Betriebsanleitung

09/10 MN05013008Z-DE
ersetzt 02/08 AWB2528-1479D

EASY221-CO
CANopen-Slave-Anschaltung

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 2002, Redaktionsdatum 09/02
2. Auflage 2004, Redaktionsdatum 04/04
3. Auflage 2008, Redaktionsdatum 02/08
4. Auflage 2010, Redaktionsdatum 09/10
siehe Änderungsprotokoll im Kapitel „Zu diesem Handbuch"
© 2002 by Eaton Industries GmbH, 53105 Bonn

Autor: Ronny Happ, 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 50110-1/-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 60364-4-41 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 60204-1 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
	Zielgruppe	7
	Weitere Handbücher zum Gerät	7
	Referenzen	8
	Datentypen	8
	Gerätebezeichnung	9
	Abkürzungen und Symbole	11
	Lesekonventionen	12
1	**Zu EASY221-CO**	13
	Systemübersicht	14
	Aufbau des Gerätes	15
	Gerätefunktionsbeschreibung	16
	– easy600/700/800, MFD-CP8…/CP10…	16
	– easy800/MFD-CP8…/CP10…	17
	Hardware- und Betriebssystem-Voraussetzungen	17
	Sachwidriger Einsatz	18
2	**Installation**	19
	Anschluss EASY221-CO an das Basisgerät	19
	Versorgungsspannung anschließen	20
	CANopen anschließen	21
	– Anschlussbelegung CANopen	21
	– Abschlusswiderstände	22
	EMV-gerechte Verdrahtung	22
	Potentialtrennungen	23
	Übertragungsraten – automatische Baudraten-erkennung	24

3 Gerät betreiben 25
Erstes Einschalten 25
CANopen-Teilnehmeradresse einstellen 26
– Einstellen der Adresse am Basisgerät mit Anzeige 26
– Adresse mittels easySoft einstellen 28
– Adresse über spezielle Konfigurationstools einstellen 29
LED-Statusanzeigen 29
– Error-LED 29
– RUN-LED 30
– Zeitdiagramm der Zustände von Error- und RUN-LED 30
Zykluszeit easy-Basisgerät 31
EDS-Datei 31

4 CANopen-Dienste 33
Kommunikationsobjekte 33
– Servicedatenobjekte 33
– Prozessdatenobjekte 34
– PDO-Mapping 36
Systemdienste 37
– Synchronisations-Objekt 37
– Time Stamp-Objekt 37
– Emergency-Objekt 37
Netzwerkmanagement 38
– Initialisation 39
– Pre-Operational 39
– Operational 39
– Prepared 40
– Knotenüberwachung 42
Sonstige Dienste 43
– Speichern und Wiederherstellen von Einträgen 43
– Layer Setting Service 44
Geräteprofil 44

5 Objektverzeichnis 45
Kommunikationsparameter 45
Herstellerspezifische Objekte 54
Fehlermeldungen (Emergency) 55
– Drittes Datenbyte: Koppelmodul-Status 57

6 CANopen-Protokolle 59
PDO-Protokoll 59
SDO-Protokoll 60
– Allgemeine Übersicht 60
– SDO-Protokoll für easy/MFD 64
– Beispiel für easy600: Uhrzeit Lesen (7 Byte) 67
– Beispiel für easy800 : Uhrzeit lesen (8 Byte) 71
Emergency-Protokoll 75

7 PDO – Direkter Datenaustausch mit easy/MFD 77
Ausgangsdaten (2011_{hex}): Betriebsart, R1 – R16 78
Eingangsdaten (2012_{hex}): Betriebsart, S1 – S8 81

8 SDO – Steuerbefehle für easy600 85
Datum und Uhrzeit, Sommer-/Winterzeit lesen und schreiben 87
Abbilddaten lesen 91
– Generelles zum Arbeiten mit Abbilddaten 91
– Übersicht 91
– Digital-Eingänge, P-Tasten und Bedientasten 92
– Analog-Eingänge: I7 – I8 95
– Zeitrelais, Zählerrelais, Zeitschaltuhren, Analogwertvergleicher 96
– Hilfsrelais (Merker), Digital-Ausgänge, Textanzeige 99
Funktionsbausteine lesen/schreiben 102
– Übersicht 102
– Analogwertvergleicher A1 – A8: Istwerte schreiben (Funktion, Vergleichswerte) 103
– Zählerrelais C1 – C8: Istwert lesen 106
– Zählerrelais C1 – C8: Sollwert schreiben 108
– Zählerrelais C1 – C8: Sollwert lesen 110

– Zeitrelais T1 – T8: Istwert lesen (Zeitbereich, Istwert, Schaltfunktion) 112
– Zeitrelais T1 – T8: Parameter schreiben (Zeitbereich, Sollwert, Schaltfunktion) 116
– Zeitschaltuhr 1 – 4: Istwerte lesen (Kanal, ON-Zeit, OFF-Zeit) 120
– Zeitschaltuhr 1 – 4: Sollwerte schreiben (Kanal, ON-Zeit, OFF-Zeit) 124

9 SDO – Steuerbefehle für easy700 129
Datum und Uhrzeit lesen/schreiben 130
Abbilddaten lesen/schreiben 134
– Übersicht 134
– Analogwertvergleicher/Schwellwertvergleicher: A1 – A16 135
– Zähler: C1 – C16 136
– Textbausteine: D1 – D16 137
– Lokale Eingänge: I1 – I16 138
– Lokale Analog-Eingänge: IA1 – IA4 139
– Merker: M1 – M16/N1 – N16 schreiben 141
– Merker: M1 – M16/N1 – N16 lesen 143
– Betriebsstundenzähler: O1 – O4 145
– Lokale P-Tasten: P1 – P4 146
– Lokale Ausgänge: Q1 – Q8 148
– Ein-/Ausgänge von easyLink: R1 – R16/S1 – S8 149
– Zeitglieder: T1 – T16 151
– Jahresschaltuhr: Y1 – Y8 152
– Masterreset: Z1 – Z3 153
– Wochenschaltuhr: 1 – 8 154
Funktionsbaustein-Daten lesen/schreiben 155
– Allgemeine Hinweise 155
– Übersicht 155
– Analogwertvergleicher/Schwellwertschalter: A1 – A16 156
– Zählerrelais: C1 – C16 159
– Betriebsstundenzähler: O1 – O4 162
– Zeitrelais: T1 – T16 164
– Jahresschaltuhr: Y1 – Y8 167
– Wochenschaltuhr: 1 – 8 170
Analyse – Fehlercodes über easyLink 173

10 SDO – Steuerbefehle für easy800/MFD 175
Versionsgeschichte 176
Datum und Uhrzeit lesen/schreiben 177
– Winter-/Sommerzeit, Zeitumstellung 178
Abbilddaten lesen/schreiben 182
– Übersicht 182
– Lokale Analog-Eingänge: IA1 – IA4 183
– Lokale Diagnose: ID1 – ID16 185
– Lokale Eingänge: IW0 187
– Eingänge der Netzwerkteilnehmer: IW1 – IW8 189
– Merker: M… 190
– Lokale P-Tasten: P1 – P4 193
– Lokaler Analog-Ausgang: QA1 195
– Lokale Ausgänge: QW0/ Ausgänge der Netzwerkteilnehmer: QW1 – QW8 196
– Ein-/Ausgänge von easyLink: RW/SW 198
– Receive-Data Netz: RN1 – RN32/ Send-Data Netz: SN1 – SN32 200
Funktionsbaustein-Daten lesen/schreiben 202
– Allgemeine Hinweise 202
– Übersicht 202
– Analogwertvergleicher: A01 – A32 204
– Arithmetikbaustein: AR01 – AR32 206
– Block Compare: BC01 – BC32 208
– Block Transfer: BT01 – BT32 210
– Boolesche Verknüpfung: BV01 – BV32 212
– Zähler: C01 – C32 214
– Frequenzzähler: CF01 – CF04 216
– Schneller Zähler: CH01 – CH04 218
– Inkrementalzähler: CI01 – CI02 220
– Vergleicher: CP01 – CP32 222
– Textausgabe-Baustein: D01 – D32 224
– Datenbaustein: DB01 – DB32 227
– PID-Regler: DC01 – DC32 229
– Diagnose DG01…DG16 232
– Signalglättungsfilter: FT01 – FT32 234
– Empfang von Netzdaten: GT01 – GT32 236
– Wochenzeitschaltuhr: HW01 – HW32 238
– Jahreszeitschaltuhr: HY01 – HY32 241
– Bedingter Sprung JC01…JC32 244

– Wertskalierung: LS01 – LS32 246
– Masterreset: MR01 – MR32 248
– Datenmultiplexer MX01…MX32 250
– Zahlenwandler: NC01 – NC32 252
– Betriebsstundenzähler: OT01 – OT04 254
– Impulsausgabe PO01…PO02 256
– Senden von Netzdaten: PT01 – PT32 259
– Pulsweitenmodulation: PW01 – PW02 261
– Uhr synchronisieren: SC01 263
– Serielle Ausgabe SP01…SP32 264
– Schieberegister SR01…SR32 266
– Sollzykluszeit: ST01 269
– Zeitrelais: T01 – T32 271
– Tabellenfunktion TB01…TB32 274
– Wertbegrenzung: VC01 – VC32 276
Analyse – Fehlercodes über easyLink 278

11 Was ist, wenn…? 281
RUN-LED 281
Error-LED 282

Anhang 283
Technische Daten 283
Abmessungen 286

Glossar 287

Stichwortverzeichnis 301

Zu diesem Handbuch

Änderungsprotokoll

Gegenüber der letzten Ausgabe haben sich folgende wesentliche Änderungen ergeben:

Redaktions-datum	Seite	Stichwort	neu	Ände-rung	ent-fällt
04/04	allg.	easy700/800/MFD	✓		
	60	SDO-Protokoll		✓	
02/08	Kapitel 10	CANopen-Zugriff auf die geänderten/ neuen Funktionsbausteine der easy800 und MFD-CP8 ab Betriebssystem V 1.20	✓		
09/10	allg.	Umstellung auf Eaton-Bezeichnungen		✓	

Zielgruppe

Dieses Handbuch richtet sich an Automatisierungstechniker und Ingenieure. Es werden fundierte Kenntnisse zum Feldbus CANopen und zur Programmierung einer CANopen-Mastersteuerung vorausgesetzt. Außerdem sollten Sie mit der Handhabung des Steuerrelais easy bzw. des Multifunktionsdisplays MFD vertraut sein.

Weitere Handbücher zum Gerät

Grundsätzlich gelten die Bedienhandbücher zu:

- „Steuerrelais easy412, easy600“ (MN05013004Z-DE; frühere Bezeichnung AWB2528-1304-D)
- „Steuerrelais easy700“ (MN05013003Z-DE; frühere Bezeichnung AWB2528-1508D)
- „Steuerrelais easy800“ (MN04902001Z-DE; frühere Bezeichnung AWB2528-1423D)
- „Multifunktionsdisplay MFD-Titan“ (MN05002001Z-DE; frühere Bezeichnung AWB2528-1480D).

Alle Handbücher stehen im Internet als PDF-Datei zum Download zur Verfügung. Für ein schnelles Auffinden geben Sie unter http://www.eaton.com → Support als Suchbegriff die Dokumentationsnummer ein.

Referenzen

[1] CANopen – Application Layer and Communication Profile
CiA Draft Standard DS301
Version 4.01
June 1, 2000

[2] CANopen – Cabling and Connector Pin Assignment
CiA Draft Recommendation DR303-1
Version 1.0
October 10, 1999

[3] CANopen – Indicator Specification
CiA Draft Recommendation Proposal DRP303-3
Version 0.2
February, 22, 2001

[4] CANopen – Layer Setting Services and Protocol (LSS)
CiA Draft Standard Proposal DSP305
Version 1.0
May 31, 2000

Datentypen

CANopen spezifiziert seine eigenen Datentypen in [1] Kapitel 9.1 und 9.5.3. Für den CANopen Protocol Handler des EASY221-CO werden die in der folgenden Tabelle aufgeführten Datentypen genutzt.

