW10)]

LDR     %I0.3
[INC %MW100]
```

Syntax

Die Syntax richtet sich nach den verwendeten Operatoren (siehe folgende Tabelle).

Operator	**Syntax**
+,-,*,/,REM	[Op1: = Op 2 Operator Op3]
INC, DEC	[Operator Op1]
SQRT (1)	[Op1: = SQRT(Op2)]
ABS (1)	[Op1: = ABS(Op2)]

Operanden:

Typ	**Operand 1 (Op1)**	**Operanden 2 und 3 (Op2 & 3) (1)**
Wörter	%MWi, %QWi, %QWAi, %SWi	Direkter Wert, %MWi, %KWi, %INW, %IW, %IWAi, %QNW, %QW, %QWAi, %SWi, %BLK.x
Doppelwörter	%MDi	Direkter Wert, %MDi, %KDi

Hinweis: (1) Bei diesem Operator darf Op2 kein direkter Wert sein.
Die Funktion ABS kann nur mit Doppelwörtern (%MD und %KD) und Gleitkommazahlen (%MF und %KF) verwendet werden. Folglich müssen OP1 und OP2 Doppelwörter oder Gleitkommawerte sein.

Überlauf und Fehlerbedingungen

Addition

- Überlauf während der Operation an Wörtern
  Überschreitet das Ergebnis die Kapazität des Ergebniswortes, wird Bit %S18 (Überlauf) auf 1 gesetzt und das Ergebnis ist nicht signifikant (siehe Beispiel 1 auf der nächsten Seite). Das Anwenderprogramms verwaltet das Bit %S18.

Hinweis:
Für die Doppelwörter lauten die Grenzwerte -2147483648 und 21474836487.

Multiplikation

- Überlauf während der Operation
  Überschreitet das Ergebnis die Kapazität des Ergebniswortes, wird Bit %S18 (Überlauf) auf 1 gesetzt und das Ergebnis ist nicht signifikant.

Division/Divisionsrest

- Division durch 0
  Soll durch 0 dividiert werden, ist die Division nicht möglich und Systembit %S18 wird auf 1 gesetzt. Das Ergebnis ist dann falsch.
- Überlauf während der Operation
  Überschreitet der Quotient der Division die Kapazität des Ergebniswortes, wird Bit %S18 auf 1 gesetzt.

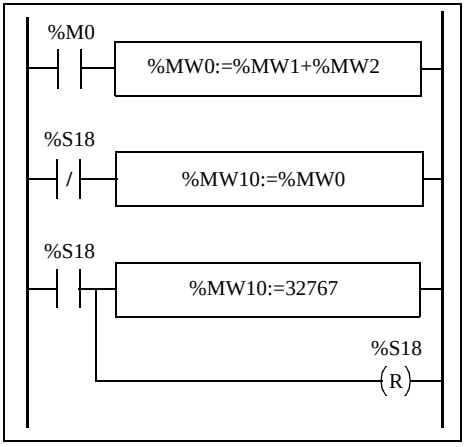
Ziehen der Quadratwurzel

- Überlauf während der Operation
  Die Quadratwurzel kann nur aus positiven Werten gezogen werden. Deshalb ist das Ergebnis immer positiv. Ist der Operand der Quadratwurzel negativ, wird Systembit %S18 auf 1 gesetzt und das Ergebnis ist falsch.

Hinweis: Das Anwenderprogramm ist verantwortlich für die Verwaltung der Systembits %S17 und %S18. Diese werden von der Steuerung auf 1 gesetzt und müssen vom Programm zurückgesetzt werden, damit sie erneut verwendet werden können (siehe Beispiel auf der vorherigen Seite).

Beispiele

Beispiel 1: Überlauf während der Addition




```
LD      %M0
[%MW0:=%MW1 + %MW2]

LDN     %S18
[%MW10:=%MW0]

LD      %S18
[%MW10:=32767]
R       %S18
```

Sind %MW1 = 23241 und %MW2 = 21853, dann kann das tatsächliche Ergebnis (45094) nicht in einem 16-Bit-Wort ausgedrückt werden und Bit %S18 wird auf 1 gesetzt. Das ausgegebene Ergebnis (-20442) ist falsch. In diesem Beispiel wird, wenn das Ergebnis größer als 32767 ist, der Wert auf 32767 festgesetzt.

Logische Anweisungen

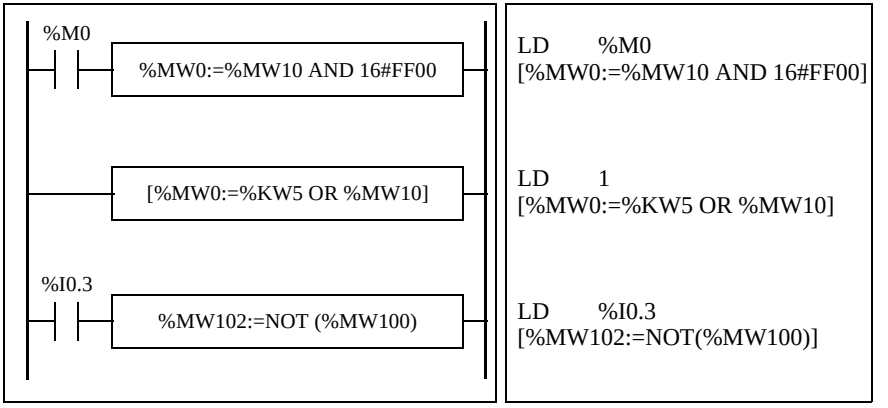
Einleitung

Die logischen Anweisungen führen eine logische Verknüpfung zwischen zwei Wortoperanden oder mit einem Wortoperand aus.
Die folgende Tabelle führt die Arten von logischen Anweisungen auf.

Anweisung	Funktion
AND	UND-Verknüpfung (bitweise) zwischen zwei Operanden
OR	Logische ODER-Verknüpfung (bitweise) zwischen zwei Operanden
XOR	Exklusiv-ODER-Verknüpfung (bitweise) zwischen zwei Operanden
NOT	Logisches Komplement (bitweise) eines Operanden

Struktur

Logische Verknüpfungen werden wie folgt ausgeführt:



Syntax

Die Syntax richtet sich nach den verwendeten Operatoren:

Operator	Syntax	Operand 1 (Op1)	Operanden 2 und 3 (Op2 und 3)
AND, OR, XOR	[Op1: = Op2 Operator Op3]	%MWi, %QWi, %QWAi, %SWi	Direkter Wert (1), %MWi, %KWi, %IW, %IWAi, %QW, %QWAi, %SWi, %BLK.x
NOT	[Op1:=NOT(Op2)]		

Hinweis: (1) Bei NOT darf Op2 kein direkter Wert sein.

Im Beispiel zur Bezugnahme

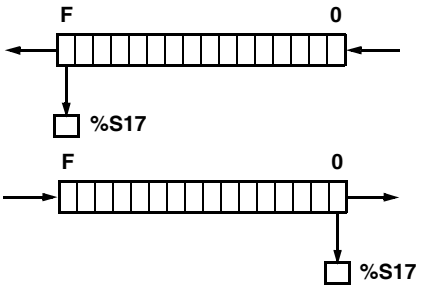
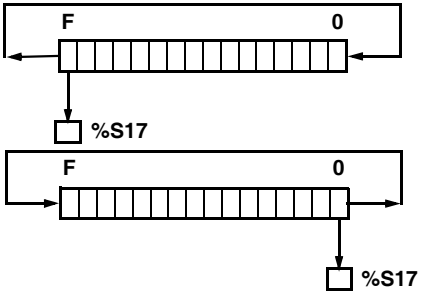
Im Folgenden sehen Sie ein Beispiel für die UND-Anweisung.

```
[%MW15:=%MW32 AND %MW12]
```

Schiebeanweisungen

Einleitung

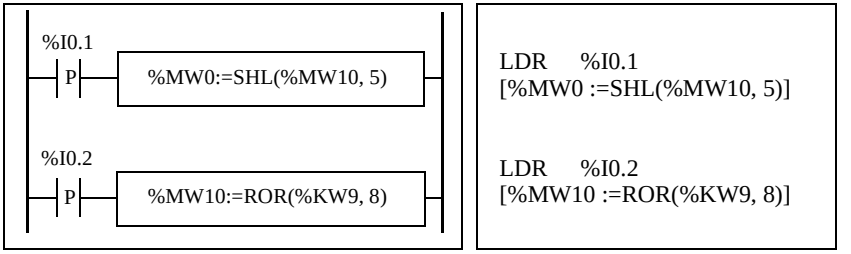
Schiebeanweisungen verschieben die Bits eines Operanden um eine bestimmte Anzahl Stellen nach rechts oder nach links.
Die folgende Tabelle führt die Arten von Schiebeanweisungen auf.

Anweisung	Funktion	
Logisches Schieben		
SHL(op2,i)	Logisches Schieben von i Stellen nach links.	
SHR(op2,i)	Logisches Schieben von i Stellen nach rechts.	
Zirkularverschiebung		
ROR(op2,i)	Zirkularverschiebung von i Stellen nach links.	
ROR(op2,i)	Zirkularverschiebung von i Stellen nach rechts.	

Hinweis: Das Systembit %S17 (Siehe *Systembits (%S), S. 620*) wird für die Leistungsüberschreitung verwendet.

Struktur

Schiebeanweisungen werden wie folgt ausgeführt:



Syntax

Die Syntax richtet sich nach den verwendeten Operatoren (siehe folgende Tabelle).

Operator	**Syntax**
SHL, SHR	[Op1: = Operator (Op2,i)]
ROL, ROR	

Operanden:

Typen	**Operand 1 (Op1)**	**Operand 2 (Op2)**
Wörter	%MWi, %QWi, %QWAi, %SWi	%MWi, %KWi, %IW, %IWAi, %QW, %QWAi, %SWi, %BLK.x
Doppelwörter	%MDi	%MDi, %KDi

Konvertierungsanweisungen

Einleitung

Konvertierungsanweisungen wandeln verschiedene Darstellungen von Zahlen in andere Darstellungen um.
Die folgende Tabelle führt die Arten von Konvertierungsanweisungen auf.

Anweisung	Funktion
BTI	BCD --> Binär-Konvertierung
ITB	Binär --> BCD-Konvertierung

Übersicht über den BCD-Code

Ein BCD-Wert (binärcodierte Dezimalzahl) stellt eine Dezimalziffer (0 bis 9) durch die Codierung von vier Bits dar. Ein 16-Bit-Wortobjekt kann also eine Zahl enthalten, die durch vier Ziffern (0000 - 9999) ausgedrückt wird, und ein 32-Bit-Doppelwortobjekt kann folglich eine Zahl enthalten, die durch acht Ziffern ausgedrückt wird.
Bei einer Konversion wird das Systembit %S18 auf 1 gesetzt, wenn es sich nicht um einen BCD-Wert handelt. Dieses Bit muss vom Programm getestet und auf 0 zurückgesetzt werden.
BCD-Darstellung von Dezimalzahlen:

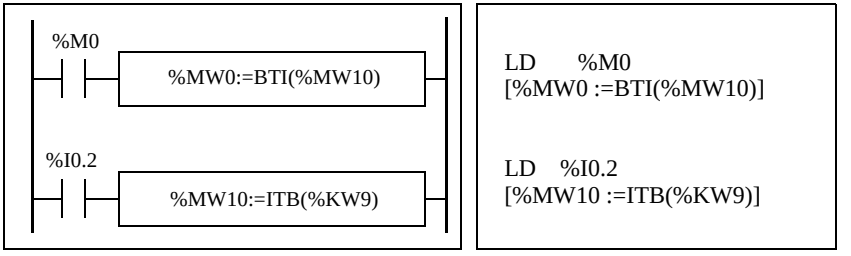
Dezimal	0	1	2	3	4	5	6	7	8	9
BCD	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001

Beispiele:
- Das Wort %MW5 drückt den BCD-Wert "2450" aus, der dem folgenden Binärwert entspricht: 0010 0100 0101 0000
- Das Wort %MW12 drückt den Dezimalwert "2450" aus, der dem folgenden Binärwert entspricht: 0000 1001 1001 0010

Das Wort %MW5 wird mit der Anweisung BTI in das Wort %MW12 konvertiert.
Das Wort %MW12 wird mit der Anweisung ITB in das Wort %MW5 konvertiert.

Struktur

Konversionsanweisungen werden wie folgt ausgeführt:



Syntax

Die Syntax richtet sich nach den verwendeten Operatoren (siehe folgende Tabelle).

Operator	Syntax
BTI, ITB	[Op1: = Operator (Op2)]

Operanden:

Typ	Operand 1 (Op1)	Operand 2 (Op2)
Wörter	%MWi, %QWi, %QWAi, %SWi	%MWi, %KWi, %IW, %IWAi, %QW, %QWAi, %SWi, %BLK.x
Doppelwörter	%MDi	%MDi, %KDi

Applikationsbeispiele:

Die Anweisung BTI verarbeitet einen Sollwert an den Steuerungseingängen über BCD-codierte Rändelschalter.
Die Anweisung ITB zeigt auf BCD-codierten Displays numerische Werte an (z.B. das Ergebnis einer Berechnung, den aktuellen Wert eines Funktionsblocks).

Anweisungen zur Konvertierung zwischen einfachen Wörtern und Doppelwörtern

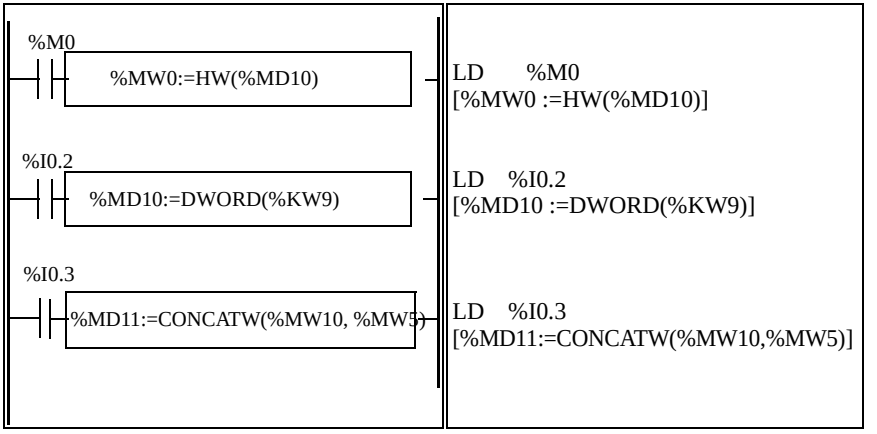
Einleitung

Die folgende Tabelle beschreibt die Anweisungen zur Konvertierung zwischen einfachen Wörtern und Doppelwörtern.

Anweisung	Funktion
LW	Extrahiert das niederwertige Byte eines Doppelwortes in ein Wort.
HW	Extrahiert das höherwertige Byte eines Doppelwortes in ein Wort.
CONCATW	Verkettet zwei Wörter, um so ein Doppelwort zu bilden.
DWORD	Konvertiert ein 16-Bit-Wort in ein 32-Bit-Doppelwort.

Struktur

Konversionsanweisungen werden wie folgt ausgeführt:



Syntax

Die Syntax richtet sich nach den verwendeten Operatoren (siehe folgende Tabelle).

Operator	Syntax	Operand 1 (Op1)	Operand 2 (Op2)	Operand 3 (Op3)
LW, HW	Op1 = Operator (Op2)	%MWi	%MDi, %KDi	[-]
CONCATW	Op1 = Operator (Op2, Op3)	%MDi	%MWi, %KWi, direkter Wert	%MWi, %KWi, direkter Wert
DWORD	Op1 = Operator (Op2)	%MDi	%MWi, %KWi	[-]

16.4 Programmanweisungen

Auf einen Blick

Inhalt des Abschnitts

Dieser Abschnitt enthält eine Einführung in Programmanweisungen.

Inhalt dieses Abschnitts

Dieser Abschnitt enthält die folgenden Themen:

Thema	Seite
END-Anweisungen	446
Anweisung NOP	448
Sprunganweisungen	449
Unterprogramm-Anweisungen	450

END-Anweisungen

Einführung

Die END-Anweisungen definieren das Ende der Ausführung eines Programmzyklus.

END, ENDC und ENDCN

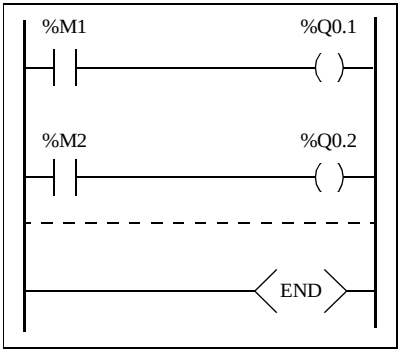
Es stehen drei verschiedene Anweisungen zur Verfügung:

- END: unbedingtes Ende des Programms
- ENDC: Ende des Programms, wenn das boolesche Ergebnis der vorherigen Testanweisung 1 ist
- ENDCN: Ende des Programms, wenn das boolesche Ergebnis der vorherigen Testanweisung 0 ist

Standardmäßig (im Normalbetrieb) werden die Ausgänge am Programmende aktualisiert und der nächste Zyklus wird gestartet.
Bei periodischer Abtastung werden die Ausgänge am Ende der Periode aktualisiert und der nächste Zyklus wird gestartet.

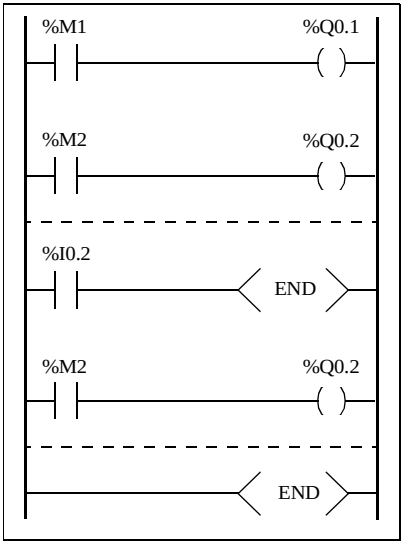
Beispiele

Beispiel für die unbedingte END-Anweisung.




```
LD      %M1
ST      %Q0.1
LD      %M2
ST      %Q0.2

.....

END
```

Beispiel für die bedingte END-Anweisung.




```
LD      %M1
ST      %Q0.1
LD      %M2
ST      %Q0.2

.....

LD      %I0.2   → Wenn %I0.2 = 1,
ENDC              Ende der
LD      %M2       Programmabtastung
ST      %Q0.2
                  Wenn %I0.2 = 0,
                  Fortsetzung
                  der
.....             Programmabtastung,
                  bis neue
                  END-Anweisung
```

Anweisung NOP

NOP	Die Anweisung NOP führt keinen Befehl aus. Mit dieser Anweisung können Sie Zeilen im Programm "reservieren", so dass Sie später Anweisungen einfügen können, ohne die Zeilennummern zu ändern.

Sprunganweisungen

Einleitung

Durch Sprunganweisungen wird die Ausführung des Programms sofort unterbrochen und in der Zeile nach der Programmzeile wieder aufgenommen, die die Markierung %Li (i = 1 bis 16 für eine kompakte Steuerung und 1 bis 63 für die anderen Steuerungen) enthält.

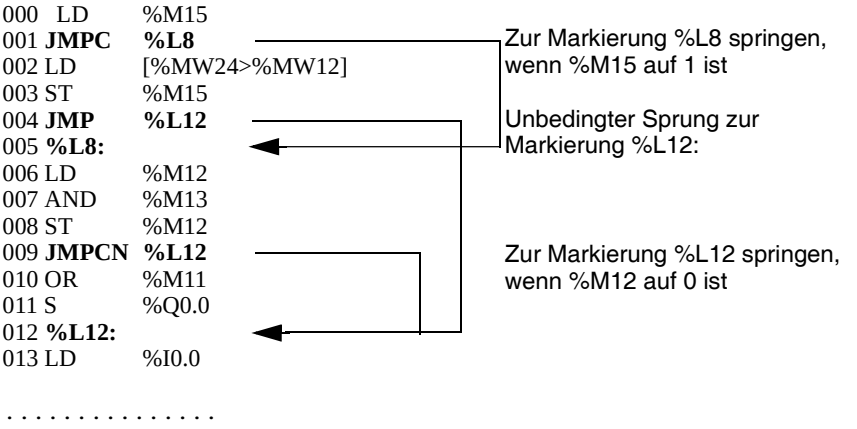
JMP, JMPC und JMPCN

Es stehen drei verschiedene Sprunganweisungen zur Verfügung:

- JMP: unbedingter Programmsprung
- JMPC: Programmsprung, wenn das boolesche Ergebnis der vorherigen Logik 1 ist
- JMPCN: Programmsprung, wenn das boolesche Ergebnis der vorherigen Logik 0 ist

Beispiele

Beispiele für Sprunganweisungen.



Richtlinien

- Sprunganweisungen sind zwischen Klammern nicht zugelassen und sie dürfen nicht zwischen den Anweisungen AND(, OR(und der Anweisung mit der geschlossenen Klammer ")" stehen.
- Die Markierung darf nur vor einer der Anweisungen LD, LDN, LDR, LDF oder BLK angeordnet werden.
- Die Markierungsnummer der Markierung %Li darf nur einmal im Programm definiert sein.
- Der Programmsprung führt zu einer Programmzeile, die sich weiter aufwärts oder weiter abwärts befindet. Geht der Sprung aufwärts, müssen Sie die Zykluszeit des Programms beobachten. Eine zu lange Zykluszeit kann zur Auslösung des Watchdogs führen.

Unterprogramm-Anweisungen

Einleitung

Die Unterprogramm-Anweisungen bewirken, dass ein Programm ein Unterprogramm ausführt und dann zum Hauptprogramm zurückkehrt.

SRn, SRn: und RET

Unterprogramme bestehen aus drei Schritten:

- Die Anweisung **SRn** ruft das Unterprogramm auf, das von der Marke SRn angegeben wird, wenn das Ergebnis der vorherigen Booleschen Anweisung 1 ist.
- Das Unterprogramm wird durch die Bezeichnung **SRn:** referenziert, wobei n für TWDLCAA10DRF und TWDLCAA16DRF einen Wert zwischen 0 und 15 und für alle anderen Steuerungen einen Wert zwischen 0 und 63 annehmen kann.
- Die Anweisung **RET** am Ende des Unterprogramms bewirkt, dass der Programmfluss zum Hauptprogramm zurückkehrt.

Im Beispiel zur Bezugnahme

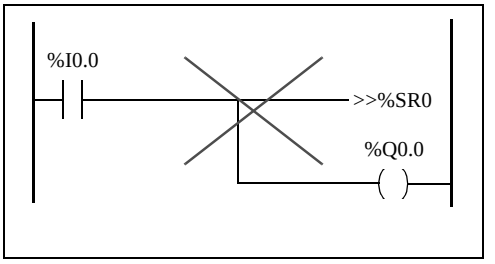
Beispiele für Unterprogramm-Anweisungen.

```
000 LD      %M15
001  AND    %M5
002  ST     %Q0.0
003  LD     [%MW24>%MW12]
004  SR8                          Sprung zum Unterprogramm SR8
005  LD     %I0.4
006  AND    M13
007  _
008  _
009  _
010  END

011  SR8:
012  LD     1
013  IN     %TM0
014  LD     %TM0.Q
015  ST     %M15
010  RET                          Rückkehr zum Hauptprogramm

. . . . . .
```

Richtlinien

- Ein Unterprogramm darf kein weiteres Unterprogramm aufrufen.
- Unterprogramm-Anweisungen sind nicht zwischen Klammern zugelassen und sie dürfen nicht zwischen den Anweisungen AND(, OR(und der Anweisung mit der geschlossenen Klammer ")" stehen.
- Die Marke darf nur vor einer Anweisung LD oder BLK angeordnet werden und kennzeichnet den Beginn einer Booleschen Gleichung (bzw. des Programmblocks).
- Auf den Aufruf eines Unterprogramms darf keine Zuweisungsanweisung folgen. Dies ist der Fall, weil das Unterprogramm den Inhalt des Booleschen Akkumulators möglicherweise ändert. Es könnte dann bei Rückkehr einen anderen Wert haben als vor dem Aufruf. Siehe hierzu folgendes Beispiel.

Beispiel für die Programmierung eines Unterprogramms.




```
LD      %I0.0
SR0
ST      %Q0.0

LD      %I0.0
ST      %Q0.0
SR0
```

Erweiterte Anweisungen

17

Auf einen Blick

Inhalt dieses Kapitels

In diesem Kapitel erfahren Sie Einzelheiten zu Anweisungen und Funktionsblöcken, mit denen Sie erweiterte Steuerungsprogramme für speicherprogrammierbare Twido Steuerungen erstellen können.

Inhalt dieses Kapitels

Dieses Kapitel enthält die folgenden Abschnitte:

Abschnitt	Thema	Seite
17.1	Erweiterte Funktionsblöcke	455
17.2	Uhrfunktionen	499
17.3	Twido PID-Kurzanleitung	509
17.4	PID-Funktion	534
17.5	Anweisungen zu Gleitkommas	588
17.6	Anweisungen für Objekttabellen:	600

17.1 Erweiterte Funktionsblöcke

Auf einen Blick

Inhalt des Abschnitts

Dieser Abschnitt gibt eine Übersicht über die erweiterten Funktionsblöcke und enthält Programmierbeispiele.

Inhalt dieses Abschnitts

Dieser Abschnitt enthält die folgenden Themen:

Thema	Seite
Zu erweiterten Funktionsblöcken gehörige Bit- und Wortobjekte	456
Prinzipien für die Programmierung von erweiterten Funktionsblöcken	458
LIFO/FIFO-Register-Funktionsblock (%Ri)	461
LIFO-Funktionsweise	462
FIFO-Funktionsweise	463
Programmierung und Konfiguration der Register	464
Impulsbreitenmodulations-Funktionsblock (%PWM)	466
Impulsgenerator-Funktionsblock (%PLS)	469
Trommelsteuerungs-Funktionsblock (%DR)	472
Funktionsweise des Trommelsteuerungs-Funktionsblocks %DRi	474
Programmieren und Konfigurieren von Trommelsteuerungen	476
Schneller Zähler-Funktionsblock (%FC)	478
Sehr schneller Zähler-Funktionsblock (%VFC)	481
Senden/Empfangen von Meldungen - die Austauschanweisung (EXCH)	494
Austauschsteuerungs-Funktionsblock (%MSGx)	495

Zu erweiterten Funktionsblöcken gehörige Bit- und Wortobjekte

Einführung

Die erweiterten Funktionsblöcke verwenden zweckbestimmte Wörter und Bits desselben Typs wie die Standard-Funktionsblöcke. Erweiterte Funktionsblöcke umfassen:

- LIFO/FIFO-Register (%R)
- Trommelsteuerungen (%DR)
- Schnelle Zähler (%FC)
- Sehr schnelle Zähler (%VFC)
- Impulsbreitenmodulation (%PWM)
- Impulsgeneratorausgang (%PLS)
- das Schiebebitregister (%SBR)
- Schrittzähler (%SC)
- Meldungssteuerungsblock (%MSG)

Vom Programm zugängliche Objekte

Die folgende Tabelle bietet einen Überblick über die Wörter und Bits, auf die das Programm zugreifen kann und die den verschiedenen erweiterten Funktionsblöcken zugewiesen sind. Beachten Sie, dass sich der Schreibzugriff in der folgenden Tabelle nach den Einstellungen "Verstellbar" während der Konfiguration richtet. Diese Einstellung ermöglicht und verhindert den Zugriff auf Wörter und Bits mit TwidoSoft oder der Bedienoberfläche.

Erweiterter Funktionsblock	Zugehörige Wörter und Bits		Adresse	Schreibzugriff
%R	Wort	Eingang des Registers	%Ri.I	Ja
	Wort	Registerausgang	%Ri.O	Ja
	Bit	Registerausgang voll	%Ri.F	Nein
	Bit	Registerausgang leer	%Ri.E	Nein
%DR	Wort	Aktuelle Schrittnummer	%DRi.S	Ja
	Bit	Letzter Schritt gleich aktueller Schritt	%DRi.F	Nein
%FC	Wort	Aktueller Wert	%FCi.V	Ja
	Wort	Voreingestellter Wert	%FCi.P	Ja
	Bit	Fertig	%FCi.D	Nein

Erweiterter Funktionsblock	Zugehörige Wörter und Bits		Adresse	Schreibzugriff
%VFC	Wort	Aktueller Wert	%VFCi.V	Nein
	Wort	Voreingestellter Wert	%VFCi.U	Ja
	Bit	Zählrichtung	%VFCi.U	Nein
	Wort	Wert erfassen	%VFCi.C	Nein
	Wort	Grenzwert 0	%VFCi.S0	Ja
	Wort	Schwellenwert 1	%VFCi.S1	Ja
	Bit	Überlauf-	%VFCi.F	Nein
	Bit	Reflexausgang 0 freigeben	%VFCi.R	Ja
	Bit	Reflexausgang 1 freigeben	%VFCi.S	Ja
	Bit	Grenzwertausgang 0	%VFCi.TH0	Nein
	Bit	Grenzwertausgang 1	%VFCi.TH1	Nein
	Bit	Zeitbasis Frequenzmessung	%VFCi.T	Ja
%PWM	Wort	Prozent der Impulse auf 1 in Beziehung zur Gesamtdauer.	%PWMi.R	Ja
	Wort	Voreingestellte Periode	%PWMi.P	Ja
%PLS	Wort	Anzahl Impulse	%PLSi.N	Ja
	Wort	Voreingestellter Wert	%PLSi.P	Ja
	Bit	Aktueller Ausgang freigegeben	%PLSi.Q	Nein
	Bit	Erzeugung erledigt	%PLSi.D	Nein
%SBR	Bit	Registerbit	%SBRi.J	Nein
%SC	Bit	Schrittzählerbit	%SCi.J	Ja
%MSG	Bit	Fertig	%MSGi.D	Nein
	Bit	Fehler	%MSGi.E	Nein

Prinzipien für die Programmierung von erweiterten Funktionsblöcken

Auf einen Blick

Alle Twido Anwendungen werden in Form von AWL-Programmen gespeichert, auch wenn Sie im KOP-Editor geschrieben wurden. Deshalb können Twido Steuerungen als AWL-"Rechner" bezeichnet werden. Der Begriff "Umkehrbarkeit" bezieht sich auf die Fähigkeit von TwidoSoft, eine AWL-Anwendung in KOP darzustellen und auch wieder in AWL zurückzuschalten. Standardmäßig sind alle KOP-Programme reversibel.

Wie bei Basis-Funktionsblöcken müssen auch bei erweiterten Funktionsblöcken die Regeln für die Umkehrbarkeit berücksichtigt werden. Für den Aufbau von reversiblen Funktionsblöcken in AWL werden die folgenden Anweisungen benötigt:

- **BLK** : Kennzeichnet den Blockanfang und den Eingangsteil des Funktionsblocks.
- **OUT_BLK**: Kennzeichnet den Beginn des Ausgangsteils des Funktionsblocks.
- **END_BLK** : Kennzeichnet das Ende des Funktionsblocks.

Hinweis: Die Verwendung dieser reversiblen Funktionsblockanweisungen ist für ein ordnungsgemäß funktionierendes AWL-Programm nicht obligatorisch. Einige Anweisungen können in AWL programmiert werden, die nicht reversibel sind.

Zweckbestimmte Eingänge und Ausgänge

Die erweiterten Funktionen Schneller Zähler, Sehr schneller Zähler, PLS und PWM nutzen zweckbestimmte Eingänge und Ausgänge, doch diese Bits sind nicht für die exklusive Verwendung durch einen einzigen Block reserviert. Die Verwendung dieser zweckbestimmten Ein- und Ausgänge muss stattdessen verwaltet werden. Wenn Sie diese erweiterten Funktionen nutzen, müssen Sie verwalten, wie die zweckbestimmten Eingänge und Ausgänge zugeordnet werden. TwidoSoft unterstützt die Konfiguration dieser Ressourcen durch die Anzeige von Einzelheiten für die Eingangs-/Ausgangskonfiguration und warnt, wenn ein zweckbestimmter Eingang oder Ausgang bereits von einem konfigurierten Funktionsblock verwendet wird.
Die folgende Tabelle bietet einen Überblick über die zweckbestimmten Eingänge und Ausgänge und bestimmte Funktionen.
Bei Verwendung in Zählfunktionen:

Eingänge	Verwendung
%I0.0.0	%VFC0 : Auf-/Ab-Einstellung oder Phase B
%I0.0.1	%VFC0 : Impulseingang oder Phase A
%I0.0.2	%FC0: Impulseingang oder voreingestellter Eingang von %VFC0
%I0.0.3	%FC1: Impulseingang oder Selbsthalteeingang von %VFC0
%I0,0,4	%FC2: Impulseingang oder Selbsthalteeingang von %VFC1
%I0.0.5	Voreingestellter Eingang von %VFC1
%I0.0.6	%VFC1: Auf-/Ab-Einstellung oder Phase B
%I0.0.7	%VFC1: Impulseingang oder Phase A

Bei Verwendung in Zähl- oder Sonderfunktionen:

Ausgänge	Verwendung
%Q0.0.0	Ausgang %PLS0 oder PWM0
%Q0.0.1	Ausgang %PLS1 oder PWM1
%Q0,0,2	Reflexausgänge für %VFC0
%Q0.0.3	
%Q0,0,4	Reflexausgänge für %VFC1
%Q0.0.5	

Verwendung von zweckbestimmten Eingängen und Ausgängen

TwidoSoft achtet auf die Einhaltung der folgenden Regeln für die Verwendung von zweckbestimmten Eingängen und Ausgängen.

- Jeder Funktionsblock, der zweckbestimmte E/A verwendet, muss konfiguriert und dann in der Anwendung referenziert werden. Zweckbestimmte E/A werden nur zugeordnet, wenn ein Funktionsblock konfiguriert wird und nicht, wenn er im Programm referenziert wird.
- Nachdem Sie einen Funktionsblock konfiguriert haben, können dessen zweckbestimmte Eingänge und Ausgänge nicht von der Anwendung oder von einem anderen Funktionsblock verwendet werden.
  Wenn Sie z.B. %PLS0 konfigurieren, können Sie %Q0.0.0 nicht in %DR0 (Trommelsteuerung) oder in der Anwendungslogik (d.h. ST %Q0.0.0) verwenden.
- Wenn ein zweckbestimmter Eingang oder Ausgang von einem Funktionsblock benötigt wird und dieser Eingang oder Ausgang bereits von der Anwendung oder von einem anderen Funktionsblock verwendet wird, kann der Funktionsblock nicht konfiguriert werden.
  Wenn Sie z.B. %FC0 als Aufwärtszähler konfigurieren, können Sie %VFC0 nicht so konfigurieren, dass %I0.0.2 als Selbsthalteeingang verwendet wird.

Hinweis: Wenn Sie die Verwendung zweckbestimmter E/A ändern möchten, müssen Sie den Funktionsblock entkonfigurieren, indem Sie den Objekttyp auf "nicht verwendet" setzen und dann die Referenzen auf den Funktionsblock in Ihrer Anwendung löschen.

LIFO/FIFO-Register-Funktionsblock (%Ri)

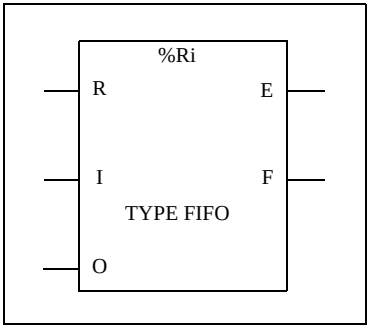
Einleitung

Ein Register ist ein Speicherblock, der bis zu 16 Worte mit je 16 Bits auf jeweils zwei verschiedene Arten speichern kann:

- Warteschlange (First In, First Out), genannt FIFO;
- Stapel (Last In, First Out), genannt LIFO.

Abbildung

Im Folgenden sehen Sie eine Abbildung des Register-Funktionsblocks.



Registerfunktionsblock

Einstellungen

Der Register-Funktionsblock hat die folgenden Parameter:

Parameter	Beschriftung	Wert
Registernummer	%Ri	0 bis 3
Typ	FIFO oder LIFO	Warteschlange oder Stapel)
Eingangswort	%Ri.I	Register-Eingangswort. Kann gelesen, getestet und geschrieben werden.
Ausgangswort	%Ri.O	Register-Ausgangswort. Kann gelesen, getestet und geschrieben werden.
Speichereingang (oder Anweisung)	I (Eingang)	Speichert bei einer steigenden Flanke den Inhalt von Wort %Ri.I im Register.
Abrufeingang (oder Anweisung)	O (Ausgang)	Lädt bei einer steigenden Flanke ein Datenwort des Registers in das Wort %Ri.O.
RESET-Eingang (oder Anweisung)	R (Rücksetzen)	Initialisiert im Zustand 1 das Register.
Ausgang Leer	E (Leer)	Das zugehörige Bit %Ri.E zeigt an, dass das Register leer ist. Kann getestet werden.
Ausgang Voll	F (Voll)	Das zugehörige Bit %Ri.F zeigt an, dass das Register voll ist. Kann getestet werden.

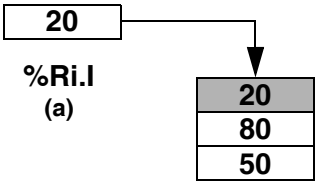
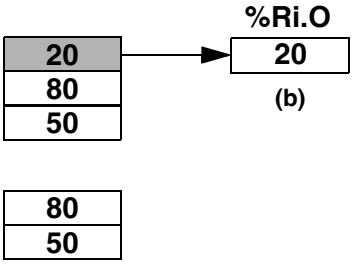
LIFO-Funktionsweise

Einleitung

Bei der LIFO-Funktionsweise (Last In, First Out) wird das Datenelement, das zuletzt eingetragen wurde, zuerst abgerufen.

Funktionsweise

Die folgende Tabelle beschreibt die LIFO-Funktionsweise.

Schritt	Beschreibung	Im Beispiel zur Bezugnahme
1	Wenn eine Speicheranforderung empfangen wird (steigende Flanke an Eingang I oder Aktivierung der Anweisung I), wird der Inhalt von Eingangswort %Ri.I (das bereits geladen ist) in der Spitze des Stapels gespeichert (Abb. a). Ist der Stapel voll (Ausgang F=1), kann nichts mehr gespeichert werden.	Speichern des Inhalts von %Ri.I im obersten Wert des Stapels 
2	Wird eine Abrufanforderung empfangen (steigende Flanke am Eingang O oder Aktivierung der Anweisung O), wird das höchste Datenwort (das zuletzt eingegebene Wort) in das Wort %Ri.O geladen (Abb. b). Ist das Register leer (Ausgang E=1), ist kein Abruf mehr möglich. Ausgangswort %Ri.O wird nicht verändert und behält seinen letzten Wert.	Abrufen des höchsten Datenworts im Stapel 
3	Der Stapel kann jederzeit zurückgesetzt werden (Zustand 1 am Eingang R oder Aktivierung der Anweisung R). Das vom Zeiger angegebene Element ist dann das höchste Element im Stapel.	

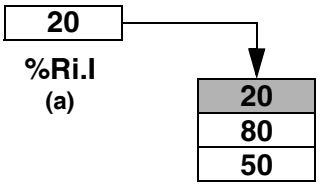
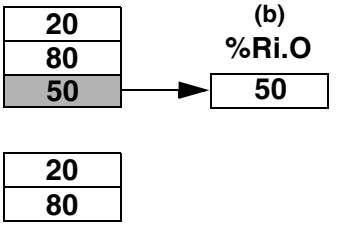
FIFO-Funktionsweise

Einleitung

Bei der FIFO-Funktionsweise (First In, First Out) wird das Datenelement, das zuerst eingetragen wurde, auch zuerst wieder abgerufen.

Funktionsweise

Die folgende Tabelle beschreibt die FIFO-Funktionsweise.

Schritt	Beschreibung	Im Beispiel zur Bezugnahme
1	Wenn eine Speicheranforderung empfangen wird (steigende Flanke an Eingang I oder Aktivierung der Anweisung I), wird der Inhalt von Eingangswort %Ri.I (das bereits geladen ist) in der Spitze der Warteschlange gespeichert (Abb. a). Ist die Warteschlange voll (Ausgang F=1), ist kein Speichern mehr möglich.	Speichern des Inhalts von %Ri.I im obersten Wert der Warteschlange 
2	Wird eine Abrufanforderung empfangen (steigende Flanke am Eingang O oder Aktivierung der Anweisung O), wird das Datenwort in der letzten Position in der Warteschlange in das Ausgangswort %Ri.O geladen. Der Inhalt des Registers wird in der Warteschlange um einen Platz nach unten verschoben (Abb. b). Ist das Register leer (Ausgang E=1), ist kein Abruf mehr möglich. Ausgangswort %Ri.O wird nicht verändert und behält seinen letzten Wert.	Abrufen des ersten Datenelements, das dann in %Ri.O geladen wird 
3	Die Warteschlange kann jederzeit zurückgesetzt werden (Zustand 1 am Eingang R oder Aktivierung der Anweisung R).	

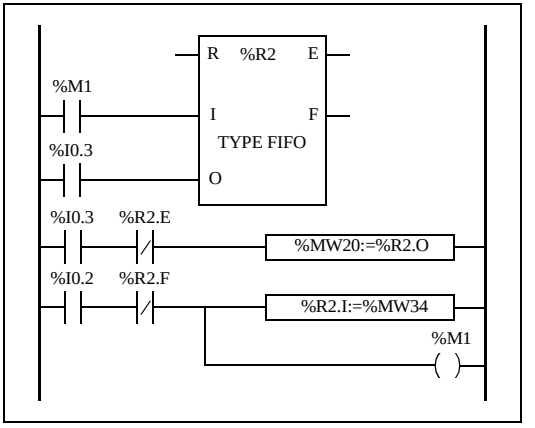
Programmierung und Konfiguration der Register

Einleitung

Im folgenden Programmierbeispiel wird der Inhalt eines Speicherworts (%MW34) durch die Speicheranforderung (%I0.2) in ein Register (%R2.I) geladen, wenn das Register %R2 nicht voll (%R2.F = 0) ist. Die Speicheranforderung im Register wird von %M1 gestellt. Die Abrufanforderung wird von Eingang %I0.3 gestellt und %R2.O wird in %MW20 geladen, wenn das Register nicht voll (%R2.E = 0) ist.

Programmier-beispiel

Das folgende Bild zeigt einen Register-Funktionsblock mit Beispielen für reversible und nicht reversible Programmierung.



Kontaktplan

```
BLK          %R2
LD           %M1
I
LD           %I0.3
O
END_BLK
LD           %I0.3
ANDN         %R2.E
[%MW20:=%R2.O]
LD           %I0.2
ANDN         %R2.F
```

Reversibles Programm

```
LD           %M1
I            %R2
LD           %I0.3
O            %R2
ANDN         %R2.E
[%MW20:=%R2.O]
LD           %I0.2
ANDN         %R2.F
[%R2.I:=%MW34]
ST           %M1
```

Nicht reversibles Programm

Konfiguration

Der einzige Parameter, der während der Konfiguration eingegeben werden muss, ist der Registertyp:

- FIFO (Voreinstellung) oder
- LIFO

Sonderfälle

Die folgende Tabelle enthält eine Liste mit Sonderfällen für die Funktionsweise:

Sonderfall	Beschreibung
Auswirkung eines Kaltstarts (%S0=1)	Initialisiert den Inhalt des Registers. Das Ausgangsbit %Ri.E, das dem Ausgang E zugewiesen ist, wird auf 1 gesetzt.
Auswirkung eines Warmstarts (%S1=1) bei Steuerungsstopp	Hat keine Auswirkungen auf den aktuellen Wert des Registers und auch nicht auf den Zustand der Ausgangsbits.

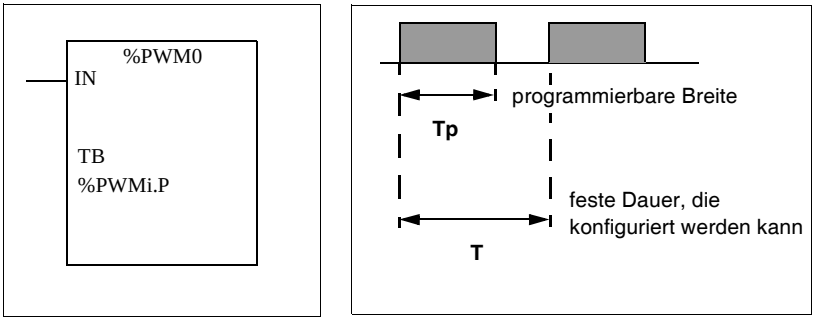
Impulsbreitenmodulations-Funktionsblock (%PWM)

Einleitung

Der Impulsbreitenmodulations-Funktionsblock (%PWM) generiert ein Rechtecksignal an den zweckbestimmten Ausgangskanälen (%Q0.0.0 oder %Q0.0.1), deren Breite und folglich das Tastverhältnis verändert werden. Steuerungen mit Relaisausgängen für diese beiden Kanäle unterstützen diese Funktion wegen einer Frequenzbegrenzung nicht.
Es stehen zwei %PWM-Blöcke zur Verfügung. %PWM0 nutzt den zweckbestimmten Ausgang %Q0.0.0 und %PMW1 nutzt den zweckbestimmten Ausgang %Q0.0.1. Die %PLS-Funktionsblöcke nutzen gemeinsam dieselben zweckbestimmten Ausgänge. Es ist daher erforderlich, eine der beiden Funktionen auszuwählen.

Abbildung

PWM-Block und Chronogramm:



Einstellungen Die folgende Tabelle führt die Parameter des PWM-Funktionsblocks auf.

Parameter	Beschriftung	Beschreibung
Zeitbasis	TB	0,142 ms, 0,57 ms, 10 ms, 1 s (Standardwert)
Voreinstellung der Periode	%PWMi.P	0 < %PWMi.P <= 32767 mit Zeitbasis von 10 ms oder 1 s 0 < %PWMi.P <= 255 mit Zeitbasis von 0,57 ms oder 0,142 s 0 = Funktion nicht verwendet
Tastverhältnis	%PWMi.R	Dieser Wert gibt den prozentualen Anteil des Signals im Zustand 1 in einer Periode an. Die Breite Tp ist deshalb gleich: Tp = T * (%PWMi.R/100). Die Applikation schreibt den Wert für %PWMi.R. Dieses Wort steuert das Tastverhältnis der Periode. Zur Definition von T finden Sie weitere Informationen weiter unten unter "Periodenbereiche". Der voreingestellte Wert ist 0 und Werte größer als 100 werden als Wert 100 betrachtet.
Impulserzeugungseingang	IN	Im Zustand 1 wird das Impulsbreitenmodulationssignal am Ausgangskanal erzeugt. Im Zustand 0 wird der Ausgangskanal auf 0 gesetzt.

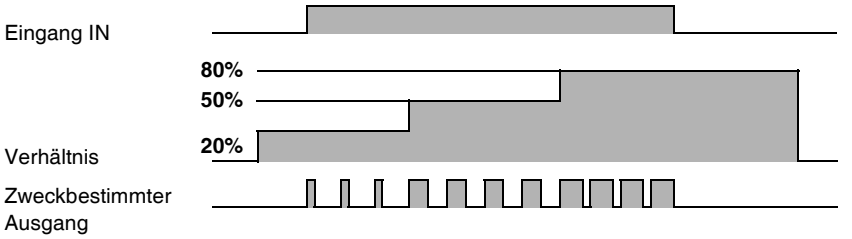
Periodenbereiche

Der voreingestellte Wert und die Zeitbasis können während der Konfiguration geändert werden. Sie dienen zum Festsetzen der Signalperiode T=%PWMi.P * TB. Je niedriger das zu erreichende Verhältnis, desto größer muss der ausgewählte Wert für %PWMi.P sein. Verfügbare Periodenbereiche:

- 0,142 ms bis 36,5 ms in 0,142 ms-Schritten (27,4 Hz bis 7 kHz)
- 0,57 ms bis 146 ms in 0,57 ms-Schritten (6,84 Hz bis 1,75 kHz)
- 10 ms bis 5,45 min in 10 ms-Schritten
- 1 s bis 9,1 Stunden in 1 s-Schritten

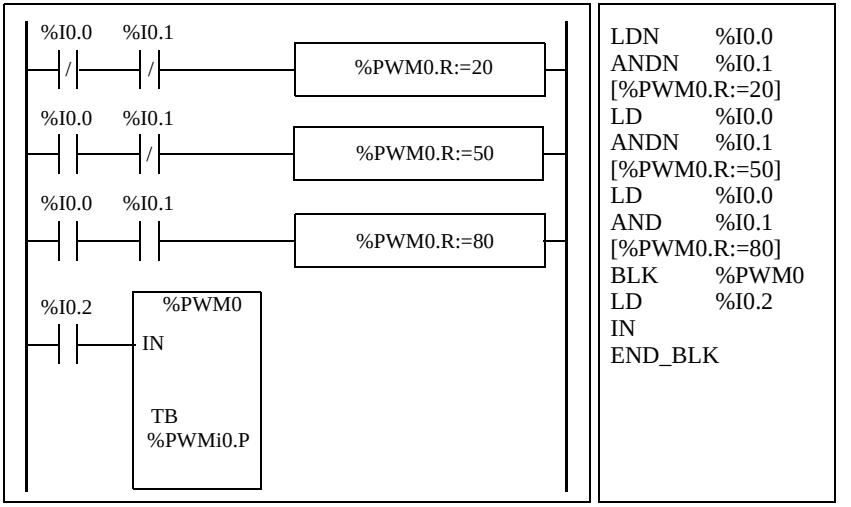
Funktionsweise

Die Frequenz des Ausgangssignals wird während der Konfiguration durch Auswählen der Zeitbasis TB und der Voreinstellung %PWMi.P eingestellt. Wird das % PWMi.R-Tastverhältnis im Programm geändert, wird die Breite des Signals moduliert. Im Folgenden sehen Sie eine Abbildung eines Impulsdiagramms für den PWM-Funktionsblock mit verschiedenen Tastverhältnissen.



Programmierung und Konfiguration

In diesem Beispiel wird die Signalbreite vom Programm entsprechend dem Zustand der Steuerungseingänge %I0.0.0 und %I0.0.1 geändert.
Wenn %I0.0.1 und %I0.0.2 auf 0 gesetzt sind, wird das %PWM0.R-Verhältnis auf 20% gesetzt und die Länge des Signals im Zustand 1 ist dann: 20 % x 500 ms = 100 ms.
Werden %I0.0.0 auf 0 und %I0.0.1 auf 1 gesetzt, wird das %PWM0.R-Tastverhältnis auf 50 % gesetzt (Länge: 250 ms).
Werden %I0.0.0 und %I0.0.1 auf 1 gesetzt, wird das %PWM0.R-Verhältnis auf 80 % gesetzt (Länge: 400 ms).
Programmierbeispiel:



Sonderfälle

Die folgende Tabelle enthält eine Liste mit Sonderfällen für die Funktionsweise des Impulsbreitenmodulations-Funktionsblocks.

Sonderfall	Beschreibung
Auswirkung eines Kaltstarts (%S0=1)	Setzt das %PWMi.R-Verhältnis auf 0. Außerdem wird der Wert von %PWMi.P auf den konfigurierten Wert zurückgesetzt, und dies übersteuert alle Änderungen, die im Animationstabellen-Editor oder im optionalen Bedienterminal vorgenommen wurde.
Auswirkung eines Warmstarts (%S1=1)	Hat keine Auswirkungen.
Fall, in dem die Ausgänge dem %PWM-Block zugewiesen sind	Das Forcieren von Ausgang %Q0.0.0 oder %Q0.0.1 mit einem Programmiergerät stoppt die Signalerzeugung nicht.

Impulsgenerator-Funktionsblock (%PLS)

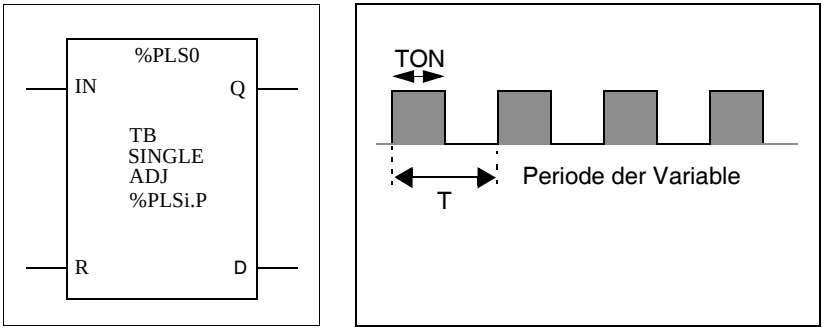
Einführung

Der Funktionsblock %PLS dient zum Erzeugen von Rechtecksignalen. An den zweckbestimmten Ausgängen %Q0.0.0 und %Q0.0.1 stehen zwei Funktionen %PLS zur Verfügung. Die Funktion %PLS erlaubt nur eine einzelne Signalbreite bzw. ein Tastverhältnis von 50 %. Sie können die Anzahl der Impulse oder den Zeitraum, während dem die Impulsfolge ausgeführt wird, begrenzen. Diese Einstellungen können Sie während der Konfiguration vornehmen und/oder sie können von der Anwenderapplikation aktualisiert werden.

Hinweis: Steuerungen mit Relaisausgängen für diese beiden Kanäle unterstützen die Funktion %PLS nicht.

Darstellung

Beispiel für den Impulsgenerator-Funktionsblock im Einzelwortmodus:



- TON=T/2 für die Zeitbasen 0,142 ms und 0,57 ms
  = (%PLSi.P*TB)/2
- TON=[ganzer Teil (%PLSi.P)/2]*TB für die Zeitbasen 10 ms bis 1 s.

Kenndaten Die folgende Tabelle enthält die Eigenschaften des Funktionsblocks PLS:

Funktion	Objekt	Beschreibung
Zeitbasis	TB	0,142 ms, 0,57 ms, 10 ms, 1 s
Voreingestellte Periode	%PLSi.P	Die Impulse am Ausgang %PLS1 werden nicht angehalten, wenn %PLS1.N oder %PLS1.ND* für die Zeitbasen 0,142 ms und 0,57 ms erreicht ist. ● 1 < %PLSi.P <= 32767 für eine Zeitbasis von 10 ms oder 1 s ● 0 < %PLSi.P <= 255 für eine Zeitbasis von 0,57 ms oder 0,142 s ● 0 = Funktion nicht verwendet Um eine hohe Tastverhältnis-Präzision mit den Zeitbasen 10 ms und 1 s zu erreichen, wird ein %PLSi >= 100 empfohlen, wenn P ungerade ist.
Anzahl Impulse	%PLSi.N %PLSi.ND*	Die Anzahl der zu erzeugenden Impulse in der Periode T kann auf den Bereich von 0 <= %PLSi.N <= 32767 im Standardmodus oder auf 0 <= %PLSi.ND <= 4294967295 im Doppelwortmodus begrenzt werden. Der Standardwert ist 0. Wenn Sie eine unbegrenzte Anzahl Impulse erzeugen möchten, setzen Sie %PLSi.N oder %PLSi.ND auf Null. Die Anzahl der Impulse kann immer verändert werden und richtet sich nicht nach der Einstellung "Verstellbar".
Verstellbar	Y/N	Wenn auf Y gesetzt, ist es möglich, den voreingestellten Wert %PLSi.P über eine Mensch/Maschine-Schnittstelle oder den Animationstabellen-Editor zu ändern. Wenn auf N gesetzt, ist die Voreinstellung nicht zugänglich.
Impulserzeugungseingang	IN	Im Zustand 1 findet die Impulserzeugung am zweckbestimmten Ausgangskanal statt. Im Zustand 0 wird der Ausgangskanal auf 0 gesetzt.
RESET-Eingang	R	Im Zustand 1 werden die Ausgänge %PLSi.Q und %PLSi.D auf 0 gesetzt. Die Anzahl der im Zeitraum T erzeugten Impulse wird auf 0 gesetzt.
Aktuelle Impulserzeugung	%PLSi.Q	Der Zustand 1 zeigt an, dass das Impulssignal am zweckbestimmten Ausgangskanal erzeugt wird.
Ausgang Impulserzeugung fertig	%PLSi.D	Im Zustand 1 ist die Signalerzeugung beendet. Die Anzahl der gewünschten Impulse wurde erzeugt.

Hinweis: (*) Bedeutet eine Doppelwortvariable.

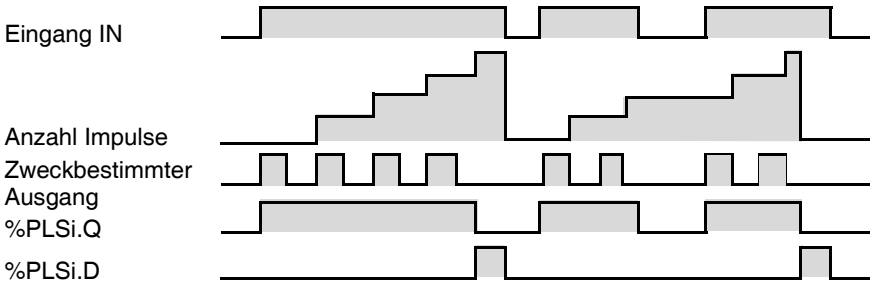
Periodenbereiche

Der voreingestellte Wert und die Zeitbasis können während der Konfiguration geändert werden. Sie dienen zum Festsetzen der Signalperiode T=%PLSi.P * TB. Verfügbare Periodenbereiche:

- 0,142 ms bis 36,5 ms in 0,142 ms-Schritten (27,4 Hz bis 7 kHz)
- 0,57 ms bis 146 ms in 0,57 ms-Schritten (6,84 Hz bis 1,75 kHz)
- 20 ms bis 5,45 min in 10 ms-Schritten
- 2 s bis 9,1 Stunden in 1 s-Schritten

Befehl

Im Folgenden sehen Sie eine Darstellung eines Funktionsblocks %PLS.



Sonderfälle

Sonderfall	Beschreibung
Auswirkung eines Kaltstarts (%S0=1)	Setzt %PLSi.P auf den während der Konfiguration definierten Wert
Auswirkung eines Warmstarts (%S1=1)	Hat keine Auswirkungen.
Auswirkung einer Änderung der Voreinstellung (%PLSi.P)	Wird sofort wirksam
Fall, in dem die Ausgänge dem %PLS-Block zugewiesen sind	Das Forcieren von Ausgang %Q0.0.0 oder %Q0.0.1 mit einem Programmiergerät stoppt die Signalerzeugung nicht.

Hinweis: %PLSx.D wird gesetzt, wenn die Anzahl der gewünschten Impulse erreicht ist. Es wird zurückgesetzt, wenn einer der Eingänge IN oder R auf 1 gesetzt wird.

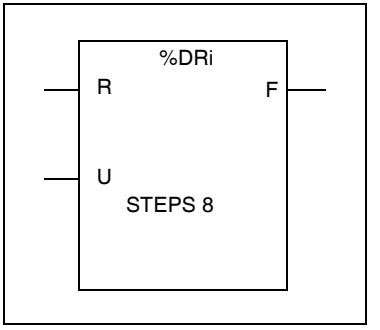
Trommelsteuerungs-Funktionsblock (%DR)

Einleitung

Die Trommelsteuerung funktioniert im Prinzip ähnlich wie eine elektromechanische Trommelsteuerung, die nach externen Ereignissen von Schritt zu Schritt wechselt. Bei jedem Schritt gibt der Hochpunkt eines Nockens einen Befehl, der automatisiert ausgeführt wird. Im Falle einer Trommelsteuerung werden diese Hochpunkte für jeden Schritt durch den Zustand 1 dargestellt und werden den Ausgangsbits %Qi.j oder den internen Bits %Mi (den sogenannten Befehlsbits) zugewiesen.

Abbildung

Das folgende Beispiel zeigt die Verwendung eines Trommelsteuerungs-Funktionsblocks.



Trommelsteuerungs-Funktionsblock

Einstellungen Der Trommelsteuerungs-Funktionsblock hat die folgenden Parameter:

Parameter	Beschriftung	Wert
Nummer	%DRi	0 bis 3 Kompaktsteuerung0 bis 7 Modulare Steuerungen
Aktuelle Schrittnummer	%DRi.S	0<%DRi.S<7. Wort, das gelesen und geschrieben werden kann. Bei einem geschriebenen Wert muss es sich um einen direkten Dezimalwert handeln. Ist das Wort geschrieben, wird sein Wert bei der nächsten Ausführung des Funktionsblocks berücksichtigt.
Anzahl der Schritte		1 bis 8 (Voreinstellung)
Eingang zum Rücksetzen auf Schritt 0 (oder auf die Anweisung)	R (Reset)	In Zustand 1; setzt die Trommelsteuerung auf den Schritt 0.
Weiterschalteingang (oder Anweisung)	U (oben)	Bei einer steigenden Flanke wird die Trommelsteuerung einen Schritt weitergeschaltet und die Steuerbits werden aktualisiert.
Ausgang	F (Voll)	Zeigt an, dass der aktuelle Schritt gleich dem letzten definierten Schritt ist. Das zugehörige Bit %DRi.F kann getestet werden (Beispiel: %DRi.F=1, wenn %DRi.S= Anzahl der konfigurierten Schritte - 1).
Steuerbits		Ausgangsbits oder interne Bits, die mit dem Schritt verbunden sind (16 Befehlsbits) und im Konfigurationseditor definiert sind.

Funktionsweise des Trommelsteuerungs-Funktionsblocks %DRi

Einführung

Die Trommelsteuerung besteht aus:

- Einer Matrix aus konstanten Daten (die Nocken), die in acht Schritten (0 bis 7) angeordnet sind, und 16 Datenbits (Zustand des Schritts), die in Spalten von 0 bis F angeordnet sind.
- Einer Liste der Steuerbits, die mit einem konfigurierten Ausgang (%Qi.j.k) oder einem Speicherwort (%Mi) verbunden ist. Während des aktuellen Schritts nehmen die Steuerbits die für diesen Schritt definierten Binärzustände an.

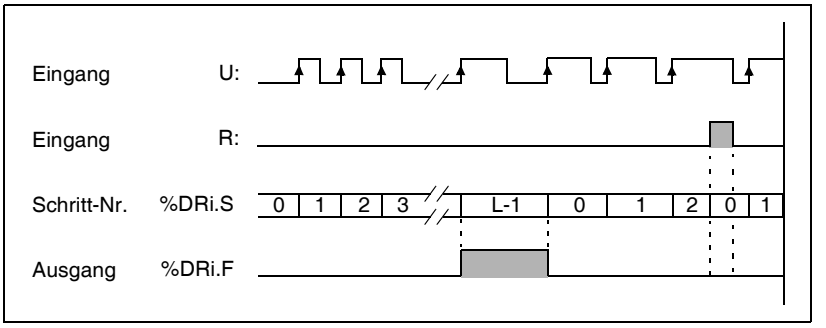
Das Beispiel in der folgenden Tabelle gibt einen Überblick über die wesentlichen Eigenschaften einer Trommelsteuerung.

Spalte	**0**	**1**	**2**		**D**	**E**	**F**
Steuerbits	%Q0.1	%Q0.3	%Q1.5		%Q0.6	%Q0.5	%Q1.0
Schritt 0	0	0	1		1	1	0
Schritt 1	1	0	1		1	0	0
Schritt 5	1	1	1		0	0	0
Schritt 6	0	1	1		0	1	0
Schritt 7	1	1	1		1	0	0

Befehl

Im Beispiel oben ist Schritt 5 der aktuelle Schritt, die Steuerbits %Q0.1, %Q0.3 und %Q1.5 sind auf 1 gesetzt, die Steuerbits %Q0.6, %Q0.5 und %Q1.0 sind auf 0 gesetzt. Die aktuelle Schrittnummer wird bei jeder steigenden Flanke am Eingang U inkrementiert (oder bei Aktivierung der Anweisung U). Der aktuelle Schritt kann vom Programm geändert werden.

Chronogramm Das folgende Diagramm zeigt die Funktionsweise der Trommelsteuerung.



Sonderfälle Die folgende Tabelle enthält eine Liste mit Sonderfällen für die Funktionsweise von Trommelsteuerungen.

Sonderfall	Beschreibung
Auswirkungen eines Kaltstarts (%S0=1)	Die Trommelsteuerung wird auf Schritt 0 zurückgesetzt (Aktualisierung der Steuerbits).
Auswirkung eines Warmstarts (%S1=1)	Die Steuerbits werden nach dem aktuellen Schritt aktualisiert.
Auswirkungen eines Programmsprungs	Wenn die Trommelsteuerung nicht mehr abgetastet wird, werden die Steuerbits dadurch nicht auf 0 zurückgesetzt.
Aktualisierung der Steuerbits	Tritt nur bei Schrittänderung oder bei Warmstart oder Kaltstart auf.

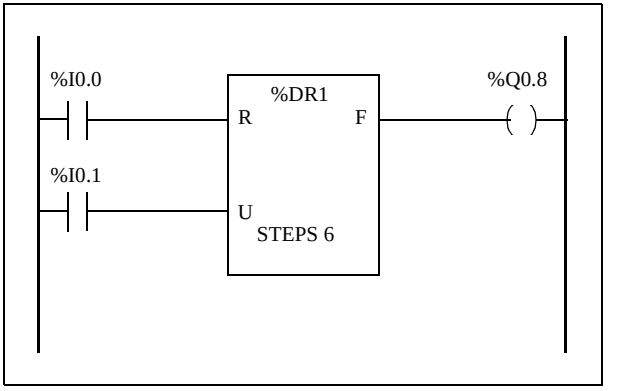
Programmieren und Konfigurieren von Trommelsteuerungen

Einleitung

Im Folgenden Beispiel für die Programmierung und Konfiguration von Trommelsteuerungen werden die ersten sechs Ausgänge %Q0.0 bis %Q0.5 nacheinander aktiviert, und zwar immer wenn Eingang %I0.1 auf 1 gesetzt wird. Eingang I0.0 setzt die Ausgänge auf 0 zurück.

Programmierbeispiel

Das folgende Bild zeigt einen Trommelsteuerungs-Funktionsblock mit Beispielen für reversible und nicht reversible Programmierung.



Kontaktplan

