10/10 MN05003003Z-DE
ersetzt 01/08 AWB2724-1584D

easyControl
Programmierbare Steuerung EC4-200

Powering Business Worldwide

Alle Marken- und Produktnamen sind Warenzeichen oder eingetragene Warenzeichen der jeweiligen Titelhalter.

Störfallservice
Bitte rufen Sie Ihre lokale Vertretung an:
http://www.eaton.com/moeller/aftersales
oder
Hotline After Sales Service:
+49 (0) 180 5 223822 (de, en)
AfterSalesEGBonn@eaton.com

Originalbetriebsanleitung
Die deutsche Ausführung dieses Dokuments ist die Originalbetriebsanleitung.

Übersetzung der Originalbetriebsanleitung
Alle nicht deutschen Sprachausgaben dieses Dokuments sind Übersetzungen der Originalbetriebsanleitung.

1. Auflage 2006, Redaktionsdatum 09/06
2. Auflage 2006, Redaktionsdatum 12/06
3. Auflage 2007, Redaktionsdatum 03/07
4. Auflage 2008, Redaktionsdatum 01/08
5. Auflage 2010, Redaktionsdatum 10/10
siehe Änderungsprotokoll im Kapitel „Zu diesem Handbuch“

© 2006 by Eaton Industries GmbH, 53105 Bonn

Autor: Peter Roersch
Redaktion: Thomas Kracht, Barbara Petrick

Alle Rechte, auch die der Übersetzung, vorbehalten.

Kein Teil dieses Handbuches darf in irgendeiner Form (Druck, Fotokopie, Mikrofilm oder einem anderen Verfahren) ohne schriftliche Zustimmung der Firma Eaton Industries GmbH, Bonn, reproduziert oder unter Verwendung elektronischer Systeme verarbeitet, vervielfältigt oder verbreitet werden.

Änderungen vorbehalten.

Gefahr!
Gefährliche elektrische Spannung!

Vor Beginn der Installationsarbeiten

- Gerät spannungsfrei schalten
- Gegen Wiedereinschalten sichern
- Spannungsfreiheit feststellen
- Erden und kurzschließen
- Benachbarte, unter Spannung stehende Teile abdecken oder abschranken.
- Die für das Gerät angegebenen Montagehinweise (AWA) sind zu beachten.
- Nur entsprechend qualifiziertes Personal gemäß EN 50 110-1/-2 (VDE 0105 Teil 100) darf Eingriffe an diesem Gerät/System vornehmen.
- Achten Sie bei Installationsarbeiten darauf, dass Sie sich statisch entladen, bevor Sie das Gerät berühren.
- Die Funktionserde (FE) muss an die Schutzerde (PE) oder den Potentialausgleich angeschlossen werden. Die Ausführung dieser Verbindung liegt in der Verantwortung des Errichters.
- Anschluss- und Signalleitungen sind so zu installieren, dass induktive und kapazitive Einstreuungen keine Beeinträchtigung der Automatisierungsfunktionen verursachen.
- Einrichtungen der Automatisierungstechnik und deren Bedienelemente sind so einzubauen, dass sie gegen unbeabsichtigte Betätigung geschützt sind.
- Damit ein Leitungs- oder Aderbruch auf der Signalseite nicht zu undefinierten Zuständen in der Automatisierungseinrichtung führen kann, sind bei der E/A-Kopplung hard- und softwareseitig entsprechende Sicherheitsvorkehrungen zu treffen.
- Bei 24-Volt-Versorgung ist auf eine sichere elektrische Trennung der Kleinspannung zu achten. Es dürfen nur Netzgeräte verwendet werden, die die Forderungen der IEC 60 364-4-41 bzw. HD 384.4.41 S2 (VDE 0100 Teil 410) erfüllen.
- Schwankungen bzw. Abweichungen der Netzspannung vom Nennwert dürfen die in den technischen Daten angegebenen Toleranzgrenzen nicht überschreiten, andernfalls sind Funktionsausfälle und Gefahrenzustände nicht auszuschließen.
- NOT-AUS-Einrichtungen nach IEC/EN 60 204-1 müssen in allen Betriebsarten der Automatisierungseinrichtung wirksam bleiben. Entriegeln der NOT-AUS-Einrichtungen darf keinen Wiederanlauf bewirken.
- Einbaugeräte für Gehäuse oder Schränke dürfen nur im eingebauten Zustand, Tischgeräte oder Portables nur bei geschlossenem Gehäuse betrieben und bedient werden.
- Es sind Vorkehrungen zu treffen, dass nach Spannungseinbrüchen und -ausfällen ein unterbrochenes Programm ordnungsgemäß wieder aufgenommen werden kann. Dabei dürfen auch kurzzeitig keine gefährlichen Betriebszustände auftreten. Ggf. ist NOT-AUS zu erzwingen.
- An Orten, an denen in der Automatisierungseinrichtung auftretende Fehler Personen- oder Sachschäden verursachen können, müssen externe Vorkehrungen getroffen werden, die auch im Fehler- oder Störfall einen sicheren Betriebszustand gewährleisten beziehungsweise erzwingen (z. B. durch unabhängige Grenzwertschalter, mechanische Verriegelungen usw.).

Inhalt

Zu diesem Handbuch		7
	Änderungsprotokoll	7
	Weiterführende Dokumentationen	7
	Lesekonventionen	8
1 Einsatz		9
	Typenübersicht EC4-200	9
2 Aufbau		11
	Eingänge	11
	– Funktions- und Cursortasten als Eingänge	12
	– Diagnose-Eingänge	12
	– Eingänge für schnelle Zähler	12
	Ausgänge	13
	Speicherkarte (MMC)	13
	– Daten der Speicherkarte	13
	– Datenzugriff auf die Speicherkarte	13
	LED-Zustandsanzeige RUN/STOP/SF und CAN/NET	13
	Echtzeituhr	14
	Schnittstellen	14
	– Programmierschnittstelle zum Anschluss eines PC	14
	– Multifunktions-Schnittstelle (MFS)	14
	– Kabelverbindungen	15
	CAN/easyNet-Schnittstellen	16
3 Erweiterungsgeräte		17
	Eingänge	17
	– Diagnose-Eingänge	17
	Ausgänge	17
4 Montage		19
	Montage auf Hutschiene	19
	Montage auf Montageplatte	19
5 Installation		21
	Versorgungsspannung anschließen	21
	Digital-Eingänge anschließen	21
	Analog-Eingänge anschließen	21
	– Sollwertgeber anschließen	22
	– Temperatursensor anschließen	22
	– 20-mA-Sensor anschließen	22
	Impuls-/Inkrementalgeber anschließen	23
	– Impulsgeber anschließen	23
	– Inkrementalgeber anschließen	23
	Ausgänge anschließen	24
	– Relais-Ausgänge anschließen	24
	– Transistor-Ausgänge anschließen	25
	– Analog-Ausgang anschließen	26
	Speicherkarte, CAN/easyNet, PC-Verbindung	27
	– Speicherkarte stecken oder ziehen	27
	– CAN/easyNet, PC-Verbindung	27
	Erweiterungen/Netzwerk-Kopplungen anschließen	28
	– Zentrale Erweiterung	28
	– Dezentrale Erweiterung	28

6 Bedienung 29
Tastenfeld 29
Menüführung und Eingabe von Werten 29
Menüpunkte wählen oder umschalten 29
– Cursor-Anzeige 29
– Wert einstellen 29
Haupt- und Sondermenü wählen 30
– Statusanzeige 30
– Statusanzeige mit Uhrzeit 30
Menüstruktur 31
– Hauptmenü ohne Passwortschutz 31
– Hauptmenü mit Passwortschutz 32
– Sondermenü 32
– Sondermenü 33

7 Beschreibung der Einstellungen 35
Passwortschutz 35
– Passwort einrichten 35
– Gültigkeitsbereich des Passwortes wählen 35
– Passwort aktivieren 35
– Zugang bei Passwortschutz 36
– Passwort, Bereich ändern oder löschen 36
Menüsprache ändern 37
Datum und Uhrzeit einstellen 37
Anlaufverhalten 37
– Anlauf- (Start-) verhalten einstellen 37
Kontrast und Hintergrundbeleuchtung LCD einstellen 38

8 Konfiguration der Ein-/Ausgänge (I/O) 39
Darstellung der Ein-/Ausgänge in der Konfiguration 39
Anzeige der lokalen Ein-/Ausgänge 39
Ordnerfunktion ändern 39
Anzeige der Ein-/Ausgänge der Erweiterungsgeräte 40

9 Betrieb 41
Allgemeines 41
– Übersicht über die Speichergrößen 41
– Speicher-Definition 41
Einschaltverhalten 41
– Einschaltverhalten mit Bootprojekt auf der Speicherkarte 41
Startverhalten in der Programmiersoftware einstellen 43
Programm START/STOP 43
– Programmstart (STOP → RUN) 43
– Verhalten nach dem Ausschalten/Unterbrechen der Spannungsversorgung 43
– Programmstopp (RUN → STOP) 43
– Programm über externen Schalter starten/stoppen 44
Programmbearbeitung und Systemzeit 44
Zykluszeit überwachen 44
Reset 44
– Reset warm 44
– Reset kalt 44
– Reset Ursprung 44
– Wiederherstellen der Werkseinstellungen (factoryset) 44
– Verhalten der Variablen nach Reset 45

Test und Inbetriebnahme 45
– Breakpoint/Einzelschritt-Betrieb 45
– Einzelzyklus-Betrieb 45
– Zwangssetzen von Variablen und Ein-/Ausgänge (Forcen) 45
– Zustandsanzeige in der Programmiersoftware 45
Schnelle Zähler (Counter) 45
– Counter-Funktionen (Ein-/Ausgänge) 46
Inkrementalwertzähler (Incremental Input) 47
– Erklärung der Ein-/Ausgangssignale (I/Q) 47
– Signalübersicht der Ein-/Ausgänge (I/Q) 48
– Funktionen der Ein-/Ausgangssignale 48
– Referenzierung 48
Systemereignisse 49
– START, COLDSTART, WARMSTART, STOP 49
– Interrupt-Eingänge I1 bis I4 50
– Counter-Interrupt 50
– Timer-Interrupt 50
Interruptverarbeitung 52
– Arbeitsschritte zur Interruptverarbeitung 52
– Beispiel zur Interruptverarbeitung 52
Direkter I/O-Zugriff 53
– Beschreibung der Funktionen 53
Fehlercode bei „Direkter I/O-Zugriff“ 54
Bootprojekt erzeugen und transferieren 55
– Bootprojekt auf Speicherkarte abspeichern 55
– Bootprojekt und Betriebssystem (BTS) auf Speicherkarte 55
– Bootprojekt löschen 55
Betriebssystem herunterladen/aktualisieren 56
– Betriebssystem vom PC in die Steuerung übertragen 56
– BTS vom PC auf die Speicherkarte übertragen 57
– BTS von der Speicherkarte in die Steuerung übertragen 57

10 Browser-Befehle 59
– Ethernet-Parameter einstellen 59
Beschreibung wichtiger Browser-Befehle 60
– canload 60
– setrtc 60

11 Bibliotheken, Funktionsbausteine und Funktionen 61
Bibliotheken handhaben 61
Weitere Systembibliotheken installieren 61
EC4-200-spezifische Funktionen 62
– Bibliothek EC_Util.lib 62
– Bibliothek EC_Visu.lib/EC_Visu2.lib 62

12 Verbindungsaufbau PC – EC4-200 63
Verbindungsaufbau über die RS232 63
Kommunikationsparameter des PCs festlegen/ändern 63
Kommunikationsparameter (Baudrate) der CPU ändern 64
Verbindungsaufbau über Ethernet 64
Abfragen/Ändern der IP-Adresse 66

13 Systemparameter über STARTUP.INI-Datei vorgeben 67
Übersicht 67
Aufbau der INI-Datei 67
Startup.INI-Datei erstellen 67
Einschalten der Steuerung bei gesteckter Speicherkarte mit Startup.INI-Datei 67
Parameter ändern 68
Startup.INI-Datei löschen 68

14 Programmieren über CAN-Netzwerk (Routing) 69
Voraussetzungen 69
Routingeigenschaften der Steuerung 69
Routing über XC200 69
Hinweise zum Routing 70
Einstellung der Node-ID/Routing-ID 70
Einstellung des Master-Teilnehmers 71
Einstellung des Device-Teilnehmers 71
Steuerungskombinationen zum Routing 72

15 RS232-Schnittstelle im Transparent-Modus 73

16 Interaktives Display 75
Darstellungsform (Anzeige) 75
– Umschaltung zwischen Statusanzeige und Ein-/Ausgabemodus 75
– Funktions-/Funktionsbaustein-Übersicht 76
Beschreibung wichtiger Funktionen/Funktionsbausteine 77
– FUNCTION Disp_EnableDisplay: BOOL (*Umschaltung Statusanzeige <-> Ein-/Ausgabemode*) 77
– Allgemeiner Ablauf zur Programmerstellung 80
– Beispiel zur Ausgabe von Texten und Werten 81
– Beispiel zur Ausgabe einer Seite mit Texten und Eingabemöglichkeiten 83
Multi-Funktions-Display MFD-CP4 an der EC4-200 86
– MFD-Aufbau 86

17 EC4-200-Netzwerk-Kopplungen 87
EASY205-ASI 87
– Zyklischer Datenaustausch 87
– Konfiguration 88
– Teilnehmer-Adresse einstellen 88
EASY221-CO, EASY204-DP, EASY222-DN 88
– Zyklischer Datenaustausch 88
– Konfiguration 89
– Einstellung der Teilnehmer-Adresse 89
– Azyklischer Datenaustausch 89

Anhang 93

Netzwerk CAN/easyNet 93
– Zubehör 93
Beispielprogramm zum START/STOP der Steuerung über externen Schalter 94
Verbindungskabel EASY800-PC-CAB 95
Abmessungen und Gewicht 95
Technische Daten 96
– Transistor-Ausgänge 101
– Analog-Ausgang 103
Zeichensätze 104

Stichwortverzeichnis 107

→ Die bisherigen Kapitel 17: „Das Netzwerk easyNet“ und Kapitel 18: „Programmieren über easyNet (Routing)“ entfallen.
Diese Informationen finden Sie weitaus detaillierter im Handbuch MN05006004Z-DE (frühere Bezeichnung 08/07 AWB2786-1593de), „Datenübertragung zwischen easy- und IEC-Steuerungen (easyNet)“.

Zu diesem Handbuch

Änderungsprotokoll

Gegenüber den früheren Ausgaben hat es folgende wesentliche Änderungen gegeben:

Redaktionsdatum	Seite	Stichwort	neu	Änderung
12/06	87	EC4-200-Netzwerk-Kopplungen		
03/07	9	Einfügen der Typen EC4P-222-…	✓	
	14	Ethernet-Schnittstelle	✓	
	15	Kabelverbindungen	✓	
	59	Browserbefehle für Ethernet	✓	
	67	Startup.INI mit Etherneteinträgen	✓	
	104	Zeichensätze	✓	
01/08	29	Menüpunkte wählen oder umschalten		✓
	33	Sondermenü		✓
	49	START, COLDSTART, WARMSTART, STOP		✓
	50	Interrupt-Eingänge I1 bis I4		✓
	51	Timer-Interrupt		✓
	87	Kapitel 17: „Das Netzwerk easyNet" und Kapitel 18: „Programmieren über easyNet (Routing)" entfallen. Diese Informationen finden Sie weitaus detaillierter im Handbuch MN05006004Z-DE (frühere Bezeichnung 08/07 AWB2786-1593) „Datenübertragung zwischen easy- und IEC-Steuerungen (easyNet)".		✓
10/10	allgemein	Umstellung auf Eaton-Bezeichnungen		✓

Weiterführende Dokumentationen

In diesem Handbuch wird an verschiedenen Stellen auf ergänzende oder vertiefende Beschreibungen in anderen Dokumentationen hingewiesen. Diese Dokumentationen werden als PDF-Datei bei der Installation der Produkt-CD auf Ihrem PC gespeichert.

Für ein schnelleres Auffinden der gesuchten Dokumentation wählen Sie im Windows-Startmenü:

Programme → Moeller Software → easy Soft CoDeSys → Dokumentation…

Darüber hinaus besteht die Möglichkeit, die PDF-Dateien über den FTP-Server herunterzuladen. Hier stehen immer die aktuellen Daten zur Verfügung.

ftp://ftp.moeller.net/DOCUMENTATION/AWB_MANUALS/

Konkrete Informationen über die Kommunikation mit CAN-Teilnehmern und deren Konfiguration finden Sie in den folgend aufgeführten Dokumentationen:

- AN27K19D: Kommunikation zwischen zwei Steuerungen mit Netzwerkvariablen über CAN (AN2700K19D.pdf)
- AN27K20D: Kopplung mehrerer autarker Steuerungen (CAN-Device) über CANopen (AN2700K20D.pdf)
- Projektierung von CAN-Teilnehmern (AN2700K27D.pdf)

(Zu finden im Windows-Startmenü unter Programme → Moeller Software → easy Soft CoDeSys → Anwendungsbeispiele…)

- MN05010001Z-DE (frühere Bezeichnung AWB2786-1554D): Bibliotheksbeschreibung CANUser.lib, CANUser_Master.lib. Mit den Funktionen der Bibliothek CANUser.lib und CANUser_Master.lib können Sie direkt auf die CAN-Objekte zugreifen.

(Zu finden im Windows-Startmenü unter Programme → Moeller Software → easy Soft CoDeSys → Dokumentation…

Lesekonventionen

Wählen Sie ‹Datei → Neu› bedeutet: Aktivieren Sie den Befehl „Neu“ im Menü „Datei“.

→ macht Sie aufmerksam auf interessante Tipps und Zusatzinformationen

Achtung!
warnt vor leichten Sachschäden.

Vorsicht!
warnt vor schweren Sachschäden und leichten Verletzungen.

Warnung!
warnt vor schweren Sachschäden und schweren Verletzungen oder Tod.

Für eine gute Übersichtlichkeit finden Sie auf den linken Seiten im Kopf die Kapitelüberschrift und auf den rechten Seiten den aktuellen Abschnitt, Ausnahmen sind Kapitelanfangsseiten und leere Seiten am Kapitelende.

1 Einsatz

Die Steuerungen aus der EC4-200-Reihe sind programmierbare Schalt- und Steuergeräte. Mit ihnen lösen Sie Aufgaben der Haustechnik und des Maschinen- und Apparatebaus. Eine EC4-200-Steuerung können Sie autark einsetzen und über die CANopen-Schnittstelle mit dezentralen Ein-/Ausgangsgeräten verbinden. Über diese Schnittstelle können Sie auch mit anderen Steuerungen (mit CANopen-Schnittstelle) kommunizieren.

Die Steuerungstypen EC4P-222-… haben zusätzlich eine Ethernetschnittstelle.

Ab der Version 2.0 des Betriebssystems verfügen die Steuerungen über folgende Eigenschaften:

- Anschluss von Erweiterungsgeräten/Steuerungen über easyLink
- Anschluss des Multi-Funktions-Display MFD-CP4 über die Multifunktions-Schnittstelle
- Transparent-Mode über die Multifunktions-Schnittstelle
- Direkter Zugriff auf die lokalen Ein-/Ausgänge und der „Schnellen Zähler"
- Integration in das easyNet-Netzwerk über die easyNet-/CAN-Schnittstelle.

Steuerungen ab der Version 2.10 können Sie über Netzwerk-Kopplungen auch an die Netzwerke ASI, PROFIBUS-DP, CAN oder DeviceNet anschließen.

Programmiert wird die Steuerung mit der Software easySoft-CoDeSys. Sie installieren die Software auf einem handelsüblichen PC mit dem Windows Betriebssystem NT, 2000 oder XP.
Weitere Informationen zur Software finden Sie im Handbuch zur Programmiersoftware (MN05010003Z-DE; frühere Bezeichnung AWB2700-1437D).

Die Software ermöglicht Ihnen einen einfachen Einstieg in die Sprachmittel der IEC wie:

- Anweisungsliste (AWL)
- Funktionsplan (FUP)
- Kontaktplan (KOP)
- Strukturierter Text (ST)
- Ablaufsprache (AS).

Es steht Ihnen eine Anzahl von Operatoren zur Verfügung wie z. B.:

- Logische Operatoren wie AND, OR, …
- Arithmetische Operatoren wie ADD, MUL, …
- Vergleichsoperatoren wie <,=, >

Mit der Programmiersoftware erstellen Sie ein Projekt, testen und dokumentieren es. Funktionen zur Analogwertverarbeitung, Reglererstellung und Funktionsblöcke wie Timer, Zähler erleichtern Ihnen die Programmierung.

Typenübersicht EC4-200

Die EC4-200-Reihe enthält Steuerungen, die sich in Display sowie in Art und Anzahl der Ein-/Ausgänge unterscheiden.

Typ	Eigenschaften				
	Tasten/Display	Transistor-Ausgänge	Relais-Ausgänge	Analog-Ausgang	Ethernet-Anschluss
EC4P-221-MTXD1	×	8	–	–	–
EC4P-221-MTXX1	–	8	–	–	–
EC4P-221-MRXD1	×	–	6	–	–
EC4P-221-MRXX1	–	–	6	–	–
EC4P-221-MTAD1	×	8	–	×	–
EC4P-221-MTAX1	–	8	–	×	–
EC4P-221-MRAD	×	–	6	×	–
EC4P-221-MRAX1	–	–	6	×	–
EC4P-222-MTXD1	×	8	–	–	×
EC4P-222-MTXX1	–	8	–	–	×
EC4P-222-MRXD1	×	–	6	–	×
EC4P-222-MRXX1	–	–	6	–	×
EC4P-222-MTAD1	×	8	–	×	×
EC4P-222-MTAX1	–	8	–	×	×
EC4P-222-MRAD	×	–	6	×	×
EC4P-222-MRAX1	–	–	6	×	×

2 Aufbau

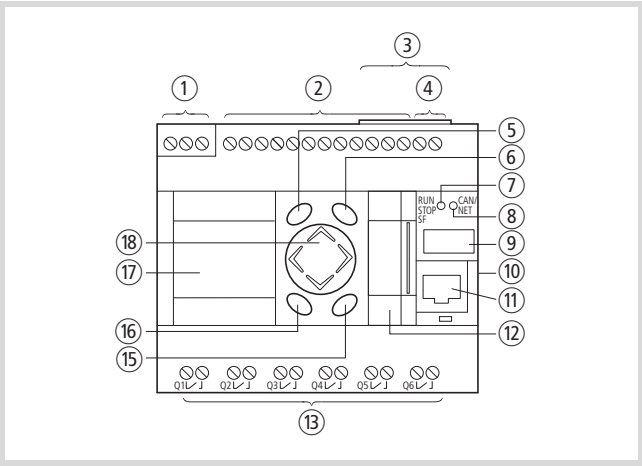


Abbildung 1: Frontdarstellung der EC4P-221-MRAD1, Legende → Abbildung 2

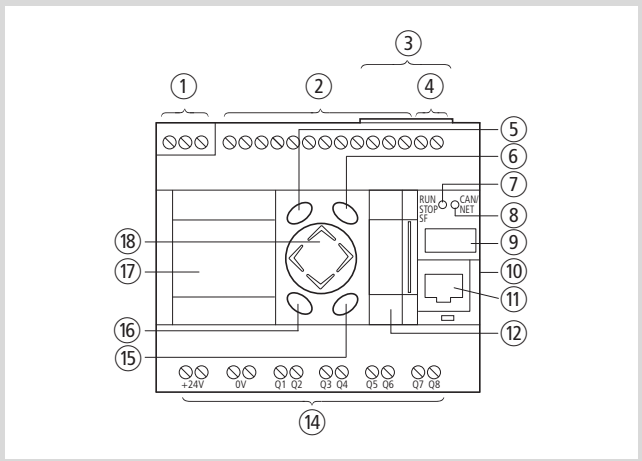


Abbildung 2: Frontdarstellung der EC4P-221-MTAD1

① 24-V-DC-Spannungsversorgung
② Eingänge
③ Schnittstelle zum Anschluss an das CAN-Netzwerk
④ analoger Ausgang, 0…10 V (nicht aktiv)
⑤ DEL-Taste
⑥ ALT-Taste
⑦ LED RUN/STOP/SF
⑧ LED CAN/NET
⑨ Feld für Gerätekennzeichnung
⑩ Schnittstelle easyLink zur Erweiterung
⑪ Programmierschnittstelle zum Anschluss eines PC
⑫ Multifunktions-Schnittstelle
⑬ Relais-Ausgänge
⑭ Transistor-Ausgänge
⑮ OK-Taste
⑯ ESC-Taste
⑰ LCD-Anzeige (EC4P-22x-M…D1)
⑱ Cursortasten P1…P4 (Wippe)

Eingänge

Tabelle 1: Art und Anzahl der Eingänge

digital	12 (I1…I12)	24 V DC
davon analog nutzbar	4 (I7, I8, I11, I12)	24 V DC/0…10 V

Die Eingänge I7, I8, I11, I12 lassen sich auch als Analog-Eingänge verwenden. Die Auswahl treffen Sie im Anwenderprogramm, indem Sie die entsprechende Syntax verwenden, die im Steuerungskonfigurator vorgegeben ist.

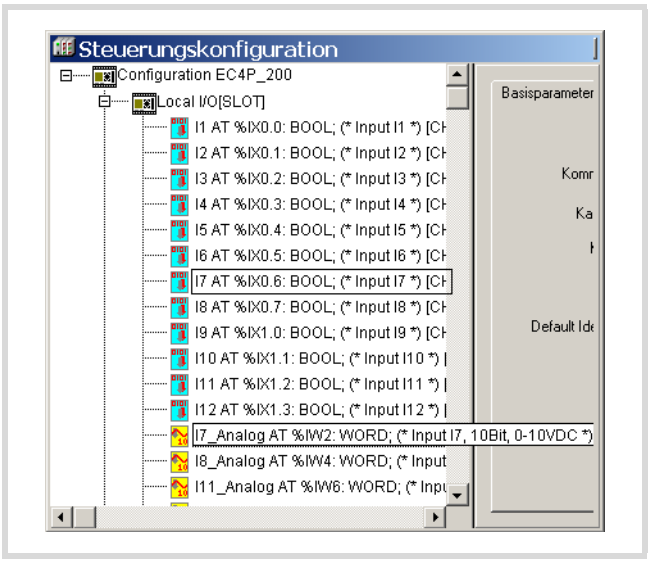


Abbildung 3: Auswahl zwischen Digital- und Analog-Eingang, z. B. I7

Wenn Sie die Eingänge im Anwenderprogramm als Digital-Eingänge programmieren, bildet die Eingangsspannung von 8 V den Grenzwert für die Signale TRUE/FALSE.

Spannung [V]	Zustand
≦ 8	FALSE
> 8	TRUE

Technische Daten: → Seite 96

Die Eingänge I1, I2, I3, I4 können Sie verwenden:

- zur Erzeugung von Interrupts (Eingänge I1, I2, I3, I4)
- zum Ansteuern schneller Zähler wie:
  - 16- oder 32-Bit-Zähler, zum Zählen von Impulsen (I1, I2), Vor-/Rückwärtszähler
  - Inkrementalwertzähler, 32 Bit, zur Verarbeitung von Signalen eines Inkrementalgebers (I1, I2, I3, I4).

Die Auswahl der Funktion erfolgt in der Steuerungskonfiguration. Die Anwendung mehrerer Funktionen gleichzeitig ist jedoch nicht möglich.

Beispiel: Nutzen Sie den Eingang I1 für einen schnellen Zähler (16 Bit), kann der I2 für einen weiteren schnellen Zähler (16 Bit) genutzt werden, nicht aber zur Erzeugung eines Interrupts.
Die Eingänge I3 und I4 können ebenfalls nicht zur Erzeugung eines Interrupts benutzt werden.

Anschlussbeschreibung → Abbildung 22 auf Seite 23.

Funktions- und Cursortasten als Eingänge

Auf der Frontplatte des Gerätes sind die Funktionstasten DEL, ALT, ESC, OK um die runde Wippe angeordnet. Die Wippe ist in 4 Tipp-Bereiche aufgeteilt, die mit den Symbolen P1 bis P4 gekennzeichnet sind und als Cursortasten bezeichnet werden. Die Funktions- und Cursortasten sind in der Steuerungskonfiguration als Eingänge dargestellt. Die Abfrage dieser Eingänge im Programm erfolgt nach den generellen Syntax-Regeln. Es darf immer nur eine Funktions- und Cursortaste betätigt werden, da sonst bei der Abfrage dieser Tasten im Programm undefinierte Zustände auftreten können.

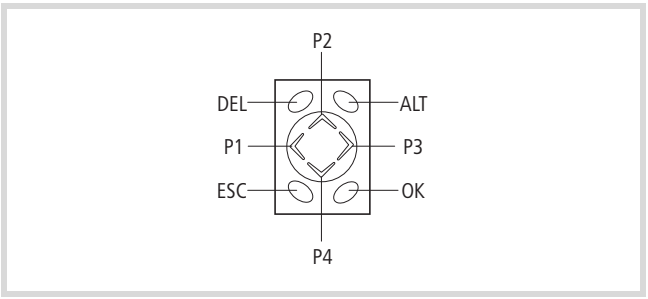


Abbildung 4: Funktionstasten und Wippe mit Cursortasten P1, P2, P3, P4

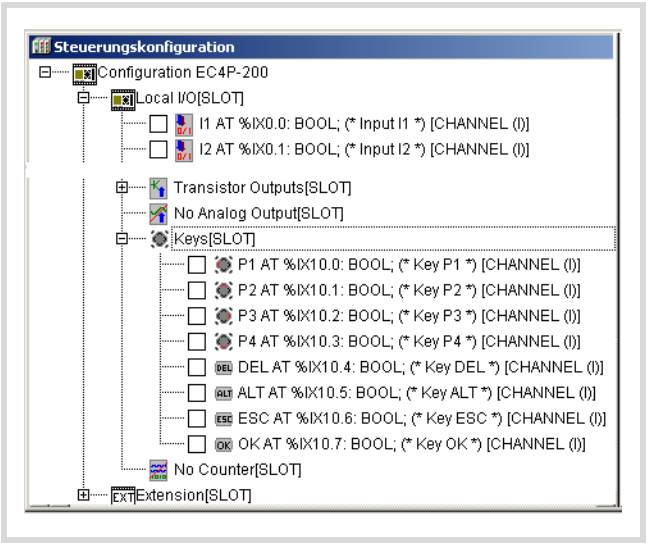


Abbildung 5: Eingänge der Wippen- und Funktionstasten

Mit Hilfe des Funktionsbausteins „Disp_GetDisplayInfo“ aus der Bibliothek „EC_Visu2.lib“ können Sie die Abfrage der Tasten in Abhängigkeit vom Menü der Steuerung beeinflussen,
→ Abschnitt „Bibliothek EC_Visu.lib/EC_Visu2.lib“, Seite 62.

Diagnose-Eingänge

Die Eingänge I13, I14, I15, I16 stellen Ihnen zusätzliche Informationen zur Verfügung:

Eingang	Funktion
I13	keine Bedeutung
I14	Verbindung zum Erweiterungsgerät über easyLink (in der Betriebssystem-Version 1.x noch nicht aktiv): 0: ok, 1: nicht ok
I15	Ausgänge Q1, Q2, Q3, Q4: 0: kein Kurzschluss, 1: Kurzschluss
I16	Ausgänge Q5, Q6, Q7, Q8: 0: kein Kurzschluss, toggelt: Kurzschluss

Die Eingänge können Sie im Programm mit den symbolischen Operanden abfragen.

Eingänge für schnelle Zähler

Sie können zwischen mehreren Funktionen wählen:

- 1 × 32-Bit-Zähler, zum Zählen von Impulsen (vor-/rückwärts)
- 2 × 16-Bit-Zähler, zum Zählen von Impulsen (vor-/rückwärts); die Zählrichtung (vor-/rückwärts) ist über den Operand DIRECTION im Programm einstellbar.
- 1 × Inkrementalwertzähler, 32 Bit, zur Verarbeitung von Signalen eines Inkrementalgebers; die Zählrichtung wird durch die Flankenfolge vom Geber vorgegeben

Den Zähler-Typ wählen Sie in der Steuerungskonfiguration aus.

Die Funktion der „Schnellen Zähler“ erfordert das Setzen der Eingänge und das Abfragen der Ausgänge in einer POU, z. B. PLC_PRG. Diese POU darf nicht vom einem Zähler erzeugten Interrupt aufgerufen werden.

Weitere Informationen stehen im Abschnitt „Schnelle Zähler (Counter)“, Seite 45.

Ausgänge

Tabelle 2: Art und Anzahl der Ausgänge

Transistor-Ausgänge EC4P-221/222-MT…	8 (Q1…Q8)	24 V DC/0,5 A
Relais-Ausgänge EC4P-221/222-MR…	6 (Q1…Q6)	250 V AC/8 A

Die Transistor-Ausgänge verfügen über eine Kurzschlussüberwachung. Tritt an einem der Ausgänge ein Kurzschuss auf, wird dies über die Diagnose-Eingänge I15/I16 gemeldet. I15 wird auf 1-Signal gesetzt, wenn an den Ausgängen Q1 bis Q4 ein Kurzschluss auftritt. Tritt an den Ausgängen Q5 bis Q6 ein Kurzschluss auf, toggelt der Eingang I16.

⚠ Vorsicht!
Fragen Sie I15/I16 im Programm ab. Im Kurzschlussfall setzen Sie die Ausgänge auf 0-Pegel, um eine thermische Überbelastung der Ausgangsbeschaltung zu vermeiden.

Speicherkarte (MMC)

Die Speicherkarte dient als Massenspeicher und unterstützt das FAT16-Dateisystem.

Daten der Speicherkarte

Auf der Speicherkarte können Sie folgende Daten ablegen:

Daten	Übertragungsmöglichkeit
Bootprojekt	Browserbefehl: copyprojtommc
Startup.INI-Datei	Browserbefehl: createstartupini
Betriebssystem (BTS)	Aktualisieren des BTS, → Seite 56
Quellcode des Projekts	Online-Betrieb/Online-Menü: Quellcode laden
Allgemeine Dateien	Online-Betrieb/Online-Menü: Datei in Steuerung schreiben Datei aus Steuerung laden

Eine Kurzbeschreibung der Browser-Befehle finden Sie ab Seite 59.

▽ Achtung!
Um Datenverluste zu vermeiden, stellen Sie sicher, dass Sie alle Dateien des Programms geschlossen haben, bevor Sie die Speicherkarte ziehen bzw. stecken oder die Spannung ausschalten.

Datenzugriff auf die Speicherkarte

Mit Funktionen wie „FileOpen" oder „FileRead" können Sie aus dem Anwenderprogramm heraus auf die Dateien der Speicherkarte zugreifen. Diese Funktionen sind in der Bibliothek „EC_File.lib" enthalten und in dem Handbuch „Funktionsbausteine" (MN05010002Z-DE; frühere Bezeichnung AWB2786-1456D) beschrieben.

LED-Zustandsanzeige RUN/STOP/SF und CAN/NET

Nach dem Einschalten der Versorgungsspannung kann die CPU folgende Zustände annehmen, die an den LEDs angezeigt werden:

Tabelle 3: LED-Zustandsanzeige

LED		Bedeutung/CPU-Zustand
RUN/STP/SF	CAN/NET	
rot	rot[1)]	Systemtest wird durchgeführt (bis 6 Sekunden nach Start; nach 6 Sekunden, wenn kein Anwenderprogramm vorhanden ist) CPU im NOT READY!
orange	orange[1)]	System-Update wird durchgeführt
rot	aus[1)]	Systemtest ohne Fehler abgeschlossen
rot blinkend	rot blinkend[1)]	Systemtest ergab einen Fehler
orange	aus	kein Anwenderprogramm vorhanden CPU im NOT READY
grün blinkend	–	Anwenderprogramm geladen CPU im STOP
grün	–	Anwenderprogramm geladen CPU im RUN
rot	–	Zykluszeit überschritten CPU im STOP
orange blinkend	–	Endlosschleife im Programm erkannt CPU im STOP
rot blinkend	rot blinkend	Fataler Fehler aufgetreten

1) LED nur beim Einschalten/Systemtest relevant

Ist die CPU im Zustand RUN, zeigt die CAN/NET-LED folgende Zustände an:

Tabelle 4: LED-Zustandsanzeige für CAN/easyNet

LED		Bedeutung
RUN/STP/SF	CAN/NET	
grün	aus	Kommunikation nicht aktiviert
grün	rot	Buszustand STOP
grün	orange	Buszustand PREOPERATIONAL Teilnehmer kann initialisiert werden, keine Übertragung von Prozessdaten
grün	grün	Buszustand OPERATIONAL Prozessdaten werden übertragen

Echtzeituhr

Die Steuerung besitzt eine Echtzeituhr, die Sie im Anwenderprogramm über Funktionen aus der Bibliothek „SysLibRTC" ansprechen können. Die Funktionen sind in der PDF-Datei „SysLibRTC" beschrieben. Diese Datei finden Sie nach der Installation im Windows-Startmenü unter <Programme → Moeller Software → easySoft-CoDeSys → Dokumentation → Automation Manuals>.

Sie können die Uhr auch mit den Browser-Befehlen „getrtc" und „setrtc" lesen bzw. setzen. Mehr Informationen finden Sie im Abschnitt „setrtc" auf Seite 60.

Bei Spannungsausfall wird die Uhr mindestens 72 Stunden gepuffert.

Schnittstellen

Programmierschnittstelle zum Anschluss eines PC

Über die Programmierschnittstelle , bestehend aus einem RJ45-Steckverbinder, erfolgt die Kommunikation zwischen der Steuerung und dem Programmiergerät.

Der Steckbverbinder enthält eine RS232-Schnittstelle und bei den Steuerungstypen EC4P- 222-… zusätzlich eine Ethernet-Schnittstelle zur Programmierung.

Tabelle 5: Signalbelegung der Programmierschnittstelle

RJ45		EC4P-221…	EC4P-222-…	
		Signal	Signal	
		RS232	RS232	Ethernet
1 2 3 4 5 6 7 8	1	–	–	Tx+
	2	–	–	Tx-
	3	–	–	Rx+
	4	GND	GND	1)
	5	TxD	TxD	–
	6	–	–	Rx-
	7	GND	GND	1)
	8	RxD	RxD	–

1) Für eine Ethernet-Verbindung ist das GND-Signal nicht erforderlich. Verwenden Sie daher ein Kabel mit unbelegten Anschlusspins 4 und 7!

Transparent-Modus

Um eine Punkt-zu-Punkt-Verbindung (ohne Handshake-Leitungen) zu einem anderen Gerät aufzubauen, schalten Sie die RS232-Schnittstelle mit Hilfe der Funktionen aus der Bibliothek EC_SysLibCom.lib in den Transparentmodus. Im Transparent-Modus wird die Schnittstelle als COM1 angesprochen.

→ Kapitel „RS232-Schnittstelle im Transparent-Modus", Seite 73.

Splitten der RS232-/Ethernet-Schnittstelle

Mit Hilfe der Kabelweiche XT-RJ45-ETH-RS232 können Sie gleichzeitig über die RS232- und die Ethernet-Schnittstelle kommunizieren. Mit dem Kabel EASY-NT-30/80/130 stellen Sie die Verbindung zwischen der Steuerung und der Kabelweiche her. Die Stiftbelegung der RS232- und Ethernet-Steckerbuchse der Kabelweiche entspricht der Stiftbelegung der Programmierschnittstelle nach Tabelle 5.

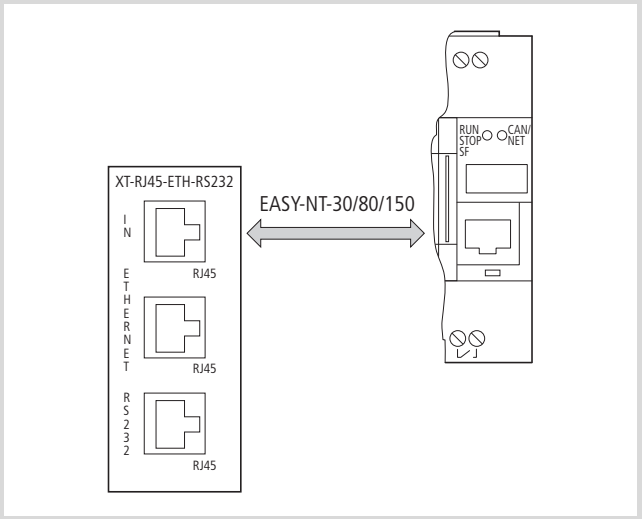


Abbildung 6: Steuerung mit XT-RJ45-ETH-RS232 verbinden

Siehe auch:

→ Abschnitt „CAN/easyNet, PC-Verbindung", Seite 27

→ Kapitel 12 „Verbindungsaufbau PC – EC4-200", Seite 63

Multifunktions-Schnittstelle (MFS)

Über diese Schnittstelle kann die Steuerung alternativ mit folgenden Geräten kommunizieren:

- Speicherkarte

Die Speicherkarte stecken Sie in einen Adapter, den Sie auf diesen Platz stecken.

- Multi-Funktions-Display (MFD) MFD-CP4

Das MFD ist ein von der Steuerung abgesetztes Display mit Bedienmöglichkeiten. Es zeigt den Inhalt des Steuerungsdisplays an. Über die integrierten Tasten können Sie Signale zur Steuerung senden und den Programmablauf steuern. Das MFD lässt sich in eine Schranktüre max. 5 m von der Steuerung entfernt einbauen. Die Geräte werden mit dem Kabel MFD-CP4-800-CAB5 verbunden.

- Terminal/Drucker

Mit Hilfe eines Terminals können Sie alphanummerische Zeichen anzeigen oder eingeben. Zur Ausgabe von Daten kann auch ein Drucker verwendet werden. Das Terminal mit einer RS232-Schnittstelle wird durch das Kabel EASY800-PC-CAB mit der Steuerung verbunden. Das Kabel, mit Bauelementen zur Anpassung an die Steuerungssignale, muss von der Terminalseite her separat versorgt werden. Die Signale und Pinbelegung der Schnittstelle sind nach der RS232-Spezifikation ausgeführt.

Zur Versorgung der Bauelemente im Kabel muss das RTS-Signal im (Terminal-) Gerät gesetzt sein, → Abschnitt „Verbindungskabel EASY800-PC-CAB“ auf Seite 95.

Um Daten zum Terminal zu senden oder zu empfangen, ist die RS232-Schnittstelle, die mit COM2 angesprochen wird, in den Transparent-Modus zu setzen.

→ Kapitel „RS232-Schnittstelle im Transparent-Modus“ auf Seite 73.

Die Funktionen zum Öffnen und Schließen der Schnittstelle sowie zum Senden und Empfangen von Daten sind in der Bibliothek EC_SysLibCom.lib beschrieben.

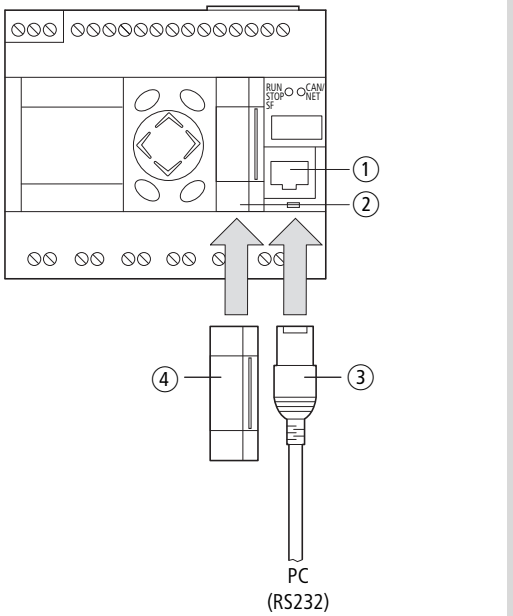
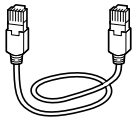
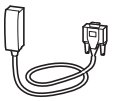
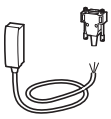


Abbildung 7: Schnittstellen

① Programmierschnittstelle zum Anschluss eines PC
② Multifunktions-Schnittstelle
③ Programmierkabel, z.B.: EU4A-RJ45-CAB1
④ Adapter mit Speicherkarte oder Kabelanschluss

Kabelverbindungen

Die folgende Übersicht zeigt die Kabeltypen, die an die Steuerung angeschlossen werden können und deren Funktion.

Schnitt-stelle	Kabeltyp	Gerät	Funktion
RJ45			
RS232	EU4A-RJ45-CAB1	PC, Terminal/ Printer	Programm, Transparentmodus (COM1)
Ethernet	XT-CAT5-X-2 	PC	Programm
MFS	MFD-CP4-800-CAB5	MFD-CP4	Display-Verlängerung
	easy800-USB-CAB	PC	Programm
	easy800-PC-CAB 	Terminal/ Printer	Transparentmodus (COM2)
	easy800-MO-CAB 	PC, Terminal/ Printer	Programm, Transparentmodus (COM1)

CAN/easyNet-Schnittstellen

Die Steuerung besitzt eine CAN/easyNet-Schnittstelle mit zwei Steckplätzen, deren Anschlüsse intern verbunden sind.