Name	Beschreibung	Bereich	
		Minimum	Maximum
UNSIGNED8	Unsigned Integer der Länge 8 Bit (b7 bis b0)	0	255
UNSIGNED16	Unsigned Integer der Länge 16 Bit (b15 bis b0)	0	65535
UNSIGNED32	Unsigned Integer der Länge 32 Bit (b31 bis b0)	0	4294967295
VISIBLE_STRINGlen	Zeichenkette der Länge len. Die Zeichenkette muss nicht mit 0_{hex} terminiert sein!	Es sind alle ASCII-Zeichen von 20_{hex} bis $7E_{hex}$ sowie 0_{hex} zulässig	
DOMAIN	Anwenderspezifisches Datenformat		

Gerätebezeichnung

In diesem Handbuch werden für die Gerätetypen folgende Kurzbezeichnungen genommen, sofern die Beschreibung auf alle diese Typen zutrifft:

- easy600 für
  - EASY6…-AC-RC(X)
  - EASY6…-DC.-.C(X)
- easy700 für
  - EASY719-AB…
  - EASY719-AC…
  - EASY719-DA…
  - EASY719-DC…
  - EASY721-DC…
- easy800 für
  - EASY819-…
  - EASY820-…
  - EASY821-…
  - EASY822-…
- MFD-CP8… für
  - MFD-CP8-ME
  - MFD-AC-CP8-ME
  - MFD-CP8-NT
  - MFD-AC-CP8-NT
- MFD-CP10… für
  - MFD-CP10-ME
  - MFD-CP10-NT
- MFD-…-CP8/CP10… für
  - MFD-CP8…
  - MFD-CP10…

- easy-AB für
  - EASY719-AB…
- easy-AC für
  - EASY6…-AC-RC(X),
  - EASY719_AC
  - EASY8…-AC-…
- easy-DC für
  - EASY6…-DC-…
  - EASY719-DC-…
  - EASY8…-.DC-…
- easy-DA für
  - EASY719-DA…

Abkürzungen und Symbole

Dieses Handbuch verwendet Abkürzungen und Symbole, die folgende Bedeutung haben:

BCD	**B**inär **C**odierter **D**ezimalcode
bin	Binärwert
CAL	**C**AN **A**pplication **L**ayer
CAN	**C**ontroller **A**rea **N**etwork
COB	**C**ommunication **Ob**ject
COB ID	**C**ommunication **Ob**ject **Id**entifier
COV	**C**hange **o**f **V**alue
dez	Dezimal (Zahlensystem mit der Basis 10)
EDS	**E**lectronic **D**ata **S**heets
EMCY	**Em**ergen**cy** Objekt
hex	Hexadezimal (Zahlensystem mit der Basis 16)
ID	**Id**entifier
len	length (Länge)
LSS	**L**ayer **S**etting **S**ervice
NMT	**N**etwork **M**anagemen**t**
NVM	**N**on-**V**olatile **M**emory (Permanenter Speicher)
NVM-PA	**N**on-**V**olatile **M**emory-**Pa**rameter (Bereich für Speichern und Laden)
NVM-RO	**N**on-**V**olatile **M**emory-**R**ead **O**nly (Bereich kann nur gelesen werden)
PC	**P**ersonal **C**omputer
PDO	**P**rocess **D**ata **O**bject
ro	**R**ead **O**nly (nur Lesezugriff)
ROM	**R**ead **O**nly **M**emory
RTR	**R**emote **T**ransmit **R**equest (Fern-Abfrage)
rw	**R**ead/**W**rite (Lese- und Schreibzugriff)
SELV	**S**afety **E**xtra **L**ow **V**oltage (Sicherheitskleinspannung)
SDO	**S**ervice **D**ata **O**bject
WE	**W**erks**e**instellung

Lesekonventionen

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.

▶ zeigt Handlungsanweisungen an.

Achtung!
warnt vor leichten Sachschäden.

Warnung!
warnt vor schweren Sachschäden und leichten Verletzungen.

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

macht Sie aufmerksam auf interessante Tipps und Zusatzinformationen

1 Zu EASY221-CO

Das Kommunikationsmodul EASY221-CO wurde für Automatisierungsaufgaben mit dem Feldbus CANopen entwickelt. Das EASY221-CO stellt ein Gateway dar und kann nur in Verbindung mit den erweiterbaren Steuerrelais easy600, easy700, easy800 oder MFD-Basisgeräten betrieben werden. Die Systemeinheit, bestehend aus dem Steuergerät easy/MFD und dem CANopen-Gateway, agiert im Feldbussystem ausschließlich als Slave-Teilnehmer.

Systemübersicht

Die easy-CANopen-Slaves werden innerhalb eines CANopen-Feldbus-Systems integriert.

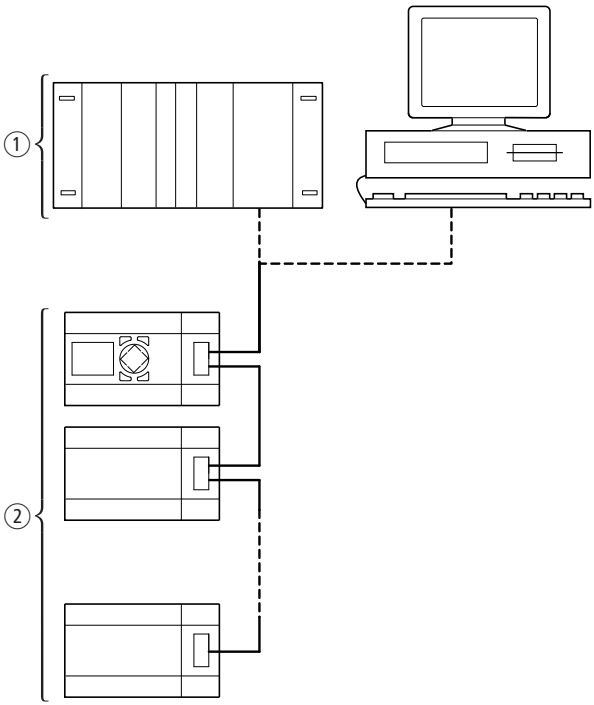


Abbildung 1: Einbindung EASY221-CO im CANopen-Netzwerk

① Masterbereich, SPS (z. B.: XC600) oder PC mit CAN-Karte

② Slave-Bereich z. B.: Steuerrelais easy mit CANopen-Anschaltung

Aufbau des Gerätes

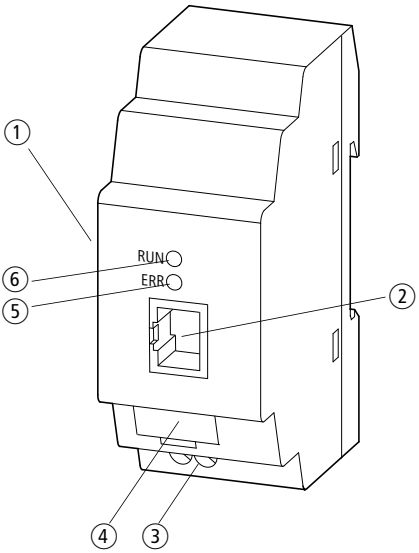


Abbildung 2: Aufbau EASY221-CO

① easyLink-Buchse
② CANopen-Anschluss, 8-polige RJ-45-Buchse
③ Spannungsversorgung 24 V ⎓
④ Gerätekennzeichnungsschild
⑤ ERR-LED (Error)
⑥ RUN-LED

Gerätefunktions-beschreibung

Das Modul EASY221-CO ermöglicht der easy- bzw. MDF-Familie die Anbindung an ein CANopen-Kommunikations-netzwerk. Mit der entsprechenden SDO/PDO-Auswahl können die nachfolgenden Daten übertragen werden:

easy600/700/800, MFD-CP8…/CP10…

- S1 bis S8
  Ausgangsdaten des Basisgerätes, RUN/STOP
  (Lesen, vom CANopen-Master aus gesehen)
- R1 bis R16
  Eingangsdaten des Basisgerätes, RUN/STOP
  (Schreiben, vom CANopen-Master aus gesehen)
- Alle Daten der Funktionsrelais
  (Lesen, vom CANopen-Master aus gesehen)
  - Zeitrelais
  - Zählerrelais
  - Schaltuhr
  - Analogwertvergleicher
  - Wochentag, Uhrzeit, Sommer-/Winterzeit
  - Alle Zustände der easy600-Kontakte

- Die Sollwerte der Funktionsrelais
  (Schreiben, vom CANopen-Master aus gesehen)
  - Zeitrelais
  - Zählerrelais
  - Schaltuhr
  - Analogwertvergleicher
  - Wochentag, Uhrzeit, Sommer-/Winterzeit

easy800/MFD-CP8…/CP10…

- sämtliche Merker und easyNet-Daten
- Funktionsbausteine
  (Lesen/Schreiben aus Sicht des Masters)
  - Arithmetikbausteine
  - Frequenzzähler, schnelle Zähler, Inkrementalzähler
  - Wochen- und Jahreszeitschaltuhr
  - Betriebsstundenzähler
  - PID-Regler
  - PWM (Pulsweitenmodulation)
  - Echtzeituhr.

Hardware- und Betriebssystem-Voraussetzungen

Das Erweiterungsgerät EASY221-CO arbeitet mit den Basisgeräten easy600, easy700, easy800 und MFD ab den nachfolgenden Betriebssystemversionen zusammen:

Basisgerät		**Erweiterungsgerät EASY221-CO**	
Geräteversion	BTS-Version	Geräteversion = 02	Geräteversion = 03
easy600			
= 04	ab 2.4	×	×
easy700			
= 01	ab 1.01.xxx	–	×
easy800			
= 04	ab 1.10.xxx	–	×
MFD-CP8…/CP10…			
= 01	ab 1.10.xxx	–	×

Die Geräteversion des entsprechenden Basis- oder Erweiterungsgerätes ist auf der rechten Gehäuseseite angegeben.
Beispiel: EASY221-CO: 03-228xxxxxxx (03 = Geräteversion)
Die Betriebssystemversion (BTS) des entsprechenden Basisgerätes können Sie über die easySoft auslesen. Bei den Geräten easy700, easy800 und MFD-CP8… besteht auch die Möglichkeit, die Information direkt am Gerät auszulesen. Lesen Sie hierzu bitte im entsprechenden Handbuch nach.
Einen Überblick über die Änderungen und Neuerungen der verschiedenen Geräteversionen des easy800 finden Sie auf Seite 176.

Sachwidriger Einsatz

easy darf nicht eingesetzt werden als Ersatz für sicherheitsrelevante Steuerungen wie:

- Brenner-,
- NOT-AUS-,
- Kran- oder
- Zweihand-Sicherheitssteuerungen.

2 Installation

Es gelten die gleichen Richtlinien wie für die easy600, easy700, easy800 und MFD-Basisgeräte mit Erweiterung.