```
BLK     %DR1
LD      %I0.0
R
LD      %I0.1
U
OUT_BLK
LD      F
ST      %Q0.8
END_BLK
```

Konfiguration

Die folgenden Informationen werden während der Konfiguration definiert:

- Anzahl der Schritte: 6
- Die Ausgangszustände (Steuerbits) der einzelnen Schritte der Trommelsteuerung.

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
Schritt 1:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Schritt 2:	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Schritt 3:	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
Schritt 4:	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
Schritt 5:	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
Schritt 6:	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0

- Zuweisung der Steuerbits.

1 :	%Q0,0	4 :	%Q0,1
2 :	%Q0,2	5 :	%Q0.3
3 :	%Q0.4	6 :	%Q0.5

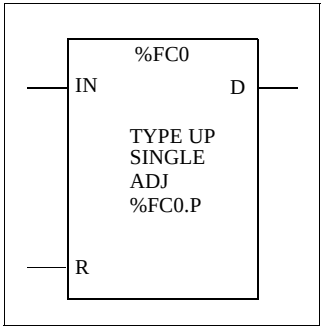
Schneller Zähler-Funktionsblock (%FC)

Einführung

Der Schneller Zähler-Funktionsblock (%FC) dient als Aufwärts- oder Abwärtszähler. Er kann die steigende Flanke von Digitaleingängen mit Frequenzen von bis zu 5 kHz im Einzelwort- oder Doppelwort-Verarbeitungsmodus zählen. Weil die schnellen Zähler von bestimmten Hardware-Interrupts verwaltet werden, richtet sich die Beibehaltung von Abtastraten mit maximaler Frequenz nach der jeweiligen Applikation und der Hardware-Konfiguration.
Die kompakten Steuerungen TWDLCA•40DRF können bis zu vier schnelle Zähler unterstützten, während alle anderen kompakten Steuerungsserien für die Verwendung von maximal drei schnellen Zählern konfiguriert werden können. Modulare Steuerungen können nur maximal zwei schnelle Zähler unterstützen. Die Funktionsblöcke "Schneller Zähler" %FC0, %FC1, %FC2 und %FC3 nutzen jeweils die zweckbestimmten Eingänge %I0.0.2, %I0.0.3, %I0.0.4 bzw. %I0.0.5. Diese Bits sind nicht für die exklusive Nutzung reserviert. Bei der Zuordnung dieser zweckbestimmten Ressourcen muss die Verwendung durch andere Funktionsblöcke berücksichtigt werden.

Abbildung

Im Folgenden sehen Sie ein Beispiel für einen Schneller Zähler-Funktionsblock im Einzelwort-Modus.



Einstellungen Die folgende Tabelle führt die Parameter des Schneller Zähler-Funktionsblocks auf.

Parameter	Markierung	Beschreibung
Funktion	TYPE	Wird während der Konfiguration ausgewählt, möglich sind Aufwärtszählen und Abwärtszählen.
Voreingestellter Wert	%FCi.P %FCi.PD	Der Ausgangswert kann gesetzt werden: ->zwischen 1 und 65635 im Standardmodus, ->zwischen 1 und 4294967295 im Doppelwortmodus.
Verstellbar	Y/N	Wenn auf Y gesetzt, können die voreingestellten Werte %FCi.P oder %FCi.PD und der aktuelle Wert %FCi.V oder %FCi.VD mit dem Bedienterminal oder dem Animationstabellen-Editor geändert werden. Wenn auf N gesetzt, ist die Voreinstellung nicht zugänglich.
Aktueller Wert	%FCi.V %FCi.VD	Der aktuelle Wert zählt die Inkremente oder Dekremente nach der eingestellten Zählrichtung aufwärts oder abwärts. Beim Aufwärtszählen wird der aktuelle Zählwert aktualisiert. Er kann bis zu 65535 im Standardmodus (%FCi.V) und bis zu 4294967295 im Doppelwortmodus (%FCi.VD) erreichen. Für die Abwärtszählung ist der aktuelle Wert der voreingestellte Wert %FCi.P oder %FCi.PD und kann bis auf Null verringert werden.
Freigabeeingang	IN	Im Zustand 1 wird der aktuelle Wert entsprechend den Impulsen am physikalischen Eingang aktualisiert. Im Zustand 0 bleibt der aktuelle Wert auf dem letzten Wert.
Rücksetzen	%FCi.R	Dient zum Initialisieren des Blocks. Im Zustand 1 wird der aktuelle Wert auf 0 zurückgesetzt, wenn ein Aufwärtszähler eingerichtet ist, oder er wird auf %FCi.P oder %FCi.PD gesetzt, wenn ein Abwärtszähler konfiguriert ist. Das Bit Fertig %FCi.D wird auf den voreingestellten Wert zurückgesetzt.
Fertig	%FCi.D	Dieses Bit wird auf 1 gesetzt, wenn %FCi.V oder %FCi.VD den Wert %FCi.P oder %FCi.PD erreicht, sofern ein Aufwärtszähler konfiguriert ist, oder wenn %FCi.V oder %FCi.VD Null erreicht, sofern ein Abwärtszähler konfiguriert ist. Dieses schreibgeschützte Bit wird nur zurückgesetzt, wenn %FCi.R auf 1 gesetzt wird.

Hinweis

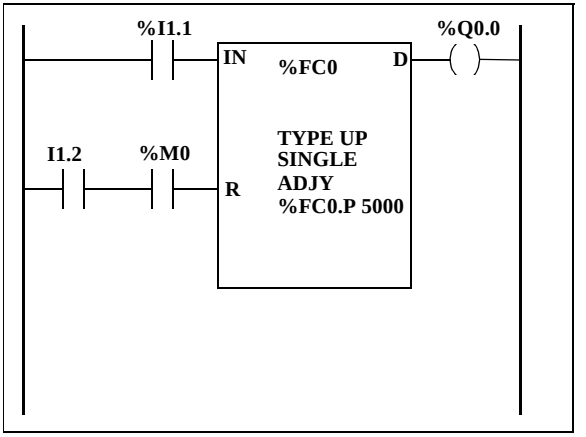
Ist der voreingestellte Wert als "Verstellbar" konfiguriert, dann kann die Applikation den voreingestellten Wert %FCi.P oder %FCi.PD und den aktuellen Wert %FCi.V oder %FCi.VD jederzeit ändern. Ein neuer Wert wird jedoch nur berücksichtigt, wenn das Rücksetzen des Eingangs aktiv ist oder wenn eine steigende Flanke an Ausgang %FCi.D auftritt. So sind unterschiedliche, aufeinander folgende Zählungen möglich, ohne dass ein einziger Impuls verloren geht.

Befehl

Ist ein Aufwärtszähler eingerichtet, wird der aktuelle Wert um 1 inkrementiert, wenn am zweckbestimmten Eingang eine steigende Flanke auftritt. Das Ausgangsbit Fertig %FCi.D wird auf 1 gesetzt, wenn der voreingestellte Wert %FCi.P oder %FCi.PD erreicht ist.
Ist ein Abwärtszähler eingerichtet, wird der aktuelle Wert um 1 dekrementiert, wenn am zweckbestimmten Eingang eine steigende Flanke auftritt. Ist der aktuelle Wert gleich Null, wird das Ausgangsbit Fertig %FCi.D auf 1 gesetzt, und der voreingestellte Wert wird in den aktuellen Wert %FCi.V oder %FCi.VD geladen.

Konfiguration und Programmierung

In diesem Beispiel zählt die Applikation eine Anzahl von Elementen bis zu 5000, während %I1.1 auf 1 gesetzt ist. Der Eingang für %FC0 ist der zweckbestimmte Eingang %I0.0.2. Ist der voreingestellte Wert erreicht, wird %FC0.D auf 1 gesetzt und behält diesen Wert, bis %FC0.R durch das Ergebnis der UND-Verknüpfung auf %I1.2 und %M0 gesetzt wird.




```
BLK      %FC0
LD       %I1.1
IN
LD       %I1.2
AND      %M0
R
OUT_BLK
LD D
ST %Q0.0
END_BLK
```

Sonderfälle

Die folgende Tabelle enthält eine Liste mit Sonderfällen für die Funktionsweise des %FC-Funktionsblocks.

Sonderfall	Beschreibung
Auswirkung eines Kaltstarts (%S0=1)	Alle %FC-Attribute werden auf die Werte zurückgesetzt, die vom Anwender oder in der Applikation konfiguriert sind.
Auswirkung eines Warmstarts (%S1=1)	Hat keine Auswirkungen.
Auswirkungen eines Steuerungsstopps	%FC zählt mit den Attributeinstellungen weiter, die wirksam waren, als die Steuerung gestoppt wurde.

Sehr schneller Zähler-Funktionsblock (%VFC)

Einführung

Der sehr schnelle Zähler-Funktionsblock (%VFC) kann in TwidoSoft für die folgenden Funktionen eingerichtet werden:

- Auf-/Abwärtszähler
- 2-Phasen-Auf-/Abwärtszähler
- Einfacher Aufwärtszähler
- Einfacher Abwärtszähler
- Frequenzmesser

Der %VFC unterstützt das Zählen von Digitaleingängen mit Frequenzen von bis zu 20 kHz im Einzelwort- oder Doppelwort-Verarbeitungsmodus zählen. Die kompakten Steuerungen TWDLCA•40DRF können bis zu zwei sehr schnelle Zähler unterstützten, während alle anderen kompakten Steuerungsserien nur für die Verwendung von einem sehr schnellen Zähler (%VFC) konfiguriert werden können. Modulare Steuerungen können für bis zu zwei sehr schnelle Zähler (%VFC) konfiguriert werden.

Zuweisungen zweckbestimmter E/A

Die sehr schnellen Zähler-Funktionsblöcke (%VFC) nutzen zweckbestimmte Eingänge sowie Hilfseingänge und Hilfsausgänge. Diese Eingänge und Ausgänge sind nicht für die exklusive Nutzung reserviert. Bei der Zuordnung dieser zweckbestimmten Ressourcen muss die Verwendung durch andere Funktionsblöcke berücksichtigt werden. Die folgende Matrix gibt einen Überblick über diese Zuweisungen:

		Haupteingänge		**Hilfseingänge**		**Reflexausgänge**	
%VFC0	Gewählte Verwendung	Eingang IA	Eingang IB	IPres	Ica	Ausgang 0	Ausgang 1
	Auf-/Abwärtszähler	%I0.0.1	%I0.0.0 (UP=0/DO=1)	%I0.0.2 (1)	%I0.0.3 (1)	%Q0.0.2 (1)	%Q0.0.3 (1)
	2-Phasen-Auf-/Abwärtszähler	%I0.0.1	%I0.0.0 (Impuls)	%I0.0.2 (1)	%I0.0.3 (1)	%Q0.0.2 (1)	%Q0.0.3 (1)
	Einfacher Aufwärtszähler	%I0.0.1	(2)	%I0.0.2 (1)	%I0.0.3 (1)	%Q0.0.2 (1)	%Q0.0.3 (1)
	Einfacher Abwärtszähler	%I0.0.1	(2)	%I0.0.2 (1)	%I0.0.3 (1)	%Q0.0.2 (1)	%Q0.0.3 (1)
	Frequenzmesser	%I0.0.1	(2)	(2)	(2)	(2)	(2)

		Haupteingänge		Hilfseingänge		Reflexausgänge	
%VFC1	Gewählte Verwendung	Eingang IA	Eingang IB)	IPres	Ica	Ausgang 0	Ausgang 1
	Auf-/Abwärtszähler	%I0.0.7	%I0.0.6 (UP=0/DO=1)	%I0.0.5 (1)	%I0.0.4 (1)	%Q0.0.4 (1)	%Q0.0.5 (1)
	2-Phasen-Auf-/ Abwärtszähler	%I0.0.7	%I0.0.6 (Impuls)	%I0.0.5 (1)	%I0.0.4 (1)	%Q0.0.4 (1)	%Q0.0.5 (1)
	Einfacher Aufwärtszähler	%I0.0.7	(2)	%I0.0.5 (1)	%I0.0.4 (1)	%Q0.0.4 (1)	%Q0.0.5 (1)
	Einfacher Abwärtszähler	%I0.0.7	(2)	%I0.0.5 (1)	%I0.0.4 (1)	%Q0.0.4 (1)	%Q0.0.5 (1)
	Frequenzmesser	%I0.0.7	(2)	(2)	(2)	(2)	(2)

Kommentare:

(1) = optional

(2) = nicht verwendet

Ipres = Preset-Eingang

Ica= Selbsthalteeingang

Eingang IA = Impulseingang

Eingang IB = Impulse oder UP/DO

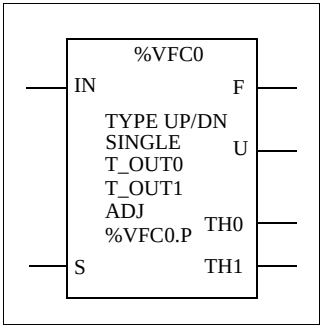
UP/DO = Auf-/Abwärtszählen

Wird der Eingang oder Ausgang nicht verwendet, bleibt er ein normaler digitaler E/A, der im Hauptzyklus von der Applikation verwaltet werden kann.

Wenn %I0.0.2 verwendet wird, ist %FC0 nicht verfügbar.
Wenn %I0.0.3 verwendet wird, ist %FC2 nicht verfügbar.
Wenn %I0.0.4 verwendet wird, ist %FC3 nicht verfügbar.

Abbildung

Im Folgenden sehen Sie eine Blockdarstellung eines sehr schnellen Zählers (%VFC) im Einzelwortmodus:



Kenndaten Die folgende Tabelle führt die Eigenschaften des sehr schnellen Zähler-Funktionsblocks (%VFC) auf.

Funktion	Beschreibung	Werte	Verwendung des %VFC	Zugriff zur Laufzeit
Aktueller Wert (%VFCi.V) (%VFCi.VD*)	Aktueller Wert, der entsprechend den physikalischen Eingängen und der gewählten Funktion erhöht oder verringert wird. Dieser Wert kann mittels des Preset-Eingangs (%VFCi.S) voreingestellt oder initialisiert werden.	%VFCi.V: 0 -> 65535 %VFCi.VD: 0 -> 4294967295	CM	Lesen
Voreingestellter Wert (%VFCi.P) (%VFCi.PD*)	Wird nur in der Funktion zum Auf-/Abwärtszählen und beim einfachen Aufwärts- oder Abwärtszählen verwendet.	%VFCi.P: 0 -> 65535 %VFCi.PD: 0 -> 4294967295	CM oder FM	Lesen und Schreiben (1)
Wert erfassen (%VFCi.C) (%VFCi.CD*)	Wird nur in der Funktion zum Auf-/Abwärtszählen und beim einfachen Aufwärts- oder Abwärtszählen verwendet.	%VFCi.C: 0 -> 65535 %VFCi.CD: 0 -> 4294967295	CM	Lesen
Zählrichtung (%VFCi.U)	Wird vom System gesetzt. Dieses Bit wird von der Funktion zum Auf-/Abwärtszählen verwendet und zeigt die Zählrichtung an: Bei einem 1-Phasen-Aufwärts- oder -Abwärtszähler gibt %I0.0.0 die Richtung für %VFC0 an und %I0.0.6 gibt die Richtung für %VFC1 an. Bei einem 2-Phasen-Auf-/Abwärtszähler legt die Phasendifferenz zwischen den beiden Signalen die Richtung fest. Für %VFC0 sind %I0.0 IB und %I0.1 IA zugewiesen. Für %VFC1 sind %I0.6 IB und %I0.7 IA zugewiesen.	0 (Abwärtszählen) 1 (Aufwärtszählen)	CM	Lesen
Reflexausgang 0 freigeben (%VFCi.R)	Freigabe Reflexausgang 0	0 (Deaktivieren) 1 (Freigeben)	CM	Lesen und Schreiben (2)
Reflexausgang 1 freigeben (%VFCi.S)	Freigabe Reflexausgang 1	0 (Deaktivieren) 1 (Freigeben)	CM	Lesen und Schreiben (2)
Grenzwert S0 (%VFCi.S0) (%VFCi.S0D*)	Dieses Wort enthält den Grenzwert 0. Die Bedeutung wird während der Konfiguration des Funktionsblocks definiert. Hinweis: Dieser Wert muss kleiner als %VFCi.S1 sein.	%VFCi.S0: 0 -> 65535 %VFCi.S0D: 0 -> 4294967295	CM	Lesen und Schreiben (1)

Funktion	Beschreibung	Werte	Verwendung des %VFC	Zugriff zur Laufzeit
Grenzwert S1 (%VFCi.S1) (%VFCi.S1D*)	Dieses Wort enthält den Grenzwert 0. Die Bedeutung wird während der Konfiguration des Funktionsblocks definiert. Hinweis: Dieser Wert muss größer als %VFCi.S0 sein.	%VFCi.S1: 0 -> 65535 %VFCi.S1D: 0 -> 4294967295	CM	Lesen und Schreiben (1)
Zeitbasis Frequenzmessung (%VFCi.T)	Konfigurationselement der Zeitbasis (100 ms oder 1000 ms).	1000 oder 100	FM	Lesen und Schreiben (1)
Verstellbar (Y/N)	Konfigurierbares Element, mit dem der Anwender, sofern es gewählt ist, die Voreinstellung, den Grenzwert und die Zeitbasis für die Frequenzmessung im Betrieb einstellen kann.	N (Nein) Y (Ja)	CM oder FM	Nein
Freigabeeingang (IN)	Dient zum Validieren oder Sperren der aktuellen Funktion.	0 (Nein)	CM oder FM	Lesen und Schreiben (3)
Preset-Eingang (S)	Je nach Konfiguration im Zustand 1: • Auf-/Abwärtszählen oder Abwärtszählen: initialisiert den aktuellen Wert mit dem Preset-Wert. • Einfaches Aufwärtszählen: Aktueller Wert wird auf Null gesetzt Außerdem werden die Grenzwertausgänge initialisiert und alle vom Anwender vorgenommenen Änderungen an den Grenzwerten berücksichtigt, die über das Bedienterminal oder das Anwenderprogramm eingegeben wurden.	0 oder 1	CM oder FM	Lesen und Schreiben
Überlaufausgang (F)	0 bis 65535 oder 65535 bis 0 im Standardmodus 0 bis 4294967295 oder 4294967295 bis 0 im Doppelwortmodus	0 oder 1	CM	Lesen
Grenzwert Bit 0 (%VFCi.TH0)	Wird auf 1 gesetzt, wenn der aktuelle Wert größer als oder gleich dem Grenzwert %VFCi.S0 ist. Es ist ratsam, dieses Bit nur einmal im Programm zu testen, weil es in Echtzeit aktualisiert wird. Die Anwenderapplikation ist verantwortlich für die Gültigkeit des Werts zum Zeitpunkt des Einsatzes.	0 oder 1	CM	Lesen

Funktion	Beschreibung	Werte	Verwendung des %VFC	Zugriff zur Laufzeit
Grenzwert S1 (%VFCi.S1) (%VFCi.S1D*)	Dieses Wort enthält den Grenzwert 0. Die Bedeutung wird während der Konfiguration des Funktionsblocks definiert. Hinweis: Dieser Wert muss größer als %VFCi.S0 sein.	%VFCi.S1: 0 -> 65535 %VFCi.S1D: 0 -> 4294967295	CM	Lesen und Schreiben (1)
Zeitbasis Frequenzmessung (%VFCi.T)	Konfigurationselement der Zeitbasis (100 ms oder 1000 ms).	1000 oder 100	FM	Lesen und Schreiben (1)
Verstellbar (Y/N)	Konfigurierbares Element, mit dem der Anwender, sofern es gewählt ist, die Voreinstellung, den Grenzwert und die Zeitbasis für die Frequenzmessung im Betrieb einstellen kann.	N (Nein) Y (Ja)	CM oder FM	Nein
Freigabeeingang (IN)	Dient zum Validieren oder Sperren der aktuellen Funktion.	0 (Nein)	CM oder FM	Lesen und Schreiben (3)
Preset-Eingang (S)	Je nach Konfiguration im Zustand 1: ● Auf-/Abwärtszählen oder Abwärtszählen: initialisiert den aktuellen Wert mit dem Preset-Wert. ● Einfaches Aufwärtszählen: Aktueller Wert wird auf Null gesetzt Außerdem werden die Grenzwertausgänge initialisiert und alle vom Anwender vorgenommenen Änderungen an den Grenzwerten berücksichtigt, die über das Bedienterminal oder das Anwenderprogramm eingegeben wurden.	0 oder 1	CM oder FM	Lesen und Schreiben
Überlaufausgang (F)	0 bis 65535 oder 65535 bis 0 im Standardmodus 0 bis 4294967295 oder 4294967295 bis 0 im Doppelwortmodus	0 oder 1	CM	Lesen
Grenzwert Bit 0 (%VFCi.TH0)	Wird auf 1 gesetzt, wenn der aktuelle Wert größer als oder gleich dem Grenzwert %VFCi.S0 ist. Es ist ratsam, dieses Bit nur einmal im Programm zu testen, weil es in Echtzeit aktualisiert wird. Die Anwenderapplikation ist verantwortlich für die Gültigkeit des Werts zum Zeitpunkt des Einsatzes.	0 oder 1	CM	Lesen

Funktion	Beschreibung	Werte	Verwendung des %VFC	Zugriff zur Laufzeit
Grenzwert Bit 1 (%VFCi.TH1)	Wird auf 1 gesetzt, wenn der aktuelle Wert größer als oder gleich dem Grenzwert %VFCi.S1 ist. Es ist ratsam, dieses Bit nur einmal im Programm zu testen, weil es in Echtzeit aktualisiert wird. Die Anwenderapplikation ist verantwortlich für die Gültigkeit des Werts zum Zeitpunkt des Einsatzes.	0 oder 1	CM	Lesen

(*)Bedeutet eine 32-Bit-Doppelwortvariable. Die Doppelwortoption ist an allen Steuerungen mit Ausnahme der Twido-Steuerungen TWDLC•A10DRF verfügbar.
(1) Kann nur geschrieben werden, wenn Einstellen auf 1 gesetzt ist.
(2) Zugriff nur verfügbar, wenn konfiguriert.
(3)Lese- und Schreibzugriff nur über die Applikation, nicht über das Bedienterminal oder den Animationstabellen-Editor.
CM = Zählmodus
FM = Frequenzmessmodus

Beschreibung der Zählfunktion

Der sehr schnelle Zähler (%VFC) funktioniert mit einer maximalen Frequenz von 20 kHz und einem Bereich von 0 bis 65535 im Standardmodus und einem Bereich von 0 bis 4294967295. Die zu zählenden Impulse werden auf folgende Weise angelegt: Tabelle:

Funktion	Beschreibung	%VFC0		%VFC1	
		IA	IB	IA	IB
Auf-/Abwärtszähler	Die Impulse liegen am physikalischen Eingang an, die aktuelle Funktionsweise (Auf-/Abwärtszählen) wird durch den Zustand des physikalischen Eingangs IB angegeben.	%I0.0.1	%I0.0.0	%I0.0.7	%I0.0.6
2-Phasen-Auf-/ Abwärtszähler	Die beiden Phasen des Gebers liegen an den physikalischen Eingängen IA und IB an.	%I0.0.1	%I0.0.0	%I0.0.7	%I0.0.6
Einfacher Aufwärtszähler	Die Impulse liegen am physikalischen Eingang IA an. IB wird nicht verwendet.	%I0.0.1	NA	%I0.0.7	NA
Einfacher Abwärtszähler	Die Impulse liegen am physikalischen Eingang IA an. IB wird nicht verwendet.	%I0.0.1	NA	%I0.0.7	NA

Hinweise zu Funktions-blöcken

Die Auf- und Abwärtszählvorgänge werden bei steigenden Flanken an Impulsen durchgeführt und nur, wenn der Zählblock aktiviert ist.
Es gibt zwei Eingänge, die für den Zählmodus verwendet werden: ICa und IPres. ICa dient zum Erfassen des aktuellen Werts (%VFCi.V oder %VFCi.VD), der in %VFCi.C oder %VFCi.CD gespeichert wird. Die Eingänge Ica werden angegeben als %I0.0.3 für %VFC0 und %I0.0.4 für %VFC1, sofern verfügbar.
Ist der Eingang IPres aktiv, wirkt sich dies folgendermaßen auf den aktuellen Wert aus:

- Beim Aufwärtszählen wird %VFCi.V oder %VFCi.VD auf 0 zurückgesetzt.
- Beim Abwärtszählen wird %VFCi.V oder %VFCi.VD mit dem Inhalt von %VFCi.P bzw. %VFCi.PD geschrieben.
- Beim Frequenzzählen werden %VFCi.V oder %VFCi.PD auf 0 gesetzt.

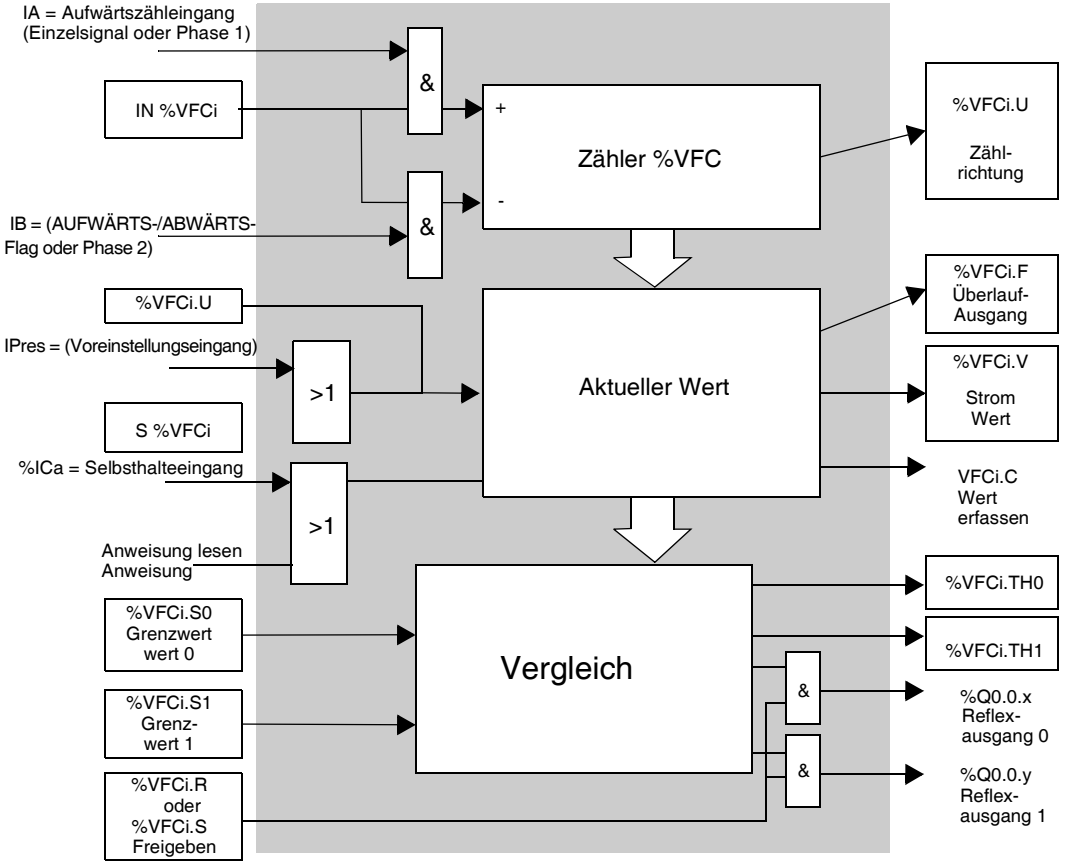
Achtung: %VFCi.F wird ebenfalls auf 0 gesetzt. Die Eingänge IPres werden angegeben als %I0.0.2 für %VFC0 und %I0.0.5 für %VFC1, sofern verfügbar.

Hinweise zu Funktionsblock-ausgängen

Bei allen Funktionen wird der aktuelle Wert mit zwei Grenzwerten verglichen (%VFCi.S0 oder %VFCi.S0D und % VFCi.S1 oder %VFCi.S1D). Je nach dem Ergebnis dieses Vergleichs werden zwei Bitobjekte (%VFCi.TH0 und %VFCi.TH1) auf 1 gesetzt, wenn der aktuelle Wert größer als oder gleich dem entsprechenden Grenzwert ist bzw. sie werden im umgekehrten Fall auf 0 zurückgesetzt. Reflexausgänge (sofern konfiguriert) werden entsprechend diesen Vergleichen auf 1 gesetzt. Hinweis: Es können keine, 1 oder 2 Ausgänge konfiguriert werden. %VFC.U ist ein Ausgang des FB, er gibt die Richtung des zugehörigen Zählers an (1 bei AUF, 0 bei AB).

Diagramm für die Zählfunktion

Im Folgenden sehen Sie ein Diagramm für die Zählfunktion im Standardmodus (im Doppelwortmodus verwenden Sie die entsprechenden Variablen der Doppelwortfunktion):



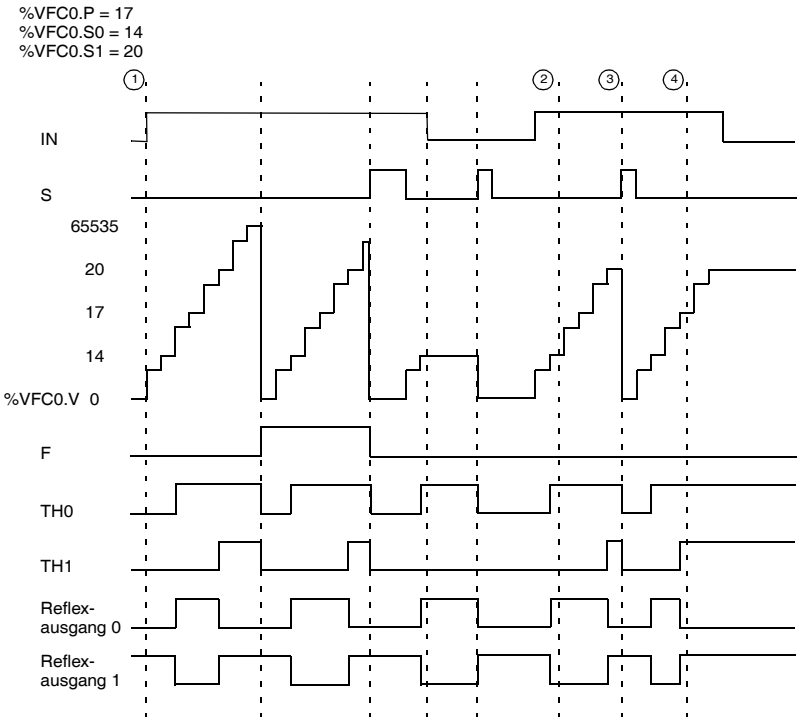
Hinweis: Die Ausgänge werden unabhängig von der Zykluszeit der Steuerung verwaltet. Die Antwortzeit liegt zwischen 0 und 1 ms.

Funktionsweise des einfachen Aufwärtszählers

Im Folgenden sehen Sie ein Beispiel für den Einsatz von %VFC als einfachen Aufwärtszähler. Die folgenden Konfigurationselemente wurden für dieses Beispiel eingestellt:
Der voreingestellte Wert %VFC0.P ist 17, während der untere Grenzwert %VFC0.S0 14 und der obere Grenzwert %VFC0.S1 20 beträgt.

Reflexausgang	Wert < %VFC.S0	%VFC0.S0 <= Wert < %VFC0.S1	Wert >= %VFC0.S1
%Q0.0.2		X	
%Q0.0.3	X		X

Im Folgenden sehen Sie ein Zeitdiagramm:



(1) : %VFC0.U = 1, weil %VFC ein Aufwärtszähler ist

(2) : %VFC0.S1 auf 17 ändern

(3) : Wenn der Eingang S aktiv ist, wird der neue Grenzwert S1 beim nächsten Zählen zugelassen.

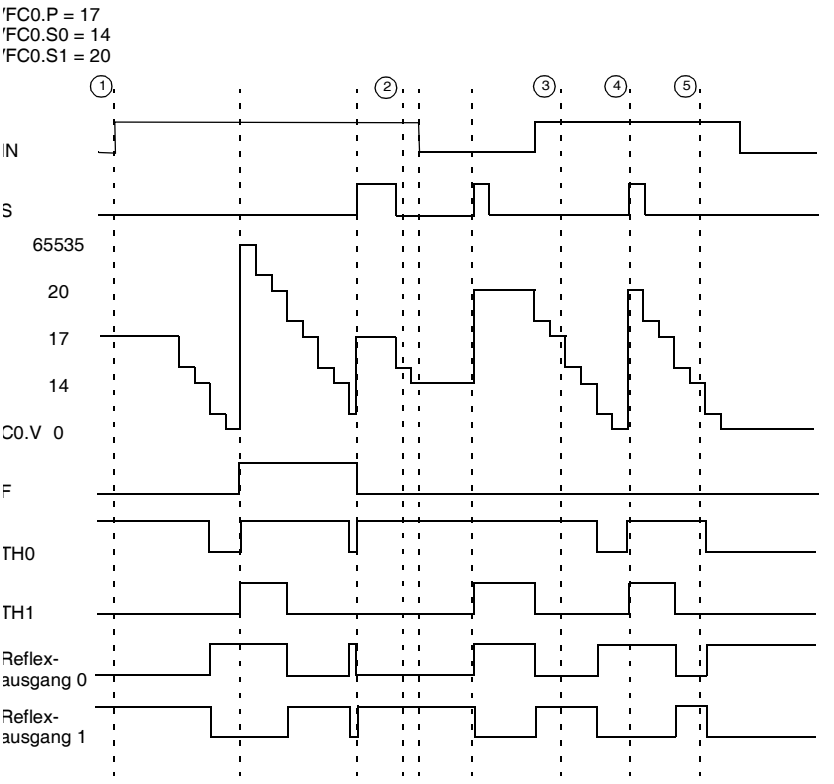
(4) : Der aktuelle Wert wird erfasst, deshalb ist %VFC0.C = 17.

Funktionsweise des einfachen Abwärtszählers

Im Folgenden sehen Sie ein Beispiel für den Einsatz von %VFC als einfachen Abwärtszähler. Die folgenden Konfigurationselemente wurden für dieses Beispiel eingestellt:
Der voreingestellte Wert %VFC0.P ist 17, während der untere Grenzwert %VFC0.S0 14 und der obere Grenzwert %VFC0.S1 20 beträgt.

Reflexausgang	Wert < %VFC.S0	%VFC0.S0 <= Wert < %VFC0.S1	Wert >= %VFC0.S1
%Q0.0.2	X		X
%Q0.0.3		X	

Beispiel:



: %VFC0.U = 0, weil %VFC ein Abwärtszähler ist

: %VFC0.P auf 20 ändern

: %VFC0.S1 auf 17 ändern

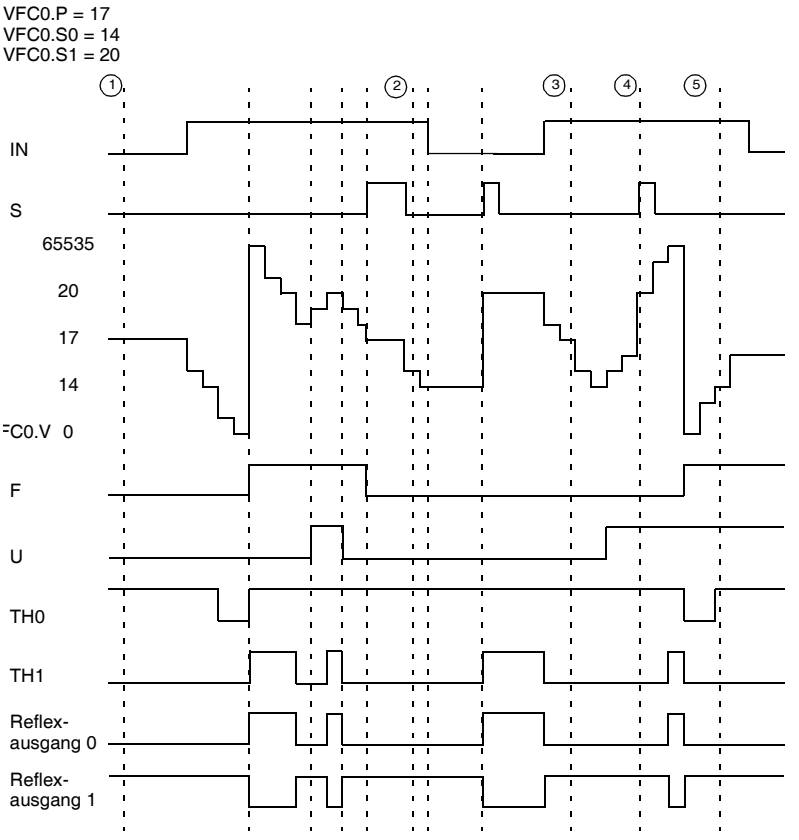
: Wenn der Eingang S aktiv ist, wird der neue Grenzwert S1 beim nächsten Zählen zugelassen.

Funktionsweise des Auf-/ Abwärtszählers

Im Folgenden sehen Sie ein Beispiel für den Einsatz von %VFC als Auf-/ Abwärtszähler. Die folgenden Konfigurationselemente wurden für dieses Beispiel eingestellt:
Der voreingestellte Wert %VFC0.P ist 17, während der untere Grenzwert %VFC0.S0 14 und der obere Grenzwert %VFC0.S1 20 beträgt.

Reflexausgang	Wert < %VFC.S0	%VFC0.S0 <= Wert < %VFC0.S1	Wert >= %VFC0.S1
%Q0.0.2			X
%Q0.0.3	X	X	

Beispiel:



: Eingang IN ist 1 und Eingang S ist 1

: %VFC0.P auf 20 ändern

: %VFC0.S1 auf 17 ändern

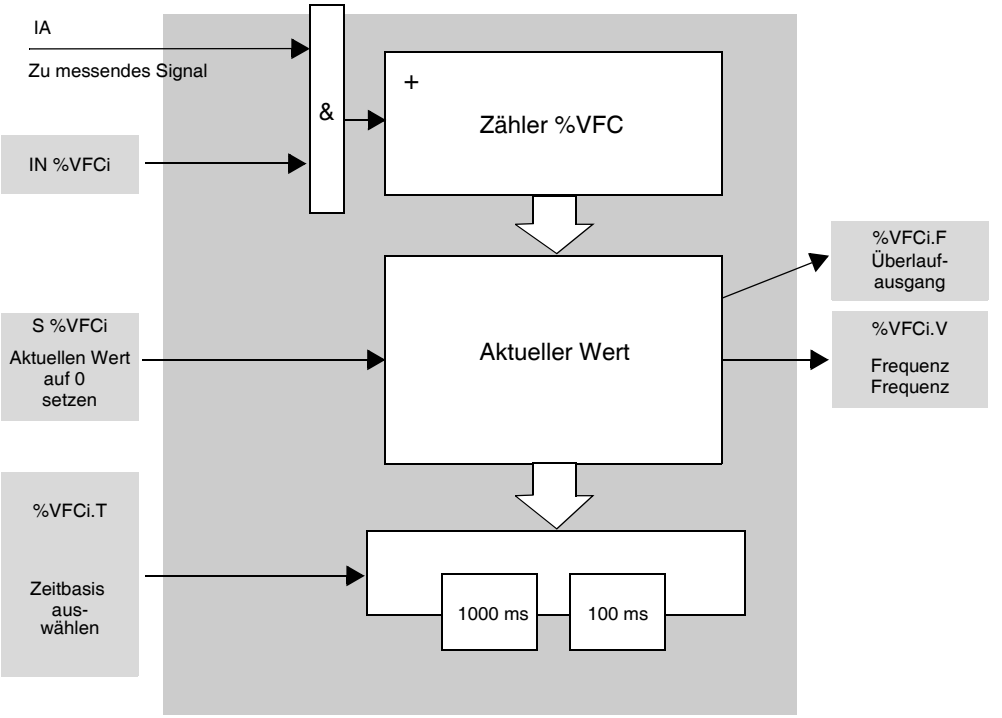
Beschreibung der Funktion des Frequenz-messers

Die Frequenzmessfunktion eines %VFC dient zum Messen der Frequenz eines periodischen Signals am Eingang IA in Hz. Der Frequenzbereich, der gemessen werden kann, liegt zwischen 10 Hz und 20 kHz. Der Anwender kann eine Zeitbasis auswählen (zwei stehen zur Auswahl), wobei die Auswahl über ein neues Objekt %VFC.T (Zeitbasis) getroffen wird. Der Wert 100 = Zeitbasis von 100 ms und der Wert 1000 = Zeitbasis von 1 Sekunde.

Zeitbasis	Messbereich	Genauigkeit	Aktualisierung
100 ms	100 Hz bis 20 kHz	0,05 % für 20 kHz, 10 % für 100 Hz	10 Mal pro Sekunde
1 s	10 Hz bis 20 kHz	0,005 % für 20 kHz, 10 % für 10 Hz	Einmal pro Sekunde

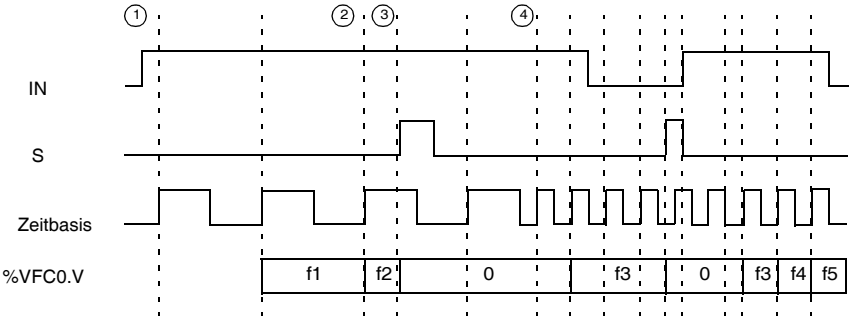
Funktionsdi-agramm für den Frequenzmesser

Im Folgenden sehen Sie ein Funktionsdiagramm für den Frequenzmesser:



Funktionsweise des Frequenz-messers

Im Folgenden sehen Sie ein Beispiel für ein Zeitdiagramm bei Einsatz von %VFC als Frequenzmesser.



① : Die erste Frequenzmessung beginnt hier.

② : Der aktuelle Frequenzwert wird aktualisiert.

③ : Eingang IN ist 1 und Eingang S ist 1

④ : %VFC0.T auf 100 ms ändern: Durch diese Änderung wird die aktuelle Messung abgebrochen und eine neue Messung begonnen.

Sonderfälle

Die folgende Tabelle enthält eine Liste mit Sonderfällen für die Funktionsweise des %VFC-Funktionsblocks.

Sonderfall	Beschreibung
Auswirkung eines Kaltstarts (%S0=1)	Alle %VFC-Attribute werden auf die Werte zurückgesetzt, die vom Anwender oder in der Anwenderapplikation konfiguriert sind.
Auswirkung eines Warmstarts (%S1=1)	Hat keine Auswirkungen.
Auswirkungen eines Steuerungsstopps	Der %VFC stoppt und die Ausgänge bleiben im aktuellen Zustand.

Senden/Empfangen von Meldungen - die Austauschanweisung (EXCH)

Einführung

Sie können eine Twido-Steuerung so konfigurieren, dass sie mit Modbus-Slave-Geräten kommunizieren oder Meldungen im Zeichenmodus (ASCII) senden und/oder empfangen kann.
TwidoSoft bietet für diese Kommunikation die folgenden Funktionen:

- Anweisung EXCH zum Senden/Empfangen von Meldungen
- Austauschsteuerungs-Funktionsblock (%MSG) für die Steuerung des Datenaustauschs

Die Twido-Steuerung nutzt das für den angegebenen Port eingerichtete Protokoll für die Bearbeitung der Anweisung EXCH. Jedem Kommunikationsport kann ein anderes Protokoll zugewiesen werden. Auf die Kommunikationsports wird zugegriffen, indem die entsprechende Portnummer an die Funktion EXCH oder %MSG angehängt wird (EXCH1, EXCH2, %MSG1, %MSG2).
Drüber hinaus implementieren Steuerungen der Serie TWDLCAE40DRF die Modbus TCP-Nachrichtenübertragung über das Ethernet-Netzwerk mittels der Anweisung EXCH3 und der Funktion %MSG3.

Anweisung EXCH

Mit der Anweisung EXCH kann eine Twido-Steuerung Informationen von ASCII-Geräten empfangen und Informationen an ASCII-Geräte senden. Der Anwender definiert eine Tabelle mit Wörtern (%MWi:L), die die Daten enthalten, die gesendet und/oder empfangen werden sollen (maximal 250 Datenbytes zum Senden und/oder Empfangen). Das Format der Worttabelle wird in den Abschnitten zu den einzelnen Protokollen beschrieben. Ein Meldungsaustausch wird mit der Anweisung EXCH durchgeführt.

Syntax

Die Anweisung EXCH hat das folgende Format:
[EXCHx %MWi:L]
wobei: x = Nummer des seriellen Ports (1 oder 2); x = Ethernet-Port (3); L = Gesamtzahl Wörter in der Worttabelle (maximal 121). Werte der internen Worttabelle %MWi:L sind Werte wie i+L <= 255.
Die Twido-Steuerung muss den Austausch einer ersten Anweisung EXCHx beenden, bevor eine zweite Austauschanweisung gestartet werden kann. Der Funktionsblock %MSG muss verwendet werden, wenn mehrere Meldungen gesendet werden sollen.

Hinweis: Ausführliche Informationen über die Modbus TCP-Nachrichtenübertragungsanweisung EXCH3 finden Sie unter *TCP Modbus-Nachrichtenübertragung, S. 190*.

Austauschsteuerungs-Funktionsblock (%MSGx)

Einführung

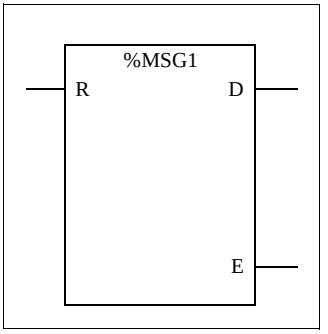
Hinweis: Das "x" in %MSGx gibt den Port der Steuerung an: "x = 1 oder 2"
- x = 1 oder 2, gibt den seriellen Port 1 bzw. 2 der Steuerung an;
- x = 3, gibt den Ethernet-Netzwerk-Port der Steuerung an (nur an Steuerungen der Serie TWDLCAE40DRF). Weitere Informationen über die Funktion %MSG3 finden Sie unter *TCP Modbus-Nachrichtenübertragung, S. 190*.

Der Funktionsblock %MSGx verwaltet den Datenaustausch und verfügt über drei Funktionen:
- Kommunikationsfehlerprüfung:
  Durch die Fehlerprüfung wird bestätigt, dass die mit der Anweisung EXCH programmierte Blocklänge (Worttabelle) groß genug ist, um die Länge der zu sendenden Meldung aufzunehmen (Vergleich mit der im niederwertigsten Byte des ersten Worts der Worttabelle programmierten Länge).
- Koordination mehrerer Meldungen:
  Um beim Senden von mehreren Meldungen die Koordination sicherzustellen, liefert der Funktionsblock %MSGx die Informationen, anhand derer festgestellt werden kann, wann die vorherige Meldung beendet ist.
- Senden von Prioritätsmeldungen:
  Der Funktionsblock %MSGx lässt es zu, dass die aktuelle Meldungsübertragung gestoppt wird, damit eine dringende Meldung sofort gesendet werden kann.

Die Programmierung des Funktionsblocks %MSGx ist optional.

Abbildung

Im Folgenden sehen Sie ein Beispiel für einen Funktionsblock %MSGx.



Einstellungen Die folgende Tabelle führt die Parameter des %MSGx-Funktionsblocks auf.

Parameter	Markierung	Wert
RESET-Eingang (oder Anweisung)	R	Im Zustand 1 wird die Kommunikation neu initialisiert: %MSGx.E = 0 und %MSGx.D = 1.
Ausgang Komm. beendet	%MSGx.D	Zustand 1, Komm. beendet, wenn: ● Ende der Übertragung (sofern Übertragung) ● Ende des Empfangs (Endezeichen empfangen) ● Fehler ● Block zurücksetzen Zustand 0, Request in Bearbeitung.
Fehlerausgang	%MSGx.E	Zustand 1, Komm. beendet, wenn: ● Falscher Befehl ● Tabelle falsch konfiguriert ● Falsches Zeichen empfangen (Geschwindigkeit, Parität usw.) ● Empfangstabelle voll (nicht aktualisiert) Zustand 0, Meldungslänge OK, Verbindung OK.

Tritt bei Ausführung einer EXCH-Anweisung ein Fehler auf, werden die Bits %MSGx.D und %MSGx.E auf 1 gesetzt und das Systemwort %SW63 enthält den Fehlercode für Schnittstelle 1 und das Systemwort %SW64 enthält den Fehlercode für Schnittstelle 2. Siehe *Systemwörter (%SW), S. 633*.

RESET-Eingang (R)

Wenn der RESET-Eingang auf 1 gesetzt wird:
- Alle Meldungen, die übertragen werden, werden gestoppt.
- Der Fehlerausgang wird auf 0 zurückgesetzt.
- Das Bit Fertig wird auf 1 gesetzt.

Jetzt kann eine neue Meldung gesendet werden.

Fehlerausgang (%MSGx.E)

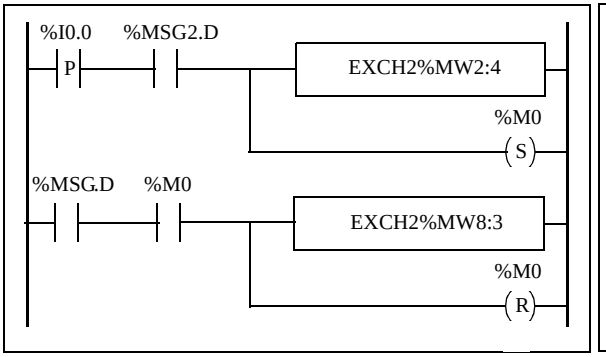
Der Fehlerausgang wird auf 1 gesetzt, weil entweder ein Kommunikationsprogrammierfehler oder ein Meldungsübertragungsfehler aufgetreten ist. Der Fehlerausgang wird auf 1 gesetzt, wenn die Anzahl der Bytes, die im Datenblock der Anweisung EXCH definiert sind (Wort 1, niederwertigstes Byte), größer als 128 ist (+80 in hexadezimaler Schreibweise je FA).
Der Fehlerausgang wird auch auf 1 gesetzt, wenn beim Senden einer Modbus-Meldung an ein Modbus-Gerät ein Fehler auftritt. In diesem Fall sollte die Verdrahtung überprüft werden und es sollte geprüft werden, dass das Zielgerät Modbus-Kommunikation unterstützt.

Ausgang Kommunikation beendet (%MSGx.D)

Wird das Bit Fertig auf 1 gesetzt, ist die Twido-Steuerung bereit, eine neue Meldung zu senden. Die Verwendung des Bits %MSGx.D ist empfohlen, wenn mehrere Meldungen gesendet werden sollen. Wird das Bit nicht verwendet, gehen möglicherweise Meldungen verloren.

Übertragung mehrerer aufeinander folgender Meldungen

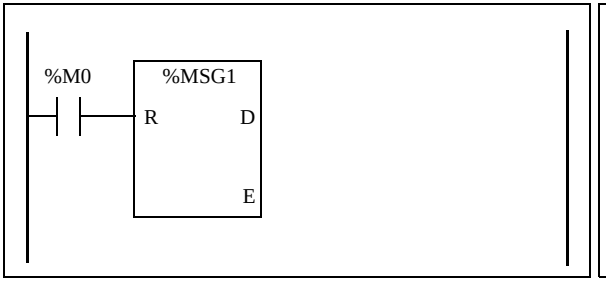
Die Ausführung der Anweisung EXCH aktiviert einen Meldungsblock im Applikationsprogramm. Die Meldung wird gesendet, wenn der Meldungsblock nicht bereits aktiv ist (%MSGx.D = 1). Werden mehrere Meldungen im gleichen Zyklus gesendet, wird nur die erste Meldung übertragen. Der Anwender ist verantwortlich für die Verwaltung der Übertragung von mehreren Meldungen im Programm.
Beispiel für eine Übertragung von zwei aufeinander folgenden Meldungen an Port 2:




```
LDR     %I0.0
AND     %MSG2.D
[EXCH2 %MW2:4]
S       %M0
LD      %MSG2.D
AND     %M0
[EXCH2 %MW8:3]
R       %M0
```

Neuinitialisierung des Austauschs

Ein Austausch wird durch Aktivierung des Eingangs (oder der Anweisung) R abgebrochen. Dieser Eingang initialisiert die Kommunikation und setzt den Ausgang %MSG.E auf 0 und den Ausgang %MSG.D auf 1 zurück. Es ist möglich, einen Datenaustausch neu zu initialisieren, wenn ein Fehler erkannt wurde.
Beispiel für die Neuinitialisierung eines Austauschs:




```
BLK     %MSG1
LD      %M0
R
END_BLK
```

Sonderfälle Die folgende Tabelle beschreibt die Sonderfälle der Funktionsweise des Funktionsblocks %MSGx.

Sonderfall	Beschreibung
Auswirkung eines Kaltstarts (%S0=1)	Forciert eine Neuinitialisierung der Kommunikation.
Auswirkung eines Warmstarts (%S1=1)	Hat keine Auswirkungen.
Auswirkungen eines Steuerungsstopps	Ist eine Meldungsübertragung in Bearbeitung, stoppt die Steuerung die Übertragung und initialisiert die Ausgänge %MSGx.D und %MSGx.E neu.

17.2 Uhrfunktionen

Auf einen Blick

Inhalt des Abschnitts

Dieser Abschnitt enthält eine Beschreibung der Funktionen zur Uhrzeitverwaltung der Twido-Steuerungen.

Inhalt dieses Abschnitts

Dieser Abschnitt enthält die folgenden Themen:

Thema	Seite
Uhrfunktionen	500
Abarbeitungsblöcke	501
Uhrzeit-/Datumsstempel	504
Einstellen von Datum und Uhrzeit	506

Uhrfunktionen

Einführung

Twido Steuerungen verfügen über eine Echtzeituhrfunktion, die die optionale Echtzeituhr (RTC) benötigt und folgendes bietet:

- **Abarbeitungsblöcke** steuern Aktionen zu vordefinierten oder berechneten Zeiten.
- Der **Zeit-/Datumsstempel** weist Ereignissen eine Uhrzeit und ein Datum zu und misst die Länge von Ereignissen.

Die Twido Echtzeituhr rufen Sie im TwidoSoft-Menü **Software** mit dem Befehl **Abarbeitungsblöcke** auf. Außerdem kann die Echtzeituhr von einem Programm eingestellt werden. Die Uhreinstellungen funktionieren noch bis zu 30 Tage nach Ausschalten der Steuerung, sofern die Batterie vor Ausschalten der Steuerung mindestens sechs Stunden lang geladen wurde.
Die Echtzeituhr hat ein 24-Stunden-Format und berücksichtigt Schaltjahre.

Korrekturwert der Echtzeituhr

Der Korrekturwert der Echtzeituhr ist für den einwandfreien Betrieb der Echtzeituhr erforderlich. Jede Echtzeituhr hat einen eigenen Korrekturwert, der auf das Gerät geschrieben ist. Dieser Wert kann in TwidoSoft im Dialogfeld **Steuerungsvorgänge** mit der Option **Echtzeituhr konfigurieren** konfiguriert werden.

Abarbeitungsblöcke

Einleitung

Abarbeitungsblöcke steuern Aktionen zu einem über Monat, Tag und Uhrzeit vordefinierten Zeitpunkt. Es dürfen maximal 16 Abarbeitungsblöcke programmiert werden. Diese Blöcke benötigen keinen Programmeintrag.

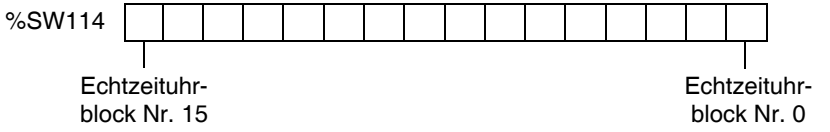
Hinweis: Überprüfen Sie das Systembit %S51 und das Systemwort %SW118, um sicherzustellen, dass die Echtzeituhr-Option (RTC) installiert ist. Siehe *Systembits (%S), S. 620*. Die Echtzeituhr-Option ist für die Verwendung von Abarbeitungsblöcken erforderlich.

Einstellungen Die folgende Tabelle führt die Parameter für einen Abarbeitungsblock auf:

Parameter	Format	Funktion/Bereich
Nummer des Abarbeitungsblocks	n	n = 0 bis 15
Konfiguriert	Kontrollkästchen	Aktivieren Sie dieses Kontrollkästchen, um die gewählte Nummer des Abarbeitungsblocks zu konfigurieren.
Ausgangsbit	%Qx.y.z	Die Ausgangszuweisung wird vom Abarbeitungsblock aktiviert: %Mi oder %Qj.k. Dieser Ausgang wird auf 1 gesetzt, wenn das aktuelle Datum und die aktuelle Uhrzeit zwischen der Einstellung für den Anfang des aktiven Zeitraums und der Einstellung für das Ende des aktiven Zeitraums liegen.
Startmonat	Januar bis Dezember	Der Monat, in dem der Abarbeitungsblock gestartet werden soll.
Endmonat	Januar bis Dezember	Der Monat, in dem der Abarbeitungsblock beendet werden soll.
Startdatum	1 - 31	Der Tag im Monat, an dem der Abarbeitungsblock gestartet werden soll.
Enddatum	1 - 31	Der Tag im Monat, an dem der Abarbeitungsblock beendet werden soll.
Startzeit	hh:mm	Uhrzeit, zu der der Abarbeitungsblock gestartet werden soll. Definiert durch die Stunde (0 bis 23), gefolgt von den Minuten (0 bis 59).
Stoppzeit	hh:mm	Uhrzeit, zu der der Abarbeitungsblock beendet werden soll. Definiert durch die Stunde (0 bis 23), gefolgt von den Minuten (0 bis 59).
Wochentag	Montag - Sonntag	Kontrollkästchen, die den Wochentag angeben, an dem der Abarbeitungsblock aktiviert werden soll.

Freigeben von Abarbeitungsblöcken

Die Bits von Systemwort %SW114 aktivieren (Bit auf 1) oder deaktivieren (Bit auf 0) den Betrieb der 16 einzelnen Abarbeitungsblöcke.
Zuweisung von Abarbeitungsblöcken in %SW114:



Standardmäßig (oder nach Kaltstart) sind alle Bits dieses Systemworts auf 1 gesetzt. Die Verwendung dieser Bits durch das Programm ist optional.

Ausgang von Abarbeitungsblöcken

Ist der gleiche Ausgang (%Mi oder %Qj.k) von verschiedenen Blöcken zugewiesen, wird diesem Objekt schließlich die ODER-Verknüpfung der Ergebnisse der einzelnen Blöcke zugewiesen (verschiedene "Betriebsbereiche" für den gleichen Ausgang sind möglich).

Im Beispiel zur Bezugnahme

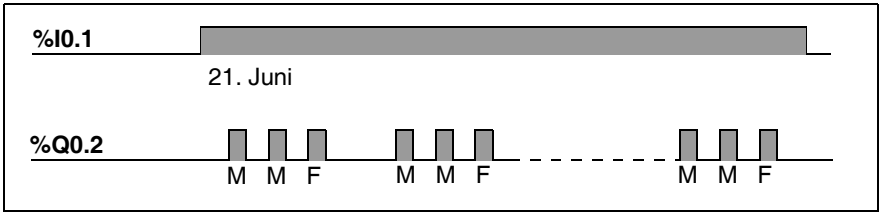
Die folgende Tabelle zeigt die Parameter für ein Programmbeispiel für eine Beregnung in den Sommermonaten:

Parameter	Wert	Beschreibung
Abarbeitungsblock	6	Abarbeitungsblock Nummer 6
Ausgangsbit	%Q0,2	Ausgang %Q0.2 aktivieren
Startmonat	Juni	Aktivität im Juni starten
Endmonat	September	Aktivität im September beenden
Startdatum	21	Aktivität am 21. Tag im Juni beginnen
Enddatum	21	Aktivität am 21. Tag im September beenden
Wochentag	Montag, Mittwoch, Freitag	Aktivität montags, mittwochs und freitags ausführen
Startzeit	21:00	Aktivität um 21:00 Uhr beginnen
Stoppzeit	22:00	Aktivität um 22:00 Uhr beenden

Mit Hilfe des folgenden Programms kann der Abarbeitungsblock über einen Schalter oder einen Feuchtigkeitssensor, der mit Eingang %I0.1 verdrahtet ist, deaktiviert werden.

```
LD    %I0.1
ST    %SW114:X6
```

Das folgende Chronogramm zeigt die Aktivierung von Ausgang %Q0.2.



Datierung durch das Programm

Das Datum und die Uhrzeit sind in den Systemwörtern %SW50 bis %SW53 verfügbar (siehe *Systemwörter (%SW), S. 633*). Im Steuerungsprogramm kann deshalb ein Uhrzeit- und Datumsstempel gesetzt werden, indem arithmetische Vergleiche zwischen dem aktuellen Datum und der aktuellen Uhrzeit und den direkten Werten oder Wörtern %MWi (oder %KWi), die Sollwerte enthalten können, durchgeführt werden.

Uhrzeit-/Datumsstempel

Einleitung

Die Systemworte %SW49 bis %SW53 enthalten die Datums- und Uhrzeitparameter im BCD-Format (siehe Abschnitt *Übersicht über den BCD-Code, S. 442*), das nützlich für die Anzeige auf einem Peripheriegerät oder für die Übertragung zu einem Peripheriegerät ist. Diese Systemworte können zum Speichern von Datums- und Uhrzeitparametern eines Ereignisses verwendet werden (siehe Abschnitt *Systemwörter (%SW), S. 633*).

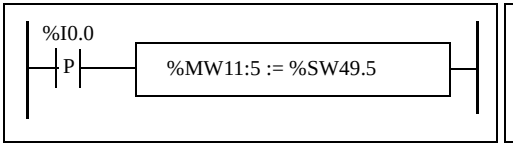
Hinweis: Die Datums- und Uhrzeitparameter können auch mit Hilfe des optionalen Bedienterminals geregelt werden (siehe Abschnitt *Echtzeituhr, S. 328*).

Datierung eines Ereignisses

Zum Datieren eines Ereignisses genügt es, Zuweisungsoperationen zu verwenden, um den Inhalt von Systemworten an interne Worte zu übertragen und diese internen Worte zu verarbeiten (z. B. die Übertragung an den Bedienterminal mit Hilfe der Anweisung EXCH).

Programmier-beispiel

Das folgende Beispiel zeigt, wie eine steigende Flanke am Eingang %I0.1 datiert wird.




```
LDR      %I0.0
[%MW11:5 := %SW49:5]
```

Wenn ein Ereignis erkannt wird, enthält die Worttabelle:

Codierung	Höherwertiges Byte	Niederwertiges Byte
%MW11		Wochentag[1]
%MW12	00	Sekunde
%MW13	Stunde	Minute
%MW14	Monat	Tag
%MW15	Jahrhundert	Jahr

Hinweis: (1) 1 = Montag, 2 = Dienstag, 3 = Mittwoch, 4 = Donnerstag, 5 = Freitag, 6 = Samstag, 7 = Sonntag.

Beispiel für eine Worttabelle

Beispiel für die Daten für Montag, den 19. April 2002, 13:40:30:

Wort	Wert (hex)	Bedeutung
%MW11	0001	Montag
%MW12	0030	30 Sekunden
%MW13	1340	13 Uhr und 40 Minuten
%MW14	0419	04 = 19. April
%MW15	2002	2002

Datum und Uhrzeit des letzten Stopps

Die Systemworte %SW54 bis %SW57 enthalten die Datums- und Uhrzeitparameter der letzten Unterbrechung der Applikation, und das Wort %SW58 enthält den Code zur Anzeige der Unterbrechungsursache im BCD-Format (siehe Abschnitt *Übersicht über den BCD-Code, S. 442*).

Einstellen von Datum und Uhrzeit

Einführung

Sie können die Einstellungen für Datum und Uhrzeit auf eine der folgenden Arten ändern:

- TwidoSoft
  Im Dialogfeld **Zeit einstellen**. Dieses Dialogfeld ist aus dem Dialogfeld **Steuerungsvorgänge** aufrufbar. Dieses wiederum wird im Menü **Steuerung** über den Befehl **Steuerung kontrollieren**.
- Systemwörter
  Mit den Systemwörtern %SW49 bis %SW53 oder dem Systemwort %SW59.

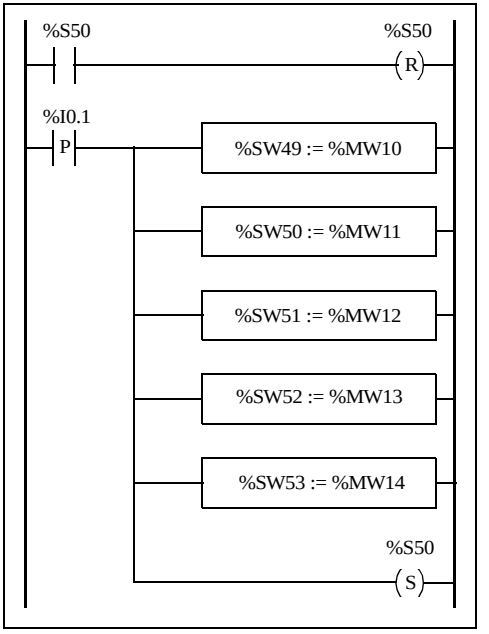
Die Einstellungen für Datum und Uhrzeit können nur aktualisiert werden, wenn die optionale Echtzeituhr-Cartridge (TWDXCPRTC) in der Steuerung installiert ist. Beachten Sie, dass die TWDLCA•40DRF-Serie der kompakten Steuerungen mit einer Echtzeituhr ausgestattet sind.

Mit %SW49 bis %SW53

Wenn Sie das Datum und die Uhrzeit mit den Systemwörtern %SW49 bis %SW53 einstellen möchten, muss das Bit %S50 auf 1 gesetzt sein. Dies bewirkt Folgendes:

- Die Aktualisierung der Wörter %SW49 bis %SW53 über die interne Uhr wird abgebrochen.
- Die Werte in den Wörtern %SW49 bis %SW53 werden an die interne Uhr übertragen.

Programmierbeispiel:




```
LD      %S50
R       %S50

LDR     %I0.1
[%SW49 := %MW10]
[%SW50 := %MW11]
[%SW51 := %MW12]
[%SW52 := %MW13]
[%SW53 := %MW14]
S       %S50
```

Die Wörter %MW10 bis %MW14 enthalten das neue Datum und die neue Uhrzeit im BCD-Format (siehe *Übersicht über den BCD-Code, S. 442*) und entsprechen der Codierung der Wörter %SW49 bis %SW53.
Die Worttabelle muss das neue Datum und die neue Uhrzeit enthalten:

Codierung	Höherwertiges Byte	Niederwertiges Byte
%MW10		Wochentag[1]
%MW11		Sekunde
%MW12	Stunde	Minute
%MW13	Monat	Tag
%MW14	Jahrhundert	Jahr

Hinweis: (1) 1 = Montag, 2 = Dienstag, 3 = Mittwoch, 4 = Donnerstag, 5 = Freitag, 6 = Samstag, 7 = Sonntag.

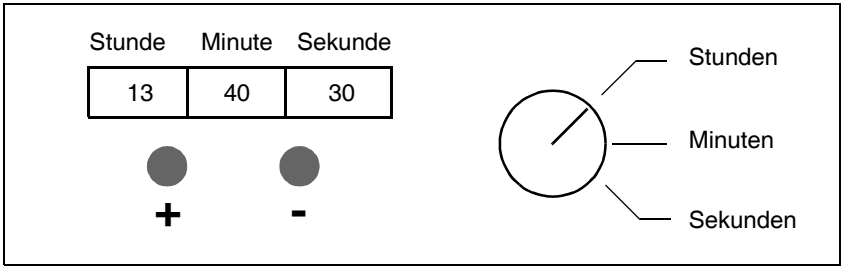
Beispieldaten für Montag, 19. April 2002:

Wort	Wert (hex)	Bedeutung
%MW10	0001	Montag
%MW11	0030	30 Sekunden
%MW12	1340	13 Uhr und 40 Minuten
%MW13	0419	04 = 19. April
%MW14	2002	2002

Mit %SW59

Eine weitere Vorgehensweise zum Aktualisieren von Datum und Uhrzeit ist die Verwendung von Systembit %S59 und des Systemworts für die Datumseinstellung %SW59.
Wird Bit %S59 auf 1 gesetzt, wird die Anpassung des aktuellen Datums und der aktuellen Uhrzeit durch das Wort %SW59 ermöglicht (siehe *Systemwörter (%SW), S. 633*). %SW59 inkrementiert oder dekrementiert alle Komponenten des Datums und der Uhrzeit bei einer steigenden Flanke.

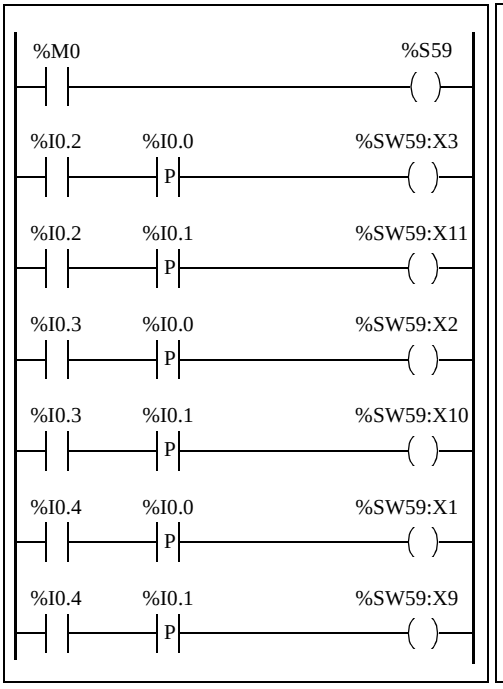
Applikationsbeispiel Die folgende Frontseite wird erstellt, um die Stunde, Minute und Sekunde der internen Uhr zu ändern.



Beschreibung der Befehle:

- Der Schalter für Stunden/Minuten/Sekunden ändert die Zeitanzeige über die jeweiligen Eingänge %I0.2, %I0.3 und %I0.4.
- Der Drucktaster "+" inkrementiert die eingestellte Zeitanzeige über den Eingang %I0.0.
- Der Drucktaster "-" dekrementiert die eingestellte Zeitanzeige über den Eingang %I0.1.

Das folgende Programm liest die Eingänge auf der Frontseite und stellt die interne Uhr ein.




```
LD    %M0
ST    %S59
LD    %I0.2          (Stunde)
ANDR  %I0.0
ST    %SW59:X3
LD    %I0.2
ANDR  %I0.1
ST    %SW59:X11
LD    %I0.3          (Minute)
ANDR  %I0.0
ST    %SW59:X2
LD    %I0.3
ANDR  %I0.1
ST    %SW59:X10
LD    %I0.4          (Sekunde)
ANDR  %I0.0
ST    %SW59:X1
LD    %I0.4
ANDR  %I0.1
ST    %SW59:X9
```

17.3 Twido PID-Kurzanleitung

Auf einen Blick

Überblick

Dieser Abschnitt enthält Informationen über die ersten Schritte mit den an Twido-Steuerungen verfügbaren PID-Steuerungs- und Auto-Tuning-Funktionen.

Inhalt dieses Abschnitts

Dieser Abschnitt enthält die folgenden Themen:

Thema	Seite
Zweck des Dokuments	510
Schritt 1 - Konfigurierung der analogen Kanäle für die Regelung	512
Schritt 2 - Vorbedingungen für die PID-Konfigurierung	514
Schritt 3 - Konfiguration des PID	516
Schritt 4 - Initialisierung der Inbetriebnahme der Regelung	523
Schritt 5 - Inbetriebnahme der Regelung AT + PID	527
Schritt 6 - Debugging der Einstellungen	531

Zweck des Dokuments

Einführung

Ziel des Schnelleinstiegs ist es, Sie mit Hilfe eines Anwendungsbeispiels durch die Einzelschritte der Konfigurierung und fachgerechten Inbetriebnahme der PID-Regelungsfunktionen der Twido-Steuerung zu führen.

Hinweis: Die Inbetriebnahme der Twido-Funktion PID erfordert keine spezifischen Kenntnisse, wohl aber Genauigkeit, um ein optimales Ergebnis unter möglichst geringem Zeitaufwand zu ermöglichen.

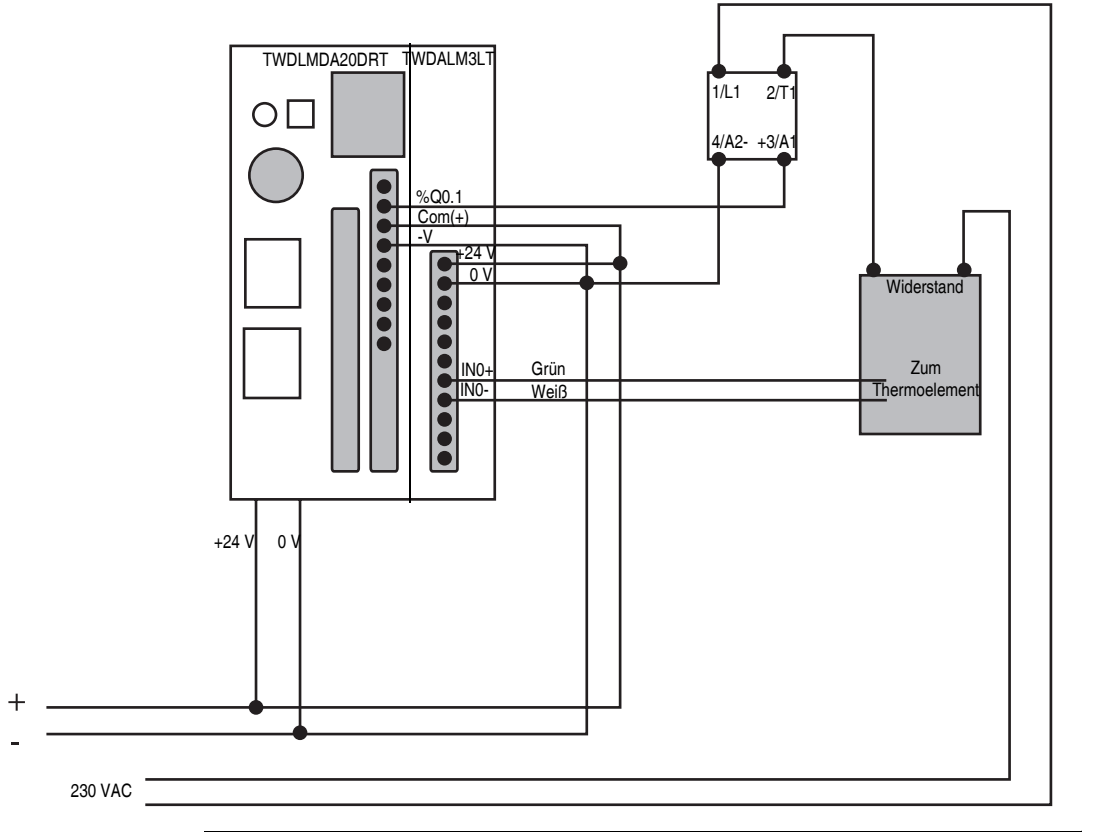
Das Dokument enthält:

Folgende Schritte werden im Einzelnen behandelt:

Schritt	Beschreibung
1	Konfigurierung der Analogkanäle für die Regelung
2	Vorbedingungen für die PID-Konfigurierung
3	PID-Konfigurierung
4	Initialisierung der Inbetriebnahme der Regelung
5	Inbetriebnahme der AT + PID-Regelung
6	Fehlerbeseitigung und Anpassung

Hinweis zu dem Beispiel im Schnelleinstieg

Wir arbeiten in diesem Beispiel mit einem Thermoelement des Typs K, 0-200°. Wir arbeiten mit einer Steuerung mit statischem Relais. Der Ausgang ist folglich ein Ausgang des SPS-Grundgeräts, der als PWM direkt über den PID-Regler gesteuert wird (siehe*Schritt 3 - Konfiguration des PID, S. 516*
Die obenstehende Abbildung zeigt den Versuchaufbau aus dem Anwendungsbeispiel:



Schritt 1 - Konfigurierung der analogen Kanäle für die Regelung

Einführung

Im Allgemeinen misst ein PID-Regler den zu regelnden Wert anhand eines analogen ("Messwert" genannten) Rückmeldesignals.
Das kann je nach Anwendung ein Pegel, eine Temperatur, ein Abstand o.ä. sein.

Beispiel für ein analoges Messsignal

Nehmen wir als Beispiel eine Temperaturmessung.
Der verwendete Geber sendet einen analogen, vom Messwert abhängigen Wert an die SPS zurück. Bei Temperaturmessungen mit Gebern wie PT100 oder Thermoelementen wird das gemessene Signal bei steigender Temperatur stärker.

Hinzufügen einer Analogkarte (Erweiterungsmodul):

Im Offline-Modus fügen Sie nach Auswahl des SPS-Grundgeräts eine Analogkarte zur Erweiterung des Grundgeräts hinzu. Die Markierung der Kanäle ist von der Kartenanordnung in der Konfiguration abhängig.

Konfigurieren der Analogkanäle

Die folgende Tabelle beschreibt, wie die Analogkanäle des Erweiterungsmoduls konfiguriert werden:

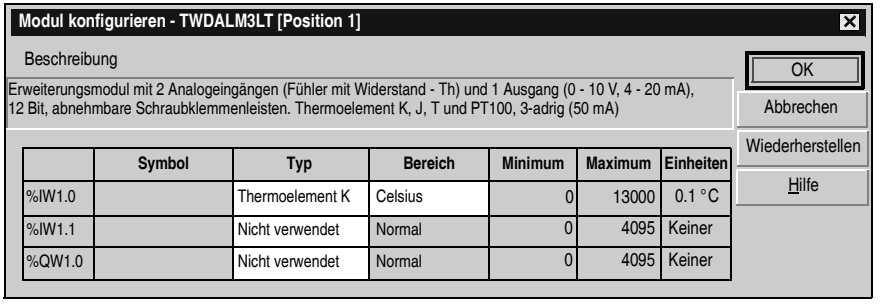
Schritt	Aktion
1	Rechter Mausklick auf das Item **Erweiterungsbus** → **Modul hinzufügen**.
2	Wählen Sie in der Liste die gewünschte Karte. Zum Beispiel TWDALM3LT für eine Temperaturmessung mit PT100 oder Thermoelement.
3	Klicken Sie auf **Hinzufügen** und auf **Beenden**, wenn nur dieses eine Erweiterungsmodul konfiguriert wird.
4	Rechter Mausklick auf die hinzugefügte Karte und auf das Menü **Konfigurieren**.
5	Wählen Sie in der Spalte **Typ** den Eingabetyp je nach verwendetem Geber (z.B. Thermoelement K, falls zutreffend).
6	Wählen Sie in der Spalte **Grösse** die Maßeinheit des Gebers. Bei Temperaturgebern wählen Sie am besten **Celsius**, um einen direkten Bezug zwischen den von der Analogkarte zurückgesendeten Punkten und der tatsächlichen Messung herzustellen.
7	Suchen Sie das Eingangssymbol auf der so konfigurierten Analogkarte. Es wird zum Ausfüllen der PID-Felder (im vorliegenden Beispiel %IW1.0) benötigt.
8	Gehen Sie bei einem zur Steuerung eines Steuersystems bestimmten Analogausgang nach dem gleichen Verfahren vor.

Konfigurations-beispiel für einen Analogkanal

Je nach angewandtem Messverfahren gibt es wie unten gezeigt mehrere Konfigurationsmöglichkeiten:

- Für das hier beschriebene Anwendungsbeispiel haben wir ein **Thermoelement vom Typ K**, 0-200° gewählt. Der abgelesene Messwert ist direkt auswertbar (2000 Pkte = 200°, da der Faktor für die Maßeinheit 0,1° beträgt).
- Für andere Messungen können Sie **0-10 V** oder **4-20 mA** in der Spalte **Typ** wählen und **Grösse** in der Spalte **Individuell**. Stimmen Sie anschließend die Werteskala ab (geben Sie **0** in die Spalte **Minimum** und**10000** in die Spalte **Maximum**ein), um direkt auswertbare Werte zu erhalten (10 V = 10000 Punkte).

Das unten aufgeführte Beispiel zeigt eine Analogkanalkonfiguration mit einem Thermoelement des Typs K:



	Symbol	Typ	Bereich	Minimum	Maximum	Einheiten
%IW1.0		Thermoelement K	Celsius	0	13000	0.1 °C
%IW1.1		Nicht verwendet	Normal	0	4095	Keiner
%QW1.0		Nicht verwendet	Normal	0	4095	Keiner

Schritt 2 - Vorbedingungen für die PID-Konfigurierung

Einführung

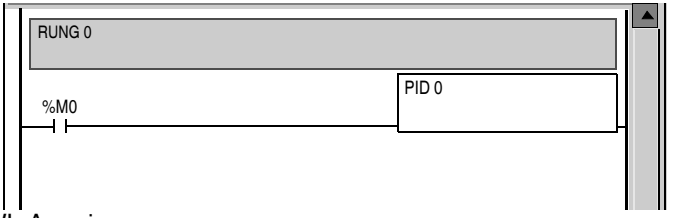
Überprüfen Sie vor der PID-Konfigurierung, dass folgende Schritte durchgeführt wurden:

Phase	Beschreibung
1	Freigabe des PID im Programm.
2	Konfigurieren der Abtastzeit

Freigabe des PID im Programm.

Der PID-Regler muss im Programm durch eine Anweisung freigegeben werden. Es handelt sich dabei um eine permanente bzw. durch einen Eingang oder ein externes Bit gegebene Anweisung. Im folgenden Beispiel wird das PID durch die Anweisung %M0 freigegeben

- Als KOP:



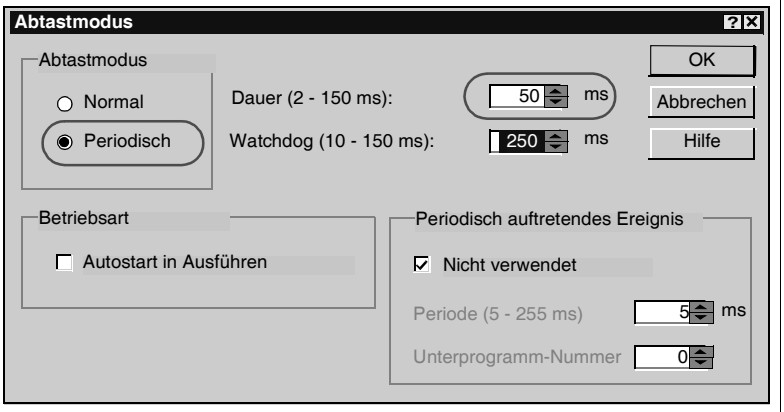
- Als AWL-Anweisung:

```
----
   0 LD     %M0
   1 [ PID 0 ]
```

Hinweis: Auf richtige Schreibung achten!
Zwischen den Buchstaben "PID" und der PID-Zahl (wie `PID<Leerzeichen>0`) muss ein Leerzeichen stehen.

Konfigurieren der Abtastzeit

Es empfiehlt sich, beim Einsatz von PID-Reglern den Abtastmodus des Steuerungszyklus auf periodisch einzustellen. Die nachfolgende Tabelle zeigt, wie der Abtastmodus konfiguriert wird.

Schritt	Aktion
1	Wählen Sie in der Menüleiste TwidoSoft **Programm** → **Bearbeiten Sie den Abtastmodus**.
2	Kreuzen Sie das Kästchen **Periodisch**an.
3	Stellen Sie die Zyklusdauer wie unten im Fenster gezeigt ein:  **Hinweis:** Die Zyklusdauer muss dem Programmumfang und den gewünschten Leistungen angepasst werden. (50 ms sind ein sinnvoller Kompromiss).

Schritt 3 - Konfiguration des PID

Einleitung

Für dieses Beispiel haben wir uns entschieden, die größtmögliche Anzahl von Funktionen des PID-Reglers für Twido zu implementieren. Bestimmte Optionen sind nicht unverzichtbar und können vereinfacht werden.

Funktion Auto-Tuning (AT)

Der PID-Regler verfügt über eine Auto-Tuning-Funktion, die es ermöglicht, die Einstellung der Regelschleife zu vereinfachen (diese Funktion wird im weiteren Verlauf des Dokuments als AT bezeichnet).

Betriebsarten

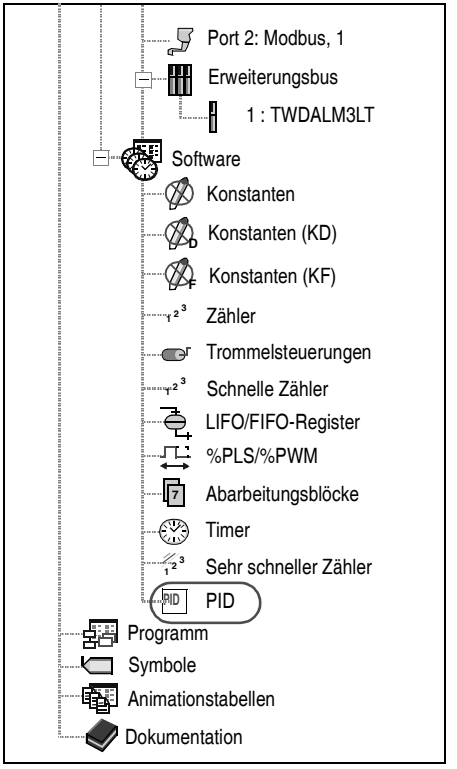
Der PID-Regler der Twido-Steuerung bietet vier verschiedenen Betriebsarten, die im Dialogfeld **PID** auf der Registerkarte **Allgemein** konfiguriert werden können:

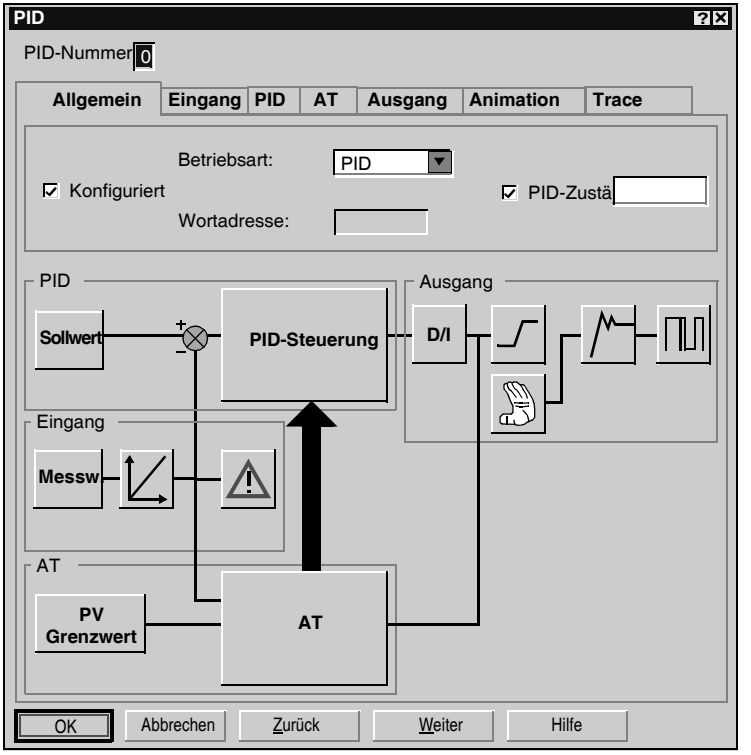
- **PID** = einfacher PID-Regler.
- **AT + PID** = Die Funktion Auto-Tuning wird beim Start des PID-Reglers aktiviert und ermöglicht die automatische Abfrage der Verstärkungswerte **Kp, Ti und Td** (Registerkarte **PID**) und des Aktionstyps des PID-Reglers (Registerkarte **Ausgang**). Nach Abschluss des Auto-Tuning geht der Regler in den PID-Modus mit dem eingestellten Sollwert und mit den durch AT eingestellten Parametern über .
- **AT** = Die Funktion Auto-Tuning wird beim Start des PID-Reglers aktiviert und ermöglicht die automatische Abfrage der Verstärkungswerte **Kp, Ti und Td** (Registerkarte **PID**) und des Aktionstyps des PID-Reglers (Registerkarte **Ausgang**). Nach Abschluss der Sequenz hält der PID-Regler an und wartet. Die Verstärkungswerte **Kp, Ti und Td** (Registerkarte **PID**) und der Aktionstyp des PID (Registerkarte **Ausgang**) werden angegeben.
- **Wortadresse** = Die Auswahl der Betriebsart des PID kann programmgesteuert erfolgen, indem der gewünschte Wert der mit dieser Option verbundenen Wortadresse zugewiesenen wird:
  - %MWxx=1: Der Regler wird im einfachen PID-Modus betrieben.
  - %MWxx=2: Der Regler wird im Modus AT + PID betrieben.
  - %MWxx=3: Der Regler wird nur im Modus AT betrieben.

  Diese Art der Konfiguration über die Wortadresse bietet dem Benutzer die Möglichkeit, per Anwendungsprogramm die Betriebsart des PID-Reglers zu steuern und ermöglicht so die Anpassung an die individuellen Bedürfnisse.

Start des PID-Dialogfelds

Die folgende Tabelle zeigt das Dialogfeld des PID sowie die Vorgehensweise für den Zugriff auf verschiedene Registerkarten zur Konfiguration der Parameter des PID.

Schritt	Aktion
1	Doppelklicken Sie auf die Option **PID** im Konfigurationsnavigator im linken Teil des TwidoSoft-Fensters (siehe Abbildung unten). 

Schritt	Aktion
2	Das PID-Dialogfeld wird im Vordergrund angezeigt. Es ermöglicht die Anzeige der verschiedenen Parameter des Reglers (siehe Abbildung unten). In diesem Bildschirm werden im Offline-Modus mehrere Registerkarten angezeigt: **Allgemein, Eingang, PID, AT, Ausgang:**  **Wichtig:** Die Registerkarten sind in der Reihenfolge auszufüllen, in der Sie im Dialogfeld des PID angezeigt werden: zunächst Allgemein, Eingang, PID, AT und dann Ausgang. **Hinweis:** Im Online-Modus werden in diesem Bildschirm zwei weitere Registerkarten angezeigt (**Animation** und **Trace**), die die Diagnose bzw. die Anzeige der Funktionsweise des Reglers ermöglichen.

Dynamische Änderung der Parameter

Für eine dynamische Änderung der Parameter des PID (während des Betriebs und im Online-Modus) empfiehlt es sich, Speicheradressen in den zugehörigen Feldern anzugeben, um so den Übergang in den Offline-Modus für jegliche Werteänderung im laufenden Betrieb zu vermeiden.

Parametrierung der Registerkarte "Allgemein"

Die folgende Tabelle zeigt die Vorgehensweise für die Parametrierung der Registerkarte **Allgemein** des Dialogfelds des PID.

Schritt	Aktion
1	Aktivieren Sie auf der Registerkarte **Allgemein** das Kontrollkästchen **Konfiguriert**, um den PID zu aktivieren und um die folgenden Registerkarten einstellen zu können.
2	Wählen Sie in der Dropdown-Liste **Betriebsart** die gewünschte Betriebsart aus (Siehe *Betriebsarten, S. 516*). **Im Beispiel:** Wir wählen den Modus "Speicheradresse" aus und geben in das zugehörige Feld das Wort %MW17 ein. Die Betriebsart des PID ist folglich mit dem in %MW17 vorhandenen Wert verbunden.

Parametrierung der Registerkarte "Eingang"

Die folgende Tabelle zeigt die Vorgehensweise für die Parametrierung der Registerkarte **Eingang** des Dialogfelds des PID.

Schritt	Aktion
1	Geben Sie auf der Registerkarte **Eingang** im entsprechenden Feld den analogen Kanal an, der als Messwert dient. **Im Beispiel:** Wir wählen **%IW1.0** aus, da er als Temperaturmesswert dient.
2	Aktivieren Sie erforderlichenfalls die Alarme an den unteren und oberen Schwellwerten, indem Sie das Kontrollkästchen aktivieren und die entsprechenden Felder ausfüllen. **Hinweis:** Die eingegebenen Werte können unveränderliche Werte (eingegeben in die entsprechenden Felder) oder veränderbare Werte sein (indem in die entsprechenden Felder die Speicheradressen eingegeben werden: %MWxx).

Parametrierung der Registerkarte "PID"

Die folgende Tabelle zeigt die Vorgehensweise für die Parametrierung der Registerkarte **PID** des Dialogfelds des PID.

Schritt	Aktion
1	Geben Sie auf der Registerkarte **PID** den Wert ein, der zur Festlegung des Sollwerts des Reglers dient. Im Allgemeinen ist dieser Wert eine Speicheradresse oder ein von einem analogen Eingang stammender Sollwert. **Im Beispiel:** Wir geben %MW0 ein, das als Sollwertwort dient.
2	Konfigurieren Sie die Parameter **Kp, Ti und Td**. **Wichtig:** Wenn der Modus **AT** oder **AT+PID** ausgewählt ist, müssen in die Felder Kp, Ti und Td unbedingt **Speicheradressen** eingetragen werden, die es der Funktion Auto-Tuning ermöglichen, automatisch die gefundenen Werte anzugeben. **Im Beispiel:** Wir geben %MW10 für Kp, %MW11 für Ti und %MW12 für Td ein. **Hinweis:** Es ist grundsätzlich ziemlich schwer, die optimalen Einstellwerte von Kp, Ti und Td für eine Anwendung anzugeben, die noch nicht erstellt worden ist. Daher empfehlen wir dringend, in diese Felder Adressen von Speicherwörtern einzutragen, die eine Festlegung dieser Werte im Online-Modus ermöglichen, um so bei jeglicher Werteänderung im laufenden Betrieb den Übergang in den Offline-Modus zu vermeiden.
3	Stellen Sie die **Abtastzeit** des PID ein. Dieser Wert wird vom Regler verwendet, um die Erfassung der Messwerte und die Aktualisierung der Ausgänge durchzuführen. **Im Beispiel:** Wir stellen die Abtastzeit des PID auf 100 ein, was 1 s entspricht. Da das gesteuerte System über eine Zeitkonstante von mehreren Minuten verfügt, scheint die Größenordnung der Abtastzeit korrekt zu sein. **Wichtig:** Es wird geraten, die Abtastzeit auf ein Vielfaches der Abtastzeit der Steuerung und konsistent zum geregelten System einzustellen.

Parametrierung der Registerkarte AT

Die folgende Tabelle zeigt die Vorgehensweise für die Parametrierung der Registerkarte **AT** des Dialogfelds des PID.

Schritt	Aktion
1	Aktivieren Sie auf der Registerkarte **AT** das Kontrollkästchen **Freigeben**, wenn Sie die AT-Funktion verwenden möchten.
2	Geben Sie den Wert **Grenzwert der Messung** an. Dies ist der Grenzwert, der von der Messung während des AT-Verfahrens nicht überschritten werden darf.
3	Geben Sie den Wert **Ausgangssollwert** an. Dies ist der am Ausgang des Reglers gesendete Wert zur Generierung des AT-Verfahrens.
Hinweis	**Ausführliche Informationen über die Einstellung dieser Werte finden Sie im Abschnitt *Registerkarte AT des PID, S. 551*.**
Ratschlag	**Wir empfehlen dringend, in diese Felder Adressen von Speicherwörtern einzutragen, die eine Festlegung dieser Werte im Online-Modus ermöglichen, um so bei jeglicher Werteänderung im laufenden Betrieb den Übergang in den Offline-Modus zu vermeiden.**

Parametrierung der Registerkarte "Ausgang"

Die folgende Tabelle zeigt die Vorgehensweise für die Parametrierung der Registerkarte **Ausgang** des Dialogfelds des PID.

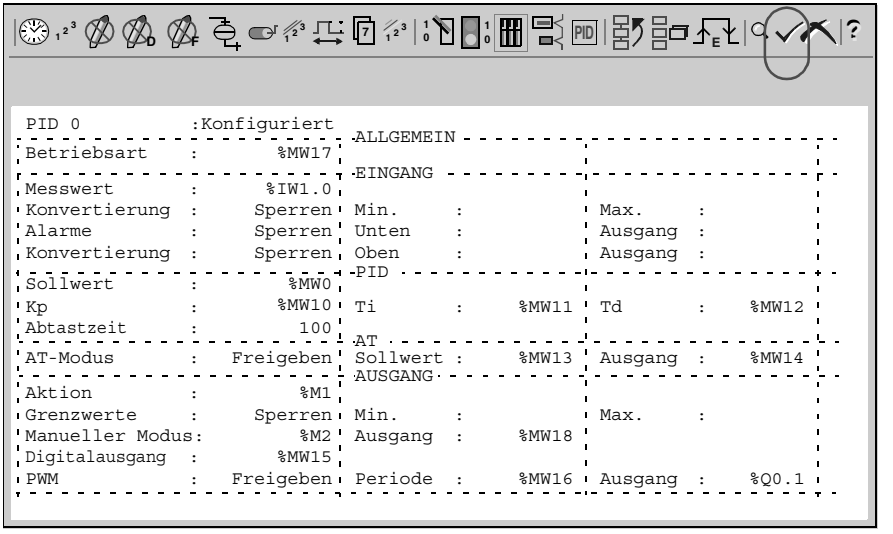
> **⚠ WARNUNG**
>
> **GEFAHR EINER SYSTEMÜBERLASTUNG**
>
> Wir weisen darauf hin, dass der manuelle Modus direkt auf den Ausgang des Reglers wirkt. Folglich wirkt das Sensen eines manuellen Sollwerts (Feld **Ausgang**) direkt auf das bei offener Schleife geregelte System. Es empfiehlt sich daher, diese Betriebsart mit äußerster Vorsicht zu nutzen.
>
> **Die Nichtbeachtung dieser Anweisung kann den Tod, körperverletzung Oder materialschäden Zur folge Haben!**

Schritt	Aktion
1	Geben Sie auf der Registerkarte **Ausgang** die Auswahl aus der Dropdown-Liste **Aktion** an. Diese Auswahl ist vom geregelten System abhängig: ● Direkte Aktion: Der Ausgang des Reglers nimmt ab, wenn der Wert der Abweichung (Sollwert - Messwert) zunimmt (Kaltregler). ● Inverse Aktion: Der Ausgang des Reglers nimmt zu, wenn der Wert der Abweichung (Sollwert - Messwert) abnimmt (Warmregler). **Wichtig:**Bei Verwendung der Funktion AT wird die Auswahl dieser Liste automatisch auf **Bitadresse** gesetzt. Die Betriebsart wird durch die Funktion AT bestimmt und in diesem Fall in dem diesem Feld zugewiesenen Bit angegeben.
2	Geben Sie erforderlichenfalls in den Feldern **Alarme** die Grenzwerte des Ausgangswerts des Reglers an. Diese Funktion kann für bestimmte Anwendungen erforderlich sein, um die Prozessalarme bei einer Überschreitung der Schwellwerte zu verwalten.
3	Legen Sie die Betriebsart des **manuellen Modus** fest. Die Dropdown-Liste bietet mehrere Optionen: ● **Sperren** = kein manueller Modus. ● **Freigeben** = der Regler funktioniert nur im manuellen Modus. ● **Bitadresse** = der Wert des Bits ermöglicht die Änderung der Funktionsweise des manuellen Modus (Bit auf 0 = automatischer Modus, Bit auf 1 = manueller Modus). **Im Beispiel:** Wir wählen %M2, um die Wahl zu aktivieren, und %MW18 für die Einstellung des manuellen Sollwerts.
4	Legen Sie das Wort **Digitalausgang** fest. Dieses Wort wird vom Regler verwendet, um den Sollwert des Reglers zu senden. Er wird direkt auf einem analogen Ausgangskanal (%QW..) oder in einem Speicherwort (%MWxx) für eine weitere Verarbeitung gesendet. **Wichtig:** Geben Sie bei Verwendung der Funktion PWM in diesem Feld eine Speicheradresse (%MWxx) ein.

Schritt	Aktion
5	Legen Sie den **Ausgang PWM** fest, wenn das System dies erfordert: **1.** Aktivieren Sie das Kontrollkästchen **Freigeben**, wenn Sie das System durch einen PWM-Aktor steuern möchten. **2.** Geben Sie die **PWM-Befehlsperiode** im entsprechenden Feld an. **3.** Geben Sie den für die Steuerung des PWM-Aktors verwendeten **Ausgang** an. Es wird geraten, die Transistorausgänge der Grundgeräte SPS für diese Funktion zu verwenden (z.B. %Q0.0 oder %Q0.1 für das Grundgerät SPS TWDLMDA20DRT).
6	Bestätigen Sie die Konfiguration des Reglers durch Anklicken der Schaltfläche **OK** unten links im Bildschirm.
7	Wenn mehrere PID-Regler konfiguriert werden müssen, verwenden Sie die Schaltfläche **Weiter**, um die Nummer des zu regelnden PID zu inkrementieren.

PID-Konfigurationseditor

Nach der Bestätigung der Parametrierung des PID müssen Sie den PID-Konfigurationseditor bestätigen, in dem alle Parameter jedes konfigurierten PID zusammengefasst sind.
Um den Bildschirm des Konfigurationseditors zu bestätigen, klicken Sie auf das Symbol **Akzeptieren** in der Shortcut-Leiste (siehe Abbildung unten).



Schritt 4 - Initialisierung der Inbetriebnahme der Regelung

Vorbedingungen für die Inbetriebnahme

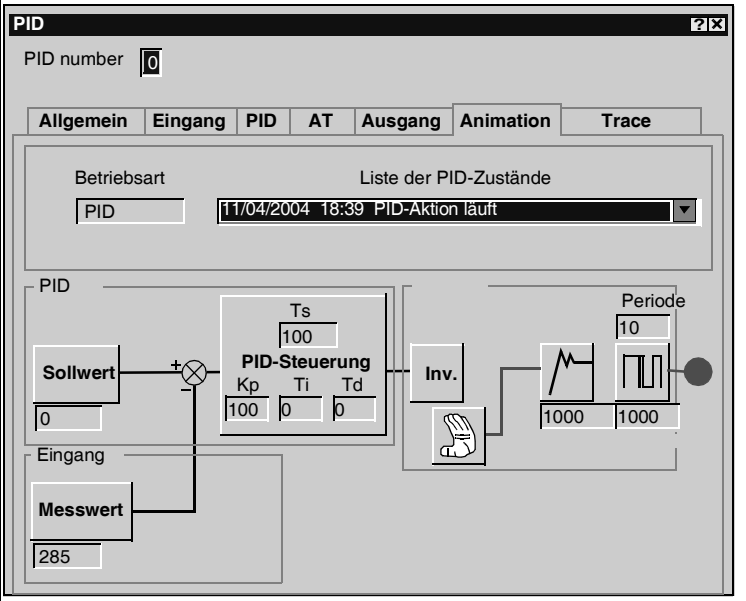
Führen Sie vor der Inbetriebnahme folgende Schritte durch:

Schritt	Aktion
1	Schließen Sie den PC an die Steuerung an und übertragen Sie die Anwendung.
2	Schalten Sie die Steuerung in den RUN-Modus.

Hinweis: Überprüfen Sie vorab, ob die Betriebsbedingungen des Geräts das Umschalten auf RUN für die restliche Anwendung ermöglichen.

Vorgehensweise Für die Initialisierung der Inbetriebnahme der Regelung sind folgende Schritte durchzuführen:

Schritt	Aktion
1	Erstellen Sie eine Animationstabelle mit den Hauptdiagnoseobjekten: **Im Beispiel** ● %MW0: Regler-Sollwert, ● %IW1.0: Messung, ● %M0: Freigabe des Reglers, ● %M1: Regleraktionsart (durch die AT-Funktion bestimmt), ● %M2: Wahl zwischen Automatik- oder Handbetrieb, ● %MW10 bis %MW12: PID-Reglerkoeffizienten, ● %MW13: Messgrenzwert, der im AT-Modus nicht überschritten werden darf, ● %MW14: Ausgangssollwert des Reglers im AT-Modus, ● %MW15: Digitalausgang des PID-Reglers (vom Regler angegeben), ● %MW16: Einstellung der PWM-Dauer, ● %MW17: Wahl der Betriebsart des PID-Reglers, ● %MW18: von Hand eingegebener Sollwert mit Bitwahl %M2.
2	Überprüfen Sie, ob der im Feld %IW1.0 gemessene Wert stimmt. **Im Beispiel** **1.** messen wir 248 Pkte bei stabilem System im Kaltzustand. **2.** Der Wert scheint zu stimmen, da der Multiplikationsfaktor zwischen Temperatur und abgelesenem Wert 10 beträgt. Man kann den Wert zur Überprüfung sicherheitshalber auch von außen beeinflussen (dazu die Temperatur an der Außenseite des Fühlers erhöhen und den Anstieg des Messwerts beobachten). **Hinweis:** Dieser Test ist wichtig, weil die Funktionsfähigkeit des Reglers in erster Linie von der Qualität und Präzision der Messung abhängt. **3.** Wenn Sie Zweifel an der Genauigkeit der Messung haben, schalten Sie die Steuerung auf STOP und kontrollieren Sie die Kabelanschlüsse an den Eingängen der Analogkarte (Voltmeter oder Amperemeter an den Ausgängen 0-10V / 4-20mA, Ohmmeter an den PT100 (100 Ohm bei 20°) bzw. Thermoelement (zwischen 10 und 100 Ohm): ● Ziehen Sie den Fühler vorher aus den Klemmen der Analogkarte. ● Überprüfen Sie, ob die Kabel vertauscht wurden (Farbmarkierung der an den Eingängen angeschlossenen Drähte, Kompensationskabel für PT100). **Achtung:** Die Eingangskanäle IN0 und IN1 haben das gleiche Potenzial an den (-)-Klemmen. ● Überprüfen Sie, ob die ersten beiden Klemmen der Analogkarte mit 24VCC gespeist werden. ● Überprüfen Sie, ob die Fühler der 4-20 mA-Eingänge mit Spannung versorgt werden. Die Analogeingangskarten des Twido sind keine Stromquelle.

Schritt	Aktion
3	Zum Starten des Reglers müssen Sie zunächst den PID-Regler auf **Hand**betrieb stellen, um die Grenzwerte für die AT-Funktion zur Kenntnis zu nehmen. So stellen Sie den Regler auf Handbetrieb: **1.** Schalten Sie die Steuerung in den RUN-Modus. **2.** Geben Sie für die Speicheradressen folgende Werte in die Animationstabelle ein: ● %M2: Wahl des Handbetriebs = 1 (M2=1 => Handbetrieb, M2=0 => Automatikbetrieb), ● %MW16: Einstellung der PWM-Dauer = 10, ● %MW17: Wahl der Betriebsart des PID-Reglers = 1 (einfaches PID), ● %MW18: von Hand eingegebener Sollwert mit Bitwahl %M2 = 1000. Die Wahl des Sollwertes kann mehrmals erfolgen, vorausgesetzt, Sie warten, bis das System jeweils wieder im Ausgangszustand ist. **Im Beispiel** haben wir den Wert 1000 gewählt, was dem durchschnittlichen Temperaturanstieg entspricht (Hinweis: 2000 Pkte = 200°). Im Kaltzustand startet das System nach dieser Einteilung bei 250 Pkten.
4	Überprüfen Sie, ob die Steuerung sich im RUN-Modus befindet. (%M0: Freigabe des Reglers = 1, in der Animationstabelle angeben.)
5	Doppelklicken Sie im Konfigurationsnavigator auf das Item **PID**.
6	Aktivieren Sie die Registerkarte **Animation** für die gewünschte PID-Nummer und überprüfen Sie, ob die Animation dem Bildschirm unten entspricht:  **Hinweis:** Die Fenster des PID-Reglers werden nur aufgefrischt, wenn der Regler freigegeben ist (und API auf RUN steht).

Schritt	Aktion
7	Aktivieren Sie die Registerkarte **Trace** für die gewünschte PID-Nummer und: **1.** stellen Sie die Drop-Down-Liste des Zeitablaufs auf **15 min**, um den Fortschritt des Messsignals anzuzeigen. **2.** Überprüfen Sie, dass der Messwert sich in einem für das System annehmbaren Rahmen bewegt. Benutzen Sie für diese Kontrolle die Registerkarte Trace, die den Messwertanstieg anzeigt. Sobald der Messwert sich stabilisiert hat, lesen Sie den entsprechenden Wert auf der Messkurve ab (im Beispiel entsprechen 350 Pkte 35°, d.h. der Anstieg bezogen auf den Ausgangszustand beträgt 10°).
8	stellen Sie die Drop-Down-Liste des Zeitablaufs auf 15 min, um den Fortschritt des Messsignals anzuzeigen. Überprüfen Sie, dass der Messwert sich in einem für das System annehmbaren Rahmen bewegt. Benutzen Sie für diese Kontrolle die Registerkarte "Trace", die den Messwertanstieg anzeigt. Sobald der Messwert sich stabilisiert hat, lesen Sie den entsprechenden Wert auf der Messkurve ab (im Beispiel entsprechen 350 Pkte 35°, d.h. der Anstieg bezogen auf den Ausgangszustand beträgt 10°).

9	Falls eines der Stellglieder nicht angesteuert wird, muss der Ausgangskreis kontrolliert werden:
	● Bei einem Analogausgang messen Sie die Spannung bzw. Stromstärke am Ausgang der Analogkarte.
	● Im Falle eines PWM-Ausgangs überprüfen Sie:
	  ● ob die LED des betreffenden Ausgangs leuchtet (im Beispiel %Q0.1),
	  ● die Verkabelung der Stromversorgung und den 0V-Kreis der Ausgänge am Grundgerät TWDLMDA20DRT,
	  ● die Leistungsversorgung des Stellglieds.

10	Schließen Sie den PID-Anzeigebildschirm und beenden Sie den Handbetrieb, indem Sie folgende Werte in die Animationstabelle eingeben: ● %M0: Freigabe des Reglers = 0 (Regler aus) ● %M2: Wahl zwischen Automatik- oder Handbetrieb = 0 (Handbetrieb aus) ● %MW17: Wahl der Betriebsart des PID-Reglers = 0 ● %MW18: von Hand eingegebener Sollwert mit Bitwahl %M2 = 0

Schritt 5 - Inbetriebnahme der Regelung AT + PID

Einleitung

Dieser Abschnitt beschreibt die Parametrierung des Reglers für die Inbetriebnahme im Modus AT+PID. In dieser Betriebsart führt der Regler automatisch die Regelung des Reglers an den Koeffizienten Kp, Ti, und Td aus.

Hinweis: Es empfiehlt sich, während dieser Phase das System nicht durch äußere Veränderungen, die die endgültigen Einstellungen beeinflussen könnten, zu stören. Außerdem empfiehlt es sich, vor dem Start der AT-Sequenz sicher zu stellen, dass sich das System in einem stabilen Zustand befindet.

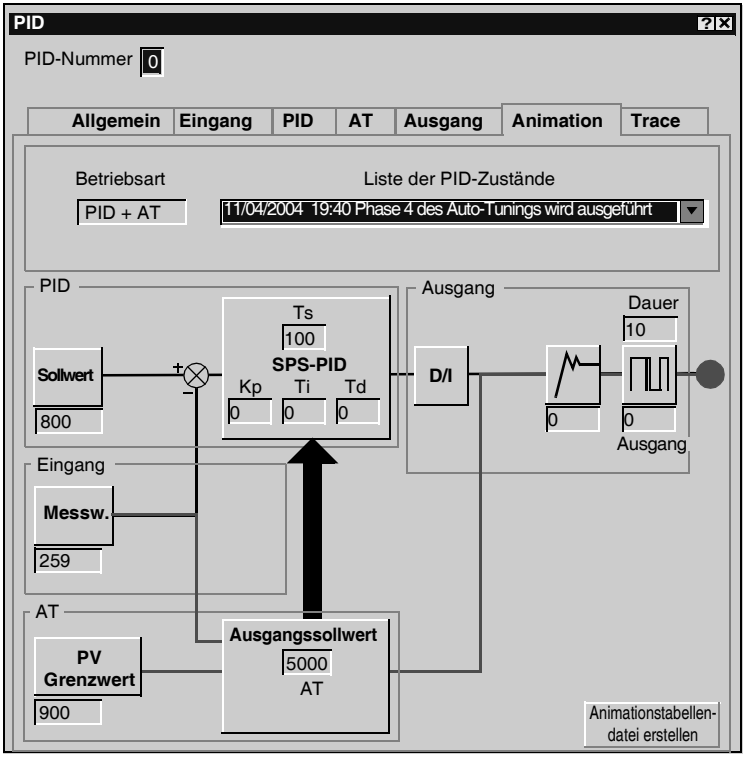
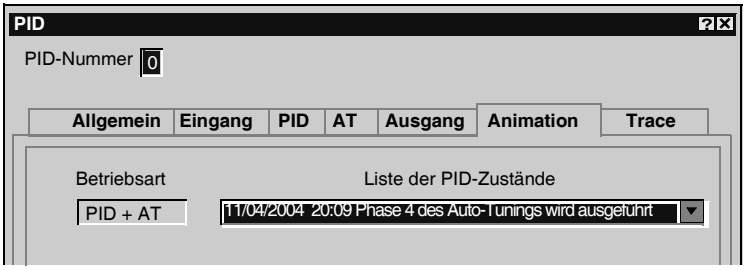
Hinweis zur Parametrierung von Kp, Ti und Td

Damit der Betrieb im Modus AT+PID möglich ist, müssen die beiden folgenden Bedingungen erfüllt sein:

- Die Koeffizienten **Kp, Ti und Td** müssen auf **Speicheradressen (%MWxx)** parametriert sein.
- Der **Aktion**styp auf der **Registerkarte Ausgang** muss auf ein **Speicherbitadresse (%Mxx)** parametriert sein.

Gehen Sie folgendermaßen vor, um den Regler in den Modus AT+PID zu versetzen:

Schritt	Aktion
1	Geben Sie für die Speicheradressen die folgenden Werte aus der Animationstabelle an: ● %M2: Wahl des automatischen oder manuellen Modus = 0, ● %MW0: Reglersollwert = 600 (im Beispiel ist der Sollwert nach der AT-Sequenz aktiv, und der Regler gewährleistet eine Temperatur von 60 °C), ● %MW10 à %MW12: Koeffizienten des PID-Reglers (auf 0 lassen, die AT-Sequenz gibt die Koeffizienten an), ● %MW13: Grenzwert des Messwerts, der im Modus AT nicht zu überschreiten ist = 900 (im Beispiel dürfen 90 °C nicht überschritten werden, da sonst ein AT-Fehler gemeldet wird), ● %MW14: Ausgangssollwert des Reglers im Modus AT = 2000 (abgeleitet aus dem Test im manuellen Modus). Es handelt sich um den auf den Prozess angewandten Wert der Schrittänderung. Im Modus AT wird der Ausgangssollwert direkt auf den Ausgang des Reglers angewandt. Dieser Wert kann entweder ein internes Wort (%MW0 bis %MW2999), eine interne Konstante (%KW0 bis %KW255) oder ein direkter Wert sein. Der Wert muss folglich zwischen 0 und 10.000 liegen. **Hinweis:** Der Sollwert des Auto-Tuning-Ausgangs muss immer größer als der zuletzt auf den Prozess angewandte Ausgang sein. ● %MW15: Digitalausgang des PID-Reglers (angegeben vom Regler), ● %MW16: Regelung der PWM-Periode (behalten Sie den zuvor eingestellten Wert 10 bei), ● %MW17: Wahl der Betriebsart des PID-Reglers = 2 (AT + PID), ● %MW18: mit der Wahl des Bits %M2 = 0 verbundener manueller Sollwert.
2	Konfigurieren Sie die Twido-Steuerung so, dass sie eine Abtastung im **periodischen Modus durchführt.**
3	Stellen Sie die **Dauer** der Abtastzeit der Twido-Steuerung so ein, dass der Wert der **Abtastzeit (Ts)** des PID-Reglers ein genaues Vielfaches hiervor ist. **Hinweis:** Ausführliche Informationen über die Bestimmung der Abtastzeit finden Sie unter *Auto-Tuning-Anforderungen, S. 569* und *Methoden zur Ermittlung der Abtastzeit (Ts), S. 570*.
4	Überprüfen Sie, ob sich die Steuerung im RUN-Modus befindet.
5	Geben Sie das Speicherbit %M0 an. %M0: Freigabe des Reglers = 1 in der Animationstabelle.
6	Doppelklicken Sie auf das Element **PID** im Konfigurationsnavigator.

Schritt	Aktion
7	Aktivieren Sie die Registerkarte **Animation** für die gewünschte PID-Nummer und überprüfen Sie, ob die Animation mit dem unten abgebildeten Fenster übereinstimmt:  **Hinweis:** Die Bildschirme des PID-Reglers werden nur aktualisiert, wenn der Regler freigegeben ist (und sich die SPS im RUN-Modus befindet).
8	Klicken Sie auf die Registerkarte **Trace** und warten sie, bis das System die AT-Sequenz startet.  **Hinweis:** Es kann mehrere Minuten dauern, bis das AT-Verfahren abgeschlossen ist.

Speicherung der berechneten Koeffizienten Kp, Ti und Td

Nach Abschluss der Auto-Tuning-Sequenz werden die berechneten Werte in den den Koeffizienten Kp, Ti und Td zugewiesenen Speicherwörtern gespeichert. Diese Werte werden in den RAM-Speicher geschrieben und in der Steuerung gespeichert, solange die Anwendung gültig ist (Abschaltung der Stromversorgung weniger als 30 Tage) und kein Kaltstart erfolgt (%S0).

Hinweis: Wenn das System keinen äußeren Schwankungen unterliegt, können die Werte **fest** in die Parametrierung des PID-Reglers geschrieben und der Regler nur in den PID-Modus versetzt werden.

Wiederholung der AT-Sequenz

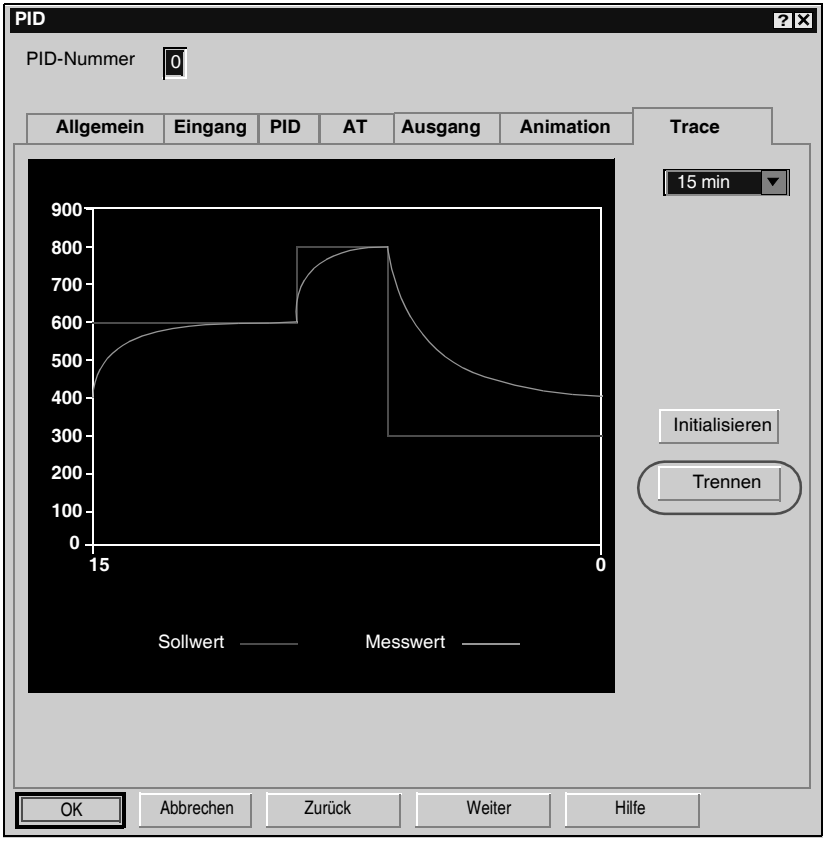
Die Auto-Tuning-Sequenz wird nach jedem Übergang in den RUN-Modus oder Kaltstart (%S0) wiederholt.
Es empfiehlt sich daher, die Diagnosewörter durch das Programm zu überprüfen, um das bei einem Neustart zu befolgende Verhalten zu bestimmen.

Schritt 6 - Debugging der Einstellungen

Zugriff auf die Animationstabelle

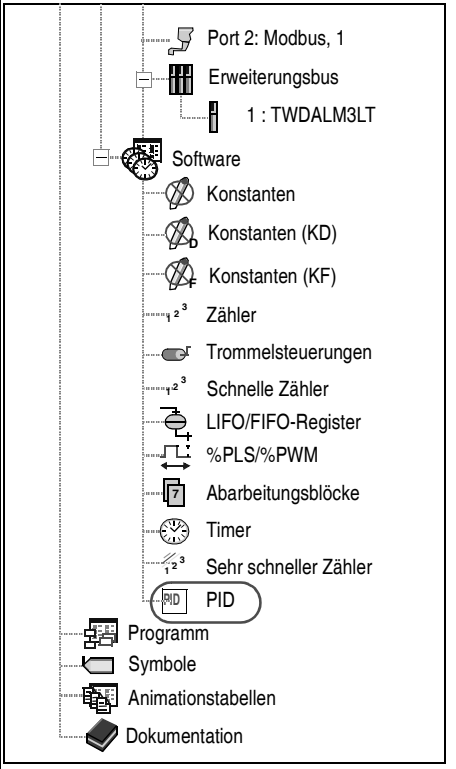
Um das Debugging des Systems zu erleichtern, ist der Zugriff auf die Animationstabelle immer möglich, wenn sich die Bildschirme des PID-Reglers im Vordergrund befinden.

Hinweis:
Bei Anzeige der Sollwert- und Messwertkurven nur über die Schaltfläche **Trennen** der Registerkarte **Trace** (siehe Fenster der Registerkarte "Trace" unten) ist der Zugriff auf die Animationstabelle über das Menü **Fenster** → **Animationstabelleneditor - Animation... möglich**



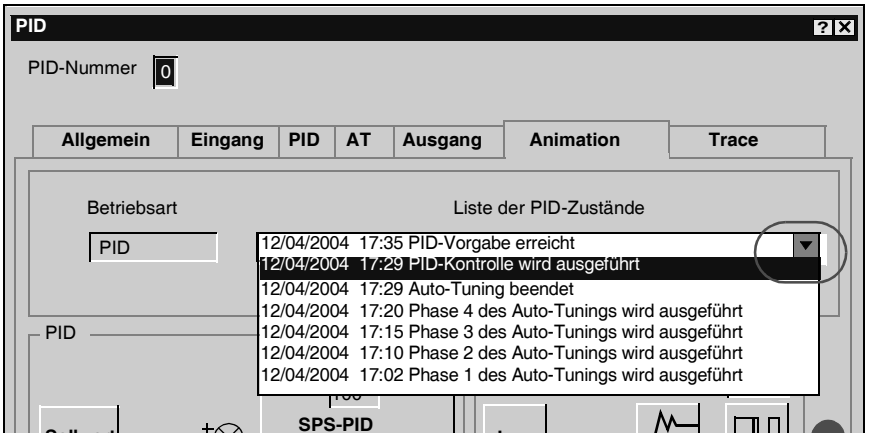
Rückkehr zu den PID-Bildschirmen

Gehen Sie folgendermaßen vor, um zu den Bildschirmen des PID-Reglers zurückzukehren, ohne den Verlauf der Kurven zu verlieren:

Schritt	Aktion
1	Doppelklicken Sie auf die Option **PID** im Konfigurationsnavigator im linken Teil des TwidoSoft-Fensters (siehe Abbildung unten). 
2	Wenn das Fenster des PID-Reglers angezeigt wird, wählen Sie die gewünschte PID-Nummer auf der Registerkarte **Allgemein** aus.

Verlauf der PID-Zustände

Auf der Registerkarte **Animation** der PID-Regler ist der Zugriff auf die letzten 15 Zustände des aktuellen Reglers durch Klicken auf die Dropdown-Liste möglich (siehe Abbildung unten).



Hinweis: Die PID-Zustände werden gespeichert, wenn der PC und die TwidoSoft-Steuerung online mit der Steuerung verbunden sind.

17.4 PID-Funktion

Auf einen Blick

Inhalt des Abschnitts

Dieser Abschnitt beschreibt das Verhalten, die Funktionen und die Inbetriebnahme der PID-Funktion.

Hinweis: Um schnell Konfigurationsinformationen über Ihre PID-Steuerung sowie über die PID-Auto-Tuning-Funktion zu finden, ziehen Sie die *Twido PID-Kurzanleitung, S. 509* zu Rate.

Inhalt dieses Abschnitts

Dieser Abschnitt enthält die folgenden Themen:

Thema	Seite
Allgemeine Beschreibung	535
Prinzip der Regelkreise	536
Methode zur Entwicklung einer Regelungsapplikation	537
Kompatibilität und Leistungsdaten	538
Ausführliche Beschreibung der Merkmale der PID-Funktion	539
Verfahren zum Zugreifen auf die Konfiguration des PID	542
Registerkarte "Allgemein" des PID	544
Registerkarte "Eingang" des PID	547
Registerkarte "PID" des PID	549
Registerkarte AT des PID	551
Registerkarte "Ausgang" des PID	556
Verfahren zum Zugreifen auf des Debugging des PID	559
Registerkarte "Animation" des PID	561
Registerkarte "Trace" des PID	563
PID-Zustände und Fehlercodes	565
PID-Einstellung mittels Auto-Tuning (AT)	569
PID-Parameter-Anpassungsverfahren	577
Rolle und Einfluss der Paramater eines PID	580
Anhang 1: Grundlegendes zur PID-Theorie	584
Anhang 2: Modell erster Ordnung mit Zeitverzögerung	586

Allgemeine Beschreibung

Allgemein

Die PID-Regelungsfunktion ist eine Funktion der Programmiersprache TwidoSoft. Sie ermöglicht die Programmierung von PID-Regelschleifen an mit Twido ab der Version 2.0 kompatiblen Steuerungen.

Diese Funktion ist speziell geeignet für:
- Die Erfüllung der Anforderungen sequentieller Verfahren, bei denen Hilfsregelungsfunktionen erforderlich sind (wie zum Beispiel: Plastikfolienverpackungsmaschinen, Maschinen zur Oberflächenbehandlung, Pressen usw.),
- die Erfüllung der Anforderungen von einfachen Regelungsprozessen (wie zum Beispiel: Öfen zur Metallbearbeitung, Keramiköfen, kleine Kühlgruppen usw.).

Ihre Implementierung ist sehr einfach, da sie über folgende Menüs erfolgt:
- Konfiguration
- Debug-Menü

in Verbindung mit einer Programmzeile (Operationsblock in KOP oder einfacher PID-Aufruf in AWL), welche die Nummer des verwendeten PID angibt.
Beispiel einer Programmzeile in der Kontaktplansprache:

Hinweis: In ein und derselben Twido-Automatisierungsanwendung können maximal 14 PID-Funktionen konfiguriert werden.

Hauptfunktionen

Nachfolgend sind die Hauptfunktionen aufgeführt:
- Analogeingang
- lineare Konvertierung des konfigurierbaren Messwerts
- Überwachung des oberen und unteren Grenzwerts am konfigurierbaren Eingang
- Analogausgang oder Impulsbreitenmodulation
- Spitzenbegrenzung des konfigurierbaren Ausgangs
- Konfigurierbare direkte oder inverse Aktion

Prinzip der Regelkreise

Auf einen Blick

Der Betrieb eines Regelkreises beinhaltet drei verschiedene Phasen:

- Die Erfassung der Daten:
  - vom Impulsgeber (Analogimpulsgeber, Geber) stammende (n) Messwerts (Messwerte) ,
  - des (der) allgemein von den internen Variablen der Steuerung oder von den von einer TwidoSoft-Animationstabelle ausgegebenen Daten stammende (n) Sollwerts (Sollwerte).
- Ausführung des Regelungsalgorithmus des PID-Reglers,
- Übermittlung der an die Eigenschaften der Stellglieder angepassten Befehle, die über die digitalen Ausgänge (PWM) oder über die analogen Ausgänge zu regeln sind.

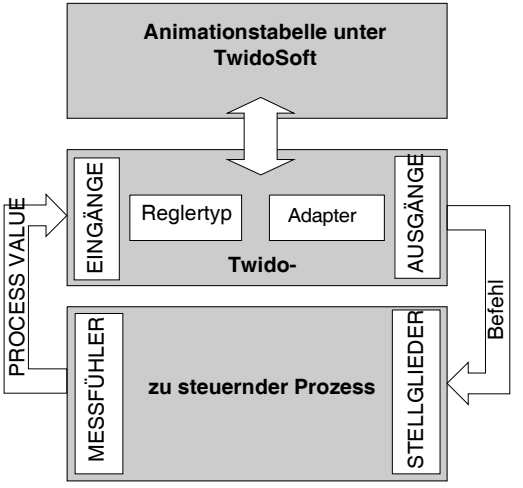
Der PID-Regleralgorithmus bearbeitet das auszugebende Befehlssignal:

- des vom Eingangsmodul korrigierten Messwerts,
- des entweder vom Bediener oder vom Programm festgelegten Sollwerts,
- der Werte der verschiedenen Reglerparameter.

Das vom Regler ausgegebene Signal wird entweder direkt über eine analoge Ausgangskarte der an das Stellglied angeschlossenen Steuerung bearbeitet oder über eine PWM-Anpassung zur Regelung über einen digitalen Ausgang der Steuerung.

Abbildung

Die unten dargestellte Abbildung zeigt das Prinzip eines Regelkreises.

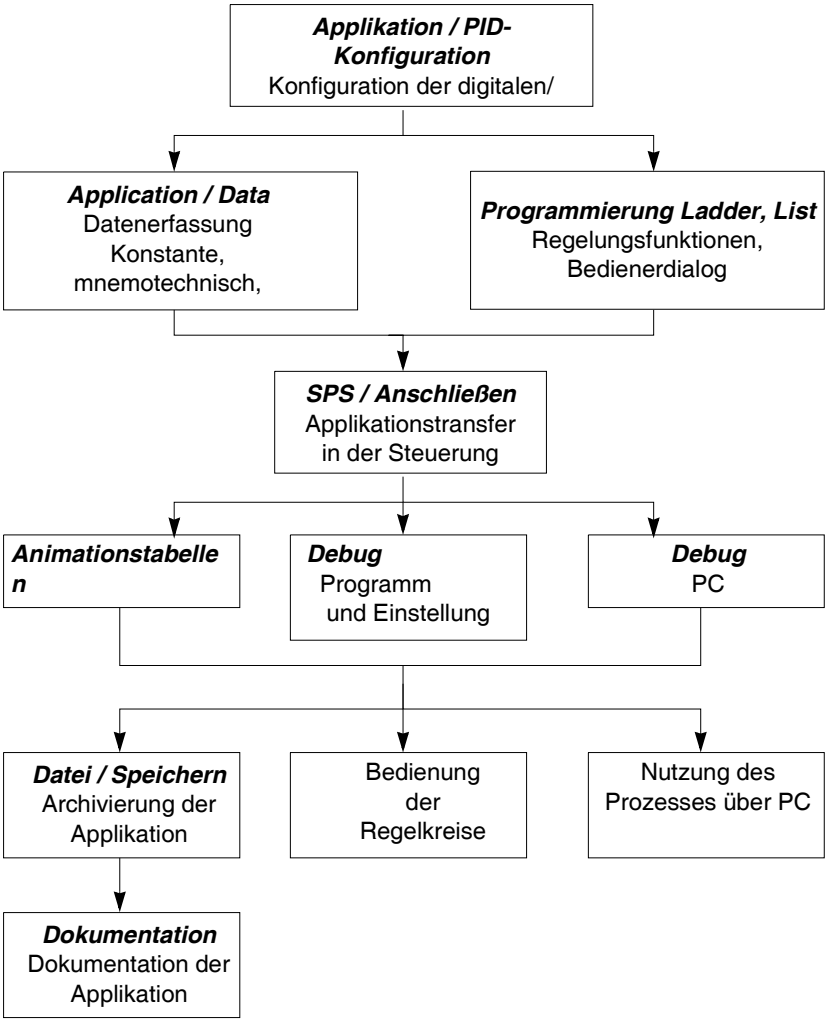


Methode zur Entwicklung einer Regelungsapplikation

Ablaufdiagramm

Das folgende Diagramm zeigt die Tasks bei Erstellung und Debug einer Regelungsapplikation.

Hinweis: Die festgelegte Reihenfolge ist abhängig von Ihrer Arbeitsmethode, sie ist hier nur zu Informationszwecken angegeben.



Kompatibilität und Leistungsdaten

Auf einen Blick

Die PID-Funktion von Twido ist eine für mit TwidoSoft ab der Version 2.0 kompatible Steuerungen verfügbare Funktion. Daher unterliegt ihre Nutzung einer Reihe von Hard- und Software-Kompatibilitätsanforderungen, die in den folgenden Abschnitten beschrieben sind.
Darüber hinaus erfordert diese Funktion Ressourcen, die im Abschnitt **Leistungen** beschrieben sind.

Kompatibilität

Die PID-Funktion von Twido ist ab der Twido-Steuerungsversion 2.0 verfügbar. Wenn Sie über Twido-Steuerungen mit einer Softwareversion unter 1.2 verfügen, müssen Sie die Firmware aktualisieren, um die PID-Funktion nutzen zu können.

Hinweis: Die analogen Ein- und Ausgangsmodule der Version 1.0 können ohne Aktualisierung als PID-Ein- oder Ausgänge verwendet werden.

Um einen PID an diesen verschiedenen Hardwareversionen konfigurieren und programmieren zu können, müssen Sie über die Version **2.0 oder höher der Software TwidoSoft** verfügen.

Leistungsdaten

Die PID-Regelschleifen weisen die folgenden Leistungsdaten auf:

Beschreibung	Dauer
Ausführungszeit einer Schleife	0,4 ms

Ausführliche Beschreibung der Merkmale der PID-Funktion

Allgemeines

Die PID-Funktion führt ausgehend von einem Messwert und einem analogen Sollwert im Standardformat [0 -10.000] eine PID-Korrektur durch und liefert einen analogen Befehl im selben Format oder eine Impulsbreitenmodulation (PWM) an einem Digitalausgang.
Alle Parameter des PID sind in den zur Konfiguration des PID vorgesehenen Fenstern beschrieben. Wir beschränken uns hier in einer Funktionsübersicht auf die Zusammenfassung der verfügbaren Funktionen, auf die Angabe ihrer Messbereiche und auf die Beschreibung ihrer Integration ins PID.

Hinweis: Für eine Nutzung im Vollskalabereich (optimale Auflösung) können Sie Ihren an den Messzweig des PID angeschlossenen Analogeingang im Bereich von 0 bis 10.000 konfigurieren. Wenn Sie jedoch die Standardkonfiguration von 0 bis 4.095 nutzen, funktioniert der Regler richtig.

Hinweis: Damit die Regelung richtig funktionieren kann, muss sich die Twido-Steuerung **unbedingt im periodischen Modus befinden**. Die PID-Funktion wird in diesem Modus periodisch bei jedem Zyklus ausgeführt, und die Abtastung der Eingangsdaten des PID berücksichtigt das in der Konfiguration festgelegte Intervall (siehe folgende Tabelle).

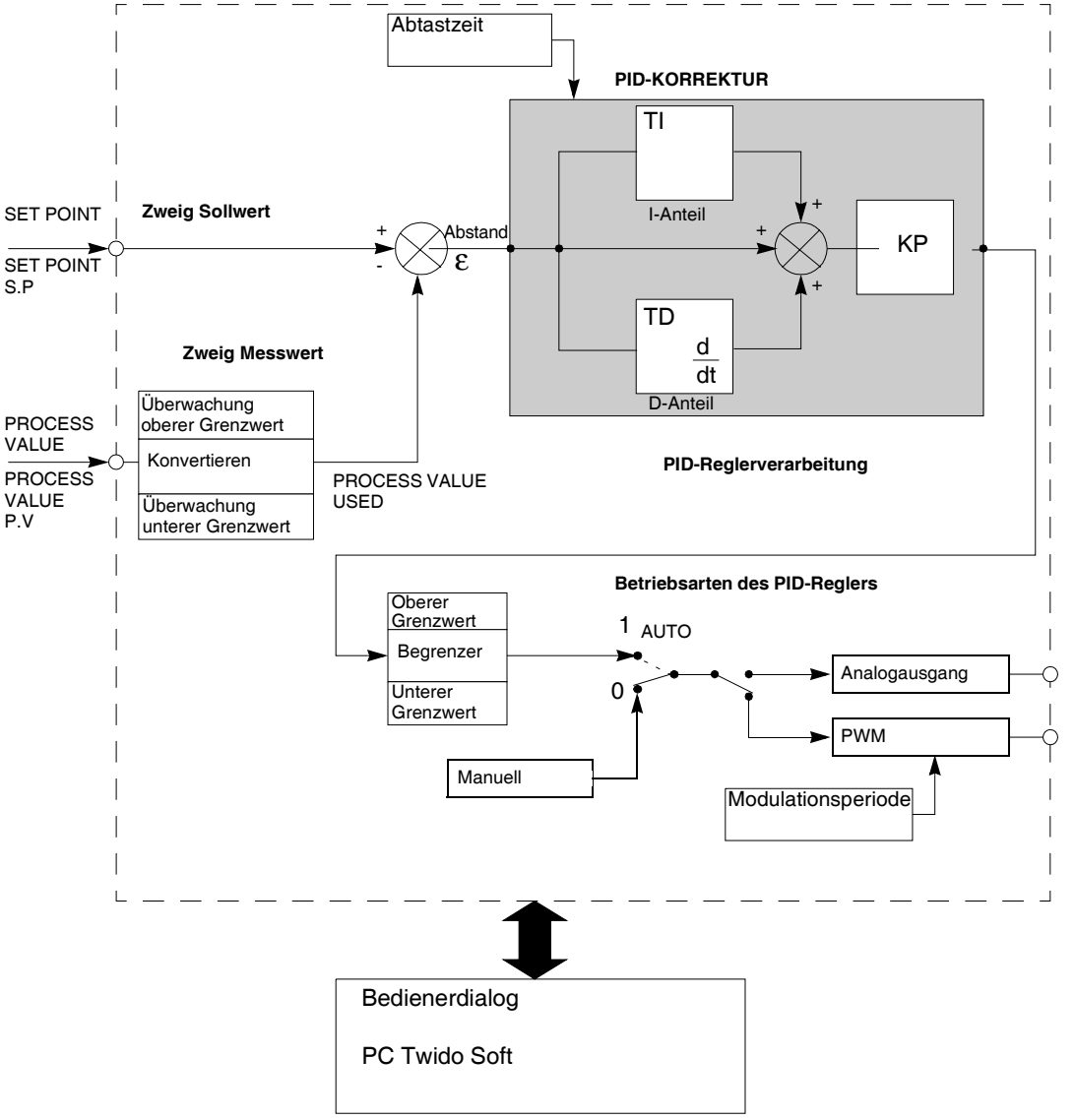
Überblick über die verfügbaren Funktionen

In der folgenden Tabelle sind die verschiedenen verfügbaren Funktionen und ihre Skalen beschrieben.

Funktion	Skala und Kommentar
Lineare Konvertierung des Eingangs	Die Konvertierung ermöglicht die Umwandlung eines Wertes zwischen 0 und 10.000 (Auflösung des analogen Eingangsmoduls) in einen Wert zwischen -32.768 und 32.767.
Proportionale Verstärkung	Verknüpft mit einem Faktor von 100 liegt ihr Wert zwischen 1 und 10.000. Dies entspricht einem Verstärkungsfaktor von 0,01 bis 100. **Hinweis:** Wenn Sie einen ungültigen Verstärkungsbereich (negativ oder Null) eingeben, ignoriert TwidoSoft diese Benutzereinstellung und weist diesem Faktor automatisch den Standardwert 100 zu.
Integralzeit	Verknüpft mit einer Zeitbasis von 0,1 Sekunden, liegt ihr Wert zwischen 0 und 20.000. Dies entspricht einer Integralzeit zwischen 0 und 2000,0 Sekunden.
Abgeleitete Zeit	Verknüpft mit einer Zeitbasis von 0,1 Sekunden, liegt ihr Wert zwischen 0 und 10.000. Dies entspricht einer abgeleiteten Zeit zwischen 0 und 1000,0 Sekunden.
Abtastzeit	Verknüpft mit einer Zeitbasis von 0,01 Sekunden, liegt ihr Wert zwischen 1 und 10.000. Dies entspricht einer Zeit zwischen 0,01 und 100 Sekunden.
PWM-Ausgang	Verknüpft mit einer Zeitbasis von 0,1 Sekunden, liegt ihr Wert zwischen 1 und 500. Dies entspricht einer Modulationszeit zwischen 0,1 und 50 Sekunden.
Analogausgang	Wert zwischen 0 und +10000
Alarm, Messbereich überschritten	Dieser Alarm wird nach der Konvertierung definiert. Er liegt zwischen -32.768 und 32.767, wenn die Konvertierung aktiv ist, und zwischen 0 und 10.000, wenn die Konvertierung nicht aktiv ist.
Alarm, Messbereich unterschritten	Dieser Alarm wird nach der Konvertierung definiert. Er liegt zwischen -32.768 und 32.767, wenn die Konvertierung aktiv ist, und zwischen 0 und 10.000, wenn die Konvertierung nicht aktiv ist.
Oberer Grenzwert am Ausgang	Dieser Grenzwert liegt zwischen 0 und 10.000 für einen analogen Ausgangswert. Wenn PWM aktiv ist, entspricht der Grenzwert einem Prozentsatz der modulierten Dauer. 0% für 0 und 100% für 10000.
Unterer Grenzwert am Ausgang	Dieser Grenzwert liegt zwischen 0 und 10.000 für einen analogen Ausgangswert. Wenn PWM aktiv ist, entspricht der Grenzwert einem Prozentsatz der modulierten Dauer. 0% für 0 und 100% für 10000.
Manueller Modus	Wenn der manuelle Modus aktiviert ist, ist der Ausgang identisch mit einem vom Benutzer parametrierten Festwert. Dieser Ausgang liegt zwischen 0 und 10.000 (0 bis 100 % für einen PWM-Ausgang).
Direkte oder inverse Aktion	Die direkte oder inverse Aktion ist verfügbar, sie wirkt direkt auf den Ausgang.
Auto-Tuning (AT)	Diese Funktion führt eine automatische Einstellung der Parameter "Kp", "Ti", "Td" und "Direkte/Inverse Aktion" durch, um eine optimale Konvergenz des Steuerungsprozesses zu erreichen.

Hinweis: Ziehen Sie für ein besseres Verständnis der Funktionsweise aller in der vorhergehenden Tabelle aufgeführten Funktionen die nachfolgende Übersicht zu Rate.

Funktionsweise Das folgende Schema veranschaulicht die Funktionsweise der PID-Funktion.



Hinweis: Die verwendeten Parameter sind in der Tabelle auf der vorherigen Seite und in den Konfigurationsfenstern beschrieben.

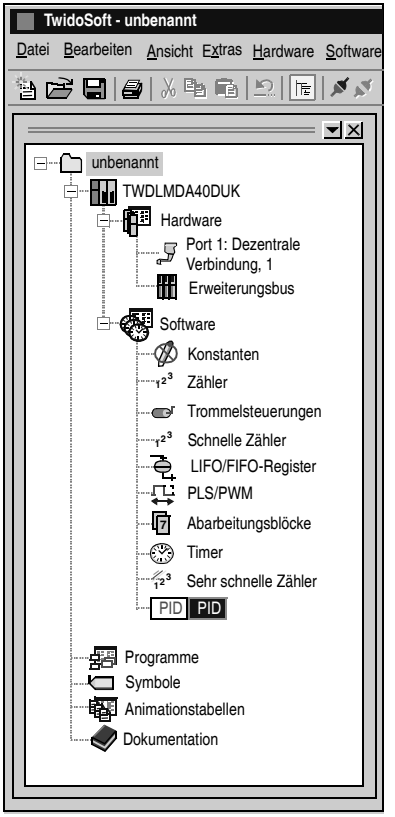
Verfahren zum Zugreifen auf die Konfiguration des PID

Auf einen Blick

Im Folgenden wird der Zugriff auf die Konfigurationsfenster eines PID an TWIDO-Steuerungen beschrieben.

Vorgehensweise

Die folgende Tabelle beschreibt die Vorgehensweise zum Zugriff auf die Konfigurationsfenster eines PID:

Schritt	Aktion
1	Vergewissern Sie sich, dass Sie sich im **Offline-Modus** befinden.
2	Öffnen Sie den Navigator. **Ergebnis:** 

Schritt	Aktion
3	Doppelklicken Sie auf **PID**. **Ergebnis:** Das Parametrierungsfenster des PID wird angezeigt. Standardmäßig erscheint es auf der Registerkarte **Allgemein** (Siehe *Registerkarte "Allgemein" des PID, S. 544*). **Hinweis**: Sie können auch mit der rechten Maustaste auf **PID** klicken und die Option **Bearbeiten** wählen oder das Menü **Software** → **PID** auswählen oder das Menü **Programm** → **Konfigurationseditor** → **PID-Symbol** verwenden oder im letzten Fall den PID auswählen und auf das Symbol **Lupe** klicken, um einen bestimmten PID auszuwählen.

Registerkarte "Allgemein" des PID

Auf einen Blick

Wenn Sie den PID vom Navigator aus öffnen, öffnen Sie das PID-Konfigurationsfenster. Dieses Fenster ermöglicht:

- Konfigurieren jedes PID von TWIDO,
- Debuggen jedes PID von TWIDO,

Wenn Sie das Fenster aufrufen und sich in folgendem Modus befinden:

- Offline-Modus: öffnet sich standardmäßig die Registerkarte **Allgemein**, und Sie haben Zugriff auf die Konfigurationsparameter,
- Online-Modus: öffnet sich die Registerkarte **Animation**, und Sie haben Zugriff auf die Debug- und Einstellparameter,

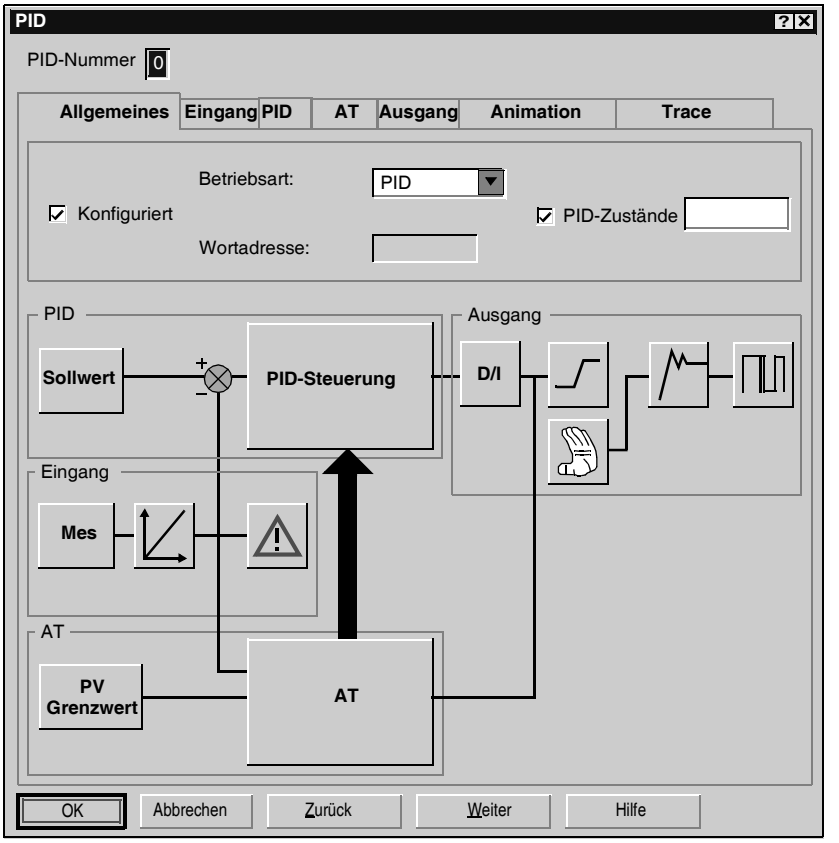
Hinweis: In einigen Fällen sind die abgeblendeten Registerkarten und Felder möglicherweise aus einem der beiden nachfolgend aufgeführten Gründe nicht verfügbar: Der Betriebsmodus "Nur PID" ist ausgewählt, was den Zugriff auf die Parameter der Registerkarte "AT" verhindert, die nicht länger benötigt werden.

- Der aktuell aktive Betriebsmodus (offline oder online) ermöglicht keinen Zugriff auf diese Parameter.
- Der Betriebsmodus "Nur PID" ist ausgewählt, was den Zugriff auf die Parameter der Registerkarte "AT" verhindert, die nicht länger benötigt werden.

Im Folgenden wird die Registerkarte **Allgemein** beschrieben.

Registerkarte "Allgemein" des PID

Das folgende Fenster ermöglicht die Angabe der allgemeinen Parameter des PID.



Beschreibung In der folgenden Tabelle sind die Parameter beschrieben, die Sie definieren können.

Feld	Beschreibung
PID-Nummer	Geben Sie hier die Nummer des PID ein, den Sie konfigurieren möchten. Der Wert liegt zwischen 0 und 13; pro Applikation sind maximal 14 PID zulässig.
Konfiguriert	Um die PID-Regelung zu konfigurieren, muss dieses Kontrollkästchen aktiviert sein. Ansonsten kann keine Aktion in diesen Bildschirmen ausgeführt werden und die PID-Regelung kann, obwohl sie in der Applikation vorhanden ist, nicht genutzt werden.
Betriebsmodus	Geben Sie hier den gewünschten Betriebsmodus an. Sie können aus drei Betriebsmodi und einer Wortadresse wie folgt auswählen: ● PID ● AT ● AT+PID ● Wortadresse
Wortadresse	Sie können in dieses Textfeld ein internes Wort (%MW0 bis %MW2999) eingeben, das verwendet wird, um den Betriebsmodus per Programm festzulegen. Das interne Wort kann abhängig von dem Betriebsmodus, den Sie einstellen möchten, drei mögliche Werte annehmen: ● %MWx = 1 (zum Einstellen von "Nur PID) ● %MWx = 2 (zum Einstellen von "AT + PID") ● %MWx = 3 (zum Einstellen von "Nur AT")
PID-Zustände	Wenn Sie diese Option aktivieren, können Sie in dieses Textfeld ein Speicherwort (%MW0 bis %MW2999) eingeben, das vom PID-Regler verwendet wird, um den aktuellen PID-Zustand während der Ausführung des PID-Reglers und/oder der Auto-Tuning-Funktion zu speichern (ausführliche Informationen finden Sie unter *PID-Zustände und Fehlercodes, S. 565*.)
Übersicht	Die Übersicht ermöglicht Ihnen die Anzeige der verschiedenen Möglichkeiten zur Konfiguration Ihres PID.

Registerkarte "Eingang" des PID

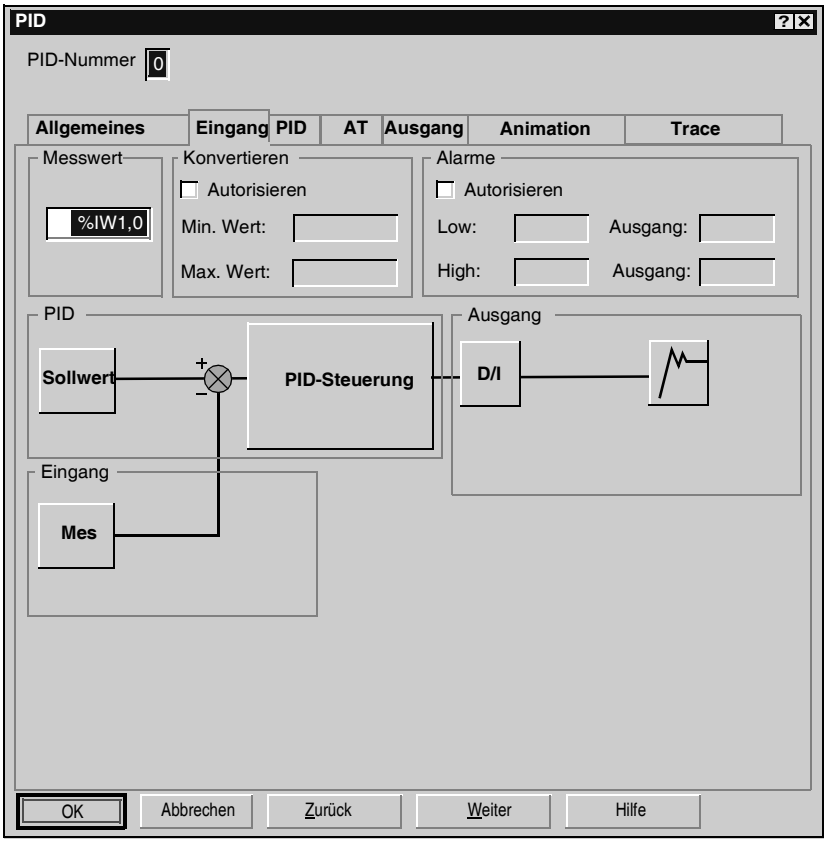
Auf einen Blick

Diese Registerkarte ermöglicht die Festlegung der Eingangsparameter des PID.

Hinweis: Diese Registerkarte ist im Offline-Modus zugänglich.

Registerkarte "Eingang" der PID-Funktion

Das folgende Fenster ermöglicht die Angabe der Eingangsparameter des PID.



Beschreibung In der folgenden Tabelle sind die Parameter beschrieben, die Sie definieren können.

Feld	Beschreibung
PID-Nummer	Geben Sie hier die Nummer des PID ein, den Sie konfigurieren möchten. Der Wert liegt zwischen 0 und 13; pro Applikation sind maximal 14 PID zulässig.
Istwert	Geben Sie hier die Variable an, die den Messwert des zu regulierenden Prozesses enthält. Die Standardskala reicht von 0 bis 10000. Sie können entweder ein internes Wort (%MW0 bis %MW2999) oder einen analogen Eingang (%IWx.0 bis %IWx.1) eingeben.
Konvertieren	Aktivieren Sie dieses Kontrollkästchen, wenn Sie die als einen PID-Eingang angegebene Prozessvariable konvertieren wollen. Wenn dieses Kontrollkästchen aktiviert ist, haben Sie Zugriff auf die beiden Felder **Min. Wert** und **Max. Wert**. Die Konvertierung ist linear und wandelt einen Wert zwischen 0 und 10.000 in einen Wert um, dessen Minimum und Maximum zwischen -32768 und +32767 liegen können.
Min. Wert **Max. Wert**	Geben Sie den Minimal- und Maximalwert der Konvertierungsskala an. Die Prozessvariable wird dann automatisch innerhalb des Intervalls **[Min.-Wert bis Max.-Wert]** neu skaliert. **Hinweis**: Der **Min. Wert** muss auf jeden Fall kleiner als der **Max. Wert** sein. **Min. Wert** oder **Max. Wert** können entweder interne Wörter (%MW0 bis %MW2999) oder interne Konstanten (%KW0 bis %KW255) oder ein Wert zwischen -32768 und +32767 sein.
Alarme	Aktivieren Sie dieses Kontrollkästchen, wenn Sie die Alarme für Eingansvariablen aktivieren wollen. **Hinweis**: Die Alarmwerte sind anhand der nach der Konvertierungsphase erhaltenen Prozessvariable festzulegen. Sie müssen daher zwischen **Min.-Wert** und **Max.-Wert** liegen, wenn die Konvertierung aktiv ist. Ansonsten liegen sie zwischen **0 und 10000.**
Low Ausgang	Geben Sie den oberen Alarmwert in das Feld **Low** ein. Dieser Wert kann entweder ein internes Wort (%MW0 bis %MW2999), eine interne Konstante (%KW0 bis %KW255) oder ein direkter Wert sein. **Ausgang** muss die Adresse enthalten, die auf 1 gesetzt wird, wenn der untere Grenzwert erreicht ist. **Ausgang** kann entweder ein internes Bit (%M0 bis %M255) oder ein Ausgang sein (%Qx.0 bis %Qx.32).
High Ausgang	Geben Sie den unteren Alarmwert in das Feld **High** ein. Dieser Wert kann entweder ein internes Wort (%MW0 bis %MW2999), eine interne Konstante (%KW0 bis %KW255) oder ein direkter Wert sein. **Ausgang** muss die Adresse enthalten, die auf 1 gesetzt wird, wenn der obere Grenzwert erreicht ist. **Ausgang** kann entweder ein internes Bit (%M0 bis %M255) oder ein Ausgang sein (%Qx.0 bis %Qx.32).
Übersicht	Die Übersicht ermöglicht Ihnen die Anzeige der verschiedenen Möglichkeiten zur Konfiguration Ihres PID.

Registerkarte "PID" des PID

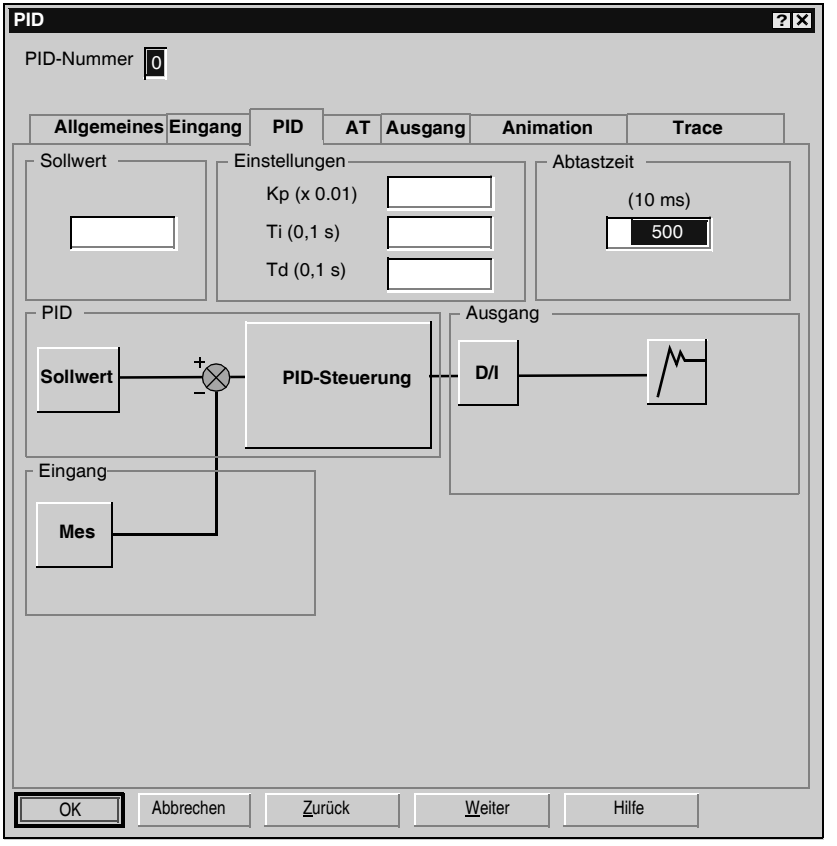
Auf einen Blick

Diese Registerkarte ermöglicht die Festlegung der internen Parameter des PID.

Hinweis: Diese Registerkarte ist im Offline-Modus zugänglich.

Registerkarte "PID" des PID

Das folgende Fenster ermöglicht die Angabe der internen Parameter des PID.



Beschreibung In der folgenden Tabelle sind die Parameter beschrieben, die Sie definieren können.

Feld	Beschreibung
PID-Nummer	Geben Sie hier die Nummer des PID ein, den Sie konfigurieren möchten. Der Wert liegt zwischen 0 und 13; pro Applikation sind maximal 14 PID zulässig.
Sollwert	Geben Sie hier des Sollwert des PID an. Dieser Wert kann entweder ein internes Wort (%MW0 bis %MW2999), eine interne Konstante (%KW0 bis %KW255) oder ein direkter Wert sein. Dieser Wert muss zwischen 0 und 10000 liegen, wenn die Konvertierung deaktiviert ist. Wenn die Konvertierung aktiviert ist, muss der Wert zwischen dem **Minimal- und dem Maximalwert** der Konvertierung liegen.
Kp * 100	Geben Sie hier den Proportionskoeffizienten des PID multipliziert mit 100 ein. Dieser Wert kann entweder ein internes Wort (%MW0 bis %MW2999), eine interne Konstante (%KW0 bis %KW255) oder ein direkter Wert sein. Der gültige Bereich für den Kp-Parameter lautet: 0 < Kp < 10000. **Hinweis:** Wenn Kp versehentlich auf 0 gesetzt wird (Kp ≤ 0 ist ungültig), wird der Standardwert Kp=100 automatisch durch die PID-Funktion zugewiesen.
TI (0,1 s)	Geben Sie hier den I-Anteilskoeffizienten in Bezug auf eine Zeitbasis von 0,1 Sekunde an. Dieser Wert kann entweder ein internes Wort (%MW0 bis %MW2999), eine interne Konstante (%KW0 bis %KW255) oder ein direkter Wert sein. Der Wert muss zwischen 0 und 20000 liegen. **Hinweis:** Setzen Sie diesen Koeffizienten auf 0, um die integrale Aktion des PID zu deaktivieren.
Td (0,1 sec)	Geben Sie hier den D-Anteilskoeffizienten in Bezug auf eine Zeitbasis von 0,1 Sekunde an. Dieser Wert kann entweder ein internes Wort (%MW0 bis %MW2999), eine interne Konstante (%KW0 bis %KW255) oder ein direkter Wert sein. Der Wert muss zwischen 0 und 10000 liegen. **Hinweis:** Setzen Sie diesen Koeffizienten auf 0, um die abgeleitete Aktion des PID zu deaktivieren.
Abtastzeit	Geben Sie hier die Abtastzeit des PID in Bezug auf eine Zeitbasis von 10^{-2} Sekunden (10 ms) an. Dieser Wert kann entweder ein internes Wort (%MW0 bis %MW2999), eine interne Konstante (%KW0 bis %KW255) oder ein direkter Wert sein. Der Wert muss zwischen 1 (0,01 s) und 10000 (100 s) liegen.
Übersicht	Die Übersicht ermöglicht Ihnen die Anzeige der verschiedenen Möglichkeiten zur Konfiguration Ihres PID.

Hinweis: Wenn AT aktiviert ist, werden die Parameter "Kp", "Ti" und "Td" nicht mehr durch den Benutzer gesetzt, da sie automatisch und programmgesteuert durch den AT-Algorithmus gesetzt werden. In diesem Fall müssen Sie in diese Felder nur ein **internes Wort** (%MW0 bis %MW2999) eingeben.
Vorsicht: Geben Sie keine interne Konstante oder einen Direktwert ein, wenn AT aktiviert ist, da dies bei der Ausführung Ihrer PID-Applikation zu einem Fehler führt.

Registerkarte AT des PID

Auf einen Blick

Die Einstellung korrekter PID-Parameter kann mühsam, zeitaufwendig und fehleranfällig sein. All dies macht die Konfiguration der Prozesssteuerung für den erfahrenen, jedoch nicht unbedingt professionellen Prozesssteuerungsanwender schwierig. Daher kann es manchmal schwierig sein, eine optimale Einstellung zu erreichen.
Der PID-Auto-Tuning-Algorithmus dient der automatischen und adäquaten Ermittlung der folgenden vier PID-Elemente:

- Verstärkungsfaktor,
- Integralwert,
- abgeleiteter Wert,
- direkte oder inverse Aktion.

Folglich kann die AT-Funktion eine schnelle und optimale Einstellung für die Prozessschleife bieten.

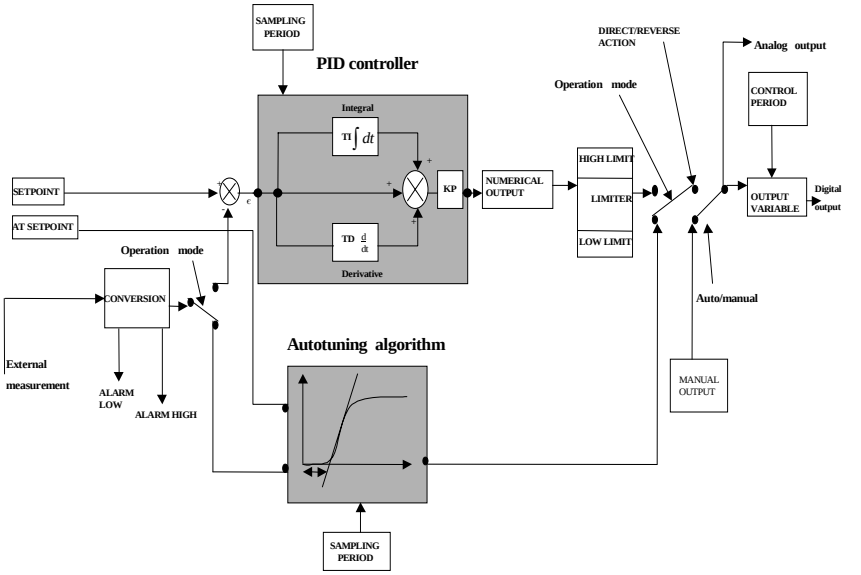
AT-Anforderungen

Das PID-Auto-Tuning ist ideal geeignet für Temperatursteuerungsprozesse.
Im Allgemeinen müssen die Prozesse, die mittels der AT-Funktion gesteuert werden können, die folgenden Anforderungen erfüllen:

- der Prozess ist im gesamten Betriebsbereich größtenteils linear,
- die Prozessreaktion auf eine Pegeländerung des Analogausgangs folgt einem vorübergehend asymptotischen Muster, und