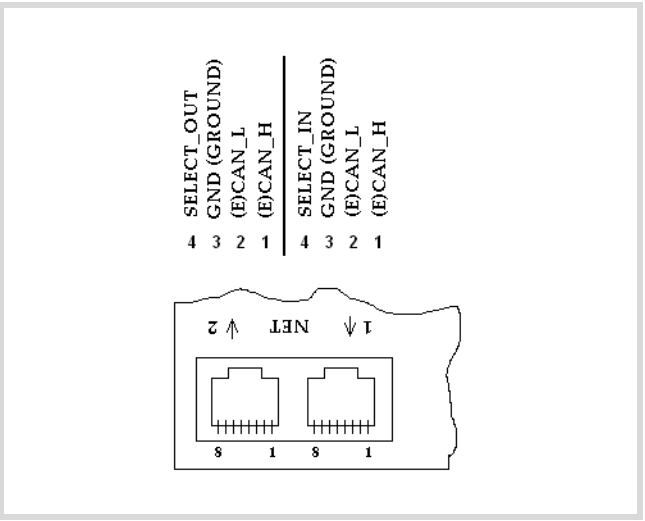


Abbildung 8: CAN/easyNet-Schnittstellen

CANopen

Die CAN-Schnittstelle wird als CANopen-Schnittstelle nach CIA-Spezifikation DS301V4.0 ausgelegt. Die Steuerung kann sowohl als NMT-Master als auch als CAN-Device an CAN-Netzwerken betrieben werden. Als CAN-Device benötigt die Steuerung eine Adresse (= Node-ID) zur Identifikation am Bus. Mögliche Node-Ids sind 1…127. Die Konfiguration des Masters und des Device nehmen Sie im Steuerungskonfigurator vor.

→ Abschnitt „Netzwerk CAN/easyNet“, Seite 93.

3 Erweiterungsgeräte

Die Erweiterungsgeräte schließen Sie über die easyLink-Schnittstelle direkt an die Steuerung an. Mit den folgenden Erweiterungsgeräten können Sie die Anzahl der Ein- und Ausgänge der Steuerungen erhöhen.

Typenübersicht Erweiterungsgeräte

TYP	Versorgungsspannung	Eingänge	Ausgänge
EASY618-AC-RE	100…230 V AC	12 AC	6 Relais
EASY618-DC-RE	24 V DC	12 DC	6 Relais
EASY620-DC-TE	24 V DC	12 DC	8 Transistor
EASY202-RE	–	–	2 Relais mit gemeinsamer Versorgung für mehrere Ausgänge

Mit Hilfe des Koppelgerätes EASY200-EASY kann ein Erweiterungsgerät über eine 30 m lange 2-Draht- oder Mehrader-Leitung dezentral mit der Steuerung verbunden werden.

Ein-/Ausgangs-Übersicht

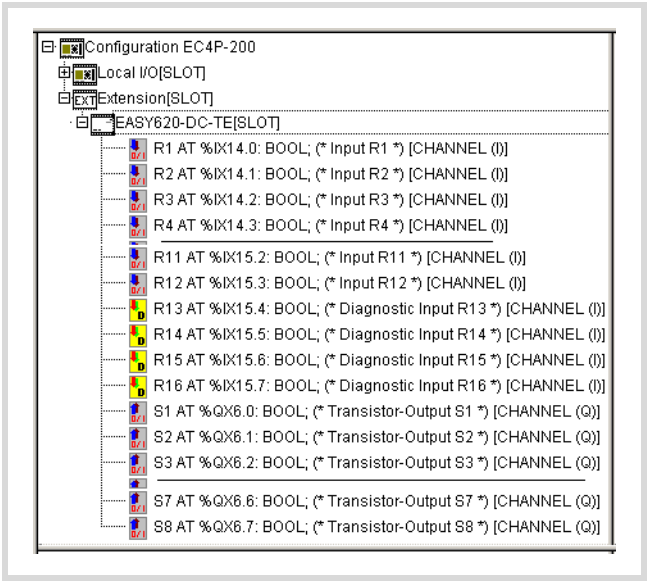


Abbildung 9: I/Os der EASY620-DC-TE

Eingänge

Tabelle 6: Anzahl der Eingänge und symbolische Operanden

TYP	Anzahl	Operand
EASY6..-..-..	12	R1,…,R12
EASY620-DC-TE	4 (Diagnose)	R13,…,R16

Die Auswahl treffen Sie im Anwenderprogramm, indem Sie die entsprechende Syntax verwenden, die im Steuerungskonfigurator vorgegeben ist.

Diagnose-Eingänge

Die Eingänge R15, R16 stellen Ihnen zusätzliche Informationen zur Verfügung:

Tabelle 7: Funktionen der Diagnose-Eingänge

Eingang	Funktion
R13,R14	keine Bedeutung
R15	Ausgänge S1, S2, S3, S4: 0: kein Kurzschluss, toggelt: Kurzschluss
R16	Ausgänge S5, S6, S7, S8: 0: kein Kurzschluss, toggelt: Kurzschluss

Die Eingänge können Sie im Programm mit den symbolischen Operanden abfragen.

Ausgänge

Tabelle 8: Anzahl der Ausgänge und symbolische Operanden

TYP	Anzahl	Operand
EASY618	6	S1,…, S6
EASY620	8	S1,…, S8
EASY202-RE	2	S1,S2

Die Transistor-Ausgänge verfügen über eine Kurzschlussüberwachung. Tritt an einem der Ausgänge ein Kurzschuss auf, wird dies über die Diagnose-Eingänge R15/R16 gemeldet. R15 wird auf 1-Signal gesetzt, wenn an den Ausgängen S1 bis S4 ein Kurzschluss auftritt. Tritt an den Ausgängen S5 bis S6 ein Kurzschluss auf, toggelt der Eingang R16.

4 Montage

Bauen Sie die Steuerung in einen Schaltschrank, einen Installationsverteiler oder in ein Gehäuse ein, sodass die Anschlüsse der Versorgungsspannung und die Klemmenanschlüsse im Betrieb gegen direktes Berühren geschützt sind.

Sie können die Steuerung senkrecht oder waagerecht auf eine Hutschiene nach IEC/EN 60715 oder mit Gerätefüßen auf der Montageplatte befestigen.

Um die Verdrahtung zu erleichtern, halten Sie auf den Klemmenseiten einen Abstand von mindestens 3 cm zur Wand oder zu benachbarten Geräten ein.

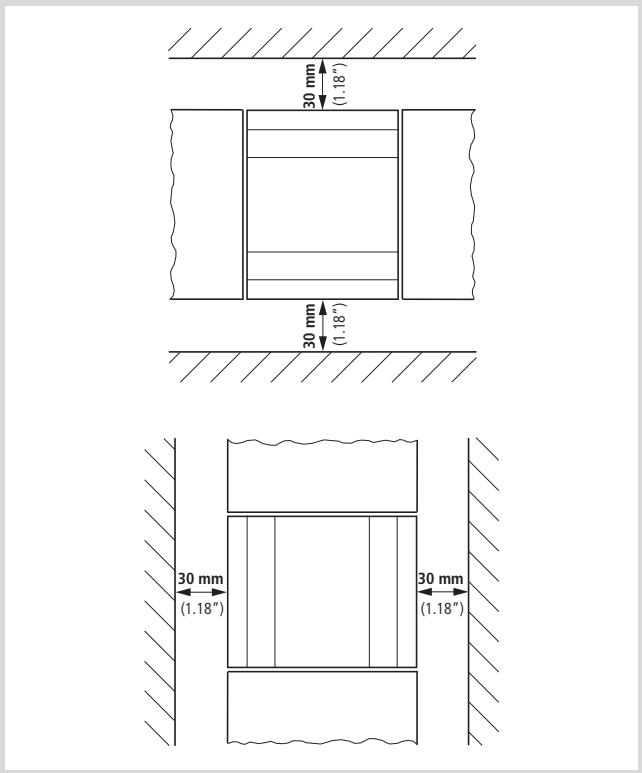


Abbildung 10: Abstände zur Verdrahtung einhalten

Montage auf Hutschiene

- ▶ Setzen Sie das Gerät schräg auf die Oberkante der Hutschiene auf. Drücken Sie das Gerät leicht nach unten und an die Hutschiene, bis es über die Unterkante der Hutschiene schnappt. Durch den Federmechanismus rastet es automatisch ein.
- ▶ Prüfen Sie das Gerät kurz auf festen Halt.

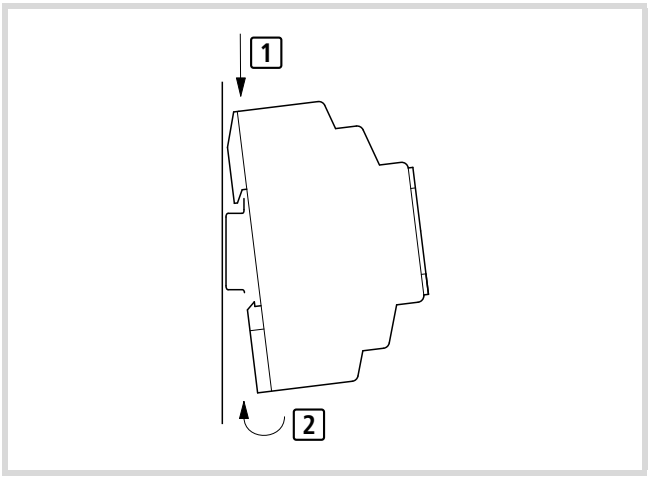


Abbildung 11: Montage auf Hutschiene

Die senkrechte Montage auf einer Hutschiene wird in gleicher Weise ausgeführt.

Montage auf Montageplatte

Für die Schraubmontage benötigen Sie Gerätefüße, die Sie auf der Rückseite des Gerätes einsetzen können. Die Gerätefüße erhalten Sie als Zubehör.

→ Für ein Gerät mit vier Befestigungspunkten reichen drei Gerätefüße.

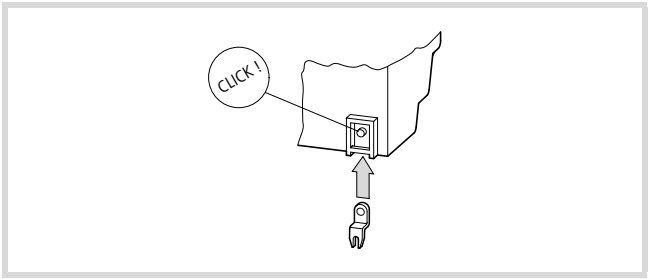


Abbildung 12: Gerätefuß einsetzen

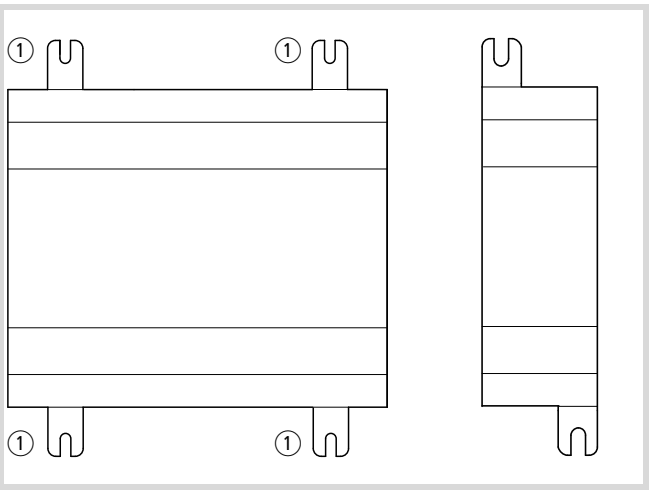


Abbildung 13: Geräte anschrauben

① Gerätefüße

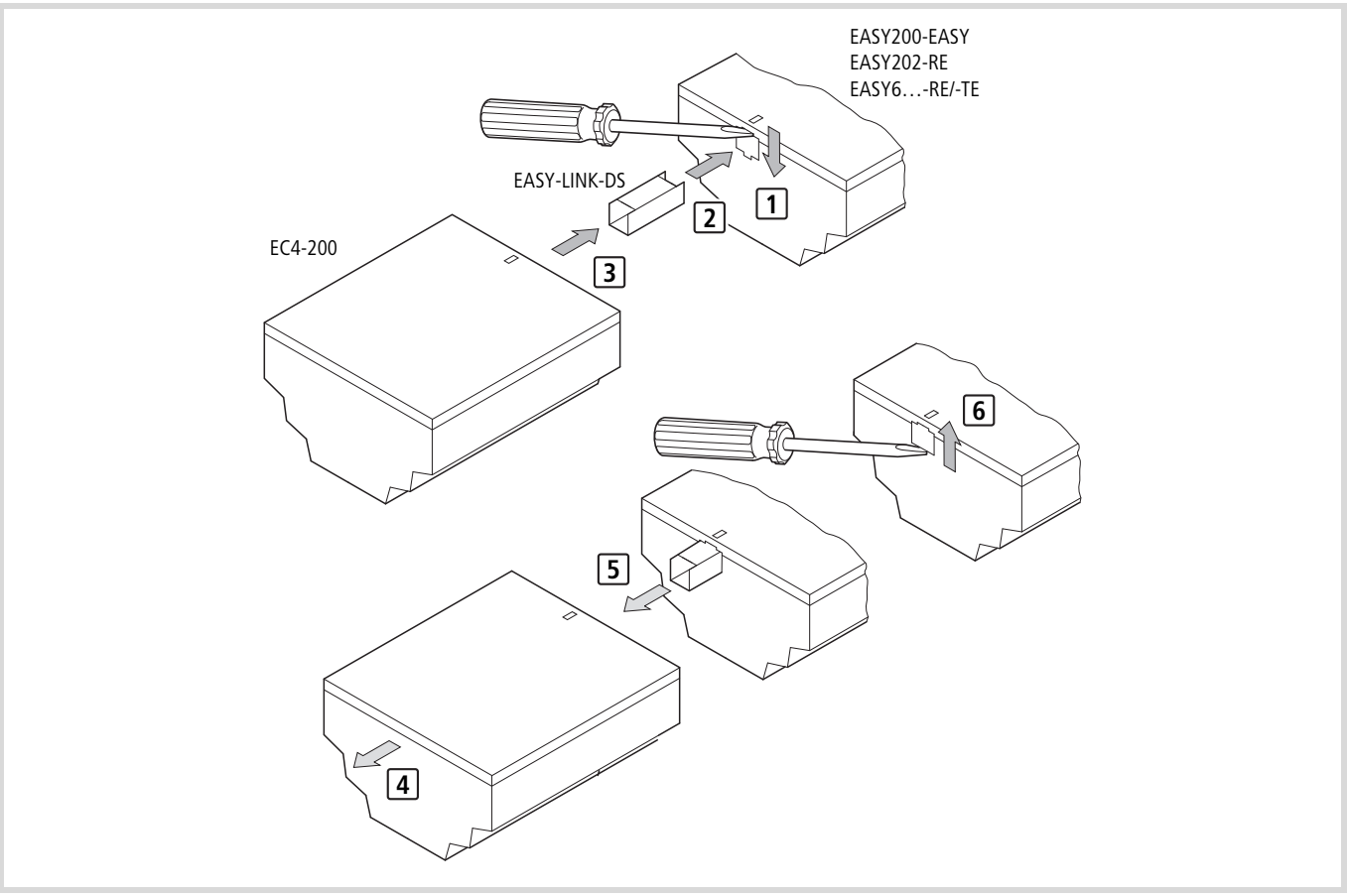


Abbildung 14: Erweiterung/Netzwerk-Kopplung an EC4-200 anschließen

5 Installation

Versorgungsspannung anschließen

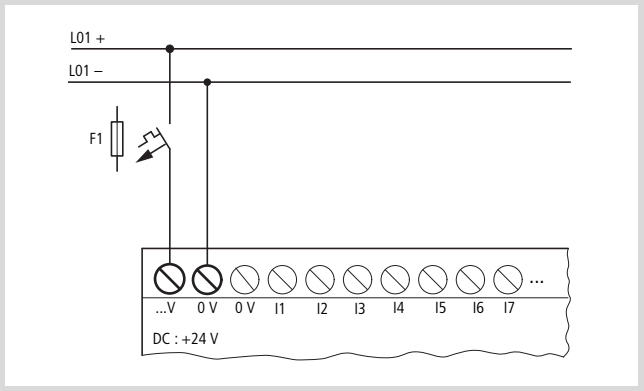


Abbildung 15: Versorgungsspannung anschließen
Die beiden 0-V-Anschlüsse sind intern miteinander verbunden!

→ Die EC4-200 ist verpolungsgeschützt.

→ Die erforderlichen Anschlussdaten finden Sie im Kapitel „Technische Daten", Seite 96.

Leitungsschutz
Schützen Sie die Versorgungsleitungen durch einen Leitungsschutz (F1) von mindestens 1 A (T) an.

→ Beim ersten Einschalten verhält sich die Steuerung kapazitiv. Das Schaltgerät und das Versorgungsgerät zum Einschalten der Versorgungsspannung müssen dafür vorgesehen sein; d. h. keine Reedrelaiskontakte, keine Näherungsinitiatoren.

Digital-Eingänge anschließen

Schließen Sie Taster, Schalter, 3- oder 4-Draht-Näherungsschalter an den Eingangsklemmen I1 bis I12 an. Setzen Sie wegen des hohen Reststroms keine 2-Draht-Näherungsschalter ein.

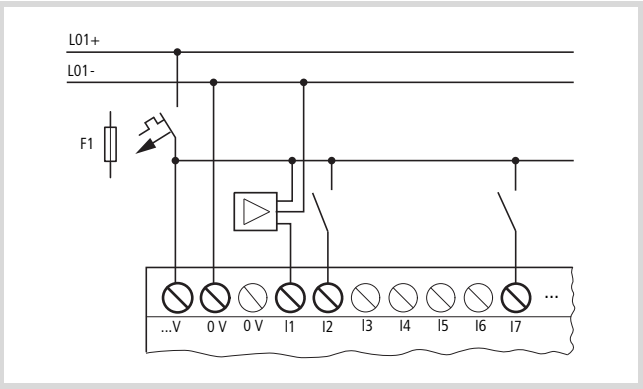


Abbildung 16: Digital-Eingänge anschließen

Analog-Eingänge anschließen

Über die Eingänge I7, I8, I11 und I12 können Sie auch analoge Spannungen im Bereich 0 bis 10 V anschließen.

Die Auflösung beträgt 10 Bit = 0 bis 1023.

⚠ **Vorsicht!**
Beachten Sie beim Verlegen und Anschließen der Analogleitungen die folgende Hinweise:

- ▶ Verwenden Sie geschirmte, paarweise verdrillte Leitungen, um Störeinkopplungen auf die Analogsignale zu vermeiden.
- ▶ Erden Sie den Schirm der Leitungen bei kurzen Leitungslängen beidseitig und vollflächig. Ab einer Leitungslänge von etwa 30 m kann die beidseitige Erdung zu Ausgleichsströmen zwischen beiden Erdungsstellen und damit zur Störung von Analogsignalen führen. Erden Sie die Leitung in diesem Fall nur einseitig.
- ▶ Verlegen Sie Signalleitungen nicht parallel zu Energieleitungen.
- ▶ Schließen Sie induktive Lasten, die Sie über die Ausgänge schalten, an eine separate Versorgungsspannung an oder verwenden Sie eine Schutzbeschaltung für Motoren und Ventile. Wenn die Steuerung mit Motoren, Magnetventilen oder Schützen über die gleiche Versorgungsspannung betrieben wird, kann das Schalten zu einer Störung der analogen Eingangssignale führen.

Die folgenden Schaltungen zeigen Beispiele für den Einsatz der Analogwert-Erfassung.

→ Stellen Sie eine galvanische Verbindung des Bezugspotentials her. Verbinden Sie die 0 V des Netzteiles von den in den Beispielen dargestellten Sollwertgeber bzw. den verschiedenen Sensoren mit den 0 V der Versorgungsspannung.

Sollwertgeber anschließen

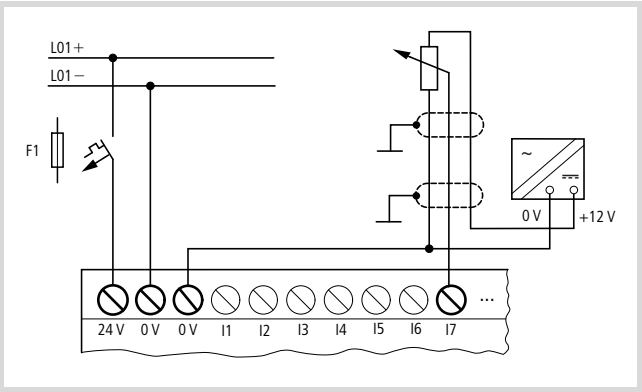


Abbildung 17: Sollwertgeber

Setzen Sie ein Potentiometer mit dem Widerstandswert ≦ 1 kΩ, z. B. 1 kΩ, 0,25 W ein.

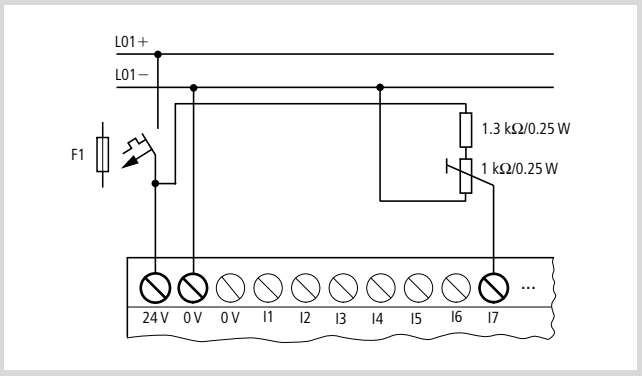


Abbildung 18: Sollwertgeber mit vorgeschaltetem Widerstand

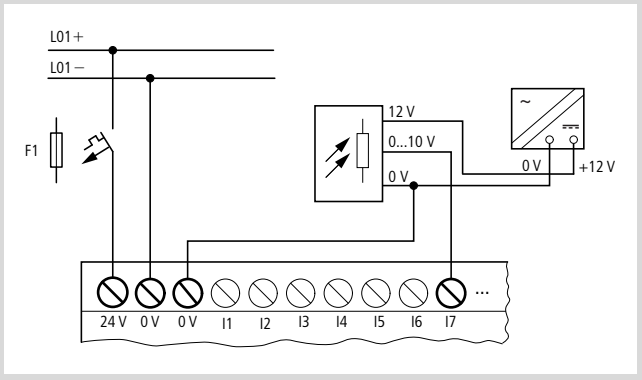


Abbildung 19: Helligkeitssensor

Temperatursensor anschließen

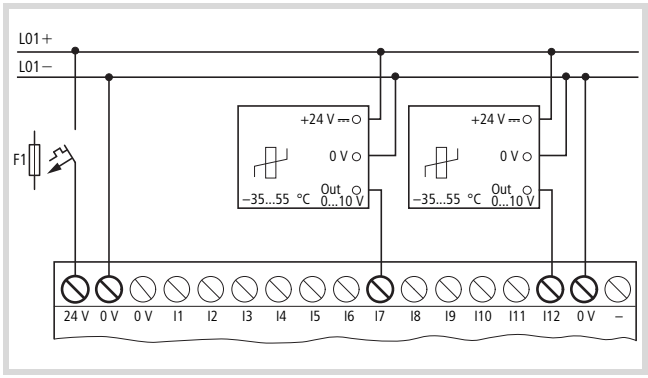


Abbildung 20: Temperatursensor

20-mA-Sensor anschließen

Der Anschluss eines 4 bis 20 mA (0 bis 20 mA)-Sensors ist mit einem externen Widerstand von 500 Ω problemlos möglich.

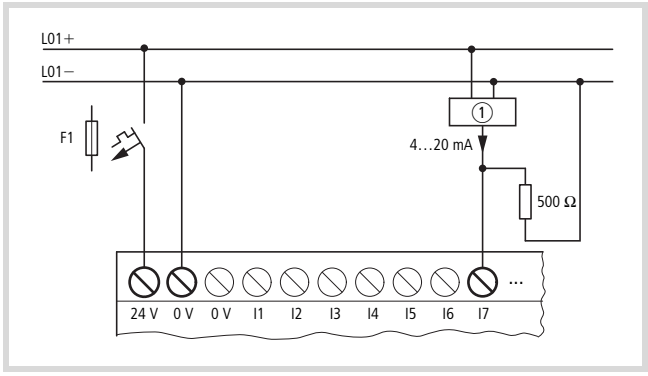


Abbildung 21: 20-mA-Sensor

① Analog-Sensor

Folgende Werte ergeben sich:

- 4 mA = 1,9 V
- 10 mA = 4,8 V
- 20 mA = 9,5 V

(nach $U = R \times I = 478\ \Omega \times 10\ \text{mA} \sim 4{,}8\ \text{V}$)

Impuls-/Inkrementalgeber anschließen

Die Eingänge I1 bis I4 sind so ausgelegt, dass schnelle Signale von Impuls-/Inkrementalgebern gezählt werden können.

Sie können alternativ folgende Geber anschließen:

- 1 × Impulsgeber (32 Bit)
- 2 × Impulsgeber (16 Bit)
- 1 × Inkrementalgeber (32 Bit).

Impulsgeber anschließen

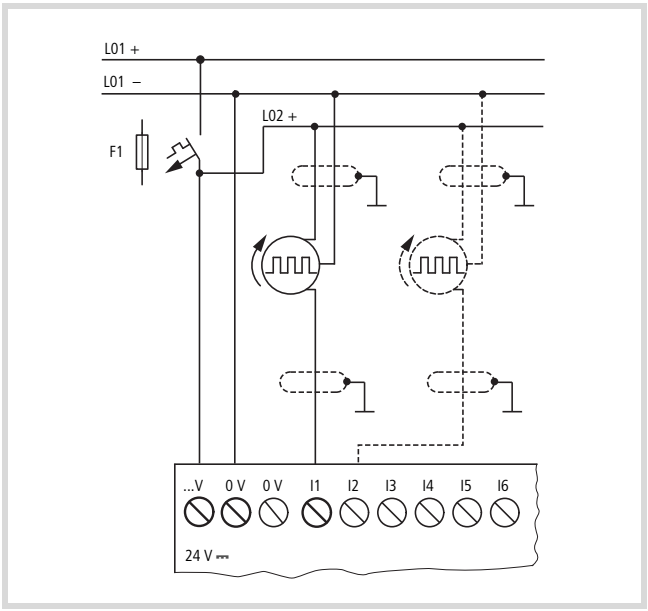


Abbildung 22: Impulsgeber anschließen

Die Abbildung zeigt den Anschluss eines Impulsgebers, dessen Impulse auf den Eingang I1 geführt werden. Ein interner Zähler verarbeitet die Impulse. Sie können zwischen einem 16- Bit-Zähler (max. 65535) und 32 Bit-Zähler (max. 4294967295) wählen. Der Impulsgeber für den 32-Bit-Zähler darf nur an I1 angeschlossen werden. Nur wenn an I1 ein 16- Bit-Zähler verwendet wurde, kann ein weiterer Impulsgeber (16 Bit) an I2 angeschlossen werden.

Inkrementalgeber anschließen

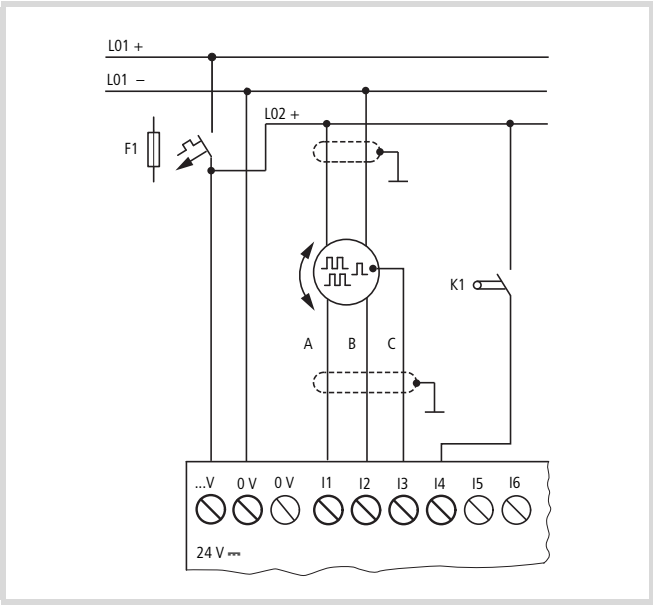


Abbildung 23: Inkrementalgeber anschließen

A, B: Rechteckförmige Inkremental-Signale, die elektrisch um 90 Grad phasenverschoben sind
C: Referenz-Signal
K1: Referenzfenster-Schalter

Ausgänge anschließen

Mit den Relais- oder Transistor-Ausgängen schalten Sie Lasten wie z. B. Leuchtstoffröhren, Glühlampen, Schütze, Relais oder Motoren. Beachten Sie vor der Installation die technischen Grenzwerte und Daten der Ausgänge (→ Seite 100, 101).

Relais-Ausgänge anschließen

EC4P-221/222-MR…, EASY6..-DC-RE

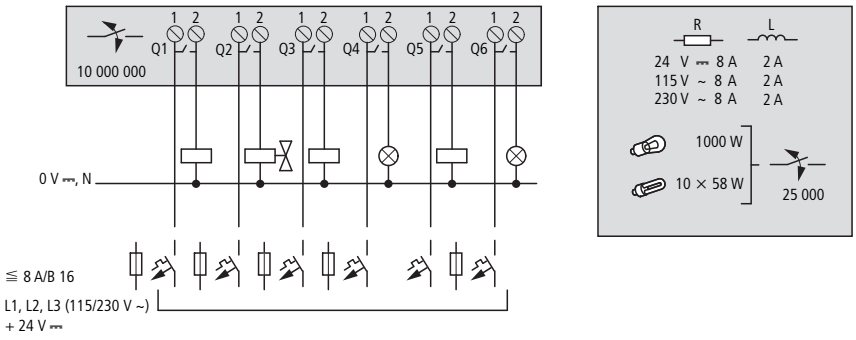


Abbildung 24: Relais-Ausgänge EC4P-221/222-MR…

Im Gegensatz zu den Eingängen können Sie an die Relais-Ausgänge EC4P-221/222-MR…, EASY6..-..RE verschiedene Außenleiter anschließen.

Vorsicht!
Halten Sie die obere Spannungsgrenze von 250 V AC am Kontakt eines Relais ein. Eine höhere Spannung kann zu Überschlägen am Kontakt führen und damit das Gerät oder eine angeschlossene Last zerstören.

Transistor-Ausgänge anschließen

EC4P-221/222-MT…, EASY6…-DC-TE

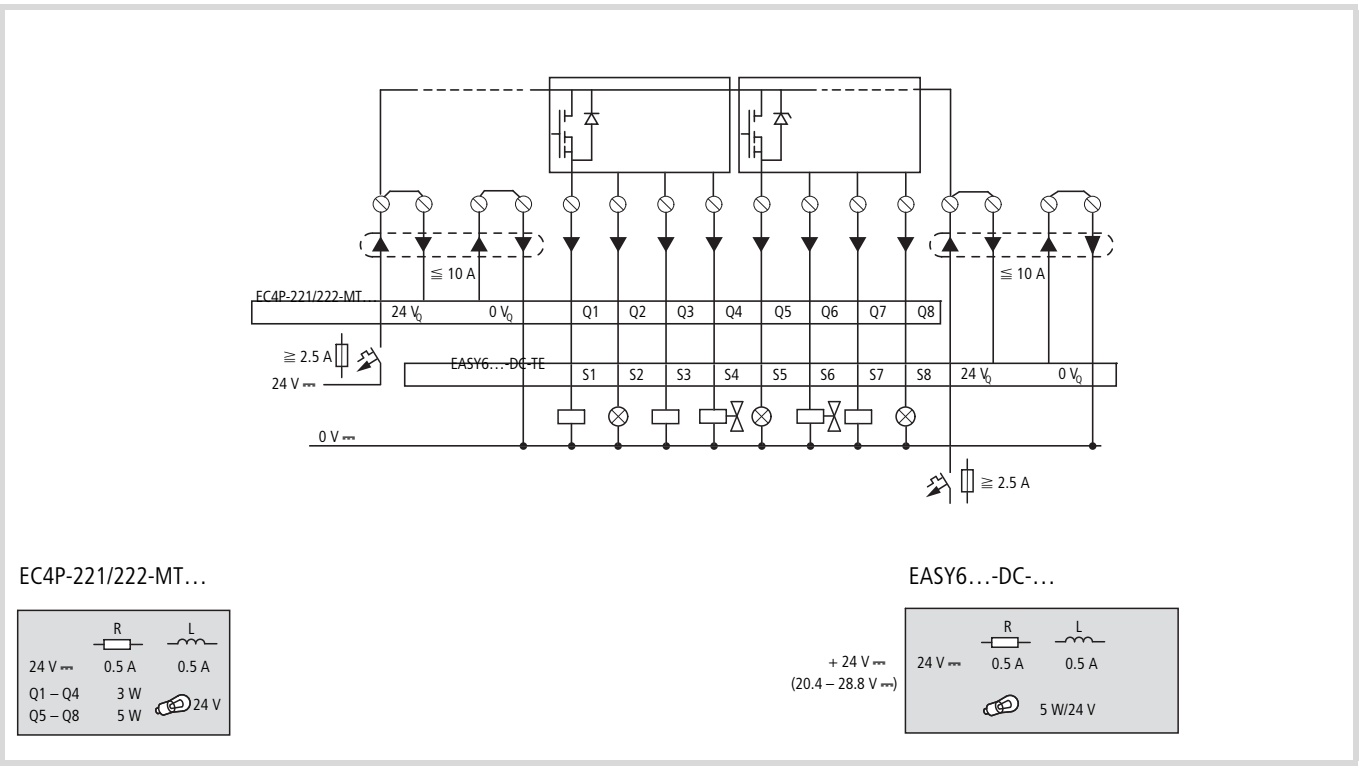


Abbildung 25: Transistor-Ausgänge EC4P-221/222-MT…, EASY6..-DC-TE

Parallelschaltung:
Zur Leistungserhöhung können bis zu maximal vier Ausgänge parallel geschaltet werden. Dabei addiert sich der Ausgangsstrom auf maximal 2 A.

Vorsicht!
Beim Abschalten von induktiven Lasten ist Folgendes zu beachten:
Schutzbeschaltete Induktivitäten verursachen weniger Störungen im gesamten elektrischen System. Es empfiehlt sich generell, die Schutzbeschaltung möglichst nahe an der Induktivität anzuschließen.

Vorsicht!
Nur innerhalb einer Gruppe (Q1 bis Q4 oder Q5 bis Q8) dürfen die Ausgänge parallel geschaltet werden; z. B. Q1 und Q3 oder Q5, Q7 und Q8. Parallel geschaltete Ausgänge müssen gleichzeitig angesteuert werden.

Werden Induktivitäten nicht schutzbeschaltet, gilt:
Es dürfen nicht mehrere Induktivitäten gleichzeitig abgeschaltet werden, um die Treiberbausteine im ungünstigsten Fall nicht zu überhitzen. Wird im NOT-AUS-Fall die +24-V-DC-Versorgung mittels Kontakt abgeschaltet und kann dabei mehr als ein angesteuerter Ausgang mit Induktivität abgeschaltet werden, müssen Sie die Induktivitäten mit einer Schutzbeschaltung versehen (→ folgende Abbildungen).

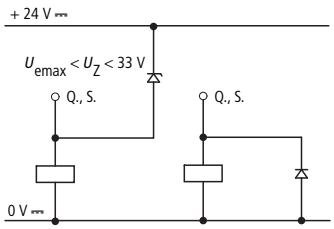


Abbildung 26: Induktivität mit Schutzbeschaltung

Verhalten bei Kurzschluss/Überlast

Tritt Kurzschluss oder Überlast an einem Transistor-Ausgang auf, schaltet dieser Ausgang ab. Nach einer von der Umgebungstemperatur und der Höhe des Stromes abhängigen Abkühlzeit schaltet der Ausgang erneut bis zur maximalen Temperatur ein. Besteht der Fehler weiterhin, schaltet der Ausgang so lange aus und ein, bis der Fehler behoben ist, bzw. die Versorgungsspannung ausgeschaltet wird.

Analog-Ausgang anschließen

Die EC4-200 besitzt einen Analog-Ausgang QA 01, 0 V bis 10 V DC, 10 Bit Auflösung (0 bis 1023). Über den Analog-Ausgang können Sie Servoventile oder andere Stellglieder ansteuern.

Servoventil anschließen

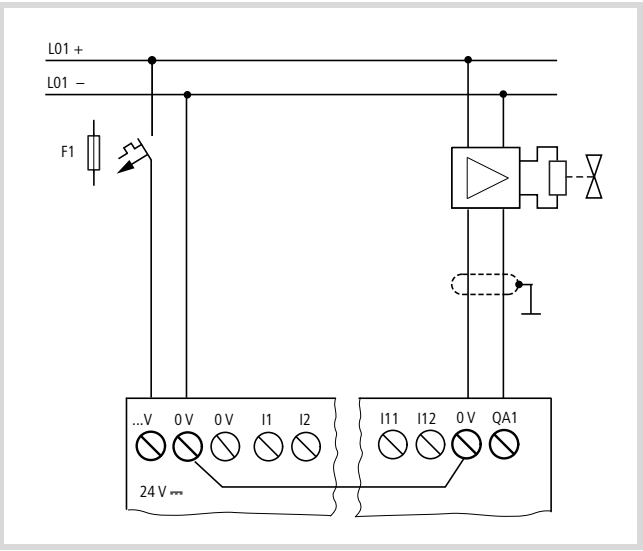


Abbildung 27: Servoventil anschießen

Sollwertvorgabe für einen Antrieb

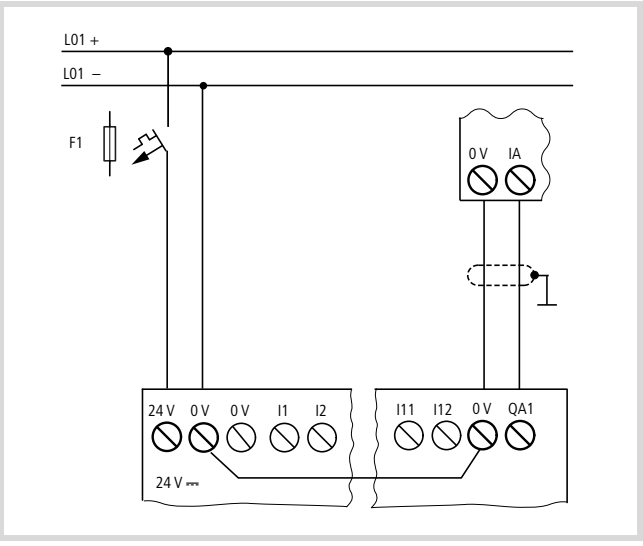


Abbildung 28: Sollwertvorgabe für einen Antrieb

Vorsicht!
Analogsignale sind störempfindlicher als digitale Signale, sodass die Signalleitungen sorgfältiger verlegt und angeschlossen werden müssen. Unsachgemäßer Anschluss kann zu nicht gewollten Schaltzuständen führen.

Speicherkarte, CAN/easyNet, PC-Verbindung

Um eine Speicherkarte zu stecken bzw. eine CAN/easyNet oder eine PC-Verbindung herzustellen, müssen Sie zunächst die entsprechende Verschlusskappe entfernen.

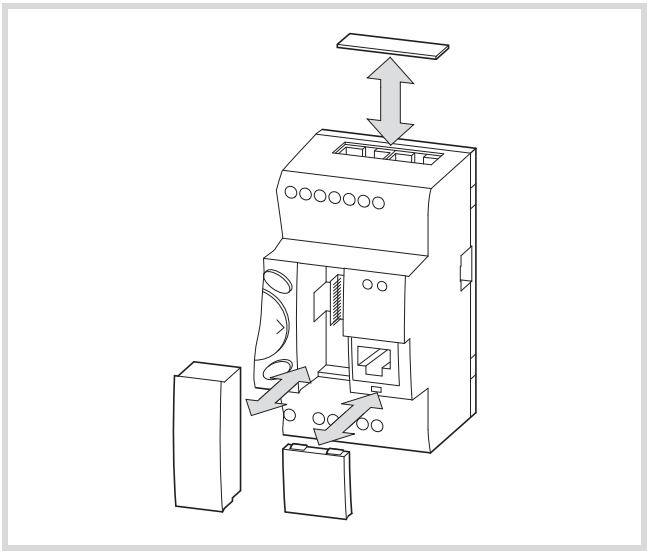


Abbildung 29: Entfernen der Verschlusskappe/des Adapters:
oben: für CAN/easyNet-Verbindung
unten links: Adapter für Speicherkarte
unten rechts: PC-Verbindung

Speicherkarte stecken oder ziehen

Die Speicherkarte befindet sich im Adapter ③.

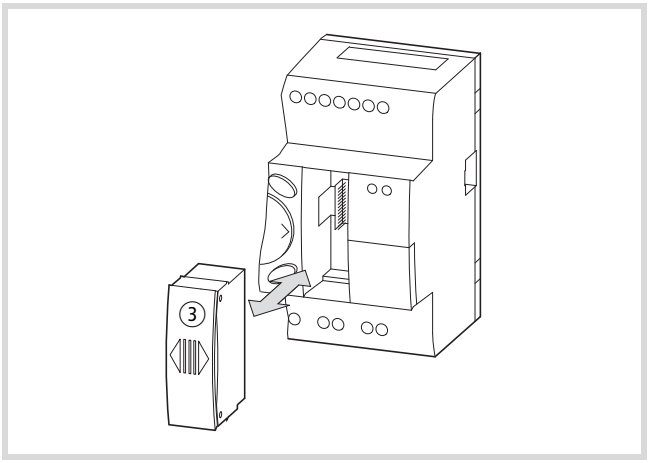


Abbildung 30: Adapter mit Speicherkarte

- ▶ Um die Speicherkarte zu stecken, drücken Sie diese bis zum Einrasten ein.
- ▶ Um die Speicherkarte zu ziehen, müssen Sie diese drücken, bis sie entrastet.

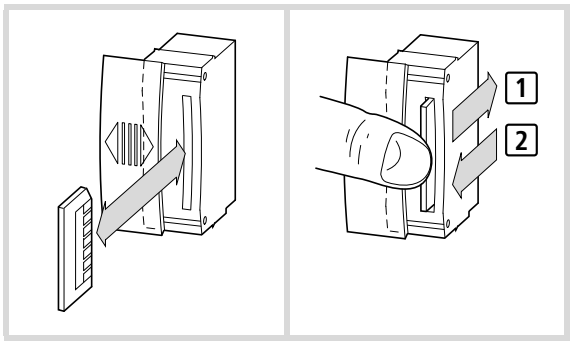


Abbildung 31: Speicherkarte stecken/ziehen

CAN/easyNet, PC-Verbindung

- ▶ Den Stecker für die CAN/easyNet-Verbindung stecken Sie in die Öffnung oben am Gerät ①.
- ▶ Den Stecker für die PC-Verbindung stecken Sie in die Öffnung unten rechts am Gerät ②.

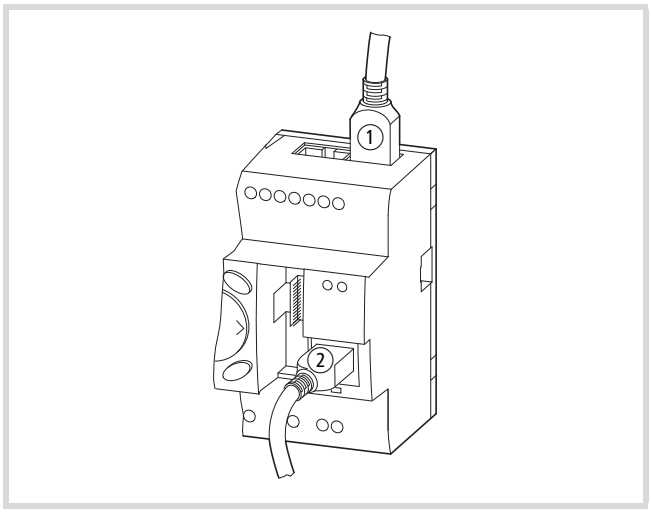


Abbildung 32: Stecker für die CAN/easyNet-Verbindung ① und die PC-Verbindung ②

→ Weitere Informationen → Abschnitt „Netzwerk CAN/easyNet“, Seite 93.

Achtung!
So schützen Sie die EC4-200 und Speicherkarte vor Entladung statischer Elektrizität: Entladen Sie sich elektrostatisch gegen eine geerdete Fläche, bevor Sie die Speicherkarte stecken oder ziehen.

Erweiterungen/Netzwerk-Kopplungen anschließen

Zentrale Erweiterung

▶ Schließen Sie die Geräte zur Erweiterung oder Netzwerk-Kopplung über den Verbindungsstecker EASY-LINK-DS an.