Anschluss EASY221-CO an das Basisgerät

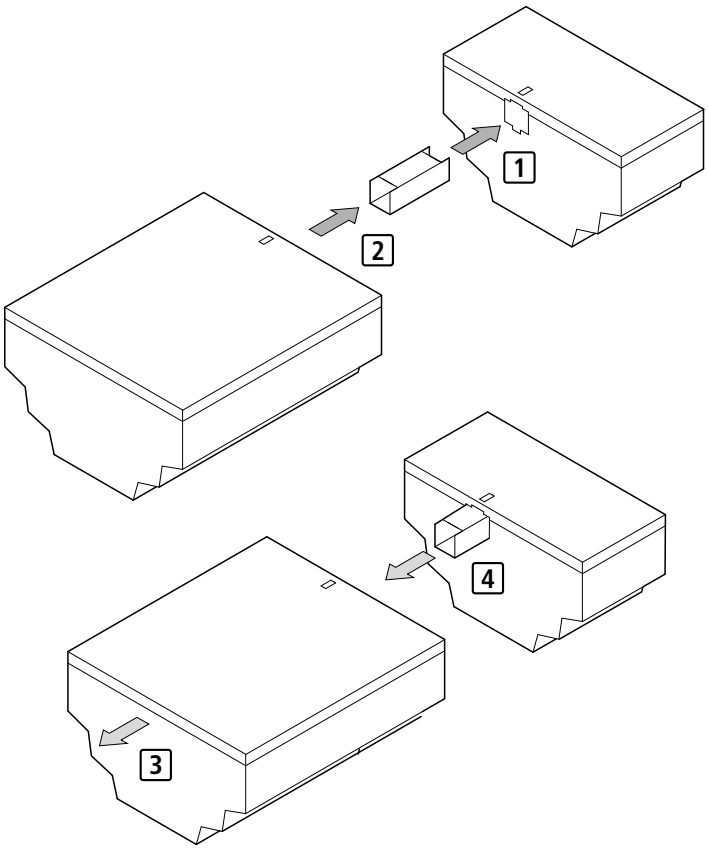


Abbildung 3: EASY221-CO an das Basisgerät montieren

1 +2 Einbau

3 +4 Ausbau

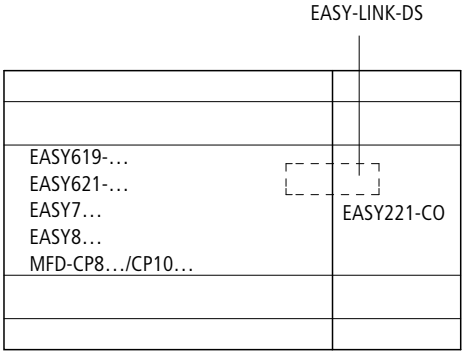


Abbildung 4: Verbindung zwischen Basisgerät und EASY221-CO

Versorgungsspannung anschließen

Das Gerät EASY221-CO wird mit einer 24-V-DC-Versorgungsspannung betrieben (→ Technische Daten unter „Stromversorgung", Seite 285).

Gefahr!
Bei der 24-V-Versorgung ist auf eine sichere elektrische Trennung der Kleinspannung (SELV) zu achten.

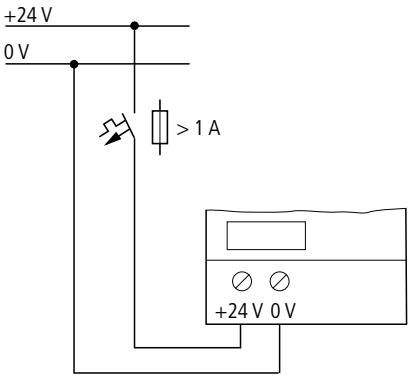


Abbildung 5: Versorgungsspannung EASY221-CO

CANopen anschließen

Das zu verwendende Kabel, die Verbindungstücke, als auch die Abschlusswiderstände sind in der ISO 11898 spezifiziert.

Für das EASY221-CO wird ein geschirmter 8-poliger RJ-45-Stecker verwendet. Die Belegung des Steckers ist sowohl in der CiA DR-303-1 als auch im Folgenden aufgeführt.

Anschlussbelegung CANopen

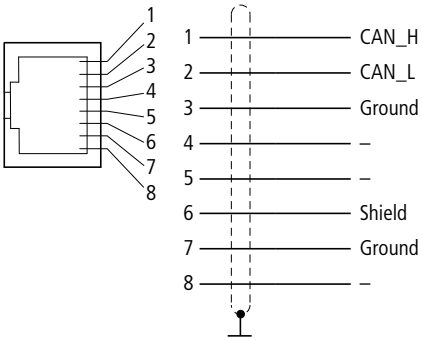


Abbildung 6: Anschlussbelegung Gerätebuchse

Pin	Signal	Beschreibung
1	CAN_H	CAN-Bus-Signal (dominant high)
2	CAN_L	CAN-Bus-Signal (dominant low)
3, 7	CAN_GND	CAN-Masse (Ground)
6	CAN_SHILD	Optionaler Schirm
4, 5, 8	–	nicht belegt

Abschlusswiderstände

Der erste und der letzte Teilnehmer in einem CANopen-Netzwerk muss mit einem Busabschlusswiderstand von 120 Ω abgeschlossen sein. Dieser wird zwischen CAN_H und CAN_L geschaltet.

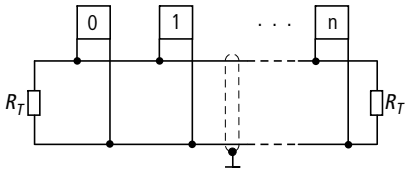


Abbildung 7: Abschlusswiderstände R_T:
Anschluss CAN_H und CAN_L

R_T = 120 Ω

EMV-gerechte Verdrahtung

Durch eventuelle elektromagnetische Beeinflussung des Kommunikationsfeldbusses, können unerwünschte Störungen auftreten. Diese lassen sich durch die Verwendung des oben beschriebenen geschirmten Kabels, eines geschirmten RJ-45-Steckers und natürlich durch Auflegen des Schirms wesentlich verringern.

Die folgenden zwei Abbildungen stellen das richtige Auflegen der Abschirmung dar.

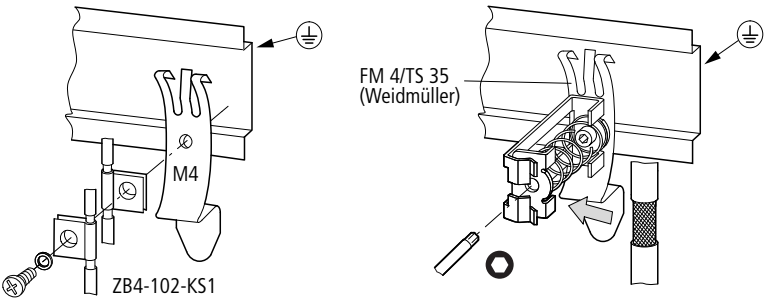


Abbildung 8: Schirmbefestigung auf Hutschiene

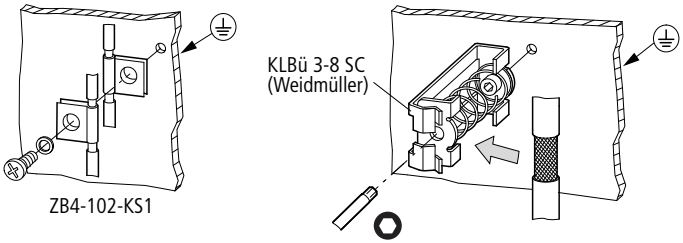


Abbildung 9: Schirmbefestigung auf Montageplatte

Potentialtrennungen

Für die Schnittstellen des EASY221-CO gelten folgende Potentialtrennungen:

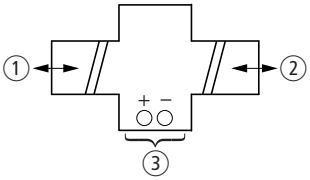


Abbildung 10: Potentialtrennung zwischen Versorgungsspannung und Ausgängen

① sichere elektrische Trennung zwischen easyLink und 240 V ~

② einfache elektrische Trennung zum CANopen-Kommunikationsbus

③ Versorgungsspannung 24 V DC

Übertragungsraten – automatische Baudraten-erkennung

Nach dem Einschalten erkennt das Modul EASY221-CO automatisch die im Kommunikationsnetzwerk verwendete Übertragungsrate. Hierzu muss allerdings mindestens ein Teilnehmer im Netzwerk gültige Nachrichten verschicken. In diesem Zustand blinken die Error- und die RUN-LED des EASY221-CO schnell.

Wenn eine korrekte CANopen-Nachricht empfangen wurde, gilt die verwendete und somit eingestellte Baudrate als korrekt und das Gerät sendet eine BootUp-Nachricht. Die RUN-LED wechselt in den blinkenden Zustand und die Error-LED verlischt.

EASY221-CO unterstützt die von der CiA spezifizierten Übertragungsraten. Die folgende Tabelle gibt einen Überblick zur empfohlenen Bitrate und der entsprechenden maximalen Leitungslänge.

Bitrate **kBaud**	**Maximale Leitungslänge** **m**	**Empfohlener Leitungs-querschnitt** **mm^2**
10	5000	> 0,8
20	2500	> 0,8
50	1000	0,75 bis 0,8
100	650	0,34 bis 0,6
125	500	0,34 bis 0,6
250	250	0,34 bis 0,6
500	100	0,25 bis 0,34
800	50	0,25 bis 0,34
1000	25	0,25 bis 0,34

3 Gerät betreiben

Erstes Einschalten

- ▶ Prüfen Sie vor dem Einschalten, ob die Stromversorgung, der Busanschluss und die Verbindung zum Basisgerät richtig angeschlossen sind.
- ▶ Schalten Sie die Versorgungsspannung für das Basisgerät und EASY221-CO ein.

Die LEDs des EASY221-CO flimmern. Das Gerät befindet sich im Modus zum Erkennen der richtigen Baudrate (→ Abschnitt „Übertragungsraten – automatische Baudratenerkennung“ auf Seite 24). Am Basisgerät werden die GW-Information (intelligenter Teilnehmer angekoppelt) angezeigt.

Basisgerät	Geräteversion	GW-Anzeige
easy600	04	statisch
easy700	ab 01	blinkend
easy800	04	statisch
	ab 05	blinkend
MFD-CP8…	01	statisch
	ab 02	blinkend
MFD-CP10…	ab 01	blinkend

Sobald das Gerät im Netzwerkmanagement in den Status „Operational“ geschaltet wird, wechselt der Zustand des GW auch bei den Geräten mit blinkenden GW auf statisch, → Abschnitt „Netzwerkmanagement“ auf Seite 38).

Hat EASY221-CO die Werkseinstellungen (Node ID = 127), so müssen Sie die CANopen-Teilnehmeradresse einstellen.

CANopen-Teilnehmeradresse einstellen

Jeder CANopen-Teilnehmer benötigt eine eindeutige Adresse (Node ID) in der CANopen-Struktur. In einer CANopen-Struktur können Sie maximal 127 Adressen (1 bis 127) vergeben. Jede Node ID darf in der gesamten Busstruktur nur einmal vergeben werden.

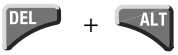
Die CANopen-Adresse können Sie bei EASY221-CO auf drei Arten einstellen:

- Mit der integrierten Anzeige und Tastatur am „easy"- oder MFD-Titan-Basisgerät; Adressbereich: 1 bis 127
- Mittels easySoft Version 3.01 oder höher am PC
- Über die Konfigurationssoftware der verwendeten Master-Steuerung (eventuell unter Verwendung einer expliziten Nachricht).

Einstellen der Adresse am Basisgerät mit Anzeige

Voraussetzungen:

- Das entsprechende Basisgerät (easy600, easy700, easy800 oder MFD-Titan) und EASY221-CO sind mit Spannung versorgt.
- Das Basisgerät ist aufgeschlossen (kein Passwortschutz aktiviert).
- Das Basisgerät besitzt eine gültige Betriebssystemversion.
- Das Basisgerät muss im STOP-Modus sein.



▶ Wechseln Sie durch gleichzeitiges Betätigen von DEL + ALT in das Sondermenü.

```
PASSWORT...
SYSTEM...
GB D F E I
KONFIGURATOR
```

```
PASSWORT...
SYSTEM...
GB D F E I
KONFGURATOR
```

▶ Wechseln Sie mit den Cursor-Tasten ∧ oder ∨ zu KONFIGURATOR.