- es gibt nur geringfügige Störungen bei den Prozessvariablen. (Im Fall eines Temperatursteuerungsprozesses bedeutet dies, dass kein abnormal hoher Wärmeaustausch zwischen dem Prozess und seiner Umgebung vorliegt.)

AT-Funktionsweise

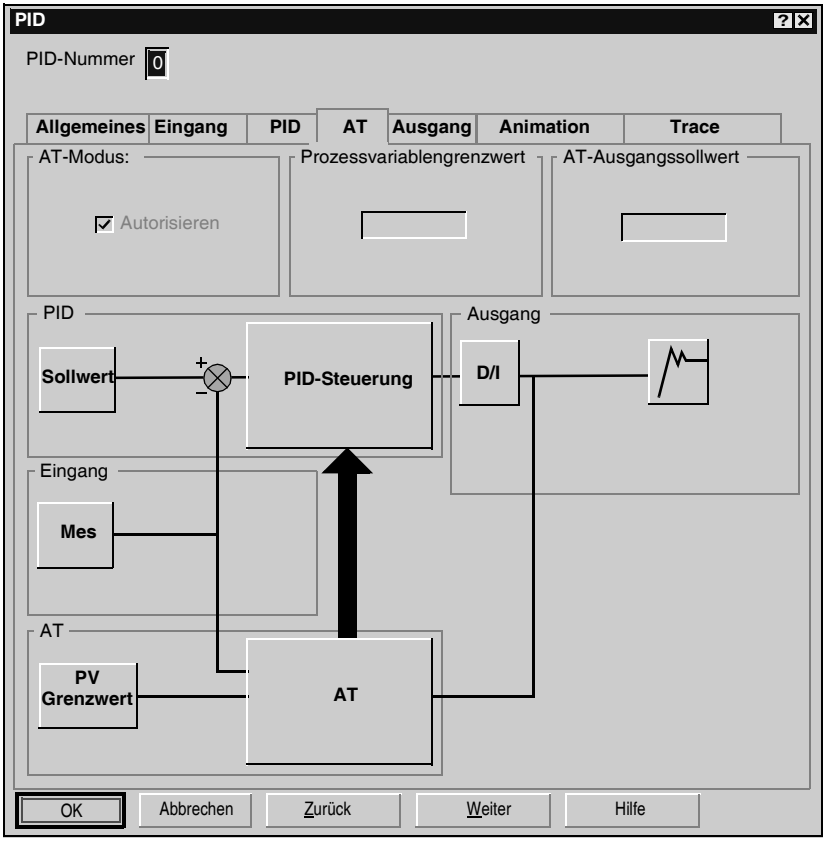
Das folgende Diagramm beschreibt das Arbeitsprinzip der AT-Funktion und wie sie mit den PID-Schleifen interagiert.



Registerkarte "AT" der PID-Funktion

Das folgende Fenster ermöglicht die Aktivierung/Deaktivierung der AT-Funktion und die Eingabe der AT-Parameter.

Hinweis: Diese Registerkarte ist nur im Offline-Modus zugänglich.



Beschreibung

⚠ WARNUNG

DIE PROZESSVARIABLEN- (PV) GRENZWERTE UND DIE AUSGANGS-SOLLWERTE MÜSSEN SORGFÄLTIG AUSGEWÄHLT WERDEN.

PID-Auto-Tuning ist ein rückführungsfreier Prozess, der ohne Vorschriften oder irgendwelche anderen als die durch den Prozessvariablen-Grenzwert oder den Ausgangs-Sollwert vorgegebenen Einschränkungen direkt auf den Steuerungsprozess wirkt. Daher müssen beide Werte sorgfältig innerhalb des zulässigen, durch den Prozess vorgegebenen Bereichs ausgewählt werden, um eine mögliche Überlastung des Prozesses zu vermeiden.

Die Nichtbeachtung dieser Anweisung kann den Tod, körperverletzung Oder materialschäden Zur folge Haben!

In der folgenden Tabelle sind die Parameter beschrieben, die Sie definieren können.

Feld	Beschreibung
Autorisieren	Aktivieren Sie dieses Kontrollkästchen, wenn Sie den AT-Modus aktivieren möchten. Es gibt zwei Möglichkeiten, dieses Kontrollkästchen zu verwenden. Sie sind davon abhängig, ob Sie die Betriebsart manuell oder über eine Wortadresse auf der Registerkarte "Allgemein" der PID-Funktion auswählen: • Wenn Sie den **Betriebsmodus** auf der Registerkarte **Allgemein** auf **PID+AT** oder **AT** setzen (siehe *Registerkarte "Allgemein" des PID, S. 544*), dann ist die Option "Autorisieren" automatisch aktiviert und abgeblendet (sie kann nicht deaktiviert werden). • Wenn Sie den Betriebsmodus über eine Wortadresse festlegen %MWx (%MWx = 2: PID+AT; %MWx = 3: AT), dann müssen Sie die Option "Autorisieren" manuell aktivieren, um die Konfiguration der AT-Parameter zu ermöglichen. **Ergebnis:** In beiden oben aufgeführten Fällen sind alle Felder dieses Konfigurationsfensters der Registerkarte "AT" aktiv, und Sie müssen die entsprechenden Werte in die Felder "Sollwert" und "Ausgang" eingeben.
Prozessvariablengrenzwert	Geben Sie den Grenzwert an, den die gemessene Prozessvariable während des AT-Prozesses nicht überschreiten darf. Dieser Parameter bietet Sicherheit für das Steuerungssystem, da AT ein rückführungsfreier Prozess ist. Dieser Wert kann ein internes Wort (%MW0 bis maximal %MW2999, abhängig vom verfügbaren Systemspeicher), eine interne Konstante (%KW0 bis %KW255) oder ein Direktwert sein. Dieser Wert muss zwischen 0 und 10000 liegen, wenn die Konvertierung deaktiviert ist. Wenn die Konvertierung aktiviert ist, muss der Wert zwischen dem Minimal- und dem Maximalwert der Konvertierung liegen.

Feld	Beschreibung
AT-Ausgangssollwert	Geben Sie hier den AT-Ausgangswert an. Dies ist der Wert der Schrittänderung, der auf den Prozess angewandt wird. Dieser Wert kann entweder ein internes Wort (%MW0 bis %MW2999), eine interne Konstante (%KW0 bis %KW255) oder ein direkter Wert sein. Dieser Wert muss zwischen 0 und 10000 liegen. **Hinweis:** Der AT-Ausgangssollwert muss immer größer als der zuletzt auf den Prozess angewandte Ausgang sein.

Hinweis: Wenn die AT-Funktion aktiviert ist, sind Konstanten (%KWx) oder Direktwerte nicht länger zulässig. In den folgenden PID-Feldern sind nur Speicherwörter zulässig:

- Die Parameter **Kp, Ti** und **Td** müssen als **Speicherwörter** (%MWx) auf der Registerkarte "PID" gesetzt werden.
- Das Feld **Aktion** wird auf der Registerkarte "OUT" automatisch auf **"Adressbit"** gesetzt.
- Auf der Registerkarte OUT muss im Feld **Bit** ein adäquates **Speicherbit** (%Mx) eingegeben werden.

Berechnete Kp-, Ti- und Td-Koeffizienten

Wenn der AT-Prozess abgeschlossen ist:

- werden die berechneten Kp-, Ti- und Td-Koeffizienten in den jeweiligen Speicherwörtern (%MWx) gespeichert und
- können nur im TwidoSoft-Online-Modus auf der Registerkarte **Animation** eingesehen werden.

Registerkarte "Ausgang" des PID

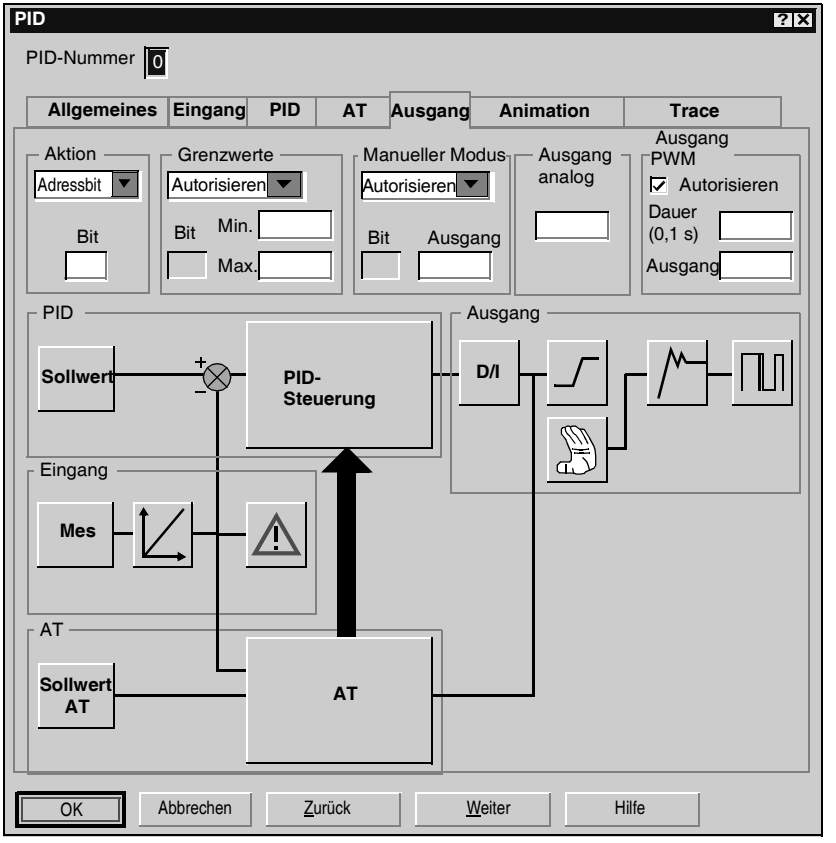
Auf einen Blick

Diese Registerkarte ermöglicht die Festlegung der Ausgangsparameter des PID.

Hinweis: Diese Registerkarte ist im Offline-Modus zugänglich.

Registerkarte "Ausgang" des PID

Das folgende Fenster ermöglicht die Angabe der internen Parameter des PID.

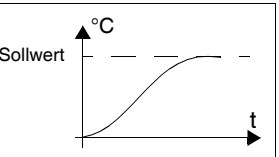


Beschreibung In der folgenden Tabelle sind die Parameter beschrieben, die Sie definieren können.

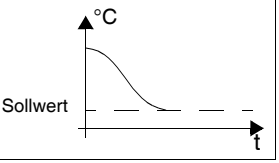
Feld	Beschreibung
PID-Nummer	Geben Sie hier die Nummer des PID ein, den Sie konfigurieren möchten. Der Wert liegt zwischen 0 und 13; pro Applikation sind maximal 14 PID zulässig.
Aktion	Geben Sie hier den Typ der auf den Prozess wirkenden Aktion des PID an. Drei Optionen sind verfügbar: **Invers**, **Direkt** oder **Bitadresse**. Wenn Sie **Bitadresse** ausgewählt haben, können Sie diesen Typ per Programm ändern, indem Sie das zugehörige Bit ändern, bei dem es sich entweder um ein internes Bit (%M0 bis %M255) oder um eine Eingang (%Ix.0 bis %Ix.32) handelt. Die Aktion ist direkt, wenn das Bit auf 1 gesetzt ist; andernfalls ist sie invers. **Hinweis:** Wenn AT aktiviert ist, bestimmt der Auto-Tuning-Algorithmus automatisch den richtigen direkten oder inversen Aktionstyp für den Steuerungsprozess. In diesem Fall ist nur eine Option in der Aktions-Dropdown-Liste verfügbar: **Adressbit.** Sie müssen dann in das entsprechende Textfeld **Bit** ein **internes Wort** (%MW0 bis %MW2999) eingeben. Geben Sie keine interne Konstante oder einen Direktwert in das Textfeld **Bit** ein, da dies zu einem Ausführungsfehler führt.
Grenzwerte Bit	Geben Sie hier den Aktionstyp des PID auf den Prozess an. Drei Optionen sind verfügbar: **Freigeben**, **Sperren** oder **Bitadresse**. Wenn Sie **Bitadresse** ausgewählt haben, können Sie die Verwaltung der Grenzwerte per Programm freigeben (Bit auf 1) oder sperren (Bit auf 0), indem Sie das zugehörige Bit ändern, bei dem es sich entweder um ein internes Bit (%M0 bis %M255) oder um eine Eingang (%Ix.0 bis %Ix.32) handelt.
Min. **Max.**	Setzen Sie hier den oberen und unteren Grenzwert für den PID-Ausgang. **Hinweis**: **Min.** muss kleiner als **Max.** sein. **Min.** oder **Max.** können entweder interne Wörter (%MW0 bis %MW2999) oder interne Konstanten (%KW0 bis %KW255) oder ein Wert zwischen 1 und 10000 sein.
Manueller Modus Bit **Ausgang**	Geben Sie hier an, ob Sie den PID in den manuellen Modus versetzen möchten oder nicht. Drei Optionen sind verfügbar: **Freigeben**, **Sperren** oder **Bitadresse**. Wenn Sie **Bitadresse** ausgewählt haben, können Sie per Programm in den manuellen Modus (Bit auf 1) oder in den automatischen Modus (Bit auf 0) wechseln, indem Sie das zugehörige Bit ändern, bei dem es sich entweder um ein internes Bit (%M0 bis %M255) oder um eine Eingang (%Ix.0 bis %Ix.32) handelt. Der **Ausgang** des manuellen Modus muss den Wert enthalten, den Sie dem Analogausgang zuweisen möchten, wenn sich das PID im manuellen Modus befindet. Dieser **Ausgang** kann entweder ein Wort (%MW0 bis %MW2999) oder ein direkter Wert im Format [0-10000] sein.
Analogausgang	Geben Sie hier den Ausgang des PID im Automatikmodus an. Dieser **Analogausgang** kann vom Typ %MW (%MW0 bis %MW2999) oder %QW (%QWx.0) sein.

Feld	Beschreibung
Zulässiger PWM-Ausgang **Dauer (0,1 s)** **Ausgang**	Aktivieren Sie das Kontrollkästchen, wenn Sie die PWM-Funktion des PID nutzen möchten. Geben Sie die zu modulierende Dauer in **Dauer (0,1 s)** an. Diese Dauer muss zwischen 1 und 500 liegen; sie kann entweder ein internes Wort (%MW0 bis %MW2999) oder eine interne Konstante (%KW0 bis %KW255) sein. Geben Sie das PWM-Ausgangsbit als den Wert in **Ausgang** an. Dieser kann entweder ein internes Bit (%M0 bis %M255) oder ein Ausgang (%Qx.0 bis %Qx.32) sein.
Übersicht	Die Übersicht ermöglicht Ihnen die Anzeige der verschiedenen Möglichkeiten zur Konfiguration Ihres PID.

Hinweis: Der Begriff "Invers" im Feld "Aktion" wird verwendet, um einen oberen Sollwert zu erreichen (Bsp.: zum Heizen)



Der Begriff "Direkt" im Feld "Aktion" wird verwendet, um einen unteren Sollwert zu erreichen (Bsp.: zum Abkühlen)



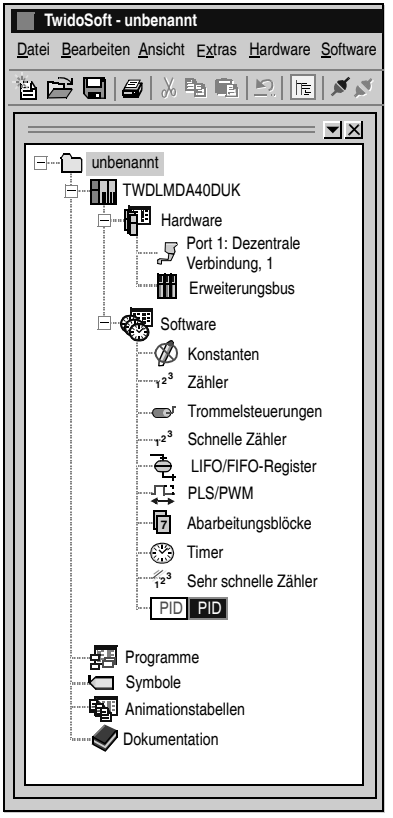
Verfahren zum Zugreifen auf des Debugging des PID

Auf einen Blick

Im Folgenden wird der Zugriff auf die Debug-Fenster eines PID an TWIDO-Steuerungen beschrieben.

Vorgehensweise

Die folgende Tabelle beschreibt die Vorgehensweise zum Zugriff auf die Debug-Fenster eines PID:

Schritt	Aktion
1	Vergewissern Sie sich, dass Sie sich im **Online-Modus** befinden.
2	Öffnen Sie den Navigator. **Ergebnis:** 

Schritt	Aktion
3	Doppelklicken Sie auf **PID**. **Ergebnis:** Das Parametrierungsfenster des PID wird angezeigt. Standardmäßig erscheint es auf der Registerkarte **Animation** (Siehe *Registerkarte "Animation" des PID, S. 561*). **Hinweis**: Sie können auch mit der rechten Maustaste auf **PID** klicken und die Option **Bearbeiten** wählen oder das Menü **Software** → **PID** auswählen oder das Menü **Programm** → **Konfigurationseditor** → **PID-Symbol** verwenden oder im letzten Fall den PID auswählen und auf das Symbol **Lupe** klicken, um einen bestimmten PID auszuwählen.

Registerkarte "Animation" des PID

Auf einen Blick

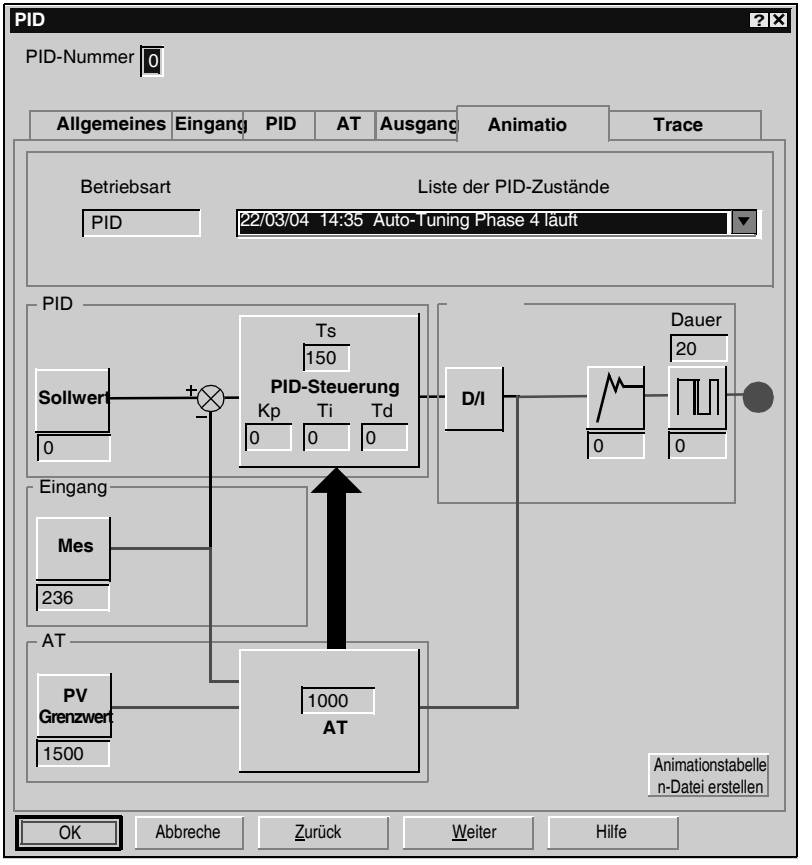
Diese Registerkarte ermöglicht des Debuggen des PID.
Die angezeigte Übersicht ist abhängig von dem von Ihnen erstellen PID-Typ. Es werden nur die konfigurierten Elemente angezeigt.

Die Anzeige ist dynamisch. Die aktiven Verbindungen erscheinen rot, die inaktiven Verbindungen erscheinen schwarz.

Hinweis: Diese Registerkarte ist im Online-Modus zugänglich.

Registerkarte "Animation" des PID

Das folgende Fenster ermöglicht das Anzeigen und Debuggen des PID.



Beschreibung Die folgende Tabelle beschreibt die verschiedenen Bereiche des Fensters.

Feld	Beschreibung
PID-Nummer	Geben Sie hier die Nummer des PID ein, den Sie debuggen möchten. Der Wert liegt zwischen 0 und 13; pro Applikation sind maximal 14 PID zulässig.
Betriebsmodus	In diesem Feld wird der aktuelle Betriebsmodus des PID angezeigt.
Liste der PID-Zustände	Diese Dropdown-Liste ermöglicht die Anzeige der 15 letzten Zustände des PID in Echtzeit. Jede Zustandsänderung aktualisiert die Liste, indem das Datum und die Uhrzeit sowie der aktuelle Zustand angegeben werden.
Animationstabellen-Datei erstellen	Klicken Sie auf die Schaltfläche **Animationstabellen-Datei erstellen**, um eine Datei mit allen in der Übersicht angezeigten Variablen zu erstellen, diese online ändern zu können und Ihren PID zu debuggen.

Registerkarte "Trace" des PID

Auf einen Blick

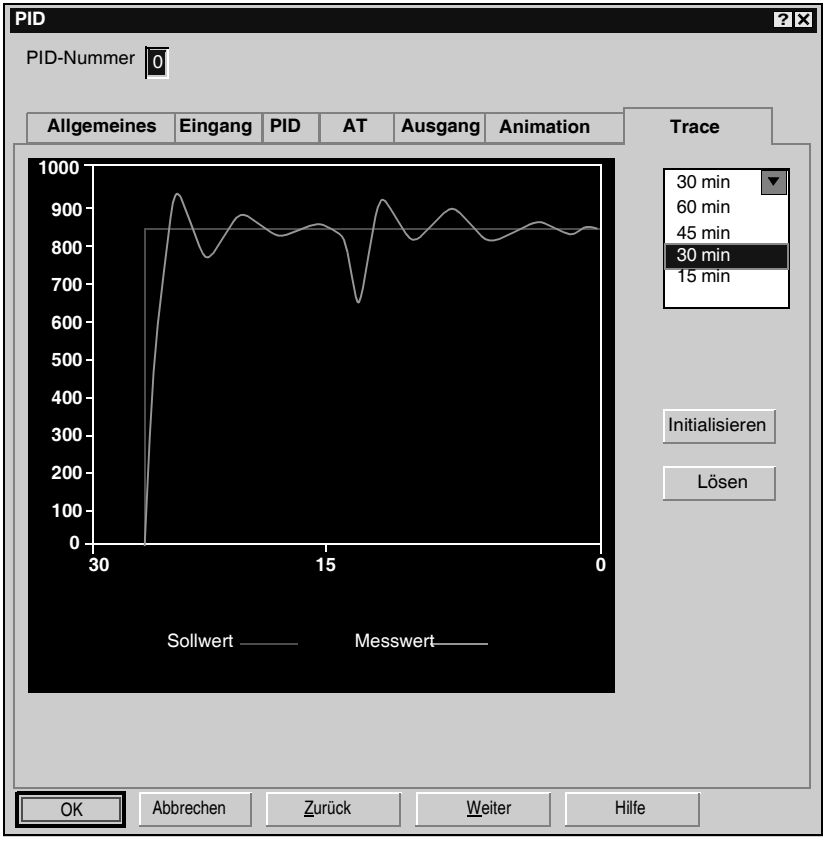
Diese Registerkarte ermöglicht das Anzeigen der Funktion des PID und das Regeln seines Verhaltens.

Der Kurvenverlauf beginnt mit der Anzeige des Debug-Fensters.

Hinweis: Diese Registerkarte ist im Online-Modus zugänglich.

Registerkarte "Animation" des PID

Das folgende Fenster ermöglicht das Anzeigen der Regelung des PID.



Beschreibung Die folgende Tabelle beschreibt die verschiedenen Bereiche des Fensters.

Feld	Beschreibung
PID-Nummer	Geben Sie hier die Nummer des PID ein, den Sie anzeigen möchten. Der Wert liegt zwischen 0 und 13; pro Applikation sind maximal 14 PID zulässig.
Diagramm	In diesem Bereich werden die Kurven des **Sollwerts** und des **Messwerts** angezeigt. Die Skala der horizontalen Achse (X) wird durch das Menü oben rechts festgelegt. Die Skala der vertikalen Achse wird durch die Konfigurationswerte des PID-Eingangs (mit oder ohne Konvertierung) festgelegt. Sie wird automatisch optimiert, um die Kurven so gut wie möglich anzuzeigen.
Menü für Skala der horizontalen Achse	Mit Hilfe dieses Menüs kann die Skala der horizontalen Achse geändert werden. Sie können zwischen vier Werten wählen: 15, 30, 45 oder 60 Minuten.
Initialisieren	Diese Schaltfläche bewirkt das Löschen der Kurve und den Neustart der Anzeige der Verläufe.

PID-Zustände und Fehlercodes

Auf einen Blick

Zusätzlich zur **Liste der PID-Zustände**, die im Dialogfeld **Animation** verfügbar ist (siehe *Registerkarte "Animation" des PID, S. 561*) und die eine Anzeige von bzw. die Umschaltung zu einem der 15 letzten PID-Zustände ermöglicht, kann die Twido-PID-Steuerung auch den aktuellen Zustand des PID-Reglers und des AT-Prozesses in einem benutzerdefinierten Speicherwort speichern.
Informationen über die Aktivierung und Konfiguration des **PID-Status-Speicherworts** (%MWi) finden Sie unter *Registerkarte "Allgemein" des PID, S. 544*.

PID-Status-Speicherwort

Das PID-Status-Speicherwort kann jede beliebige der drei folgenden PID-Informationstypen speichern:

- Aktueller Status des PID-Reglers (PID-Status)
- Aktueller Status des Auto-Tuning-Prozesses (AT-Status)
- PID- und AT-Fehlercodes

Hinweis: Das PID-Status-Speicherwort ist schreibgeschützt.

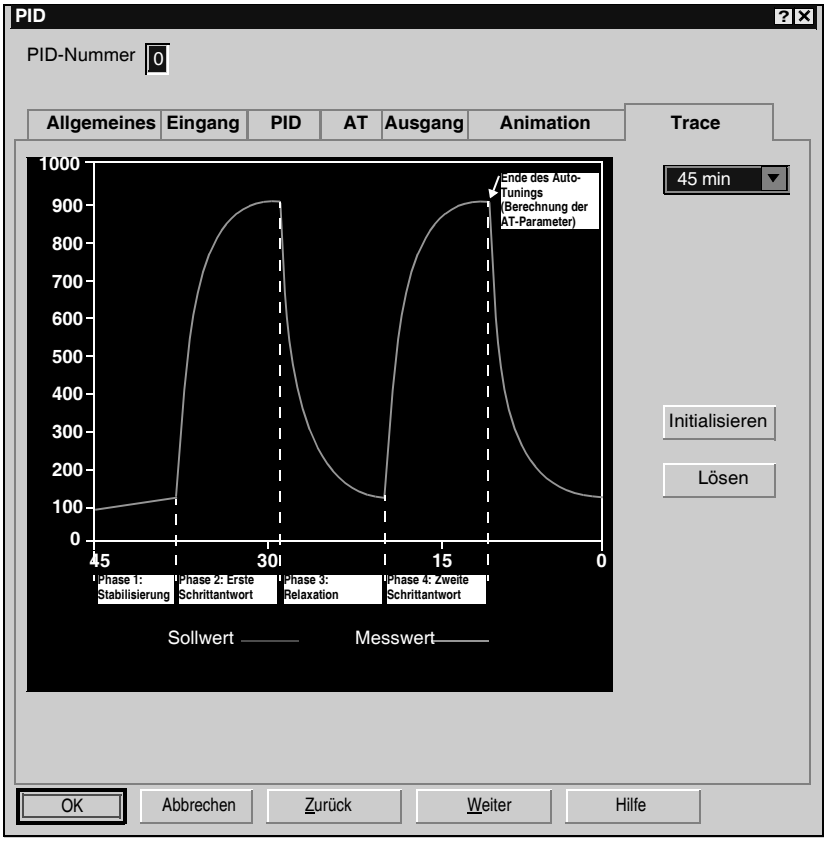
PID-Status-Speicherwort

In der folgenden Tabelle ist die Konkordanz des PID-Reglerstatus mit dem hexadezimal codierten Speicherwort aufgeführt.

Hexadezimale Schreibweise des PID-Status	Beschreibung
0000h	PID-Regelung nicht aktiv
2000h	PID-Regelung läuft
4000h	PID-Sollwert erreicht

Beschreibung des AT-Status

Der Auto-Tuning-Prozess ist in 4 aufeinander folgende Phase unterteilt. Jede Phase des Prozesses muss erfüllt sein, damit der Auto-Tuning-Prozess erfolgreich abgeschlossen werden kann. Die folgende Prozessansprechkurve und -tabelle beschreibt die 4 Phasen des Twido-PID-Auto-Tunings.



In der folgenden Tabelle sind die Auto-Tuning-Phasen beschrieben.

AT-Phase	Beschreibung
1	**Phase 1** ist die Stabilisierungsphase. Sie beginnt, wenn der Benutzer den AT-Prozess startet. Während dieser Phase führt die Auto-Tuning-Funktion der Twido-Steuerung Überprüfungen durch, um zu gewährleisten, dass sich die Prozessvariable in einem gleichbleibenden Zustand befindet. **Hinweis:** Der zuletzt vor Beginn des Auto-Tuning auf den Prozess angewandte Ausgang wird sowohl als Startpunkt als auch als Relaxationspunkt für den Auto-Tuning-Prozess verwendet.
2	**Phase 2** wendet die erste Schrittänderung auf den Prozess an. Sie erzeugt eine Prozessschrittantwort, die mit der in der Abbildung oben gezeigten Antwort identisch ist.

AT-Phase	Beschreibung
3	**Phase 3** ist die Relaxationsphase, die beginnt, wenn sich die erste Schrittantwort stabilisiert hat. **Hinweis:** Die Relaxation tritt in der Nähe des Gleichgewichts auf, das als der zuletzt vor dem Start des Auto-Tunings auf den Prozess angewandte Ausgang definiert ist.
4	**Phase 4** wendet in gleichem Maß und auf gleiche Weise wie in der oben beschriebenen Phase 2 die zweite Schrittänderung auf den Prozess an. Der Auto-Tuning-Prozess endet, und die AT-Parameter werden berechnet und nach Abschluss der Phase 4 in ihren jeweiligen Speicherwörtern gespeichert. **Hinweis:** Nach Abschluss dieser Phase wird die Prozessvariable wieder den Ausgangspegel gesetzt, der zuletzt vor Beginn des Auto-Tuning auf den Prozess angewandt wurde.

AT-Status-Speicherwort

In der folgenden Tabelle ist die Konkordanz des PID-Reglerstatus mit dem hexadezimal codierten Speicherwort aufgeführt.

Hexadezimale Schreibweise des AT-Status	Beschreibung
0100h	Auto-Tuning-Phase 1 läuft
0200h	Auto-Tuning-Phase 2 läuft
0400h	Auto-Tuning-Phase 3 läuft
0800h	Auto-Tuning-Phase 4 läuft
1000h	Auto-Tuning-Prozess abgeschlossen

PID- und AT-Fehlercodes

Die folgende Tabelle beschreibt die möglichen Ausführungsfehler, die während der PID-Regelungs- und Auto-Tuning-Prozesse auftreten können.

PID/AT-Prozesse	Fehlercode (hexadezimal)	Beschreibung
PID-Fehler	8001h	Betriebsmoduswert außerhalb des zulässigen Bereichs
	8002h	Lineare Konvertierung min. und max. gleich
	8003h	Oberer Grenzwert für den Digitalausgang geringer als der untere Grenzwert
	8004h	Prozessvariablengrenzwert außerhalb des linearen Konvertierungsbereichs
	8005h	Prozessvariablengrenzwert geringer als 0 oder größer als 10000
	8006h	Sollwert außerhalb des linearen Konvertierungsbereichs
	8007h	Sollwert geringer als 0 oder größter als 10000
	8008h	Steuerungsaktion nicht mit der beim AT-Start festgelegten Aktion identisch

PID/AT-Prozesse	Fehlercode (hexadezimal)	Beschreibung
Auto-Tuning-Fehler	8009h	Auto-Tuning-Fehler: Der Prozessvariablengrenzwert wurde erreicht
	800Ah	Auto-Tuning-Fehler: entweder Überabtastung oder zu geringer Ausgangs-Sollwert
	800Bh	Auto-Tuning-Fehler: Kp gleich Null
	800Ch	Auto-Tuning-Fehler: Die Zeitkonstante ist negativ.
	800Dh	Auto-Tuning-Fehler: Verzögerung ist negativ
	800Eh	Auto-Tuning-Fehler: Fehler bei der Berechnung von Kp
	800Fh	Auto-Tuning-Fehler: Zeitkonstante über Verzögerungsverhältnis > 20
	8010h	Auto-Tuning-Fehler: Zeitkonstante über Verzögerungsverhältnis < 2
	8011h	Auto-Tuning-Fehler: Der Grenzwert für Kp wurde überschritten.
	8012h	Auto-Tuning-Fehler: Der Grenzwert für Ti wurde überschritten.
	8013h	Auto-Tuning-Fehler: Der Grenzwert für Td wurde überschritten.

PID-Einstellung mittels Auto-Tuning (AT)

Überblick über die PID-Einstellung

Die PID-Steuerungsfunktion beruht auf den drei folgenden benutzerdefinierten Parametern: Kp, Ti und Td. Die PID-Einstellung hat die präzise Festlegung dieser Prozessparameter zum Ziel, um eine optimale Steuerung des Prozesses zu gewährleisten.

Umfang des Auto-Tuning

Die AT-Funktion der Twido-SPS ist besonders für die automatische Einstellung von thermischen Prozessen geeignet. Da die Werte der PID-Parameter von einem Steuerungsprozess zum anderen erheblich variieren können, kann Sie die Auto-Tuning-Funktion der Twido-SPS mit geringem Aufwand bei der Ermittlung von präziseren Werten als einfach geschätzten Werten unterstützen.

Auto-Tuning-Anforderungen

Vergewissern Sie sich bei der Verwendung der Auto-Tuning-Funktion, dass der Steuerungsprozess und die Twido-SPS allen vier nachfolgend aufgeführten Anforderungen entsprechen:

- Der Steuerungsprozess muss ein rückführungsfreies, stabiles System sein.
- Zu Beginn des Auto-Tunings muss sich der Steuerungsprozess in einem stationären Betriebszustand mit einem Prozesseingang gleich Null befinden (z.B.: Ein Ofen oder ein Kessel muss sich bei Umgebungstemperatur befinden.)
- Stellen Sie sicher, dass während des Auto-Tuning-Vorgangs keine Störungen des Prozesses auftreten können, die zu falsch berechneten Parametern oder zum Fehlschlagen des Auto-Tuning-Vorgangs führen können (z.B.: Die Ofentür darf noch nicht einmal momentan geöffnet werden.)
- Konfigurieren Sie die Twido-SPS so, dass Sie im **Periodischen Modus** abgetastet wird. Wenn Sie die richtige Abtastzeit (Ts) für das Auto-Tuning ermittelt haben, muss die Zykluszeit so konfiguriert werden, dass die Abtastzeit (Ts) ein exaktes Vielfaches der Zykluszeit der Twido-SPS ist.

Hinweis: Um eine ordnungsgemäße Ausführung der PID-Steuerung und des Auto-Tuning-Vorgangs zu gewährleisten, ist es von entscheidender Bedeutung, dass die Twido-SPS so konfiguriert werden kann, dass sie die Zyklen im periodischen (und nicht im zyklischen) Modus ausführt. Im periodischen Modus beginnt jeder Zyklus der SPS in regelmäßigen Zeitintervallen. Dies gewährleistet, dass die Abtastrate während der gesamten Messdauer konstant ist (im Gegensatz zum zyklischen Modus, in dem ein Zyklus startet, sobald der vorherige Zyklus abgeschlossen ist, wodurch die Abtastzeit von einem Zyklus zum nächsten variiert).

AT-Betriebsmodus

Auto-Tuning kann entweder autonom (AT-Modus) oder in Verbindung mit der PID-Steuerung (AT + PID) verwendet werden:

- **AT-Modus:** Nach der Konvergenz des AT-Vorgangs und der erfolgreichen Ermittlung mit der Festlegung der PID-Steuerungsparameter Kp, Ti und Td (oder nach Erkennung eines Fehlers im AT-Algorithmus) wird der numerische AT-Ausgang auf 0 gesetzt, und die folgende Meldung wird im Dropdown-Listenfeld **Liste der PID-Zustände** angezeigt: "Auto-Tuning abgeschlossen."
- **AT + PID-Modus:** Als erstes wird AT gestartet. Nach erfolgreicher Beendigung des AT wird der PID-Regelkreis gestartet (auf der Grundlage der durch das AT berechneten Parameter Kp, TI und Td).
  Hinweis zu AT+PID: Wenn der AT-Algorithmus auf einen Fehler trifft:
  - wird kein PID-Parameter berechnet;
  - wird der numerische AT-Ausgang auf den Wert gesetzt, der zuletzt vor Beginn des Auto-Tuning auf den Prozess angewandt wurde;
  - wird eine Fehlermeldung im Dropdown-Listenfeld "Liste der PID-Zustände" angezeigt;
  - wird die PID-Steuerung abgebrochen.

Hinweis: Stoßfreie Transition
Im **Modus AT+PID** ist die Transition von AT zu PID stoßfrei.

Methoden zur Ermittlung der Abtastzeit (Ts)

Wie in den folgenden beiden Abschnitten (siehe *Anhang 1: Grundlegendes zur PID-Theorie, S. 584* und *Anhang 2: Modell erster Ordnung mit Zeitverzögerung, S. 586*) erklärt wird, ist die **Abtastzeit (Ts)** ein entscheidender Parameter der PID-Steuerung. Die Abtastzeit kann von der AT-**Zeitkonstante (τ) abgeleitet werden.** Es gibt zwei Methoden zur Ermittlung der korrekten Abtastzeit (Ts) mittels der Auto-Tuning-Funktion. Diese Methoden werden in den folgenden Abschnitten beschrieben.

- Die Prozessansprechkurven-Methode
- Die Versuch-und-Irrtum-Methode

Beide Methoden werden in den folgenden Teilabschnitten beschrieben.

Beschreibung der Prozessan-sprechkurven-Methode

Dieses Methode besteht im Setzen einer Schrittänderung am Steuerungsprozes-seingang und der Aufzeichnung der Prozessausgangskurve mit Zeit.
Die Prozessansprechkurven-Methode geht von der folgenden Voraussetzung aus:

- Der Steuerungsprozess kann durch die folgende Transferfunktion adäquat als ein Modell erster Ordnung mit Zeitverzögerung beschrieben werden:

$$\frac{S}{U} = \frac{k}{1 + \tau p} \cdot e^{-\theta p}$$

(Für ausführliche Informationen siehe Anhang 2: Modell erster Ordnung mit Zeitverzögerung)

Verwendung der Prozessan-sprechkurven-Methode

Gehen Sie wie nachfolgend beschrieben vor, um die Abtastzeit (Ts) mittels der Prozessansprechkurven-Methode zu ermitteln.

Schritt	Aktion
1	Es wird davon ausgegangen, dass Sie bereits die verschiedenen Einstellungen auf den Registerkarten "Allgemein", "Eingang", "PID", "AT" und "Ausgang" der PID-Steuerung konfiguriert haben.
2	Wählen Sie im Applikationsnavigator die Registerkarte **PID > Ausgang** aus.
3	Wählen Sie aus dem Dropdown-Listenfeld **Manueller Modus** die Option **Autorisieren** oder **Adressbit** aus, um einen manuellen Ausgang zu ermöglichen, und setzen Sie das Feld **Ausgang** auf einen hohen Level (im Bereich [5000-10000]).
4	Wählen Sie aus der Menüleiste die Option **SPS > Übertragung PC => SPS...** aus, um das Applikationsprogramm in die Twido-SPS zu laden.
5	Schalten Sie im PID-Konfigurationsfenster in den Modus **Trace**.
6	Starten Sie die PID-Steuerung und überprüfen Sie den Anstieg der Ansprechkurve.
7	Beenden Sie die PID-Messung, wenn die Ansprechkurve einen gleich bleibenden Status erreicht hat. **Hinweis:** Lassen Sie das "Trace"-Fenster der PID-Steuerung aktiv.
8	Verwenden Sie die folgende graphische Methode, um die Zeitkonstante (τ) des Steuerungsprozesses zu bestimmen: **1.** Berechnen Sie mittels der folgenden Formel den Prozessvariablenausgang bei 63% Steigung ($S_{[63\%]}$): $S_{[63\%]} = S_{[initial]} + (S_{[ending]}-S_{[initial]})x63\%$ **2.** Ermitteln Sie graphisch die Zeitabszisse ($t_{[63\%]}$), die S(63%) entspricht. **3.** Ermitteln Sie graphisch die Initialzeit ($t_{[initial]}$), die dem Start des Anstiegs der Prozessansprechkurve entspricht. **4.** Berechnen Sie mittels der folgenden Beziehung die Zeitkonstante (τ) des Steuerungsprozesses: $\tau = t_{[63\%]}-t_{[initial]}$
9	Berechnen Sie mittels der folgenden Formel die Abtastzeit (Ts) auf der Grundlage des Werts von (τ), den Sie zuvor im vorigen Schritt ermittelt haben: $Ts = \tau/75$ **Hinweis:** Die Grundeinheit für die Abtastzeit ist 10 ms. Daher müssen Sie den Wert von Ts auf die nächsten vollen 10 ms auf- bzw. abrunden.
10	Wählen Sie **Programm > Abtastmodus bearbeiten** aus und gehen Sie folgendermaßen vor: **1.** Setzen Sie den **Abtastmodus** der Twido-SPS auf **Periodisch**. **2.** Stellen Sie die **Zykluszeit** mittels der folgenden Formel so ein, dass die Abtastzeit (Ts) ein **exaktes Vielfaches** der Zykluszeit ist: Zykluszeit = Ts / n, wobei "n" eine positive Ganzzahl ist. **Hinweis:** Sie müssen "n" so wählen, dass die resultierende Zykluszeit eine positive Ganzzahl im Bereich von [2 - 150 ms] ist.

Beispiel für die Prozessan-sprechkurve

Dieses Beispiel zeigt Ihnen, wie die Zeitkonstante (τ) eines einfachen thermischen Prozesses mittels der im vorigen Teilabschnitt beschriebenen Prozessansprech-kurven-Methode gemessen wird.

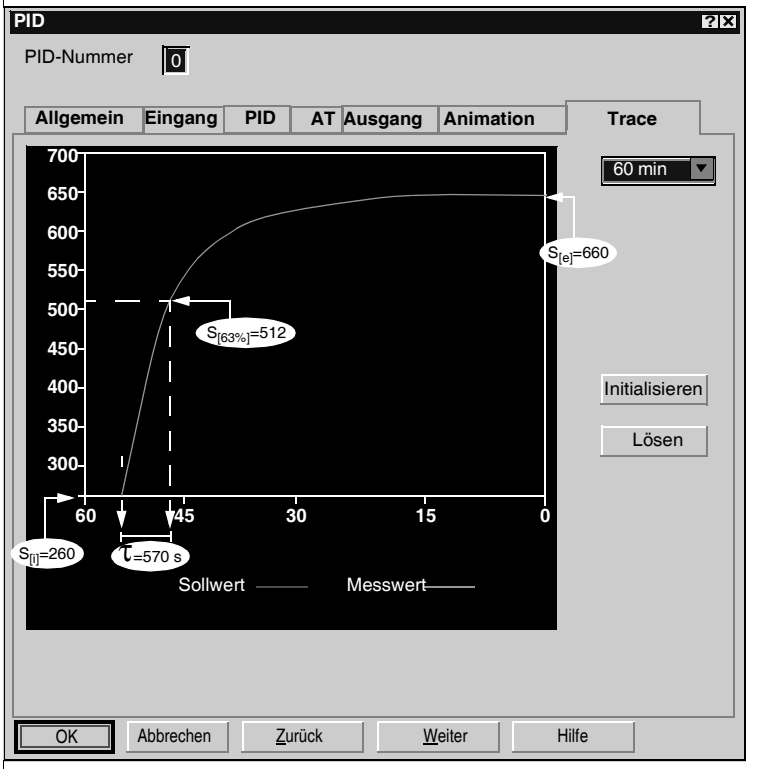
Das experimentelle Setup für die Zeitkonstantenmessung lautet wie folgt:

- Der Steuerungsprozess besteht aus einem Umluftofen, der mit einer Heizlampe ausgestattet ist.
- Die Temperaturmessungen werden von der Twido-SPS über einen Pt100-Sensor erfasst, und die Temperaturdaten werden in °C gespeichert.
- Die Twido-SPS steuert die Heizlampe über den diskreten PWM-Ausgang des PID-Reglers.

Das Experiment wird folgendermaßen durchgeführt:

Schritt	Aktion
1	Im PID-Konfigurationsbildschirm wird die PID-Registerkarte **Ausgang** ausgewählt.
2	**Manueller Modus** ist auf der Registerkarte "Ausgang" ausgewählt.
3	Der **Ausgang** im manuellen Modus ist auf 10000 gesetzt.
4	Die PID-Ausführung ist von der PID-Registerkarte **Trace gestartet.**
5	Die PID-Ausführung wird gestoppt, wenn die Ofentemperatur einen gleich bleibenden Status erreicht hat.

Schritt	Aktion
6	Die folgenden Informationen werden direkt wie in folgender Abbildung dargestellt durch die graphische Analyse der Prozessansprechkurve ermittelt.



wobei

- $S_{[i]}$ = Initialwert der Prozessvariable = 260
- $S_{[e]}$ = Endwert der Prozessvariable = 660
- $S_{[63\%]}$ = Prozessvariable bei 63% Anstieg = $S_{[i]} + (S_{[i]} - S_{[e]}) \times 63\%$
  = 260+(660-260)x63%
  = 512
- τ = Zeitkonstante
  = verstrichene Zeit vom Start des Anstiegs, bis $S_{[63\%]}$ erreicht ist
  = 9 min 30 s = 570 s

7	Die Abtastzeit (Ts) wird mittels folgender Beziehung ermittelt: $Ts = \tau/75$ = 570/75 = 7,6 s (7600 ms)
8	Im Dialogfeld **Programm > Abtastmodus bearbeiten** muss die **Zykluszeit** so gesetzt sein, dass die Abtastzeit (Ts) ein exaktes Vielfaches der Zykluszeit ist (wie in folgendem Beispiel): Zykluszeit = Ts/76 = 7600/76 = 100 ms (was die folgende Bedingung erfüllt: 2 ms ≤ Zykluszeit ≤ 150 ms.)

Versuch-und-Irrtum-Methode

Die Versuch-und-Irrtum-Methode besteht darin, eine Reihe von Schätzungen der Abtastzeit in die Auto-Tuning-Funktion einzugeben, bis der Auto-Tuning-Algorithmus erfolgreich zu Kp-, Ti- und Td-Werten konvergiert, die vom Benutzer als zufrieden stellend beurteilt werden.

Hinweis: Im Gegensatz zur Prozessansprechkurven-Methode basiert die Versuch-und-Irrtum-Methode nicht auf irgendeiner auf die Prozessreaktion angewandten Näherungsregel. Sie hat jedoch den Vorteil, dass der Wert zu einem Wert der Abtastzeit konvergiert, der sich in derselben Größenordnung wie der aktuelle Wert befindet.

Befolgen Sie die nachfolgend aufgeführten Schritte, um eine Versuch-und-Irrtum-Schätzung der Auto-Tuning-Parameter durchzuführen:

Schritt	Aktion
1	Wählen Sie im PID-Konfigurationsbildschirm die Registerkarte **AT** aus.
2	Setzen Sie die **Ausgangsbegrenzung** des AT auf **10000.**
3	Wählen Sie im PID-Konfigurationsbildschirm die Registerkarte **PID** aus.
4	Geben Sie die erste oder n-te Schätzung in das Feld **Abtastzeit** ein. **Hinweis:** Wenn Sie über keine erste Ahnung des möglichen Bereichs für die Abtastzeit verfügen, setzen Sie diesen Wert auf den minimal möglichen Wert: 1 (1 Einheit à 10 ms).
5	Wählen Sie aus der Menüleiste die Option **SPS > Übertragung PC => SPS...** aus, um das Applikationsprogramm in die Twido-SPS zu laden.
6	Starten Sie **Auto-Tuning.**
7	Wählen Sie im PID-Konfigurationsbildschirm die Registerkarte **Animation** aus.
8	Warten Sie, bis der Auto-Tuning-Vorgang abgeschlossen ist.
9	Zwei Fälle sind möglich: • **Auto-Tuning wird erfolgreich abgeschlossen:** Sie können den Vorgang mit Schritt 9 fortsetzen. • **Auto-Tuning schlägt fehl:** Das bedeutet, dass die aktuelle Schätzung der Abtastzeit (Ts) falsch ist. Führen Sie eine neue Schätzung von Ts durch und wiederholen Sie die Schritte 3 bis 8 so oft wie erforderlich, bis der Auto-Tuning-Vorgang möglicherweise konvergiert. Befolgen Sie diese Richtlinien, um eine neue Schätzung von Ts durchzuführen: ◦ AT endet mit der Fehlermeldung **"Die berechnete Zeitkonstante ist negativ!":** Das bedeutet, dass die Abtastzeit **Ts zu groß ist.** Sie müssen den Wert von Ts verringern, um eine neue Schätzung abzugeben. ◦ AT endet mit der Fehlermeldung **"Abtastfehler!":** Das bedeutet, dass die Abtastzeit **Ts zu gering ist.** Sie müssen den Wert von Ts erhöhen, um eine neue Schätzung abzugeben.
10	Sie können jetzt die PID-Steuerungsparameter (Kp, Ti und Td) auf der Registerkarte Animation sehen und sie dann im PID-Konfigurationsbildschirm wie gewünscht auf der Registerkarte "PID" anpassen. **Hinweis:** Wenn die von diesen Steuerungsparametern gelieferte PID-Einstellung nicht vollständig zufrieden stellende Ergebnisse liefert, können Sie die Versuch-und-Irrtum-Schätzung der Abtastzeit noch weiter verfeinern, bis Sie die richtigen Kp-, Ti- und Td-Steuerungsparameter erhalten.

Anpassen der PID-Parameter

Um die von den PID-Parametern (Kp, Ti, Td) per Auto-Tuning gelieferte Prozessregelung feiner anzupassen, haben Sie auch die Möglichkeit, diese Parameterwerte direkt ausgehend von der Registerkarte "PID" des PID-Konfigurationsbildschirms oder über die entsprechenden Speicherwörter (%MW) manuell anzupassen.

Einschränkungen hinsichtlich der Verwendung von Auto-Tuning und der PID-Steuerung

Auto-Tuning ist ideal für Prozesse, deren Zeitkonstante (τ) und Verzögerungszeit (θ) den folgenden Anforderungen entsprechen: ($\tau + \theta$) < 2700 s (d.h.: 45 min)
Die **PID-Steuerung** ist ideal für die Regelung von Prozessen, die die folgende Bedingung erfüllen: $2 < (\tau/\theta) < 20$, wobei (τ) die Zeitkonstante des Prozesses und (θ) die Verzögerungszeit ist.

Hinweis: Abhängig vom Verhältnis (τ/θ):

- (τ/θ) < 2: Die PID-Regelung hat ihre Grenzwerte erreicht, in diesem Fall sind höher entwickelte Regelungstechniken erforderlich.
- (τ/θ) > 20: In diesem Fall kann ein einfacher Ein-/Aus-Regler (Zweipunktregler) anstelle der PID-Steuerung verwendet werden.

Beseitigung von Fehlern der Auto-Tuning-Funktion

Die folgende Tabelle enthält eine Liste der Auto-Tuning-Fehlermeldungen sowie deren mögliche Ursachen und Maßnahmen zur Behebung der Fehler.

Fehlermeldung	Mögliche Ursache	Erklärung / Mögliche Lösung
Auto-Tuning-Fehler: Der Prozessvariablengrenzwert wurde erreicht	Die Prozessvariable hat den maximal zulässigen Grenzwert erreicht.	Dies ist eine Systemsicherheit. Da das AT ein rückführungsfreier Prozess ist, fungiert der Prozessvariablengrenzwert als ein oberer Grenzwert.
Auto-Tuning-Fehler: entweder Überabtastung oder zu geringer Ausgangs-Sollwert	Eine beliebige der beiden Ursachen: • Abtastzeit zu gering. • AT-Ausgang zu gering eingestellt.	Erhöhen Sie entweder die Abtastzeit oder den Sollwert des AT-Ausgangs.
Auto-Tuning-Fehler: Die Zeitkonstante ist negativ.	Die Abtastzeit ist möglicherweise zu groß.	Ausführliche Informationen finden Sie unter *PID-Einstellung mittels Auto-Tuning (AT), S. 569*.
Auto-Tuning-Fehler: Fehler bei der Berechnung von Kp	Der AT-Algorithmus ist fehlgeschlagen (keine Konvergenz).	Überprüfen Sie die PID- und AT-Parameter und nehmen Sie Anpassungen vor, die zu einer Verbesserung der Konvergenz führen können. Überprüfen Sie auch, dass keine Störung vorliegt, die die Prozessvariable beeinflussen könnte.
Auto-Tuning-Fehler: Zeitkonstante über Verzögerungsverhältnis > 20	$\tau/\theta > 20$	Die PID-Regelung ist nicht länger gewährleistet. Ausführliche Informationen finden Sie unter *PID-Einstellung mittels Auto-Tuning (AT), S. 569*.
Auto-Tuning-Fehler: Zeitkonstante über Verzögerungsverhältnis < 2	$\tau/\theta < 2$	Die PID-Regelung ist nicht länger gewährleistet. Ausführliche Informationen finden Sie unter *PID-Einstellung mittels Auto-Tuning (AT), S. 569*.
Auto-Tuning-Fehler: Der Grenzwert für Kp wurde überschritten.	Der berechnete Wert der statischen Verstärkung (Kp) ist größer als 10000.	Die Messempfindlichkeit einiger Applikationsvariablen ist eventuell zu gering. Der Messbereich der Applikation muss innerhalb des Intervalls [0-10000] neu eingestellt werden.
Auto-Tuning-Fehler: Der Grenzwert für Ti wurde überschritten.	Der berechnete Wert der integralen Zeitkonstante (Ti) ist größer als 20000.	Der rechentechnische Grenzwert wurde erreicht.
Auto-Tuning-Fehler: Der Grenzwert für Td wurde überschritten.	Der berechnete Wert der abgeleiteten Zeitkonstante (Td) ist größer als 10000.	Der rechentechnische Grenzwert wurde erreicht.

PID-Parameter-Anpassungsverfahren

Einleitung

Es gibt zahlreiche Verfahren zur Anpassung der PID-Parameter. Wir empfehlen das Ziegler- und Nichols-Verfahren, von dem es zwei Varianten gibt:

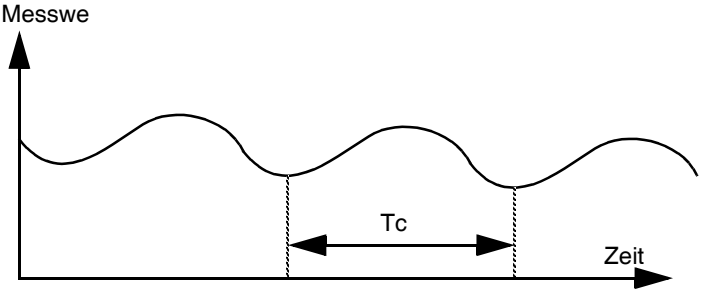
- adaptive Regelung mit Rückführung,
- adaptive Regelung ohne Rückführung.

Vor der Durchführung eines dieser Verfahren müssen Sie die Richtung der PID-Aktion festlegen:

- Wenn eine Erhöhung des OUT-Ausgangs zur Erhöhung der PV-Messung führt, invertieren Sie den PID (KP > 0),
- wenn dies jedoch zu einer PV-Verringerung führt, machen Sie den PID direkt (KP < 0).

Adaptive Regelung mit Rückführung

Das Prinzip besteht in der Verwendung eines proportionalen Befehls (Ti = 0, Td = 0), um den Prozess mit einer Erhöhung der Produktion zu starten, bis diese wiederum oszilliert, nachdem ein Pegel auf den PID-Korrektor-Sollwert angewandt wird. All das ist erforderlich, um den kritischen Produktionslevel (Kpc) zu erhöhen, der dazu geführt hat, dass die nicht gedämpfte Oszillation und die Oszillationsdauer (Tc) die Werte verringert haben, was zu einer optimalen Einstellung des Reglers führt.



Je nach Art des Reglers (PID oder PI), wird die Anpassung der Koeffizienten mit den folgenden Werten ausgeführt:

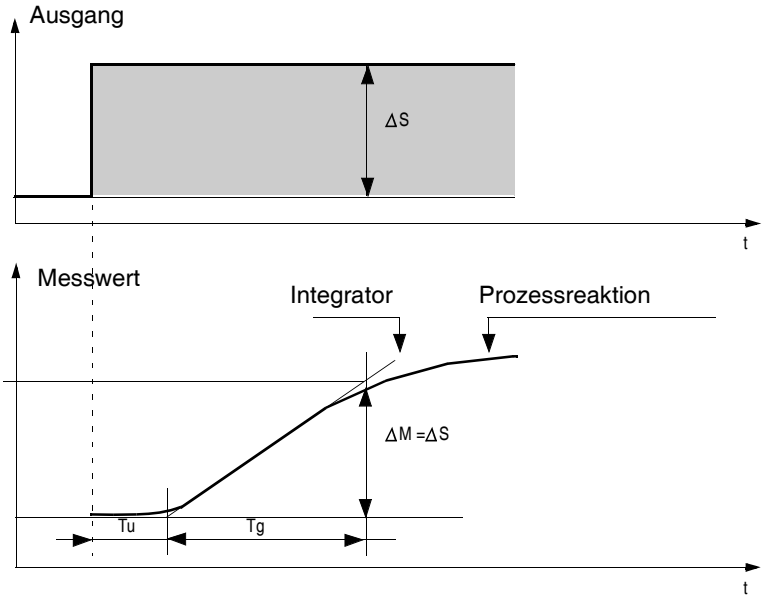
-	Kp	Ti	Td
PID	Kpc/1,7	Tc/2	Tc/8
PI	Kpc/2,22	0,83 x Tc	-

wobei Kp = proportionale Produktion, Ti = Integrationszeit und TD = Umleitungszeit.

Hinweis: Dieses Anpassungsverfahren bietet eine äußerst dynamische Steuerung, was sich in einem unerwünschten Überschwingen während des Wechsels der Sollwertimpulse äußern kann. Verringern Sie in diesem Fall den Produktionswert, bis Sie das gewünschte Verhalten erreichen.

Adaptive Regelung ohne Rückführung

Da sich der Regler im manuellen Modus befindet, legen Sie einen Pegel am Ausgang an und lassen die Prozessantwort genauso starten wie ein Integrator mit reiner Verzögerungszeit.



Der Kreuzungspunkt auf der rechten Seite, der repräsentativ für den Integrator mit der Zeitachse ist, bestimmt die Zeit Tu. Als nächstes wird die Zeit Tg als die für die kontrollierte Variable (Messwert) erforderliche Zeit bestimmt, um dieselbe Variationsgröße (% der Skala) wie den Reglerausgang zu erhalten.
Je nach Art des Reglers (PID oder PI), wird die Anpassung der Koeffizienten mit den folgenden Werten ausgeführt:

-	Kp	Ti	Td
PID	-1,2 Tg/Tu	2 x Tu	0,5 x Tu
PI	-0,9 Tg/Tu	3,3 x Tu	-

wobei Kp = proportionale Produktion, Ti = Integrationszeit und TD = Umleitungszeit.

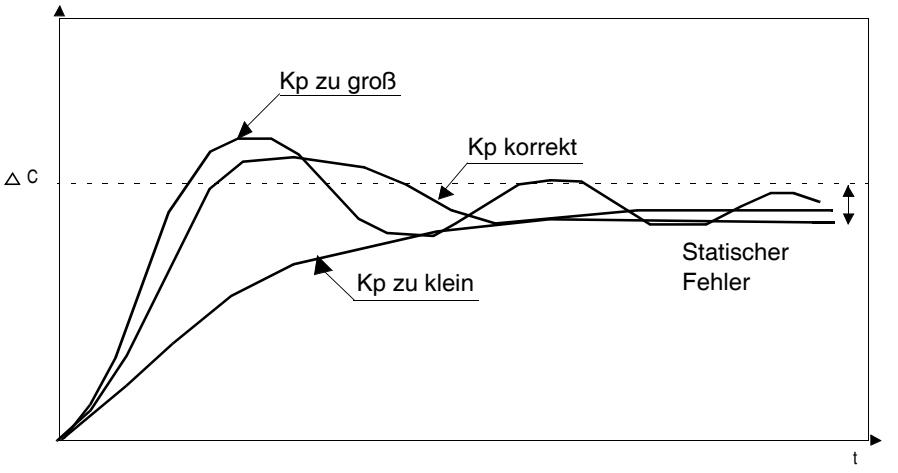
Hinweis: Beachten Sie die Einheiten! Wenn die Anpassung in PL7 ausgeführt wird, so multiplizieren Sie den für KP erhaltenen Wert mit 100.

Dieses Anpassungsverfahren bietet außerdem eine äußerst dynamische Steuerung, was sich in einem unerwünschten Überschwingen während des Wechsels der Sollwertimpulse äußern kann. Verringern Sie in diesem Fall den Produktionswert, bis Sie das gewünschte Verhalten erreichen. Dieses Verfahren ist interessant, da es keine Annahmen über die Art und Weise des Verfahrens erfordert. Sie können es sowohl auf das stabile Verfahren als auch auf reale integrierende Verfahren anwenden. Es ist wirklich interessant im Fall von langsamen Verfahren (Glasindustrie, …), da der Benutzer nur den Beginn der Antwort benötigt, um die Koeffizienten Kp, Ti und Td zu regeln.

Rolle und Einfluss der Paramater eines PID

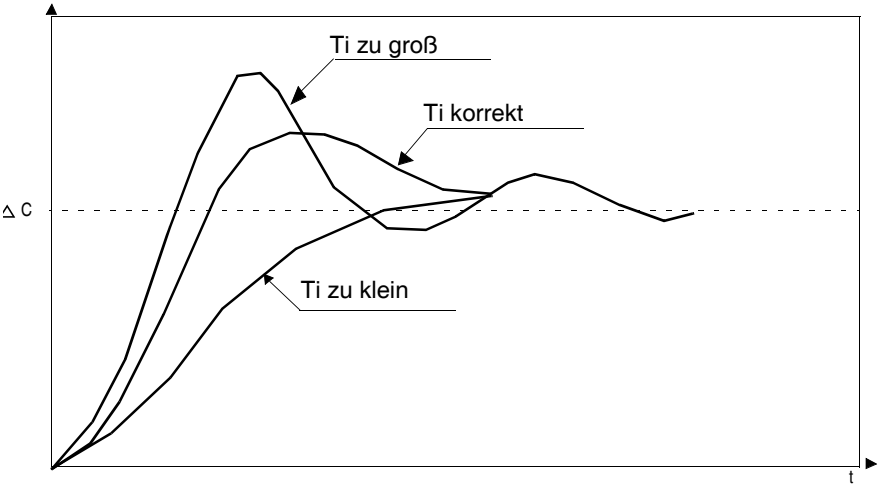
Einfluss der proportionalen Verarbeitung

Mit der proportionalen Verabeitung kann die Antwortgeschwindigkeit des Prozesses beeinflusst werden. Je größer die Verstärkung ist, desto mehr beschleunigt sich die Antwort und desto geringer wird der statische Fehler (bei reiner Proportionalität); doch umso mehr nimmt auch die Stabilität ab. Man muss einen guten Kompromiss zwischen Geschwindigkeit und Stabilität finden. Der Einfluss der proportionalen Verabeitung auf die Antwort des Prozesses auf einer Stufe stellt sich wie folgt dar:



Einfluss der Integralaktion

Mit Hilfe der Integralaktion kann der statische Fehler (Abweichung zwischen Regelgröße und Führungsgröße) annulliert werden. Je höher die Integralaktion ist (Ti klein), desto schneller wird die Antwort und desto mehr nimmt die Stabilität ab. Man muss auch hier einen guten Kompromiss zwischen Geschwindigkeit und Stabilität finden. Der Einfluss der Integralaktion auf die Antwort des Prozesses auf einer Stufe stellt sich wie folgt dar:

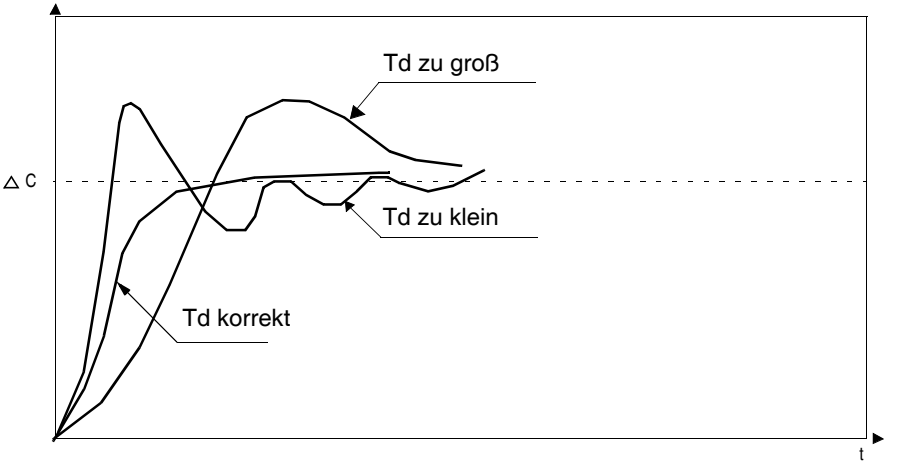


Hinweis: Ein kleines Ti bedeutet, dass die Integralaktion hoch ist.

oder Kp = proportionale Verstärkung, Ti = Integrationszeit und Td = Abzweigungszeit.

Einfluss des D-Verhaltens

Das D-Verhalten ist vorwegnehmend. Es fügt nämlich einen Wert hinzu, der die Variationsgeschwindigkeit der Abweichung berücksichtigt; dies gestattet eine Vorwegnahme durch Beschleunigung der Antwort des Prozesses, wenn die Abweichung größer wird, und eine Verlangsamung, wenn die Abweichung geringer wird. Je höher das D-Verhalten ist (Td groß), desto mehr wird die Reaktion beschleunigt. Auch hier muss man einen guten Kompromiss zwischen Geschwindigkeit und Stabilität finden. Der Einfluss des D-Verhaltens auf die Antwort des Prozesses auf einer Stufe stellt sich wie folgt dar:



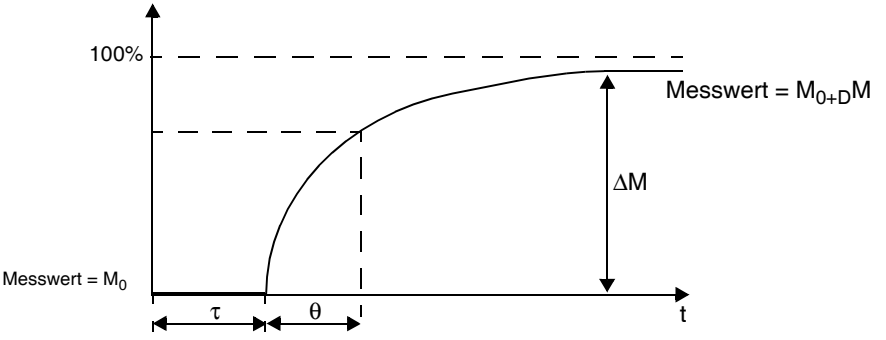
Grenzen der PID-Einstellung

Wenn man den Prozess einem Vorgang erster Ordnung mit reiner Verzögerung durch die Transferfunktion angleicht: $(H(p)) = K\frac{(e^{(-\tau)p})}{(1 + \theta p)}$

mit :

τ = Verzögerung des Modells

θ = Zeitkonstante des Modells



dann hängen die Leistungen der Regelung ab vom Verhältnis $\frac{\tau}{\theta}$

Die PID- Regelung gilt als angemessen im folgenden Bereich:2- $\frac{\tau}{\theta}$ -20

Für $\frac{\tau}{\theta}$ <2, d.h. für schnelle Regelkreise (θ klein) oder für Prozesse mit starker Verzögerung (groß) ist die PID-Regelung nicht mehr geeignet, es müssen höher entwickelte Algorithmen verwendet werden.

Für $\frac{\tau}{\theta}$ > 20 ist eine Regelung mit Schwellwert plus Hysterese ausreichend.

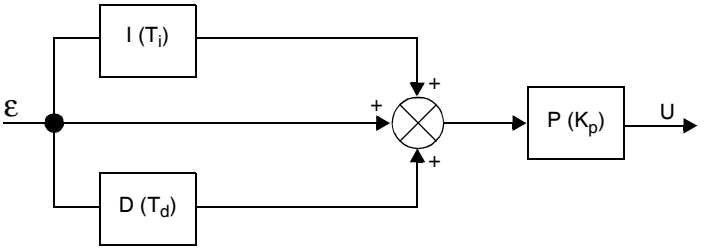
Anhang 1: Grundlegendes zur PID-Theorie

Einführung

Die in allen Twido-Steuerungen integrierte PID-Regelungsfunktion bietet eine effiziente Steuerung für einfache industrielle Prozesse, die aus einem System-Stimulus (in diesem Dokument als Sollwert bezeichnet) und einer messbaren Eigenschaft des Systems (als Messwert oder Prozessvariable bezeichnet) bestehen.

Das PID-Steuerungs-modell

Die Twido-PID-Regelung implementiert eine kombinierte (seriell - parallel) PID-Korrektur (siehe nachfolgende Abbildung eines PID-Modells) über einen analogen Messwert und Sollwert im Format [0-10000] und leitet einen analogen Befehl im selben Format an den gesteuerten Prozess weiter.
Die kombinierte Form des PID-Steuerungsmodells ist in folgender Abbildung beschrieben.



wobei
wobei gilt:

- I = die **integrale** Aktion (die unabhängig und parallel zur abgeleiteten Aktion wirkt),
- D = die **abgeleitete** Aktion (die unabhängig und parallel zur integralen Aktion wirkt),
- P = die **proportionale** Aktion (die seriell auf den kombinierten Ausgang der integralen und abgeleiteten Aktionen wirkt),
- U = der PID-Reglerausgang (der später als Eingang an den gesteuerten Prozess weitergeleitet wird).

Das PID-Regelungsgesetz

Die PID-Regelung besteht aus der Kombination (seriell - parallel) der Regelungsverstärkung (Kp) und den integralen (Ti) und abgeleiteten (Td) Zeitkonstanten. Folglich hat das von der Twido-Steuerung verwendete PID-Regelungsgesetz die folgende Form *(Gleichung 1):*

$$u(i) = K_P \cdot \left\{ \varepsilon(i) + \frac{T_s}{T_i} \sum_{j=1}^{i} \varepsilon(j) + \frac{T_d}{T_s}[\varepsilon(i) - \varepsilon(i-1)] \right\}$$

wobei

- Kp = die proportionale Verstärkung der Steuerung,
- Ti = die integrale Zeitkonstante,
- Td = die abgeleitete Zeitkonstante,
- Ts = die Abtastzeit,
- $\varepsilon(i)$ = die Abweichung ($\varepsilon(i)$ = Sollwert - Prozessvariable).

Hinweis: Es werden abhängig vom Wert der integralen Zeitkonstante (Ti) zwei Rechenalgorithmen verwendet:

- Ti $\neq$ 0: In diesem Fall wird ein inkrementaler Algorithmus verwendet.
- Ti = 0: Dies ist der Fall für nicht-integrierende Prozesse. In diesem Fall wird ein Positionsalgorithmus zusammen mit einem Offset von +5000, der auf die PID-Ausgangsvariable angewandt wird, verwendet.

Eine ausführliche Beschreibung von Kp, Ti und Td finden Sie unter *Registerkarte "PID" des PID, S. 549*.

Wie aus *(Gleichung 1)* und *(Gleichung 1')* gefolgert werden kann, ist der wichtigste Parameter für die PID-Regelung die **Abtastzeit (Ts)**. Die Abtastzeit hängt eng von der **Zeitkonstante (τ)** ab, einem Parameter des durch den PID-Regler gesteuerten Prozesses (siehe *Anhang 2: Modell erster Ordnung mit Zeitverzögerung, S. 586*.)

Anhang 2: Modell erster Ordnung mit Zeitverzögerung

Einführung

Dieser Abschnitt beschreibt das Modell erster Ordnung mit Zeitverzögerung, das zur Beschreibung einer Vielzahl einfacher, jedoch nicht unwichtiger industrieller Prozesse einschließlich thermischer Prozesse verwendet wird.

Modell erster Ordnung mit Zeitverzögerung

Es wird weithin angenommen, dass einfache (aus einem Stimulus bestehende) thermische Prozesse in angemessener Art und Weise durch ein Modell erster Ordnung mit Zeitverzögerung angenähert werden können.
Die Übertragungsfunktion eines derartigen rückführungsfreien Prozesses erster Ordnung hat die folgende Form in der Laplace-Domain *(Gleichung 2):*

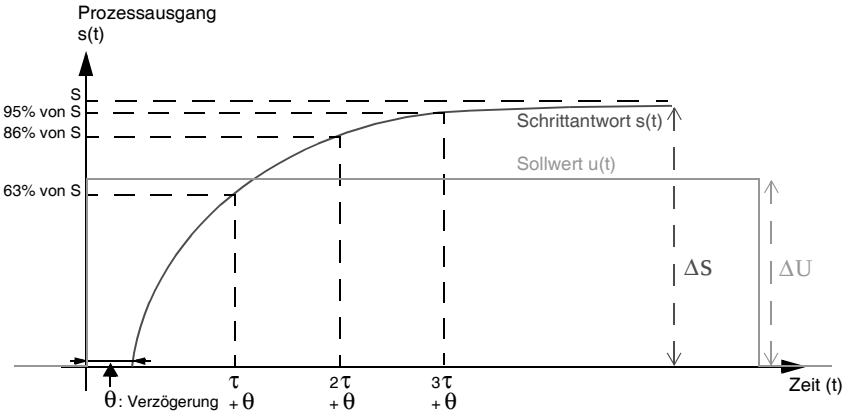
$$\frac{S}{U} = \frac{k}{1 + \tau p} \cdot e^{-\theta p}$$

wobei

- k = die statische Verstärkung,
- τ = die Zeitkonstante,
- θ = die Verzögerungszeit,
- U = der Prozesseingang (dies ist der Ausgang des PID-Reglers),
- S = der Prozessausgang.

Die Prozesszeitkonstante τ

Der wichtigste Parameter des Prozessantwortgesetzes *(Gleichung 2)* ist die **Zeitkonstante τ.** Dies ist ein Parameter des zu steuernden Prozesses.
Die Zeitkonstante (τ) eines Systems erster Ordnung ist definiert als die Zeit (in Sekunden), die vergeht, bis die Systemausgangsvariable ab dem Moment, in dem das System die Reaktion auf den Schrittstimulus u(t) begonnen hat, 63% des endgültigen Ausgangs erreicht.
Die folgende Abbildung zeigt eine typische Reaktion des Prozesses erster Ordnung auf einen Schrittstimulus.



wobei

- k = die statische Verstärkung, berechnet als das Verhältnis ΔS/ΔU,
- τ = die Zeit beim 63%igen Anstieg = die Zeitkonstante,
- 2τ = die Zeit beim 86%igen Anstieg,
- 3τ = die Zeit beim 95%igen Anstieg.

Hinweis: Wenn Auto-Tuning implementiert ist, muss die Abtastzeit (Ts) innerhalb des folgenden Bereichs ausgewählt werden: [τ/125 <Ts < τ/25]. Idealerweise sollten Sie [Ts= τ/75] verwenden (siehe *PID-Einstellung mittels Auto-Tuning (AT), S. 569*.)

17.5 Anweisungen zu Gleitkommas

Auf einen Blick

Inhalt des Abschnitts

In diesem Abschnitt werden die erweiterten Anweisungen für Gleitkommas (Siehe *Gleitkomma- und Doppelwortobjekte, S. 33*) in der TwidoSoft-Sprache beschrieben. Die Vergleichs- und Zuweisungsanweisungen sind im Abschnitt *Numerische Verarbeitung, S. 426* beschrieben.

Inhalt dieses Abschnitts

Dieser Abschnitt enthält die folgenden Themen:

Thema	Seite
Arithmetische Anweisungen für Gleitkommawerte	589
Trigonometrische Anweisungen	593
Konvertierungsanweisungen	595
Anweisungen zur Konvertierung von Ganzzahlen <-> Gleitpunktzahlen	597

Arithmetische Anweisungen für Gleitkommawerte

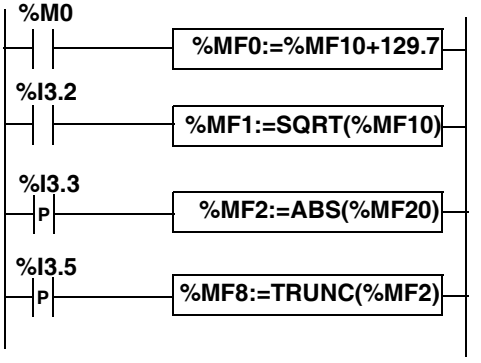
Allgemeines

Mit diesen Anweisungen kann eine arithmetische Operation zwischen zwei Operanden oder mit einem Operanden ausgeführt werden.

+	Addition zweier Operanden	**SQRT**	Quadratwurzel eines Operanden
-	Subtraktion zweier Operanden	**ABS**	Absolutwert eines Operanden
*****	Multiplikation zweier Operanden	**TRUNC**	ganzer Teil eines Gleitkommawertes
/	Division zweier Operanden	**EXP**	natürliche Exponentialfunktion
LOG	Zehnerlogarithmus	**EXPT**	Potenz einer Ganzzahl mit einer Realzahl
LN	natürlicher Logarithmus		

Struktur

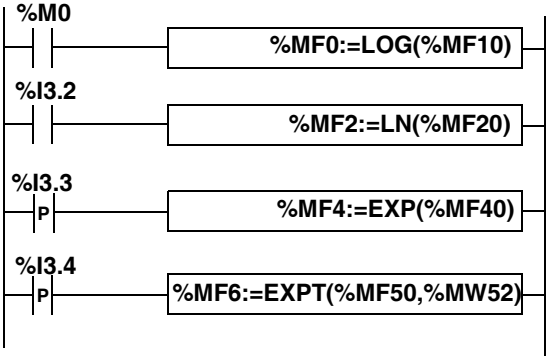
Kontaktplan(sprache)



Anweisungsliste (AWL)