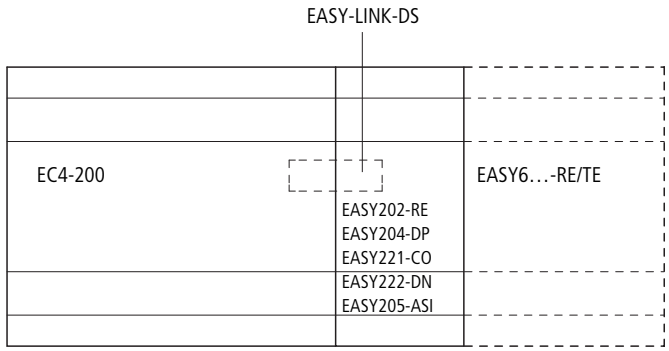


Abbildung 33: Erweiterungsgeräte mit EC4-200 verbinden

Zwischen dem Basisgerät und einem Erweiterungsgerät besteht eine elektrische Trennung (Trennung immer im lokalen Anschluss der Erweiterung):

- einfache Trennung 400 V AC (+10 %)
- sichere Trennung 240 V AC (+10 %)

Wird der Wert 400 V AC +10 % überschritten, kann dies zur Zerstörung der Geräte und zu Fehlfunktionen der Anlage oder Maschine führen!

→ Basisgerät und Erweiterungsgerät können mit verschiedenen DC-Spannungsversorgungen gespeist werden.

Dezentrale Erweiterung

Bei der dezentralen Erweiterung können Sie die Erweiterungsgeräte bis zu 30 m entfernt vom Basisgerät installieren und betreiben.

Warnung!
Die 2-Draht- oder Mehrader-Leitung zwischen den Geräten muss die Isolationsspannung einhalten, die für die Installationsumgebung notwendig ist. Anderenfalls kann ein Fehlerfall (Erdschluss, Kurzschluss) zur Zerstörung der Geräte oder zu Personenschäden führen.

Eine Leitung z. B. NYM-0 mit einer Betriebsbemessungsspannung von U_e = 300/500 V AC reicht im Normalfall aus.

→ Die Klemmen E+ und E– des EASY200-EASY sind kurzschluss- und verpolungssicher. Die Funktionsfähigkeit ist nur gegeben, wenn E+ mit E+ und E– mit E– verbunden ist.

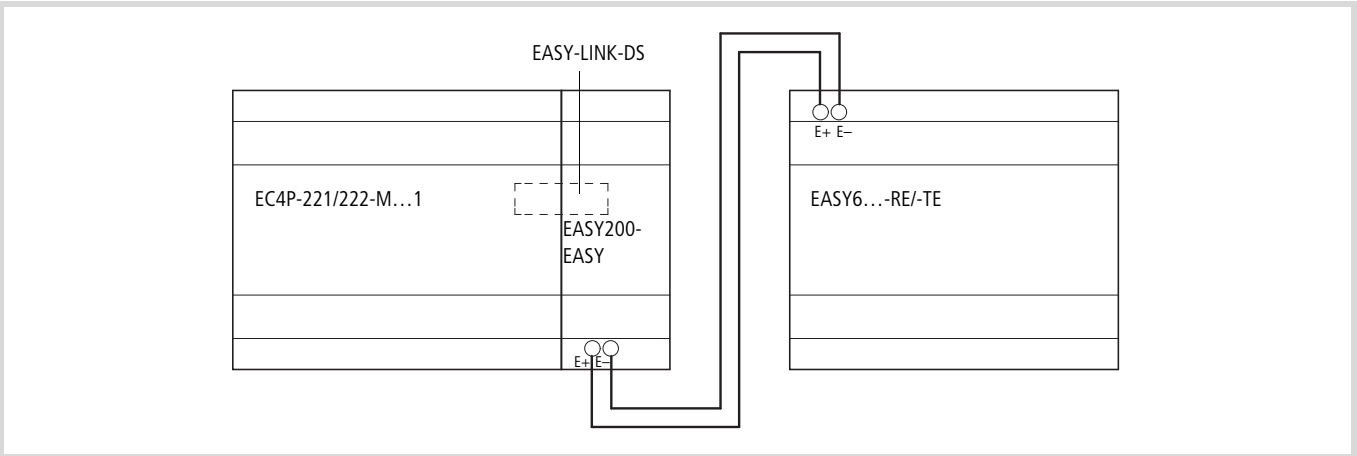
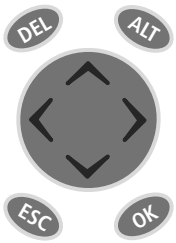


Abbildung 34: Dezentrale Erweiterungen an EC4-200 anschließen

6 Bedienung

Das folgende Kapitel beschreibt die Bedienung der Tasten und die Anzeige (Display) auf der Frontplatte.

Tastenfeld



DEL: Löschen

ALT: Sonderfunktionen, Statusanzeige

Cursortasten 〈 〉 ∧ ∨:
Cursor bewegen
Menüpunkte wählen
Zahlen und Werte einstellen

OK: Weiterschalten, Speichern

ESC: Zurück wechseln, Abbrechen

Menüführung und Eingabe von Werten

DEL und ALT — Sondermenü aufrufen

OK — Zur nächsten Menüebene wechseln
Menüpunkt aufrufen
Eingaben aktivieren, ändern, speichern

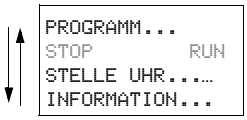
ESC — Zur vorherigen Menüebene wechseln
Eingaben ab letztem **OK** zurücknehmen

∧ Menüpunkt wechseln
∨ Wert ändern
〈 〉 Stelle wechseln

P-Tasten-Funktion:

〈 Eingang P1, ∧ Eingang P2
〉 Eingang P3, ∨ Eingang P4

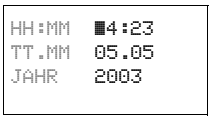
Menüpunkte wählen oder umschalten



Cursor ∧ ∨

OK — wählen oder umschalten

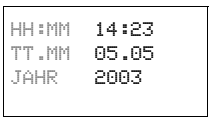
Cursor-Anzeige



Der Cursor blinkt im Wechsel.

Voll-Cursor ▇/:

- Cursor mit 〈 〉 ∧ ∨ bewegen

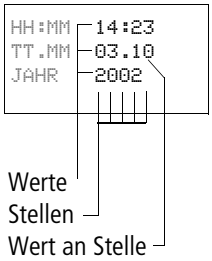


Wert M/ṀM

- Position mit 〈 〉 ändern
- Werte mit ∧ ∨ ändern

Blinkende Werte werden im Handbuch grau dargestellt.

Wert einstellen



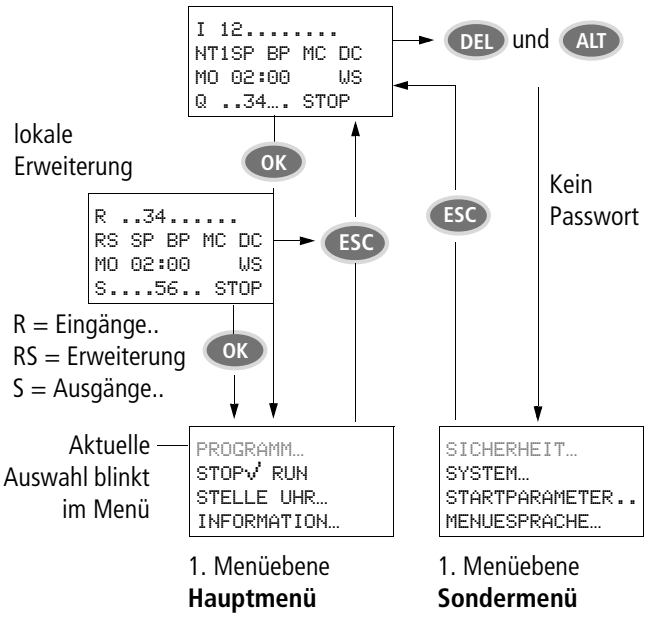
Wert wählen ∧ ∨
Stelle wählen 〈 〉
Wert an Stelle ändern ∧ ∨

OK — Einstellung speichern

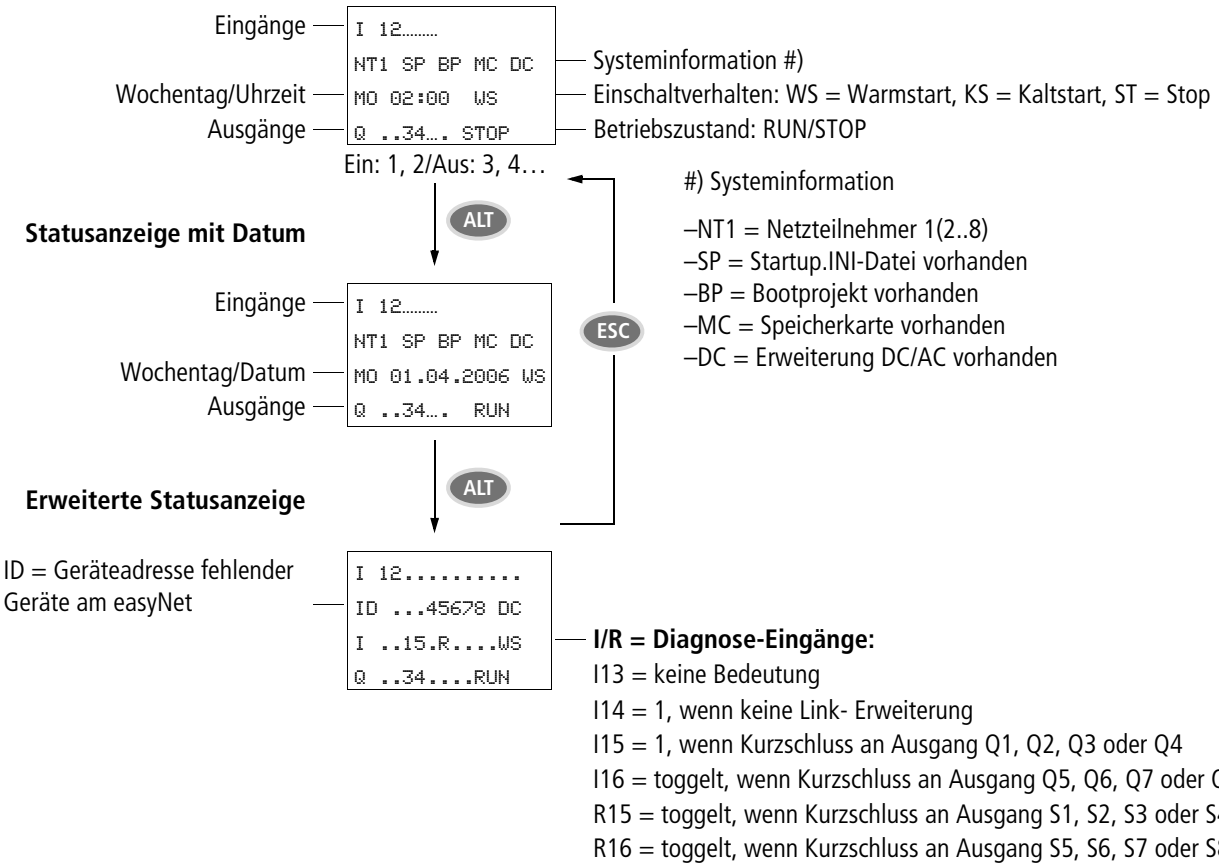
ESC — vorherigen Wert behalten

Haupt- und Sondermenü wählen

Statusanzeige



Statusanzeige mit Uhrzeit



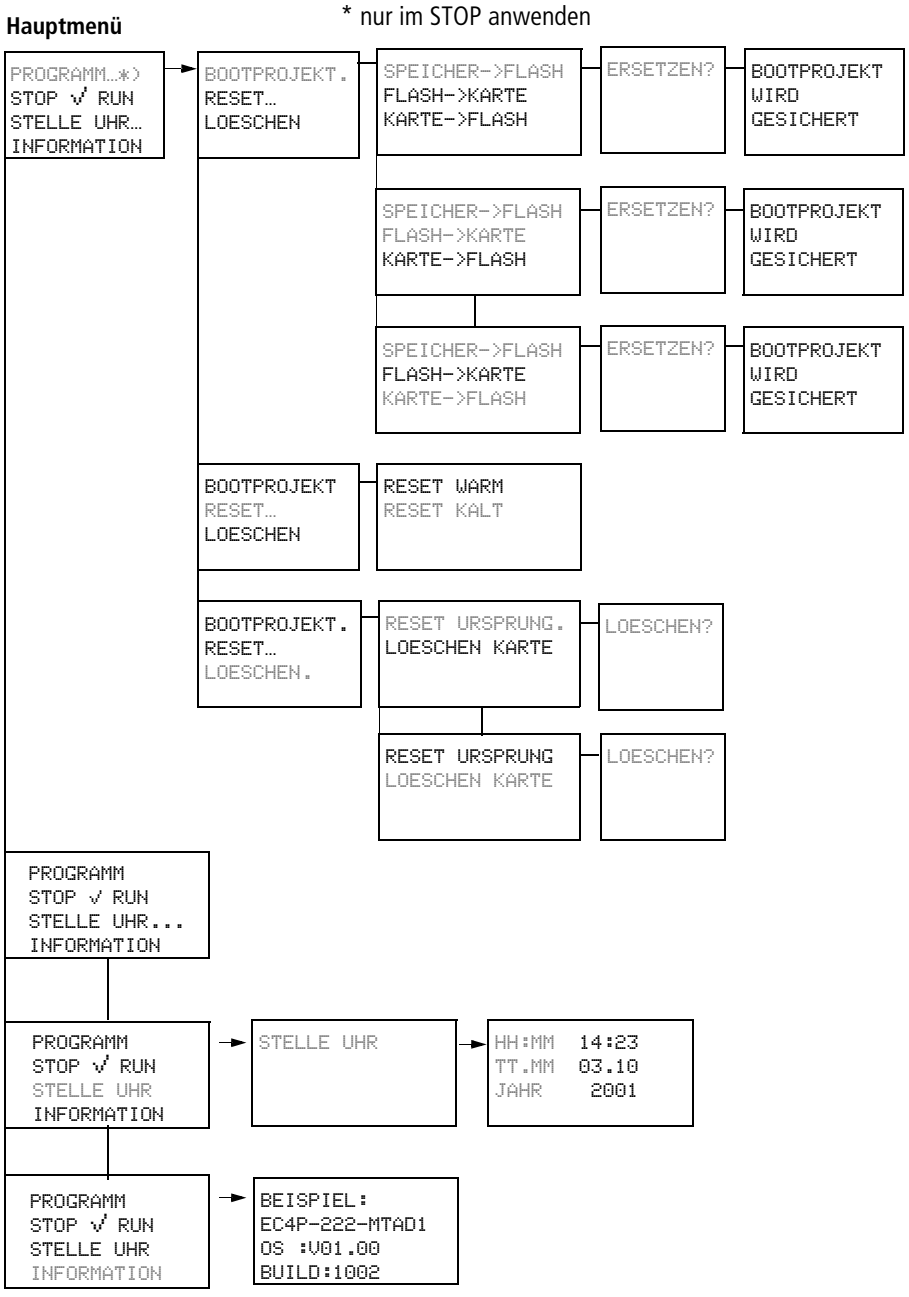
#) Systeminformation

–NT1 = Netzteilnehmer 1(2..8)
–SP = Startup.INI-Datei vorhanden
–BP = Bootprojekt vorhanden
–MC = Speicherkarte vorhanden
–DC = Erweiterung DC/AC vorhanden

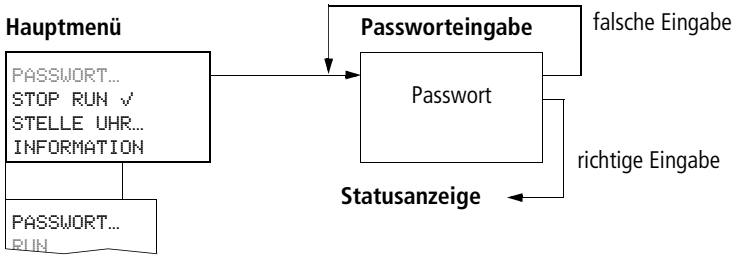
Menüstruktur

Hauptmenü ohne Passwortschutz

▶ Mit Betätigen von **OK** gelangen Sie ins Hauptmenü.

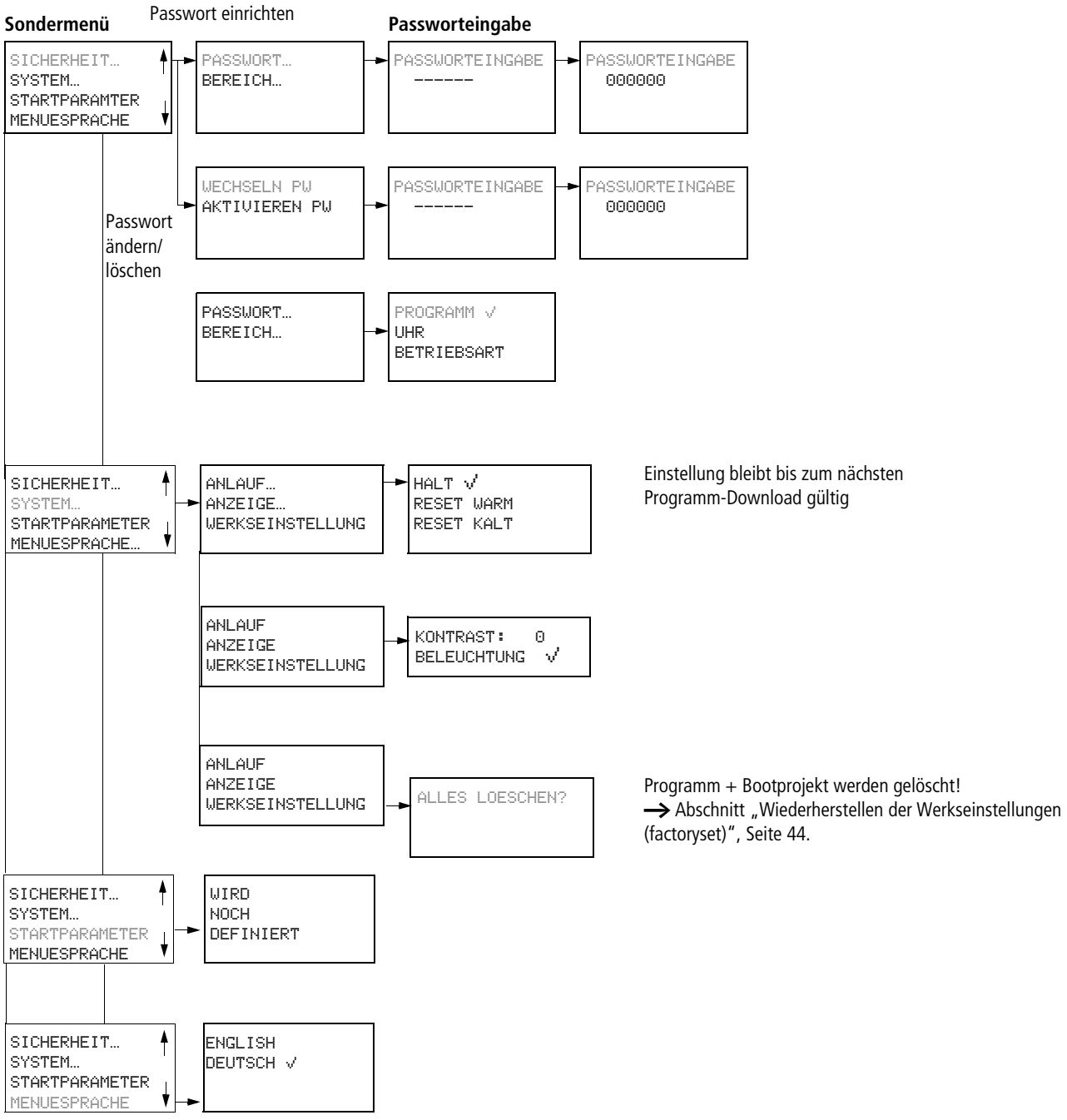


Hauptmenü mit Passwortschutz

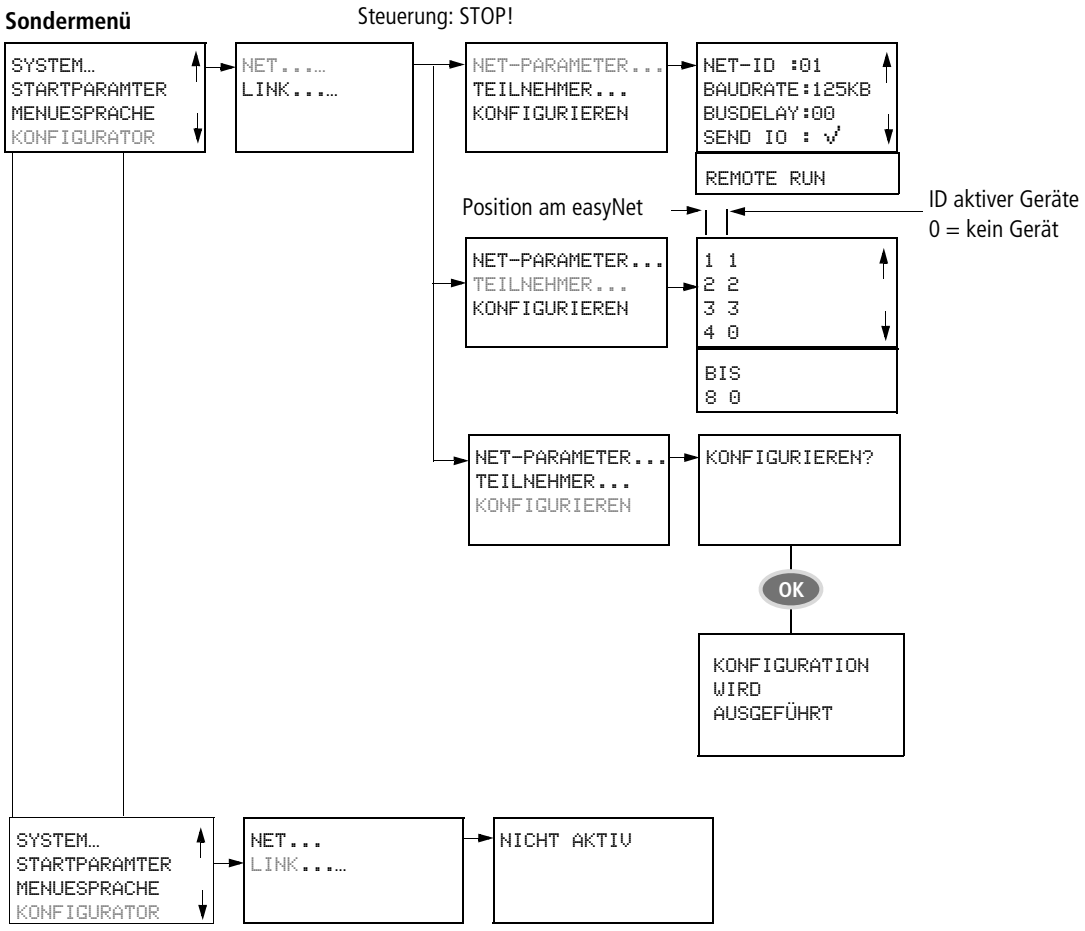


Sondermenü

▶ Mit gleichzeitigem Betätigen von **DEL** und **ALT** gelangen Sie ins Sondermenü.



Sondermenü


Sondermenü
Steuerung: STOP!
SYSTEM…
STARTPARAMTER
MENUESPRACHE
KONFIGURATOR
NET.....
LINK.....
NET-PARAMETER...
TEILNEHMER...
KONFIGURIEREN
NET-ID :01
BAUDRATE:125KB
BUSDELAY:00
SEND IO : √
REMOTE RUN
Position am easyNet
ID aktiver Geräte
0 = kein Gerät
NET-PARAMETER...
TEILNEHMER...
KONFIGURIEREN
1 1
2 2
3 3
4 0
BIS
8 0
NET-PARAMETER...
TEILNEHMER...
KONFIGURIEREN
KONFIGURIEREN?
OK
KONFIGURATION
WIRD
AUSGEFÜHRT
SYSTEM…
STARTPARAMTER
MENUESPRACHE
KONFIGURATOR
NET...
LINK.....
NICHT AKTIV

7 Beschreibung der Einstellungen

Alle Einstellungen nehmen Sie über die Bedienelemente der Steuerung vor.

Passwortschutz

Sie können den Zugang zum Hauptmenü inklusive Sondermenü, das Einstellen der Uhr und der Betriebsart (RUN/STOP) mit einem Passwort schützen. Über das Menü <Sicherheit → Bereich> aktivieren Sie die einzelnen Einstellmöglichkeiten.

Das Sondermenü wird bei einem aktivierten Passwort immer geschützt.

Als Passwort wird ein Wert zwischen 000001 und 999999 eingegeben. Mit der Zahlenkombination 000000 löschen Sie ein Passwort.

Passwort einrichten

Ein Passwort können Sie über das Sondermenü einrichten, unabhängig von der Betriebsart RUN oder STOP. Wenn bereits ein Passwort aktiviert ist, können Sie nicht ins Sondermenü wechseln.

- ▶ Rufen Sie mit **DEL** und **ALT** das Sondermenü auf.
- ▶ Starten Sie die Passworteingabe über den Menüpunkt SICHERHEIT…
- ▶ Betätigen Sie die Taste **OK** und wechseln Sie zu dem Menü PASSWORT…
- ▶ Mit einem weiteren Betätigen von **OK** sind Sie in der Passworteingabe.

Ist kein Passwort eingetragen, erscheinen sechs Striche: Kein Passwort vorhanden.

```
EINGABE PASSWORT
 ▮-----
```

- ▶ Betätigen Sie **OK**, sechs Nullen erscheinen
- ▶ Stellen Sie das Passwort mit den Cursortasten ein:
  – 〈 〉 Stelle im Passwort auswählen,
  – ∧∨ einen Wert zwischen 0 bis 9 einstellen.

```
EINGABE PASSWORT
  000042
```

- ▶ Speichern Sie das neue Passwort mit **OK**.

Mit **OK** verlassen Sie die Passwortanzeige und gehen mit **ESC** und ∨ auf das Menü BEREICH…

Der Gültigkeitsbereich des Passwortes wurde noch nicht betrachtet. Das Passwort ist gültig, aber noch nicht aktiviert.

Gültigkeitsbereich des Passwortes wählen

- ▶ Betätigen Sie die Taste **OK**.
- ▶ Wählen Sie die zu schützende Funktion oder das Menü aus.
- ▶ Betätigen Sie die Taste **OK** um die Funktion oder das Menü zu schützen (Haken = geschützt).

```
PROGRAMM √
UHR
BETRIEBSART
```

→ Der Standardschutz liegt auf dem Programm.

Mindestens eine Funktion oder ein Menü muss geschützt sein.

- PROGRAMM: Das Menü PROGRAMM ist geschützt.
- UHR: Datum und Uhrzeit sind mit dem Passwort geschützt.
- BETRIEBSART: Die Umstellung der Betriebsart RUN oder STOP ist geschützt.

Passwort aktivieren

Ein vorhandenes Passwort kann auf vier Wege aktiviert werden:

- Automatisch beim erneuten Einschalten der Steuerung
- Automatisch nach dem Laden des Programms
- Automatisch wenn auf der PC-Schnittstelle 30 Minuten nach der Passwort-Eingabe kein Telegramm gesendet wurde.
- Über das Passwortmenü.

- ▶ Rufen Sie mit **DEL** und **ALT** das Sondermenü auf.
- ▶ Öffnen Sie das Passwortmenü über den Menüpunkt SICHERHEIT…

das Passwortmenü wird nur angezeigt, wenn ein Passwort vorhanden ist.

```
WECHSELN PW
AKTIVIEREN
```

Der Standardschutz liegt auf dem Programm.

→ Bevor Sie Ihr Passwort aktivieren, notieren Sie sich das Passwort. Ist der Passworteintrag nicht mehr bekannt, können Sie nicht mehr in das Sondermenü wechseln.

- ▶ Wählen Sie AKTIVIEREN PW und betätigen Sie **OK**.

Das Passwort ist jetzt aktiv. Es wird zur Statusanzeige gewechselt.

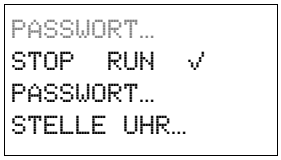
Bevor Sie nun eine geschützte Funktion, ein geschütztes Menü ausführen oder in das Sondermenü wechseln können, müssen Sie das Passwort eingeben.

Zugang bei Passwortschutz

Nach der Eingabe des Passwortes ist der Passwortschutz deaktiviert. Sie können den Passwortschutz später wieder über das Passwortmenü oder durch Aus- und Einschalten der Versorgungsspannung aktivieren.

▶ Wechseln Sie mit **OK** in das Hauptmenü.

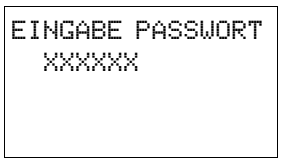
Der Eintrag PASSWORT… blinkt.



▶ Wechseln Sie mit **OK** zur Passworteingabe.

→ Wird im Hauptmenü PROGRAMM… statt PASSWORT… angezeigt, ist kein Passwortschutz aktiv.

Das Feld zur Passworteingabe wird eingeblendet.



▶ Stellen Sie das Passwort mit den Cursortasten ein.
▶ Bestätigen Sie mit **OK**.

Wenn das Passwort stimmt, erfolgt der Wechsel zurück zur Statusanzeige.

Der Menüpunkt PROGRAMM… ist freigegeben.

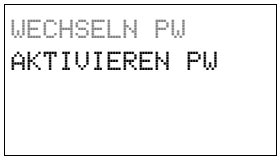
Das Sondermenü ist ebenso erreichbar.

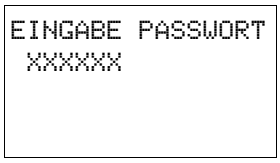
Passwort, Bereich ändern oder löschen

▶ Geben Sie Ihr Passwort ein.
▶ Rufen Sie mit **DEL** und **ALT** das Sondermenü auf.
▶ Öffnen Sie das Passwortmenü über den Menüpunkt SICHERHEIT und PASSWORT…

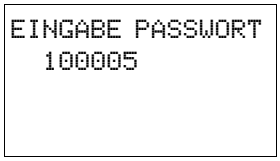
Der Eintrag WECHSELN PW blinkt.

Dieses Menü wird nur angezeigt, wenn ein Passwort vorhanden ist.





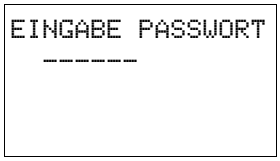
▶ Rufen Sie mit **OK** die Passworteingabe auf.
▶ Wechseln Sie mit **OK** auf das 6-stellige Eingabefeld.
▶ Das aktuelle Passwort wird angezeigt.
▶ Ändern Sie die sechs Passwortstellen mit den Cursortasten.
▶ Bestätigen Sie mit **OK**.



Mit **ESC** verlassen Sie den Sicherheitsbereich.

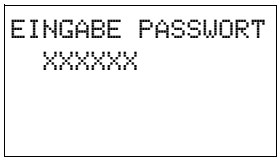
Löschen
Löschen Sie ein Passwort mit dem Wert „000000".

Ist kein Passwort eingetragen, werden sechs Striche angezeigt.



Passwort fehlerhaft eingegeben oder nicht mehr bekannt

Sie haben ein fehlerhaftes Passwort eingegeben?



▶ Geben Sie das Passwort erneut ein.

Dies kann beliebig oft erfolgen!

Wenn Sie ESC betätigen, gelangen Sie zurück ins Ausgangsmenü.

⚠ Sollten Sie das Passwort vergessen haben, können Sie nur den Browser-Befehl „factoryset" aufrufen.

Passwort, Anwenderprogramm und Bootprojekt werden dann gelöscht und die Steuerung mit den Defaultparametern initialisiert → Abschnitt „Reset", Seite 44.

Menüsprache ändern

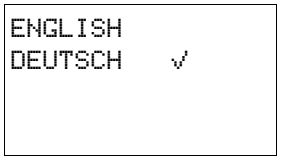
Es stehen zwei Menüsprachen zur Auswahl, die Sie über das Sondermenü einstellen können.

Sprache	Anzeige
Englisch	ENGLISH
Deutsch	DEUTSCH

→ Die Sprachauswahl steht nur zur Verfügung, wenn die Steuerung nicht durch ein Passwort gesichert ist.

- ▶ Rufen Sie mit **DEL** und **ALT** das Sondermenü auf.
- ▶ Wählen Sie MENUESPRACHE… zur Änderung der Menüsprache.

Die Sprachauswahl für den ersten Eintrag ENGLISH wird angezeigt.



- ▶ Wählen Sie mit ∧ oder ∨ die neue Menüsprache aus.
- ▶ Bestätigen Sie mit **OK**. Die „Sprache" erhält einen Haken.
- ▶ Verlassen Sie das Menü mit **ESC**.

Die neue Menüsprache wird aktiv.

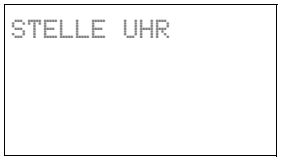
Mit **ESC** wechseln Sie zurück zur Statusanzeige.

Datum und Uhrzeit einstellen

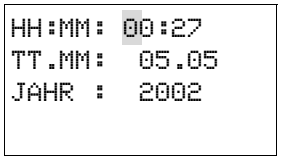
Die Geräte sind mit einer Echtzeituhr mit Datum und Uhrzeit ausgestattet. Stellen Sie bei der ersten Inbetriebnahme Stunde, Minute, Tag, Monat und Jahr ein.

- ▶ Wählen Sie im Hauptmenü STELLE UHR…

Das Menü zur Uhreinstellung wird eingeblendet.



- ▶ Wählen Sie STELLE UHR.



- ▶ Stellen Sie die Werte für Uhrzeit, Tag, Monat und Jahr ein.
- ▶ Betätigen Sie die Taste **OK**, um in den Eingabemodus zu gelangen.
  - – 〈 〉 Die Stelle wählen
  - – ∧∨ Den Wert ändern.
  - – **OK** Tag und Zeit speichern
  - – **ESC** Vorherige Einstellung beibehalten.

Mit **ESC** verlassen Sie die Anzeige der Uhreinstellung.

Anlaufverhalten

Anlauf- (Start-) verhalten einstellen

Über das Menü können Sie folgende Startmöglichkeiten einstellen:

- HALT
- WARMSTART
- KALTSTART.

- ▶ Wechseln Sie in das Sondermenü.

→ Ist die Steuerung durch ein Passwort geschützt, steht das Sondermenü nur nach der Eingabe des Passwortes zur Verfügung (→ Abschnitt „Zugang bei Passwortschutz",Seite 36.

- ▶ Stellen Sie das Anlaufverhalten ein.

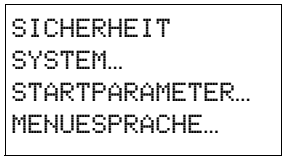
Kontrast und Hintergrundbeleuchtung LCD einstellen

Die Hintergrundbeleuchtung der LCD-Anzeige kann abgeschaltet werden. Der Kontrast der Anzeige kann in 5 Stufen eingestellt werden. Im Betrieb wird die Anzeige nicht benötigt. Nur im Wartungsfall oder falls Texte angezeigt werden müssen, wird die Hintergrundbeleuchtung benötigt.

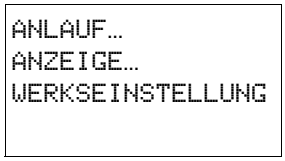
▶ Wechseln Sie in das Sondermenü.

→ Ist die Steuerung durch ein Passwort geschützt, steht das Sondermenü nur nach der Eingabe des Passwortes zur Verfügung (→ Abschnitt „Zugang bei Passwortschutz" Seite 36.

▶ Wählen Sie das Menü SYSTEM.
▶ Betätigen Sie die Taste **OK**.



▶ Wählen Sie mit der Cursortaste ∨ das Menü ANZEIGE und betätigen Sie **OK**.



Es werden die Menüs für die Kontrasteinstellung und die Hintergrundbeleuchtung angezeigt.

▶ Betätigen Sie die Taste **OK** und wechseln damit in die Eingabe des Kontrastes.

Mit den Cursortasten ∧ und ∨ verändern Sie den Kontrast zwischen dem Wert –2 bis +2.

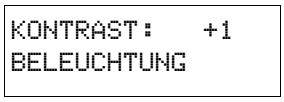
▶ Wählen Sie ihre Einstellung.

▶ Bestätigen Sie die Einstellung mit der Taste **OK**.

Die Kontrasteinstellung bleibt so lange erhalten, bis Sie geändert wird.

▶ Wechseln Sie mit den Cursortasten ∧ und ∨ auf das Menü BELEUCHTUNG.
▶ Betätigen Sie die Taste **OK**.

▶ Die Hintergrundbeleuchtung ist ausgeschaltet.



▶ Möchten Sie die Hintergrundbeleuchtung wieder einschalten, betätigen Sie die Taste **OK**.

▶ Der Haken √ signalisiert, dass die Hinergrundbeleuchtung eingeschaltet ist.

→ Die Grundeinstellung bei Auslieferung ist:
Der Kontrast steht auf der Einstellung 0.
Die Hintergrundbeleuchtung ist ständig eingeschaltet.
Einstellung des Menüs: BELEUCHTUNG √

8 Konfiguration der Ein-/Ausgänge (I/O)

Darstellung der Ein-/Ausgänge in der Konfiguration

In der Steuerungskonfiguration sind den direkten Adressen der Ein-/Ausgänge bereits die symbolischen Namen zugeordnet.

Symbolischer Operand	Physikalischer Operand	Datentyp
I1	AT % IX0.0	BOOL

Die symbolischen Operanden können Sie direkt im Programm verwenden.

Anzeige der lokalen Ein-/Ausgänge

- ▶ Zur Anzeige der lokalen Ein-/Ausgänge klicken Sie zuerst auf das Knotenzeichen vor „Configuration EC4P-200“, dann auf das Knotenzeichen vor „Local I/O“.
- ▶ Unter dem Ordner „Local I/O“ werden auch die folgenden Ordner angezeigt, deren Funktion dem realen Steuerungstyp anzupassen ist.

  – Transistor Outputs
  – No Analog Outputs
  – No Keys
  – No Counter.

Bei einer Steuerung mit Relais-Ausgängen ist die Ordnerfunktion „Transistors Outputs“ in „Relais Outputs“ zu ändern,
→ Abschnitt „Ordnerfunktion ändern“.

- ▶ Zur Anzeige der Ein-/Ausgänge der einzelnen Ordner klicken Sie mit der linken Maustaste auf das Knotenzeichen vor einem Ordner.

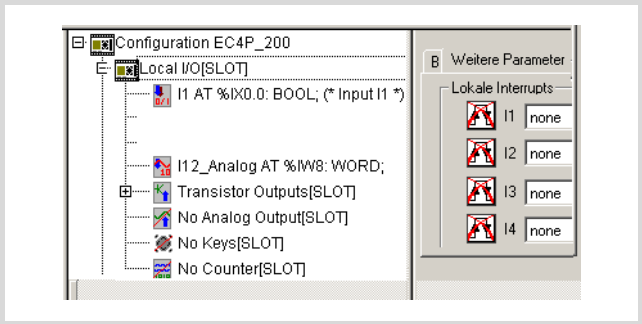


Abbildung 35: Wählbare Ein-/Ausgänge

Ordnerfunktion ändern

Transistor Outputs ↔ Relais Outputs

Die Transistor Outputs werden als Default-Konfiguration einer Steuerung angezeigt. Setzen Sie eine Steuerung mit Relais ein, müssen Sie den „Output-Typ“ ändern:

- ▶ Klicken Sie mit der rechten Maustaste auf den Schriftzug „Transistor Outputs“.
- ▶ Ziehen den Cursor auf den Menüpunkt „Elemente ersetzen“ und klicken „Relais Outputs“ an.

Generell gilt: Zur Anzeige der direkten und symbolischen Adressen der Ausgänge klicken Sie auf den Knoten „xxx Output“.

No Analog Output

Die Default-Konfiguration „No Analog Output“ kann ab Betriebssystem V2.0 durch „Analog Output“ ersetzt werden.

No Keys

Der Begriff „Keys“ steht für die Tasten auf der Frontseite der Steuerung wie ALT, DEL, ESC und OK sowie die 4 Tasten der Cursorwippe. Die Tasten werden in der Steuerungskonfiguration als Eingänge dargestellt.

Möchten Sie die Tastenzustände im Programm abfragen, können Sie die direkten oder die symbolischen Adressen der Eingänge programmieren.

- ▶ Klicken Sie mit der rechten Maustaste auf den Schriftzug „NoKeys“.
- ▶ Ziehen den Cursor auf den Menüpunkt „Elemente ersetzen“ und klicken „Keys“ an.
- ▶ Klicken Sie auf den Knoten vor „Keys“.

No Counter

Benötigen Sie für Ihre Applikation schnelle Zähler, müssen Sie diese aktivieren:

- ▶ Klicken Sie mit der rechten Maustaste auf den Schriftzug „No Counter“.
- ▶ Ziehen den Cursor auf den Menüpunkt „Elemente ersetzen“ und klicken Sie eine von 3 Zähler-Funktionen an.

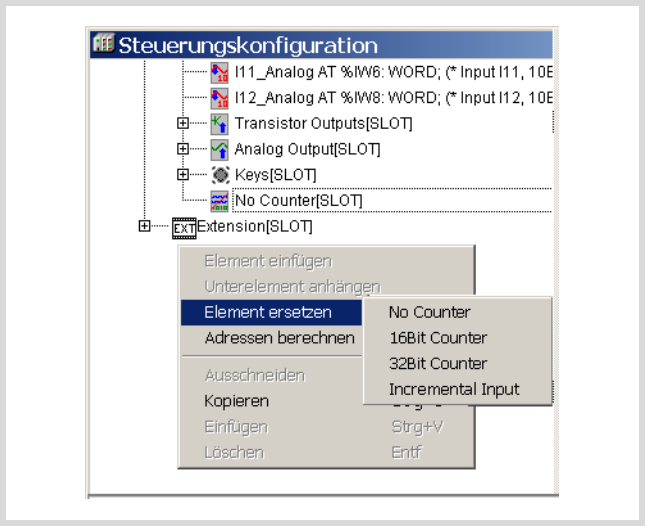


Abbildung 36: Counter auswählen

Das Untermenü erscheint:

- ▶ Wählen Sie eine Zählerart aus, z. B.: 32 Bit Counter.
- ▶ Die Bezeichnung „No Counter“ wird durch „32 Bit Counter“ ersetzt.
- ▶ Ein weiterer Klick auf das Auflösungszeichen und die Ein-/Ausgänge des Zählers werden angezeigt.

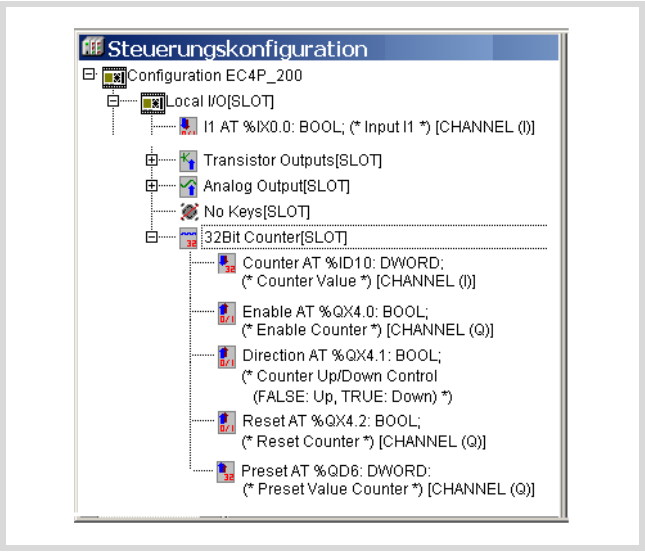


Abbildung 37: Counter (32 Bit) konfigurieren

Anzeige der Ein-/Ausgänge der Erweiterungsgeräte

- ▶ Klicken Sie auf das Auflösungszeichen vor dem Ordner „Extension“.
- ▶ Klicken Sie mit der rechten Maustaste auf den Ordner „No Extension“.
- ▶ Wählen Sie aus dem Menü „Element ersetzen“ ein Gerät aus.

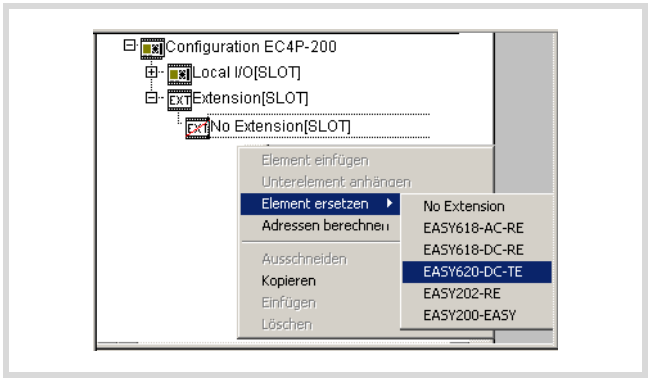


Abbildung 38: Erweiterungsgerät auswählen

- ▶ Klicken Sie auf das Auflösungszeichen vor dem neuen Geräteordner, damit Ihnen die Ein- und Ausgänge inklusive der Diagnoseeingänge angezeigt werden.