▶ Bestätigen Sie mit OK.

```
NET...
LINK...
```

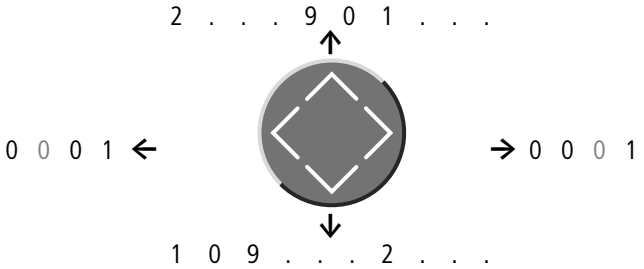
▶ Wählen Sie bei den Geräten easy800/MFD den Menüpunkt LINK…

▶ Bestätigen Sie mit OK.

Es erscheint das Menü CANOPEN.

```
CANOPEN
(MAX 127)
NODE ID 0127
221-01.20- B
```

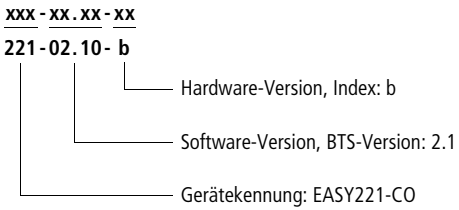
▶ Stellen Sie die Adresse mit den Cursor-Tasten ein:
- Den Wert der aktuellen Ziffer stellen Sie mit ∧ oder ∨ ein.
- Die aktuelle Ziffer wechseln Sie mit < oder >.



▶ Übernehmen Sie die Adresse mit OK.

▶ Mit ESC brechen Sie die Adresseingabe ab.

Information zur 4. Displayzeile:



Adresse mittels easySoft einstellen

Bei easySoft, Version 3.1
‹Menü → Online → Erweiterungsgeräte konfigurieren›

Bei easySoft, ab Version 4.01
‹Menü → Kommunikation → Konfiguration → Erweiterungsgeräte → EASY221-CO›.

Das Menü ist nur in der Kommunikationsansicht verfügbar; aktivieren Sie deshalb die Registerkarte „Kommunikation".

Für Geräte der Geräteversionskennung 01 gilt:

Nachdem Sie die NodeID über das Basisgerät abgeändert haben, müssen Sie EASY221-CO neu starten. Hierzu schalten Sie die Versorgungsspannung aus und wieder ein. EASY221-CO-Geräte mit einer Versionskennung > 01 übernehmen die Adresse automatisch.

Adresse über spezielle Konfigurationstools einstellen

Eine weitere Möglichkeit zum Einstellen oder Ändern der Node ID des Gateway besteht über spezielle Konfigurationstools, mit denen das CANopen-Netzwerk generell konfiguriert werden kann. Das Gateway unterstützt zu diesem Zweck den Dienst des LSS (Layer-Setting-Services).

LED-Statusanzeigen

Das Erweiterungsgerät EASY221-CO besitzt zwei LED-Anzeigeelemente, eine grüne RUN-LED und eine rote Error-LED. Diese geben Ihnen den momentanen Modulzustand an sowie eine Unterstützung zur schnellen Problemanalyse.

Error-LED

Nr.	Error-LED	Status	Beschreibung
1	AUS	Kein Fehler	EASY221-CO arbeitet korrekt. Ist die RUN LED ebenfalls aus, so ist EASY221-CO entweder ausgeschaltet oder befindet sich im Reset-Zustand.
2	Einmal aufblitzen	Alarmgrenze erreicht	Mindestens ein Fehlerzähler des CANopen-Controllers hat das „Warning Limit" erreicht oder überschritten. Es sind zu viele Fehler auf dem CANopen-Bus aufgetreten.
3	Flimmernd	AutoBaud/LSS	Die Auto Baudrate Erkennung läuft gerade (flimmert abwechselnd mit der RUN-LED).
4	Zweimal aufblitzen	Fehlerkontrollereignis (Error Control Event)	Ein Schutzereignis durch „Guard Event" oder ein Lebenszeichenereignis durch „Heartbeat Event" ist aufgetreten.
5	EIN	Bus ausgeschaltet	Der CANopen-Controller hat in den BUS-OFF Status gewechselt.

RUN-LED

Nr.	RUN-LED	Status	Beschreibung
1	AUS	Reset	EASY221-CO ist entweder ausgeschaltet oder befindet sich im Reset-Zustand.
2	Flimmernd	AutoBaud	Die Auto Baudrate Erkennung läuft gerade (flimmert abwechselnd mit der Error-LED).
3	Einmal aufblitzen	STOPPED[1)]	Das Gerät befindet sich im Zustand STOPPED.
4	Blinkend	PRE-OPERATIONAL[1)]	Das Gerät befindet sich im Zustand PRE-OPERATIONAL.
5	EIN	OPERATIONAL[1)]	Das Gerät befindet sich im Zustand OPERATIONAL.

1) Detailliertere Erläuterungen zu den verschieden Stadien finden Sie im Abschnitt „Netzwerkmanagement", Seite 38.

Zeitdiagramm der Zustände von Error- und RUN-LED

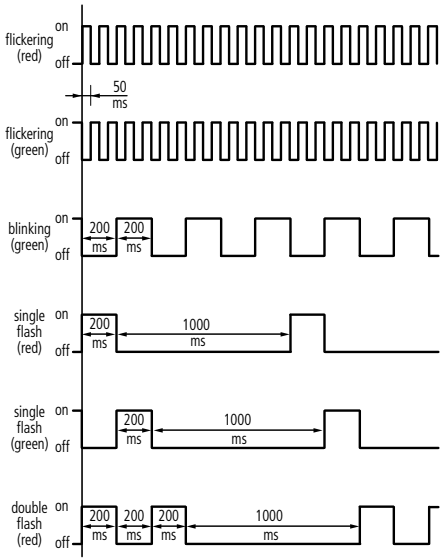


Abbildung 11: Error- und RUN-LED

Zykluszeit easy-Basisgerät

Die Kommunikation zwischen dem Basisgerät und EASY221-CO über den Datenstecker easyLink verlängert die Zykluszeit des Basisgerätes.

Im Extremfall kann sich die Zykluszeit um 25 ms verlängern.

Bitte beachten Sie dieses bei den Reaktionszeiten des Basisgerätes.

EDS-Datei

EASY221-CO kann in die CANopen-Struktur mit Hilfe einer standardisierten EDS-Datei eingebunden werden (Electronic Data Sheet = Elektronisches Datenblatt). Das EDS beschreibt die Funktionalität eines CANopen-Gerätes in maschinenlesbarer Form. In dem EDS sind alle Objekte aufgeführt, die unterstützten Baudraten, der Hersteller und viele weitere Angaben.

Die jeweils aktuelle Version der EDS-Datei ist direkt bei Eaton erhältlich. Ein weitere Möglichkeit zum Aktualisieren besteht durch Herunterladen der Datei von der Eaton-Homepage:

http://www.eaton.com → Support → Suche „EASY221-CO“

4 CANopen-Dienste

Die funktionelle Beschreibung zur Ansteuerung des EASY221-CO am CANopen-Bus erfolgt anhand der CANopen-Dienste.

Kommunikationsobjekte

EASY221-CO unterstützt Servicedatenobjekte und Prozessdatenobjekte des CANopen Pre-defined Connection Set.

Servicedatenobjekte

Servicedatenobjekte (SDO – Service Data Object) dienen dem lesenden und schreibenden Zugriff auf die Einträge des Objektverzeichnisses.

Server SDO
Es wird die erste Server SDO unterstützt; damit kann auf das lokale Objektverzeichnis lesend und schreibend zugegriffen werden.

EASY221-CO unterstützt den Expedited Transfer (für bis zu 4 Datenbyte) und den Segmented Transfer (für mehr als 4 Datenbyte).

→ Der Blocktransfer wird nicht unterstützt!

Der Ablauf ist im Abschnitt „PDO-Protokoll“, Seite 59, detaillierter beschrieben.

Client SDO
Die Client SDOs dienen dazu, in das Objektverzeichnis eines anderen CANopen-Gerätes im Netzwerk Daten zu schreiben oder zu lesen.

→ Die Client SDOs werden vom EASY221-CO nicht unterstützt!

Prozessdatenobjekte

Der Austausch von Prozessdaten erfolgt in CANopen mit Hilfe von Prozessdatenobjekten (PDO – Process Data Object). Der Ablauf ist im Abschnitt „Herstellerspezifische Objekte“, Seite 54, detaillierter beschrieben.

Die Prozessdaten mit den zugehörigen PDOs sind in der folgenden Tabelle aufgeführt.

PDO	Prozessdaten	Länge
Empfang-PDO	Befehl oder Kennung für die Abbilddaten R16 bis R1 vom easy/MFD-Basisgerät (Ausgangsdaten zum easy)	3 Byte
Sende-PDO	Befehl oder Status für Abbilddaten S8 bis S1 vom easy/MFD-Basisgerät (Eingangsdaten vom easy)	3 Byte

→ Die detaillierte Zusammensetzung der Prozessdaten finden Sie im Abschnitt „Herstellerspezifische Objekte“, Seite 54.

Empfang-PDO
EASY221-CO erhält Daten per Empfang-PDO vom CANopen-Netzwerk (PDO Consumer) und schreibt diese über easyLink als Befehl oder Kennung für die Abbilddaten R16 bis R1 zum easy/MFD-Basisgerät.

Sende-PDO
Umgekehrt werden Befehl oder Status der Abbilddaten S8 bis S1 vom easy/MFD-Basisgerät über easyLink gelesen und als Sende-PDO des EASY221-CO dem CANopen-Netzwerk zur Verfügung gestellt (PDO Producer).

PDO-Mapping
EASY221-CO unterstützt **statisches PDO-Mapping**, hierbei werden die Prozessdaten den einzelnen PDOs fest zugeordnet, die Granularität beträgt 1 Byte. Das PDO-Mapping ist fest gespeichert und kann von Ihnen nicht verändert werden.

Übertragungsarten der PDOs
Empfang-PDO:
Die Übertragungsart (Transmission Type) der Empfang-PDO wird per Default auf „Asynchron" eingestellt (Wert: $255_{dez} = FF_{hex}$).

Sende-PDO:
Die Übertragungsart (Transmission Type) der Sende-PDO wird per Default auf „Asynchron" eingestellt (Wert: $255_{dez} = FF_{hex}$).

Inhibit Time
Die Inhibit Time wird nur bei Transmit PDOs ausgewertet. Die Inhibit Time ist die Sperrzeit zwischen zwei Transmit PDOs und wird in Schritten von 100 µs angegeben. Der übergebene Wert wird auf die nächst kleinere Millisekunde abgerundet. Werte kleiner 1 ms werden als 0 gespeichert. In diesem Fall überträgt das Modul die PDOs so schnell wie möglich.

Eine Inhibit Time wird per Default nicht aktiviert, da die über das easyLink-Protokoll übertragenen Daten nur alle 180 ms aktualisiert werden. Es besteht jedoch die Möglichkeit, das Definieren einer Inhibit Time bei Bedarf für die Sende-PDO zu aktivieren.

Event Timed PDOs
Das Ablaufen eines Zählers kann als Ereignis betrachtet werden, woraufhin eine PDO gesendet wird. Event Timed PDOs werden bei EASY221-CO per Default nicht unterstützt, können jedoch bei Bedarf für die Sende-PDO aktiviert werden.

Multiplexed PDOs
Multiplexed PDOs enthalten neben den reinen Prozessdaten eine Adressinformation bestehend aus Index und Subindex, mit der die PDO an eine bestimmte Adresse im Objektverzeichnis des konsumierenden Gerätes geschrieben werden kann.

→ EASY221-CO unterstützt keine Multiplexed PDOs.

PDO-Mapping

Die Prozessdaten werden einem Empfang- und einem Sende-PDO folgendermaßen zugeordnet.