```
LD %M0
[%MF0:=%MF10+129.7]

LD %I3.2
[%MF1:=SQRT(%MF10)]

LDR %I3.3
[%MF2:=ABS(%MF20)]

LDR %I3.5
[%MF8:=TRUNC(%MF2)]
```

Kontaktplan(sprache)



Anweisungsliste (AWL)

```
LD %M0
[%MF0:=LOG(%MF10]

LD %I3.2
[%MF2:=LN(%MF20)]

LDR %I3.3
[%MF4:=EXP(%MF40)]

LDR %I3.4
[%MF6:=EXPT(%MF50,%MW52)]
```

Syntax

Operatoren und Syntax der arithmetischen Anweisungen für Gleitkommawerte

Operatoren	Syntax
+, - *, /	Op1:=Op2 Operator Op3
SQRT, ABS, TRUNC, LOG, EXP, LN	Op1:=Operator(Op2)
EXPT	Op1:=Operator (Op2,Op3)

Hinweis: Bei der Addition oder Subtraktion zweier Gleitkommazahlen müssen die beiden Operanden die Bedingung $\mathrm{Op1} > \mathrm{Op2} \times 2^{-24}$ einhalten, wobei Op1>Op2. Wenn diese Bedingung nicht eingehalten wird, entspricht das Ergebnis dem Operanden 1 (Op1). Dieses Verhalten hat keine große Auswirkung, wenn es sich um eine isolierte Operation handelt, da die resultierende Abweichung sehr gering ist (2^{-24}). Bei iterativen Berechnungen kann es jedoch zu unerwarteten Folgen kommen.
Beispiel: Gegeben sei die Anweisung **%MF2:= %MF2 + %MF0**, die sich unendlich wiederholt. Wenn die Anfangsbedingungen %MF0 = 1.0 und %MF2= 0 sind, ist eine Blockierung des Wertes %MF2 bei 16777216 zu beobachten.
Iterative Berechnungen sollten daher nur mit großer Vorsicht programmiert werden. Wenn Sie dennoch diese Art von Berechnung programmieren wollen, muss die Client-Anwendung die Abbrechfehler verwalten.

Operanden der arithmetischen Anweisungen für Gleitkommawerte:

Operatoren	Operand 1 (Op1)	Operand 2 (Op2)	Operand 3 (Op3)
+, - *, /	%MFi	%MFi, %KFi, direkter Wert	%MFi, %KFi, direkter Wert
SQRT, ABS, LOG, EXP, LN	%MFi	%MFi, %KFi	[-]
TRUNC	%MFi	%MFi, %KFi	[-]
EXPT	%MFi	%MFi, %KFi	%MWi, %KWi, direkter Wert

Verwendungs-regeln

- die Operationen mit Gleitkommawerten und mit Ganzzahlen können nicht direkt gemischt werden. (Die Konvertierungsoperationen (Siehe *Anweisungen zur Konvertierung von Ganzzahlen <-> Gleitpunktzahlen, S. 597*) gewährleisten die Konvertierung in das eine oder andere dieser Formate.)
- Das Systembit %S18 wird auf die gleiche Weise verwaltet wie die Operationen mit Ganzzahlen (Siehe *Arithmetische Anweisungen für Ganzzahlen, S. 435*), das Wort %SW17 (Siehe *Systemwörter (%SW), S. 633*) zeigt die Fehlerursache an.
- Wenn der Operand der Funktion ein ungültiger Wert ist (Beispiel: Logarithmus einer negativen Zahl, so erzeugt er ein unbestimmtes oder ein unendliches Ergebnis und setzt das Bit %S18 auf 1; das Wort %SW17 gibt den Grund des Fehlers an.

Trigonometrische Anweisungen

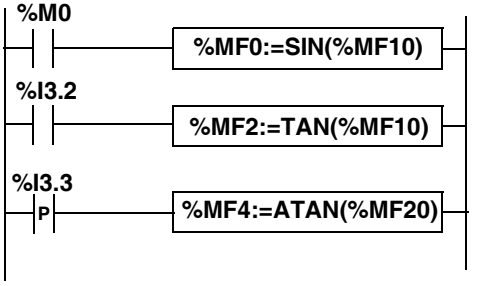
Allgemein

Diese Anweisungen ermöglichen die Durchführung trigonometrischer Rechenoperationen.

SIN	Sinus eines Winkels, ausgedrückt im Bogenmaß,	**ASIN**	Arkussinus (resultieren zwischen $-\frac{\pi}{2}$ und $\frac{\pi}{2}$)
COS	Kosinus eines Winkels, ausgedrückt im Bogenmaß,	**ACOS**	Arkuskosinus (resultieren zwischen 0 und π)
TAN	Tangens eines Winkels, ausgedrückt im Bogenmaß.	**ATAN**	Arkustangens (resultieren zwischen $-\frac{\pi}{2}$ und $\frac{\pi}{2}$)

Struktur

Kontaktplan(sprache)



Anweisungsliste (AWL)

```
LD %M0
[%MF0:=SIN(%MF10)]

LD %I3.2
[%MF2:=TAN(%MF10)]

LDR %I3.3
[%MF4:=ATAN(%MF20)]
```

Strukturierte Textsprache

```
IF %M0 THEN
 %MF0:=SIN(%MF10);
END_IF;
IF %I3.2 THEN
 %MF2:=TAN(%MF10);
END_IF;
IF %I3.3 THEN
 %MF4:=ATAN(%MF20);
END_IF;
```

Syntax

Operatoren, Operanden und Syntax der trigonometrischen Rechenanweisungen:

Operatoren	Syntax	Operand 1 (Op1)	Operand 2 (Op2)
SIN, COS, TAN, ASIN, ACOS, ATAN	Op1:=Operator(Op2)	%MFi	%MFi, %KFi

Verwendungsregeln

- Wenn der Operand der Funktion ein ungültiger Wert ist (Beispiel: Arkuskosinus einer Zahl über 1), so erzeugt er ein unbestimmtes oder ein unendliches Ergebnis und setzt das Bit %S18 auf 1; das Wort %SW17 (Siehe *Systemwörter (%SW), S. 633*) gibt den Grund des Fehlers an.
- Die Funktionen SIN/COS/TAN lassen als Parameter einen Winkel zwischen -4096π und 4096π zu, aber ihre Genauigkeit nimmt progressiv für Winkel außerhalb des Intervalls -2π und $+2\pi$ ab. Dies ist auf die Ungenauigkeit des Modulo 2π zurückzuführen, das vor jeder Operation auf den Parameter angewandt wird.

Konvertierungsanweisungen

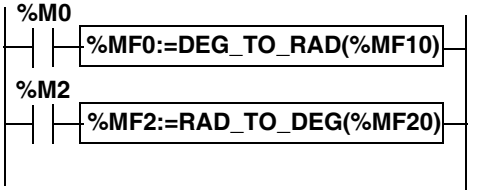
Allgemein

Mit diesen Anweisungen können Konvertierungsoperationen ausgeführt werden.

DEG_TO_RAD	Konvertierung von Grad in Radiant, das Ergebnis ist der Wert des Winkels zwischen 0 und 2π
RAD_TO_DEG	Konvertierung eines Winkels in Radiant; das Ergebnis ist der Wert des Winkels zwischen 0 und 360 Grad

Struktur

Kontaktplan(sprache)



Anweisungsliste (AWL)

```
LD %M0
[%MF0:=DEG_TO_RAD(%MF10)]

LD %M2
[%MF2:=RAD_TO_DEG(%MF20)]
```

Strukturierte Textsprache

```
IF %M0 THEN
 %MF0:=DEG_TO_RAD(%MF10);
END_IF;
IF %M2 THEN
 %MF2:=RAD_TO_DEG(%MF20);
END_IF;
```

Syntax

Operatoren, Operanden und Syntax der Konvertierungsanweisungen:

Operatoren	Syntax	Operand 1 (Op1)	Operand 2 (Op2)
DEG_TO_RAD **RAD_TO_DEG**	Op1:=Operator(Op2)	%MFi	%MFi, %KFi

Verwendungs-regeln

Der zu konvertierende Winkel muss zwischen -737280.0 und +737280.0 (für Konvertierungen des Typs DEG_TO_RAD) oder zwischen -4096π und 4096π (für Konvertierungen des Typs RAD_TO_DEG) liegen.
Bei Werten, die nicht zwischen diesen Grenzen liegen, ist das Ergebnis + 1.#NAN, wobei die Bits %S18 und %SW17:X0 auf 1 gesetzt sind.

Anweisungen zur Konvertierung von Ganzzahlen <-> Gleitpunktzahlen

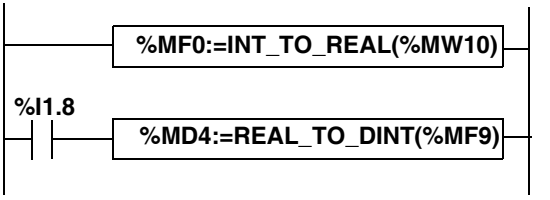
Allgemein

Es sind vier Konvertierungsanweisungen verfügbar.
Liste der Anweisungen zur Konvertierung von Ganzzahlen <-> Gleitpunktzahlen:

INT_TO_REAL	Konvertierung Ganzzahlwort --> Gleitpunktwort
DINT_TO_REAL	Konvertierung Doppelwort (Ganzzahl) --> Gleitpunktwort
REAL_TO_INT	Konvertierung Gleitpunktwort --> Ganzzahlwort (das Ergebnis ist der nächste algebraische Wert)
REAL_TO_DINT	Konvertierung Gleitpunktwort --> Ganzzahl-Doppelwort (das Ergebnis ist der nächste algebraische Wert)

Struktur

Kontaktplan(sprache)



Anweisungsliste (AWL)

```
LD TRUE
[%MF0:=INT_TO_REAL(%MW10)]

LD I1.8
[%MD4:=REAL_TO_DINT(%MF9)]
```

Strukturierte Textsprache

```
%MF0:=INT_TO_REAL(%MW10);
IF %I1.8 THEN
 %MD4:=REAL_TO_DINT(%MF9);
END_IF;
```

Syntax

Operatoren und Syntax (Konvertierung Ganzzahlwort --> Gleitpunktwort):

Operatoren	Syntax
INT_TO_REAL	Op1=INT_TO_REAL(Op2)

Operanden (Konvertierung Ganzzahlwort --> Gleitpunktwort):

Operand 1 (Op1)	Operand 2 (Op2)
%MFi	%MWi,%KWi

Beispiel: Konvertierung Ganzzahlwort --> Gleitpunktwort: 147 --> 1.47e+02

Operatoren und Syntax (Konvertierung ganzes Doppelwort --> Gleitpunkt-Doppelwort):

Operatoren	Syntax
DINT_TO_REAL	Op1=DINT_TO_REAL(Op2)

Operanden (Konvertierung ganzes Doppelwortes --> Gleitpunkt-Doppelwort):

Operand 1 (Op1)	Operand 2 (Op2)
%MFi	%MDi,%KDi

Beispiel: Konvertierung ganzes Doppelwort --> Gleitpunkt-Doppelwort: 68905000 --> 6.8905e+07

Operatoren und Syntax (Konvertierung Gleitpunktwort/Gleitpunkt-Doppelwort --> Vollwort oder ganzes Doppelwort):

Operatoren	Syntax
REAL_TO_INT	Op1=Operator(Op2)
REAL_TO_DINT	

Operanden (Konvertierung Gleitpunktwort/Gleitpunkt-Doppelwort --> Vollwort oder ganzes Doppelwort):

Typ	Operand 1 (Op1)	Operand 2 (Op2)
Worte	%MWi	%MFi, %KFi
Doppelworte	%MDi	%MFi, %KFi

Beispiel:
Konvertierung Gleitpunktwort --> Vollwort: 5978.6 --> 5979
Konvertierung Gleitpunkt-Doppelwort --> ganzes Doppelwort: -1235978.6 --> -1235979

Hinweis: Wenn während der Konvertierung einer Realzahl in ein Ganzzahlwort (oder einer Realzahl in ein Ganzzahl-Doppelwort) der Gleitpunktwert außerhalb der Grenzen des Worts (oder Doppelworts) liegt, wird das Bit %S18 auf 1 gesetzt.

Rundungs-genauigkeit

Die Norm IEEE 754 schreibt 4 Rundungsverfahren für Operationen mit Gleitpunktwerten vor.
Das von den oben aufgeführten Anweisungen angewandte Verfahren ist das Verfahren der "nächsten Rundung":
"Wenn die nächsten darstellbaren Werte sich in einem gleichen Abstand zum theoretischen Ergebnis befinden, ist der gelieferte Wert der, dessen niederwertigstes Bit gleich "0" ist.
In bestimmten Fällen kann das Rundungsergebnis ein Standardwert oder ein aufgerundeter Wert sein.

Beispiel:
Rundung des Wertes 10,5 -> 10
Rundung des Wertes 11,5 -> 12

17.6 Anweisungen für Objekttabellen:

Auf einen Blick

Inhalt des Abschnitts

In diesem Abschnitt sind die spezifischen Anweisungen in Tabellen beschrieben:
- in Doppelworttabellen,
- in Gleitkommatabellen.

Die Zuweisungsanweisungen in Tabellen sind im Kapitel "Basis-Anweisungen" (Siehe *Zuweisung der Wort-, Doppelwort- oder Gleitkommawerttabellen, S. 431*) beschrieben.

Inhalt dieses Abschnitts

Dieser Abschnitt enthält die folgenden Themen:

Thema	Seite
Summierungsfunktion in Tabellen	601
Tabellenvergleichsfunktion	602
Suchfunktionen an Tabellen	604
Suchfunktionen nach Minimal- und Maximalwerten in Tabellen	606
Anzahl der Vorkommen eines Werts in einer Tabelle	607
Funktion Zirkularverschiebung in einer Tabelle	608
Sortierfunktion in Tabellen	610
Interpolationsfunktion in Gleitkommazahltabellen	612
Funktion zur Bildung des Mittelwertes der Werte einer Gleitkommawerttabelle	617

Summierungsfunktion in Tabellen

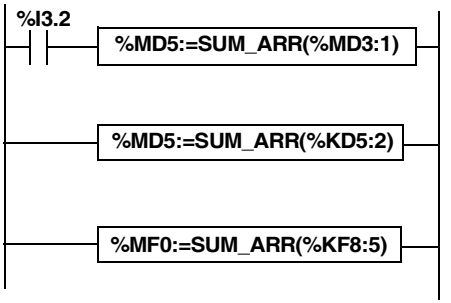
Allgemeines

Die Funktion SUM_ARR bildet die Summe sämtlicher Elemente einer Objekttabelle:

- Besteht die Tabelle aus Doppelworten, wird das Ergebnis in Form eines Doppelwortes ausgegeben.
- Besteht die Tabelle aus Gleitkommazahlworten, wird das Ergebnis in Form eines Gleitkommazahlwortes ausgegeben.

Struktur

Kontaktplan(sprache)



Anweisungsliste (AWL)

```
LD %I3.2
[%MD5:=SUM_ARR(%MD3:1)]
%MD5:=SUM_ARR(%KD5:2)
%MF0:=SUM_ARR(%KF8:5)
```

Syntax

Syntax von Summierungsanweisungen an Tabellen:

```
Erg:=SUM_ARR(Tab)
```

Parameter von Summierungsanweisungen an Tabellen

Typ	Ergebnis (Erg)	Tabelle (Tab)
Doppelworttabellen	%MDi	%MDi:L,%KDi:L
Gleitpunktzahltabellen	%MFi	%MFi:L,%KFi:L

Hinweis: Das Bit %S18 wird auf 1 gesetzt, wenn das Ergebnis entsprechend dem Tabellenoperanden außerhalb der Grenzwerte des Doppelwortformats liegt.

Im Beispiel zur Bezugnahme

```
%MD5:=SUM(%MD30:4)
```

wobei %MD30=10, %MD31=20, %MD32=30, %MD33=40
%MD5=10+20+30+40=100

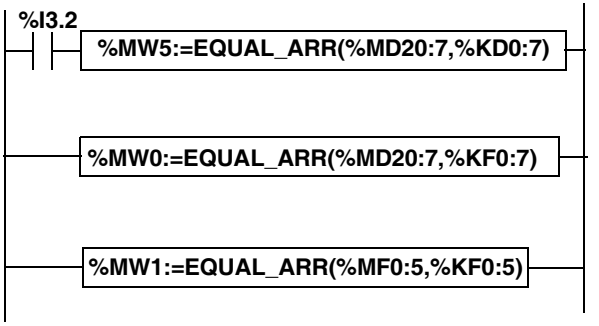
Tabellenvergleichsfunktion

Allgemeines

Mit der Funktion und EQUAL_ARR werden 2 Tabellen Element für Element miteinander verglichen.
Wird ein Unterschied ermittelt, wird die Reihenfolgenummer der ersten ungleichen Elemente in Form eines Wortes zurückgesendet, ansonsten ist der Wert des zurückgesendeten Wortes gleich –1.
Der Vergleich wird über die gesamte Tabelle vorgenommen.

Struktur

Kontaktplan(sprache)



Anweisungsliste (AWL)

```
LD %I3.2
[%MW5:=EQUAL_ARR(%MD20:7,KD0:7)]
```

Strukturierte Textsprache

```
%MW0:=EQUAL_ARR(%MD20:7,%KF0:7)

%MW1:=EQUAL_ARR(%MF0:5,%KF0:5)
```

Syntax

Syntax von Vergleichsanweisungen an Tabellen:

```
Erg:=EQUAL_ARR(Tab1,Tab2)
```

Parameter von Vergleichsanweisungen an Tabellen:

Typ	Ergebnis (Erg)	Tabellen (Tab1 und Tab2)
Doppelworttabellen	%MWi	%MDi:L,%KDi:L
Gleitpunktzahltabellen	%MWi	%MFi:L,%KFi:L

Hinweis:
- Die Tabellen müssen unbedingt identischer Länge und vom gleichen Typ sein.

Im Beispiel zur Bezugnahme

```
%MW5:=EQUAL_ARR(%MD30:4,%KD0:4)
```

Vergleich von 2 Tabellen:

Reihenfolgenummer	Worttabelle	Konstantentabellen	Unterschied
0	%MD30=10	%KD0=10	=
1	%MD31=20	%KD1=20	=
2	%MD32=30	%KD2=60	unterschiedlich
3	%MD33=40	%KD3=40	=

Der Wert des Wortes %MW5 ist gleich 2 (erste unterschiedliche Reihenfolgenummer)

Suchfunktionen an Tabellen

Allgemeines

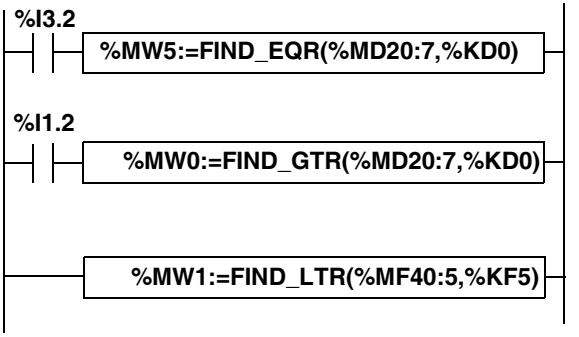
Es stehen 3 Suchfunktionen zur Verfügung:

- **FIND_EQR**: Suche in einer Doppelworttabelle oder einer Gleitkommatabelle nach der Position des ersten Elements, dessen Wert gleich einem gegebenen Wert ist.
- **FIND_GTR** : Suche in einer Doppelworttabelle oder einer Gleitkommatabelle nach der Position des ersten Elements, dessen Wert größer als ein gegebener Wert ist.
- **FIND_LTR** : Suche in einer Doppelworttabelle oder einer Gleitkommatabelle nach der Position des ersten Elements, dessen Wert niedriger als ein gegebener Wert ist.

Das Ergebnis dieser Anweisungen ist gleich der Reihenfolgenummer des ersten gefundenen Elements oder –1, wenn die Suche zu keinem Ergebnis führte.

Struktur

Kontaktplan(sprache)



Anweisungsliste (AWL)