9 Betrieb

Allgemeines

Übersicht über die Speichergrößen

Folgende maximale Speicher/Programmbausteine stehen zur Verfügung:

Programm (Code)	256 kByte
Globale Variablen (Global)	224 kByte
Datenspeicher (Memory)	16 kByte
Eingangsabbild (Input)	4 kByte
Ausgangsabbild (Output)	4 kByte
Remanente Variablen (Retain)	8 kByte
Max. Anzahl von Bausteinen (POUs)	ca. 2000

→ Wird in einem Funktionsbaustein (FB) eine Variable als RETAIN deklariert, nehmen alle Variablen in diesem Baustein den Status RETAIN an.

Speicher-Definition

Die Steuerung hat folgende Speicher:

- Arbeitsspeicher (SRAM), nicht gepuffert.
  – Inhalt, z. B.: Programm, Daten
- Systemspeicher (FLASH), gepuffert.
  – Inhalt, z. B.: Bootprojekt
- Speicherkarte
  – Inhalt, z. B.: Bootprojekt, Betriebssystem.

Einschaltverhalten

Die Steuerung hat keine Batterie zum Backup des Arbeitsspeichers, der das Programm enthält. Zur Sicherung des Programms bei Spannungsausfall sollten Sie ein Bootprojekt von diesem Programm erzeugen, das im spannungsausfallsicheren Systemspeicher abgelegt wird.

Nach dem Einschalten der Versorgungsspannung führt die CPU einen System-Selbsttest durch. Im Fehlerfall blinken die LEDs RUN/STOP/SF und CAN/NET rot. Nach dem fehlerfreien Selbsttest überprüft die Steuerung, ob:

- ein Betriebssystem-Update auf einer gesteckten Speicherkarte vorhanden ist. Gegebenfalls muss es geladen werden.
- ein Bootprojekt verfügbar ist. Ist das der Fall, wird es in den Arbeitsspeicher der Steuerung geladen und in Abhängigkeit des parametrierten Startverhaltens gestartet. Ist kein Bootprojekt vorhanden, bleibt die Steuerung im Zustand NOT READY.

Einschaltverhalten mit Bootprojekt auf der Speicherkarte

Beim Einschalten der Steuerung hat ein auf der Speicherkarte vorhandenes Bootprojekt Vorrang vor einem im Systemspeicher gespeicherten. Sind die Bootprojekte unterschiedlich, wird das Bootprojekt von der Speicherkarte in den Systemspeicher kopiert und anschließend zur Ausführung gebracht. Durch den Kopiervorgang verlängert sich die Hochlaufphase der Steuerung einmalig um mehrere Sekunden.

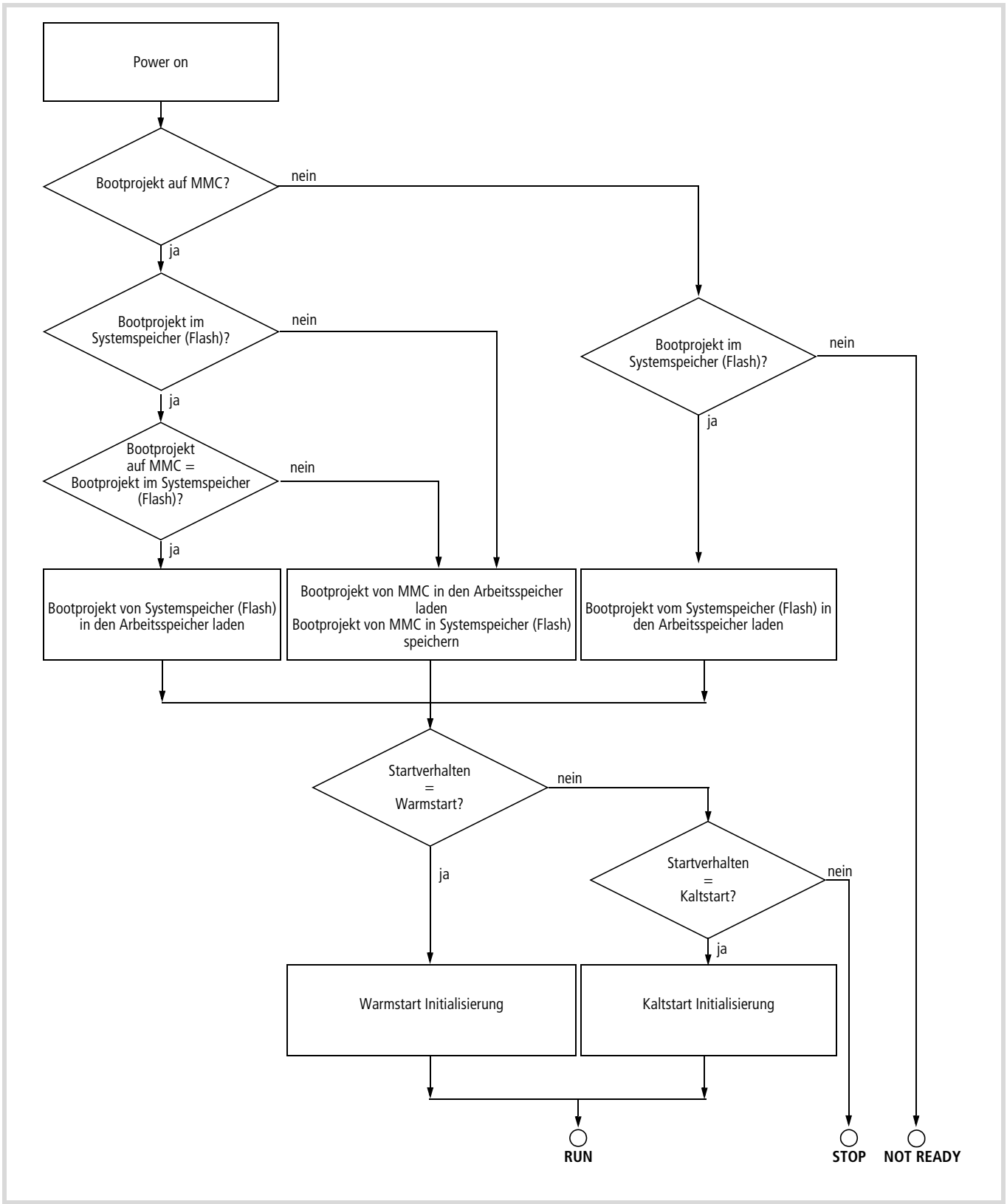


Abbildung 39: Einschaltverhalten mit Bootprojekt

Startverhalten in der Programmiersoftware einstellen

Mit der Einstellung des Startverhaltens bestimmen Sie das Anlaufverhalten der Steuerung beim Einschalten der Versorgungsspannung.

Die Einstellung können Sie im Steuerungskonfigurator oder über die Bedienelemente der Steuerung vornehmen. Die Einstellmöglichkeiten sind nicht priorisiert. Die letzte Eingabe ist gültig.

Aktivieren Sie im Steuerungskonfigurator das Register „Allgemeine Parameter" und wählen Sie aus dem Listenfeld die die gewünschte Startbedingung aus.

- HALT
- WARMSTART
- COLDSTART.

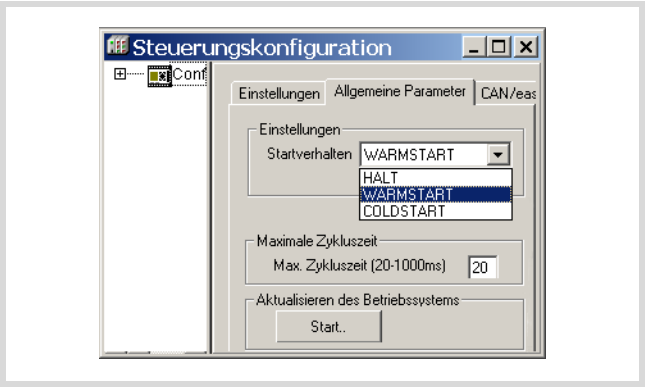


Abbildung 40: Startverhalten festlegen

Programm START/STOP

Programmstart (STOP → RUN)

Sie haben zwei Möglichkeiten, das Programm zu starten:

- Im Online-Betrieb den Befehl START anwählen; z. B. nach dem Laden eines Programms.
- Über die Bedienelemente der Steuerung.
  – Wählen Sie im Hauptmenü im Menü Programm „START" an.

Verhalten nach dem Ausschalten/Unterbrechen der Spannungsversorgung

Wenn Sie die Spannung ausschalten oder unterbrechen, wird der Programmzyklus sofort unterbrochen. Das Programm wird bis zum Zyklusende nicht mehr abgearbeitet. Dies erfolgt auch nicht bei Spannungswiederkehr. Dann beginnt die Programmbearbeitung am Anfang des Programms. Dies kann dazu führen, dass remanente Daten, z. B. Variablen im Doppelwortformat, je nach Programm-Abbruchstelle, nicht mehr konsistent sind!

Sind inkonsistente Daten für eine Anwendung nicht praktikabel, können Sie z. B. eine unterbrechungsfreie Stromversorgung mit Akku-Pufferung einsetzen.

Bei Spannungsausfall werden alle Ausgänge auf 0 gesetzt bzw. abgeschaltet.

Das Verhalten remanenter Variablen nach eingestellter Startbedingung zeigt Tabelle 9.

Der Start der Steuerung erfolgt gemäß den Einstellungen im Steuerungskonfigurator → Abbildung 40.

Tabelle 9: Verhalten der Variablen beim Start

Startbedingungen	**Art der Variablen**	
	Nicht remanent	**Remanent (Retain)**
COLDSTART	Aktivieren der Initialwerte	
WARMSTART	Aktivieren der Initialwerte	Werte bleiben erhalten
Programm im Online-Betrieb laden und starten	Aktivieren der Initialwerte	
Start/Stop/Start…	Werte bleiben erhalten	

Programmstopp (RUN → STOP)

Sie haben zwei Möglichkeiten, das Programm zu stoppen:

- Im Online-Betrieb den Befehl STOP wählen.
- Über das Menü der Steuerung.
  – Wählen Sie im Hauptmenü im Menü Programm STOP an

Wenn Sie den STOP-Befehl ausführen, wechselt die CPU in den Zustand STOP, sobald der Programmzyklus beendet ist. Die Ausgänge werden auf 0 gesetzt.

Programm über externen Schalter starten/stoppen

Mit Hilfe eines externen Schalters, der auf einen Eingang geführt wird, können Sie die Programmbearbeitung starten oder stoppen. Es sind einige zusätzliche Programmanweisungen erforderlich, die Sie aus dem Beispiel im Anhang entnehmen können (→ Seite 94). Für den Start benötigen Sie die Funktion SysStartPlcProgram, für den Stop die Funktion SysStopPlcProgram aus der Bibliothek SysLibPlcCtrl.lib.

Für diesen Fall ist das Startverhalten der Steuerung im Steuerungskonfigurator unter <Weitere Parameter → Einstellungen> auf WARMSTART zu setzen!

Nach wie vor können Sie die Steuerung auch über den PC im Online-Betrieb in den Zustand START bzw. STOP setzen.

Programmbearbeitung und Systemzeit

Das Anwenderprogramm wird zyklisch abgearbeitet. Vor jedem Programmzyklus wird das Abbild der Eingänge eingelesen und am Ende des Zyklus wird das Ausgangsabbild auf die physikalischen Ausgänge geschrieben.

Die Zykluszeit hängt von der Länge und der Struktur des Anwenderprogramms ab. Um schnell auf Ereignisse zu reagieren, gibt es die Möglichkeit, Programm-Routinen zu programmieren, die beim Auftreten von Systemereignissen gestartet werden, → Abschnitt „Systemereignisse" auf Seite 49.

Zykluszeit überwachen

Die Zykluszeit des Anwenderprogramms wird überwacht. Überschreitet die Zykluszeit die eingestellte Zeit, wird die Steuerung in den Zustand STOP gesetzt und die Ausgänge werden abgeschaltet.

Die max. zulässige Zeit stellen Sie im Steuerungskonfigurator unter „Weitere Parameter" ein. Der kleinste Zeitwert beträgt 20 ms (Defaultwert), der größte 1 000 ms.

Reset

Einen Reset können Sie über den PC im Online-Betrieb oder über das Menü der Steuerung durchführen. Wählen Sie dazu im Steuerungskonfigurator bzw. im Menü der Steuerung den Menüpunkt an.

Es gibt folgende Reset-Befehle im Menü:

Konfigurator (Online-Menü)	Steuerungsmenü
Reset warm	Reset warm
Reset kalt	Reset kalt
Reset Ursprung	LÖSCHEN-> RESET URSPRUNG

Reset warm

- Das Programm wird angehalten.
- Die nicht remanenten Variablen werden initialisiert, die „Retain"-Variablen bleiben erhalten.
- Das Programm kann wieder gestartet werden.

Reset kalt

- Das Programm wird angehalten.
- Alle Variablen werden initialisiert.
- Das Programm kann wieder gestartet werden.

Reset Ursprung

- Das Programm im Arbeitsspeicher und das Bootprojekt im Systemspeicher der Steuerung werden gelöscht.
- Bei gesteckter Speicherkarte:
  - Alle auf der Speicherkarte projektspezifischen Dateien, das Betriebssystem und das Bootprojekt werden gelöscht
  - Alle benutzerspezifischen Dateien sowie die Startup.ini-Datei bleiben unverändert
- Die Steuerung wird in den Zustand NOT READY gesetzt.

Wiederherstellen der Werkseinstellungen (factoryset)

Mit Hilfe des Browerbefehls „factoryset" oder des Menüpunkts der Steuerung <SYSTEM → WERKSEINSTELLUNG> wird ein „Reset Ursprung" durchgeführt (→ Abschnitt „Reset Ursprung"). Zusätzlich werden die Startup.ini-Datei auf der Speicherkarte und die System-Parameter in der Steuerung gelöscht. Nach einem Start abeitet die Steuerung wieder mit den STARTUP-Daten. Die Schnittstellen werden mit ihren Defaultwerten initialisiert.

Verhalten der Variablen nach Reset

Reset	Art der Variablen	
	Nicht remanent	Retain
Reset warm	Aktivieren der Initialwerte	Werte bleiben erhalten
Reset kalt	Aktivieren der Initialwerte	
Reset Ursprung[1)]	Keine Variablen mehr vorhanden, das Programm wird gelöscht	

1) Nach Reset-Ursprung muss das Programm neu geladen werden. Im Online-Betrieb können Sie danach die Steuerung starten.

Test und Inbetriebnahme

Die Steuerung unterstützt folgende Test- und Inbetriebnahme-Funktionen:

- Breakpoint/Einzelschritt-Betrieb
- Einzelzyklus-Betrieb
- Forcen
- Online-Änderung
- Ablaufanzeige (Power Flow).

→ Für Breakpoint/Einzelschritt-Betrieb und Einzelzyklus-Betrieb gilt:

Wenden Sie diese Inbetriebnahme-Funktionen nicht in den Programm-Routinen, z. B. für Start, an. Ein Fehlverhalten kann zu einem undefinierten Steuerungszustand führen.

Lassen sich die Inbetriebnahme-Funktionen nicht ausführen, aktivieren Sie die Debugging-Funktion (Defaultzustand): Klicken Sie im Menü <Projekt → Optionen → Übersetzungsoptionen> das Feld „Debugging" an.

Breakpoint/Einzelschritt-Betrieb

Innerhalb des Anwenderprogramms können Breakpoints gesetzt werden. Bei der Ausführung einer mit einem Breakpoint versehenen Anweisung wird das Programm an dieser Stelle angehalten. Das Programm kann nun im Einzelschritt-Betrieb ausgeführt werden. Die Zykluszeitüberwachung ist deaktiviert.

⚠ **Vorsicht!**
Die zu diesem Zeitpunkt gesetzten Ausgänge bleiben gesetzt!

Einzelzyklus-Betrieb

Im Einzelzyklus-Betrieb wird ein einzelner Programmzyklus in Echtzeit ausgeführt. Die Ausgänge sind während des Zyklus freigegeben. Die Zykluszeit-Überwachung ist aktiv.

⚠ **Vorsicht!**
Die zu diesem Zeitpunkt gesetzten Ausgänge bleiben gesetzt!

Zwangssetzen von Variablen und Ein-/Ausgänge (Forcen)

Alle Variablen eines Anwenderprogramms können mittels „Forcen" zwangsweise auf feste Werte gesetzt werden. Zwangsgesetzte lokale Ausgänge werden nur im Zustand RUN zur Peripherie durchgeschaltet.

→ Ein Zwangssetzen des über den CANopen-Feldbus angeschlossenen Ein-/Ausgänge ist nicht möglich.

Zustandsanzeige in der Programmiersoftware

- Der Signalzustand der physikalischen, booleschen Eingänge wird sowohl im Zustand RUN der CPU als auch im STOP angezeigt.
- Der Signalzustand der physikalischen, booleschen Ausgänge wird nur im RUN-Zustand angezeigt; im STOP-Zustand sind sie mit FALSE gekennzeichnet.
- Alle anderen Variablen werden mit dem aktuellen Variablenwert angezeigt.

Schnelle Zähler (Counter)

Für jede Zähler-Funktion ist im Abschnitt „Impuls-/Inkrementalgeber anschließen" (Seite 23) der Steuerungseingang zur Impulsverarbeitung aufgeführt. Die weiteren Ein-/Ausgänge eines Zählers werden in der Steuerungskonfiguration mit Symbolen – wie z. B. Reset – dargestellt, nachdem Sie die Zählerart – z. B. 32BitCounter – ausgewählt haben (→ Kapitel „Konfiguration der Ein-/Ausgänge (I/O)", Seite 39). Die symbolischen Ein-/Ausgänge parametrieren Sie im Programm (Arbeitsweise mit Symbolen).

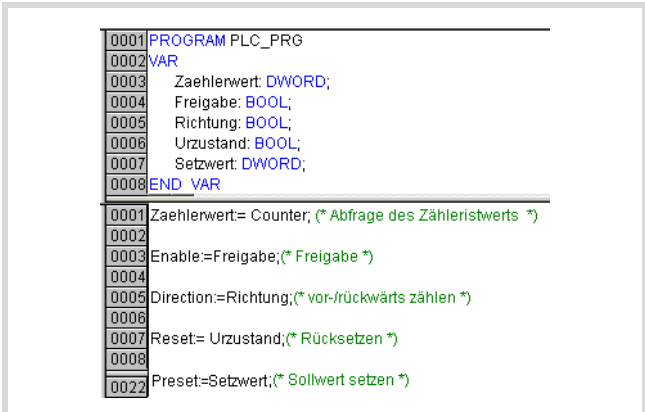


Abbildung 41: Ein-/Ausgänge des schnellen 32-Bit-Zählers programmieren

Dabei ist zu berücksichtigen, dass die Operationen wie z.B. Reset, über das Programmabbild geführt werden. Der Ausgang „Reset AT%QX1.2" wird erst am Ende des Programms aktiv.

Sie können auch schnellere Zugriffszeiten unter Umgehung des Abbildes erreichen, wie es z. B. in einer Interrupt-Routine erforderlich sein kann. Setzen Sie in diesem Fall Funktionsbausteine ein – alternativ zur Arbeitsweise mit Symbolen. In der Bibliothek EC_Util2.lib gibt es für jeden Zählertyp einen Funktionsbaustein. Der Funktionsbaustein für den 32 Bit-Zähler hat folgende Parameter:

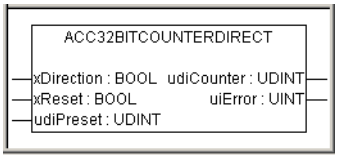


Abbildung 42: Funktionsbaustein 32 Bit Counter

Die Ein-/Ausgänge der Funktionsbausteine entsprechen im Wesentlichen den Ein-/Ausgängen, die in der Steuerungskonfiguration aufgeführt sind.

→ Sowohl für den 16 Bit- als auch für den 32-Bit-Counter müssen Sie zusätzlich zu den Eingängen des Funktionsbausteins auch den „Enable"-Eingang aus der Steuerungskonfiguration zur Zählerfreigabe programmieren.

Counter-Funktionen (Ein-/Ausgänge)

Die Beschreibung der Funktionen der Ein-/Ausgänge in den folgenden Abschnitten gilt für die Ein-/Ausgänge der Funktionsbausteine und die der Steuerungskonfiguration.

32 Bit Counter

Die Funktion des 32-Bit-Counter steht einmal zur Verfügung. Der Impulsgeber muss mit dem externen Eingang I1 verbunden werden. Er nimmt die Impulse mit einer max. Frequenz von 50 kHz auf. Die CPU zählt diese Impulse und stellt sie als Istwert (= Counter) zur Verfügung. Den Istwert können Sie im Anwenderprogramm abfragen. Ob der Istwert bei Eintreffen eines Zählimpulses inkrementiert oder dekrementiert wird, entscheiden Sie im Anwenderprogramm durch das Ansteuern des Ausgangs „Richtung" (Direction).

→ Wenn der Istwert gleich dem Sollwert ist, kann ein Interrupt erzeugt werden. Dieser bewirkt, dass eine Programm-Routine abgearbeitet wird.
Dazu müssen Sie den Interrupt in der Taskkonfiguration aktivieren und die Programm-Routine zuordnen
→ Abschnitt „Interruptverarbeitung", Seite 52.

Über das Programm können Sie folgenden Eigenschaften des Zählers festlegen:

- Freigabe (Enable):
  - TRUE: Impulse werden gezählt.
  - FALSE: Impulse werden nicht gezählt.
  - Durch ein „1"-Signal am „Enable"-Eingang wird der Zähler freigegeben: Die eingehenden Impulse werden gezählt. Mit der 0 → 1-Flanke des Enable-Signals wird der Istwert auf 0 gesetzt und der Zustand am Eingang „Richtung" sowie der am Eingang „Sollwert bereitstellen" (Preset) übernommen. Eine Änderung der Richtung wird während des Betriebs nicht erkannt.
- die Richtung (Direction)
  - Inkrementieren (Direction = FALSE): Der Zähler zählt bis zum vorgegebenen Sollwert (PRESET). Wird der Sollwert erreicht, wird der parametrierte Interrupt aktiviert, der in eine Programm-Routine verzweigt (→ Seite 52).
    Beim nächsten Zählimpuls beginnt der Zähler von 0.
  - Dekrementieren (Direction = TRUE): Mit dem erstenZählimpuls wird der Istwert von 0 auf den Sollwert gesetzt. Ist ein Interrupt parametriert, wird die damit verknüpfte Programm-Routine aufgerufen (→ Seite 52).
    Mit jedem weiteren Impuls wird der Istwert dekrementiert, bis er 0 erreicht. Beim nächsten Zählimpuls wird der Sollwert wieder übernommen und die Programm-Routine erneut aufgerufen.
- zurücksetzen (Reset):
  - Durch eine 0 → 1-Flanke am „Reset"-Eingang wird – unabhängig vom Zustand des Freigabesignals – der Istwert auf 0 gesetzt und Richtung und Sollwert übernommen.
- Sollwert bereitstellen (Preset) .

Beispiel: Programm mit FB für 32 Bit Counter

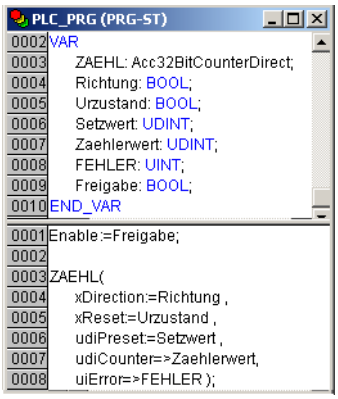


Abbildung 43: Programm mit FB für 32 Bit Counter

16 Bit Counter

Die Funktion steht zweimal zur Verfügung. Sie entspricht der Funktion des schnellen Zählers (32 Bit). Zur Unterscheidung der beiden 16-Bit-Zähler haben die symbolischen Operanden eine Nummer: 0 oder 1. Die Operanden mit der Nummer 0 steuern die Zählimpulse, die am Eingang I1 anliegen, die mit der Nummer 1 erfassen die Zählimpulse von I2.

Externe Eingänge:

Zählernummer	Impulseingang
0	I1
1	I2

Die Zählernummer ist aus den symbolischen Operanden aus dem Steuerungskonfigurator, im Ordner „16Bit Counter“ ersichtlich.

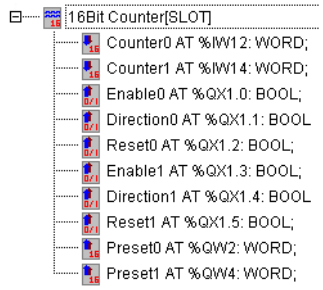


Abbildung 44: Ein-/Ausgänge der 16 Bit Counter 0 und 1

→ Wenn der Istwert gleich dem Sollwert ist, kann ein Interrupt erzeugt werden, der bewirkt, dass eine Programm-Routine abgearbeitet wird. Dazu müssen Sie den Interrupt in der Taskkonfiguration aktivieren und die Programm-Routine zuordnen → Abschnitt „Interruptverarbeitung“, Seite 52.

Inkrementalwertzähler (Incremental Input)

Die Funktion des Inkrementalwertzählers steht einmal zur Verfügung. Auf die externen Eingänge I1 und I2 werden die Inkrementalsignale A und B des Gebers geführt, auf den Eingang I3 das Referenzsignal, das der Geber einmal pro Umdrehung erzeugt. An den Eingang I4 wird der Referenzschalter angeschlossen, der im geschlossenen Zustand das Referenzfenster bildet, in dem das Referenzsignal verarbeitet wird.

Die Inkrementalsignale A und B sind zur Ermittlung der Zählrichtung um 90 Grad phasenverschoben. Es werden die steigenden und fallenden Flanken ausgewertet (4fach-Auswertung).
Die maximale Eingangsfrequenz beträgt 40 kHz. Damit ergibt sich eine Gesamtfrequenz von 160 kHz. Der Zähler erzeugt keinen Interrupt.

Mit den folgenden Signalen können Sie den Zähler steuern und an die Anwendung anpassen. Im Programm fragen Sie die Signal-Eingänge ab und setzen die Signal-Ausgänge. Die Signal-Bezeichnung geht aus der Steuerungskonfiguration hervor.

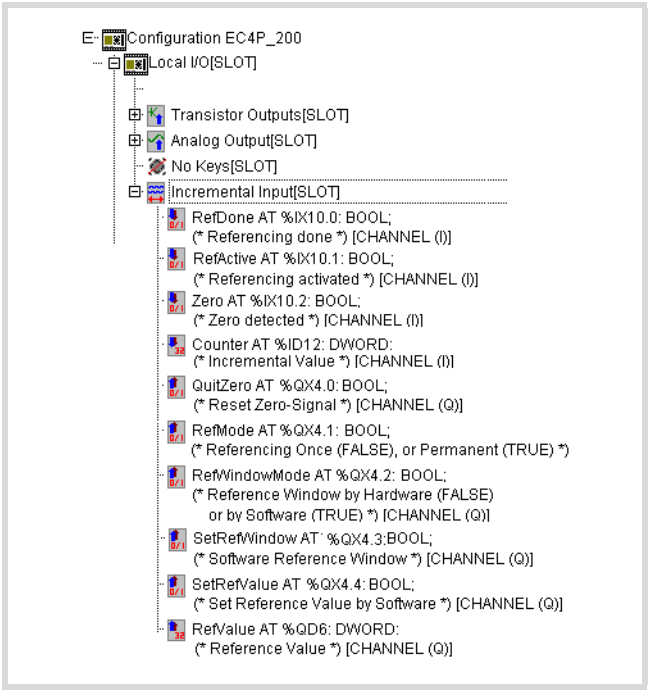


Abbildung 45: Ein-/Ausgangssignale des Inkrementalwertzählers

Erklärung der Ein-/Ausgangssignale (I/Q)

Signal	I/Q	Erklärung
RefDone	I	Referenzieren ausgeführt (Rückmeldung von SetRefWindow)
RefActive	I	Referenzieren aktiviert (Rückmeldung von SetRefWindow oder I4)
Zero	I	Nulldurchgang des Istwertes
Counter	I	Istwert des Zählers
QuitZero	Q	Quittierung des ZERO-Signals
RefMode	Q	Anzahl der Referenzierungen 0 = einmalig 1 = permanent
RefWindowMode	Q	Aktivierung des Referenzfensters durch 0 = externer Eingang I4 1 = im Programm mit „SetRefWindow“
SetRefWindow	Q	Aktivieren des Referenzfensters, wenn „RefWindowMode“ = 1
SetRefValue	Q	Referenzwert überschreibt Istwert (Reset)
RefValue	Q	Referenzwert

Signalübersicht der Ein-/Ausgänge (I/Q)

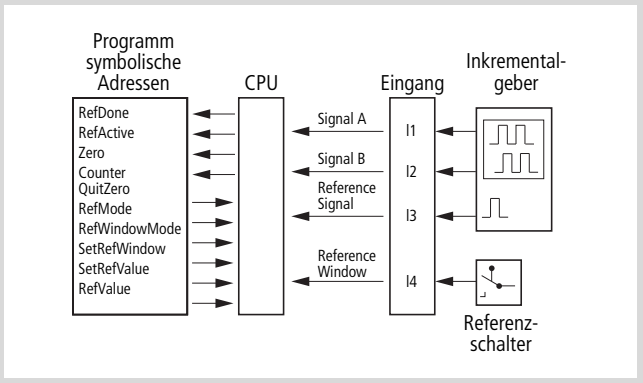


Abbildung 46: Ein-/Ausgänge des Inkrementalwertzählers

Funktionen der Ein-/Ausgangssignale

Mit dem HALT → RUN-Zustandswechsel der CPU wird der Zähler freigegeben: Die eingehenden Impulse werden gezählt.

SetRefValue (Reset)
Mit einer 0 → 1-Flanke am Eingang wird der Istwert mit dem am Eingang „RefValue" anstehenden Wert überschrieben.

Counter (Istwert)
Der Istwert des Zählers steht am Eingang „Counter" an.

Zero (Nulldurchgang)
Wenn der Istwert den Wert 0 erreicht, wird der Ausgang Zero gesetzt. Er bleibt gesetzt, bis er durch eine 0 → 1-Flanke am Eingang „QuitZero" quittiert wird.

RefWindowMode (Referenzfenster aktivieren)
Mit diesem Signal entscheiden Sie, ob das Signal zum Setzen des Referenzfensters über den Eingang I4 oder über das Anwenderprogramm mit dem Signal „SetRefWindow" erfolgt.

RefMode (Art der Referenzierung)
Das Signal am Eingang entscheidet, ob einmalig (0 am Eingang) oder permanent (1 am Eingang) referenziert wird. Der Istwert wird mit dem Referenzwert überschrieben, wenn das Referenzfenster gesetzt ist und am Eingang I3 ein Referenzimpuls erfolgt. Je nach Vorgabe des RefMode-Signals erfolgt dies einmalig (wenn die Bedingungen nach dem Start der Steuerung zutreffen) oder permanent (bei jedem Referenz-Impuls im Referenzfenster).

Referenzierung

In vielen Positioniersteuerungen wird zum Start der Positionierung ein Referenzpunkt angefahren. Dazu fährt z. B. ein Werkzeugschlitten in seine Grundposition. In dieser Position wird mechanisch ein „Referenzschalter" geschlossen, dessen Signal auf den Eingang I4 geführt wird. Das Schaltersignal kann auch durch das Signal „SetRefWindow" ersetzt werden, das der Anwender im Programm ansteuern kann. Als Rückmeldung wird das Signal „RefActive" gesetzt. Zur exakten Angabe der Schlittenposition erzeugt ein mit dem Schlitten verbundener Inkrementalgeber einen Referenzimpuls. Dieser wird am Eingang I3 erkannt, wenn der Referenzschalter geschlossen bzw. das Referenzfenster geöffnet ist. Der Referenzimpuls löst das Überschreiben des Zählers mit dem Referenzwert aus, den Sie in der Steuerungskonfiguration angegeben haben. „RefActive" wird zurückgesetzt und „RefDone" gesetzt, bis das Referenzfenster erneut geöffnet wird.

→ Wählen Sie das Referenzfenster so groß, dass das Referenzsignal nur einmal ansteht, aber dennoch zuverlässig ausgewertet werden kann.

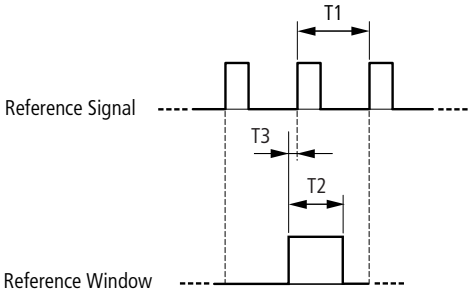


Abbildung 47: Beziehung zwischen Referenzsignal und Referenzfenster

T1 Impuls-Wiederholzeit zweier aufeinander folgender Referenzimpulse bei einer Umdrehung des Inkremental-Drehgebers

T2 Maximal zulässige Zeitdauer des Referenzfensters. Muss hinreichend kleiner sein als T1, sodass kein zweiter Referenzimpuls erkannt wird.

T3 Muss so lange sein, dass die L/H-Flanke des Referenzimpulses sicher erkannt werden kann.

T2 und T3 sind abhängig von der Impulsfolgefrequenz des Referenzimpulses und je nach Anwendung eventuell experimentell zu ermitteln.

Systemereignisse

Systemereignisse sind:

START	Start des Anwenderprogramms (Kalt- und Warmstart)
COLDSTART	Kaltstart des Anwenderprogramms
WARMSTART	Warmstart des Anwenderprogramms
STOP	Stopp des Anwenderprogramms (gilt nicht bei Zykluszeitüberschreitung oder Hardware-Watchdog)
IO-Interrupt 1, 2, 3, 4	Spannungswechsel an den Eingängen I1, I2, I3, I4
Counter-Interrupt1	Ist = Sollwert, bei 16 Bit Counter 0
Counter-Interrupt2	Ist = Sollwert, bei 16 Bit Counter 1 oder 32 Bit Counter
TIMER-INTERRUPT	Ein vom Anwender parametrierter Timer löst einen Interrupt aus.

Auf Systemereignisse der Steuerung können Sie reagieren, indem Sie eine Programm-Routine (POU) erstellen, die bei Auftreten des Ereignisses einmal ausgeführt wird. Die Ausführung wird zeitlich überwacht. Als Zeitbasis dient der für die maximal zulässige Zykluszeit parametrierte Wert.

START, COLDSTART, WARMSTART, STOP

Bei Auftreten des Ereignisses, z. B. Warmstart der Steuerung, wird ein Interrupt erzeugt (→ Seite 52), der die ihm zugeordnete Programm-Routine aufruft. Die Zuordnung nehmen Sie in der Taskkonfiguration vor.

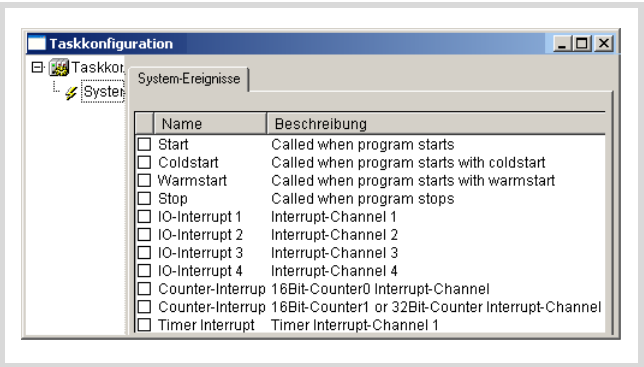


Abbildung 48: Systemereignisse

Interrupt-Eingänge I1 bis I4

Die Eingänge I1 bis I4 können als Interrupt-Eingänge konfiguriert werden. Eine Flanke am Eingang erzeugt einen Interrupt (→ Seite 52), der die zugeordnete Programm-Routine aufruft.

- ▶ Legen Sie zunächst die Art der Flanke des Eingangssignals im Steuerungskonfigurator fest.
- ▶ Ordnen Sie in der Taskkonfiguration dem Eingang die Programm-Routine zu.

Die Eingänge sind in Gruppen priorisiert. I1 und I2 haben eine hohe Priorität, I3 und I4 haben eine niedrige Priorität.

Nieder priore Interrupts können von höher prioren unterbrochen werden.

Counter-Interrupt

Verwenden Sie die Funktion „Schneller Zähler", vergleicht die Steuerung ständig den Istwert mit dem Sollwert des Zählers. Sind beide identisch, wird ein Interrupt erzeugt (→ Seite 52), der die von Ihnen erstellte Programm-Routine (POU) aufruft.

Dazu müssen Sie zunächst die Art des Zählers im Steuerungskonfigurator festlegen. Dann erfolgt die Zuordnung zwischen dem Eingang, der die Zählimpulse empfängt, und der POU, die Sie in der Taskkonfiguration vornehmen.

Timer-Interrupt

Sie können eine Programm-Routine erstellen, die in einem festen Intervall aufgerufen wird. Gestartet wird die Funktion TIMERINTERRUPTENABLE mit Hilfe einer booleschen Variablen oder eines externen Eingangs. Die Zuordnung der Programm-Routine zu dem Timer-Interrupt erfolgt in der Taskkonfiguration. Die Periodendauer kann von 500 – 2 500 000 Mikrosekunden eingestellt werden. Die Periodendauer programmieren Sie, in dem Sie die Funktion TIMERINTERRUPTENABLE aus der Bibliothek EC_Util.lib in Ihr Anwenderprogramm mit einbinden.

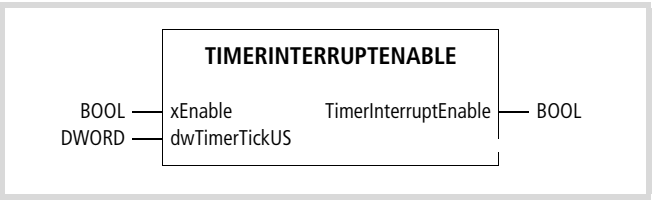


Abbildung 49: Funktion TimerInterruptEnable

Am Eingang „dwTimerTickUS" geben Sie die Periodendauer an.

Der Wert wird beim Start des Timers übernommen und kann während des Ablaufs nicht geändert werden. Wird der Wert von 500 unter- bzw. der Wert von 2 500 000 überschritten, liefert die Funktion als Rückgabewert FALSE und der Timer wird nicht ausgeführt.

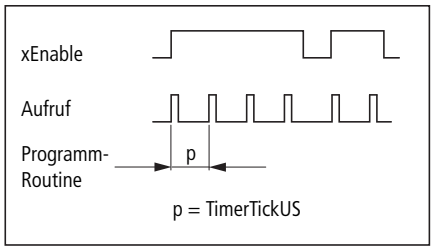


Abbildung 50: Periodischer Aufruf der Programm-Routine

Soll z. B. die Periodendauer von 2 Sekunden von dem externen Eingang I0.0 gestartet werden, müssen Sie folgende Programmzeile in das Anwenderprogramm schreiben:

```
TimerInterruptEnable(%IX0.0,2000000)
```

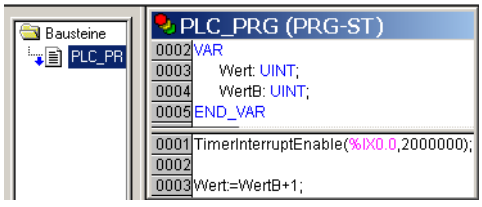


Abbildung 51: Funktion in das Programm einfügen

Beispiel

- Programm mit Funktionsaufruf erstellen

Erstellen Sie ein Programm mit der Funktion TIMERINTERRUPTENABLE nach Abbildung 51.

- Programm-Routine erstellen

▶ Öffnen Sie im Verzeichnis „Ressourcen“ das Unterverzeichnis „Task-Konfiguration“ mit einem Doppelklick.

▶ Klicken Sie hier den Ordner „Systemereignisse“ an. Das Register „System-Ereignisse“ ist aktiv.

▶ Aktivieren Sie den Timer-Interrupt, indem Sie auf das Kästchen links neben dem Namen „Timer-Interrupt“ klicken.

▶ Tragen Sie im Feld „aufgerufene POU“ den Namen der Programm-Routine ein, z. B. „Zeit_Int“.

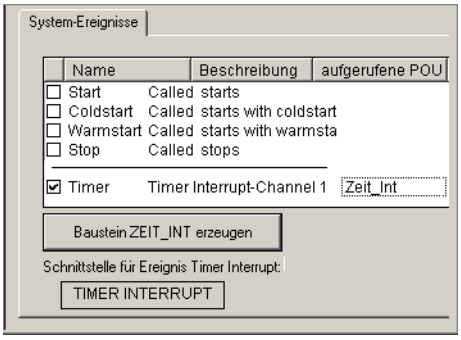


Abbildung 52: Programm-Routine erstellen

▶ Klicken Sie nochmals auf den Namen „Timer-Interrupt“. Jetzt wird die Schaltfläche „Baustein erzeugen“ aktiv und zeigt den Namen der POU an.

▶ Klicken Sie auf diese Schaltfläche. Im Fenster BAUSTEINE wird unter der POU „PLC_PRG“ ein Ordner (POU) mit dem Namen „Zeit_Int“ eingefügt.

▶ Öffnen Sie die POU und schreiben Sie Ihre Programm-Routine:

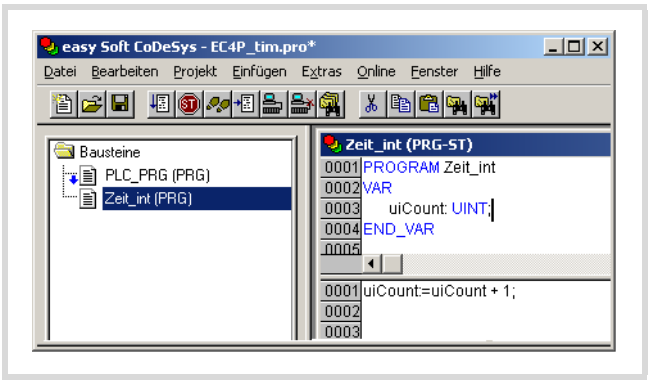


Abbildung 53: Programm-Routine schreiben

Wird der Eingang IX0.0 an Spannung gelegt, wird periodisch die POU „Zeit_Int“ aufgerufen und die Variable „uiCount“ inkrementiert.

→ Der Interrupt kann durch höher priorisierte Systeminterrupts unterbrochen werden. Während der Ausführung des Timer-Interrupts ist die Zykluszeitüberwachung aktiv.

Treten zu häufige Timer-Interrupts auf, kann dies zur Überschreitung der vorgewählten Programm-Zykluszeit führen. In diesem Fall wechselt die Steuerung vom Betriebszustand RUN in den Zustand STOP.

Der Timer-Interrupt kann aus dem Anwenderprogramm heraus gesperrt und freigegeben werden. Hierzu stehen die Funktionen „DisableInterrupt“ und „EnableInterrupt“ aus der Bibliothek EC_UTIL.lib zur Verfügung.

Interruptverarbeitung

Tritt ein Interrupt auf, wird das Anwenderprogramm unterbrochen und die mit dem Systemereignis verknüpfte Programm-Routine bearbeitet. In der Abbildung 54 sind die Interruptquellen aufgeführt.

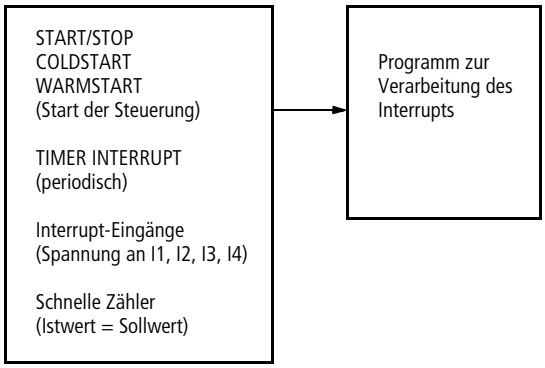


Abbildung 54: Interruptquellen

→ Die Ausführung der Programm-Routinen wird zeitlich überwacht.

Die vom Interrupt aufgerufene Programm-Routine kann durch einen neuen Interrupt (anderer Kanal) unterbrochen werden.

Folgt dem aktuellen Interrupt ein neuer Interrupt (gleicher Kanal), wird der neue Interrupt erst nach Beendigung des aktuellen ausgeführt.

Die Interrupts werden im Zustand RUN der CPU freigegeben und im STOP gesperrt. In der Konfiguration nicht freigegebene Interruptquellen lösen keinen Interrupt aus.

Sie können die Interrupt-Eingänge I1…I4 und den Timer-Interrupt aus dem Programm heraus sperren und wieder freigeben. Hierzu gibt es die Funktionen „DisableInterrupt" und „EnableInterrupt". Ein Aufrufparameter bestimmt, ob ein einzelner Interrupt oder alle Interrupts gesperrt/freigegeben werden. Die Freigabe eines gesperrten Interrupts muss mit dem gleichen Parameter erfolgen, wie die Sperrung.

Die beiden Funktionen „DisableInterrupt" und „EnableInterrupt" sind Bestandteil der Biblithek „EC_Util.lib". Diese Bibliothek- müssen Sie – falls noch nicht vorhanden – vom Bibliotheksverwalter der Programmiersoftware zu Ihrem Projekt einbinden.

DisableInterrupt: Mit dieser Funktion schalten Sie einen parametrierten physikalischen Interrupt aus dem Anwenderprogramm heraus inaktiv.

EnableInterrupt: Mit dieser Funktion geben Sie den zuvor inaktiv geschalteten physikalischen Interrupt wieder als aktiven Interrupt frei.