1. Empfang-PDO:
Die folgende Tabelle zeigt das Mapping der 1. Empfang-PDO.

Datenbyte	Inhalt	Beschreibung
1. Datenbyte	Zyklischer Befehl und Kennung	Eingangsdaten des easy/MFD-Basisgerät schreiben (aus Sicht des Masters) (Index 2011, Subindex 00_{hex})
2. Datenbyte	Abbilddaten R16 bis R9	
3. Datenbyte	Abbilddaten R8 bis R1	
4. bis 8. Datenbyte	Wird nicht übertragen	

→ Die detaillierte Zusammensetzung der Prozessdaten finden Sie im Abschnitt „Ausgangsdaten (2011hex): Betriebsart, R1 – R16“, Seite 78.

2. bis 4. Empfang-PDO:
Die 2. bis 4. Empfang-PDO des Pre-defined Connection Set werden nicht unterstützt.

1. Sende-PDO:
Die folgende Tabelle zeigt das Mapping der 1. Sende-PDO.

Datenbyte	Inhalt	Beschreibung
1. Datenbyte	Zyklischer Befehl und Status	Ausgangsdaten vom easy/MFD-Basisgerät lesen (aus Sicht des Masters) (Index 2012_{hex}, Subindex 00_{hex})
2. Datenbyte	Abbilddaten S8 bis S1	
3. Datenbyte	leer (00_{hex})	
4. bis 8. Datenbyte	Wird nicht übertragen	

→ Die detaillierte Zusammensetzung der Prozessdaten finden Sie im Abschnitt „Eingangsdaten (2012hex): Betriebsart, S1 – S8“, Seite 81.

2. bis 4. Sende-PDO:
Die 2. bis 4. Sende-PDO des Pre-defined Connection Set werden nicht unterstützt.

Systemdienste

Synchronisations-Objekt

EASY221-CO unterstützt als Consumer das Synchronisations-Objekt (Index: 1005_{hex}), um die synchrone Übertragung der PDOs zu ermöglichen.

Time Stamp-Objekt

Mit Hilfe des Time Stamp-Objekts (Index: 1012_{hex}) stellt ein Zeit-Producer allen anderen Teilnehmern im System eine gemeinsame Zeitreferenz zur Verfügung.
Das Time Stamp-Objekt wird vom EASY221-CO nicht unterstützt.

Emergency-Objekt

EASY221-CO unterstützt das Emergency-Objekt (Index: 1014_{hex}), um das Netzwerk über aufgetretene Fehler im Gerät zu unterrichten. Der Inhalt der Emergency-Nachricht richtet sich nach dem aufgetretenen Fehler. Die erkannten Fehler sind im Abschnitt „Fehlermeldungen (Emergency)“, Seite 55, beschrieben.

Netzwerkmanagement

In einem CANopen-Netzwerk gibt es immer nur einen NMT-Master (NMT = Network Management), alle anderen Geräte sind NMT-Slaves. Der NMT-Master hat die komplette Kontrolle über alle Geräte und kann deren Zustand verändern.

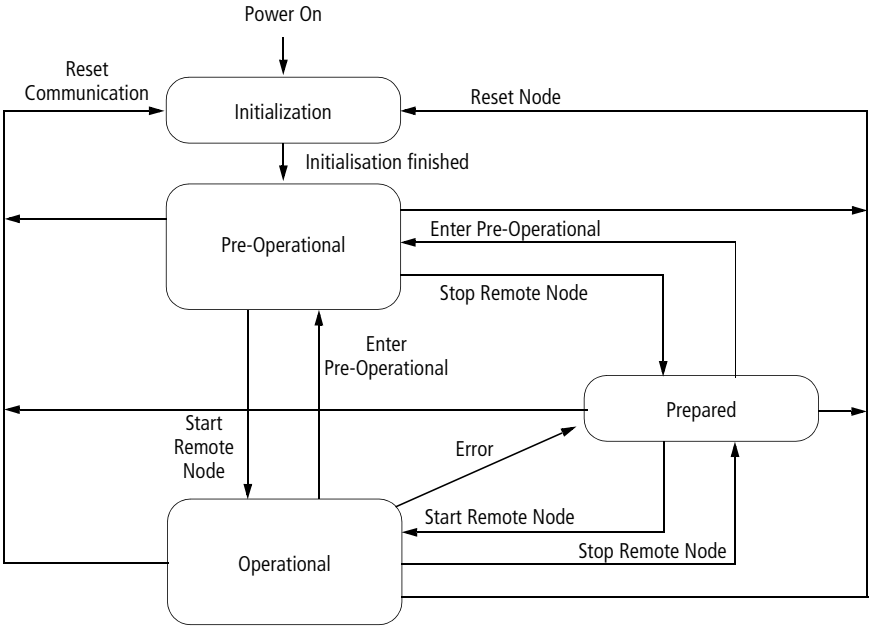


Abbildung 12: Netzwerkmanagement

Es werden folgende Zustände unterschieden:

- Initialisation,
- Pre-Operational,
- Operational und
- Prepared.

Initialisation

Dies ist der Zustand, den der Knoten nach dem Einschalten durchläuft. Innerhalb dieser Phase erfolgt die Autobaud-Erkennung, Initialisierung der Geräte-Applikation sowie der Geräte-Kommunikation. Anschließend geht der Knoten selbstständig in den Zustand „Pre-Operational".

Pre-Operational

In diesem Zustand kann mit dem Knoten über SDOs kommuniziert werden (z. B. Einstellen der Guard-Time, Lifetime-Faktor). Der Knoten ist nicht in der Lage, eine PDO-Kommunikation durchzuführen und sendet auch keine Emergency-Botschaften aus.

Die RUN-LED des EASY221-CO signalisiert diesen Zustand blinkend.

Ab Geräteversion 03 wird der Zustand außerdem am Basisgerät mit der blinkenden Status-Anzeige GW dargestellt. Der Diagnoseeingang I14 (am Basisgerät) bleibt so lange gesetzt, bis GW am Display statisch dargestellt ist. Dies wird erreicht, indem der CAN-Knoten in den Operational-Mode gesetzt wird.

Operational

In diesem Zustand hat der CANopen-Knoten die volle Betriebsbereitschaft und kann selbstständig Nachrichten übertragen (PDOs, Emergency).

Die RUN-LED des EASY221-CO signalisiert diesen Zustand statisch.

Ab Geräteversion 03 wird der Zustand außerdem am Basisgerät mit der statischen Status-Anzeige GW dargestellt. Der Diagnoseeingang I14 (am Basisgerät) wird logisch Null gesetzt.

Prepared

In diesem Zustand wird ein Knoten vollständig vom Netz abgeschaltet, es sind weder SDO- noch PDO-Kommunikation möglich. Der Knoten kann nur durch ein entsprechendes Netzwerk-Kommando (z. B. Start-Node) in einen anderen Netzwerk-Zustand gebracht werden.

Um zu signalisieren, dass ein Gerät nach dem Einschalten betriebsbereit ist, wird eine sogenannte „Boot-Up Message" ausgesendet. Diese Botschaft benutzt den Identifer des NMT-Error-Control Protokolls und ist fest an die eingestellte Geräteadresse gebunden (1792_{dez} + Geräteadresse).

Für den PDO- und SDO-Transfer lesen bitte auch die in Abschnitt „PDO-Protokoll" auf Seite 59 angeführten Informationen.

Um einen Prozessdatenaustausch mit Hilfe von PDOs durchzuführen, ist das Modul mit dem Service „Start Remote Node" in den Zustand OPERATIONAL zu überführen. Sind TxPDOs mit dem Transmission Type 254 oder 255 konfiguriert, werden mit jedem Wechsel in den Zustand OPERATIONAL die PDOs gesendet. Diese PDOs werden unabhängig von einem Eingangsdatenwechsel gesendet.

Tritt ein Fehler auf, geht das Modul in den Zustand PREPARED. In diesem Zustand ist keine Kommunikation über SDOs und PDOs möglich. Das Modul reagiert nur auf die NMT Services:

- Start Remote Node, Übergang in den Zustand OPERATIONAL, Datenübertragung über SDOs und PDOs ist möglich.
- Enter Pre-Operational, Übergang in den Zustand PRE-OPERATIONAL, Datenübertragung über SDOs ist möglich.
- Reset Node und
- Reset Communication, Übergang in den Zustand INITIALISATION, es werden die zuletzt gespeicherten Einstellungen, bzw. wenn nichts gespeichert wurde, die werkseitigen Standardeinstellungen geladen. Danach geht das Modul in den Zustand PRE-OPERATIONAL.

Aufbau der NMT Services:

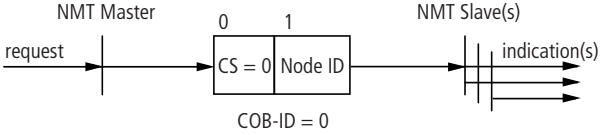


Abbildung 13: Aufbau der NMT Services, Start Remote Node

Node ID = 0: Alle vorhandenen Knoten werden in den Zustand OPERATIONAL überführt.

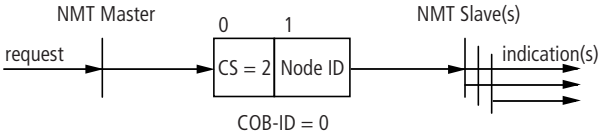


Abbildung 14: Aufbau der NMT Services, Stop Remote Node

Node ID = 0: Alle vorhandenen Knoten werden in den Zustand PREPARED überführt.

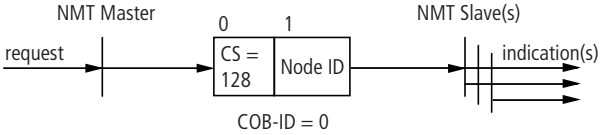


Abbildung 15: Aufbau der NMT Services, Pre Operational State

Node ID = 0: Alle vorhandenen Knoten werden in den Zustand PRE-OPERATIONAL überführt.

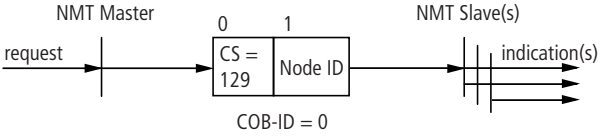


Abbildung 16: Aufbau der NMT Services, Reset Node

Node ID = 0: Bei allen vorhandenen Knoten wird ein Reset ausgeführt.

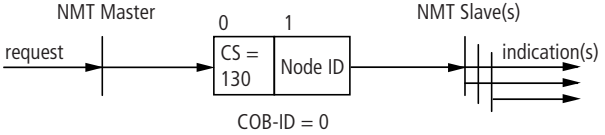


Abbildung 17: Aufbau der NMT Services, Reset Communication

Node ID = 0: Bei allen vorhandenen Knoten wird ein Reset ausgeführt.

Knotenüberwachung

Die Prüfung eines CANopen-Knotens ist insbesondere dann gefordert, wenn dieser nicht ständig Botschaften sendet (zyklische PDOs). Zur Überwachung von CANopen-Knoten sind zwei Mechanismen vorhanden, die alternativ genutzt werden können.

→ EASY221-CO unterstützt die Betriebsarten Node-Guarding und Heartbeat-Producer zur Knotenüberwachung.

Node-Guarding
Der NMT-Master pollt jeden NMT-Slave in bestimmten zeitlichen Abständen („Node-Guard-Time") mit einem knotenspezifischen Remote-Transmission-Request-Telegramm (RTR). Der NMT-Slave beantwortet diese Anfrage durch Senden seines Kommunikationsstatus. Der NMT-Master meldet an seine Applikation ein „Node-Garding-Event", falls ein Knoten seine Statusabfrage nicht innerhalb seiner spezifischen „Node-Life-Time" beantwortet hat.

Ausfall des Node-Guarding
Ist die Life-Time abgelaufen, ohne dass ein Node Guard Telegramm vom EASY221-CO empfangen wurde, wird dies als Kommunikationsfehler behandelt.

In diesem Fall werden die R-Daten für das easy-Basisgerät auf Null gesetzt. Der Ausfall des Guardings wird durch zweimaliges Blinken der Error-LED angezeigt.