```
LD %I3.2
[%MW5:=FIND_EQR(%MD20:7,KD0)]
LD %I1.2
[%MW0:=FIND_GTR(%MD20:7,%KD0)]
%MW1:=FIND_LTR(%MF40:5,%KF5)
```

Syntax

Syntax von Suchanweisungen an Tabellen:

Funktion	Syntax
FIND_EQR	Erg:=Funktion(Tab,Wert)
FIND_GTR	
FIND_LTR	

Parameter der Suchanweisungen für Gleitkomma- oder Doppelworttabellen:

Typ	Ergebnis (Erg)	Tabelle (Tab)	Wert (Wert)
Gleitpunktzahltabellen	%MWi	%MFi:L,%KFi:L	%MFi,%KFi
Doppelworttabellen	%MWi	%MDi:L,%KDi:L	%MDi,%KDi

Im Beispiel zur Bezugnahme

```
%MW5:=FIND_EQR(%MD30:4,%KD0)
```

Suche nach der Position des ersten Doppelwortes =%KD0=30 in der Tabelle:

Reihenfolgenummer	Worttabelle	Ergebnis
0	%MD30=10	-
1	%MD31=20	-
2	%MD32=30	%MW5=2 (Wert der Reihenfolgennummer)
3	%MD33=40	-

Suchfunktionen nach Minimal- und Maximalwerten in Tabellen

Allgemeines

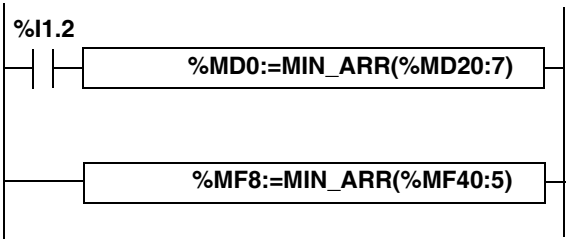
Es stehen 2 Suchfunktionen zur Verfügung:

- **MAX_ARR**: Suche nach dem Maximalwert in einer Doppelwort- oder Gleitkommatabelle
- **MIN_ARR**: Suche nach dem Minimalwert in einer Doppelwort- oder Gleitkommatabelle

Das Ergebnis dieser Anweisungen ist gleich dem in der Tabelle gefundenem Maximalwert (bzw. Minimalwert).

Struktur

Kontaktplan(sprache)



Anweisungsliste (AWL)

```
LD %I1.2
[%MD0:=MIN_ARR(%MD20:7)]
%MF8:=MIN_ARR(%MF40:5)
```

Syntax

Syntax von Suchanweisungen für Maximal- und Minimalwerte an Tabellen:

Funktion	Syntax
MAX_ARR	Erg:=Funktion(Tab)
MIN_ARR	

Parameter der Suchanweisungen für Maximal- und Minimalwerte an Tabellen:

Typ	Ergebnis (Erg)	Tabelle (Tab)
Doppelworttabellen	%MDi	%MDi:L,%KDi:L
Gleitpunktzahltabellen	%MFi	%MFi:L,%KFi:L

Anzahl der Vorkommen eines Werts in einer Tabelle

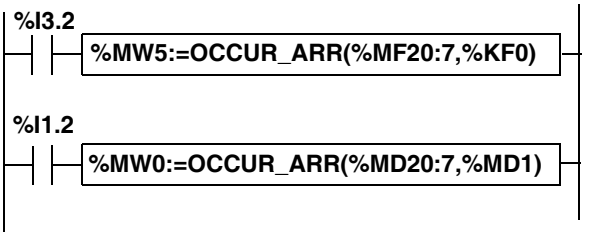
Allgemeines

Die zur Verfügung stehende Suchfunktion ist:

- **OCCUR_ARR**: Suche in einer Doppelworttabelle oder einer Gleitkommatebelle nach der Anzahl der Elemente, die einem gegebenen Wert entsprechen.

Struktur

Kontaktplan(sprache)



Anweisungsliste (AWL)

```
LD %I3.2
[%MW5:=OCCUR_ARR(%MF20:7,%KF0)]
LD %I1.2
[%MW0:=OCCUR_ARR(%MD20:7,%MD1)
```

Syntax

Syntax von Suchanweisungen für Maximal- und Minimalwerte an Tabellen:

Funktion	Syntax
OCCUR_ARR	Erg:=Funktion(Tab,Wert)

Parameter der Suchanweisungen für Maximal- und Minimalwerte an Tabellen:

Typ	Ergebnis (Erg)	Tabelle (Tab)	Wert (Wert)
Doppelworttabellen	%MWi	%MDi:L,%KDi:L	%MDi,%KDi
Gleitpunktzahltabellen	%MFi	%MFi:L,%KFi:L	%MFi,%KFi

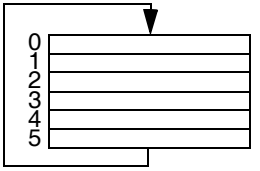
Funktion Zirkularverschiebung in einer Tabelle

Allgemeines

Es stehen 2 Verschiebungsfunktionen zur Verfügung:

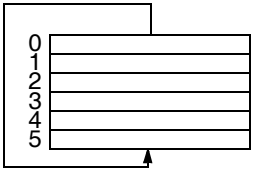
- **ROL_ARR** : führt an Elementen der Gleitpunkzahltabelle die Zirkularverschiebung von n Positionen von oben nach unten aus.

Veranschaulichung der ROL_ARR-Funktionen



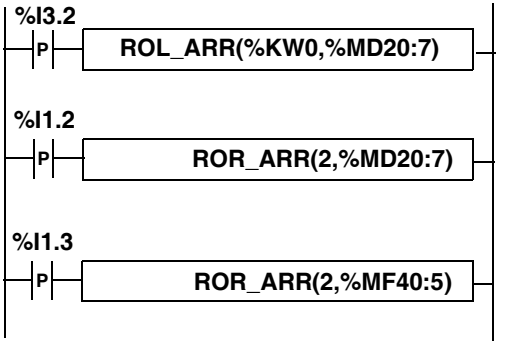
- **ROR_ARR** : führt an Elementen der Gleitpunktzahltabelle die Zirkularverschiebung von n Positionen von unten nach oben aus.

Veranschaulichung der ROR_ARR-Funktion



Struktur

Kontaktplan(sprache)



Anweisungsliste (AWL)

```
LDR %I3.2
[ROL_ARR(%KW0,%MD20:7)]
LDR %I1.2
 [ROR_ARR(2,%MD20:7)]
LDR %I1.3
[ROR_ARR(2,%MF40:5)]
```

Syntax

Syntax von Anweisungen zur Zirkularverschiebung in Doppelwort- oder Gleitkommazahltabellen **ROL_ARR** und **ROR_ARR**

Funktion	Syntax
ROL_ARR	Funktion(n,Tab)
ROR_ARR	

Parameter von Anweisungen zur Zirkularverschiebung an Gleitpunktzahltabellen: **ROL_ARR** und **ROR_ARR**:

Typ	Anzahl der Positionen (n)	Tabelle (Tab)
Gleitpunktzahltabellen	%MWi, direkter Wert	%MFi:L
Doppelworttabellen	%MWi, direkter Wert	%MDi:L

Hinweis: Ist der Wert n negativ oder Null, wurde keinerlei Verschiebung ausgeführt.

Sortierfunktion in Tabellen

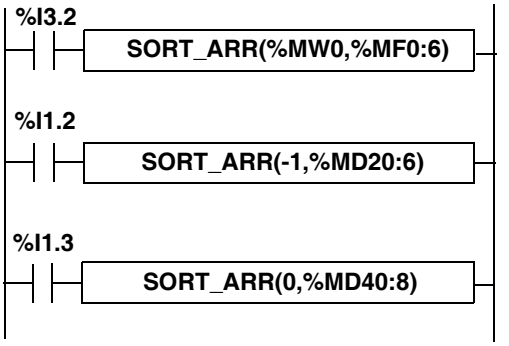
Allgemeines

Die Sortierfunktion ist:

- **SORT_ARR** : Sortiert die Elemente einer Doppelwort- oder Gleitkommazahltabelle in aufsteigender oder absteigender Reihenfolge und speichert das Ergebnis in eben dieser Tabelle ab.

Struktur

Kontaktplan(sprache)



Anweisungsliste (AWL)

```
LD %I3.2
[SORT_ARR(%MW20,%MF0:6)]
LD %I1.2
[SORT_ARR(-1,%MD20:6)]
LD %I1.3
[SORT_ARR(0,%MF40:8)
```

Syntax

Syntax von Sortieranweisungen an Tabellen:

Funktion	Syntax
SORT_ARR	Funktion(Richtung,Tab)

- Der Parameter "Richtung" gibt die Sortierrichtung an. Richtung > 0: aufsteigende Sortierrichtung, Richtung < 0: absteigende Sortierrichtung, Richtung = 0: keine Sortierung.
- Das Ergebnis (sortierte Tabelle) wird im Parameter Tab (zu sortierende Tabelle) zurückgesendet.

Parameter von Sortieranweisungen an Tabellen:

Typ	Sortierrichtung	Tabelle (Tab)
Doppelworttabellen	%MWi, direkter Wert	%MDi:L
Gleitpunktzahltabellen	%MWi, direkter Wert	%MFi:L

Interpolationsfunktion in Gleitkommazahltabellen

Überblick

Die **LKUP**-Funktion wird verwendet, um eine Reihe von X-Gleitkommadaten mit einer Reihe von Y-Gleitkommadaten für einen bestimmten X-Wert zu interpolieren.

Interpolationsregel

Die LKUP-Funktion verwendet die lineare Interpolationsregel wie in folgender Gleichung definiert:

(Gleichung 1:) $$Y = Y_i + \left[\frac{(Y_{i+1} - Y_i)}{(X_{i+1} - X_i)} \cdot (X - X_i)\right]$$

für $X_i \le X \le X_{i+1}$, wobei $i = 1 \ldots (m-1)$;

wobei davon ausgegangen wird, dass X_i-Werte in aufsteigender Reihenfolge angeordnet sind: $X_1 \le X_2 \le \ldots X \ldots \le X_{m-1} \le X_m$.

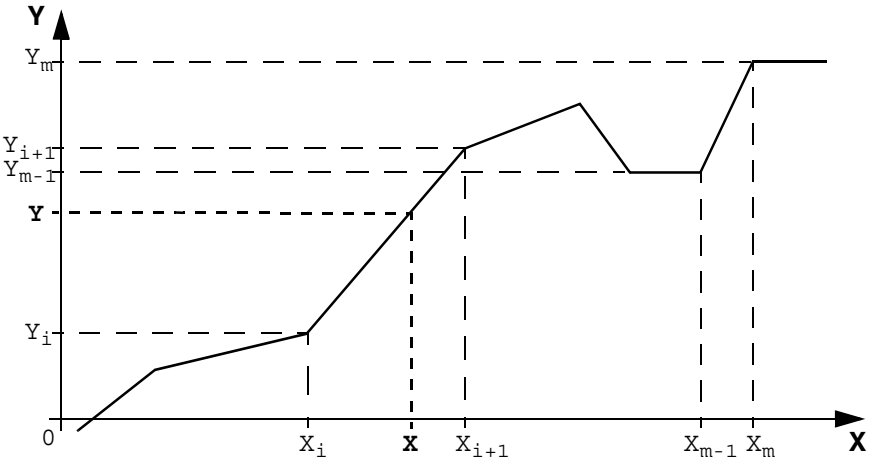
Hinweis: Wenn zwei beliebige aufeinander folgende Xi-Werte gleich sind ($X_i=X_{i+1}=X$), führt die Gleichung (1) zu einer ungültigen Ausnahme. In diesem Fall wird, um diese Ausnahme verarbeiten zu können, der folgende Algorithmus anstelle der Gleichung (1) verwendet:

(Gleichung 2:) $$Y = \left[\frac{(Y_{i+1} - Y_i)}{2}\right]$$

für $X_i = X_{i+1} = X$, wobei $i = 1 \ldots (m-1)$.

Graphische Darstellung der linearen Interpolati-onsregel

Das folgende Diagramm veranschaulicht die oben beschriebene lineare Interpolationsregel:



Syntax der LKUP-Funktion

Die LKUP-Funktion verwendet drei Operanden, von denen zwei Funktionsattribute sind (wie in folgender Tabelle beschrieben):

Syntax	**Operand 1 (Op1)** **Ausgangsvariable**	**Operand 2 (Op2)** **Benutzerdefinierter (X) Wert**	**Operand 3 (Op3)** **Benutzerdefinierter (X_i,Y_i)** **Variablenbereich**
[Op1: = LKUP(Op2,Op3)]	%MWi	%MF0	Ganzzahlwert, %MWi oder %KWi

Definition von Op1

Op1 ist das Speicherwort, das die Ausgangsvariable der Interpolationsfunktion enthält.
Der Wert von Op1 zeigt dem Benutzer an, ob die Interpolation erfolgreich war oder fehlgeschlagen ist und was zum Fehler geführt hat (siehe folgende Tabelle).

Op1 (%Mwi)	Beschreibung
0	Erfolgreiche Interpolation
1	Interpolationsfehler: Falscher Bereich, $X_m < X_{m-1}$
2	Interpolationsfehler: Op2 außerhalb des zulässigen Bereichs, $X < X_1$
4	Interpolationsfehler: Op2 außerhalb des zulässigen Bereichs, $X > X_m$
8	Unzulässige Größe des Datenbereichs: ● Op3 ist als ungerade Zahl gesetzt, oder ● Op3 < 6.

Hinweis: Op1 enthält **nicht** den berechneten Interpolationswert (Y). Für einen bestimmten (X) Wert ist das Ergebnis der Interpolation (Y) in %MF2 des Op3-Bereichs enthalten (siehe *Definition von Op3* unten).

Definition von Op2

Op2 ist die Gleitkommavariable (%MF0 des Op3-Gleitkommabereichs), die den benutzerdefinierten (X) Wert enthält, für den der interpolierte (Y) Wert berechnet werden soll:

- Der zulässige Bereich für Op2 lautet wie folgt: $X_1 \leq Op2 \leq X_m$.

Definition von Op3

Op3 legt die Größte (Op3 / 2) des Gleitkommabereichs fest, in dem die (X_i,Y_i) Datenpaare gespeichert sind.
X_i- und Y_i-Daten werden in Gleitkommaobjekten mit geraden Indexen beginnend bei %MF4 gespeichert (beachten Sie, dass die Gleitkommaobjekte %MF0 und %MF2 für den benutzerdefinierten Sollwert X bzw. den interpolierten Wert Y reserviert sind).

Für einen bestimmten Bereich (m) von Datenpaaren (X_i,Y_i) wird der obere Index (u) des Gleitkommabereichs (%MFu) mittels der folgenden Beziehungen gesetzt:

- *(Gleichung 3:)* $Op3 = 2 \cdot m$;
- *(Gleichung 4:)* $u = 2 \cdot (Op3 - 1)$.

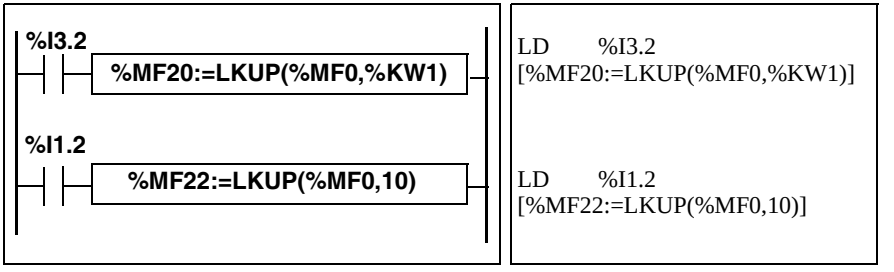
Der Gleitkommabereich Op3 (%MFi) hat eine Struktur, die identisch mit der des folgenden Beispiels ist (wobei Op3=8):

(X)		(X_1)		(X_2)		(X_3)	
%MF0		**%MF4**		**%MF8**		**%MF12**	
	%MF2		**%MF6**		**%MF10**		**%MF14**
	(Y)		(Y_1)		(Y_2)		(Y_3)
							(Op3=8)

Hinweis: Als Ergebnis der oben aufgeführten Struktur des Gleitkommabereichs muss Op3 die beiden folgenden Anforderungen erfüllen. Ansonsten führt dies zu einem Fehler der LKUP-Funktion:
- Op3 ist eine gerade Zahl und
- Op3 ≥ 6 (weil mindestens 2 Datenpunkte vorhanden sein müssen, um eine lineare Interpolation zu ermöglichen).

Struktur

Die Interpolationsoperationen werden wie folgt ausgeführt:



Beispiel

Nachfolgend ist die beispielhafte Verwendung einer LKUP-Interpolationsfunktion aufgeführt:

```
[%MW20:=LKUP(%MF0,10)]
```

In diesem Beispiel gilt Folgendes:

- %MW20 ist Op1 (die Ausgangsvariable).
- %MF0 ist der benutzerdefinierte (X) Wert, dessen entsprechender (Y) Wert durch die lineare Interpolation berechnet werden muss.
- %MF2 speichert den durch die lineare Interpolation berechneten Wert (Y).
- 10 ist Op3 (wie durch *Gleichung 3* oben vorgegeben). Dieser legt die Größe des Gleitkommabereichs fest. Das höchstrangige Element ist %MFu, wobei u=18 durch *Gleichung 4* oben vorgegeben ist.

Es gibt 4 Datenpunktpaare, die im Bereich Op3 [%MF4..%MF18] gespeichert sind:

- %MF4 enthält X_1, %MF6 enthält Y_1.
- %MF8 enthält X_2, %MF10 enthält Y_2.
- %MF12 enthält X_3, %MF14 enthält Y_3.
- %MF16 enthält X_4, %MF18 enthält Y_4.

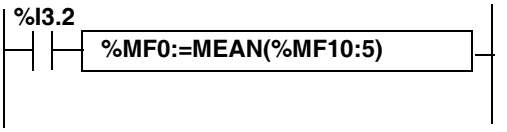
Funktion zur Bildung des Mittelwertes der Werte einer Gleitkommawerttabelle

Allgemeines

Die Funktion **MEAN** ermöglicht die Berechnung des Mittelwertes der Werte einer bestimmten Anzahl von Elementen einer Gleitkommawerttabelle.

Struktur

Kontaktplan(sprache)



Anweisungsliste (AWL)