Arbeitsschritte zur Interruptverarbeitung

▶ Legen Sie die Interrupt-Eigenschaften fest:

Startverhalten	Art auswählen
TIMER INTERRUPT	Funktion aufrufen TIMERINTERRUPTENABLE
Interrupt Eingänge	Flanken bestimmen
Schnelle Zähler	Typ auswählen

▶ Erstellen Sie die Programm-Routine (POU).
Zur bestehenden POU PLC_PRG ist eine weitere Programm-Routine (POU) vom Typ PRGzu erstellen, die ein Interrupt aufruft.

▶ Ordnen Sie der Programm-Routine eine Interruptquelle zu:
- Rufen Sie dazu den Steuerungskonfigurator auf und klicken Sie die Taskkonfiguration l System-Ereignisse an.
  Im Register „System-Ereignisse" sind die Interruptquellen (Namen) aufgeführt und ein freies Eingabefeld für den Namen der „aufgerufenen POU".
- Geben Sie den Interrupt frei, indem Sie auf das Kästchen neben dem gewünschten Interrupt klicken und tragen Sie in der gleichen Zeile den Namen der POU ein. Die Details sind im Beispiel zur Interruptverarbeitung beschrieben.

Beispiel zur Interruptverarbeitung

Eine POU „PLC_PRG" soll ständig abgearbeitet werden. Eine weitere Programm-Routine (POU) „Fastprog" soll abgearbeitet werden, wenn eine steigende (L → H) Flanke am Eingang I3 einen Interrupt erzeugt.

▶ Erstellen Sie die POUs „PLC_PRG"und „Fastprog" gemäß Abbildung 55.

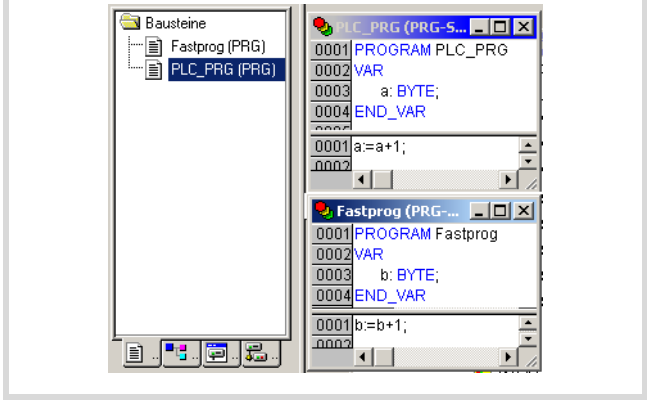


Abbildung 55: Programm erstellen

▶ Wechseln Sie in die Steuerungskonfiguration, klicken Sie auf den Ordner Local I/O[SLOT] und öffnen Sie das Register „Weitere Parameter".
▶ Weisen Sie dem Eingang I3 den Typ „steigende Flanke" zu.

Abbildung 56: Interrupt-Flanke auswählen

▶ Wechseln Sie in die Taskkonfiguration und öffnen Sie den Ordner „System-Ereignisse".

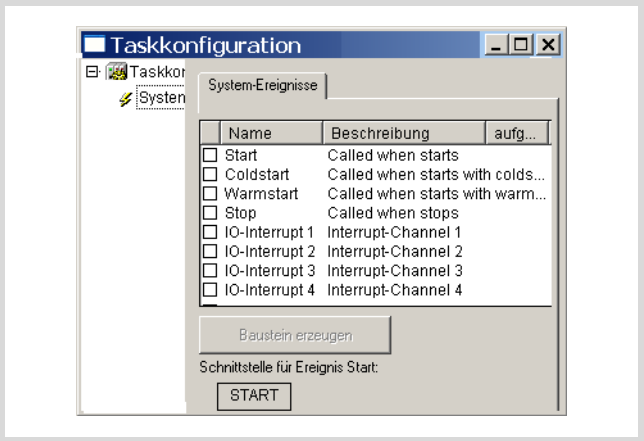


Abbildung 57: Interrupt freigeben

▶ Geben Sie den IO-Interrupt 3 frei, indem Sie auf das Kästchen links neben dem Namen „IO-Interrupt3" klicken.
Die Bestätigung wird durch einen Haken dargestellt.
▶ Markieren Sie den Bereich von Spalte „ aufgerufene POU" und Zeile „IO-Interrupt3".
▶ Stellen Sie den Cursor auf den markierten Bereich und betätigen Sie die Funktionstaste F2.

Das Fenster „Eingabehilfe" wird geöffnet. Dort sind alle vordefinierten Programme aufgelistet:

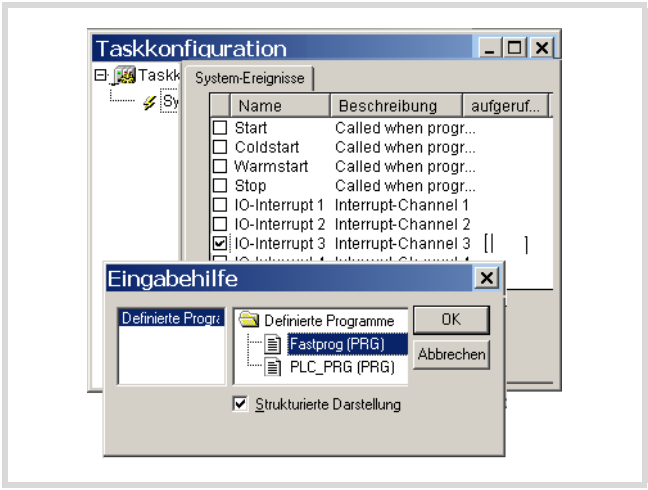


Abbildung 58: Zuweisung Interruptquelle → POU

▶ Wählen Sie die POU „Fastprog" aus und bestätigen Sie mit OK.
▶ Speichern Sie das Projekt. Sie können es nun testen.

Mit jeder steigenden Flanke am Eingang I3 wird die Variable „b" um eins erhöht.

Direkter I/O-Zugriff

Mit den Funktionen der Bibliothek EC_Util.lib zum direkten I/O-Zugriff greifen Sie direkt auf die lokalen Ein- und Ausgänge der Steuerung zu. Der Zugriff erfolgt sofort aus dem Anwenderprogramm, nicht über das Ein-/Ausgangs-Abbild. Für die folgende Ein-/Ausgänge wird der Zugriff nicht unterstützt:

- Ein-/Ausgänge der Erweiterungsgeräte
- Lokale Diagnosebits
- Tasten der Schaltmatte
- Ein-/Ausgänge der Geräte, die über ein Bussystem eingebunden sind.

Der Zugriff auf die „Schnellen Zähler" kann mit Funktionsbausteinen der Bibliothek EC_Util2.Lib erfolgen.

Beschreibung der Funktionen

Die Funktion „ReadBitDirect" wird beispielhaft für alle weiteren Funktionen beschrieben.

ReadBitDirect

Mit dieser Funktion können Sie den Zustand eines Eingangsbits direkt lesen. Er wird in der Variablen abgelegt, auf die der parametrierte Pointer „ptr_xValue" zeigt. Die Pointervariable wird nicht verändert, wenn bei der Bearbeitung ein Fehler auftritt.

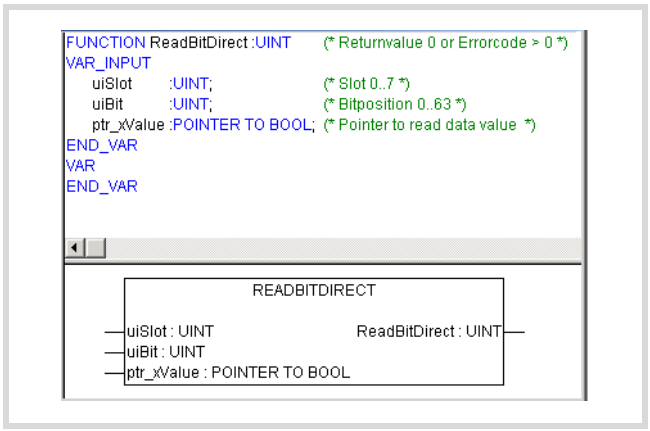


Abbildung 59: Funktion ReadBitDirect

Funktionsaufruf: ReadBitDirect(uiSlot, uiBit, ptr_xValue)

Die folgenden Tabellen zeigen die Zugriffsmöglichkeiten.

Tabelle 10:Funktionen zum Ansteuern der I/Os

EC_UTIL.Lib	**EC4-200-I/O**			
	I1-I12 digital	**I7,I8,I11,I12 analog**	**Q1-Q8 digital**	**QA1 analog**
ReadBitDirect	Bit 0-11	–	–	–
ReadByteDirect	Byte 0+1	–	–	–
ReadWordDirect	–	Offset 2,4,6,8	–	–
WriteBitDirect	–	–	Bit 0-7	–
WriteByteDirect	–	–	Byte 0	–
WriteWord Direct	–	–	–	Offset 2

Tabelle 11:FBs für die „Schnellen Zähler"

EC_UTIl2.Lib	**Schnelle Zähler**		
Funktionsbaustein	**16 Bit**	**32 Bit**	**Incremental**
Acc16BitCounterDirect	Offset 0+1	–	–
Acc32BitCounterDirect	–	ok	–
AccIncremental InputDirect	–	–	ok

Fehlercode bei „Direkter I/O-Zugriff"

Alle Funktionen prüfen soweit wie möglich die Gültigkeit der Aufrufparameter. Bei einem festgestellten Fehler wird der Zugriff nicht durchgeführt und ein Fehlercode ausgegeben.

Folgende Rückgabewerte sind möglich:

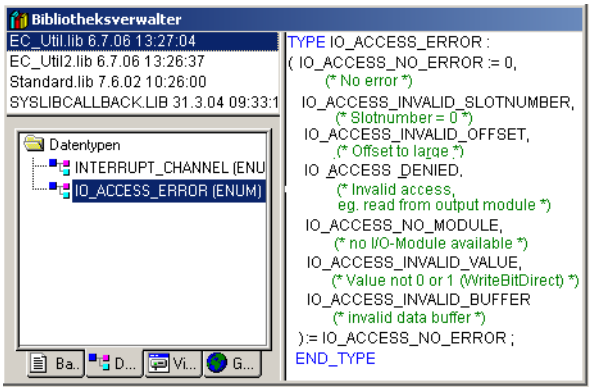


Abbildung 60: Rückgabe der EC-UTIL.Lib-Funktionen

Tabelle 12:Übersicht über die Rückgabewerte

	IO_ACCESS_INVALID_SLOTNUMBER	**IO_ACCESS_INVALID_OFFSET**	**IO_ACCESS_DENIED**	**IO_ACCESS_NO_MODUL**	**IO_ACCESS_INVALID_BUFFER**
READBITDIRECT	x	x	x		x
READBYTEDIRECT	x	x	x		x
READWORDDIRECT	x	x	x		x
WRITEBITDIRECT	x	x	x		
WRITEBYTEDIRECT	x	x	x		
WRITEWORDDIRECT	x	x	x		
ACCESS16BITCOUNTERDIRECT		x	x		
ACCESS32BITCOUNTERDIRECT			x		
ACCESSINCREMENTALINPUTDIRECT			x		

Bootprojekt erzeugen und transferieren

Die CPU arbeitet das im Arbeitsspeicher gespeicherte Anwenderprogramm ab. Da der Arbeitsspeicher nicht gepuffert ist, wird das Programm bei Spannungsausfall gelöscht. Um das Programm remanent zu sichern, erstellen Sie ein Bootprojekt.

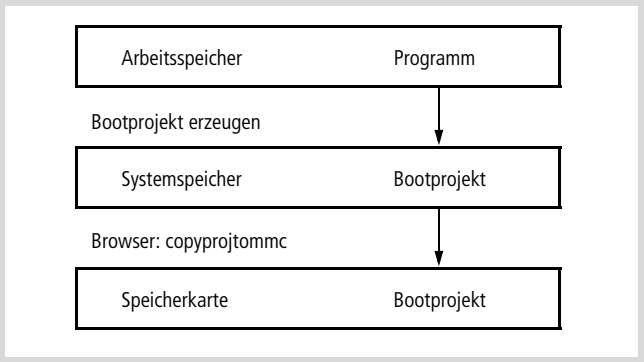


Abbildung 61: Bootprojekt speichern

Das Bootprojekt können Sie im Online-Betrieb oder über das Menü der Steuerung erzeugen. Das Bootprojekt wird mit Hilfe des aktuellen Betriebssystems der Steuerung erzeugt!

Im Online-Betrieb sind folgende Schritte nötig:

- ▶ Wechseln Sie in das Menü „Online" und wählen Sie den Befehl „Einloggen".
- ▶ Im RUN-Zustand werden Sie aufgefordert, die Steuerung anzuhalten.
- ▶ Wählen Sie den Befehl „Bootprojekt erzeugen".

Es erscheint folgende Abfrage:

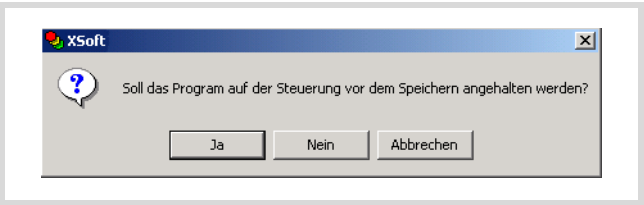


Abbildung 62: Bootprojekt erzeugen

- ▶ Beantworten Sie diese Frage mit „Ja".

Es erscheint für eine kurze Zeit das folgende Fenster:

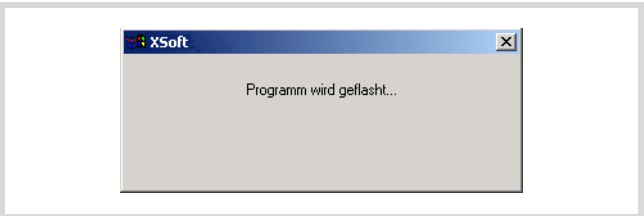


Abbildung 63: Bootprojekt erstellen

Mit dem automatischen Ausblenden des Fensters ist das Bootprojekt erstellt. Sie können nun die Steuerung wieder starten.

Bootprojekt auf Speicherkarte abspeichern

Das im Systemspeicher (Flash) abgelegte Bootprojekt kann zusätzlich auf der Speicherkarte abgepeichert werden. Das geschieht durch den Aufruf des Browserkommandos „copyprojtommc" im Online-Betrieb oder über die Bedienelemente der Steuerung mit der Auswahl aus dem Hauptmenü: PROGRAMM → BOOTPROJEKT → FLASH → KARTE.

Bootprojekt und Betriebssystem (BTS) auf Speicherkarte

Das Bootprojekt arbeitet nur mit dem aktuellen Betriebssytem (BTS) zusammen, unter dem es erstellt wurde!

Stecken Sie die Speicherkarte mit einem BTS in die Steuerung, so wird nach dem Einschalten das BTS der Steuerung aktualisiert und ein Bootprojekt in die Steuerung geladen. Wurde das Bootprojekt nicht mit diesem BTS erstellt, wird es von der Steuerung nicht erkannt. In diesem Fall laden Sie das Programm und erstellen Sie ein neues Bootprojekt!

Bootprojekt löschen

Der Browser-Befehl „Remove" löscht sowohl das Bootprojekt im Systemspeicher (Flash) als auch auf der Speicherkarte. Mit dem Browser-Befehl „removeprojfrommmc" wird das Bootprojekt und die Startup.INI-Datei auf der Speicherkarte gelöscht. Das Löschen des Bootprojektes auf der Speicherkarte kann auch über das Menü der Steuerung erfolgen: PROGRAMM → LÖSCHEN → KARTE LÖSCHEN.

Betriebssystem herunterladen/aktualisieren

Bei der EC4-200 haben Sie die Möglichkeit, das gespeicherte Betriebssystem (BTS) durch ein aktuelleres zu ersetzen.
Eaton bietet die jeweils aktuelle BTS-Version im Internet als Download an: http://www.eaton.com/moeller → **Support**
Zusätzlich ist das aktuelle Betriebssystem auch auf der jeweils neuesten „easySoft-CoDeSys"-CD enthalten.

Achtung!
Der Download kann nur im Offline-Betrieb über die RS232-Schnittstelle erfolgen! Beim Download des BTS werden alle auf der Steuerung/Speicherkarte befindlichen Dateien gelöscht. Die Steuerung führt einen „Reset Ursprung " aus → Seite 44.

Sie haben zwei Möglichkeiten, das Betriebssystem zu übertragen:

- Direkt vom PC in die Steuerung
- Vom PC auf die Speicherkarte. Beim Start der Steuerung wird das BTS von der MMC in die Steuerung kopiert.

Betriebssystem vom PC in die Steuerung übertragen

▶ Öffnen Sie ein Projekt und aktivieren Sie unter ‹Ressourcen → Steuerungskonfigurator› das Register „Allgemeine Parameter".
▶ Klicken Sie im Feld „Aktualisieren des Betriebssystems" auf die Schaltfläche „Start".

Das Fenster „Betriebssystem übertragen" öffnet sich.

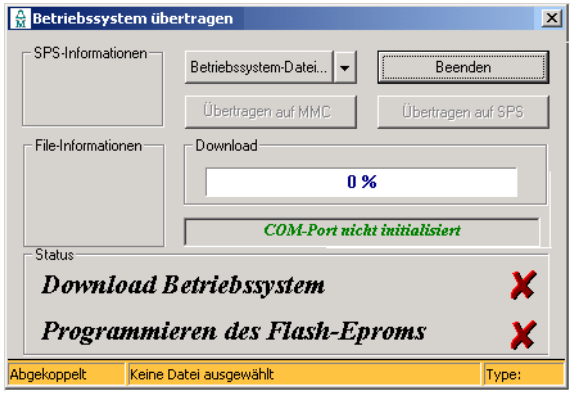


Abbildung 64: Betriebssystem übertragen

Das System meldet, dass der COM-Port nicht initialisiert ist.

▶ Klicken Sie auf die Schaltfläche „Betriebssystem-Datei" und wählen Sie die gewünschte Betriebssystem-Datei (*.hex) aus.

→ Zuletzt geöffnete Dateien können Sie aus dem Listenfeld (Drop-Down-Menü) auswählen.

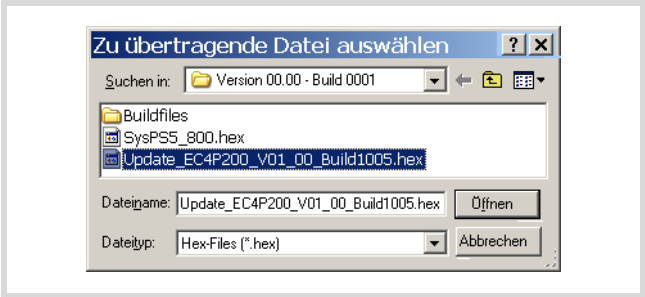


Abbildung 65: Betriebssystem-Datei auswählen

Nach der Auswahl der BTS-Datei erhalten Sie eine Information über den Zieltyp und die Fileversion.

▶ Betätigen Sie die Schaltfläche „Übertragen auf SPS".
▶ Wählen Sie die RS232-Schnittstelle aus.

Die Übertragung beginnt. Das Programmieren des „Flash-Eproms" dauert ca. 20 bis 30 Sekunden.

→ Wenn während der Übertragung ein Warnzeichen im Feld „Programmieren des Flash-Eproms" erscheint, darf die Spannung nicht abgeschaltet werden!

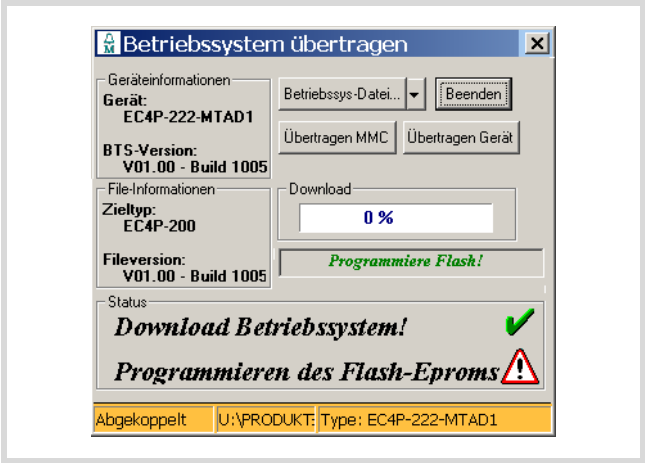


Abbildung 66: Warnung beim Download

Warten Sie die folgende Anzeige ab.

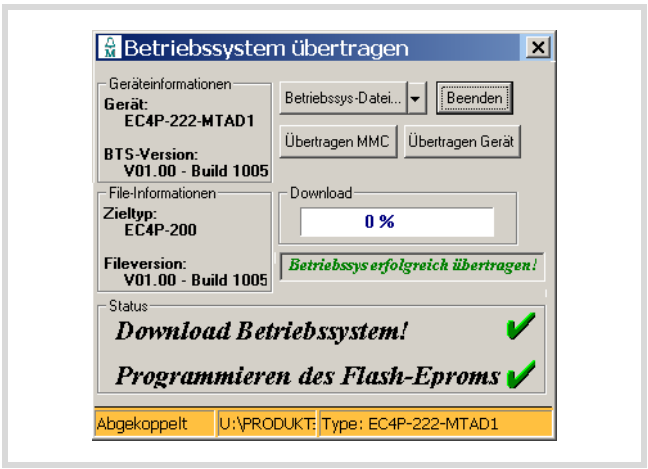


Abbildung 67: Betriebssystem auf SPS übertragen

▶ Klicken Sie in diesem Fenster auf die Schaltfläche „Beenden“.

BTS vom PC auf die Speicherkarte übertragen

Wenn Sie ein BTS auf die Speicherkarte laden, wird das bestehende BTS und das Bootprojekt auf der Speicherkarte und das Anwenderprogramm in der Steuerung gelöscht. Die Vorgehensweise ist analog zur Beschreibung im Abschnitt „Betriebssystem vom PC in die Steuerung übertragen“. Klicken Sie in diesem Fall auf die Schaltfläche „Übertragen auf MMC“, → Abbildung 64 auf Seite 56.

BTS von der Speicherkarte in die Steuerung übertragen

▶ Stecken Sie die Speicherkarte im ausgeschalteten Zustand in die Steuerung.
▶ Schalten Sie die Steuerung ein.

Beim Einschalten wird das BTS der Steuerung aktualisiert und ein Bootprojekt in die Steuerung geladen. Die Übertragung kann über 30 Sekunden dauern, da die CPU mehrfach gebootet wird.

→ Unterbrechen Sie den Vorgang nicht, indem Sie z. B. die Versorgungsspannung ausschalten.

10 Browser-Befehle

Der PLC-Browser ist ein textbasierter Steuerungsmonitor. Hier geben Sie Kommandos zur Abfrage von Statusinformationen der Steuerung in einer Eingabezeile ein. Sie werden als String zur Steuerung gesendet. Die Antwort zeigt ein Ergebnisfenster des Browsers an. So können Sie eine Diagnose oder ein Debugging durchführen.

→ Die Browser-Befehle sind nur online anwendbar.

Um diese Befehle aufzurufen, sind folgende Schritte notwendig:

▶ Wählen Sie in der Programmiersoftware im Verzeichnis „Ressourcen" mit einem Doppelklick den „PLC-Browser" aus.

Es erscheint im Feld rechts ein neues Fenster mit der Bezeichnung „PLC-Browser".

▶ Klicken Sie die Schaltfläche an.

Im Auswahlfeld werden die zur Verfügung stehenden Browser-Befehle angezeigt.

▶ Wählen Sie den gewünschten Befehl mit einem Doppelklick aus.

Jetzt erscheint im Fenster „PLC-Browser" der gewählte Befehl.

▶ Betätigen Sie die Eingabe-Taste, um die Antwort der Steuerung auf den Browser-Befehl im Ergebnisfenster zu sehen.

→ Weiterführende Informationen zu dem ausgewählten Browser-Befehl erhalten Sie, wenn Sie vor den ausgewählten Browser-Befehl ein „?" mit einem anschließenden Leerzeichen setzen und die Eingabetaste (RETURN-Taste) drücken.

Die Beschreibung der Befehle finden Sie auch im Handbuch zur Programmiersoftware (MN05010003Z-DE) im Kapitel ‹Die Ressourcen → PLC-Browser›.

Die Steuerung unterstützt die Browser-Befehle aus Tabelle 13.

Ethernet-Parameter einstellen

Wenn Sie einen Browserbefehl zum Einstellen der IP-Adresse/Subnetzmaske und Gateway-Adresse ausführen, wird dieser erst mit einer der folgenden Aktionen wirksam:

- Aus- und Einschalten des Gerätes
- Eingabe des Befehls „reboot"
- Ethernet-Kabel an der EC4-200 ziehen und wieder stecken.

Tabelle 13: Browser-Befehle

?	Liste der implementierten Kommandos holen
pinf	Projektinformationen ausgeben
cycle	Zykluszeit ausgeben
canload*	Belastung des CAN-Bus anzeigen
copyprojtommc	Kopieren des aktuellen Bootprojekts auf die Speicherkarte
createstartupini	Erzeugen der Startup.INI-Datei auf der Speicherkarte
factoryset	Werkseinstellungen übernehmen
format	Formatieren der Speicherkarte
GetNodeId	Anzeigen der CANopen Node-Id der CAN-Schnittstelle
GetRoutingId	Anzeigen der Routing Node-Id und der Routing-Schnittstelle
getipgateway	Anzeige der eingestellten Gateway-Adresse (Default-Adresse: 0.0.0.0)
getipconfig	Anzeige von eingestellter IP-Adresse und Subnetzmaske Anzeige Defaultwerte: IP-Adresse: 192.168.119.60 Subnetzmaske: 255.255.255.0
getmacaddress	Anzeige der MAC-Adresse des EthernetControllers. Anzeige z. B. 00-80-99-05-11-22
getrtc	Echtzeituhr auslesen
metrics	SPS-Informationen ausgeben
reboot	Reboot der EC4-200 ▽ Nach der Bestätigung des Befehls erfolgt die sofortige Ausführung – ohne Rückfrage!
reload	Bootprojekt vom FLASH in die Steuerung laden
remove	Bootprojekt im FLASH löschen
removeprojfrommmc	Löschen des Bootprojekts (und Datei Startup.INI) von der Speicherkarte
removestartupini	Löschen der Datei Startup.INI von der Speicherkarte
setipconfig	Setzen der IP-Adresse und der Subnetzmaske Syntax (Beispiel): setipconfig 192.168.119.60 255.255.255.0 -> ok oder Fehlermeldung
setipgateway	Setzen der Gateway-Adresse Syntax (Beispiel): setipconfig 192.168.119.10 -> ok oder Fehlermeldung
setrtc*	Echtzeituhr setzen

Zu den mit * markierten Befehlen finden Sie gleich anschließend noch weitere Informationen.

Beschreibung wichtiger Browser-Befehle

canload

Zeigt die Auslastung des CANopen-Feldbusses an.

Beispiel:

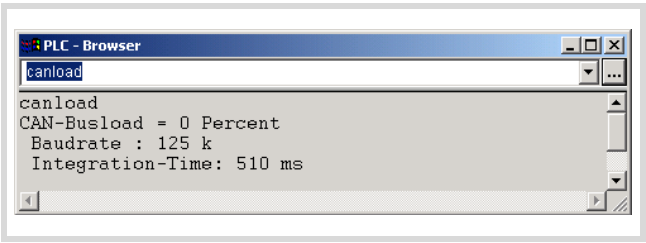


Abbildung 68: Browser-Befehl „canload"

Nach dem Aufruf des Browser-Befehls erhalten Sie z. B. folgende Infomation:

- CAN-Busload = 0 Percent
- Baudrate 125 kBaud
- Integration-Time: 510 ms

Achtung!
Bei einer Busbelastung von 75 Prozent oder mehr erscheint zusätzlich der Warnhinweis ATTENTION: HIGH BUSLOAD. Die Überlastung des lokalen CAN-Busses in Zusammenhang mit weiteren kurzzeitigen Lastspitzen kann zu CAN-Datenverlust führen.

→ Zusätzlich zum Browser-Befehl steht die Funktion CAN_BUSLOAD zur Ermittlung der CAN-Busauslastung aus dem Anwenderprogramm heraus zur Verfügung,
→ Abschnitt „Funktion CAN_BUSLOAD" auf Seite 62.

setrtc

Stellt oder verändert das Datum und/oder die Uhrzeit in der Steuerung.

Syntax:

```
<setrtc_YY:MM:DD:DW_HH:MM:SS>
```

Legende:

_	Leerzeichen
YY	die beiden letzten Ziffern der Jahreszahl (00 F YY F 99)
MM	Monat (01 F YY F 12)
DD	Tag (01 F DD F 31)
DW	Wochentag (01 F DW F 07; 01 = Montag, 07 = Sonntag)
HH	Stunde (00 F HH F 23)
MM	Minute (00 F MM F 59)
SS	Sekunde (00 F SS F 59)

11 Bibliotheken, Funktionsbausteine und Funktionen

Die Bibliotheken enthalten IEC-Funktionsbausteine und Funktionen, die Sie z. B. für folgende Aufgaben anwenden können:

- Datenaustausch über den CANopen-Bus
- Bedienung der Echtzeituhr
- Ermittlung der Busbelastung des CANopen-Bus
- Interrupt ausführen
- Daten über die Schnittstellen senden/empfangen.

Die Bibliotheken finden Sie in den Verzeichnissen:

- Lib_Common für alle Steuerungen
- Lib_EC4P_200 für die Steuerung EC4-200

Bibliotheken handhaben

Beim Öffnen eines Projekts werden die Bibliotheken „Standard.lib" und „SYSLIBCALLBACK.lib" in den Bibliotheksverwalter kopiert. Sind für die Anwendung weitere Bibliotheken notwendig, müssen Sie diese nachinstallieren.

Die im Bibliotheksverwalter stehenden Bibliotheken werden nach dem Speichern dem Projekt zugeordnet. Öffnen Sie das Projekt wieder, werden auch diese Bibliotheken aufgerufen.

Die folgende Übersicht listet die Dokumente auf, in denen die Funktionsbausteine und Funktionen beschrieben sind.

Dokument	Bibliothek
MN05010003Z-DE (frühere Bezeichnung AWB2700-1437D)	Standard.lib Util.lib XX_Util. Lib
Online-Hilfe oder PDF-Dateien (im Windows-Startmenü unter <Programme → Moeller Software → easySoft-CoDeSys → Dokumentation → Automation Manuals>	SysLib…pdf
MN05010002Z-DE (frühere Bezeichnung AWB2786-1456D)	XS40_MoellerFB. Lib/Visu. Lib/…
AN2700K20	3S_CANopenDevice. Lib 3S_CANopenManager. Lib
AN2700K19	3S_CANopenNetVar. Lib
AN2700K27	SysLibCan. Lib
MN05010001Z-DE (frühere Bezeichnung AWB2786-1554D)	CANUserLib. lib CANUser_Master. lib

Weitere Systembibliotheken installieren

Zum Nachinstallieren von Bibliotheken sind folgende Schritte notwendig:

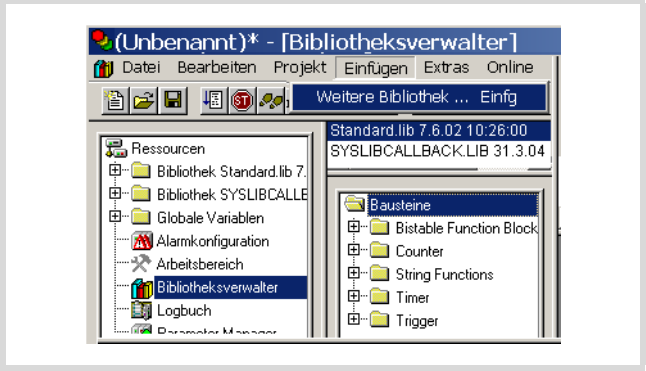


Abbildung 69: Bibliotheken nachinstallieren

▶ Klicken Sie im angelegten Projekt auf das Register „Ressourcen".
▶ Klicken Sie das Verzeichnis „Bibliotheksverwalter" mit der linken Maustaste doppelt an.
▶ Klicken Sie auf den Menüpunkt ‹Einfügen → Weitere Bibliothek… Einfg›.

Im neuen Fenster werden die zur Verfügung stehenden, vom Zielsystem abhängigen, Bibliotheken dargestellt.

▶ Wählen Sie die nachzuinstallierende Bibliothek aus und klicken Sie auf die Schaltfläche „Öffnen".

Die Bibliothek erscheint danach im Bibliotheksverwalter.

EC4-200-spezifische Funktionen

Bibliothek EC_Util.lib

Diese Bibliothek stellt die in der folgenden Abbildung dargestellten Funktionen zu Verfügung:

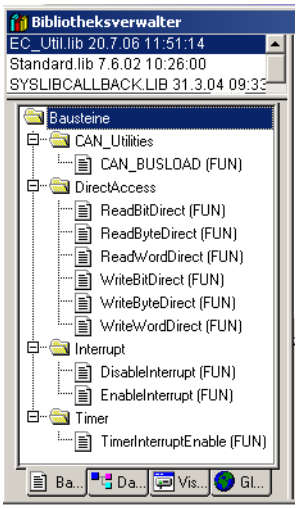


Abbildung 70: Funktionen der Bibliothek EC_Util.lib

Funktion CAN_BUSLOAD

Die Funktion kann in einem Anwenderprogramm zyklisch aufgerufen werden. Wenn ein Lesezyklus erfolgreich beendet wurde, liefert die Funktion den Wert TRUE zurück und schreibt die ermittelten Werte für Integrationszeit und Busauslastung auf die übergebenen Adressen.

Ist die Berechnung der Buslast noch nicht beendet, oder ist der CAN-Controller nicht initialisiert, liefert die Funktion als Rückgabewert FALSE.

Informationen zur Auswertung des Rückgabewertes finden Sie beim Browser-Befehl „canload" auf Seite 60.

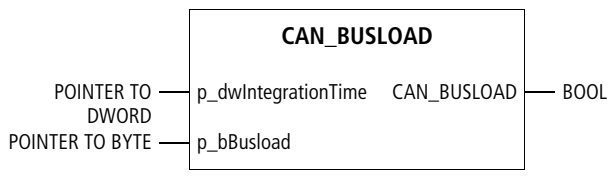


Abbildung 71: Funktion CAN_BUSLOAD

Die übrigen Funktionen finden Sie auf folgenden Seiten:

- Direkter I/O-Zugriff (DirectAccess) → Seite 53
- TimerInterruptEnable → Seite 50
- DisableInterrupt/EnableInterrupt → Seite 52

Bibliothek EC_Visu.lib/EC_Visu2.lib

In der Bibliothek EC_Visu2.lib sind die Funktionsblöcke zum Ansteuern des LC-Displays zusammengefasst.

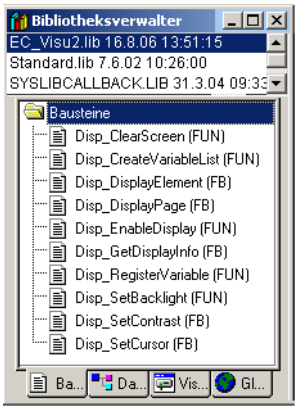


Abbildung 72: Bibliothek EC_Visu2.lib

→ Die bereits bestehenden Funktionen/Funktionsbausteine „SetBacklight", „SetContrast" und „GetDisplayInfo" aus der Bibliothek EC_Visu.lib können Sie weiter nutzen. Sie werden jedoch durch die in der Bibliothek EC_Visu2.lib enthaltenen Funktionen/Funktionsbausteine ersetzt (→ Tabelle)

EC_Visu.lib	EC_Visu2.lib
SetBacklight	Disp_SetBacklight
SetContrast	Disp_SetContrast
GetDisplayInfo	Disp_GetDisplayInfo

Eine Beschreibung dieser Funktionen finden Sie auf Seite 76.

12 Verbindungsaufbau PC – EC4-200

Zur Herstellung der Verbindung zwischen PC und Steuerung müssen die Kommunikationsparameter beider Geräte übereinstimmen. An Geräten, die erstmalig zum Einsatz kommen, sind bereits die Default-Parameter nach Abbildung 73 eingestellt. Wählen Sie nur noch die PC-Schnittstelle COM… aus. Weitere Einstellungen sind nicht erforderlich.

→ Falls Sie eine Fehlermeldung erhalten, sind die Standardeinstellungen der CPU bereits verändert worden. Versuchen Sie es in diesem Fall mit allen anderen Baudraten oder stellen Sie die Werkseinstellung ein.

Anschließend können Sie die Parameter der CPU neu festlegen (→ Abbildung 74). Diese Parameter-Änderungen müssen Sie dann für den PC erneut anpassen.

Die Verbindung zwischen PC und der Programmierschnittstelle der Steuerung können Sie herstellen über:

- die RS232-Schnittstelle
- die Ethernet-Schnittstelle (bei den Typen EC4P-222-… zusätzlich zur RS232)

Verbindungsaufbau über die RS232

Passen Sie zunächst die Kommunikationsparameter des PCs an die Standardparameter der Steuerung an → Abschnitt „Kommunikationsparameter des PCs festlegen/ändern".

Die RS232-Schnittstelle der Steuerung (COM1) hat folgende Standardparameter:

Baudrate	38400 Bit/s
Parity	No
Stopbits	1
Motorola Byte	No

Kommunikationsparameter des PCs festlegen/ändern

Sie können die COM1- bis COMx-Schnittstelle des PCs verwenden. In der Programmiersoftware legen Sie die Kommunikationsparameter der Schnittstelle fest.

▶ Wählen Sie den Menüpunkt ‹Online → Kommunikationsparameter› aus.
▶ Legen Sie den Port (COM1- oder COM2-Schnittstelle) fest, → Abschnitt „Parameter ändern"
▶ Übernehmen Sie die weiteren Parameter aus der Abbildung 73.
▶ Bestätigen Se die Parameter mit OK.
▶ Loggen Sie sich in die Steuerung ein.

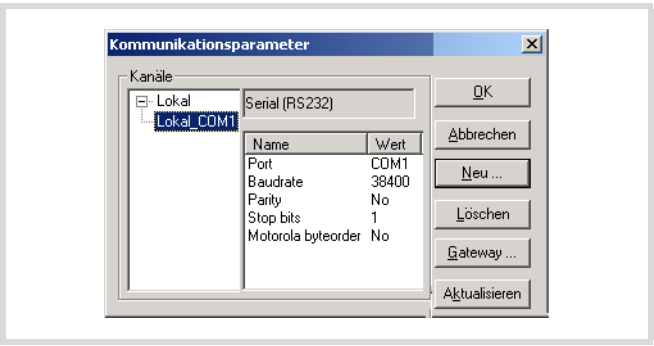


Abbildung 73: Kommunikationsparameter des PCs festlegen

Parameter ändern

Um Parameter wie z. B. die Baudrate oder die Port-Angabe zu verändern, gehen Sie wie folgt vor:

▶ Klicken Sie doppelt auf den Wert, z. B. 38400. Das Feld wird grau unterlegt.
▶ Geben Sie den gewünschten Wert an.

Mit weiteren Doppelklicks in diesem Feld wählen Sie die gewünschte Baudrate, z. B. 57600 Bit/s aus.

Kommunikationsparameter (Baudrate) der CPU ändern

- ▶ Öffnen Sie die Steuerungskonfiguration›.
- ▶ Wählen Sie die Registerkarte „Kommunikation".
- ▶ Wählen Sie im Listenfeld „Baudrate" die Baudrate aus (z. B. 57600 Bit/s wie in Abbildung 74).

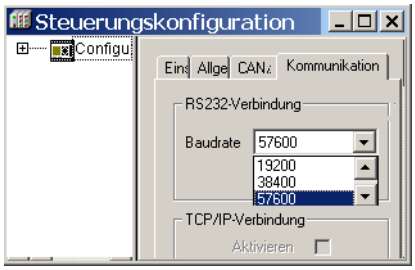


Abbildung 74: Kommunikationsparameter der CPU festlegen

- ▶ Loggen Sie sich in die Steuerung ein.

Sie erhalten folgende Abfrage:

Abbildung 75: Abfrage nach Programmänderung

- ▶ Beantworten Sie diese Frage mit „Ja".

Das Programm wird geladen. Nach einer Verzögerungvon ca. 2 Min. erhalten Sie die Fehlermeldung für Kommunikationsfehler, da die Baudrate zwischen CPU und PC nicht mehr übereinstimmt:

Abbildung 76: Kommunikationsfehler

- ▶ Bestätigen Sie die Fehlermeldung mit OK.

Um den PC wieder ankoppeln zu können, müssen Sie die Baudrate des PCs erneut anpassen und zwar an die des Projekts.

Verbindungsaufbau über Ethernet

Nachdem Sie den PC mit der Steuerung durch ein Ethernet-Kabel verbunden haben, wählen Sie in der Programmiersoftware im Fenster „Kommunikationsparameter" den Kommunikationskanal TCP/IP aus und geben die IP-Adresse der Steuerung an. Die Steuerung hat die Default-Adresse 192.168.119.60.

Kommunikationskanal und Adresse auswählen

- ▶ Rufen Sie das Menü ‹Online → Kommunikationsparameter› auf.

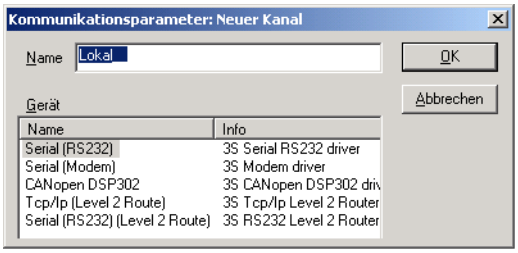


Abbildung 77: Kanalauswahl

- ▶ Betätigen Sie die Schaltfläche „Neu…".
- ▶ Wählen Sie aus der Übersicht den Kommunkationskanal TCP/IP (Level2Route) aus und ändern Sie den Namen „Lokal" z. B. in „Ethernet-Test"
- ▶ Bestätigen Sie mit OK.

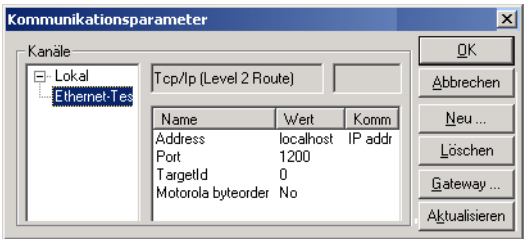


Abbildung 78: IP-Adresse eingeben

- ▶ Führen Sie einen Doppelklick auf das Feld „localhost" aus und tragen Sie die Default-Adresse 192.168.119.60 ein.
- ▶ Bestätigen Sie Ihre Angaben, indem Sie zunächst auf ein beliebiges anderes Feld und erst dann auf OK klicken.

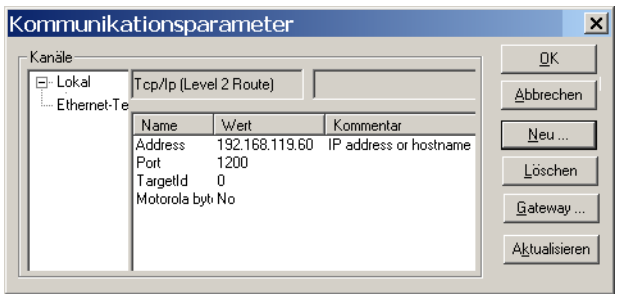


Abbildung 79: Kommunikationsparameter mit IP-Adresse

▶ Übersetzen Sie das Programm und loggen Sie sich ein.

Datenübertragung mit TCP/IP (Level2Route)-Protokoll
Diese Datenübertragung wird von den Steuerungstypen XC200, MFD4 und EC4-200 (nur EC4P-222…) unterstützt.

Die Daten zwischen PC und Steuerung werden in Datenblöcken mit einer Größe von 128 kByte (Einstellung bei Auslieferung des Gerätes) übertragen. Bei der EC4P-222… gilt folgende Einschränkung: Werden größere Datenmengen übertragen, kann dies aufgrund des zu geringen Speicherplatzes zu einer Fehlermeldung führen. Setzen Sie in diesem Fall die Blockgröße auf 4 kByte.
Die Einstellung nehmen Sie im Ordner „Konfiguration" im Register „Kommunikation" vor:

▶ Klicken Sie einfach auf die Schaltfläche „Einstellungen anpassen". Die 4-kByte-Blockgröße wird dann automatisch eingestellt.

Ist die Schaltfläche gegraut wie in Abbildung 80, ist die Blockgröße bereits auf 4 kByte eingestellt.

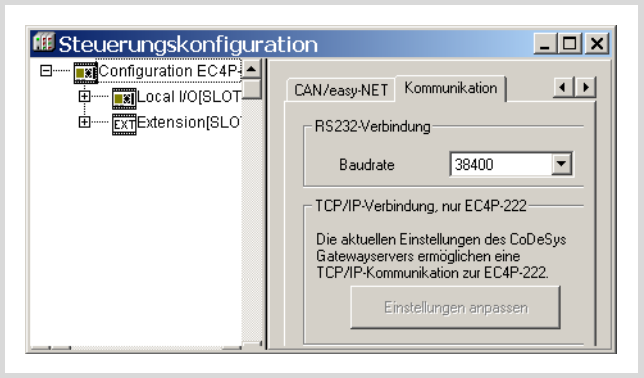


Abbildung 80: Blockgröße der TCP/IP-Verbindung einstellen

Stellen Sie die Blockgröße wieder auf 128 kByte zurück, wenn Sie eine XC200 oder ein MFD4 parametrieren. Bei diesen Steuerungen ist mehr Speicherplatz verfügbar. Die Einstellung können Sie beliebig mit der Anwendung „BlockSizeEditor.exe" verändern (zu finden im Windows-Startmenü unter Programme → Moeller Software → easy Soft CoDeSys → Kommunikation…).