Sobald das Node Guarding Protokoll wieder aufgenommen ist, erlischt die Error-LED und die Ausgänge des easy-Basisgerätes können wieder mit den PDO-Daten bedient werden.

Heartbeat-Producer

EASY221-CO signalisiert seinen Kommunikationsstatus durch zyklisches Senden einer sogenannten Heartbeat-Nachricht. Falls ein verantwortlicher Heartbeat-Consumer die Haertbeat-Nachricht nicht innerhalb der „Heartbeat Consuming Time" empfängt, wird dies seiner Anwendung als Heartbeat-Fehler angezeigt. Der zweite Parameter in Verbindung mit dem Heartbeat-Protokoll ist die im Gateway EASY221-CO einstellbare „Heartbeat-Producer-Time". Diese Zeit bestimmt den Zeitabstand zwischen zwei Heartbeat-Nachrichten des Knotens.

Wenn eine „Heartbeat Producer Time" am EASY221-CO Knoten ungleich Null konfiguriert ist, dann wird die erste Hartbeatnachricht mit dem Zustandsübergang von „Initializing" nach „Preoperational" übertragen. Die Anwendung beider Knotenüberwachungsmethoden ist nicht erlaubt. Wenn die „Haertbeat Producer Time" ungleich Null konfiguriert ist, wird das Heartbeat-Protokoll angewendet.

→ EASY221-CO unterstützt nicht die Betriebsart Heartbeat-Consumer, um Heartbeat-Nachrichten anderer CANopen-Geräte zu empfangen.

Sonstige Dienste

Speichern und Wiederherstellen von Einträgen

EASY221-CO unterstützt das Speichern und Wiederherstellen der Objektverzeichniseinträge 1000_{hex} bis $1FFF_{hex}$ im nichtflüchtigen Speicher (EEPROM oder FRAM). Der Bereich wird in den Tabellen zum Objektverzeichnis mit NVM-PA bezeichnet, im Gegensatz zu den herstellerspezifischen Einträgen, die im NVM-RO abgelegt sind.

Das Speichern von Parametern erfolgt durch das Objekt 1010_{hex} (SAVE-Signatur), es werden immer alle Parameter gespeichert.

Die Werkseinstellungen (WE) im Bereich 1000_{hex} bis $1FFF_{hex}$ können mit dem Objekt 1011_{hex} (LOAD-Signatur) wiederhergestellt werden. Es werden immer alle Werkseinstellungen wiederhergestellt.

Layer Setting Service

Der Layer Setting Service dient der Einstellung der Knotennummer über das CANopen-Netzwerk. Dieser Dienst wird vom EASY221-CO für die beiden spezifizierten Slave-Betriebsarten „Switch Mode Global" und „Switch Mode Selective" unterstützt.

→ Die Änderung der Node ID wird im EASY221-CO direkt wirksam. Damit jedoch die korrekte Node ID auch am easy- Basisgerät (unter dem Menüpunkt Konfigurator) angezeigt wird, müssen Sie das Koppelmodul erneut einschalten.

Geräteprofil

In Erweiterung zum Kommunikationsprofil nach CiA-DS-301, welches die Kommunikationsmechanismen zwischen den Teilnehmern beschreibt, definiert CANopen sogenannte Geräteprofile für die wichtigsten Geräteklassen. In den Geräteprofilen wird die Gerätefunktionalität beschrieben. EASY221-CO kann keinem bestehenden Geräteprofil zugeordnet werden.

5 Objektverzeichnis

Das Objektverzeichnis von EASY221-CO enthält die im Folgenden beschriebenen Einträge.

Kommunikationsparameter

Die Kommunikationsparameter sind in der CiA-Spezifikation [1] Kapitel 9.6.3 detailliert beschrieben.

Die Objekte 1000_{hex}, 1001_{hex} und 1018_{hex} sind für alle CANopen-Geräte erforderlich, alle anderen Objekte sind optional, hiervon unterstützt EASY221-CO die in den folgenden Tabellen aufgeführten Objekte.

Die folgende Tabelle zeigt die Objektverzeichniseinträge 1000_{hex} bis 1018_{hex}.

Index hex	Sub Index hex	Objektname	Datentyp	Zugriff Ort	WE hex	Bedeutung
1000	00	Device Type	UNSIGNED32	ro ROM	00000000	CANopen-Gerät ohne Geräteprofil
1001	00	Error Register	UNSIGNED8	ro RAM		Angabe der Fehler: 00_{hex} kein Fehler
1003	00	Predefined Error Field	UNSIGNED8	rw RAM	00	Fehlerhistorie
	01 bis 10[1)]	Standard Error Field	UNSIGNED32	ro RAM		Beschreibung des Fehlers (→[1] Seite 9-65)
1005	00	COB-ID SYNC Message	UNSIGNED32	rw NVM-PA	00000080	COB-ID vom SYNC-Objekt, Gerät konsumiert die SYNC-Nachricht

Index hex	Sub Index hex	Objekt-name	Datentyp	Zugriff Ort	WE hex	Bedeutung
1008	00	Manufac-turer Device Name	VISIBLE_STRING[2)]	ro NVM-RO	454153 593232 312D43 4F	Gerätename des Moduls (EASY221-CO)
1009	00	Manufac-turer Hardware Version	VISIBLE_STRING8	ro NVM-RO	0001.000 (Beispiel)	Hardware-Version des Moduls
100A	00	Manufac-turer Soft-ware Version	VISIBLE_STRING8	ro NVM-RO	0001.001 (Beispiel)	Software-Version des Moduls
100C	00	Guard Time	UNSIGNED16	rw NVM-PA	00 00_{hex} Auflösung in 1 ms	Guard Time in Millisekunden
100D	00	Life Time Factor	UNSIGNED8	rw NVM-PA	00_{hex}	Multiplikator mit der Guard Time, Ergebnis gleich maximale Dauer zwischen zwei Guarding-Tele-grammen
1010	00	Store Parame-ters	UNSIGNED8	ro ROM	01	maximale Anzahl der Speicheropti-onen
	01	SAVE all Parame-ters	UNSIGNED32	rw RAM	wr: 65766173 rd: 00000001	→[1] Seite 9-70

Index hex	Sub Index hex	Objekt-name	Datentyp	Zugriff Ort	WE hex	Bedeutung
1011	00	Restore default Parame-ters	UNSIGNED8	ro ROM	01	Laden der Default-Para-meter
	01	LOAD all Parame-ters	UNSIGNED32	rw RAM	wr: 64616F6C rd: 00000001	Gerät stellt die werksseitig vorgegebenen Parameter wieder her. Die Parameter bleiben bis zum nächsten Power On erhalten (→ [1] Seite 9-72)
1014	00	COB-ID EMCY Message	UNSIGNED32	ro ROM	00000080 + Node-ID	CAN-Identifier der Emergency-Nachricht
1015	00	Inhibit Time EMCY	UNSIGNED16	rw NVM-PA	0000 Auflösung in 100 µs	Zeit zwischen dem Senden von zwei EMCY-Nachrichten
1017	00	Producer Heartbeat Time	UNSIGNED16	rw NVM-PA	0000 Auflösung in 1 ms	Zeit zwischen dem Senden von zwei Heartbeat-Nachrichten

Index hex	Sub Index hex	Objekt-name	Datentyp	Zugriff Ort	WE hex	Bedeutung
1018	00	Identity Object	UNSIGNED8	ro NVM-RO	04	allgemeine Informationen über das Gerät
	01	Vendor ID	UNSIGNED32	ro NVM-RO	00000003	Hersteller
	02	Product Code	UNSIGNED32	ro NVM-RO	0323353	Produktnummer
	03	Revision Number	UNSIGNED32	ro NVM-RO	00010001 (Beispiel)	Version
	04	Serial Number	UNSIGNED32	ro NVM-RO	4010016 (Beispiel)	Seriennummer

1) EASY221-CO unterstützt bis zu 16 Fehler in der Fehlerhistorie.

2) Die maximale String-Länge ist begrenzt auf 31 Zeichen inklusive Endezeichen „\0".

EASY221-CO unterstützt die erste Server SDO des Predefined Connection Set. Die folgende Tabelle zeigt den Objektverzeichniseintrag 1200_{hex}: Server SDO Parameter der ersten Server SDO.

Index hex	Sub Index hex	Objekt-name	Datentyp	Zugriff Ort	WE hex	Bedeutung
1200	00	Server SDO Parameter	UNSIGNED8	ro ROM	02	Anzahl der gültigen Subindizes
	01	COB-ID Client → Server (rx)	UNSIGNED32	ro ROM	00000600 + Node-ID	COB-ID der Empfangs-SDO ID ist aus Predefined Connection Set abgeleitet.
	02	COB-ID Server → Client (tx)	UNSIGNED32	ro ROM	00000580 + Node-ID	COB-ID der Sende-SDO ID ist aus Pre-defined Connection Set abgeleitet.

EASY221-CO unterstützt die erste Empfang-PDO des Predefined Connection Set. Die zweite bis vierte Empfang-PDO werden nicht unterstützt. Die folgende Tabelle zeigt den Objektverzeichniseintrag 1400_{hex}: Receive PDO Communication Parameter der 1. Empfang-PDO.

Index hex	Sub Index hex	Objekt-name	Datentyp	Zugriff Ort	WE hex	Bedeutung
1400	00	Receive PDO Parameter	UNSIGNED8	ro NVM-PA	02	Kommunikationsparameter 1. Rx-PDO Anzahl der gültigen Subindizes
	01	COB-ID	UNSIGNED32	rw NVM-PA	00000200 + Node-ID	COB-ID der 1. Rx PDO, entsprechend [1]
	02	Transmission Type	UNSIGNED8	rw NVM-PA	FF	Übertragungsart der PDO: asynchron

Mit der ersten Empfang-PDO werden die Ausgangsdaten im Objektverzeichnis (Index 2011_{hex}, Subindex 00_{hex}) abgelegt und per Standardprotokoll über easyLink an das Basisgerät übertragen. Die folgende Tabelle zeigt den Objektverzeichniseintrag 1600_{hex}: Receive PDO Mapping Parameter der 1. Empfang-PDO.

Index hex	Sub Index hex	Objekt-name	Datentyp	Zugriff Ort	WE hex	Bedeutung
1600	00	Receive PDO Mapping	UNSIGNED8	ro ROM	01	Mappingparameter der 1. Rx PDO Anzahl der gültigen Subindizes
	01	1. Mapped Object	UNSIGNED32	ro ROM	2011001	Index 2011_{hex}, Subindex 00_{hex}, Länge 24 Bit

EASY221-CO unterstützt die erste Sende-PDO des Predefined Connection Set. Die zweite bis vierte Sende-PDO werden nicht unterstützt. Die folgende Tabelle zeigt den Objektverzeichniseintrag 1800_{hex}: Transmit PDO Communication Parameter der 1. Sende-PDO.

Index hex	Sub Index hex	Objekt-name	Datentyp	Zugriff Ort	WE hex	Bedeutung
1800	00	Transmit PDO Parameter	UNSIGNED8	ro NVM-PA	05	Kommunikationsparameter 1. Tx PDO Anzahl der gültigen Subindizes
	01	COB-ID	UNSIGNED32	rw NVM-PA	00000180 + Node-ID	COB-Identifier, entsprechend [1]
	02	Transmission Type	UNSIGNED8	rw NVM-PA	FF	Übertragungsart der PDO: Asynchron
	03	Inhibit Time	UNSIGNED16	rw NVM-PA	0000	Sperrzeit (min. Zeit die vergehen muss, ehe die nächste PDO gesendet wird) in ms 0000_{hex} = sofort senden
	05	Event Timer	UNSIGNED16	rw NVM-PA	0000	Ereigniszähler 0000_{hex} = nicht genutzt

Mit der ersten Sende-PDO werden die Eingangsdaten aus dem Objektverzeichnis (Index 2012_{hex}, Subindex 00_{hex}) ausgelesen und übertragen. Der Transfer erfolgt nach dem Empfang der ersten Empfang-PDO. Die folgende Tabelle zeigt den Objektverzeichniseintrag $1A00_{hex}$: Transmit PDO Mapping Parameter der 1. Sende-PDO.