```
LD %I3.2
[%MF0:=MEAN(%MF10:5)]
```

Syntax

Syntax der Funktion zur Berechnung des Mittelwertes der Werte einer Gleitkommawerttabelle:

Funktion	Syntax
MEAN	Result=Fonction(Op1)

Parameter der Funktion zur Berechnung einer bestimmten Anzahl L von Werten einer Gleitkommawerttabelle:

Operand (Op1)	Ergebnis (Result)
%MFi:L, %KFi:L	%MFi

Systembits und Systemworte

18

Auf einen Blick

Inhalt dieses Kapitels

Dieses Kapitel bietet einen Überblick über die Systembits und Systemworte, mit denen Sie Steuerungsprogramme für Twido Steuerungen erstellen können.

Inhalt dieses Kapitels

Dieses Kapitel enthält die folgenden Themen:

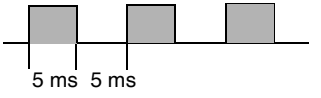
Thema	Seite
Systembits (%S)	620
Systemwörter (%SW)	633

Systembits (%S)

Einführung

Der folgende Abschnitt bietet ausführliche Informationen zur Funktion von Systembits und dazu, wie diese gesteuert werden.

Ausführliche Beschreibung Die folgende Tabelle bietet Informationen zu den Systembits sowie eine Beschreibung, wie diese gesteuert werden.

Systembit	Funktion	Beschreibung	Ausgangszustand	Steuerung
%S0	Kaltstart	Dieses normalerweise im Status 0 befindliche Bit wird auf 1 gesetzt: ● bei Spannungsrückkehr mit Datenverlust (Batteriefehler) ● durch das Anwenderprogramm oder den Animationstabellen-Editor ● durch das Bedienterminal Dieses Bit wird während des ersten kompletten Zyklus auf 1 gesetzt. Es wird vom System vor dem nächsten Zyklus auf 0 zurückgesetzt.	0	S oder U->S
%S1	Warmstart	Dieses normalerweise im Status 0 befindliche Bit wird auf 1 gesetzt: ● bei Spannungsrückkehr mit Datensicherung ● durch das Anwenderprogramm oder den Animationstabellen-Editor ● durch das Bedienterminal Es wird vom System am Ende des kompletten Zyklus auf 0 zurückgesetzt.	0	S oder U->S
%S4 %S5 %S6 %S7	Zeitbasis: 10 ms Zeitbasis: 100 ms Zeitbasis: 1 s Zeitbasis: 1 min	Die Zustandsänderungen dieser Bits werden durch eine interne Uhr gesteuert. Sie sind nicht mit dem Zyklus der Steuerung synchronisiert. Beispiel: %S4  	-	S

Systembit	Funktion	Beschreibung	Ausgangszustand	Steuerung
%S8	Verdrahtungsprüfung	Dieses ursprünglich im Status 1 befindliche Bit wird für die Prüfung der Verdrahtung verwendet, wenn sich die Steuerung TSX 37 im Status "nicht konfiguriert" befindet. Um den Wert dieses Bits zu ändern, verwenden Sie die Tasten des Bedienterminals, und ändern Sie dann den Zustand der gewünschten Ausgänge: ● Im Status 1, Rücksetzen der Ausgänge auf Null ● Im Status 0, Freigabe der Verkabelungsprüfung	1	U
%S9	Ausgänge zurücksetzen	Normalerweise auf 0 gesetzt. Es kann vom Programm oder vom PG (im Animationstabellen-Editor) auf 1 gesetzt werden: ● Im Status 1: Die Ausgänge werden auf 0 forciert, wenn die Steuerung im Betriebszustand RUN ist. ● Im Status 0: Die Ausgänge werden normal aktualisiert.	0	U
%S10	E/A-Fehler	Normalerweise auf 1 gesetzt. Dieses Bit kann vom System auf 0 gesetzt werden, wenn ein E/A-Fehler erkannt wird.	1	S
%S11	Watchdog-Überlauf	Normalerweise auf 0 gesetzt. Dieses Bit kann vom System auf 1 gesetzt werden, wenn die Programmausführungszeit (Zykluszeit) die maximale Zykluszeit (Software-Watchdog) überschreitet. Bei einem Watchdog-Überlauf geht die Steuerung in den Modus HALT über.	0	S
%S12	Steuerung im RUN-Modus	Dieses Bit reflektiert den Ausführungszustand der Steuerung. Das System setzt das Bit auf 1, wenn die Steuerung läuft. Bei einem Halt, während der Initialisierung des Systems sowie in jedem anderen Status wird dieses Bit auf 0 gesetzt.	0	S

Systembit	Funktion	Beschreibung	Ausgangszustand	Steuerung
%S13	Erster Zyklus im RUN-Modus	Dieses normalerweise im Status 0 befindliche Bit wird vom System während des ersten Zyklus nach Versetzen der Steuerung in den Modus RUN auf 1 gesetzt.	1	S
%S17	Leistungsüberschreitung	Dieses normalerweise im Status 0 befindliche Bit wird vom System auf 1 gesetzt: ● während einer Rotations- oder Schiebeoperation. Das System setzt den Bitausgang auf 1. Es muss nach jedem Befehl mit Überlaufrisiko vom Anwenderprogramm getestet und vom Anwender auf 0 zurückgesetzt werden, wenn ein Überlauf auftritt.	0	S->U
%S18	Überlauf oder arithmetischer Fehler	Normalerweise auf 0 gesetzt. Wird auf 1 gesetzt, wenn ein Überlauf bei Ausführung eines 16-Bit-Befehls auftritt, d.h.: ● bei einem Ergebnis über + 32.767 oder unter - 32.768 einfacher Länge ● bei einem Ergebnis über + 2.147.483.647 oder unter - 2.147.483.648 doppelter Länge ● bei einem Ergebnis über + 3,402824E+38 oder unter - 3,402824E+38 im Gleitkommaformat ● bei Division durch 0 ● bei Quadratwurzel einer negativen Zahl ● bei nicht signifikanter BTI- oder ITB-Konvertierung: BCD-Wert außerhalb des zulässigen Bereichs Es muss vom Anwender nach jedem Befehl mit Überlaufrisiko getestet und dann vom Anwender auf 0 zurückgesetzt werden, wenn ein Überlauf auftritt.	0	S->U

Systembit	Funktion	Beschreibung	Ausgangszustand	Steuerung
%S19	Zykluszeitüberlauf (periodische Abtastung)	Normalerweise auf 0. Dieses Bit wird vom System auf 1 gesetzt, wenn ein Zykluszeitüberlauf auftritt (Zykluszeit ist größer als die Dauer, die vom Anwender in der Konfiguration festgelegt oder in %SW0 programmiert wurde). Dieses Bit wird vom Anwender auf 0 zurückgesetzt.	0	S->U
%S20	Indexüberlauf	Dieses normalerweise auf 0 befindliche Bit wird auf 1 gesetzt, wenn die Adresse des indexierten Objekts kleiner als 0 oder größer als ihre maximale Größe wird. Es muss vom Anwenderprogramm nach jedem Befehl mit Überlaufrisiko getestet und auf 0 gesetzt werden, wenn ein Überlauf auftritt.	0	S->U
%S21	GRAFCET-Initialisierung	Dieses normalerweise im Status 0 befindliche Bit wird auf 1 gesetzt: ● durch einen Kaltstart, %S0=1 ● durch das Anwenderprogramm, nur im Vorverarbeitungsabschnitt des Programms, mittels der Anweisung 'Set' (S %S21) oder mittels einer SET-Spule -(S)- %S21. ● durch das PG Im Status 1 bewirkt es die GRAFCET-Initialisierung. Aktive Schritte werden deaktiviert, und die Ausgangsschritte werden aktiviert. Es wird vom System nach der GRAFCET-Initialisierung auf 0 gesetzt.	0	U->S
%S22	Rücksetzen des GRAFCET	Dieses normalerweise auf 0 gesetzte Bit kann nur während der Vorverarbeitung vom Programm auf 1 gesetzt werden. Im Status 1 bewirkt es, dass die aktiven Schritte im gesamten GRAFCET deaktiviert werden. Es wird vom System zu Beginn der Ausführung der Ablaufbearbeitung auf 0 zurückgesetzt.	0	U->S

Systembit	Funktion	Beschreibung	Ausgangszustand	Steuerung
%S23	GRAFCET voreinstellen und einfrieren	Dieses normalerweise im Status 0 befindliche Bit kann nur im Programmmodul für die Vorbearbeitung auf 1 gesetzt werden. Im Status 1 validiert es die Voreinstellung des GRAFCET. Wird dieses Bit auf 1 gehalten, wird GRAFCET eingefroren (der Graph wird eingefroren). Es wird vom System zu Beginn der Ausführung der Ablaufbearbeitung auf 0 zurückgesetzt, um die Entwicklung des GRAFCET ausgehend von der eingefrorenen Situation sicherzustellen.	0	U->S
%S24	Bedienterminal	Dieses normalerweise im Status 0 befindliche Bit kann vom Benutzer auf 1 gesetzt werden. ● Im Status 0 funktioniert das Bedienterminal normal. ● Im Status 1 ist das Bedienterminal eingefroren, die aktuelle Anzeige bleibt erhalten, das Blinken ist deaktiviert, und Tastenbetätigungen werden nicht mehr berücksichtigt.	0	U->S

Systembit	Funktion	Beschreibung	Ausgangszustand	Steuerung
%S25	Auswählen eines Anzeigemodus am Bedienterminal	Sie können zwischen zwei Anzeigemodi am zweizeiligen Bedienterminal wählen: "Datenmodus" und "Normaler Modus". ● Wenn %S25=0, dann ist der normale Modus aktiviert. In die erste Zeile können Sie einen Objektnamen schreiben (ein Systemwort, ein Speicherwort, ein Systembit etc.). In der zweiten Zeile können Sie seinen Wert lesen. ● Wenn %S25=1, dann ist der Datenmodus aktiviert. In der ersten Zeile können Sie den Wert %SW68 anzeigen. In der zweiten Zeile können Sie den Wert %SW69 anzeigen. Wenn %S25=1, dann ist die Bedienertastatur deaktiviert. **Hinweis:** Die Firmwareversion muss V3.0 oder höher sein.	0	U
%S26	Auswählen eines Werts mit oder ohne Vorzeichen am Bedienterminal	Sie können zwischen zwei Wertetypen auswählen: Mit oder ohne Vorzeichen. ● Wenn %S26=0, dann ist die Anzeige von Werten mit Vorzeichen (-32768 bis 32767) aktiviert. Die Zeichen +/- werden am Anfang jeder Zeile angezeigt. ● Wenn %S26=1, dann ist die Anzeige von Werte ohne Vorzeichen (0 bis 65535) aktiviert. %S26 kann nur verwendet werden, wenn %S25=1. **Hinweis:** Die Firmwareversion muss V3.0 oder höher sein.	0	U

Systembit	Funktion	Beschreibung	Ausgangszustand	Steuerung
%S31	Ereignismaske	Normalerweise auf 1. ● Im Status 0 können Ereignisse nicht ausgeführt werden, und werden in die Warteschlange gestellt. ● Im Status 1 können Ereignisse ausgeführt werden. Dieses Bit kann vom Benutzer und vom System (bei einem Kaltstart) in seinem Initialzustand von 1 gesetzt werden.	1	U->S
%S38	Berechtigung der Ereignisse, sich in die Ereigniswarteschlange einzureihen	Normalerweise auf 1. ● Im Status 0 können Ereignisse nicht in die Warteschlange eingereiht werden. ● Im Status 1 werden die Ereignisse in die Ereigniswarteschlange eingereiht, sobald sie erkannt werden. Dieses Bit kann vom Benutzer und vom System (bei einem Kaltstart) in seinem Initialzustand von 1 gesetzt werden.	1	U->S
%S39	Sättigungszustand der Ereigniswarteschlange	Normalerweise auf 0. ● Im Status 0 werden alle Ereignisse gemeldet. ● Im Status 1 geht mindestens ein Ereignis verloren. Dieses Bit kann vom Benutzer und vom System (bei einem Kaltstart) auf 0 initialisiert werden.	0	U->S
%S50	Aktualisierung des Datums und der Uhrzeit mit den Wörtern %SW49 bis %SW53	Dieses normalerweise im Status 0 befindliche Bit kann vom Programm oder vom Bedienterminal auf 1 oder 0 gesetzt werden. ● Im Status 0 können das Datum und die Uhrzeit gelesen werden. ● Im Status 1 können das Datum und die Uhrzeit aktualisiert werden. Bei fallender Flanke von %S50 wird die interne Echtzeituhr der Steuerung aktualisiert.	0	U->S

Systembit	Funktion	Beschreibung	Ausgangszustand	Steuerung
%S51	Status der Echtzeituhr	Dieses normalerweise im Status 0 befindliche Bit kann vom Programm oder vom Bedienterminal auf 1 oder 0 gesetzt werden. ● Im Zustand 0 sind das Datum und die Uhrzeit kohärent. ● Im Zustand 1 müssen das Datum und die Uhrzeit vom Anwender initialisiert werden. Wenn dieses Bit auf 1 gesetzt ist, sind die Daten der Echtzeituhr ungültig. Es ist möglich, dass das Datum und die Uhrzeit nie konfiguriert worden sind, dass die Batterie zu schwach ist, oder dass die Korrekturkonstante der Steuerung ungültig ist (nie konfiguriert worden ist, Unterschied zwischen der Korrektur der Echtzeituhr und dem gespeicherten Wert oder Wert außerhalb des zulässigen Bereichs). Bei einem Zustandswechsel von 1 auf 0 wird das Schreiben der Korrekturkonstante in die Echtzeituhr forciert.	0	U->S
%S52	Echtzeituhr fehlerhaft	Dieses vom System generierte Bit gibt an, dass die Korrektur der Echtzeituhr nicht eingegeben wurde und dass das Datum und die Uhrzeit falsch sind. ● Im Zustand 0 sind das Datum und die Uhrzeit kohärent. ● Im Zustand 1 müssen das Datum und die Uhrzeit initialisiert werden.	0	S
%S59	Aktualisierung des Datums und der Uhrzeit mit dem Wort %SW59	Dieses normalerweise im Status 0 befindliche Bit kann vom Programm oder vom Bedienterminal auf 1 oder 0 gesetzt werden. ● Im Status 0 wird das Systemwort %SW59 nicht verwaltet. ● Im Status 1 werden das Datum und die Uhrzeit abhängig von den steigenden Flanken der in %SW59 gesetzten Steuerbits inkrementiert oder dekrementiert.	0	U

Systembit	Funktion	Beschreibung	Ausgangszustand	Steuerung
%S66	BAT LED-Anzeige aktivieren/deaktivieren (nur an Steuerungen, die eine externe Batterie unterstützen: TWDLCA•40DRF-Steuerungen)	Dieses Systembit kann vom Benutzer gesetzt werden. Es ermöglicht das Ein-/Ausschalten der BAT LED: ● Wenn das Systembit auf 0 gesetzt ist, ist die BAT LED aktiviert (es wird beim Einschalten vom System auf 0 zurückgesetzt). ● Wenn das Systembit auf 1 gesetzt ist, ist die BAT LED deaktiviert (die LED bleibt auch dann aus, wenn die externe Batterie schwach ist oder keine externe Batterie im Batteriefach eingesetzt ist).	0	S oder U->S
%S69	LED-Anzeige STAT	Im Status 0 ist die LED STAT ausgeschaltet. Im Status 1 ist die LED STAT eingeschaltet.	0	U
%S75	Status der externen Batterie (nur an Steuerungen, die eine externe Batterie unterstützen: TWDLCA•40DRF-Steuerungen)	Dieses Systembit wird vom System gesetzt. Es zeigt an, dass der Status der externen Batterie vom Benutzer gelesen werden kann: ● Wenn dieses Bit auf 0 gesetzt ist, funktioniert die externe Batterie normal. ● Wenn dieses Bit auf 1 gesetzt ist, ist die externe Batterie schwach oder es ist keine externe Batterie im Batteriefach einlegt.	0	S
%S95	Wiederherstellung der Speicherwörter	Dieses Bit kann gesetzt werden, wenn zuvor Speicherwörter im internen EEPROM gespeichert wurden. Nach Beendigung des Vorgangs setzt das System dieses Bit auf 0 zurück. Die Anzahl der wiederhergestellten Speicherwörter wird in %SW97 gesetzt.	0	U
%S96	Backup-Programm OK	Dieses Bit kann jederzeit gelesen werden (entweder vom Programm oder während einer Einstellung), insbesondere nach einem Kaltstart oder einem Warmstart. ● Im Status 0 ist das Backup-Programm ungültig. ● Im Status 1 ist das Backup-Programm gültig.	0	S

Systembit	Funktion	Beschreibung	Ausgangszustand	Steuerung
%S97	Speichern von %MW OK	Dieses Bit kann jederzeit gelesen werden (entweder vom Programm oder während einer Einstellung), insbesondere nach einem Kaltstart oder einem Warmstart. ● Im Status 0 ist das Speichern von %MW nicht OK. ● Im Status 1 ist das Speichern von %MW OK.	0	S
%S100	Anschluss des TwidoSoft-Kommunikationskabels	Gibt an, ob das TwidoSoft Kommunikationskabel angeschlossen ist. ● Im Status 1 ist das TwidoSoft Kommunikationskabel entweder nicht angeschlossen oder TwidoSoft ist angeschlossen. ● Im Status 0 ist das TwidoSoft Kabel für die dezentrale Verbindung angeschlossen.	-	S
%S101	Änderung der Adresse eines Ports (Modbus-Protokoll)	Wird verwendet, um eine Port-Adresse mittels der Systemwörter %SW101 (Port 1) und %SW102 (Port 2) zu ändern. Hierzu muss %S101 auf 1 gesetzt werden. ● Im Status 0 kann die Adresse nicht geändert werden. Der Wert von %SW101 und %SW102 entspricht der aktuellen Adresse des Ports. ● Im Status 1 ist es möglich, die Adresse durch Änderung der Werte von %SW101 (Port 1) und %SW102 (Port 2) zu ändern. Nach der Änderung der Werte der Systemwörter muss %S101 wieder auf 0 zurückgesetzt werden.	0	U

Systembit	Funktion	Beschreibung	Ausgangszustand	Steuerung
%S103 %S104	Verwendung des ASCII-Protokolls	Ermöglicht die Verwendung des ASCII-Protokolls an COM 1 (%S103) oder COM 2 (%S104). Das ASCII-Protokoll wird mittels der Systemwörter %SW103 und %SW105 für COM 1 bzw. %SW104 und %SW106 für COM 2 konfiguriert. ● Im Status 0 ist das verwendete Protokoll das in Twido Soft konfigurierte Protokoll. ● Im Status 1 wird das ASCII-Protokoll an COM 1 (%S103) oder COM 2 (%S104) verwendet. In diesem Fall müssen zuvor die Systemwörter %SW103 und %SW105 für COM 1 bzw. %SW104 und %SW106 für COM 2 konfiguriert werden.	0	U
%S110	Datenaustausch über dezentrale Verbindung	Dieses Bit wird vom Programm oder vom PG auf 0 zurückgesetzt. ● Im Status 1 für einen Master ist der gesamte Datenaustausch über die dezentrale Verbindung (nur dezentrale E/A) beendet. ● Im Status 1 für einen Slave ist der Austausch mit dem Master beendet.	0	S->U
%S111	Einzelner Datenaustausch über dezentrale Verbindung	● Im Status 0 für einen Master ist ein einzelner Datenaustausch über eine dezentrale Verbindung beendet. ● Im Status 1 für einen Master ist ein einzelner Datenaustausch über eine dezentrale Verbindung aktiv.	0	S
%S112	Verbindung über dezentrale Verbindung	● Im Status 0 für einen Master ist die dezentrale Verbindung aktiviert. ● Im Status 1 für einen Master ist die dezentrale Verbindung deaktiviert.	0	U

Systembit	Funktion	Beschreibung	Ausgangszustand	Steuerung
%S113	Konfiguration/Betrieb der dezentralen Verbindung	• Im Status 0 für einen Master oder Slave ist die Konfiguration/der Betrieb der dezentralen Verbindung OK. • Im Status 1 für einen Master ist in der Konfiguration/im Betrieb der dezentralen Verbindung ein Fehler aufgetreten. • Im Status 1 für einen Slave ist in der Konfiguration/im Betrieb der dezentralen Verbindung ein Fehler aufgetreten.	0	S->U
%S118	Fehler dezentrale E/A	Normalerweise auf 1 gesetzt. Dieses Bit kann auf 0 gesetzt werden, wenn ein E/A-Fehler in der dezentralen Verbindung erkannt wird.	1	S
%S119	Fehler lokale E/A	Normalerweise auf 1 gesetzt. Dieses Bit kann auf 0 gesetzt werden, wenn ein E/A-Fehler in der dezentralen Verbindung erkannt wird. %SW118 gibt die Art des Fehlers an. Wird auf 1 zurückgesetzt, wenn der Fehler behoben wurde.	1	S

Beschreibung der in der Tabelle verwendeten Abkürzungen

Tabelle der Abkürzungen

Abkürzung	Beschreibung
S	Vom System gesteuert
U	Vom Benutzer gesteuert
U->S	Vom Benutzer auf 1 gesetzt, vom System auf 0 zurückgesetzt
S->U	Vom System auf 1 gesetzt, vom Benutzer auf 0 zurückgesetzt

Systemwörter (%SW)

Einführung Der folgende Abschnitt bietet ausführliche Informationen zur Funktion von Systemwörtern und dazu, wie diese gesteuert werden.

Ausführliche Beschreibung Die folgende Tabelle bietet ausführliche Informationen zur Funktion von Systemwörtern und dazu, wie diese gesteuert werden.

Systemwörter	Funktion	Beschreibung	Steuerung
%SW0	Zykluszeit der Steuerung (periodische Task)	Ändert die vom Anwenderprogramm in der Konfiguration definierte Zykluszeit der Steuerung im Animationstabellen-Editor	U
%SW1	Wert eines periodischen Ereignisses speichern	Ändert die Zykluszeit [5-255 ms] eines periodischen Ereignisses, ohne den im Fenster "Abtastmodus" im Feld "Periodisches Ereignis" gespeicherten Periodenwert zu verlieren. Ermöglicht Ihnen, den im Feld "Periodisches Ereignis" gespeicherten Periodenwert wiederherzustellen: • bei einem Kaltstart oder • wenn der Wert, den Sie in %SW1 schreiben, außerhalb des Bereichs [5-255] liegt. Der Wert %SW1 kann am Ende jedes Zyklus im Programm oder in der Animationstabelle geändert werden, ohne dass hierzu das Programm gestoppt werden muss. Die Zykluszeiten können während der Programmausführung korrekt überwacht werden.	U
%SW6	Steuerungszustand	Steuerungszustand: 0 = NO CONFIG 2 = STOPP 3 = RUN 4 = HALT	S

Systemwörter	Funktion	Beschreibung	Steuerung

%SW7	Steuerungszustand	• Bit [0]: Backup/Wiederherstellung läuft:
		  • im Status 1, wenn Backup/Wiederherstellung läuft
		  • im Status 0, wenn Backup/Wiederherstellung beendet oder nicht aktiv ist
		• Bit [1]: Konfiguration der Steuerung OK:
		  • im Status 1, wenn Konfiguration OK
		• Bit [3..2] EEPROM-Statusbits:
		  • 00 = Kein Speichermodul
		  • 01 = EEPROM-Speichermodul, 32 KB
		  • 10 = EEPROM-Speichermodul, 64 KB
		  • 11 = Für zukünftige Verwendung reserviert
		• Bit [4]: Applikation im RAM nicht mit dem EEPROM identisch:
		  • im Status 1, wenn Applikation im RAM nicht mit dem EEPROM identisch
		• Bit [5]: Applikation im RAM nicht mit der Kassette identisch:
		  • im Status 1, wenn Applikation im RAM nicht mit der Kassette identisch
		• Bit [6] nicht verwendet (Status 0)
		• Bit [7]: Steuerung reserviert:
		  • im Status 1, wenn reserviert
		• Bit [8]: Applikation im Schreibmodus:
		  • im Status 1, wenn die Applikation geschützt ist
		• Bit [9] nicht verwendet (Status 0)
		• Bit [10]: Zweiter serieller Port installiert:
		  • im Status 1, wenn installiert
		• Bit [11]: Zweiter serieller Port: (0 = EIA RS-232, 1 = EIA RS-485):
		  • im Status 0 = EIA RS-232
		  • im Status 1 = EIA RS-485
		• Bit [12]: Gültige Applikation im internen Speicher:
		  • im Status 1, wenn Applikation gültig
		• Bit [13] Gültige Applikation in der Sicherungskassette:
		  • im Status 1, wenn Applikation gültig
		• Bit [14] Gültige Applikation im RAM:
		  • im Status 1, wenn Applikation gültig
		• Bit [15]: betriebsbereit:
		  • im Status 1, wenn betriebsbereit
			S

%SW11	Wert des Software-Watchdogs	Enthält den maximalen Watchdog-Wert. Der Wert (10 bis 500 ms) wird in der Konfiguration definiert.	U
%SW14	Handelsversion, Vxx.yy	Zum Beispiel, wenn %SW14=0232: • 8 MSB=02 hexadezimal, dann xx=2 dezimal • 8 LSB=32 hexadezimal, dann yy=50 dezimal Folglich ist die Handelsversion V2.50. **Hinweis:**Die Firmwareversion muss V2.5 oder höher sein.	S

Systemwörter	Funktion	Beschreibung	Steuerung
%SW15	Firmware-Patch, Pzz	Zum Beispiel, wenn %SW15=0005: ● 8 MSB wird nicht verwendet. ● 8 LSB=05 hexadezimal, dann zz=5 dezimal Folglich ist das Firmware-Patch P05. **Hinweis:**Die Firmwareversion muss V2.5 oder höher sein.	S
%SW16	Firmwareversion, Vxx.yy	Zum Beispiel, wenn %SW16=0232: ● 8 MSB=02 hexadezimal, dann xx=2 dezimal ● 8 LSB=32 hexadezimal, dann yy=50 dezimal Folglich ist die Firmwareversion V2.50. **Hinweis:**Die Firmwareversion muss V2.5 oder höher sein.	S
%SW17	Fehlerstatus bei Gleitkommaoperation	Bei Erkennung eines Fehlers in einer Gleitkommaarithmetikoperation wird das Bit %S18 auf 1 gesetzt, und der Fehlerstatus %SW17 wird gemäß folgender Codierung aktualisiert: ● Bit [0]: ungültige Operation, das Ergebnis ist keine Zahl (1.#NAN oder -1.#NAN) ● Bit 1: reserviert ● Bit 2: Division durch 0, das Ergebnis ist unendlich (-1.#INF oder 1.#INF) ● Bit 3: absoluter Wert des Ergebnisses höher als +3,402824e+38, das Ergebnis ist unendlich (-1.#INF oder 1.#INF)	S und U
%SW18- %SW19	Absoluttimer mit 100-ms-Zähler	Der Zähler arbeitet mit zwei Wörtern: ● **%SW18** stellt das niederwertige Wort dar ● **%SW19** stellt das höherwertige Wort dar	S und U
%SW20 bis %SW27	Gibt den Status von CANopen-Slaves mit der Knotenadresse 1 bis 16 an.	Ausführliche Informationen finden Sie unter *Spezifische, für den CANopen-Slave reservierte Systemwörter , S. 281*.	S
%SW30	Letzte Zykluszeit	Zeigt die Ausführungszeit des letzten Zyklus der Steuerung an (in ms). **Hinweis:** Diese Zeit entspricht der Zeit, die zwischen dem Beginn (Erfassung der Eingänge) und dem Ende (Aktualisierung der Ausgänge) eines Zyklus verstrichen ist.	S

Systemwörter	Funktion	Beschreibung	Steuerung
%SW31	Maximale Zykluszeit	Zeigt die Ausführungszeit des längsten Zyklus der Steuerung seit dem letzten Kaltstart an (in ms). **Hinweise:** ● Diese Zeit entspricht der Zeit, die zwischen dem Beginn (Erfassung der Eingänge) und dem Ende (Aktualisierung der Ausgänge) eines Zyklus verstrichen ist. ● Um eine ordnungsgemäße Erkennung eines Impulssignals zu gewährleisten, wenn die Option "Eingangsverriegelung" ausgewählt ist, müssen die Impulsbreite (T_{ON}) und die Zykluszeit (T_{pulse}) die beiden folgenden Bedingungen erfüllen: ● $T_{ON} \geq 1$ ms ● Die Zykluszeit des Eingangssignals muss der Nyquist-Shannon-Abtastregel entsprechen, die besagt, dass die Zykluszeit (T_{pulse}) des Eingangssignals mindestens dem doppelten Wert der maximalen Programmzykluszeit (%SW31) entsprechen muss: $T_{pulse} \geq 2$ x %SW31. **Hinweis:** Wenn diese Bedingung nicht erfüllt ist, können einige Impulse verloren gehen.	S
%SW32	Minimale Zykluszeit	Zeigt die Ausführungszeit des kürzesten Zyklus der Steuerung seit dem letzten Kaltstart an (in ms). **Hinweis:** Diese Zeit entspricht der Zeit, die zwischen dem Beginn (Erfassung der Eingänge) und dem Ende (Aktualisierung der Ausgänge) eines Zyklus verstrichen ist.	S
%SW48	Anzahl der Ereignisse	Zeigt die Anzahl der Ereignisse an, die seit dem letzten Kaltstart ausgeführt wurden. (Zählt alle Ereignisse mit Ausnahme von periodischen Ereignissen.) **Hinweis:** Auf 0 initialisiert (nach Laden der Applikation und Kaltstart), wird mit jeder Ereignisausführung inkrementiert.	S

Systemwörter	Funktion	Beschreibung		Steuerung
%SW49 %SW50 %SW51 %SW52 %SW53	Echtzeituhr (RTC)	Echtzeituhr-Funktion (RTC): Wörter, die das aktuelle Datum und die aktuelle Uhrzeit enthalten (in BCD):		S und U
		%SW49	xN Wochentag (N=1 für Montag)	
		%SW50	00SS Sekunden	
		%SW51	HHMM Stunde und Minuten	
		%SW52	MMJJ Monat und Tag	
		%SW53	SSAA Jahrhundert und Jahr	
		Diese Wörter werden vom System gesteuert, wenn das Bit **%S50** auf 0 gesetzt ist. Diese Wörter können vom Anwenderprogramm oder vom PG geschrieben werden, wenn das Bit **%S50** auf 1 gesetzt ist. Bei fallender Flanke von **%S50** wird die interne Echtzeituhr der Steuerung ausgehend von den in diese Wörter geschriebenen Werten aktualisiert.		
%SW54 %SW55 %SW56 %SW57	Datum und Uhrzeit des letzten Stopps	Systemwörter, die die Uhrzeit und das Datum des letzten Spannungsverlusts bzw. des letzten Steuerungsstopps enthalten (in BCD):		S
		%SW54	SS Sekunden	
		%SW55	HHMM Stunde und Minuten	
		%SW56	MMJJ Monat und Tag	
		%SW57	SSAA Jahrhundert und Jahr	
%SW58	Code des letzten Stopp	Zeigt Code an, der die Ursache für den letzten Stopp angibt:		S
		1 =	Eingangsflanke Run/Stop	
		2 =	Stopp wegen Software-Fehler (Zyklusüberschreitung der Steuerung)	
		3 =	Stoppbefehl	
		4 =	Spannungsausfall	
		5 =	Stopp wegen Hardware-Fehler	

Systemwort	Funktion	Beschreibung			Steuerung
%SW59	Aktuelles Datum einstellen	Passt das aktuelle Datum an. Enthält zwei Sätze aus 8 Bits, um das aktuelle Datum anzupassen. Dieser Vorgang wird immer bei einer steigenden Flanke des Bits durchgeführt. Dieses Wort wird aktiviert von Bit **%S59**.			U
		Inkrementieren	**Dekrementieren**	**Parameter**	
		Bit 0	Bit 8	Wochentag	
		Bit 1	Bit 9	Sekunden	
		Bit 2	Bit 10	Minuten	
		Bit 3	Bit 11	Stunden	
		Bit 4	Bit 12	Tage	
		Bit 5	Bit 13	Monat	
		Bit 6	Bit 14	Jahre	
		Bit 7	Bit 15	Jahrhunderte	
%SW60	Korrektur der Echtzeituhr	Echtzeituhr-Korrekturwert			U
%SW63	Fehlercode EXCH1-Block	Fehlercode EXCH1: 0 - Vorgang war erfolgreich 1 - Anzahl zu sendender Bytes zu hoch (> 250) 2 - Sendetabelle zu klein 3 - Worttabelle zu klein 4 - Überlauf Empfangstabelle 5 - Timeout abgelaufen 6 - Senden 7 - Falscher Befehl in Tabelle 8 - Ausgewählter Port nicht konfiguriert/verfügbar 9 - Empfangsfehler 10 - %KW kann nicht beim Empfang verwendet werden 11 - Sende-Offset größer als Sendetabelle 12 - Empfangs-Offset größer als Empfangstabelle 13 - Steuerung hat Bearbeitung von EXCH gestoppt			S
%SW64	Fehlercode EXCH2-Block	Fehlercode EXCH2: Siehe %SW63.			S

Systemwort	Funktion	Beschreibung	Steuerung
%SW65	Fehlercode EXCH3-Block	Der Fehlercode EXCH3 ist nur auf Ethernet-fähigen TWDLCAE40DRF Twido-Steuerungen implementiert. 1-4, 6-13: Siehe %SW63. (Bitte beachten Sie, dass Fehlercode 5 ungültig ist und durch die nachfolgend beschriebenen, Ethernet-spezifischen Fehlercodes 109 und 122 ersetzt wurde.) Die nachfolgend aufgeführten Codes sind Ethernet-spezifische Fehlercodes: 101 - keine derartige IP-Adresse 102 - Die TCP-Verbindung ist unterbrochen 103 - Kein Socket verfügbar (alle Verbindungskanäle sind belegt) 104 - Netzwerk ausgefallen 105 - Keine Verbindung zum Netzwerk möglich 106 - Unterbrechung der Netzwerkverbindung beim Reset 107 - Verbindung durch Partnergerät getrennt 108 - Verbindung durch Partnergerät zurückgesetzt 109 - Verbindungs-Timeout abgelaufen 110 - Zurückweisung bei Verbindungsaufbau 111 - Host ausgefallen 120 - Unbekannter Index (das dezentrale Gerät ist in der Konfigurationstabelle nicht indexiert) 121 - schwer (MAC, Chip, doppelte IP)122 - Empfangs-Timeout nach dem Senden der Daten abgelaufen 123 - Ethernet-Initialisierung läuft	S
%SW67	Funktion und Typ der Steuerung	Enthält die folgenden Informationen: • Bits für Steuerungstyp [0 -11] • 8B0 = TWDLC•A10DRF • 8B1 = TWDLC•A10DRF • 8B2 = TWDLMDA20DUK/DTK • 8B3 = TWDLC•A24DRF • 8B2 = TWDLMDA20DUK/DTK • 8B6 = TWDLMDA20DRT • 8B8 = TWDLCAA40DRF • 8B9 = TWDLCAE40DRF • Bit 12, 13, 14, 15 nicht verwendet = 0	S

Systemwörter	Funktion	Beschreibung	Steuerung
%SW68 und %SW69	Simultan am zweizeiligen Bedienterminal anzuzeigende Elemente	Wenn %S25=1, dann ist der Datenmodus aktiviert. Die Bedienertastatur ist deaktiviert. %SW68 und %SW69 können am zweizeiligen Bedienterminal angezeigt werden: • Wert %SW68 in der ersten Zeile, • Wert %SW69 in der zweiten Zeile. **Hinweis:** Die Firmwareversion muss V3.0 oder höher sein.	U
%SW73 und %SW74	Status des AS-Interface-Systems	• Bit [0]: im Status 1, wenn Konfiguration OK • Bit [1]: im Status 1, wenn Datenaustausch aktiviert • Bit [2]: im Status 1, wenn Modul im Offline-Modus • Bit [3]: im Status 1, wenn Anweisung ASI_CMD beendet • Bit [4]: im Status 1, wenn Anweisung ASI_CMD fehlerhaft	S und U
%SW76 bis %SW79	Abwärtszähler 1-4	Diese 4 Wörter dienen als 1-ms-Timer. Sie werden vom System jede ms einzeln dekrementiert, wenn sie einen positiven Wert besitzen. So stehen 4 Abwärtszähler zum Abwärtszählen in ms zur Verfügung, was einem Betriebsbereich von 1 ms bis 32767 ms entspricht. Wird Bit 15 auf 1 gesetzt, kann das Dekrementieren gestoppt werden.	S und U
%SW80	E/A-Status Grundgerät	Bit [0] Kanäle im Normalbetrieb (für alle Kanäle) Bit [1] Modul wird initialisiert (oder Initialisierungsinformationen für alle Kanäle) Bit [2] Hardwarestörung (Ausfall der externen Stromversorgung, für alle Kanäle gemeinsam) Bit [3] Modulkonfigurationsfehler Bit [4] Datenkonvertierung Eingangskanal 0 läuft Bit [5] Datenkonvertierung Eingangskanal 1 läuft Bit [6] Eingang Thermoelementkanal 0 nicht konfiguriert Bit [7] Eingang Thermoelementkanal 1 nicht konfiguriert Bit [8] Frei Bit [9] Nicht verwendet Bit [10] Bereichsüberschreitung Datenkanal Analogeingang 0 Bit [11] Bereichsüberschreitung Datenkanal Analogeingang 1 Bit [12] Falsche Verdrahtung (Bereichsunterschreitung Datenkanal Analogeingang 0, Stromschleife offen) Bit [13] Falsche Verdrahtung (Bereichsunterschreitung Datenkanal Analogeingang 1, Stromschleife offen) Bit [14] Frei Bit [15] Ausgangskanal nicht verfügbar	S

Systemwörter	Funktion	Beschreibung	Steuerung
%SW81	• Status Erweiterungs-E/A-Modul 1: Gleiche Definitionen wie %SW80 • Status des CANopen-Mastermoduls an Erweiterungsadresse 1: • Bit [0] Konfigurationsstatus (1 = Konfiguration OK; 0 = Konfigurationsfehler) Bit [1] Betriebsstatus (1 = PDO-Austausch AN; 0 = PDO-Austausch AUS) Bit [2] Ausgangszustand (1 = Ausgangszustand AN; 0 = Ausgangszustand AUS) Bit [3] CAN_CMD-Anweisung vollständig (1 = vollständig; 0 = wird ausgeführt) Bit [4] CAN_CMD-Anweisungsfehler (1 = Fehler; 0 = OK) Bit [5] Initialisierungsfehler (1 = Fehler; 0 = OK) Bit [6] Verlust der Meldung, Stromversorgungsfehler (1 = Fehler; 0 = OK)		S
%SW82	Status Erweiterungs-E/A-Modul 2: Gleiche Definitionen wie %SW80 Status des CANopen-Mastermoduls an Erweiterungsadresse 2: Gleiche Definitionen wie %SW81		S
%SW83	Status Erweiterungs-E/A-Modul 3: Gleiche Definitionen wie %SW80 Status des CANopen-Mastermoduls an Erweiterungsadresse 3: Gleiche Definitionen wie %SW81		S
%SW84	Status Erweiterungs-E/A-Modul 4: Gleiche Definitionen wie %SW80 Status des CANopen-Mastermoduls an Erweiterungsadresse 4: Gleiche Definitionen wie %SW81		S
%SW85	Status Erweiterungs-E/A-Modul 5: Gleiche Definitionen wie %SW80 Status des CANopen-Mastermoduls an Erweiterungsadresse 5: Gleiche Definitionen wie %SW81		S
%SW86	Status Erweiterungs-E/A-Modul 6: Gleiche Definitionen wie %SW80 Status des CANopen-Mastermoduls an Erweiterungsadresse 6: Gleiche Definitionen wie %SW81		S
%SW87	Status Erweiterungs-E/A-Modul 7: Gleiche Definitionen wie %SW80 Status des CANopen-Mastermoduls an Erweiterungsadresse 7: Gleiche Definitionen wie %SW81		S
%SW94	Signatur der Applikation	Wenn sich eine Applikation hinsichtlich der Konfigurations- oder Programmierungsdaten ändert, ändert sich folglich auch die Signatur (Summe aller Prüfsummen). Wenn %SW94=91F3 in hexadezimaler Schreibweise, dann lautet die Signatur in hexadezimaler Schreibweise 91F3. **Hinweis:** Die Firmwareversion muss V2.5 oder höher sein.	S

Systemwörter	Funktion	Beschreibung	Steuerung

%SW96	Befehl und/oder Diagnose für die Funktion zum Speichern/ Wiederherstellen des Anwendungsprogramms und von %MW.	• Bit [0]: gibt an, dass die Speicherwörter %MWi im EEPROM gesichert werden müssen:
		  • im Status 1, wenn ein Backup angefordert ist
		  • im Status 0, wenn das laufende Backup nicht abgeschlossen ist
		• Bit [1]: Dieses Bit wird von der Firmware gesetzt, um anzuzeigen, wann der Speichervorgang beendet ist:
		  • im Status 1, wenn das Backup beendet ist
		  • im Status 0, wenn ein neues Backup angefordert worden ist
		• Bit [2]: Speicherfehler, weitere Informationen finden Sie bei den Beschreibungen der Bits 8, 9, 10 und 14:
		  • im Status 1, wenn ein Fehler aufgetreten ist
		  • im Status 0, wenn ein neues Backup angefordert worden ist
		• Bit [6]: im Status 1, wenn die Steuerung eine gültige Applikation enthält.
		• Bit [8]: Gibt an, dass die Anzahl von %MWs, die in %SW97 angegeben ist, größer ist als die Anzahl von %MWs, die in der Applikation konfiguriert ist:
		  • im Status 1, wenn der Fehler erkannt wurde
		• Bit [9]: Gibt an, dass die Anzahl von %MWs, die in %SW97 angegeben ist, größer ist als die maximale Anzahl von %MWs, die in einer TwidoSoft Applikation definiert werden kann.
		  • im Status 1, wenn der Fehler erkannt wurde
		• Bit [10]: Unterschied zwischen internem RAM und internem EEPROM (1 = ja).
		  • im Status 1, wenn es einen Unterschied gibt.
		• Bit [14]: Gibt an, wenn ein EEPROM-Schreibfehler aufgetreten ist:
		  • im Status 1, wenn der Fehler erkannt wurde
			S und U

%SW97	Befehle oder Diagnose für die Funktion Speichern/ Wiederherstellen	Beim Speichern von Speicherwörtern stellt dieser Wert die physikalische Anzahl von %MW dar, die im internen EEPROM gespeichert werden soll. Beim Wiederherstellen von Speicherwörtern wird dieser Wert mit der Anzahl der Speicherwörter aktualisiert, die im RAM wiederhergestellt wird. Ist dieser Wert für die Speicherfunktion auf 0 gesetzt, werden keine Speicherwörter gespeichert. Der Benutzer muss das Anwenderprogramm definieren. Ansonsten wird es in der Steuerungsapplikation auf 0 gesetzt, außer im folgenden Fall: Bei Kaltstart wird dieses Wort auf -1 gesetzt, wenn im internen EEPROM kein **%MW**-Speicherwort vorhanden ist. Enthält der interne Flash-EEPROM bei Kaltstart eine Liste mit **%MW**-Sicherwörtern, muss die Anzahl der in der Datei gespeicherten Speicherwörter in das Systemwort **%SW97** geschrieben werden.	S und U

<table>
<tr><th>Systemwörter</th><th>Funktion</th><th>Beschreibung</th><th>Steuerung</th></tr>
<tr><td>%SW101
%SW102</td><td>Wert der Modbus-Adresse des Ports</td><td>Wenn Bit %S101 auf 1 gesetzt ist, können Sie die Modbus-Adresse von Port 1 oder Port 2 ändern. Die Adresse von Port 1 ist %SW101 und die von Port 2 ist %SW102.</td><td>S</td></tr>
<tr><td>%SW103
%SW104</td><td>Konfiguration für die Nutzung des ASCII-Protokolls</td><td>Wenn das Bit %S103 (COM 1) oder %S104 (COM 2) auf 1 gesetzt ist, wird das ASCII-Protokoll verwendet. Das Systemwort %SW103 (COM 1) oder %SW104 (COM 2) muss abhängig von den nachfolgend aufgeführten Elementen parametriert werden:
<table>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="8">Ende der Zeichenkette</td><td>Daten-bit</td><td>Stoppbit</td><td colspan="2">Parität</td><td>RTS / CTS</td><td colspan="3">Baudrate</td></tr>
</table>

Baudrate:
0: 1200 Baud
1: 2400 Baud
2: 4800 Baud
3: 9600 Baud
4: 19200 Baud
5: 38400 Bauds
RTS/CTS:
0: deaktiviert
1: aktiviert
Parität:
00: Keine
10: Ungerade
11: Gerade
Stoppbit:
0: 1 Stoppbit
1: 2 Stoppbits
Datenbits:
0: 7 Datenbits
1: 8 Datenbits
</td><td>S</td></tr>
<tr><td>%SW105
%SW106</td><td>Konfiguration für die Nutzung des ASCII-Protokolls</td><td>Wenn das Bit %S103 (COM 1) oder %S104 (COM 2) auf 1 gesetzt ist, wird das ASCII-Protokoll verwendet. Das Systemwort %SW105 (COM 1) oder %SW106 (COM 2) muss abhängig von den nachfolgend aufgeführten Elementen parametriert werden:
<table>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="8">Timeout-Frame in ms</td><td colspan="8">Timeout-Reaktion
als Vielfaches von 100 ms</td></tr>
</table></td><td>S</td></tr>
</table>

Systemwörter	Funktion	Beschreibung	Steuerung
%SW111	Status dezentrale Verbindung	Hinweis: Bit 0 entspricht der dezentralen SPS 1, Bit 1 der dezentralen SPS 2 etc. Bits [0] bis [6]: ● im Status 0 = dezentrale Steuerung 1-7 nicht vorhanden ● im Status 1 = dezentrale Steuerung 1-7 vorhanden Bit [8] bis Bit [14]: ● im Status 0 = dezentrale E/A an dezentraler SPS 1-7 erkannt ● im Status 1 = SPS-Erweiterung in dezentraler Steuerung 1-7 erkannt	S
%SW112	Fehlercode (Konfiguration /Betrieb) der dezentralen Verbindung	00: Operationen erfolgreich 01: Timeout erkannt (Slave) 02: Prüfsummenfehler erkannt (Slave) 03: Nichtübereinstimmung der Konfiguration (Slave) Wird vom System auf 1 gesetzt und muss vom Anwender zurückgesetzt werden.	S
%SW113	Konfiguration dezentrale Verbindung	Hinweis: Bit 0 entspricht der dezentralen SPS 1, Bit 1 der dezentralen SPS 2 etc. Bits [0] bis [6]: ● im Status 0 = dezentrale Steuerung 1-7 nicht konfiguriert ● im Status 1 = dezentrale Steuerung 1-7 konfiguriert Bit [8] bis Bit [14]: ● im Status 0 = dezentrale E/A konfiguriert als dezentrale Steuerung 1-7 ● im Status 1 = Partnersteuerung konfiguriert als dezentrale Steuerung 1-7	S
%SW114	Abarbeitungsblöcke freigeben	Aktiviert oder deaktiviert den Betrieb von Abarbeitungsblöcken durch das Anwenderprogramm oder das Bedienterminal. Bit 0: 1 = Abarbeitungsblock #0 wird aktiviert ... Bit 15: 1 = Abarbeitungsblock #15 wird aktiviert Am Anfang sind alle Abarbeitungsblöcke aktiviert. Sind Abarbeitungsblöcke konfiguriert, ist der voreingestellte Wert FFFF. Sind keine Abarbeitungsblöcke konfiguriert, ist der voreingestellte Wert 0.	S und U
%SW118	Statuswort Basissteuerung	Zeigt in der Master-Steuerung erkannte Fehler. Bit 9: 0= externer Fehler oder Komm.- fehler Bit 12: 0 = Echtzeituhr nicht installiert Bit 13: 0 = Konfigurationsfehler (E/A-Erweiterung ist zwar konfiguriert, sie fehlt jedoch oder ist fehlerhaft). Alle anderen Bits dieses Worts sind auf 1 gesetzt und reserviert. Bei einer fehlerfreien Steuerung ist der Wert dieses Worts FFFFh.	S

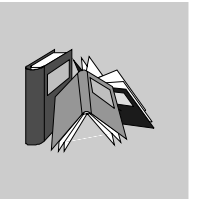
Systemwörter	Funktion	Beschreibung	Steuerung
%SW120	Zustand Erweiterungs-E/A-Modul	Ein Bit pro Modul. Adresse 0 = Bit 0 1 = Nicht gut 0 = OK	S

Beschreibung der in der Tabelle verwendeten Abkürzungen

Tabelle der Abkürzungen

Abkürzung	Beschreibung
S	Vom System gesteuert
U	Vom Benutzer gesteuert

Glossar



!

% Präfix, das interne Speicheradressen in der Steuerung kennzeichnet, in denen Werte von Programmvariablen, Konstanten, E/A usw. gespeichert werden.

A

Abarbeitungsblöcke Ein Funktionsblock für die Programmierung von Datums- und Uhrzeitfunktionen zur Steuerung von Ereignissen. Optionale Echtzeituhr erforderlich.

Abtastmodus Gibt an, wie die Steuerung ein Programm abtastet. Es gibt zwei Arten von Abtastmodi: Normal (zyklisch), die Steuerung tastet zyklisch ab. Periodisch, die Steuerung tastet für eine bestimmte Dauer (Bereich von 2 - 150 ms) ab, bevor der nächste Zyklus gestartet wird.

Adressen Interne Register in der Steuerung, in denen Werte für Programmvariablen, Konstanten, E/A usw. gespeichert werden. Adressen werden durch ein Prozentzeichen (%) als Präfix gekennzeichnet. %I0.1 z.B. gibt eine Adresse im RAM-Speicher der Steuerung an, die den Wert von Eingangskanal 1 enthält.

Analoges Potentiometer Eine anliegende Spannung, die eingestellt werden kann und in einen Digitalwert für den Einsatz in einer Applikation umgewandelt werden kann.

Animationstabelle	Tabelle, die in einem Spracheditor oder in einem Bedienbild erstellt wurde. Wenn ein PC an die Steuerung angeschlossen ist, werden die Variablen der Steuerung angezeigt und es können Werte zur Fehlerbehebung forciert werden. Kann als eigene Datei mit der Erweiterung *.tat gespeichert werden.
Animationstabelleneditor	Ein spezielles Fenster in der TwidoSoft Applikation zum Anzeigen und Erstellen von Animationstabellen.
Anweisungsliste (AWL)	Ein in der Anweisungsliste (AWL) geschriebenes Programm besteht aus einer Folge von Anweisungen, die von der Steuerung der Reihe nach ausgeführt werden. Jede Anweisung besteht aus einer Zeilennummer, einem Anweisungscode und einem Operanden.
Applikationsdatei	Twido Applikationen werden in Dateien vom Typ *.twd gespeichert.
Applikationsnavigator	Ein spezielles Fenster in TwidoSoft, das eine grafische, baumartige Darstellung einer Applikation anzeigt. Dient der komfortablen Konfiguration und Darstellung einer Applikation.
ASCII	(American Standard Code for Information Interchange) Ein Kommunikationsprotokoll, das alphanumerische Zeichen wie Buchstaben, Zahlen und einige grafische Zeichen sowie Steuerzeichen darstellt.
ausführen	Eine TwidoSoft Applikation besteht aus einem Programm, Konfigurationsdaten, Symbolen und Dokumentation.
Ausgangszustand	Der Modus von TwidoSoft, der in der Statuszeile angezeigt wird, wenn TwidoSoft gestartet wird oder wenn keine Applikation geöffnet ist.
Automatische Zeilenbestätigung	Wenn Sie AWL-Anweisungen einfügen oder ändern, ermöglicht diese optionale Einstellung bei der Eingabe von Programmzeilen eine Prüfung auf Fehler und nicht aufgelöste Symbole. Alle Elemente müssen korrekt eingegeben sein, damit Sie die Zeile verlassen können. Wird im Dialogfeld "Vorzüge" eingestellt.
Automatisches Laden	Eine Funktion, die immer aktiviert ist und eine Applikation automatisch von einer Sicherungs-Cartridge in den RAM der Steuerung überträgt, wenn eine Applikation verloren geht oder beschädigt wird. Beim Anlauf vergleicht die Steuerung die Applikation, die sich derzeit im RAM der Steuerung befindet, mit der Applikation im optionalen Speichermodul (sofern installiert). Gibt es Unterschiede, wird die Kopie aus der Sicherungs-Cartridge in die Steuerung und in den internen EEPROM kopiert. Ist die Sicherungs-Cartridge nicht installiert, wird die Applikation im internen EEPROM in die Steuerung kopiert.
AWL-Editor	Einfacher Programm-Editor für die Erstellung und Bearbeitung von AWL-Programmen.

B

Backup Ein Befehl, der die Applikation aus dem RAM der Steuerung in den internen EEPROM der Steuerung und in das optionale Speichermodul (sofern installiert) kopiert.

Bemerkungen Kommentare sind Texte, die Sie zur Dokumentation eines Programms eingeben. Bei KOP-Programmen können Sie maximal drei Zeilen Text in die Programmblock-Kopfzeile eingeben, um den Zweck des Programmblocks zu beschreiben. Jede Zeile darf maximal 64 Zeichen umfassen. Bei AWL-Programmen können Sie Text in nicht nummerierten Programmzeilen eingeben. Kommentare müssen zwischen Klammern und Sternchen stehen: (*HIER STEHEN DIE KOMMENTARE*).

Beobachtungs-zustand Der Modus von TwidoSoft, der in der Statuszeile angezeigt wird, wenn an eine Steuerung im schreibgeschützten Modus ein PC angeschlossen ist.

Betriebszu-stände Gibt den TwidoSoft-Status an. Wird in der Statusleiste angezeigt. Es gibt vier Betriebszustände: Ausgangszustand, Offline, Online und Beobachten.

BootP Ein UDP/IP-basiertes Protokoll (Bootstrap-Protokoll), das es einem hochfahrenden Host ermöglicht, sich selbst dynamisch und ohne Überwachung zu konfigurieren. BootP bietet eine Möglichkeit, einen Host über seine zugewiesenen IP-Adresse zu informieren.

C

CAN **Controller Area Network**: Ursprünglich für Anwendungen im Automobilbereich entwickelter Feldbus, der jetzt in vielen Sektoren angefangen von der Industrie bis hin zum Dienstleistungssektor verwendet wird.

CiA **CAN in Automation**: Internationale Organisation der Anwender und Hersteller von CAN-Produkten.

Client Ein Computerprozess, der einen Dienst von anderen Computerprozessen fordert.

COB — **Communication Object (dt.: Kommunikationsobjekt)**: Übertragungseinheit auf dem CAN-Bus. Ein COB ist durch eine eindeutige Kennung identifiziert, die auf 11 Bits codiert ist [0, 2047]. Ein COB enthält maximal 8 Datenbytes. Die Priorität einer COB-Übertragung wird durch seine Kennung angegeben - je geringer die Kennung, je höher die Priorität des zugehörigen COB.

D

Datenvariable — Siehe Variable.

Datum-/ Uhrfunktionen — Ermöglicht die Steuerung von Ereignissen nach Monat, Tag und Uhrzeit. Siehe Abarbeitungsblöcke.

Dezentrale Steuerung — Eine Twido Steuerung, die für die Kommunikation als Master-Steuerung in einem Netzwerk über die dezentrale Verbindung konfiguriert ist.

Dezentrale Verbindung — Hochgeschwindigkeits-Master/Slave-Bus, der für die Kommunikation geringer Datenmengen zwischen Master-Steuerung und maximal sieben dezentralen (Slave-)Steuerungen ausgelegt ist. Es gibt zwei Arten von dezentralen Steuerungen, die für die Übertragung von Daten an eine Master-Steuerung konfiguriert werden können: Partnersteuerungen übertragen Applikationsdaten und dezentrale E/A-Steuerungen übertragen E/A-Daten. Ein Netzwerk über die dezentrale Verbindung besteht aus einer Mischung aus beiden Arten von dezentralen Steuerungen.

E

E/A-Erweiterungsmodule — Optionale Erweiterungs-E/A-Module sind verfügbar, um zusätzliche E/A-Punkte für eine Twido Steuerung aufzunehmen. (Nicht alle Steuerungen können erweitert werden.)

Echtzeituhr — Eine Option, die die Uhrzeit liefert, auch wenn die Steuerung für eine begrenzte Zeit nicht mit Spannung versorgt wird.

EDS — **Electronic Data Sheet (dt.: elektronisches Datenblatt)**: Beschreibende Datei für jedes CAN-Gerät (wird vom Hersteller geliefert).

EEPROM — Electrically Erasable Programmable Read-Only Memory. Twido verfügt über einen internen EEPROM und ein optionales externes EEPROM-Speichermodul.

Erweiterungsbus Erweiterungs-E/A-Module werden über diesen Bus an die Basissteuerung angeschlossen.

Executive Loader Eine 32-Bit-Windows-Applikation zum Laden eines neuen Firmware Executive-Programms in eine Twido Steuerung.

F

FIFO First In, First Out. Ein Funktionsblock für den Betrieb von Warteschlangen.

Firmware Executive Firmware Executive ist das Betriebssystem, das Ihre Applikation ausführt und den Betrieb der Steuerung verwaltet.

Forcieren Absichtliches Einstellen von Eingängen und Ausgängen der Steuerung auf 0 oder 1, auch wenn die tatsächlichen Werte unterschiedlich sind. Dient zur Fehlerbehebung während der Animation eines Programms.

Frame Eine Gruppe von Bits, die einen diskreten Informationsblock bilden. Frames enthalten Netzwerk-Steuerungsinformationen oder Daten. Die Größe und Zusammensetzung eines Frames wird durch die verwendete Netzwerktechnologie bestimmt.

Funktionsblock Eine Programmeinheit mit Eingängen und Variablen, die auf der Basis einer definierten Funktion, z.B. Timer oder Zähler, Werte für Ausgänge berechnet.

G

Gateway Ein Gerät, das Netzwerke mit nicht gleichartigen Netzwerkarchitekturen verbindet und das auf der Applikationsebene betrieben wird. Dieser Begriff kann auch einen Router bezeichnen.

Grafcet Grafcet ermöglicht die graphische und strukturierte Darstellung der Funktionsweise einer sequentiellen Steuerung.
Hierbei handelt es sich um ein analytisches Verfahren, bei dem Ablaufsteuerungssysteme in eine Reihe von Schritten unterteilt werden, denen Aktionen, Transitionen und Bedingungen zugewiesen sind.

Grenzwertausgänge Spulen, die direkt von sehr schnellen Zählern (%VFC) entsprechend den während der Konfiguration vorgenommenen Einstellungen gesteuert werden.

H

Host Ein Knoten in einem Netzwerk.

Hub Ein Gerät, das eine Reihe von flexiblen und zentralisierten Modulen verbindet, um ein Netzwerk zu errichten.

I

Initialisieren Ein Befehl, der alle Datenwerte auf Ausgangszustände setzt. Die Steuerung muss sich im Stopp- oder Fehlermodus befinden.

Instanz Ein eindeutiges Objekt in einem Programm, das zu einer bestimmten Art von Funktionsblock gehört. Im Timer-Format %TMi ist i z.B. eine Zahl, die eine Instanz darstellt.

Internet Die globale Vernetzung von TCP/IP-basierenden Computer-Kommunikationsnetzwerken.

IP Internetprotokoll. Ein gängiges Schicht-3-Protokoll. IP wird meistens mit TCP verwendet.

IP-Adresse Internetprotokolladresse. Eine 32-Bit-Adresse, die Hosts zugewiesen wird, die TCP/IP verwenden.

K

Kaltstart oder Neustart Ein Anlauf der Steuerung, bei dem alle Daten mit Standardwerten initialisiert werden. Das Programm beginnt am Anfang und alle Variablen sind gelöscht. Alle Software- und Hardware-Einstellungen werden initialisiert. Ein Kaltstart kann dadurch ausgelöst werden, dass eine neue Applikation in den RAM der Steuerung geladen wird. Alle Steuerungen ohne Batterie-Backup laufen immer im Kaltstart hoch.

Kippeingang Eingehende Impulse werden erfasst und für die spätere Untersuchung durch die Applikation aufgezeichnet.

Knoten Ein adressierbares Gerät in einem Kommunikationsnetzwerk.

Kommentarzeilen In AWL-Programmen können Kommentare in eigenen Zeilen und nicht in den Anweisungszeilen eingegeben werden. Kommentarzeilen haben keine Zeilennummern und müssen in Klammern und Sternchen stehen: (*HIER STEHEN DIE KOMMENTARE*).

Kompakte Steuerung Art von Twido Steuerung, die über eine einfache, kompakte Konfiguration mit begrenzter Erweiterung verfügt. Die andere Art von Twido Steuerung ist die modulare Steuerung.

Konfigurations-editor Spezielles TwidoSoft Fenster, in dem die Hardware- und Software-Konfiguration verwaltet wird.

Konstanten Ein konfigurierter Wert, der vom ausgeführten Programm nicht geändert werden kann.

Kontakt Ein KOP-Element, das einen Eingang der Steuerung darstellt.

Kontaktplan (KOP) Ein im Kontaktplan geschriebenes Programm besteht aus der grafischen Darstellung von Anweisungen eines Steuerungsprogramms, wobei Kontakte, Spulen und Blöcke als Symbole in einer Folge von Programmblöcken dargestellt werden, die von der Steuerung der Reihe nach ausgeführt werden.

KOP-Editor Spezielles TwidoSoft Fenster für die Bearbeitung eines KOP-Programms.

KOP-Programmblock Zeigt Teile eines AWL-Programms an, die sich nicht in KOP umwandeln lassen.

L

LIFO Last In, First Out. Ein Funktionsblock für den Betrieb von Stapeln.

Löschen Dieser Befehl ermöglicht das Löschen der Applikation in der Steuerung. Er umfasst zwei Optionen:

- Zum Löschen des Inhalts des Steuerungs-RAM, des internen Steuerungs-EEPROM und der externen Sicherungs-Cartridge (falls vorhanden).
- Nur zum Löschen des Inhalts der optionalen Sicherungskassette.

M

MAC-Adresse Media Access Control-Adresse. Die Hardwareadresse eines Geräts. Eine MAC-Adresse wird einem Ethernet-TCP/IP-Modul werkseitig zugewiesen.

Master-Steuerung Eine Twido Steuerung, die als Master in einem Netzwerk über die dezentrale Verbindung konfiguriert ist.

MBAP Modbus-Applikationsprotokoll

Modbus Ein Master/Slave-Kommunikationsprotokoll, bei dem ein einziger Master Reaktionen von Slaves anfordern kann.

Modulare Steuerung Art von Twido Steuerung, die eine flexible Konfiguration mit Erweiterungsmöglichkeiten bietet. Die andere Art von Twido Steuerung ist die kompakte Steuerung.

N

Netzwerk Vernetzte Geräte, die gemeinsam einen Datenpfad und ein Protokoll für die Kommunikation nutzen.

Nicht aufgelöstes Symbol Ein Symbol ohne Variablenadresse.

O

Offline-Betrieb Ein Modus von TwidoSoft, wenn an die Steuerung kein PC angeschlossen ist und die Applikation im PC-Speicher eine andere ist als die Applikation im Speicher der Steuerung. Sie können im Offline-Betrieb eine Applikation anlegen und entwickeln.

Offline-Zustand Der Modus von TwidoSoft, der in der Statuszeile angezeigt wird, wenn an eine Steuerung kein PC angeschlossen ist.

Online-Betrieb Ein Modus von TwidoSoft, wenn an die Steuerung ein PC angeschlossen ist und die Applikation im PC-Speicher die gleiche ist wie die Applikation im Speicher der Steuerung. Im Online-Modus debuggen Sie eine Applikation.

Online-Zustand Der Modus von TwidoSoft, der in der Statuszeile angezeigt wird, wenn an die Steuerung ein PC angeschlossen ist.

Operand Nummer, Adresse oder Symbol zur Darstellung eines Werts, den ein Programm in einer Anweisung ändern kann.

Operator Ein Symbol oder Code zur Angabe der Funktionsweise, die von einer Anweisung ausgeführt werden soll.

P

Paket Die über ein Netzwerk gesendete Dateneinheit.

PC Personal Computer.

PLS Impulserzeugung. Ein Funktionsblock, der ein Rechtecksignal mit einem Betriebsspiel von 50% ein und 50 % aus erzeugt.

Präferenzen Ein Dialogfeld mit wählbaren Optionen zum Einrichten der KOP- und AWL-Programmeditoren.

Programm analysieren Ein Befehl, der ein Programm kompiliert und auf Programmfehler prüft: Syntax- und Strukturfehler, Symbole ohne entsprechende Adressen, vom Programm verwendete Ressourcen, die nicht verfügbar sind, und ob das Programm in den verfügbaren Steuerungsspeicher passt. Fehler werden in der Programmfehleranzeige angezeigt.

Programmblock Ein Programmblock wird in einem Gitter zwischen zwei Potentialschienen eingegeben und besteht aus einer Gruppe graphischer Elemente, die durch horizontale oder vertikale Verbindungen miteinander verbunden sind. Die maximalen Ausmaße eines Programmblocks sind sieben Reihen und elf Spalten

Programmblock-Kopfzeile Ein Feld, das sich direkt oberhalb eines KOP-Programmblocks befindet und für die Dokumentation des Programmblocks verwendet werden kann.

Programmfehleranzeige Spezielles TwidoSoft Fenster zum Anzeigen von Programmfehlern und Warnungen.

Programmierbare Steuerung Eine Twido Steuerung. Es gibt zwei Arten von Steuerungen: Kompaktsteuerungen und modulare Steuerungen.

Protokoll Beschreibt Meldungsformate und eine Reihe von Regeln, die von zwei oder mehr Geräten verwendet werden, um mittels dieser Formate zu kommunizieren.

PWM Impulsbreitenmodulation. Ein Funktionsblock, der ein Rechtecksignal mit variablem Betriebsspiel erzeugt, das vom Programm eingestellt werden kann.

Q

Querverweis-anzeige Ein spezielles Fenster in der TwidoSoft Applikation zum Anzeigen von Querverweisen.

Querverweise Erzeugung einer Liste mit Operanden, Symbolen, Zeilen-/Programmblocknummern und Operatoren, die in einer Applikation verwendet werden. Dient zur Vereinfachung der Erstellung und Verwaltung von Applikationen.

R

Rahmungstypen Zwei gängige Rahmungstypen sind Ethernet II und IEEE 802.3.

RAM Random Access Memory. Twido Applikationen werden in den internen, nicht-flüchtigen RAM zur Ausführung geladen.

Reflexausgang In einer Zählart wird der aktuelle Wert des sehr schnellen Zählers (%VFC.V) gegen die konfigurierten Grenzwerte gemessen, um den Zustand dieser zweckbestimmten Ausgänge zu ermitteln.

Register Besondere Register in der Steuerung, die für die LIFO/FIFO-Funktionsblöcke gedacht sind.

Ressourcen-Manager Eine Komponente von TwidoSoft, die den Speicherbedarf einer Applikation während der Programmierung und Konfiguration beobachtet, indem Referenzen auf Software-Objekte, die von einer Applikation gemacht werden, verfolgt werden. Ein Objekt gilt als von der Applikation referenziert, wenn es als Operand in einer Anweisungsliste oder in einem KOP-Programmblock verwendet wird. Zeigt Statusinformationen zum Prozentsatz des genutzten Speichers an und gibt eine Warnung aus, wenn wenig Speicher zur Verfügung steht. Siehe Speicherauslastungsanzeige.

Reversible Anweisungen Eine Programmiertechnik, bei der die Anweisungen sowohl in AWL als auch in KOP angezeigt werden können.

Router Ein Gerät, das zwei oder mehr Sections eines Netzwerks miteinander verbindet und den Informationsfluss zwischen diesen Sections ermöglicht. Ein Router überprüft jedes von ihm empfangene Paket und entscheidet, ob das Paket für den Rest des Netzwerks blockiert oder an den Rest des Netzwerks übertragen werden soll. Der Router wird versuchen, das Paket auf dem effektivsten Pfad über das Netzwerk zu senden.

RTC Siehe Echtzeituhr.

RTU Remote Terminal Unit. Ein Protokoll mit acht Bits für die Kommunikation zwischen einer Steuerung und einem PC.

Run Ein Befehl, der bewirkt, dass die Steuerung ein Anwendungsprogramm ausführt.

S

Schnelle Zähler Funktionsblöcke, die schneller aufwärts/abwärts zählen als die verfügbaren Zähler-Funktionsblöcke. Ein schneller Zähler kann mit einer Geschwindigkeit bis zu 5 kHz zählen.

Schritt Ein Grafcet-Schritt gibt einen Zustand in der Ablaufsteuerung an.

Schutz Bezieht sich auf zwei unterschiedliche Arten von Applikationsschutz: Passwortschutz für die Zugriffssteuerung und Applikationsschutz, der verhindert, dass das Anwendungsprogramm gelesen und geschrieben werden kann.

Sehr schnelle Zähler Ein Funktionsblock, der schneller zählen kann als Zähler und schnelle Zähler. Ein sehr schneller Zähler kann mit einer Geschwindigkeit bis zu 20 kHz zählen.

Server Ein Computerprozess, der Dienste für Clients bietet. Dieser Begriff kann auch den Computerprozess bezeichnen, auf dem der Dienst beruht.

Speicherauslastungsanzeige Ein Teil der Statuszeile im TwidoSoft Hauptfenster, das den Prozentsatz des Gesamtspeichers der Steuerung anzeigt, der von der Applikation verwendet wird. Zeigt eine Warnung an, wenn wenig Speicher zur Verfügung steht.

Speichermodul Optionales Speichermodul, das zum Sichern und Wiederherstellen einer Applikation dient (Programm- und Konfigurationsdaten). Zwei Größen sind erhältlich: 32 und 64 KB.

SPS-Erweiterung Eine Twido Steuerung, die als Slave in einem Netzwerk über die dezentrale Verbindung konfiguriert ist. Im Speicher der Partnersteuerung kann eine Applikation ausgeführt werden und das Programm kann auf lokale Daten und auf Erweiterungs-E/A-Daten zugreifen, doch die E/A-Daten können nicht an die Master-Steuerung weitergegeben werden. Das Programm, das in der Partnersteuerung läuft, gibt die Informationen über Netzwerkworte (%INW und %QNW) an die Master-Steuerung weiter.

Spule Ein KOP-Element, das einen Ausgang der Steuerung darstellt.

Standard-Gateway Die IP-Adresse des Netzwerks oder Host, an den alle an ein unbekanntes Netzwerk oder an einen unbekannten Host adressierten Pakete gesendet werden. Das Standard-Gateway ist normalerweise ein Router oder ein anderes Gerät.

Steuerung Speicherprogrammierbare Twido Steuerung. Es gibt zwei Arten von Steuerungen: Kompaktsteuerungen und modulare Steuerungen.

Stopp Ein Befehl, der bewirkt, dass die Steuerung die Ausführung eines Anwendungsprogramms stoppt.

Subnetz Ein physikalisches oder logisches Netzwerk innerhalb eines IP-Netzwerks, das eine Netzwerkadresse gemeinsam mit anderen Abschnitten des Netzwerks nutzt.

Subnetzmaske Eine Bitmaske, die verwendet wird, um zu ermitteln oder festzulegen, welche Bits in einer IP-Adresse der Netzwerkadresse entsprechen und welche Bits den Subnetzabschnitten der Adresse entsprechen. Die Subnetzmaske besteht aus der Netzwerkadresse plus den für die Identifizierung der Subnetzwerke reservierten Bits.

Switch Ein Netzwerkgerät, das zwei oder mehr separate Netzwerksegmente miteinander verbindet und den Datenverkehr zwischen diesen Segmenten ermöglicht. Ein Switch legt fest, ob ein Frame geblockt oder abhängig von seiner Zieladresse weitergeleitet werden soll.

Symbol Ein Symbol ist eine Zeichenkette mit maximal 32 alphanumerischen Zeichen, von denen das erste Zeichen ein Buchstabe ist. Mit Symbolen können Sie ein Steuerungsobjekt personalisieren, um die Pflegbarkeit der Applikation zu erhöhen.

Symboltabelle Eine Tabelle mit den Symbolen, die in einer Applikation verwendet werden. Wird im Symbol-Editor angezeigt.

T

TCP Transmission Control Protocol.

TCP/IP Ein Protokollpaket, das aus dem Transmission Control Protocol und dem Internet Protocol besteht; das Kommunikationsprotokollpaket, auf dem das Internet basiert.

Timer Ein Funktionsblock zur Einstellung eines Zeitraums für die Steuerung eines Ereignisses.

Trommelsteuerung Ein Funktionsblock, der ähnlich wie eine elektromechanische Trommelsteuerung funktioniert, die nach externen Ereignissen von Schritt zu Schritt wechselt.

Twido Eine Produktreihe von Steuerungen von Schneider Electric, die aus zwei Arten von Steuerungen (Kompaktsteuerungen und modularen Steuerungen), Erweiterungsmodulen für zusätzliche E/A-Punkte und Optionen wie Echtzeituhr, Kommunikation, Bedienerterminal und Speichermodul besteht.

TwidoSoft Eine graphische 32-Bit-Windows-Entwicklungssoftware für die Konfiguration und Programmierung von Twido Steuerungen.

U

UDP Ein Kommunikationsprotokoll (User Datagram Protocol), das Teil des TCP/IP-Softwarepakets ist, das von Anwendungen zur Übertragung von Datagrammen verwendet wird. UDP ist auf der Teil von TCP/IP, der für die Port-Adressen verantwortlich ist.

V

Variable Speichereinheit, die vom Programm adressiert und geändert werden kann.

W

Warmstart Ein Anlauf von der Steuerung nach Spannungsverlust, wobei die Applikation nicht geändert wird. Die Steuerung kehrt in den Zustand zurück, in dem sie sich vor dem Spannungsverlust befand, und beendet den in Bearbeitung gewesenen Zyklus. Alle Applikationsdaten bleiben erhalten. Diese Funktion ist nur bei modularen Steuerungen verfügbar.

Z

Zähler Ein Funktionsblock, der Ereignisse zählt (Aufwärts- oder Abwärtszähler).

Zyklus Eine Steuerung tastet ein Programm ab und führt im Wesentlichen drei Funktionen aus. Als erstes werden die Eingänge gelesen und deren Werte im Speicher abgelegt. Danach wird das Anwendungsprogramm Anweisung für Anweisung ausgeführt. Die Ergebnisse werden im Speicher abgelegt. Zum Schluss werden die Ausgänge anhand der Ergebnisse aktualisiert.

Index

Symbols

-, 589
%Ci, 414
%DR, 472
%FC, 478
%INW, 43
%MSG, 495
%PLS, 469
%PWM, 466
%QNW, 43
%S, 620
%S0, 621
%S1, 621
%S10, 622
%S100, 630
%S101, 630
%S103, 631
%S104, 631
%S11, 622
%S110, 631
%S111, 631
%S112, 631
%S113, 632
%S118, 632
%S119, 632
%S12, 622
%S13, 623
%S17, 623
%S18, 623
%S19, 624
%S20, 624
%S21, 71, 624
%S22, 71, 624
%S23, 71, 625
%S24, 625
%S25, 626
%S26, 626
%S31, 627
%S38, 627
%S39, 627
%S4, 621
%S5, 621
%S50, 627
%S51, 628
%S52, 628
%S59, 628
%S6, 621
%S66, 629
%S69, 629
%S7, 621
%S75, 629
%S8, 622
%S9, 622
%S95, 629
%S96, 629
%S97, 630
%SBR, 420
%SCi, 422
%SW, 633
%SW0, 633
%SW1, 633
%SW101, 643
%SW102, 643
%SW103, 643

%SW104, 643
%SW105, 643
%SW106, 643
%SW11, 634
%SW111, 644
%SW112, 644
%SW113, 644
%SW114, 644
%SW118, 644
%SW120, 645
%SW14, 634
%SW15, 635
%SW16, 635
%SW17, 635
%SW18, 635
%SW19, 635
%SW20 -%SW27, 635
%SW20 -.%SW27, 281
%SW30, 635
%SW31, 636
%SW32, 636
%SW48, 636
%SW49, 637
%SW50, 637
%SW51, 637
%SW52, 637
%SW53, 637
%SW54, 637
%SW55, 637
%SW56, 637
%SW57, 637
%SW58, 637
%SW59, 638
%SW6, 633
%SW60, 638
%SW63, 638
%SW64, 638
%SW65, 639
%SW67, 639
%SW68, 640
%SW69, 640
%SW7, 634
%SW73, 640
%SW74, 640
%SW76, 640
%SW77, 640
%SW78, 640
%SW79, 640
%SW80, 640
%SW81. - %SW87, 641
%SW81..%SW87, 280
%SW94, 641
%SW96, 642
%SW97, 642
%TM, 411
%VFC, 481
*, 589
+, 589
/, 589

A

ABS, 589
Absolutwert, 435
Abtastung
    Periodisch, 66
    Zyklisch, 64
ACOS, 593
Adaptive Regelung mit Rückführung, 577
Adaptive Regelung ohne Rückführung, 578
Addieren, 435
Adressierung
    Indexiert, 50
Adressierung von analogen E/A-Modulen, 199
Akkumulator, 360
Aktionszone, 336
Allgemeine Beschreibung
    PID, 535
Analogkanal, 196
Analogmodul
    Beispiel, 208
    Funktion, 198
Analogmodule
    Adressierung, 199
    E/A-Konfiguration, 200
Anschluss des Verbindungskabels, 91
Anschlussbelegung
    Buchse des Kommunikationskabels, 93
    Stecker des Kommunikationskabels, 93
Anweisung EXCH, 494
Anweisung NOP, 448

Anweisung NOT, 399
Anweisungen
  Arithmetisch, 435
  END, 446
  JMP, 449
  Konvertierung, 442
  Laden, 389
  Logisch, 439
  NOP, 448
  NOT, 399
  RET, 450
  SR, 450
  UND, 393
  Vergleich, 433
  XOR, 397
Anweisungen zur Konvertierung zwischen einfachen Wörtern und Doppelwörtern, 444
Arithmetische Anweisungen, 435
ASCII
  Hardwarekonfiguration, 122
  Kommunikation, 89, 122
  Konfigurieren der Schnittstelle, 125
  Softwarekonfiguration, 124
ASCII-Verbindung
  Beispiel, 131
ASIN, 593
AS-Interface V2-Bus
  Adressierung der E/A, 237
  Allgemeine Funktionsbeschreibung, 213
  Ändern der Adresse eines Slaves, 227
  Automatische Adressierung eines Slaves, 234
  Berücksichtigung der neuen Konfiguration, 232
  Beschreibung, 212
  Betriebsart, 243
  Debug-Fenster, 224
  Debugging des Busses, 229
  Diagnose der Slaves, 226
  Einfügen eines Slaves, 235
  Expliziter Austausch, 238
  Fehlerhafte Slaves, 236
  impliziter Austausch, 237
  Prinzip der softwaretechnischen Inbetriebnahme, 215
  Programmierung und Diagnose des AS-Interface V2-Busses, 238
  Übertragung der Abbildung eines Slave, 230
AS-Interface-Bus V2
  Konfigurationsfenster, 216
  Softwarekonfiguration, 218
ATAN, 593
Austausch-Funktionsblock, 495
AWL-Anweisungen, 361
AWL-Sprache
  Überblick, 358
AWL-Zeilenkommentare, 354

B

Backup und Wiederherstellung
  32KB-Sicherungs-Cartridge, 59
  Ohne Sicherungs-Cartridge, 57
Basis-Funktionsblöcke, 402
Bedieneranzeige
  Korrektur der Echtzeituhr, 329
Bedienterminal
  Echtzeituhr, 328
  Einstellungen des seriellen Ports, 327
  Steuerungskennung und Steuerungszustände, 319
  Systemobjekte und Systemvariablen, 321
  Übersicht, 316
Beispiel
  Aufwärtszähler/Abwärtszähler, 419
Berechnung, 435
Betriebszustände, 70
Bitobjekte, 456
  Adressierung, 37
  Übersicht, 27
Bitzeichenketten, 46
BLK, 352
Blöcke
  In Kontaktplänen, 338
Boolesche Anweisungen, 384
  In diesem Handbuch verwendetes Format, 387
  ODER, 395
  Zuweisung, 391
Boolescher Akkumulator, 360

BootP, 170
Boot-up, 252

C

CAN_CMD, 283
CAN-Busleitung, 249
CAN-high, 249
CAN-low, 249
CANopen
  Beschreibung, 249
  Das Protokoll, 249
CANopen-Bus
  Konfigurationsverfahren, 264
CANopen-Feldbus
  Expliziter Austausch, 280
  Impliziter Austausch, 279
  Programmierung und Diagnose für den CANopen-Feldbus, 280
CANopen-Master
  PDO-Adressierung, 279
COS, 593

D

Datensicherung und Wiederherstellung
  Speicherstruktur, 54
Datensicherung und -wiederherstellung
  64 KB-Erweiterungs-Sicherungskassette, 61
Debug
  PID, 559
DEG_TO_RAD, 595
Dekrementieren, 435
Dezentrale Verbindung
  Beispiel, 118
  Hardwarekonfiguration, 110
  Kommunikation, 89, 109
  Konfiguration der dezentralen Steuerung, 112
  Konfiguration der Master-Steuerung, 112
  Softwarekonfiguration, 111
  Zugriff auf Daten der dezentralen E/A, 114
  Zyklussynchronisierung der dezentralen Steuerung, 113
DINT_TO_REAL, 597
Direkte Adressierung, 50
Dividieren, 435
Divisionsrest, 435
Dokumentieren Ihres Programms, 354
Doppelwortobjekte, 45
  Adressierung, 40
  Überblick, 33
D-Verhalten, 582

E

E/A
  Kennzeichnung, 41
Empfangen von Meldungen, 494
END_BLK, 352
END-Anweisungen, 446
EQUAL_ARR, 602
Ereignistasks
  Die verschieden Ereignisquellen, 81
  Ereignisverwaltung, 82
  Überblick, 80
Erweiterte Funktionsblöcke
  Bit- und Wortobjekte, 456
  Programmierprinzipien, 458
Ethernet
  Konfiguration der Verbindungen, 187
  Netzwerkverbindung, 168
  TCP/IP-Setup, 174
EXCH, 494
EXCH3, 190
  Fehlercode, 192
exklusives OR, Anweisungen, 397
EXP, 589
EXPT, 589

F

Fehler, 437
FIFO
  Einführung, 461
  Funktionsweise, 463
FIND_, 604
Flankenerkennung
  fallend, 385
  Steigend, 384

Funktionsblock %MSG3
Anweisung, 190
Funktionsblöcke
Abarbeitungsblöcke, 501
Grafisches Element, 344
Im Programmiergitter, 339
Programmieren von Standard-Funktionsblöcken, 404
PWM, 466
Register, 461
Schiebebitregister (%SBR), 420
Schrittzähler (%SCi), 422
Timer, 406, 411
Trommelsteuerung, 472, 476
Übersicht über Basis-Funktionsblöcke, 402
Zähler, 414

G

Gateway-Adresse, 169
Geräte-ID, 184
Gleitkommaobjekte
Adressierung, 39
Überblick, 33
Grafcet
Ablaufbearbeitung, 375
Anweisungen, 370
Beispiele, 371
Vorbearbeitung, 374
Zugehörige Aktionen, 377
Grafcet-Methoden, 70
Grafische Elemente
Kontaktpläne, 342

H

Header des Strompfads, 337

I

Impulsbreitenmodulation, 466
Impulserzeugung, 469
Indexüberlauf, 51
Initialisierung der Objekte, 78
Inkrementieren, 435
INT_TO_REAL, 597
Integralaktion, 581
IP-Adresse, 169
BootP, 170
Standard-IP-Adresse, 170

J

JMP, 449

K

Kaltstart, 76
Kennzeichnung der E/A, 41
Klammern
Modifikatoren, 365
Verschachteln, 365
Verwendung in Programmen, 364
Kommunikation
ASCII, 122
Dezentrale Verbindung, 109
Modbus, 134
Kommunikation per Modem, 91
Konfiguration
PID, 542
Sende-/Empfangstabelle für ASCII, 126
Konfiguration der Verbindungen, 187
Konfigurieren
Port für Modbus, 138
Schnittstelle für ASCII, 125
Kontakte, 338
Grafisches Element, 342
Kontaktplan-Diagramme
Einführung, 334
Kontaktpläne
Blöcke, 338
Grafische Elemente, 342
OPEN und SHORT, 345
Programmierprinzipien, 336
Konvertierungsanweisungen, 442
KOP/AWL-Programmblock, 353
KOP-Programm
Umkehren in AWL, 351
KOP-Programmblöcke, 335
Korrektur der Echtzeituhr, 500
Korrekturfaktor der Echtzeituhr, 329

L

LAN ACT, 189
LAN ST, 189
LD, 389
LDF, 385, 389
LDN, 389
LDR, 384, 389
Life guarding, 258
Life time, 258
LIFO
  Einführung, 461
  Funktionsweise, 462
LKUP, 612
LN, 589
LOG, 589
Logische Anweisungen, 439

M

MAC-Adresse, 170
Markiertes IP, 179
MAX_ARR, 606
MEAN, 617
Merkmale der PID-Funktion, 539
MIN_ARR, 606
Modbus
  Hardwarekonfiguration, 134
  Kommunikation, 90, 134
  Konfigurieren des Ports, 138
  Master, 90
  Slave, 90
  Softwarekonfiguration, 138
  Standard-Requests, 153
  TCP Client/Server, 161
  TCP Modbus-Nachrichtenübertragung, 190
Modbus TCP/IP
  Dezentrale Geräte, 183
Modbus-Verbindung
  Beispiel 1, 146
  Beispiel 2, 149
Modus
  Operational, 254
  Pre-operational, 254
MPP, 367
MPS, 367
MRD, 367
Multiplizieren, 435

N

Netzwerk
  Adressierung, 43
Nicht bedingte Programmblöcke, 353
Nicht reversible Programmierung, 458
Node guarding, 258
NOP, 448
Numerische Anweisungen
  Schieben, 440
  Zuweisung, 428
Numerische Verarbeitung
  Übersicht, 427

O

Objekte
  Bitobjekte, 27
  Doppelwort, 33
  Funktionsblöcke, 44
  Gleitkomma, 33
  Strukturiert, 46
  Wörter, 30
Objekttabellen, 46
Objektvalidierung, 26
OCCUR_ARR, 607
ODER-Anweisung, 395
OPEN, 345
Operanden, 360
Operationsbausteine, 341
  Grafisches Element, 344
OUT_BLK, 352

P

Parameter, 407
Physikalische Schicht, 249
  CAN-Busleitung, 249

PID
Allgemeine Beschreibung, 535
Debug, 559
Konfiguration, 542
Registerkarte "Allgemein", 544
Registerkarte "Animation", 561
Registerkarte "Ausgang", 556
Registerkarte "Eingang", 547
Registerkarte "PID", 549
Registerkarte "Trace", 563
Registerkarte AT, 551
Polarisierung (extern, 137
Potentiometer, 194
Programmblöcke
Nicht bedingt, 353
Programmblock-Header
Kommentare, 355
Programmiergitter, 336
Programmierprinzipien, 458
Programmiersprachen
Übersicht, 21
Programmierung
Dokumentieren Ihres Programms, 354
Proportionale Verabeitung, 580
Protokoll
Modbus TCP/IP, 90
Protokolle, 89
Prüfen der Zykluszeit, 69

Q

Quadratwurzel, 435

R

RAD_TO_DEG, 595
REAL_TO_DINT, 597
REAL_TO_INT, 597
Register
FIFO, 463
LIFO, 462
Programmierung und Konfiguration, 464
Registerkarte "Allgemein"
PID, 544
Registerkarte "Animation"
PID, 561
Registerkarte "Ausgang"
PID, 556
Registerkarte "Eingang"
PID, 547
Registerkarte "PID"
PID, 549
Registerkarte "Trace"
PID, 563
Registerkarte AT
PID, 551
RET, 450
Reversible Programmierung, 458
ROL_ARR, 608
ROR_ARR, 608
RS-485 EIA-Leitung), 137
Run/Stop-Bit, 73

S

Schiebeanweisungen, 440
Schiebebitregister, 420
Schneller Zähler-Funktionsblock, 478
Schrittzähler, 422
Sehr schneller Zähler-Funktionsblock (%VFC), 481
Senden von Meldungen, 494
SHORT, 345
SIN, 593
Software-Watchdog, 69
SORT_ARR, 610
Spannungsrückkehr, 72
Spannungsunterbrechung, 72
Speicher
32KB-Sicherungs-Cartridge, 59
64 KB-Sicherungskassette, 61
Ohne Sicherungs-Cartridge, 57
Struktur, 54
Speicherbits, 27
Speicherwörter, 30
Sprunganweisungen, 449
Spulen, 338
Grafische Elemente, 343
SQRT, 589
SR, 450
Stapel, 461
Stapelanweisungen, 367

Steuerparameter
ASCII, 126
Steuertabelle
Modbus, 140
Subnetzmaske, 169
Subtrahieren, 435
SUM_ARR, 601
Symbolisieren, 52
Systembits, 620
Systemwörter, 633

T

TAN, 593
Task-Zyklus, 69
TCP Client/Server, 161
TCP/IP
Protokoll, 90
TCP/IP-Setup, 174
Testzone, 336
Timeout (Ethernet), 181
Timer, 407
Einführung, 406
Programmieren und Konfigurieren, 411
Typ TOF, 408
Typ TON, 409
Typ TP, 410
Zeitbasis von 1 ms, 412
Tipps für die Programmierung, 346
TOF-Timer, 408
TON-Timer, 409
TP-Timer, 410
Trommelsteuerungen
Programmieren und Konfigurieren, 476
Trommelsteuerungs-Funktionsblock, 472
TRUNC, 589
TwidoSoft
Einführung, 20

U

Überlauf, 437
Index, 51
Übersicht über die Kommunikation, 89
Uhrfunktionen
Abarbeitungsblöcke, 501
Einstellen von Datum und Uhrzeit, 506
Übersicht, 500
Uhrzeit-/Datumsstempel, 504
Umkehrbarkeit
Einführung, 351
Empfehlungen, 352
UND-Anweisungen, 393
Unterprogramm-Anweisungen, 450

V

Vergleichsanweisungen, 433
Vergleichsbaustein
Grafisches Element, 344
Vergleichsbausteine, 340
Verknüpfungselemente
Grafische Elemente, 342

W

Warmstart, 74
Warteschlange, 461
Wortobjekte, 456
Adressierung, 38
Überblick, 30

X

XOR, 397

Z

Zähler, 414
Programmierung und Konfiguration, 418
Zugriff auf das Debugging
PID, 559
Zugriff auf die Konfiguration
PID, 542
Zuweisungsanweisungen
Numerisch, 428
Zuweisungsbefehle, 391
Zykluszeit, 69