Abfragen/Ändern der IP-Adresse

Für die Änderung und das Abfragen der IP-Adresse stehen Ihnen die Browser-Befehle „setipconfig“ und „getipconfig“ zur Verfügung → Abschnitt „Browser-Befehle“ auf Seite 59.

Nachdem Sie die IP-Adresse geändert haben, starten Sie die Steuerung neu. Achten Sie darauf, dass die IP-Adresse des Programmiergerätes der gleichen Adressenfamilie angehört. Das bedeutet, dass die IP-Adresse des Programmiergerätes und der Steuerung in folgenden Zifferngruppen bei einer Subnetmaske von 255.255.255.0 überein stimmt:

Beispiel 1
IP-Adresse Steuerung:192.168.119.xxx
IP-Adresse PC: 192.168.119.yyy

Beispiel 2
IP-Adresse Steuerung:192.168.100.xxx
IP-Adresse PC: 192.168.100.yyy

Im Beispiel 1 und 2 gelten folgende Bedingungen:

- xxx ungleich yyy
- die Adressen müssen in den Grenzen 1 und 254 liegen.
- Adressen müssen der gleichen Adressfamilie angehören.

Kommt keine Verbindung zu Stande, können Sie den Übertragungsweg mit der „PING“-Funktion überprüfen, um sicher zu sein, dass der Verbindungaufbau nicht am Übertragungsweg scheitert. Hierzu sind folgende Schritte erforderlich:

▶ Öffnen Sie das DOS-Fenster über das „START“Feld und den Befehl „Ausführen“.
▶ Geben Sie in das Eingabefeld „CMD“ ein und bestätigen Sie dies mit „OK“.

Sie erhalten ein Fenster mit einer Laufwerksangabe und einem blinkenden Cursor hinter der Laufwerksbezeichnung.

▶ Hier geben Sie konkret für dieses Beispiel den folgenden Text ein: „ping 192.168.119.60“ und bestätigen dies mit „OK“.

Ist das Routing in Ordnung, bekommen Sie eine Antwort mit Angabe einer Response-Zeit. Andernfalls werden Sie auf eine Zeitüberschreitung beim Verbindungsaufbau hingewiesen.

Die nachfolgende Abbildung zeigt das Ergebnis eines ordnungsgemäßen Verbindungsaufbaus.

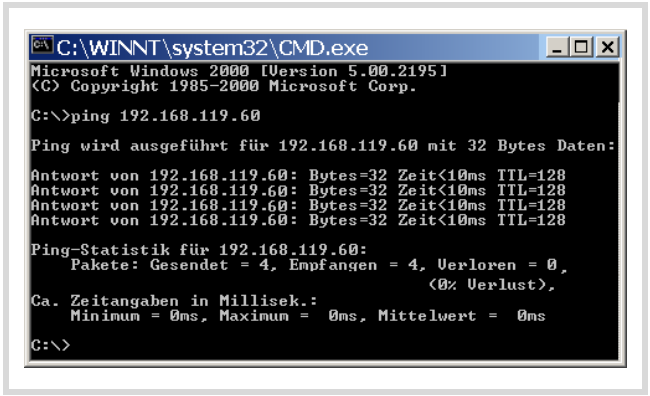


Abbildung 81: PING-Antwort bei einer ordnungsgemäßer Ethernet-Verbindung

13 Systemparameter über STARTUP.INI-Datei vorgeben

Übersicht

Projektunabhängige Systemparameter können Sie erstellen und auf der Speicherkarte speichern. Sie werden dort in der Startup.INI-Datei zusammengefasst. Die Speicherkarte können Sie auch in andere Steuerungen stecken. Beim Start übernimmt die Steuerung die Parameter. Die Startup.INI-Datei wird immer mit allen steuerungsspezifischen Einträgen erstellt (→ Tabelle 14).

Tabelle 14:Parameter in der Startup.INI-Datei

```
Einträge
COM_BAUDRATE: 4800,9600,19200,38400,57600
CAN1_BAUDRATE: 10,20,50,100,125,250,500
CAN1_NODEID: 1-127
CAN_ROUTINGID: 1-127
IP_ADDRESS=xxx.xxx.xxx.xxx
IP_SUBNETMASK=xxx.xxx.xxx.xxx
IP_GATEWAY=xxx.xxx.xxx.xxx
```

Die Parameter aus der INI-Datei haben Vorrang vor den Parametern aus der Steuerungskonfiguration. Nach einem Programm-Download oder nach Laden des Bootprojektes werden die Parameter aus der Steuerungskonfiguration nicht übernommen.

Aufbau der INI-Datei

Eine INI-Datei ist eine Textdatei mit einem festgelegten Datenformat. Ab einer namentlich benannten Sektion (in eckigen Klammern) wie z. B. [STARTUP] werden die Systemparameter aufgeführt, gefolgt von einem Gleichheitszeichen und deren Wert. Die Zeile wird mit CR/LF (Carriage/Return-Taste) abgeschlossen.

```
COM_BAUDRATE = 38400 (Carriage/Return)
```

Die mit Semikolon beginnenden Zeilen interpretiert die Steuerung als Kommentar und überspringt sie beim Einlesen:

```
; CAN_NODEID = 2
```

Die Parameter können Sie mit einem Text-Editor ändern oder erstellen, wenn Sie die Speicherkarte in den Speicherkarten-Slot des PC stecken. Stecken Sie zunächst die Speicherkarte in den mitgelieferten Adapter, dann in den PC-Slot. Die Datei STARTUP.INI wird auf der Speicherkarte im Verzeichnis „MOELLER/EC4P_200/PROJECT/“ gespeichert.

Startup.INI-Datei erstellen

Grundsätzlich arbeitet die Steuerung beim ersten Einschalten (Grundzustand) mit Default-Systemparametern, den STARTUP-Daten. Wenn Sie ein Projekt in die Steuerung laden, die sich im Grundzustand befindet, startet die Steuerung sofort mit den Systemparametern des Projektes.

Mit dem Browser-Befehl „createstartupini“ übertragen Sie die aktuellen Systemparameter aus der Steuerung auf die Speicherkarte. Dabei wird die Startup.INI-Datei erstellt, die diese Daten enthält. Voraussetzung: Die Speicherkarte muss gesteckt und formatiert sein, d. h. ohne Startup.ini-Datei.

Tabelle 15:Beispiel: STARTUP.INI Datei für EC4-200

```
[STARTUP]
TARGET = EC4P-200
IP_ADDRESS=192.168.119.60
IP_SUBNETMASK=255.255.255.0
IP_GATEWAY=0.0.0.0
COM_Baudrate = 38400
CAN1_Baudrate = 125
CAN1_NODEID = 2
CAN_ROUTINGID = 127
```

Eine bereits vorhandene Datei können Sie nicht über das Browserkommando „createstartupini“ verändern oder überschreiben. Falls Sie das Kommando trotzdem eingegeben, erscheint eine Warnung. Um eine neue Datei zu erstellen, löschen Sie die bestehende Datei zuerst, → Abschnitt „Startup.INI-Datei löschen“ auf Seite 68.

Einschalten der Steuerung bei gesteckter Speicherkarte mit Startup.INI-Datei

Beim Einschalten der Steuerung werden die Daten der Startup.INI-Datei der Speicherkarte in die Steuerung übertragen. Diese Systemparameter bleiben auch nach dem Laden eines neuen Programms aktiv.

Parameter ändern

Die Parameter bleiben erhalten, bis Sie den Browser-Befehl „removestartupini" eingeben und dann die Steuerung aus-/einschalten. Die Steuerung arbeitet nun mit den Parametern des Projekts.

Startup.INI-Datei löschen

Es stehen folgende Browser-Befehle zur Verfügung, die auf die Speicherkarte zugreifen.

- removestartupini:
  Löscht grundsätzlich die Systemparameter in der Steuerung. Wenn eine Speicherkarte gesteckt ist, wird auch die INI-Datei auf der Speicherkarte gelöscht. Beim nächsten Einschalten werden die Parameter aus dem Projekt übernommen.
- removeprojfrommmc:
  Löscht das Boot-Projekt und die INI-Datei auf der Speicherkarte. Die Systemparameter in der Steuerung bleiben erhalten.
- format:
  Löscht die gesamte Speicherkarte incl. INI-Datei.

Das Verhalten der Startup.ini-Datei bei dem Menü-Befehl Reset Ursprung, bei dem Befehl Werkseinstellung im Menü der Steuerung sowie dem Browserbefehl „factoryset" wird im Abschnitt „Reset" auf Seite 44 beschrieben.

14 Programmieren über CAN-Netzwerk (Routing)

Als Routing bezeichnet man die Möglichkeit, eine Online-Verbindung von einem Programmiergerät (PC) zu einer beliebigen (routingfähigen) Steuerung in einem CAN-Netzwerk aufzubauen, ohne dass das Programmiergerät direkt mit der Zielsteuerung verbunden sein muss. Es kann an eine andere Steuerung im Netzwerk angeschlossen werden. Über die Routing-Verbindung können Sie alle Aktionen durchführen, die auch bei einer direkten Online-Verbindung zwischen Programmiergerät und Steuerung zur Verfügung stehen:

- Programm-Download
- Online-Änderungen
- Programmtest (Debugging)
- Erzeugen von Bootprojekten
- Dateien in die Steuerung schreiben
- Dateien aus der Steuerung lesen.

Das Routing bietet den Vorteil, dass Sie von einer Steuerung, die mit dem Programmiergerät verbunden ist, Zugriff auf alle routingfähigen Steuerungen am CAN-Bus erhalten. Durch die Projektauswahl bestimmen Sie, mit welcher Steuerung Sie kommunizieren möchten. So lassen sich dezentral angeordnete Steuerungen leicht bedienen.

Allerdings ist die Datenübertragung von Routing-Verbindungen deutlich langsamer als bei Direktverbindungen (Seriell oder TCP/IP). Dies macht sich beispielsweise durch langsamere Aktualisierungszeiten von Visualisierungselementen (Variablen) oder langsamere Download-Geschwindigkeiten bemerkbar.

Voraussetzungen

Um das Routing einsetzen zu können, müssen folgende Voraussetzungen erfüllt sein:

- Sowohl die routende Steuerung als auch die Zielsteuerung müssen das Routing unterstützen.
- Beide Steuerungen müssen über den CAN-Bus verbunden sein.
- Die Steuerungen müssen über die gleiche aktive CAN-Baudrate verfügen.
- Auf beiden Steuerungen muss eine gültige Routing-Node-Id eingestellt sein.

Routingeigenschaften der Steuerung

Die Steuerung unterstützt das Routing über den CAN-Bus.

Das Routing kann ohne vorherigen Download eines Anwenderprogramms ausgeführt werden (Default:125 kBaud, Node-Id 127). Die Zielsteuerung muss hierfür nicht als CAN-Master bzw. CAN-Device konfiguriert werden.

Sie können z. B. ein Programm vom PC über eine Steuerung der XC-Gerätefamilie in die EC4-200 laden. Ordnen Sie in diesem Fall der EC4-200 (Zielsteuerung) eine Routing-Node-Id zu.

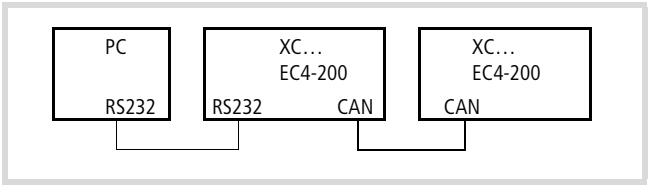


Abbildung 82: Programm-Download per Routing

Routing über XC200

Führen Sie bei einer Verbindung zwischen XC200 und PC über TCP/IP einen Programmtransfer oder ein Routing aus, müssen Sie die Blockgröße der zu übertragenen Daten einstellen. Die Blockgröße (4 kByte oder 128 kByte) ist abhängig von der Art des Transfers (Programmtransfer oder Routing) und des Betriebssystems, → Tabelle 16.

Tabelle 16:Blockgröße für die Datenübertragung

	Programm-/Datei-Transfer		**Routing**	
	BTS < V1.03.03	**BTS ≧ V1.03.03**	**BTS < V1.03.03**	**BTS ≧ V1.03.03**
Blockgröße Default: 128 kByte	128 kByte	4/128 kByte	Routing nicht möglich	4 kByte

Achtung!
Der Programmdownload mit einer Blockgröße von 4 kByte auf eine Steuerung mit dem Betriebssystem < V1.03.03 führt zu einem Fehlverhalten!

Wird ein Programm-Download vorgenommen, ändert sich die Fortschrittsanzeige auf dem Bildschirm des Programmiergeräts nur sprunghaft (ca. alle 10 Sekunden).

Das Routing mit der XC200 ist ab der BTS-Version V1.03.03 möglich.

Die Einstellung der Blockgröße (Änderung des Wertes in der Registry) wird im Folgenden beschrieben:

→ Diese Einstellung kann nur mit Administratorrechten durchgeführt werden (Zugriff auf die Registry)!

Einstellung der Blockgröße:

▶ Schließen Sie alle Anwendungen.
▶ Schließen Sie den CoDeSys Gateway Server.

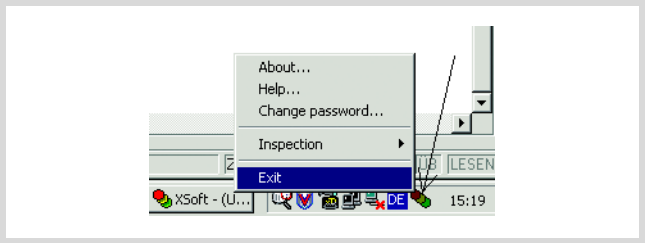


Abbildung 83: CoDeSys Gateway Server schließen

▶ Ändern Sie die Blockgröße auf den gewünschten Wert.

Rufen Sie die Anwendung BlockSizeEditor.exe im Verzeichnis „easy Soft CoDeSys" der Programmiersoftware auf und wählen Sie die Blockgröße aus.

Alternative Möglichkeit:

Zum Eintrag der Blockgröße in die Registry stehen im Installationsverzeichnis folgende *.reg-Dateien zur Verfügung:

BlockSizeDefault.reg	Trägt eine Blockgröße (Defaultwert) von 20000_{hex} = 128 kByte in die Registry ein.
BlockSizeRout.reg	Trägt eine Blockgröße von 1000_{hex} = 4 kByte in die Registry ein.

Die Download-Blockgröße wird über den folgenden Registry-Eintrag eingestellt:

```
[HKEY_LOCAL_MACHINE\SOFTWARE\3S-Smart Software Solutions GmbH\Gateway Server\Drivers\Standard\Settings\Tcp/Ip (Level 2 Route)]
"Blocksize" = dword:00020000
```

Die Default-Blockgröße beträgt 20000_{hex} (= 128 kByte), die Blockgröße für das Routing beträgt 1000_{hex} (= 4 kByte).

Hinweise zum Routing

- Werden größere Dateien in die Zielsteuerung geschrieben bzw. aus der Steuerung gelesen, kann es nach Abschluss des Übertragungsvorgangs zu einer Unterbrechung der Online-Verbindung kommen. Ein erneutes Ankoppeln ist möglich.
- Wird ein Programm mit geänderter Routing-Node-Id über eine routende Steuerung in die Zielsteuerung geladen, übernimmt die Zielsteuerung die geänderte Routing-Node-Id; die Kommunikationsverbindung wird jedoch abgebrochen. Ein Wiederankoppeln mit korrigierter Routing-Node-Id ist möglich.
- Enthält eine Steuerung ein Programm ohne gültige Routing-Parameter (Baudrate/Node-Id), kann auf diese Steuerung nicht über eine Routing-Verbindung angekoppelt werden."
- Das Routing ist unabhängig von der Konfiguration (Master/Device): Es ist möglich, auf eine Zielsteuerung zuzugreifen, die weder als Master noch als Device konfiguriert wurde. Sie muss lediglich die Grundparameter, wie Node-Id und Baudrate, sowie ein einfaches Programm enthalten.

Einstellung der Node-ID/Routing-ID

Steuerungen am CAN-Bus können als Master oder als Device konfiguriert werden. Zur eindeutigen Identifikation bei der Basiskommunikation erhalten die Steuerungen eine Node-Id/Knotennummer (Adresse). Möchten Sie mit Hilfe der Routing-Funktion auf eine (Ziel-)Steuerung zugreifen, müssen Sie der Ziel-Steuerung eine weitere (Routing-)ID zuordnen. Als Verbindung zwischen PC und z. B. EC4-200 kann die RS232-Schnittstelle oder die Ethernet-Schnittstelle genutzt werden.

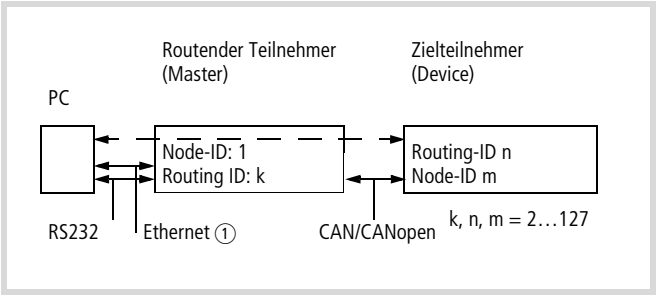


Abbildung 84: Routing über Teilnehmer mit ID1 (XC…, EC4-200, MFD4)

① Ethernet-Verbindung bei XC200, MFD4 und EC4-222… möglich

Tabelle 17:Beispiel zur Einstellung der Node-ID, Routing-ID, Baudrate

Steuerung	Funktion	Node-ID Routing-ID	Baud-rate	→ Abbildung
Routende Steuerung	Master	1 (Basis) 127 (Routing)	125 kB	86 85
Zielsteuerung	Device	3 (Basis)	125 kB	87
		54 (Routing)		85

→ Für Device-Teilnehmer gilt: Die Routing-ID muss ungleich der Node-ID (Basiskommunikation) sein!

Ausnahme XC100 mit Betriebssystem f V2.0:
Hier mussdie Routing-ID gleich der Node-ID sein!

Einstellung des Master-Teilnehmers

Im Master-Teilnehmer legen Sie zwei Node-ID fest:

- eine ID zur Routing-Funktion
- eine ID zur Basiskommunikation

zur Routing-Funktion:

Stellen Sie die Routing-ID und die CAN-Baudrate im Register „CAN/easyNet" des Ordners „Konfiguration" wie in Abbildung 85 ein.

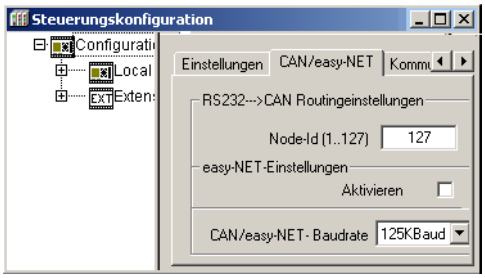


Abbildung 85: CAN-Master-Routingeinstellungen

zur Basiskommunikation:

Die ID zur Basiskommunikation und die CAN-Baudrate legen Sie im Ordner „CanMaster" im Register „CAN Parameter" wie in Abbildung 86 fest.

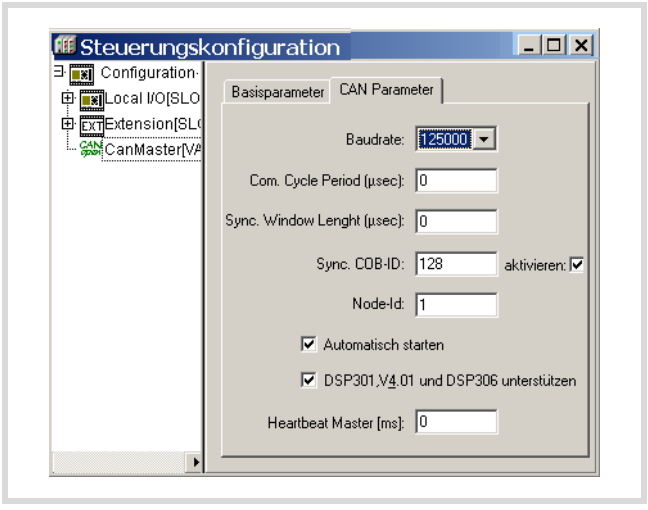


Abbildung 86: CAN-Master, ID zur Basiskommunikation

Einstellung des Device-Teilnehmers

Im Device- (Ziel-) Teilnehmer legen Sie zwei Node-ID fest:

- eine ID zur Routing-Funktion: Die Routing-ID und die CAN-Baudrate stellen Sie wie in Abbildung 85 im Register „CAN/easyNet" ein. Setzen Sie als Node-ID z.B. „54" ein.
- eine ID zur Basiskommunikation: Die ID zur Basiskommunikation und die CAN-Baudrate legen Sie im Ordner „CAN-Einstellungen", wie in Abbildung 87, fest.

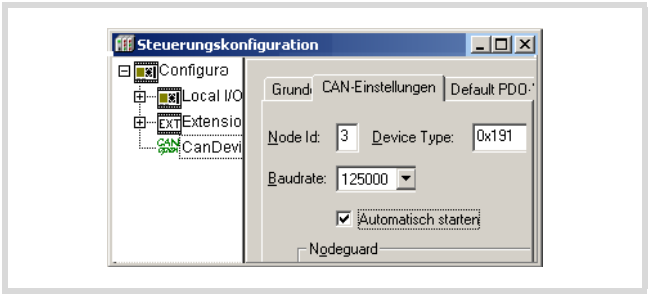


Abbildung 87: CAN-Deviceparameter

Node-ID und Baudrate werden mit dem Projektdownload in die Steuerung übertragen.

Beispiel: Zugriff auf ein Steuerungsprogramm
Im folgenden Beispiel wird die Vorgehensweise eines Zugriffs auf ein Steuerungsprogramm erklärt.

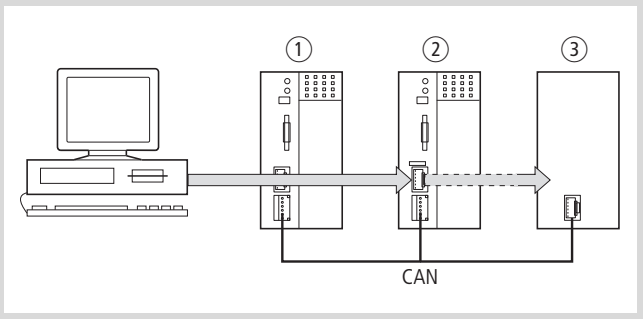


Abbildung 88: Diagnosemöglichkeiten

① XC100 mit Node-ID 1
② XC200 mit Node-ID 2, Routing-ID 127
③ Steuerung mit Node-Id 3 und Routing-ID 54 (z. B. XC100, XC200, XC121, EC4-200)

Den PC haben Sie an die Steuerung mit der Node-ID 2 angeschlossen und Sie wollen auf die Zielsteuerung mit der Routing-ID 54 zugreifen.

- ▶ Öffnen Sie das Projekt der Zielsteuerung (Node-Id 3), deren Programm Sie bearbeiten oder testen wollen.
- ▶ Parametrieren Sie zunächst die Hardware-Verbindung PC ↔ Steuerung (Node-Id 2).
- ▶ Wählen Sie im Online-Menü „Kommunikationsparameter" an.
- ▶ Klicken Sie unter Kanäle „lokal" die Taste „Neu" an.

Es erscheint das Fenster „Neuer Kanal".

- ▶ Wählen Sie im Fenster „Gerät" den Kanal aus: Serial [RS232] [Level 2 Route] oder TCP/Ip [Level 2 Route]
- ▶ Im Feld „Name" können Sie einen neuen Namen vergeben, z. B. „Rout_232".

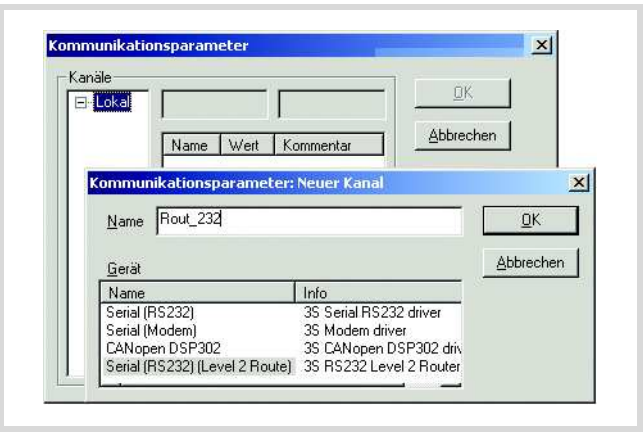


Abbildung 89: Kanal parametrieren

Sie haben nun die Parameter für die Hardware-Verbindung zwischen PC und Steuerung (Node-ID 2) festgelegt.

- ▶ Tragen Sie die Target-Id des Zielteilnehmers (im Beispiel die Zahl 54) ein. Die Target-Id ist identisch mit der Routing-ID! Zur Eingabe der Target-Id klicken Sie auf das Feld in der Spalte „Wert" rechts neben dem Begriff Target-Id. Geben Sie dort die Zahl 54 ein und bestätigen Sie mit OK.
- ▶ Loggen Sie sich ein und führen Sie die Aktion aus.

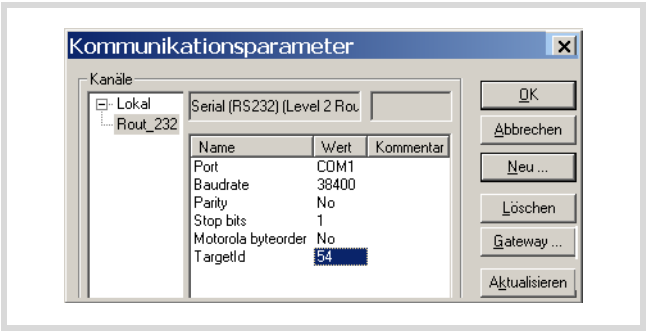


Abbildung 90: Target-Id der Zielsteuerung einstellen

Steuerungskombinationen zum Routing

Folgende Steuerungen unterstützen das Routing:

Von → Nach ↓	XC100	XC121	XC200	EC4-200	MFD4
XC100	×	×	×	×	×
XC121	×	×	×	×	×
XC200	×	×	×	×	×
MFD4	×	×	×	×	×
EC4-200	×	×	×	×	×

15 RS232-Schnittstelle im Transparent-Modus

Im Transparent-Modus erfolgt der Datenaustausch zwischen der EC4-200 und Datenendgeräten (z. B. Terminals, Drucker, PCs, Messgeräte) ohne Interpretation der Daten. Dazu ist die serielle Schnittstelle RS 232 (COM1/COM2 = Multifunktions-Schnittstelle) per Anwenderprogramm in den Transparent-Modus zu schalten.

Zum Ausführen des Transparent-Modus stehen Funktionen zum Öffnen und Schließen der Schnittstelle, zum Senden und Empfangen von Daten und zum Setzen der Schnittstellenparameter zur Verfügung. Nach dem Öffnen arbeitet die Schnittstelle mit den aktuellen Kommunikationsparametern, die Sie durch den Aufruf der Funktion "SysComSetSettings„ anpassen können.

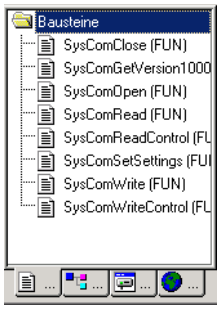


Abbildung 91: Funktionsübersicht

Die Funktionen des Transparent-Modus sind in der Bibliothek „EC_SysLibCom.lib" enthalten. Die Bibliothek muss daher in den Bibliotheksverwalter eingebunden sein. Eine Beschreibung der Funktionen finden Sie im Handbuch „Funktionsbausteine" (MN05002002Z-DE; frühere Bezeichnung AWB2786-1452D).

→ Befindet sich die RS232-Schnittstelle (COM1) im Transparent-Modus, ist eine Programmierung über diese Schnittstelle nicht möglich. Der Transparent-Modus muss erst geschlossen werden. Mit Schließen des Transparent-Modus werden die ursprünglichen Kommunikationsparameter wieder initialisiert.

COM1/COM2:

Das Schließen des Transparent-Modus erfolgt zwangsweise bei einem SPS-Zustandswechsel in den STOP oder mit dem Aufruf der Funktion „SysComClose".

16 Interaktives Display

Mit Hilfe von Funktionen und Funktionsbausteinen (FB) können Sie Variablen (Texte/Werte) auf dem Display der Steuerung darstellen und über die Cursortasten (Wippe) Werte eingeben. Zur externen Anzeige/Bedienung schließen Sie ein MFD-CP4 an die Steuerung an, das diese Funktionen parallel ausführt.

Darstellungsform (Anzeige)

Das Display der Steuerung und des MFD-CP4 hat ein Raster von 4 Zeilen mit 16 Spalten. Je Zeile können 16 Zeichen dargestellt werden. Es stehen Ihnen 3 Zeichensätze zur Verfügung.

Auf einer Seite des Displays können max. 12 Variable dargestellt werden.

Die Darstellungslänge bzw. Anzahl der Zeichen ist vom Datentyp abhängig. Enthält die Variable einen Wert, der mit einem Komma dargestellt wird, ist für das Komma eine weitere Stelle zu berücksichtigen. Der Wert der Variable kann kontinuierlich aktualisiert werden. Die Eingabe eines Wertes erfolgt über die Cursortasten (wippe).

Datentyp	min. / max. Wert	max. Stellen	max. Genauigkeit	Format
BYTE	255	4	2	1.23
WORD	65 535	6	4	1.2345
DWORD	4 294 967 295	11	9	1.234567890
USINT	255	4	2	1.23
UINT	65 535	6	4	1.2345
UDINT	4 294 967 295	11	9	1.234567890
SINT	–128/127	5	2	–1.23
INT	–32768/32767	7	4	–1.2345
DINT	–2 147 483 648/ +2 147 483 647	12	9	–1.234567890

Umschaltung zwischen Statusanzeige und Ein-/Ausgabemodus

Im Ursprungszustand zeigt das Display den Status der Steuerung an. Zur Ausgabe von anwendungsspezifischen Texte/Variablen oder Eingaben von Werten/Variablen ist der Ein-/Ausgabe-Modus der Steuerung aktiv zu schalten. Zum besseren Verständnis stellen Sie sich zwei interne Displays in der Steuerung vor, deren Anzeige kontinuierlich aufgefrischt wird. Das Erste dient zur Anzeige des Status und des Steuerungsmenüs. Das Zweite zur Anzeige von Texten und Variablen im Ein-/Ausgabemodus.

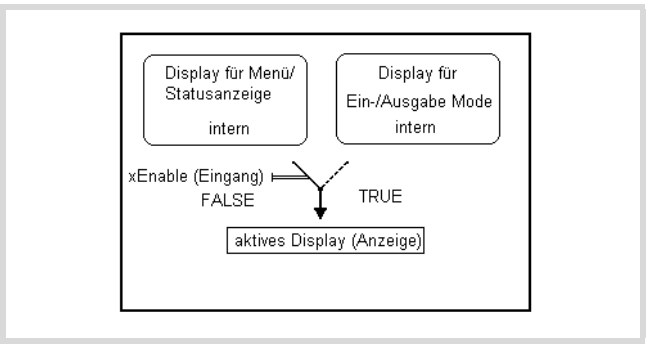


Abbildung 92: Umschaltung Statusanzeige ↔ Ein-/Ausgabe-Modus

Zur Aktivierung des Ein-/Ausgabe-Modus muss im Programm die Funktion „Disp_EnableDisplay" (→ Seite 77) aus der Bibliothek „EC_Visu2.lib" kontinuierlich abgearbeitet werden. Der Zustand am Funktionseingang „xEnable" bestimmt den Modus (→ Abbildung 92):

FALSE: Statusanzeige der Steuerung

TRUE: Ein-/Ausgabe-Modus

Im Ein-/Ausgabe-Modus zeigt das Display die Werte an, die im Anwenderprogramm erzeugt werden. Das Programm aktualisiert kontinuierlich die Werte und nimmt Eingaben über die (Wippen) Tasten an. Mit Hilfe von Funktionen und Funktionsbausteinen geben Sie die Form der Variablen, die Darstellung und die Positionierung auf dem Display an:

Mit dem Funktionsbaustein „Disp_DisplayElement" stellen Sie eine Variable auf dem Display dar.

Mit dem Funktionsbaustein „Disp_DisplayPage" wird eine Seite mit 8 Variablen dargestellt.

Funktionsbaustein „Disp_DisplayPage"
Für jede Variable legen Sie die Verwendung fest, wie z. B. Textanzeige oder Werteingabe.

Dazu sind die Eingänge des Funktionsbausteins zu parametrieren. Der Funktionsbaustein „Disp_DisplayPage" unterstützt die Cursorsteuerung. Werden mehrere Variable dargestellt, die eine Werteingabe erfordern, wird die erste Position zur Eingabe durch einen Cursor gekennzeichnet. Nach Abschluss der Eingabe springt der Cursor zur nächsten Position. Eine Anwendung mit der Darstellung von mehreren Seiten sowie der Aufruf einer Seite kann im Anwenderprogramm realisiert werden. Benutzen Sie dazu die Cursortasten P1, P2, P3, P4 und die Funktionstasten ESC, DEL, ALT und OK, deren Zustand Sie generell im Programm abfragen können.

Informationen über die Menüanzeige, aktuelle Cursorposition und Tastenzustand zeigen die Ausgänge des Funktionsbausteins „Disp_GetDisplayInfo" an.

Mit Hilfe der Funktionsbausteine „Disp_DisplayElement" und „Disp_DisplayPage" definieren Sie Elemente. Der Begriff „Elemente" bezieht sich auf die Funktionsbausteine. Ein Element ist eine Variable, die über zusätzliche Eigenschaften verfügt, wie z. B. die Positionierung auf dem Display. Die Zusatzeigenschaften werden durch die Parametrierung der Funktionsbaustein-Eingänge definiert.

Weitere Funktionen gehen aus der Funktion-/Funktionsbaustein-Übersicht hervor.

Funktions-/Funktionsbaustein-Übersicht

Die Anzeige im Ein-/Ausgabemode können Sie mit folgenden Funktionen/Funktionsbausteinen, die Bestandteil der Bibliothek „EC_Visu2.lib" sind, erstellen bzw. beeinflussen.

Funktion/ Funktionsbaustein	Beschreibung
Disp_SetBacklight	Hintergrundbeleuchtung des Displays auf der Steuerung ein/aus
Disp_RegisterVariable	Variablen registrieren
Disp_CreateVariableList	Länge der Liste für Variablen festlegen
Disp_GetDisplayInfo	Display Informationen abfragen
Disp_ClearScreen	Anzeige löschen
Disp_SetCursor	Position und Typ des Cursors festlegen
Disp_SetContrast	Kontrast des lokalen Displays definieren
Disp_DisplayElement	Anzeige eines einzelnen Elements /Variable
Disp_DisplayPage	Anzeige einer Seite mit max.12 Elementen für Texte, Werte sowie die Eingabe mehrerer Werte

Aus der folgenden Tabelle geht hervor, auf welche Anzeige (Steuerungsdisplay oder MFD-CP4) sich die Funktionen/Funktionsbausteine auswirken.

Funktion/ Funktionsbaustein	Typ	Anzeige auf Display von …
Disp_SetBacklight	Funktion	Steuerung
Disp_RegisterVariable	Funktion	–
Disp_CreateVariableList	Funktion	–
Disp_GetDisplayInfo	Funktionsbaustein	–
Disp_ClearScreen	Funktion	Steuerung, MFD[1)]
Disp_SetCursor	Funktionsbaustein	Steuerung, MFD[1)]
Disp_SetContrast	Funktionsbaustein	Steuerung
Disp_DisplayElement	Funktionsbaustein	Steuerung, MFD[1)]
Disp_DisplayPage	Funktionsbaustein	Steuerung, MFD[1)]

1) nur ausführbar, wenn Funktion Disp_EnableDisplay aktiv

Beschreibung wichtiger Funktionen/Funktionsbausteine

FUNCTION Disp_EnableDisplay: BOOL (*Umschaltung Statusanzeige <-> Ein-/Ausgabemode*)

```
VAR_INPUT
xEnable:         (* FALSE: Statusanzeige, TRUE: Ein-/Ausgabemode*)
xDisableESCKey:  (*Freischaltung der ESC-Taste am lokalen Display und MFD-CP4:
                       FALSE: Freischaltung
                       TRUE: Taste gesperrt *)
END_VAR
(* Returnwert: TRUE *)
```

Zu xDisableESCKey:

Betätigen Sie im Ein-/Ausgabe-Modus die ESC-Taste (Voraussetzung: Die ESC-Taste ist freigeschaltet), wird zur Statusanzeige umgeschaltet. Die ESC-Taste können Sie sperren, indem Sie am Eingang xDisableESCKey TRUE anlegen.

Wird am Eingang „Enable" wieder FALSE angelegt, wird die Statusanzeige der Steuerung wieder angezeigt.

FUNCTION Disp_RegisterVariable : BOOL (* Eine IEC-Variable als Display-Variable definieren *)

```
VAR_INPUT
  sName:      (* Symolischer Name der Display-Variable *)
  dwAddress:  (* Adresse der zugehörigen IEC-Variable *)
  eVarTyp:    (* Datentyp der zugehörigen IEC-Variable , siehe DISP_VARTYP*)
END_VAR
(* Returnwerte:*)
(* TRUE: Display-Variable erfolgreich registriert*)
(* FALSE: Variable-Liste voll *)

TYPE DISP_VARTYP :
( DISP_TYP_USINT := 0,
  DISP_TYP_UINT,
  DISP_TYP_UDINT,
  DISP_TYP_SINT,
  DISP_TYP_INT,
  DISP_TYP_DINT,
  DISP_TYP_BYTE,
  DISP_TYP_WORD,
  DISP_TYP_DWORD,
  DISP_TYP_STRING ) := DISP_TYP_UINT;
END_TYPE
```

Es können 50 Variable verwendet werden. Benötigen Sie mehr, ist dies über die Funktion „Disp_CreateVariableList" anzugeben.

FUNCTION_BLOCK Disp_GetDisplayInfo (* aktuelle Informationen des Displaystatus *)

```
VAR_OUTPUT
byMenueLevel :(*    Menue-Level:                                                          *)
              (*0:  Status-Menu                                                           *)
              (*1:  Haupt-Menu                                                            *)
              (*2:  Haupt -Menu / Programm                                                *)
              (*3:  Haupt -Menu / Set Clock                                               *)
              (*4:  Haupt -Menu / Information                                             *)
              (*5:  Sonder-Menu                                                           *)
              (*6:  Sonder-Menu/ Sicherheit                                               *)
              (*7:  Sonder-Menu/ System                                                   *)
              (*8:  Sonder-Menu/ Start Parameter                                          *)
              (*9:  Sonder-Menu / Menu-Sprache                                            *)
              (*10: Sonder-Menu / Konfiguration                                           *)
              (*11- 14: Nicht verwendet*)
              (*15: Ein-/Ausgabe-Modus                                                    *)
byActualLine:       (*Cursor position, Zeile 1 - 4                                       *)
byActualColumn:     (* Cursor position, Spalte 1 - 16                                    *)
xESCKeyDisabled:    (* FALSE: ESC-Taste drücken -> Status-Menü                           *)
                    (* TRUE: ESC-Taste kann im Anwenderprogramm abgefragt werden         *)
xInputEnabled #)    (* TRUE: wenn Eingänge xEnable und xEnableInput des FB Disp_DisplayPage = TRUE *)
                    (* FALSE: Ein Eingang disabled                                       *)
xInputActive #)     wenn Eingänge xEnable und xEnableInput des FB Disp_DisplayPage = TRUE und die ALT-Taste betätigt wird*)
                    (* FALSE:Eingabe nicht aktiv *)
(* #) Bei Verwendung von FB Disp_DisplayPage *)
END_VAR
```

FUNCTION_BLOCK Disp_DisplayElement (*Anzeige eines einzelnen Elements*)

```
VAR_INPUT
  xEnable:          (* Ausführung wenn Eingang = TRUE *)
  sName          :  (* Symbolischer Element Name *)
  byLine         :  (* Element anzeigen in Zeile 1 - 4 *)
  byColumn       :  (* Element anzeigen in Spalte 1 - 16 *)
  eFont          :  (* Font, nur Elemente vom Typ STRING ! Siehe DISP_FONTS*)
  byDigits       :  (* Anzahl der Zeichen, nur für numerische Elemente*)
  byPrecision    :  (* Anzahl der Zeichen nach Dezimalpunkt, nur für numerische Elemente *)
  eAttribut      :  (* Element Eigenschaften normal, reverse, blinken. Siehe DISP_ATTRIBUT*)
END_VAR
VAR_OUTPUT
  eError            (* Siehe DISP_ERROR*)
END_VAR
(* Returnwerte:*)
(* DISP_ERROR_NO_ERROR:OK, kein Fehler*)
(* DISP_ERROR_INVALID_LINE: *)
(* DISP_ERROR_INVALID_COLUMN: außerhalb des Wertebereichs*)
(* DISP_ERROR_ELEMENT_NOT_FOUND: Element nicht gefunden*)
(* DISP_ERROR_INVALID_VARIABLE_TYP:außerhalb des Wertebereichs*)

TYPE DISP_FONTS :
( DISP_FONT_LATIN1 := 0,
  DISP_FONT_LATIN2,
  DISP_FONT_CYRILLIC ) := DISP_FONT_LATIN1;
END_TYPE
...
```

```
...
TYPE DISP_ATTRIBUT :
( DISP_ATTR_NORMAL := 0,
  DISP_ATTR_REVERSE,
  DISP_ATTR_BLINK ) := DISP_ATTR_NORMAL;
END_TYPE
```

FUNCTION_BLOCK Disp_DisplayPage (* Anzeige einer Seite *)

```
VAR_INPUT
xEnable:                          (* TRUE: Anzeige aktivieren *)
xEnableInput:                     (* TRUE: Eingabe aktivieren*)
byNoOfElements:                   (* Anzahl der Elemente für diese Seite 1 - 12*)
aElementDescription:ARRAY [1..12]  (* siehe TYPE DISP_ElementDescription*)
OF DISP_ElementDescription:
END_VAR
VAR_OUTPUT
byError
END_VAR
(* Returnwerte :                  *
(* 0:         OK, alle Elemente werden angezeigt*)
(* 1 - 12: Fehler bei der Anzeige des Elements „n" oder siehe DISP_ERROR_INVALID_NO_OF_ELEMENTS*)

TYPE DISP_ElementDescription :     (* Description of one display element *)
STRUCT
              xEnable           : (* TRUE Defaulteinstellung: Element wird angezeigt; FALSE: Anzeige wird eingefroren*)
              xInputEnable      : (* FALSE: Anzeige des Elements, siehe Abbildung 93; TRUE: Anzeige (Initialisierungs-)
                                  Wert, Eingabe möglich*)
              sName             : (* Symbolischer Element Name *)
              byLine            : (* Element anzeigen in Zeile 1 - 4 *)
              byColumn:           (* Element anzeigen in Spalte 1 - 16 *)
              eFont             : (* Font ,nur Elemente vom Typ STRING ! Siehe DISP_FONTS*)
              byDigits          : (* Anzahl der Zeichen, nur für numerische Elemente*)
              byPrecision       : (* Anzahl der Zeichen nach Dezimalpunkt, nur für numerische Elemente *)
              diMinInputValue#:   (* Min-Wert für Eingabewert, nur für numerische Elemente *)
              diMaxInputValue#:   (* Max-Wert für Eingabewert, nur für numerische Elemente *)
              eAttribut:          (* Element Eigenschaften normal, reverse, blinken. Siehe DISP_ATTRIBUT*)
              xInputActiv#:       (* TRUE: Wenn Eingänge xEnable und xEnableInput des FB Disp_DisplayPage = TRUE*)
              xInputDone#:        (* TRUE: Nach Abschluß der Werteingabe mit Betätigung der
                                  Taste „OK". Ist vom Anwender wieder auf FALSE zu setzen!*)
              eError              (*Siehe DISP_ERROR*)
END_STRUCT
END_TYPE
END_VAR
                                  # aktiv, wenn xInputEnable = TRUE
(* Returnvalues: *)
(* DISP_ERROR_NO_ERROR,           OK, kein Fehler *)
(* DISP_ERROR_INVALID_LINE,       außerhalb des Wertebereichs: 1 - 4 *)
(* DISP_ERROR_INVALID_COLUMN,     außerhalb des Wertebereichs: 1- 16*)
(*DISP_ERROR_ELEMENT_NOT_FOUND,   Element nicht gefunden*)
```

Zusammenhang zwischen DISP_DisplayPage.xEnable/ xEnableInput und DISP_ElementDescription.xInputEnable für die Werteingabe

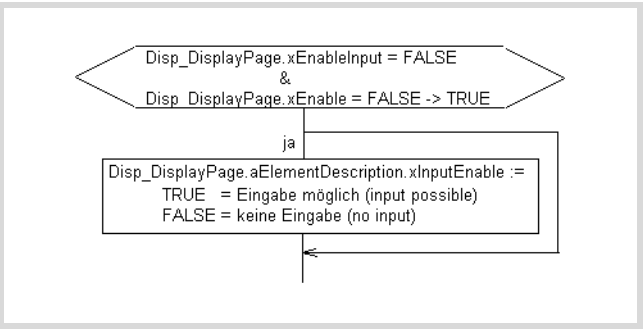


Abbildung 93: Anzeige/Werteingabe aktivieren

Ablauf zur Eingabe eines Wertes

▶ Setzen Sie im Programm:

```
Disp_DisplayPage.xEnable = TRUE       (Anzeige der
                                      Werte/Änderungen
                                      sichtbar)
Disp_DisplayPage.xEnableInput = TRUE  (Eingabe wird
                                      ermöglicht)
```

▶ Betätigen Sie die Taste ALT am Display.
Der Cursor erscheint auf dem ersten Element „aElementDescription[1]“ dessen xInputEnable auf TRUE gesetzt ist.