Index hex	**Sub Index** hex	**Objekt-name**	**Datentyp**	**Zugriff Ort**	**WE** hex	**Bedeutung**
1A00	00	Transmit PDO Mapping	UNSIGNED8	ro ROM	01	Mappingpara-meter der 1. Tx PDO Anzahl der gültigen Subin-dizes
	01	1. Mapped Object	UNSIGNED32	ro ROM	20120018	Index 2012_{hex}, Subindex 00_{hex}, Länge 24 Bit

Herstellerspezifische Objekte

Zusätzlich zu den Objekten des Geräteprofils werden herstellerspezifische Objekte im Objektverzeichnis definiert. Für diese Objekte ist ein Bereich zwischen Index 2000_{hex} und $5FFF_{hex}$ im Objektverzeichnis von EASY221-CO vorgesehen. Hierbei kommen die in der folgenden Tabelle aufgeführten herstellerspezifische Objekte zur Anwendung.

Index hex	Sub Index hex	Objekt-name	Datentyp	Zugriff Ort	Map-bar	WE hex	Bedeutung
2001[1)]	00	Koppler-Fehler	UNSIGNED8	ro easy	nein	–	Fehlerzustand des EASY221-CO
2002[1)]	00	easy-Fehler	UNSIGNED8	ro easy	nein	–	Fehlerzustand des Bassisgerätes easy/ MFD
2011	00	Ausgangs-daten	UNSIGNED24	rw easy	ja	140000	Ausgangs-daten zum Basisgerät easy/MFD
2012	00	Eingangs-daten	UNSIGNED24	ro easy	ja	–	Eingangs-daten vom Basisgerät easy/MFD
2020	00	Status	UNSIGNED8	ro easy	nein	FF	Status 00_{hex} = Daten gültig, 01_{hex} = Daten ungültig, FF_{hex} = Initia-lisierung
2021	00	Befehl	DOMAIN Länge = 7	rw easy	nein	–	Befehl zum easy619/621
2022	00	Antwort	DOMAIN Länge = 7	ro easy	nein	–	Antwort vom easy619/621

Index hex	Sub Index hex	Objekt-name	Datentyp	Zugriff Ort	Map-bar	WE hex	Bedeutung
3020	00	Status	UNSIGNED8	ro easy	nein	FF	Status 00_{hex} = Daten gültig, 01_{hex} = Daten ungültig, FF_{hex} = Initia-lisierung
3021	00	Befehl	DOMAIN Länge = 8	rw easy/ MFD	nein	–	Befehl zum easy700/800, MFD-CP8…
3022	00	Antwort	DOMAIN Länge = 8	ro easy/ MFD	nein	–	Antwort vom easy700/800, MFD-CP8…

1) Diese beiden Einträge werden ebenfalls über die Emergency-Nachricht in den ersten beiden Byte des Manufacturer Specific Error Field übertragen (→ Abschnitt „Fehlermeldungen (Emergency)“.

Fehlermeldungen (Emergency)

EASY221-CO unterstützt den in der folgenden Tabelle beschriebenen definierten Generic Error (1000_{hex}). Er wird ausgelöst, wenn im Error Register (Index 1001_{hex}, Subindex 00_{hex}) das Bit 0 für Generic Error gesetzt ist.

Im herstellerspezifischen Fehlereintrag (Manufacturer Specific Error Field) werden im Byte 0 der Fehlercode des EASY221-CO (Index 2001_{hex}, Subindex 00_{hex}) und im Byte 1 der Fehlercode des angeschlossenen easy600 (Index 2002_{hex}, Subindex 00_{hex}) ausgegeben. Dieser Wert ist momentan fest auf 00_{hex}.

Daten-byte	Inhalt	Wert	Beschreibung
1.	Generic Error Code	1000_{hex}	Generic Error (→ [1] Kapitel 9.2.5.1)
2.			
3.	Error Register	01_{hex}	Fehlerregister (Index 1001_{hex}, Subindex 00_{hex})
4.	Manufacturer Specific Error Field (0)	xx_{hex}	Koppler-Fehler (Index 2001_{hex}, Subindex 00_{hex})
5.	Manufacturer Specific Error Field (1)	00_{hex}	easy-Fehler (Index 2002_{hex}, Subindex 00_{hex})
6.	Manufacturer Specific Error Field (2)	00_{hex}	nicht genutzt
7.	Manufacturer Specific Error Field (3)	00_{hex}	nicht genutzt
8.	Manufacturer Specific Error Field (4)	00_{hex}	nicht genutzt

Im Objektverzeichniseintrag 1003_{hex} Pre-defined Error Field werden die letzten 16 Fehler gespeichert und können per Server SDO abgefragt werden. Die Einträge des Standard Error Fields (Subindex 01_{hex} bis 10_{hex}) haben folgendes Format:

Daten-byte	Inhalt	Wert	Beschreibung
1.	Error Code	1000_{hex}	Generic Error (→ [1] Kapitel 9.2.5.1)
2.			
3.	Additional Information	xx_{hex}	Koppler-Fehler (Index 2001_{hex}, Subindex 00_{hex})
4.		00_{hex}	easy-Fehler (Index 2002_{hex}, Subindex 00_{hex})

Drittes Datenbyte: Koppelmodul-Status

Wert 00_{hex}
Das easy-Basisgerät ist mit dem Gateway EASY221-CO über easyLink verbunden (connected).

Wert 04_{hex}
Das easy-Basisgerät ist nicht eingeschaltet oder nicht mit dem Gateway EASY221-CO über easyLink verbunden.

Achtung!
Mit Unterbrechen der Kommunikation zwischen dem Basisgerät easy600 und dem Erweiterungsgerät EASY221-CO wird der entsprechende Fehler-Code im 3. Datenbyte generiert. Außerdem werden die R/S-Daten des Gateway mit dem Wert 00_{hex} gesendet.

6 CANopen-Protokolle

Für die Übertragung von Daten über den CANopen-Bus werden folgende Protokolle verwendet:

- PDO-Protokoll für die Übertragung der Ein-/Ausgangsdaten und der Betriebsart.
  Informationen zu den Dateninhalten → Kapitel 7.
- SDO-Protokoll für die Übertragung von Steuerbefehlen:
  – Datum und Uhrzeit, Sommer-/Winterzeit
  – Abbild lesen/schreiben
  – Funktionsbausteine lesen/schreiben.

Informationen zu den Dateninhalten → Kapitel 8 (easy600), 9 (easy700) und 10 (easy800/MFD).

- Emergency-Protokoll
  Informationen zu den Dateninhalten → Seite 55.

PDO-Protokoll

Beim EASY221-CO kommt standardmäßig das Write PDO Protocol, wie in der folgenden Abbildung dargestellt, zum Einsatz. Das Read PDO Protocol (nicht dargestellt) kann bei Bedarf aktiviert werden.

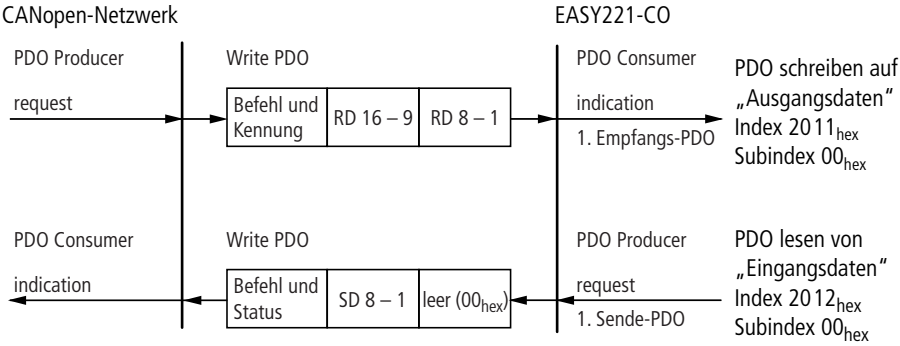


Abbildung 18: Write PDO Protocol

Die Applikation wird über Indication benachrichtigt, dass neue Daten über die 1. Empfangs-PDO empfangen und diese im Objektverzeichniseintrag „Ausgangsdaten" (Index 2011_{hex}, Subindex 00_{hex}) abgelegt werden können. Daraufhin veranlasst sie über Request das Versenden der im Objektverzeichniseintrag „Eingangsdaten" (Index 2012_{hex}, Subindex 00_{hex}) liegenden Daten über die 1. Sende PDO.

SDO-Protokoll

Allgemeine Übersicht

Über Dienstdatenobjekte („Service Data Objects", SDOs) erfolgt der bestätigte Transfer von Daten beliebiger Länge zwischen zwei Netzteilnehmern. Der Datentransfer von einem Teilnehmer zu einem anderen Teilnehmer wird im Client-Server-Modell beschrieben. Ein „SDO-Client" (initiierender Teilnehmer) hat hierbei einen direkten Zugriff auf die Einträge des Objektverzeichnisses eines „SDO-Servers" und kann Datensätze beliebiger Länge zu einem Server laden (Download) bzw. von einem Server lesen (Upload). Der zu transferierende Datensatz wird spezifiziert durch Angabe von Index und Subindex des Objektverzeichnis (OV)-Eintrags, der den Datensatz repräsentiert. Da pro Übertragungsrichtung je ein Nachrichtenidentifier benötigt wird, erfordert die Verbindung zwischen einem SDO-Client und einem SDO-Server zwei CAN-Identifier. Die Verbindung zwischen einem Client und einem Server wird auch als „SDO-Kanal" bezeichnet.

Da über einen SDO-Kanal die Übertragung von Daten beliebiger Länge möglich sein soll, die maximale Übertragungskapazität eines CAN-Telegramms jedoch nur 8 Byte beträgt, ist eine segmentierte Übertragung erforderlich. Basis hierfür sind die unter [CIA-301] spezifizierten SDO-Protokolle.

Segmentiertes Protokoll
Erfordert ein Zugriff auf das Objektverzeichnis den Transfer von mehr als 4 Byte, wird im Rahmen einer bestätigten Initialisierungssequenz zunächst der Zugriff auf den Objektverzeichniseintrag mit 16-Bit-Index und 8-Bit-Subindex spezifiziert. Anschließend erfolgt die bestätigte, segmentierte

Übertragung der Daten. Hierbei werden pro Transfer 7 Byte Daten übertragen. Das zugrunde gelegte Protokoll gewährleistet eine empfangsgesteuerte Flusskontrolle sowie die Erkennung eventuell doppelt übertragener Datensegmente. Den Datentransfer kann sowohl der Client als auch der Server abbrechen.

Die Übertragung wird durch eine „Initiate Download"-Sequenz für einen segmentierten (non-expedited) Datentransfer eingeleitet. Anschließend erfolgt die segmentweise Übertragung der Daten. Abbildung 19 zeigt das Grundprinzip des segmentierten SDO-Transfers.

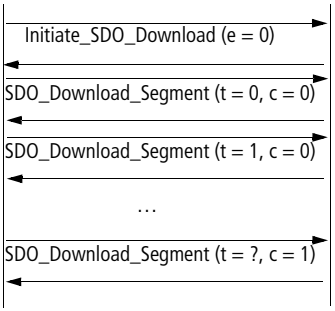


Abbildung 19: Segmentiertes SDO-Download-Protokoll

Verkürztes Protokoll

Sind dagegen maximal 4 Byte Daten zu übertragen, kann die Übertragung mit Hilfe eines verkürzten Protokolls („expedited transfer protocol") durchgeführt werden. Die zu übertragenden Daten werden hierbei bereits zusammen mit einer 1 Byte langen Protokollinformation sowie der Adresse des OV-Eintrages (Index, Subindex) innerhalb der Initialisierungssequenz übertragen (→ Abbildung 20).