▶ Betätigen Sie die OK-Taste.
Der Wert wird in der Grundform des Datentyps dargestellt, z. B. der TYP UINT : 00000)

▶ Mit den Cursortasten können Sie den Wert verändern:
- Wählen Sie mit den Tasten > oder < die Stelle des Wertes an.
- Betätigen Sie die ∧"∨ Taste, um den Wert zu verändern.

▶ Bestätigen Sie die Eingabe mit der OK-Taste.
Der Cursor springt auf die nächste Eingabemöglichkeit, z. B. das zweite Element.

Mit der ALT-Taste gelangen Sie in den Ein-/Ausgabe-Modus zurück.

Allgemeiner Ablauf zur Programmerstellung

- Deklaration der (Display-)Variablen zur Anzeige Eingabe in der Liste „Globale_Variablen_Display“ → Abbildung 95
- Programmerstellung
  Es werden 3 Programme (für jedes Beispiel) erstellt:
  - Startprogram: Erzeugung eines Startimpuls (1. Zyklus)
  - PLC_PRG: Anwenderprogramm mit Aufruf des Programms „Visualisierung“
  - Visualisierung: Programm zur Darstellung der Variablen auf dem Display

Aufbau des Programms „Visualisierung“

- Im 1. Zyklus:
  - Anzahl der Display-Variablen festlegen -> Funktion Disp_CreateVariableList (nur falls mehr als 50 Display-Variablen benötigt werden!)
  - Registrieren der Display-Variablen -> Funktion Disp_RegisterVariable (generell ausführen)

- Im folgenden Zyklus (abhängig von der Anwendung):
  - Display löschen -> Funktion Disp_ClearScreen
  - Beleuchtung des lokalen Displays -> Funktion Disp_SetBacklight
  - Cursor setzen -> Funktion Disp_SetCursor
  - Kontrast des lokalen Displays definieren-> Funktion Disp_SetContrast
  - Eigenschaften, wie Position, der Variablen festlegen-> FB Disp_DisplayElement oder FB Disp_DisplayPage

- Alle Zyklen:
  - Start der Displayanzeige -> Start des FB Disp_DisplayElement oder FB Disp_DisplayPage
  - Anzeige aktiv schalten-> Start der Funktion Disp_DisplayEnable
  - Abfrage der Displayzustände -> Funktion Disp_GetDisplayInfo

Beispiel zur Ausgabe von Texten und Werten

(Mit dem FB Disp_DisplayElement) Das Display soll die Werte der Variablen „motor1“ und „motor2“ anzeigen. Beide Werte werden vom Anwenderprogramm kontinuierlich verändert.

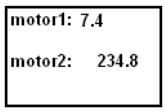


Abbildung 94: Beispiel zur Ausgabe von Texten und Werten

Aktionen über die Steuerungseingänge ausführen

- I1 = FALSE: Statusanzeige
- I1 = TRUE: Ein-/Ausgabe-Modus
- I2 = FALSE: ESC-Taste aktiv
- I2 = TRUE: ESC-Taste gesperrt
- I3 = TRUE: Die erste Zeile wird auf dem Display dargestellt.
- I5 = TRUE: Die dritte Zeile wird auf dem Display dargestellt.

Ausführung

Das Beispielprogramm besteht aus den Programmen:

- STARTPROGRAM
  - Das Startprogramm wird beim Systemereignis „Start“ aufgerufen.
  - Die Hilfsvariable g_xFirstCycleAfterStartProgram wird gesetzt.

- PLC_PRG
  - 2 Werte werden hochgezählt.
  - Das Programm „Visualisierung“ wird aufgerufen.

- VISUALISIERUNG
  - Registrierung und Positionierung der Variablen auf dem Display im ersten Zyklus
  - Die Hilfsvariable g_xFirstCycleAfterStartProgram wird zurückgesetzt.
  - Einschalten des Ein-/Ausgabe-Modus (I1)
    Anzeige starten (I3,I5)

Variablen deklarieren

▶ Deklarieren Sie zunächst für jedes Textelement, das Sie anzeigen möchten, wie z. B. „motor1“, in der Liste „Globale_Variablen_Display“ eine Variable vom Typ „String“ nach folgendem Beispiel (siehe auch Abbildung 95):

```
VAR GLOBAL
            g_sDisp_String1 :STRING:='Motor1';
END_VAR
```

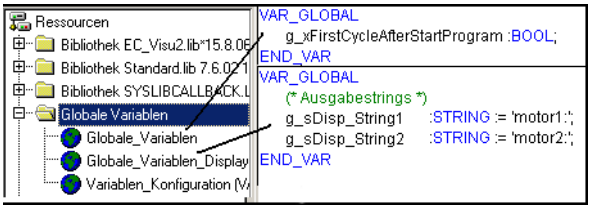


Abbildung 95: Deklaration der Display-Variablen

Hilfsvariablen erstellen

▶ Für den ersten Programmzyklus rufen Sie das Programm „Startprogram“ mit dem Systemereignis „Start“ auf.

▶ Setzen Sie in diesem Programm eine Hilfsvariable „g_xFirstCycleAfterStartProgram“, die Sie nach Beendigung des ersten Zyklus zurücksetzen. Die Hilfsvariable ist global zu deklarieren → Abbildung 95.

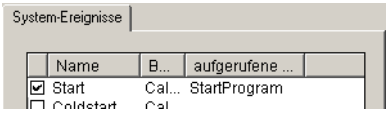


Abbildung 96: Systemereignis definieren

Programm „StartProgram“ erstellen

▶ Schreiben Sie das Programm „StartProgram“ wie in → Abbildung 97.

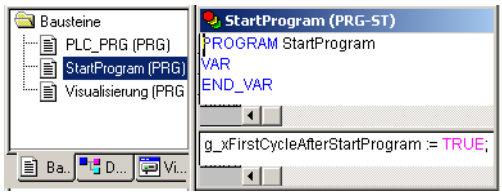


Abbildung 97: Startprogramm erstellen

Programm „PLC_PRG“ erstellen

```
PROGRAM PLC_PRG
VAR
  fbTimer1                                :TON;
  (* Anzeigewerte der Applikation *)
  byValue                                 :BYTE;
  wValue                                  :WORD;
END_VAR
-----------------------------------------------------------
fbTimer1(IN:=NOT fbTimer1.Q , PT:=t#50ms );
IF fbTimer1.Q = TRUE THEN
  byValue := byValue + 1;
  wValue := wValue + 1;
END_IF

Visualisierung(); (* Visualisierung aufrufen *)
```

Programm „Visualisierung“ erstellen

In Abhängigkeit der Hilfsvariablen „g_xFirstCycleAfterStartProgram“ registrieren Sie die Variablen, deren Text/Wert angezeigt werden soll:

- ▶ Programmieren Sie dazu die Funktion „Disp_RegisterVariable“ mit folgenden Beispiel-Parametern: Disp_RegisterVariable ('S1',ADR(g_sDisp_String1), Disp_TYP_STRING). Damit wird der Variablen der Name S1 zugeordnet.
- ▶ Zur Anzeige eines Wertes (Typ Byte) programmieren Sie eine Variable mit dem Funktionsaufruf Disp_RegisterVariable ('V1', ADR(byValue), Disp_TYP_BYTE).

```
FUNCTION Disp_RegisterVariable : BOOL
(* Register one IEC-Variable for using as display variable *)
VAR_INPUT                   (* Symbolic name for display variable *)
    sName      :STRING(16);  (* Address of corresponding IEC-variable *)
    dwAddress  :DWORD;       (* Datatyp of corresponding IEC-variable *)
    eVarTyp    :DISP_VARTYP; (* Returnvalue *)
END_VAR
```

Abbildung 98: Funktion Disp_RegisterVariable

In diesem Programmteil können Sie auch den Standort der Variablen auf der Anzeige festlegen, indem Sie Zeile (Line) und Spalte (Column) angeben. Dazu rufen Sie den Funktionsbaustein (FB) „Disp_DisplayElement“ auf und parameterieren die Eingänge sName, byLine, und byColumn, z. B.:

```
fbDisplayElement1.sName := 'S1';
fbDisplayElement1.byLine := 1;
fbDisplayElement1.byColumn := 1;
```

Das Element S1 mit dem Text „motor1“ würde in der ersten Zeile ab der ersten Spalte angezeigt werden.

Zur Anzeige/Eingabe mehrerer Elemente rufen Sie in dem folgenden Programmteil, der kontinuierlich abgearbeitet wird, den Funktionsbaustein „Disp_DisplayElement“ mehrfach auf und ordnen den Eingängen „xEnable“ externe Eingänge – z. B. I3 – zu.

```
VAR
    xIsDisplayEnabled: BOOL;
    fbGetDisplayInfo:               Disp_GetDisplayInfo;
    fbDisplayElement1: Disp_DisplayElement;
    fbDisplayElement2: Disp_DisplayElement;
    fbDisplayElement3: Disp_DisplayElement;
    fbDisplayElement4: Disp_DisplayElement;
    byError:                        BYTE;
    byValue:                        BYTE;
    wValue:                         WORD;
END_VAR
------------------------------------------------------------
(* Initialisierung im ersten Zyklus nach Start des Programms *)
IF g_xFirstCycleAfterStartProgram = TRUE THEN
Disp_ClearScreen(xEnable:=TRUE);
Disp_RegisterVariable('S1', ADR(g_sDisp_String1),
DISP_TYP_STRING);
Disp_RegisterVariable('S2', ADR(g_sDisp_String2),
DISP_TYP_STRING);
Disp_RegisterVariable('V1', ADR(PLC_PRG.byValue),
DISP_TYP_BYTE);
Disp_RegisterVariable('V2', ADR(PLC_PRG.wValue),
DISP_TYP_WORD);

    fbDisplayElement1.sName          := 'S1';
    fbDisplayElement1.byLine         := 1;
    fbDisplayElement1.byColumn       := 1;

    fbDisplayElement2.sName          := 'S2';
    fbDisplayElement2.byLine         := 3;
    fbDisplayElement2.byColumn       := 1;

    fbDisplayElement3.sName          := 'V1';
    fbDisplayElement3.byLine         := 1;
    fbDisplayElement3.byColumn       := 8;
    fbDisplayElement3.byDigits       := 4;
    fbDisplayElement3.byPrecision    := 1;

    fbDisplayElement4.sName          := 'V2';
    fbDisplayElement4.byLine         := 3;
    fbDisplayElement4.byColumn       := 8;
    fbDisplayElement4.byDigits       := 6;
    fbDisplayElement4.byPrecision    := 1;

(* Der erste Zyklus ist beendet, Flag zurücksetzen *)

g_xFirstCycleAfterStartProgram := FALSE;
END_IF

xIsDisplayEnabled := Disp_EnableDisplay(I1, I2);
fbDisplayElement1( xEnable:= I3 );
fbDisplayElement2( xEnable:= I5 );
fbDisplayElement3( xEnable:= I3 );
fbDisplayElement4( xEnable:= I5);
```

- ▶ Starten Sie die Programme.

Beispiel zur Ausgabe einer Seite mit Texten und Eingabemöglichkeiten

(Mit dem Funktionsbaustein Disp_DisplayPage)

Es soll folgende Anzeige realisiert werden.

Die Inhalte der Variablen MO11 und TEMP8 werden vom Anwenderprogramm kontinuierlich verändert.

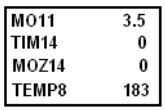


Abbildung 99: Beispiel einer Seite zur Ein- und Ausgabe

Aktionen über die Steuerungseingänge

- I1 = FALSE: Statusanzeige
- I1 = TRUE: Ein-/Ausgabe-Modus
- I2 = FALSE: ESC-Taste aktiv
- I2 = TRUE: ESC-Taste gesperrt
- I3 = TRUE: Die Werte werden vom Programm aktualisiert.
- I4 = TRUE: Eingabe aktiv.

Ausführung

Das Beispielprogramm besteht aus den Programmen:

- STARTPROGRAM: (wird beim Systemereignis „Start" aufgerufen)
  - Die Hilfsvariable „g_xFirstCycleAfterStartProgram" wird gesetzt.

- PLC_PRG:
  - 2 Werte werden hochgezählt
  - Das Programm „Visualisierung" wird aufgerufen.

- VISUALISIERUNG
  - Registrierung und Positionierung der Variablen auf dem Display im ersten Zyklus.
  - Die Hilfsvariable g_xFirstCycleAfterStartProgram wird zurückgesetzt.
  - Einschalten des Ein-/Ausgabe-Modus (I1).
  - Freigabe ESC-Taste (I2).
  - Anzeige starten (I3).
  - Eingabe starten (I4).

Display-Variablen deklarieren

▶ Deklarieren Sie zunächst für jedes Textelelement, das Sie anzeigen möchten, wie z. B. „MO11", im Ordner „Globale_Variablen_Display" eine Variable vom Typ String nach folgendem Beispiel:

```
VAR_GLOBAL
  g_sDisp_String1    :STRING := 'MO11 :';
  g_sDisp_String2    :STRING := 'TIM14 :';
  g_sDisp_String3    :STRING := 'MOZ14 :';
  g_sDisp_String4    :STRING := 'TEMP8 :' ;
END_VAR
```

▶ Erstellen Sie eine Hilfsvariable und schreiben Sie das Programm „Startprogram" wie im „Beispiel zur Ausgabe von Texten und Werten".

▶ Schreiben Sie die Programme PLC_PRG und Visualisierung nach folgendem Beispiel:

```
PROGRAM PLC_PRG (**********************)
VAR
    fbTimer1       :TON;
    (* Anzeigewerte der Applikation *)
    byValue        :BYTE;
    wValue         :WORD;
    dwValue        :DWORD;
    usiValue       :USINT;
    siValue        :SINT;
END_VAR
-----------------------------------------------------------
fbTimer1(IN:=NOT fbTimer1.Q , PT:=t#50ms );
IF fbTimer1.Q = TRUE THEN
    usiValue := usiValue + 1;
    byValue:=byValue+1;
END_IF

Visualisierung(); (* Visualisierung aufrufen *)
```

```
PROGRAM Visualisierung (***************)
VAR
     xIsDisplayEnabled                  :BOOL;
     fbDisplayPage1                     :Disp_DisplayPage;
     byError                            :BYTE;
     siValue                            :SINT;
END_VAR
-------------------------------------------------------------
(* Initialisierung im 1. Zyklus nach Start des Programms *)
IF g_xFirstCycleAfterStartProgram = TRUE THEN

Disp_RegisterVariable('S1', ADR(g_sDisp_String1), DISP_TYP_STRING);
Disp_RegisterVariable('S2', ADR(g_sDisp_String2), DISP_TYP_STRING);
Disp_RegisterVariable('S3', ADR(g_sDisp_String3), DISP_TYP_STRING);
Disp_RegisterVariable('S4', ADR(g_sDisp_String4), DISP_TYP_STRING);
Disp_RegisterVariable('V1', ADR(PLC_PRG.byValue), DISP_TYP_BYTE);
Disp_RegisterVariable('V2', ADR(PLC_PRG.wValue), DISP_TYP_WORD);
Disp_RegisterVariable('V3',ADR(PLC_PRG.dwValue),
DISP_TYP_DWORD);
Disp_RegisterVariable('V4', ADR(PLC_PRG.usiValue), DISP_TYP_USINT);

fbDisplayPage1.aElementDescription[1].sName             := 'S1';
fbDisplayPage1.aElementDescription[1].byLine            := 1;
fbDisplayPage1.aElementDescription[1].byColumn          := 1;
fbDisplayPage1.aElementDescription[2].sName             := 'S2';
fbDisplayPage1.aElementDescription[2].byLine            := 2;
fbDisplayPage1.aElementDescription[2].byColumn          := 1;
fbDisplayPage1.aElementDescription[3].sName             := 'S3';
fbDisplayPage1.aElementDescription[3].byLine            := 3;
fbDisplayPage1.aElementDescription[3].byColumn          := 1;
fbDisplayPage1.aElementDescription[4].sName             := 'S4';
fbDisplayPage1.aElementDescription[4].byLine            := 4;
fbDisplayPage1.aElementDescription[4].byColumn          := 1;

fbDisplayPage1.aElementDescription[5].sName             := 'V1';
fbDisplayPage1.aElementDescription[5].byLine            := 1;
fbDisplayPage1.aElementDescription[5].byColumn          := 13;
fbDisplayPage1.aElementDescription[5].byDigits          := 4;
fbDisplayPage1.aElementDescription[5].byPrecision       := 1;
fbDisplayPage1.aElementDescription[5].xInputEnable      := FALSE;
fbDisplayPage1.aElementDescription[5].diMinInputValue   := 1;
fbDisplayPage1.aElementDescription[5].diMaxInputValue   := 100;
fbDisplayPage1.aElementDescription[6].sName             := 'V2';
fbDisplayPage1.aElementDescription[6].byLine            := 2;
fbDisplayPage1.aElementDescription[6].byColumn          := 12;
fbDisplayPage1.aElementDescription[6].byDigits          := 5;
fbDisplayPage1.aElementDescription[6].byPrecision       := 0;
fbDisplayPage1.aElementDescription[6].xInputEnable      := TRUE;
fbDisplayPage1.aElementDescription[6].diMinInputValue   := 0;
fbDisplayPage1.aElementDescription[6].diMaxInputValue   := 33333;
fbDisplayPage1.aElementDescription[7].sName             := 'V3';
```

```
fbDisplayPage1.aElementDescription[7].byLine            := 3;
fbDisplayPage1.aElementDescription[7].byColumn          := 8;
fbDisplayPage1.aElementDescription[7].byDigits          := 9;
fbDisplayPage1.aElementDescription[7].byPrecision       := 0;
fbDisplayPage1.aElementDescription[7].xInputEnable      := TRUE;
fbDisplayPage1.aElementDescription[7].diMinInputValue   := 0;
fbDisplayPage1.aElementDescription[7].diMaxInputValue   := 4444444;
fbDisplayPage1.aElementDescription[8].sName             := 'V4';
fbDisplayPage1.aElementDescription[8].byLine            := 4;
fbDisplayPage1.aElementDescription[8].byColumn          :=13;
fbDisplayPage1.aElementDescription[8].byDigits          := 4;
fbDisplayPage1.aElementDescription[8].byPrecision       := 0;
fbDisplayPage1.aElementDescription[8].xInputEnable      := TRUE;
fbDisplayPage1.aElementDescription[8].diMinInputValue   := 4;
fbDisplayPage1.aElementDescription[8].diMaxInputValue   := 400;

(* Der erste Zyklus ist beendet, Flag zurücksetzen *)
g_xFirstCycleAfterStartProgram := FALSE;
END_IF

xIsDisplayEnabled := Disp_EnableDisplay(I1, I2);
fbDisplayPage1( xEnable:= I3 , xEnableInput:= I4,
byNoOfElements:= 8, byError =>byError );

IF fbDisplayPage1.aElementDescription[7].xInputDone = TRUE THEN
siValue := PLC_PRG.siValue; (*Wert zur internen Verarbeitung*)
fbDisplayPage1.aElementDescription[7].xInputDone        := FALSE;
END_IF
```

▶ Starten Sie die Programme.

Multi-Funktions-Display MFD-CP4 an der EC4-200

Mit dem Multi-Funktions-Display (MFD-CP4) haben Sie die Möglichkeit, die gleichen Anzeige- und Bedienmöglichkeiten der Steuerung extern aufzubauen.

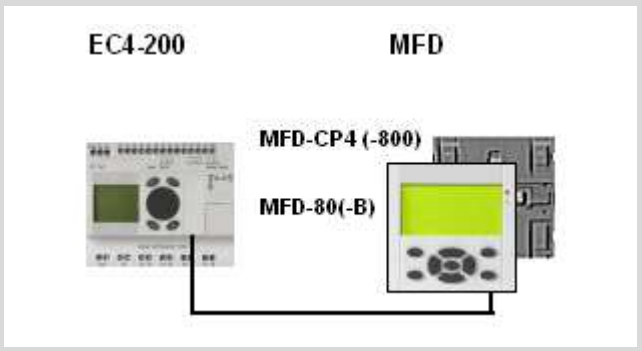


Abbildung 100: EC4-200 mit MFD-CP4

Wird das an die EC4-200 angeschlossene MFD-CP4 an Spannung gelegt, befindet es sich im Terminal-Modus. In diesem Modus übernimmt es die Information des Steuerungsdisplays und zeigt sie auf dem (MFD-)Display an.

Zum Einstellen der MFD-Parameter schalten Sie das MFD-CP4 in den Lokal-Modus. Die Parameter sind:

- Kontrast
- Beleuchtung
- Menüsprache: Anpassen der Parameter-Bezeichnung an die Landessprache
- COM-Schnittstelle
- ID=Identnummer 0, 1,..,8
  - 0: Das MFD-CP4 kommuniziert mit dem real verbundenen Gerät.
  - 1…8: ID der easyNet-Teilnehmer: Anwahl der Teilnehmer (-ID) am easyNet
    Ist die EC4-200 ein Teilnehmer am easyNet, kann das MFD-CP4 über die EC4-200 mit dem angewählten Teilnehmer kommunizieren.
- Baudrate: 9600 (19200) Baud.

Um zwischen dem Terminal-Modus und Lokal-Modus des MFD-CP4 zu wechseln, betätigen Sie die „*“-Taste (nur am MFD-CP4 vorhanden).

Der Wechsel in den Terminal-Modus kann nur aus dem Hauptmenü des Lokal-Modus erfolgen.

Siehe auch Handbuch zum MFD-CP4 (MN05013011Z-DE; frühere Bezeichnung AWB2528-1548D), Kapitel „Einstellungen“.

Hauptmenü: COM…
MENÜSPRACHE…
BELEUCHTUNG: 80%
KONTRAST: +1

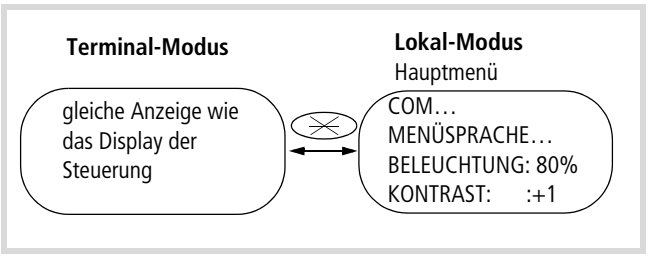


Abbildung 101: Umschaltung Terminal Modus ↔ Lokal-Modus

Im Lokal-Modus sind die MFD-Display-Tasten aktiv. Bedienungshandbuch zum Netzteil/Kommunikationsmodul MFD-CP4 (MN05013011Z-DE; frühere Bezeichnung AWB2528-1548D).

MFD-Aufbau

Das MFD-CP4 ist ein zusammengesetztes Gerät. Das eigentliche Display, die Anzeige- und Bedieneinheit MFD-80(-B), ist für den Einbau in die Frontseite einer Schaltschranktüre vorgesehen. Sie wird auf das rückseitig befestigte Netzteil/Kommunikationsmodul MFD-CP4(-800) aufgeschnappt. Die Verbindung mit der EC4-200 (Multifunktionsschnittstelle) stellen Sie über das Kabel MFD-CP4-800-CAB5 her.

Weitere Informationen über die Handhabung, den Anschluss und die Technischen Daten des Gerätes finden Sie in dem Bedienungshandbuch zum Netzteil/Kommunikationsmodul MFD-CP4 (MN05013011Z-DE; frühere Bezeichnung AWB2528-1548D).

17 EC4-200-Netzwerk-Kopplungen

Mit Hilfe der Netzwerk-Kopplungen EASY205-ASI, EASY221-CO, EASY204-DP, EASY222-DN können Sie die EC4-200 als Slave an die Netzwerke ASI, CAN, PROFIBUS-DP oder DeviceNet (→ Tabelle 18) ankoppeln. Zusätzlich kann die Steuerung Teilnehmer in einem easyNet-Netzwerk sein.

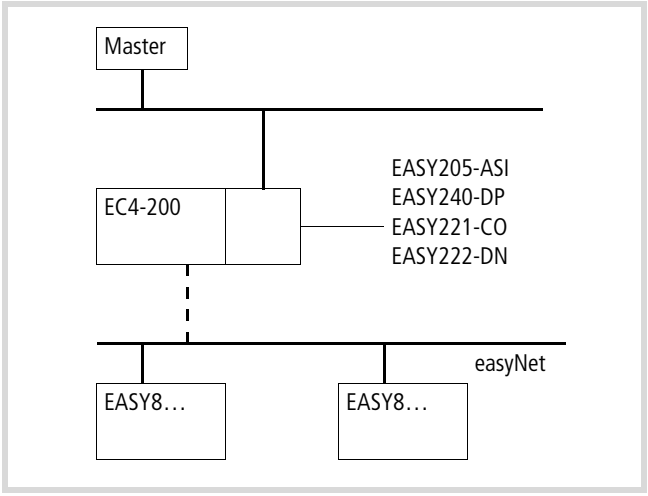


Abbildung 102: EC4-200 mit Netzwerk-Kopplung

Die Art des Datenaustauschs zwischen Master und den Netzwerk-Kopplungen geht aus Tabelle 18 hervor.

Tabelle 18:Übersicht der Netzwerk-Kopplungen

Netzwerk-Kopplung	Netzwerk	Datenaustausch
EASY205-ASI	ASi	Zyklisch
EASY204-DP	PROFIBUS-DP	Zyklisch + azyklisch
EASY221-CO	CANopen	Zyklisch + azyklisch
EASY222-DN	DeviceNet	Zyklisch + azyklisch

→ Die Netzwerkmodule werden im Zusammenhang mit der easy800 in separaten Handbüchern ausführlich beschrieben (→ Tabelle 19). Diese Handbücher treffen auch auf die EC4-200 in Verbindung mit den Netzwerk-Kopplungen zu, da sie sich genauso verhält wie eine easy800. In den folgenden Abschnitten zu den einzelnen Netzwerk-Kopplungen wird deshalb nur noch auf Abweichungen bzw. besonderes Verhalten hingewiesen.

Tabelle 19:Handbücher zu den Netzwerkmodulen

Typ	Handbuch (MN bzw. vormals AWB)
EASY204-DP	MN05013005Z-DE (frühere Bezeichnung AWB2528-1401D)
EASY221-CO	MN05013008Z-DE (frühere Bezeichnung AWB2528-1479D)
EASY222-DN	MN05013007Z-DE (frühere Bezeichnung AWB2528-1427D)

EASY205-ASI

Zyklischer Datenaustausch

Der Master sendet an die mit der EC4-200 verbundene Netzwerk-Kopplung EASY205-ASI 8 Bit: 4 Bit Output-Daten und 4 Bit Parameter. Von der EC4-200 empfängt er 4 Bit Input-Daten.

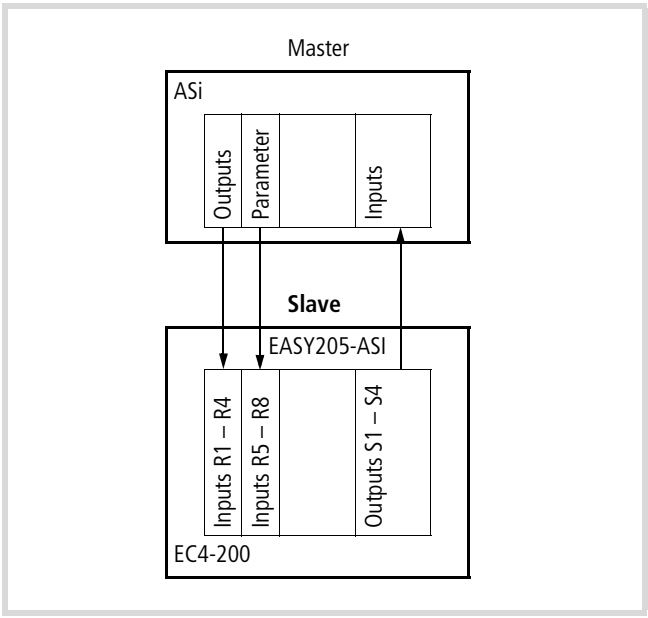


Abbildung 103: Zyklischer Datenaustausch der EASY205-ASI

Tabelle 20:Ein-/Ausgangsdaten der EC4-200

Master → EC4				
Master-Outputs	Q0	→	R1	EC4-Inputs
	Q1	→	R2	
	Q2	→	R3	
	Q3	→	R4	
Master-Parameter	P0	→	R5	EC4-Inputs
	P1	→	R6	
	P2	→	R7	
	P3	→	R8	

EC4 → Master				
EC4-Outputs	S1	→	I0	Master-Inputs
	S2	→	I1	
	S3	→	I2	
	S4	→	I3	

Konfiguration

Die Konfiguration erfolgt in der Steuerungskonfiguration der Programmiersoftware easySoft-CoDeSys. Die Netzwerk-Kopplung wird als Erweiterungsmodul in den Konfigurationsbaum eingetragen. Er besitzt vordefinierte Eingangs- und Ausgangs-Kanäle (R1…R8, S1…S4) zur zyklischen Übertragung der Daten.

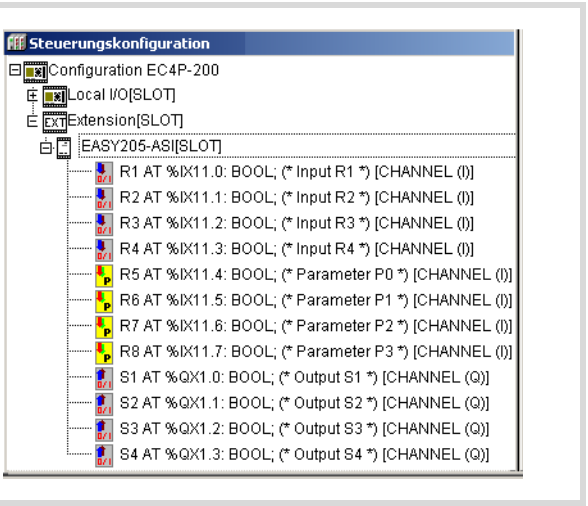


Abbildung 104: EASY205-ASI konfigurieren

Teilnehmer-Adresse einstellen

Die Teilnehmeradresse erhält das EASY205-ASI über ein externes Programmiergerät.

EASY221-CO, EASY204-DP, EASY222-DN

Das Verfahren des Datenaustauschs zwischen den EASY-Netzwerk-Kopplungen und einem Master wird in separaten Handbüchern ausführlich beschrieben, → Tabelle 19.

Zyklischer Datenaustausch

Das Prinzip des zyklischen Datenaustauschs ist für die Netzwerk-Kopplungen EASY204-DP, EASY221-CO, EASY222-DN gleich.

Der Master tauscht mit den an die EC4-200 verbundenen Netzwerk-Kopplungen 3 Byte Daten in jeder Richtung aus.

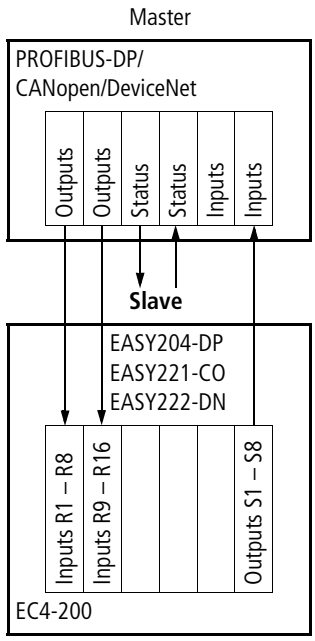


Abbildung 105: Zyklischer Datenaustausch zwischen Master und EASY204-DP, EASY221-CO, EASY222-DN

Aus der Sicht des Masters werden diese Daten in die EC4-200 geschrieben.

Byte	Bedeutung[1)]
0	Status (z. B. RUN/HALT)
1	R9…R16 (Inputs)
2	R1…R8 (Inputs)

1) Die Bedeutung der Bits, z. B. zeigt ein Bit von Byte 0 den RUN/HALT-Zustand der Steuerung an, ist in separaten Handbüchern beschrieben, → Tabelle 19.

Aus der Sicht des Masters werden diese Daten aus der EC4-200 gelesen.

Byte	Bedeutung[1)]
0	Status (z. B. RUN/HALT)
1	S1…S8 (Outputs)
2	Nicht belegt

1) Die Bedeutung der Bits, z. B. zeigt ein Bit von Byte 0 den RUN/HALT-Zustand der Steuerung an, ist in separaten Handbüchern beschrieben, → Tabelle 19.

Konfiguration

Die Konfiguration erfolgt in der Steuerungskonfiguration der Programmiersoftware easySoft-CoDeSys. Eine Netzwerk-Kopplung wird als Extension-Modul in den Konfigurationsbaum eingetragen. Er besitzt vordefinierte Eingangs-/Ausgangs-Kanäle (R1…R16, S1…S8) in denen die zyklischen Daten übertragen werden.

Einstellung der Teilnehmer-Adresse

Die Teilnehmer-Adresse der Netzwerk-Kopplung wird in einem speziellen Parameterdialog in der Steuerungskonfiguration eingestellt. Die Adresse wird dem Modul durch den Download des Programms, bzw. durch das Laden des Bootprojekts übergeben.

Die in der Steuerungskonfiguration eingestellte Adresse kann durch einen Eintrag in der Startup.ini-Datei überschrieben werden. Der zugehörige Eintrag in der Startup.ini-Datei lautet:

```
EXTENSION_SLAVE_ADDRESS = <Adresse>
```

Abbildung 106 zeigt am Beispiel des EASY204-DP, an welcher Stelle der Adresseintrag erfolgt.

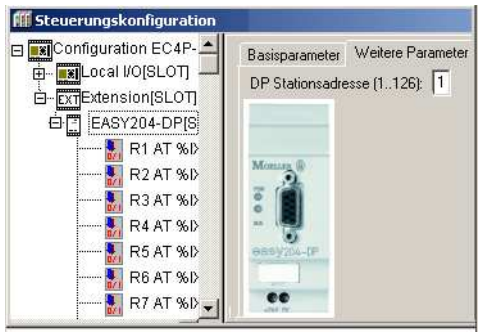


Abbildung 106: Adresse eintragen

→ Hinweis zum EASY204-DP:

Sie können die Bus-Adresse nur ändern, wenn die Kommunikation mit dem Master nicht aktiv ist.

Wurde einem Modul eine gültige Adresse zugewiesen, speichert es diese intern ab und übernimmt sie bei jedem Neustart. Wenn Sie eine neue Adresse in der Steuerungskonfiguration einstellen und einen Programmdownload ausführen, wird diese Adresse nur übergeben, wenn während des Downloads keine Kommunikation mit dem Master stattfindet, z. B. durch Abziehen des DP-Buskabels!

Wenn Sie das Programm vom PC in die Steuerung laden, überprüft die Steuerung, ob die aktuell von der Steuerung verwendete Adresse mit der konfigurierten Adresse übereinstimmt. Bei Ungleichheit erscheint eine Warnung.

Azyklischer Datenaustausch

Der azyklische Datenaustausch ermöglicht den Zugriff auf die definierten Objekte der EC4-200. Bei den Objekten handelt es sich um eine Untermenge der von easy800/MFD unterstützten Objekte.

Die in der Tabelle aufgeführten Objekte werden von der EC4-200 unterstützt und können von einem Master am CAN-, PROFIBUS-DP- oder DeviceNet angesprochen werden.

Tabelle 21:Objekte der EC4-200

Objekt-Name	Zugriffsart (R/W)
Betriebsart1)	R/W
Identifikation (nur bei EASY204-DP)	R
Eingänge I1…I16	R
Analogeingänge I7, I8, I11, I122)	R
Eingänge R1…R161)	R
Ausgänge Q1…Q8	R
Analog Ausgang QA1	R
Ausgänge S1…S81)	R
Lokale Diagnose ID1-ID163)	R
Eingänge Netzwerkteilnehmer IW1…IW8[3)]	R
Eingänge Netzwerkteilnehmer RW1…RW8[3)]	R
Ausgänge Netzwerkteilnehmer QW1…QW8[3)]	R
Ausgänge Netzwerkteilnehmer SW1…SW8[3)]	R
Receive-Daten Netzwerkteilnehmer RNW1…RNW8[3)]	R
Send-Daten Netzwerkteilnehmer SNW1…SNW8[3)]	R
Bit-Merker M1…M96[4)]	R/W
Byte-Merker MB1…MB96[4)]	R/W
Wort-Merker MW1…MW96[4)]	R/W
Doppelwort-Merker MD1…MD96[4)]	R/W
8 Byte Daten (MD67 - MD68)[4)]	R/W
16 Byte Daten (MD69…MD72)[4)]	R/W
32 Byte Daten (MD73…MD80)[4)]	R/W
64 Byte Daten (MD81…MD96)[4)]	R/W

1) Bei PROFIBUS-DP nur für Master der Klasse 2
2) IA1…IA4 in den Bedienungshandbüchern → Seite 87
3) Mit Netzwerkteilnehmer sind die Teilnehmer am easyNet-Netzwerk gemeint!
4) Die Abbildung der easy800/MFD-Merker auf die EC4P-200 gemäß Tabelle 22.

Der Zugriff auf andere easy800/MFD Objekte führt zu einer Fehlermeldung.

→ Nur bei EASY204-DP: Die Daten und Merker im Wort- und D-Wort-Format werden im Motorola-Format übertragen. Ein Byte-Swapping findet nicht statt!

Startadressen für Ein-/Ausgänge und Merker

Die Startadressen für die Adressbereiche der Ein- und Ausgänge können Sie in der Steuerungskonfiguration einstellen. Der Merkerbereich geht aus Tabelle 22 hervor.

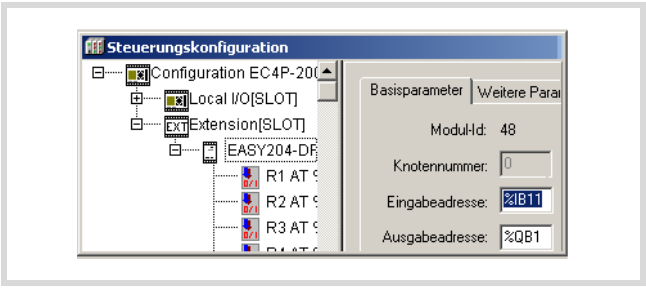


Abbildung 107: Einstellung der Adressbereiche

Die Konfiguration und die Einstellung der Teilnehmer-Adresse wurde bereits im Abschnitt „Zyklischer Datenaustausch" beschrieben.

Tabelle 22:Abbildung des EASY800-Merkerbereichs auf die EC4-200 (Werte der EC4-200 stehen in Klammern)

Bit	96–89 (11.7–11.0)	88–81 (10.7–10.0)	80–73 (9.7–9.0)	72–65 (8.7–8.0)	64–57 (7.7–7.0)	56–49 (6.7–6.0)	48–41 (5.7–5.0)	40–33 (4.7–4.0)	32–25 (3.7–3.0)	24–17 (2.7–2.0)	16–9 (1.7–1.0)	8–1 (0.0–0.7)
Byte	12 (11)	11 (10)	10 (9)	9 (8)	8 (7)	7 (6)	6 (5)	5 (4)	4 (3)	3 (2)	2 (1)	1 (0)
Word	6 (10)		5 (8)		4 (6)		3 (4)		2 (2)		1 (0)	
DWord	3 (8)				2 (4)				1 (0)			
Byte	24 (23)	23 (22)	22 (21)	21 (20)	20 (19)	19 (18)	18 (17)	17 (16)	16 (15)	15 (14)	14 (13)	13 (12)
Word	12 (22)		11 (20)		10 (18)		9 (16)		8 (14)		7 (12)	
DWord	6 (20)				5 (16)				4 (12)			
Byte	36 (35)	35 (34)	34 (33)	33 (32)	32 (31)	31 (30)	30 (29)	29 (28)	28 (27)	27 (26)	26 (25)	25 (24)
Word	18 (34)		17 (32)		16 (30)		15 (28)		14 (26)		13 (24)	
DWord	9 (32)				8 (28)				7 (24)			
Byte	48 (47)	47 (46)	46 (45)	45 (44)	44 (43)	43 (42)	42 (41)	41 (40)	40 (39)	39 (38)	38 (37)	37 (36)
Word	24 (46)		23 (44)		22 (42)		21 (40)		20 (38)		19 (36)	
DWord	12 (44)				11 (40)				10 (36)			
Byte	60 (59)	59 (58)	58 (57)	57 (56)	56 (55)	55 (54)	54 (53)	53 (52)	52 (51)	51 (50)	50 (49)	49 (48)
Word	30 (58)		29 (56)		28 (54)		27 (52)		26 (50)		25 (48)	
DWord	15 (56)				14 (52)				13 (48)			
Byte	72 (71)	71 (70)	70 (69)	69 (68)	68 (67)	67 (66)	66 (65)	65 (64)	64 (63)	63 (62)	62 (61)	61 (60)
Word	36 (70)		35 (68)		34 (66)		33 (64)		32 (62)		31 (60)	
DWord	18 (68)				17 (64)				16 (60)			
Byte	84 (83)	83 (82)	82 (81)	81 (80)	80 (79)	79 (78)	78 (77)	77 (76)	76 (75)	75 (74)	74 (73)	73 (72)
Word	42 (82)		41 (80)		40 (78)		39 (76)		38 (74)		37 (72)	
DWord	21 (80)				20 (76)				19 (72)			
Byte	96 (95)	95 (94)	94 (93)	93 (92)	92 (91)	91 (90)	90 (89)	89 (88)	88 (87)	87 (86)	86 (85)	85 (84)
Word	48 (94)		47 (92)		46 (90)		45 (88)		44 (86)		43 (84)	
DWord	24 (92)				23 (88)				22 (84)			

Word	54 (106)	53 (104)	52 (102)	51 (100)	50 (98)	49 (96)
DWord	27 (104)		26 (100)		25 (96)	
Word	60 (118)	59 (116)	58 (114)	57 (112)	56 (110)	55 (108)
DWord	30 (116)		29 (112)		28 (108)	
Word	66 (130)	65 (128)	64 (126)	63 (124)	62 (122)	61 (120)
DWord	33 (128)		32 (124)		31 (120)	
Word	72 (142)	71 (140)	70 (138)	69 (136)	68 (134)	67 (132)
DWord	36 (140)		35 (136)		34 (132)	
Word	78 (154)	77 (152)	76 (150)	75 (148)	74 (146)	73 (144)
DWord	39 (152)		38 (148)		37 (144)	
Word	84 (166)	83 (164)	82 (162)	81 (160)	80 (158)	79 (156)
DWord	42 (164)		41 (160)		40 (156)	
Word	90 (178)	89 (176)	88 (174)	87 (172)	86 (170)	85 (168)
DWord	45 (176)		44 (172)		43 (168)	
Word	96 (190)	95 (188)	94 (186)	93 (184)	92 (182)	91 (180)
DWord	48 (188)		47 (184)		46 (180)	
DWord	51 (200)		50 (196)		49 (192)	
DWord	54 (212)		53 (208)		52 (204)	
DWord	57 (224)		56 (220)		55 (216)	
DWord	60 (236)		59 (232)		58 (228)	
DWord	63 (248)		62 (244)		61 (240)	
DWord	66 (260)		65 (256)		64 (252)	
DWord	69 (272)		68 (268)		67 (264)	
DWord	72 (284)		71 (280)		70 (276)	
DWord	75 (296)		74 (292)		73 (288)	
DWord	78 (308)		77 (304)		76 (300)	
DWord	81 (320)		80 (316)		79 (312)	
DWord	84 (332)		83 (328)		82 (324)	
DWord	87 (344)		86 (340)		85 (336)	
DWord	90 (356)		89 (352)		88 (348)	
DWord	93 (368)		92 (364)		91 (360)	
DWord	96 (380)		95 (376)		94 (372)	

Anhang

Netzwerk CAN/easyNet

Zubehör

- RJ45-Stecker, Typ: EASY-NT-RJ45 (8-polig)

→ Konfektionierte Leitungen haben RJ45-Stecker an beiden Seiten.

Tabelle 23: Konfektionierte Leitungen

Leitungslänge cm	Typ
30	EASY-NT-30
80	EASY-NT-80
150	EASY-NT-150

- Frei konfektionierbare Leitung, Typ: EASY-NT-CAB (100 m 4 × 0,18 mm²)
- Crimpzange für RJ45-Stecker, Typ: EASY-RJ45-TOOL.
- Busabschlusswiderstand, Typ: EASY-NT-R RJ45-Stecker mit integriertem Busabschlusswiderstand 120 Ω

Leitungslänge und Querschnitte

Für den ordnungsgemäßen Betrieb des Netzwerkes ist es erforderlich, dass die Leitungslänge, der Querschnitt und der Leitungswiderstand der folgenden Tabelle entsprechen.

Leitungslänge	Leitungswiderstand	Querschnitt	
m	mΩ/m	mm²	AWG
bis 40	≦ 140	0,13	26
bis 175	≦ 70	0,25 bis 0,34	23, 22
bis 250	≦ 60	0,34 bis 0,5	22, 21, 20
bis 400	≦ 40	0,5 bis 0,6	20, 19
bis 600	≦ 26	0,75 bis 0,8	18
bis 1000	≦ 16	1,5	16

Der Wellenwiderstand der verwendeten Leitungen muss 120 Ω betragen.

→ Weitere Informationen zu den CAN-Kabellängen und CAN-Anschlüssen finden Sie in der ISO-Norm 11898.

Leitungslänge bei bekanntem Widerstand der Leitung berechnen

Ist der Widerstand der Leitung pro Längeneinheit bekannt (Widerstandsbelag R' in Ω/m), darf der gesamte Leitungswiderstand R_L folgende Werte nicht überschreiten. R_L ist abhängig von den gewählten Baudraten:

Baudrate kBaud	Leitungswiderstand R_L Ω
10 bis 125	≦ 30
250	≦ 25
500	≦ 12

l_{max} = maximale Länge der Leitung in m

R_L = gesamter Leitungswiderstand in Ω

R' = Widerstand der Leitung pro Längeneinheit in Ω/m

$$l_{max} = \frac{R_L}{R'}$$

Querschnitt bei bekannter Leitungslänge berechnen

Für die bekannte maximale Ausdehnung des Netzwerkes wird der minimale Querschnitt ermittelt.

l = Länge der Leitung in m

S_{min} = minimaler Leitungsquerschnitt in mm²

ρ_{cu} = spezifischer Widerstand von Kupfer, falls nichts anderes angegeben 0,018 Ωmm²/m

$$S_{min} = \frac{l \times \rho_{cu}}{12{,}4}$$

→ Wenn das Ergebnis der Berechnung keinen Normquerschnitt ergibt, nehmen Sie den nächst größeren Querschnitt.

Leitungslänge bei bekanntem Querschnitt berechnen

Für einen bekannten Leitungsquerschnitt wird die maximale Leitungslänge berechnet:

l_{max} = Länge der Leitung in m

S = Leitungsquerschnitt in mm²

ρ_{cu} = spezifischer Widerstand von Kupfer, falls nichts anderes angegeben 0,018 Ωmm²/m

$$l_{max} = \frac{S \times 12{,}4}{\rho_{cu}}$$

Beispielprogramm zum START/STOP der Steuerung über externen Schalter

Für den Start benötigen Sie die Funktion SysStartPlcProgram, für den Stop die Funktion SysStopPlcProgram aus der Bibliothek SysLibPlcCtrl.lib.

Für diesen Fall ist das Startverhalten der Steuerung im Steuerungskonfigurator unter <Weitere Parameter → Einstellungen> auf WARMSTART zu setzen!

Zur Funktion

Über die POU „StartPrg", die einmalig bei jedem Start der Steuerung aufgerufen wird, findet eine Registrierung der Funktion „FuncCalledWhenPlcIsInStop" auf das „Event" „EVENT_TASKCODE_NOT_CALLED" statt. Diese Registrierung bewirkt, dass die Funktion „FuncCalledWhenPlcIsInStop" über das „Event" „EVENT_TASKCODE_NOT_CALLED" aufgerufen wird, wenn die Steuerung im STOP ist. Mit Hilfe der Funktion „StartStopFunction" wird der Status des Eingangs überwacht und bei einer Statusänderung die Funktion zum Starten bzw. Stoppen der Steuerung aufgerufen.

Da die POU „StartPrg" einmalig aufgerufen wird, sollten in dieser POU keine Ausgänge oder Parameter gesetzt werden. Anwenderspezifische Programme sollten Sie in separaten POUs programmiert werden.

► Aktivieren Sie das Systemereignis „Start" und benennen Sie die „aufgerufene POU" mit dem Namen Startprg.

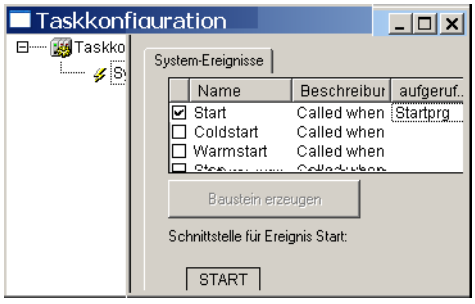


Abbildung 108: Systemereignis aktivieren

► Öffnen Sie im Ordner BAUSTEINE eine neue POU mit dem Namen „Startprg" und programmieren Sie die Funktion „SysCallbackRegister" , welche die Start/Stop-Funktionen dem Betriebssystem „vorstellt".

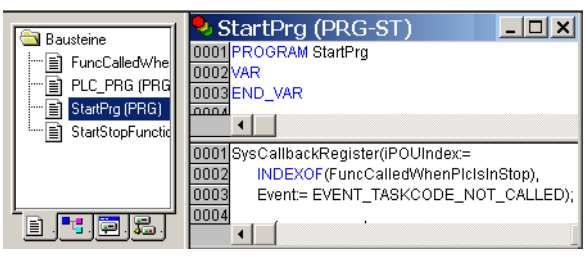


Abbildung 109: Funktion „Startprg"

► Deklarieren Sie die folgenden globalen Variablen.

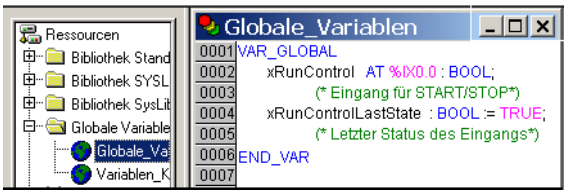


Abbildung 110: Globale Variable deklarieren

► Geben Sie das Programm für PLC_PRG nach Abbildung 111 ein. Es ist wichtig, dass das Anwenderprogramm bzw. die Aufrufe der POUs gemäß der Abbildung 111 eingefügt werden.

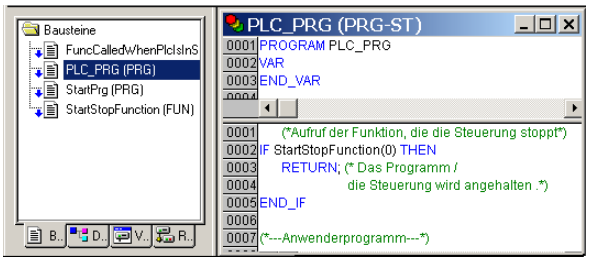


Abbildung 111: Abfrage START/STOP

► Geben Sie die Funktion „FuncCalledWhenPlcInStop" und „StartStopFunction" ein.

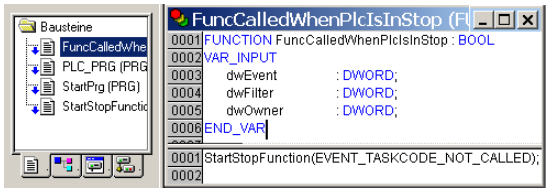


Abbildung 112: Aufruf der Funktion FuncCalledWhenPlcIsInStop

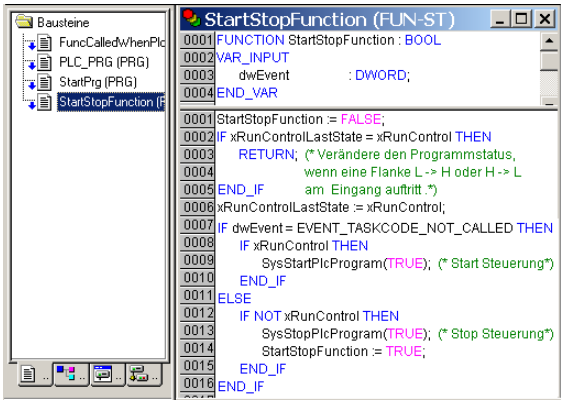


Abbildung 113: Funktion, die den Eingang überwacht

Verbindungskabel EASY800-PC-CAB

9 poliger Buchsen-Stecker am Kabel (Terminal/PC-Stecker)

Pin	Signal
2	RxD
3	TxD
4	DTR
5	GND
7	RTS

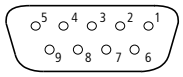


Abbildung 114: 9-poliger Buchsen-Stecker

→ Zur Funktion des Kabels muss das RTS-Signal gesetzt sein, da die Spannung an der RTS-Leitung die Bauelemente im Kabelstecker versorgt.

Abmessungen und Gewicht

Abmessungen B × H × T	
[mm] mit Adapter für MMC	107,5 × 90 × 72 107,5 × 90 × 79
[inches] mit Adapter für MMC	4,23 × 3,54 × 2,84 4,23 × 3,54 × 3,11
Teilungseinheiten (TE) breit	6
Gewicht	
[g]	320
[lb]	0,705
Montage	Hutschiene DIN 50022, 35 mm oder Schraubmontage mit 3 Gerätefüßen ZB4-101-GF1

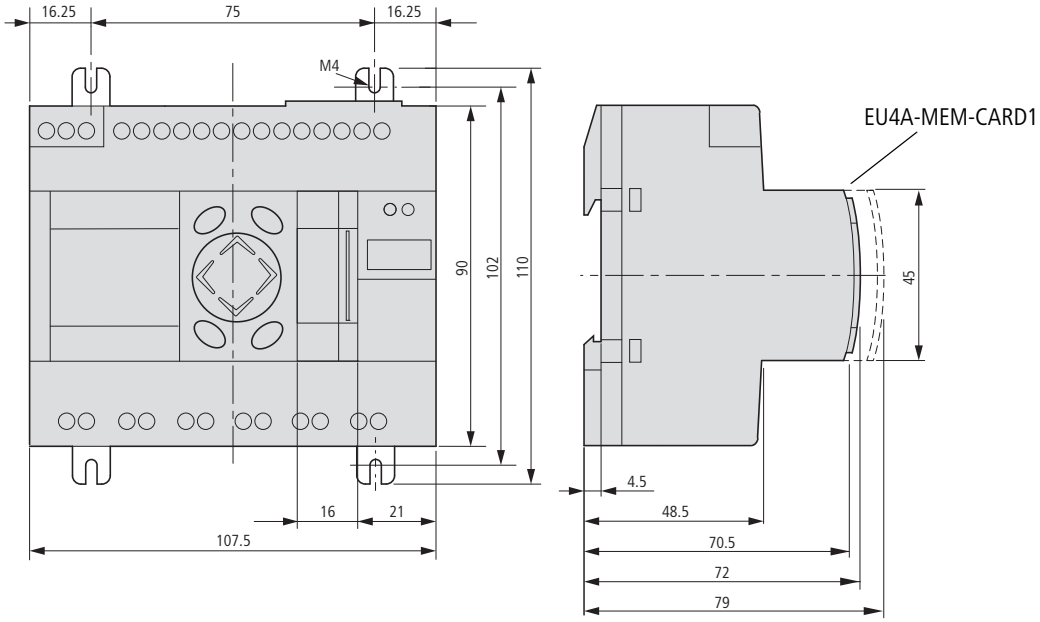


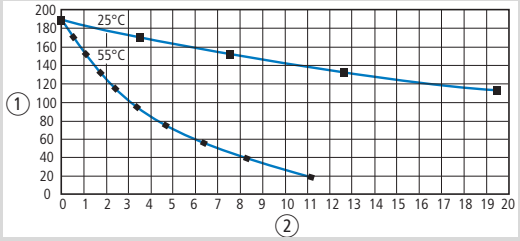
Abbildung 115: Abmessung in mm (Angaben in inches → Tabelle 24)

Tabelle 24: Abmessungen in inches

mm	inches	mm	inches
4,5	0,177	79	3,11
16,25	0,64	90	3,54
48,5	1,91	102	4,01
70,5	2,78	107,5	4,23
72	2,83	110	4,33
75	2,95		

Technische Daten

Klimatische Umgebungsbedingungen (Kälte nach IEC 60068-2-1, Wärme nach IEC 60068-2-2)			
Betriebsumgebungstemperatur waagerechter/senkrechter Einbau		°C, (°F)	–25 bis 55, (–13 bis 131)
Betauung			Betauung durch geeignete Maßnahmen verhindern
LCD-Anzeige (Sicher lesbar)		°C, (°F)	0 bis 55, (32 bis 131)
Lager-/Transporttemperatur		°C, (°F)	–40 bis 70, (–40 bis 158)
Relative Luftfeuchte (IEC 60068-2-30), keine Betauung		%	5 bis 95
Luftdruck (Betrieb)		hPa	795 bis 1080
Mechanische Umgebungsbedingungen			
Schutzart (IEC/EN 60529)			IP20
Schwingungen (IEC/EN 60068-2-6)			
konstante Amplitude 3,5 mm		Hz	5 bis 9
konstante Beschleunigung 1 g		Hz	9 bis 150
Schocken (IEC/EN 60068-2-27) Halbsinus 15 g/11 ms		Schocks	18
Kippfallen (IEC/EN 60068-2-31)	Fallhöhe	mm	50
Freier Fall, verpackt (IEC/EN 60068-2-32)		m	1
Einbaulage			waagrecht,senkrecht
Elektromagnetische Verträglichkeit (EMV)			
Elektrostatische Entladung (ESD), (IEC/EN 61000-4-2, Schärfegrad 3)			
Luftentladung		kV	8
Kontaktentladung		kV	6
Elektromagnetische Felder (RFI), (IEC/EN 61000-4-3)		V/m	10
Funkentstörung Grenzwertklasse			EN 55011, EN 55022 Klasse B
Burst Impulse (IEC/EN 61000-4-4, Schärfegrad 3)			
Versorgungsleitungen		kV	2
Signalleitungen		kV	2
Energiereiche Impulse (Surge) (IEC/EN 61000-4-5, Schärfegrad 2)		kV	0,5 symmetrisch 1 unsymmetrisch
Einströmung (IEC/EN 61000-4-6)		V	10
Isolationsfestigkeit			
Bemessung der Luft- und Kriechstrecken			EN 50178
Isolationsfestigkeit			EN 50178
Überspannungskategorie/Verschmutzungsgrad			II/2
Werkzeug und Anschlussquerschnitte			
eindrähtig, minimal bis maximal		mm2	0,2 bis 4
		AWG	22 bis 12
feindrähtig mit Aderendhülse, minimal bis maximal		mm2	0,2 bis 2,5
		AWG	22 bis 12
factory wiring		AWG	30

Schlitzschraubendreherbreite	mm	3,5 × 0,8
	inch	0,14 × 0,03
Anzugsdrehmoment	Nm	0,6
CPU		
Speicherangaben		
Programmkode	kByte	256
Programmdaten	kByte	14 Segmente a 16KB
Merker/Input/Output/Retaindaten	kByte	16/4/4/8
Zykluszeit für 1 k Anweisungen		< 0,3
Pufferung/Genauigkeit der Echtzeituhr		
Pufferung der Uhr		 ① Pufferzeit in Stunden ② Betriebsdauer in Jahren
Genauigkeit der Echtzeituhr		
pro Tag	s/Tag	± 5
pro Jahr	h/Jahr	± 0,5
Schnittstellen		
Programmierschnittstelle		
Anschlusstechnik		RJ45, 8polig
RS232 (ohne Steuerleitungen)		
Steuerungsport		COM1
Potentialtrennung		keine
Betriebsart Programmierung		
Datenübertragungsrate		4.8, 9.6, 19.2, 38.4, 57.6
Zeichenformat		8 Datenbit, keine Parität, 1 Stopp-Bit
Betriebsart Transparentmodus		
Datenübertragungsrate		0.3, 0.6, 1.2, 2.4, 4.8, 9.6, 19.2, 38.4, 57.6
Zeichenformat		8E1, 8O1, 8N1, 8N2, 7E2, 7O2, 7N2, 7E1
Anzahl der Sendebyte in einem Block		190
Anzahl der Empfangsbyte in einem Block		190
Ethernet		
Datenübertragungsrate	MBit/s	10
Potentialtrennung		ja
Multifunktions-Schnittstelle (RS232) ohne Steuerleitungen		
Betriebsart Transparentmodus		
Steuerungsport		COM2
Potentialtrennung		ja, im Kabel easy800-PC-CAB
Anschlusstechnik		Kabel easy800-PC-CAB
Datenübertragungsrate	kBit/s	9.6, 19.2

CAN(open)/easyNet		
Datenübertragungsrate	kBit/s	10, 20, 50, 100, 125, 250, 500 Default: 125
Potentialtrennung zu Ein-/Ausgängen/Versorgung		ja
Busabschlusswiderstand		extern: 120 Ω oder Stecker EASY-NT-R (inkl. Busabschlusswiderstand 120 Ω)
Anschlusstechnik		2 x RJ45, 8polig
Betriebsart CAN(open):		
– Teilnehmer	Anzahl	max. 126
– PDO-Art		asynchron, zyklisch, azyklisch
– Geräteprofil		nach DS301 V4
Betriebsart easyNet:		
– Teilnehmer	Anzahl	max. 8

Spannungsversorgung		
Bemessungsspannung		
Nennwert	V DC, (%)	24, (–15, +20)
Zulässiger Bereich	V DC	20,4 bis 28,8
Restwelligkeit	%	≦ 5
Eingangsstrom bei 24 V DC, typisch	mA	140
Spannungseinbrüche, IEC/EN 61131-2	ms	10
Verlustleistung bei 24 V DC, typisch	W	3,4

Eingänge		
Digital-Eingänge		
Anzahl		12
Eingänge für analoge Signale nutzbar		I 7,8,11,12
Eingänge für Impuls-Signale nutzbar (Schnelle Zähler)		I 1,2,3,4
Eingänge zur Interrupterzeugung		I 1,2,3,4
Zustandsanzeige		LC-Display
Potentialtrennung		
zur Versorgungsspannung, PC-Schnittstelle		Nein
gegeneinander		Nein
zu den Ausgängen, zu CAN-Schnittstellen		Ja
Bemessungsspannung		
Nennwert	V DC	24
bei Zustand „0"		
I1 bis I6 und I9 bis I10	V DC	< 5
I7, I8, I11, I12	V DC	< 8
bei Zustand „1"		
I1 bis I6 und I9 bis I10	V DC	> 15
I7, I8, I11, I12	V DC	> 8
Eingangsstrom bei Zustand „1" (bei 24 V DC)		
I1 bis I6, I9 bis I10	mA	3,3
I7, I8, I11, I12	mA	2,2

Verzögerungszeit von „0" nach „1"		
I1 bis I4	ms	0,02
I5 bis I12	ms	0,25
Verzögerungszeit von „1" nach „0"		
I1 bis I4	ms	0,02
I5 bis I12	ms	0,25
Leitungslänge (ungeschirmt)	m	100
Weitere Funktionen der Eingänge		
Eingänge für analoge Signale		
Anzahl		4 (I7,I8,I11,I12)
Signalbereich	V DC	0 bis 10
Auflösung analog	V	0,01
Auflösung digital	Bit	10
	Wert	0 bis 1023
Eingangsimpedanz	kΩ	11,2
Genauigkeit vom Istwert		
zwei Geräte	%	± 3
innerhalb eines Gerätes	%	± 2
Eingangsstrom	mA	< 1
Leitungslänge (geschirmt)	m	30
Eingänge für schnelle Zähler		I 1, I2
Anzahl/Wertebereich	Bit	2 × 16 Bit (I1,I2) 1 × 32 Bit (I1)
max. Frequenz	kHz	50
Zählrichtung über Software umschaltbar		inkrementieren/dekrementieren
Leitungslänge (geschirmt)	m	20
Impulsform		Rechteck
Impuls-Pausenverhältnis		1:1
Inkrementalwertzähler		I1,I2, I3, I4
Anzahl		1
Wertebereich	Bit	32 Bit
max. Frequenz	kHz	40
Leitungslänge (geschirmt)	m	20
Impulsform		Rechteck
Zähleingänge		I1,I2 = Zähleingang I3 = Referenzimpuls I4 = Referenzfenster
Signalversatz		90°
Puls-Pausenverhälnis		1:1
Eingänge zur Interrupterzeugung		I1, I2, I3, I4
max. Frequenz	kHz	3

Relais-Ausgänge		
Anzahl der Ausgänge		6
Parallelschaltung von Ausgängen zur Leistungserhöhung		nicht zulässig
Absicherung eines Ausgangsrelais		
Leitungsschutzschalter B16	A	16
oder Sicherung (Träge)	A	8
Potentialtrennung		Ja
Sichere Trennung	V AC	300
Basisisolierung	V AC	600
Mechanische Lebensdauer	Schalt-spiele	10 × 106
Strombahnen Relais		
Konventioneller therm. Strom	A	8
Empfohlen für Last bei 12 V AC/DC	mA	> 500
Kurzschlussfest cos φ = 1 Charakteristik B (B16) bei 600 A	A	16
Kurzschlussfest cos φ = 0,5 bis 0,7 Charakteristik B (B16) bei 900 A	A	16
Bemessungsstoßspannungsfestigkeit U_{imp} Kontakt-Spule	kV	6
Bemessungsisolationsspannung U_i		
Bemessungsbetriebsspannung U_e	V AC	250
Sichere Trennung nach EN 50178 zwischen Spule und Kontakt	V AC	300
Sichere Trennung nach EN 50178 zwischen zwei Kontakten	V AC	300
Einschaltvermögen, IEC 60947		
AC-15 250 V AC, 3 A (600 S/h)	Schaltspiele	300000
DC-13 L/R ≦ 150 ms 24 V DC, 1 A (500 S/h)	Schalt-spiele	200000
Ausschaltvermögen, IEC 60947		
AC-15 250 V AC, 3 A (600 S/h)	Schalt-spiele	300000
DC-13 L/R ≦ 150 ms 24 V DC, 1 A (500 S/h)	Schalt-spiele	200000
Glühlampenlast		
1000 W bei 230/240 V AC	Schalt-spiele	25000
500 W bei 115/120 V AC	Schalt-spiele	25000
Leuchtstoffröhrenlast 10 × 58 W bei 230/240 V AC		
Leuchtstoffröhren - mit elektrischen Vorschaltgerät - konventionell kompensiert - unkompensiert	Schalt-spiele	25000

Schaltfrequenz Relais		
mechanische Schaltspiele	Schalt-spiele	10 Mio. (107)
mechanische Schaltfrequenz	Hz	10
ohmsche/Lampenlast	Hz	2
induktive Last	Hz	0,5

Transistor-Ausgänge		
Anzahl der Ausgänge		8
Bemessungsspannung U_e	V DC	24
zulässiger Bereich	V DC	20,4 bis 28,8
Restwelligkeit	%	≦ 5
Versorgungsstrom		
bei Zustand „0", typisch/maximal	mA	18/32
bei Zustand „1", typisch/maximal	mA	24/44
Verpolungsschutz ▽**Achtung!** Wird bei verpolter Versorgungsspannung an die Ausgänge Spannung gelegt, entsteht ein Kurzschluss.		Ja
Potentialtrennung		Ja
Bemessungsstrom I_e bei Zustand „1", maximal	A	0,5
Lampenlast ohne R_V	W	5
Reststrom bei Zustand „0" pro Kanal	mA	< 0,1
maximale Ausgangsspannung		
bei Zustand „0" mit externer Last, 10 MΩ	V	2,5
bei Zustand „1", I_e = 0,5 A		$U = U$e – 1 V
Kurzschlussschutz (thermisch) Gruppe Q1 bis Q4 /Gruppe Q5 bis Q8. Auswertung erfolgt mit Diagnose-Eingang I16 (Q1 bis Q4), I17 (Q5 bis Q8) ▽**Achtung!** Setzen Sie die Ausgangsgruppe im Programm auf 0-Pegel um eine Überlastung des Ausgangs zu vermeiden		Ja
Kurzschlussauslösestrom für R_a ≦ 10 mΩ (abhängig von der Anzahl der aktiven Kanäle und deren Belastung)	A	0,7 ≦ I_e ≦ 2
maximal gesamter Kurzschlussstrom	A	16
Spitzenkurzschlussstrom	A	32
thermische Abschaltung		Ja
maximale Schaltfrequenz bei konstanter ohmscher Belastung R_L = 100 kΩ (abhängig vom Programm und Belastung)	Schalt-spiele/h	40000

Parallelschaltbarkeit der Ausgänge bei ohmscher Belastung; induktiver Belastung mit externer Schutzbeschaltung (→ Abschnitt „Transistor-Ausgänge anschließen", Seite 25); Kombination innerhalb einer Gruppe		Ja
Gruppe 1: Q1 bis Q4		
Gruppe 2: Q5 bis Q8		
Anzahl der Ausgänge maximal		4
gesamter Maximalstrom	A	2
▽ **Achtung!** Ausgänge müssen gleichzeitig und von gleicher Zeitlänge angesteuert werden.		
Zustandsanzeige der Ausgänge		LC-Display

Induktive Belastung **ohne äußere Schutzbeschaltung**

Allgemeine Erläuterungen:
$T_{0,95}$ = Zeit in Millisekunden, bis 95 % des stationären Stromes erreicht sind

$$T_{0,95} \approx 3 \times T_{0,65} = 3 \times \frac{L}{R}$$

Gebrauchskategorien in Gruppen zu Q1 bis Q4, Q5 bis Q8

T0,95 = 1 ms R = 48 Ω L = 16 mH	Gleichzeitigkeitsfaktor je Gruppe g =		0,25
	relative Einschaltdauer	%	100
	maximale Schaltfrequenz f = 0,5 Hz maximale Einschaltdauer ED = 50 %	Schalt-spiele/h	1500
DC13 T0,95 = 72 ms R = 48 Ω L = 1,15 H	Gleichzeitigkeitsfaktor g =		0,25
	relative Einschaltdauer	%	100
	maximale Schaltfrequenz f = 0,5 Hz maximale Einschaltdauer ED = 50 %	Schalt-spiele/h	1500

Andere induktive Lasten:

T0,95 = 15 ms R = 48 Ω L = 0,24 H	Gleichzeitigkeitsfaktor g =		0,25
	relative Einschaltdauer	%	100
	maximale Schaltfrequenz f = 0,5 Hz maximale Einschaltdauer ED = 50 %	Schalt-spiele/h	1500
Induktive Belastung mit äußerer Schutzbeschaltung bei jeder Last (→ Abschnitt „Transistor-Ausgänge anschließen", Seite 25)			
	Gleichzeitigkeitsfaktor g =		1
	relative Einschaltdauer	%	100
	maximale Schaltfrequenz maximale Einschaltdauer	Schalt-spiele/h	In Abhängigkeit von der Schutzbeschaltung

Analog-Ausgang		
Anzahl		1
Potentialtrennung		
zur Spannungsversorgung		nein
zu den digitalen Eingängen		nein
zu den digitalen Ausgängen		ja
zu dem Netwerk easyNet		ja
Ausgangsart		DC-Spannung
Signalbereich	V DC	0 bis 10
Ausgangsstrom maximal	mA	10
Bürdenwiderstand	kΩ	1
Kurzschluss- und Überlastsicher		ja
Auflösung analog	V	0,01
Auflösung digital	Bit	10
	Wert	0 bis 1023
Einschwingzeit	us	100
Genauigkeit		
(-25 bis 55 °C), bezogen auf den Bereich	%	2
(25 °C), bezogen auf den Bereich	%	1
Konvertierungszeit		jeder CPU-Zyklus

Zeichensätze

Latin 1, „Westeuropa"

Code	Bedeutung	Code	Bedeutung	Code	Bedeutung	Code	Bedeutung
0	Leerzeichen	64	@	128	€	192	À
1	Ā	65	A	129	‰	193	Á
2	B̄	66	B	130	↕	194	Â
3	C̄	67	C	131	?	195	Ã
4	D̄	68	D	132	„	196	Ä
5	Ē	69	E	133	┌	197	Å
6	F̄	70	F	134	└	198	Æ
7	Ḡ	71	G	135	├	199	Ç
8	H̄	72	H	136	□	200	È
9	Ī	73	I	137	┐	201	É
10	J̄	74	J	138	┘	202	Ê
11	K̄	75	K	139	┤	203	Ë
12	L̄	76	L	140	–	204	Ì
13	M̄	77	M	141	┬	205	Í
14	N̄	78	N	142	┴	206	Î
15	Ō	79	O	143	┼	207	Ï
16	P̄	80	P	144	╞	208	Ð
17	Q̄	81	Q	145	╟	209	Ñ
18	R̄	82	R	146	╭	210	Ò
19	S̄	83	S	147	√	211	Ó
20	T̄	84	T	148	⌠	212	Ô
21	Ū	85	U	149	•	213	Õ
22	V̄	86	V	150	▐	214	Ö
23	W̄	87	W	151	⌡	215	×
24	X̄	88	X	152	╩	216	Ø
25	Ȳ	89	Y	153	Š	217	Ù
26	Z̄	90	Z	154	š	218	Ú
27	↓	91	[	155	Œ	219	Û
28	↓	92	\	156	œ	220	Ü
29	→	93	]	157	Ž	221	Ý
30	←	94	^	158	ž	222	Þ
31	↙ (cursor)	95	_	159	Ÿ	223	ß
32	Leerzeichen	96	`	160	√	224	à
33	!	97	a	161	¡	225	á
34	"	98	b	162	¢	226	â
35	#	99	c	163	╠	227	ã
36	$	100	d	164	@	228	ä
37	%	101	e	165	╒	229	å
38	&	102	f	166	╘	230	æ
39	'	103	g	167	╠	231	ç
40	(	104	h	168	▐ (Cursor)	232	è
41	)	105	i	169	╗	233	é
42	*	106	j	170	╜	234	ê
43	+	107	k	171	╣	235	ë
44	,	108	l	172	=	236	ì
45	–	109	m	173	╓	237	í
46	.	110	n	174	╙	238	î
47	/	111	o	175	╫	239	ï
48	0	112	p	176	°	240	ð
49	1	113	q	177	±	241	ñ
50	2	114	r	178	²	242	ò
51	3	115	s	179	³	243	ó
52	4	116	t	180	´	244	ô
53	5	117	u	181	µ	245	õ
54	6	118	v	182	10	246	ö
55	7	119	w	183	11	247	≥
56	8	120	x	184	12	248	ø
57	9	121	y	185	13	249	ù
58	:	122	z	186	14	250	ú
59	;	123	{	187	15	251	û
60	<	124	¦	188	16	252	ü
61	=	125	}	189	⓪	253	ý
62	>	126	~	190	⓪	254	þ
63	?	127	£	191	¿	255	ÿ

Zeichensatz Latin 2, „Mitteleuropa" (für Polnisch, Ungarisch und Tschechisch)

Code	Bedeutung	Code	Bedeutung	Code	Bedeutung	Code	Bedeutung
0	Leerzeichen	64	@	128	€	192	Ŕ
1	Ā	65	A	129	¢	193	Á
2	B̄	66	B	130	‡	194	Â
3	C̄	67	C	131	‚	195	Ă
4	D̄	68	D	132	„	196	Ä
5	Ē	69	E	133		197	Ĺ
6	F̄	70	F	134		198	Ć
7	Ḡ	71	G	135		199	Ç
8	H̄	72	H	136	□	200	Č
9	Ī	73	I	137	‰	201	É
10	J̄	74	J	138	Š	202	Ę
11	K̄	75	K	139	‹	203	Ë
12	L̄	76	L	140	ś	204	Ě
13	M̄	77	M	141	ť	205	Í
14	N̄	78	N	142	Ž	206	Î
15	Ō	79	O	143	ż	207	Ď
16	P̄	80	P	144	Ţ	208	Đ
17	Q̄	81	Q	145	'	209	Ń
18	R̄	82	R	146	'	210	Ň
19	S̄	83	S	147	"	211	Ó
20	T̄	84	T	148	"	212	Ô
21	Ū	85	U	149	•	213	Ő
22	V̄	86	V	150	▓	214	Ö
23	W̄	87	W	151	–	215	≤
24	X̄	88	X	152	²	216	Ř
25	Ȳ	89	Y	153	³	217	Ů
26	Z̄	90	Z	154	š	218	Ú
27	↑	91	[	155	›	219	Ű
28	↓	92	\	156	ś	220	Ü
29	→	93	]	157	?	221	Ý
30	←	94	^	158	ż	222	◙
31	↙ (cursor)	95	_	159	ź	223	ß
32	Leerzeichen	96	`	160	√	224	ŕ
33	!	97	a	161		225	á
34	"	98	b	162		226	â
35	#	99	c	163	ł	227	ă
36	$	100	d	164		228	ä
37	%	101	e	165	Ą	229	ĺ
38	&	102	f	166	¦	230	ć
39	'	103	g	167		231	ç
40	(	104	h	168	▓ (cursor)	232	č
41	)	105	i	169	©	233	é
42	*	106	j	170	Ş	234	ę
43	+	107	k	171		235	ë
44	‚	108	l	172		236	ě
45	–	109	m	173		237	í
46	.	110	n	174		238	î
47	/	111	o	175	ż	239	d'
48	0	112	p	176	°	240	đ
49	1	113	q	177	±	241	ń
50	2	114	r	178		242	ň
51	3	115	s	179	ł	243	ó
52	4	116	t	180	´	244	ô
53	5	117	u	181	µ	245	ő
54	6	118	v	182	®	246	ö
55	7	119	w	183	®	247	≥
56	8	120	x	184		248	ř
57	9	121	y	185	ą	249	ů
58	:	122	z	186	ş	250	ú
59	;	123	{	187		251	ű
60	<	124	¦	188	Ľ	252	ü
61	=	125	}	189		253	ý
62	>	126	~	190	ľ	254	ţ
63	?	127	£	191	ż	255	

Zeichensatz „Kyrillisch“ (für Russisch)

Code	Bedeutung	Code	Bedeutung	Code	Bedeutung	Code	Bedeutung
0	Leerzeichen	64	@	128	€	192	А
1	Ā	65	A	129	ŕ	193	Б
2	B̄	66	B	130	‡	194	В
3	C̄	67	C	131	ŕ	195	Г
4	D̄	68	D	132	„	196	Д
5	Ē	69	E	133		197	Е
6	F̄	70	F	134	²	198	Ж
7	Ḡ	71	G	135	³	199	З
8	H̄	72	H	136	□	200	И
9	Ī	73	I	137	‰	201	Й
10	J̄	74	J	138	љ	202	К
11	K̄	75	K	139	<	203	Л
12	L̄	76	L	140	њ	204	М
13	M̄	77	M	141	Ќ	205	Н
14	N̄	78	N	142	ћ	206	О
15	Ō	79	O	143	џ	207	П
16	P̄	80	P	144	ђ	208	Р
17	Q̄	81	Q	145	'	209	С
18	R̄	82	R	146	'	210	Т
19	S̄	83	S	147	"	211	У
20	T̄	84	T	148	"	212	Ф
21	Ū	85	U	149	•	213	Х
22	V̄	86	V	150	▉	214	Ц
23	W̄	87	W	151	–	215	Ч
24	X̄	88	X	152	Ю	216	Ш
25	Ȳ	89	Y	153	h	217	Щ
26	Z̄	90	Z	154	њ	218	Ъ
27	□	91	[	155	>	219	Ы
28	↓	92	\	156	ћ	220	Ь
29	→	93	]	157	Ќ	221	Э
30	←	94	^	158	ћ	222	Ю
31	↙ (cursor)	95	_	159	џ	223	Я
32	Leerzeichen	96	`	160	ѵ	224	а
33	!	97	a	161	ў	225	б
34	"	98	b	162	ў	226	в
35	#	99	c	163	J	227	г
36	$	100	d	164	ђ	228	д
37	%	101	e	165	Ґ	229	е
38	&	102	f	166	¦	230	ж
39	'	103	g	167	Ё	231	з
40	(	104	h	168	▉ (cursor)	232	и
41	)	105	i	169	©	233	й
42	*	106	j	170	Є	234	к
43	+	107	k	171	≤	235	л
44	,	108	l	172		236	м
45	–	109	m	173		237	н
46	.	110	n	174	h	238	о
47	/	111	o	175	Ї	239	п
48	0	112	p	176	°	240	р
49	1	113	q	177	±	241	с
50	2	114	r	178	I	242	т
51	3	115	s	179	i	243	у
52	4	116	t	180	ґ	244	ф
53	5	117	u	181	µ	245	х
54	6	118	v	182	¶	246	ц
55	7	119	w	183	¶	247	ч
56	8	120	x	184	ё	248	ш
57	9	121	y	185	¢	249	щ
58	:	122	z	186	є	250	ъ
59	;	123	{	187	≥	251	ы
60	<	124	¦	188	j	252	ь
61	=	125	}	189	S	253	э
62	>	126	~	190	s	254	ю
63	?	127	J	191	ї	255	я

Stichwortverzeichnis

A

Abgesetztes Display 75
Abmessungen 95
Adapter für Speicherkarte 27
Adressierung, Steuerung am CANopen-Feldbus 70
Akku-Pufferung 43
Analog-Ausgang anschließen 26
Analog-Eingänge anschließen 21
Anschließen
- 20-mA-Sensor 22
- Analog-Ausgang 26
- Analog-Eingänge 21
- Ausgänge 24
- Digital-Eingänge 21
- Erweiterungen 28
- Impulsgeber 23
- Inkrementalwertgeber 23
- Näherungsschalter 21
- Netzwerk easyNet 93
- Netzwerk-Kopplungen 28
- Relais-Ausgänge 24
- Schnelle Zähler 23
- Schütze, Relais 24
- Servoventil 26
- Sollwertgeber 22
- Taster, Schalter 21
- Temperatursensor 22
- Transistor-Ausgänge 25
- Versorgungsspannung 21

Anzeige
- Ein-/Ausgänge der Erweiterungsgeräte 40
- Lokale Ein-/Ausgänge 39

Anzeige (Display) 75
Applikations-Routine 51
Aufbau, EC4-200 11
Ausgänge
- anschließen 24
- Art und Anzahl 13
- symbolische Operanden 17

Ausschalten der Spannungsversorgung 43

B

Baudrate festlegen/ändern 63
Bedienung 29
Beispiele
- Allgemeiner Ablauf zur Programmerstellung 80
- Analogwerterfassung 21
- Ausgabe von Texten und Werten 81
- Einstellung der Node-Id, Baudrate 70
- Interruptverarbeitung 52
- Programm mit Funktionsaufruf 51
- STARTUP.INI Datei für EC4-200 67
- Zugriff auf ein Steuerungsprogramm 72

Betrieb 41
Betriebssystem herunterladen/aktualisieren 56
Betriebssystem-Update 41
Bibliotheken
- CANUser.lib, CANUser_Master.lib 7
- EC_File.lib 13
- EC_SysLibCom.lib 73
- EC_Util.lib 52, 62
- EC_Visu2.lib 62
- installieren 61
- SysLibRTC 14

Blockgröße, für Datenübertragung 69
Bootprojekt 41
- erzeugen/transferieren 55
- löschen 55

Breakpoint 45
Browser-Befehle 59, 60
Browserbefehle 60
Busauslastung, CANopen-Feldbus 60, 62

C

CAN
- -Deviceparameter 71
- -Masterparameter 71
- -Routingeinstellungen 71
- -Schnittstelle 16
- -Verbindung 27

canload, Browser-Befehl 60
CoDeSys Gateway Server 70
COLDSTART 49
Counter 45
Counter-Interrupt 50
Cursor-Anzeige 29
Cursortasten, Eingänge 12

D

Datenaustausch, Netzwerk-Kopplungen
- azyklisch 89
- zyklisch 87, 88

Datenzugriff, auf MMC 13
Dezentrale Erweiterung 28
Diagnose-Eingänge 12, 17
Diagnosemöglichkeiten 72
Digital-Eingänge anschließen 21
Direkter I/O-Zugriff 53
Display, interaktiv 75
Download, Betriebssystem 56

E
easyLink 17, 28
easyNet
-Schnittstelle 16, 27
Echtzeituhr 14
Ein-/Ausgangssignale 47
Ein-/Ausgangs-Übersicht 17
Eingänge
Art und Anzahl 11
symbolische Operanden 17
Einschaltverhalten 41, 67
Einzelschritt-Betrieb 45
Einzelzyklus-Betrieb 45
Erweiterungen anschließen 20, 28
Erweiterungsgeräte 17
Externes Display 75

F
factoryset 44
Fehlercode 54
Forcen 45
Funktion 61
Funktionen 77
CAN_BUSLOAD 62
des Transparent-Modus 73
DisableInterrupt 52
Disp_EnableDisplay 77
Disp_RegisterVariable 77, 82
EnableInterrupt 52
FileOpen 13
FileRead 13
GetDisplayInfo 12
ReadBitDirect 53
TimerInterruptEnable 50
Übersicht 76
Funktionsbausteine 61, 77
16 Bit Counter 47
32 Bit Counter 46
Disp_DisplayElement 78
Disp_DisplayPage 76, 79
Disp_GetDisplayInfo 78
Übersicht 76
Funktionstasten, Eingänge 12

G
Gerätefüße 19
Gewicht 95

H
Hauptmenü, Übersicht 31
Hintergrundbeleuchtung LCD einstellen 38

I
I/O-Zugriff, direkter 53
Impulsgeber 23
Inbetriebnahme 45
Initialwerte aktivieren 45
Inkrementalgeber 23
Inkrementalwertgeber 23
Inkrementalwertzähler 47
Installation 21
Interrupt 52
Interruptquelle 53
IP-Adresse 64
abfragen/ändern 66

K
Kanal parametrieren 72
Knotennummer 70
Kommunikation mit Zielsteuerung 71
Kommunikationskanal 64
Kommunikationsparameter 63
Konfiguration
EASY204-DP, EASY221-CO, EASY222-DN 89
EASY205-ASI 88
Ein-/Ausgänge 39
XIO-EXT121-1 39
Kontrast LCD einstellen 38
Kurzschluss 25
Kurzschlussüberwachung 13, 17

L
LCD 38
LED-Zustandsanzeige 13
Leitungslänge 93
Leitungsquerschnitte 93
Leitungsschutz 21

M
Menü
Eingabe von Werten 29
-führung 29
Sprache ändern 37
-struktur 31
Merkerbereich, Abbildung von easy800 auf EC4-200 . 90
MFD-Display 75
MMC 13
Montage
Hutschiene 19
Montageplatte 19
Multi-Funktions-Display 86
Multifunktions-Schnittstelle 14

N
Netzwerk
easyNet anschließen 93
-Kopplungen 87
-Kopplungen anschließen 20, 28
No Analog Output 39
No Counter 39
No Keys 39
Node-Id 70

O Ordnerfunktion ändern 39

P Parameter ändern 37
Passwort
aktivieren 35
ändern 36
einrichten 35
falsch 36
löschen 36
-schutz entfernen 36
vergessen 36
PC-Verbindung 27
Periodendauer 50
PING-Antwort 66
PLC-Browser 59
Programm
-bearbeitung 44
-erstellung, allgemeiner Ablauf 80
-routine 44
-start 43
-stopp 43
Programmieren über CAN-Netzwerk (Routing) 69
Programmiersoftware 9
Projektierung 21
Pufferzeit, Batterie 14

R Referenzierung 48
Relais Outputs 39
Relais-Ausgänge anschließen 24
Remanente Variablen 43
Reset 44
Routing
Voraussetzungen 69
Vorgehensweise 71

S Schnelle Zähler 45
Schnelle Zähler, Eingänge 12
Schnittstelle
CAN 16
easyLink 17
easyNet 16
Kommunikationsparameter festlegen 63
Multifunktion 14
Schraubmontage 19
Sensor (20 mA) anschließen 22
Servoventil anschließen 26
setrtc, Browserbefehl 60
Signalübersicht, Ein-/Ausgänge 47
Sollwertgeber anschließen 22
Sondermenü, Übersicht 32
Spannungsversorgung ausschalten/unterbrechen 43
Speichergrößen 41
Speicherkarte 13, 27
Start, Programm 43
START, Systemereignis 49
STARTUP.INI 67
Startverhalten 41, 67
Einstellen in der Programmiersoftware 43
Statusanzeige 30
STOP 43, 49
System
-ereignisse 44
-parameter einstellen 67
-uhr, Pufferung 14
-zeit 44

T Tastenfeld 29
TCP/IP-Verbindung (beim Routing) 69
Teilnehmer-Adresse
EASY204-DP, EASY221-CO, EASY222-DN 89
Temperatursensor anschließen 22
Testfunktionen 45
Timer-Interrupt 50
Transistor-Ausgänge
anschließen 25
Output-Typ in Konfiguration ändern 39
Transparent-Modus 14, 73
Typenübersicht, Steuerung 9

U Überlast 25
Uhrzeit einstellen 37
Unterbrechen der Spannungsversorgung 43

V Variablen
Verhalten beim Start 43
Verhalten nach Reset 45
Verbindungsaufbau PC – EC4-200 63
Verbindungskabel EASY800-PC-CAB 95
Versorgungsspannung anschließen 21

W WARMSTART 49
Werkseinstellung 38
wiederherstellen 44
Wippe 12
Wochentag einstellen 37

Z Zähler 23, 45
Zentrale Erweiterung 28
Zugriffszeiten erhöhen 46
Zustandsanzeige
an der LED 13
in der Programmiersoftware 45
Zwangssetzen, Variablen und I/Os 45
Zykluszeit überwachen 44