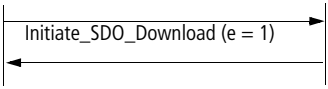


Abbildung 20: Verkürztes SDO-Download-Protokoll.

Control Byte

Das Control Byte spezifiziert die Art des Telegramms (Request/Response), Transferart (normal/expedited) und die Anzahl Bytes im Datenfeld, die keine Daten enthalten.

Abbildung 21 zeigt das Protokoll zum Schreiben eines OV-Eintrags mit Hilfe des verkürzten SDO-Protokolls. Im Client Control Byte wird angezeigt, dass es sich um einen „Initiate Download Request" handelt. Außerdem erfolgt in diesem Byte die Kennzeichnung als „expedited transfer", sowie die Angabe der Anzahl der im Datenfeld enthaltenen Datenbytes. Entsprechend zeigt das Server Control Byte an, dass es sich um eine „Initiate Download response" handelt. Anschließend an die Control Bytes wird die logische Adresse des OV-Eintrags in Form von 16-Bit-Index und 8-Bit-Subindex übertragen.

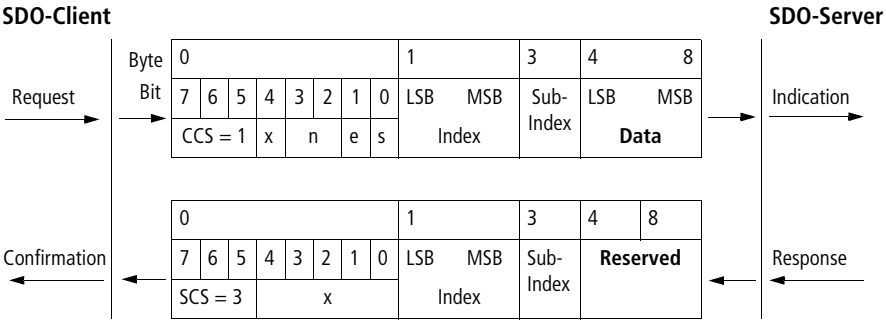


Abbildung 21: Schreiben eines OV-Eintrages mit Hilfe des „Expedited Domain Download-Protocols"

e = transfer type 0: non-expedited transfer, 1: expedited transfer

s = size indicator 0: size not indicated, 1: size indicated

n = number of bytes in data that do not contain data

x = Wert hat keine Bedeutung

CCS = 1 Client Command Specifier = initiate download request

SCS = 3 Server Command Specifier = initiate download response

Die Beschreibung des CANopen-Protokolls [CIA-301] ist im Internet, unter http://www.can-cia.org, verfügbar.

Zum besseren Verständnis der nachfolgenden Beispiele soll an dieser Stelle kurz das Download_SDO_Segment_Protokoll dargestellt werden.

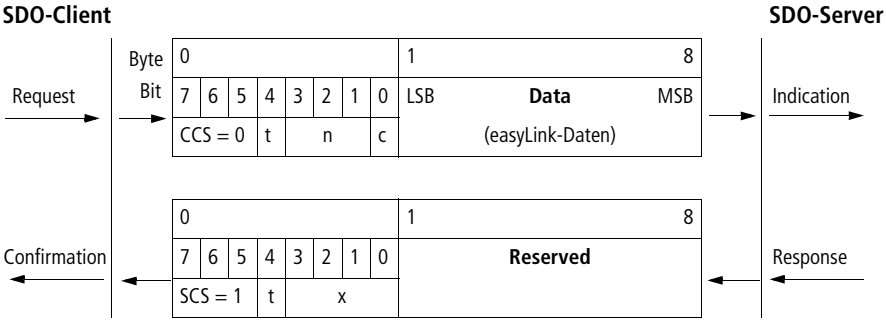


Abbildung 22: Download-Transfer-Segment nach erfolgreicher Initialisierung.

t	Toggle Bit: Das Toggle-Bit muss pro übertragenes Datenpaket gedreht werden, wobei „Request“ und „Response“ das gleiche Bit aufweisen müssen.
c	Anzeige, ob noch Segmente übertragen werden müssen: 0: weitere Segmente werden übertragen 1: keine weiteren Segmente müssen übertragen werden
n	Number of Bytes in Data that do not contain data
x	Wert hat keine Bedeutung
CCS = 0	Client Command Specifier = download segment request
SCS = 1	Server Command Specifier = download segment response

SDO-Protokoll für easy/MFD

Um auf die azyklischen Daten des Grundgerätes zugreifen zu können, müssen verschiedene CANopen-Telegrammsequenzen eingeleitet werden. Der komplette Ablauf ist in Abbildung 23 dargestellt.

1 Zuerst leitet der Client mit „Initiate SDO Download" das Schreiben per SDO auf den Objektverzeichniseintrag „Befehl" im Server ein:

Gerätefamilie	Objektverzeichniseinträge
easy600	Index 2021_{hex} Subindex 00_{hex}
easy700/800/MFD	Index 3021_{hex} Subindex 00_{hex}

2 Für den SDO-Transfer mit dem EASY221-CO werden zwei unterschiedlichen Datenlängen verwendet. Diese sind abhängig von dem verwendeten Grundgerät zu verwenden.

Gerätefamilie	Länge der easyLink-Daten
easy600	7 Byte
easy700 / 800 / MFD	8 Byte

Da die Datenlänge mehr als 4 Byte beträgt, ist danach ein „Download SDO Segment" erforderlich, um den Segmented Transfer abzuschließen. Die übertragenen Daten werden dann vom easy Protocol Handler mit dem erweiterten Protokoll zu easy/MFD übertragen.

3 Anschließend prüft der Client mit „Initiate SDO Upload", ob die Übertragung abgeschlossen ist. Dies ist erkennbar am Status im Objektverzeichnis:

Gerätefamilie	Objektverzeichniseinträge
easy600	Index 2020_{hex} Subindex 00_{hex}
easy700/800/MFD	Index 3020_{hex} Subindex 00_{hex}

Da hier nur ein Byte übertragen wird, erfolgt dies mit dem Expedited Transfer.

4 Der Client prüft zyklisch (ca. alle 50 bis 100 ms) den Status, bis der Inhalt 00_{hex} ist. Dann steht die Antwort von easy/MFD im Objektverzeichnis bereit.

Gerätefamilie	Objektverzeichniseinträge
easy600	Index 2022_{hex} Subindex 00_{hex}
easy700/800/MFD	Index 3022_{hex} Subindex 00_{hex}

5 Um die Nachricht zu lesen, leitet der Client mit „Initiate SDO Upload" den Lesevorgang ein.

6 Da auch hier die Datenlänge bis zu 7 Byte (easy600) bzw. 8 Byte (easy700/800/MFD) beträgt, müssen die restlichen Daten mit einem „Upload SDO Segment" gelesen werden.

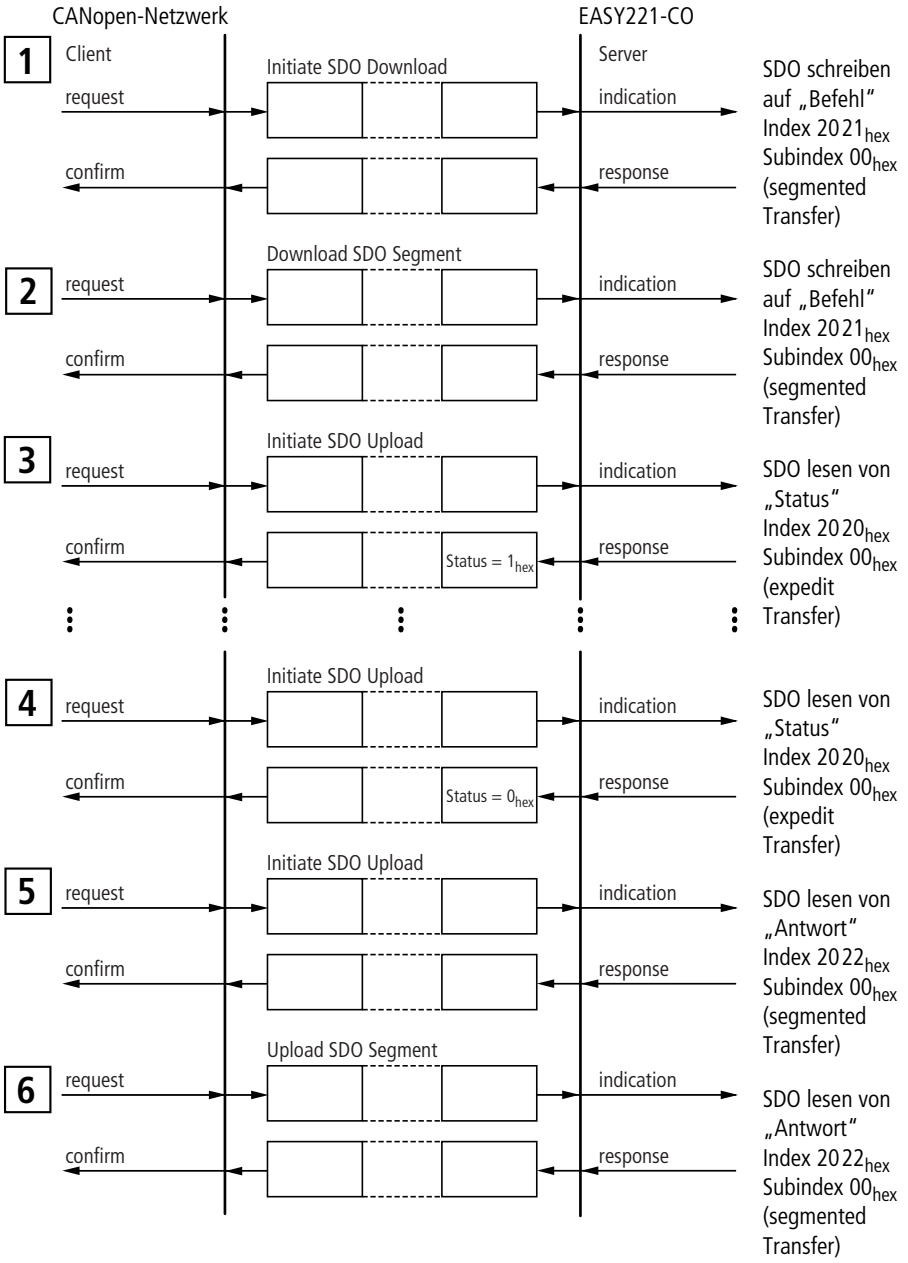


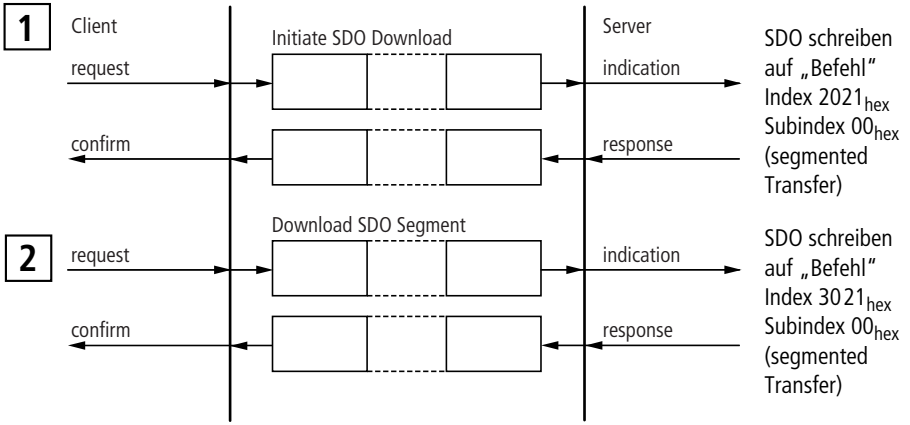
Abbildung 23: Ablauf für erweitertes SDO-Protokoll (easy600)

Beispiel für easy600: Uhrzeit Lesen (7 Byte)

Die Uhrzeit wird über den SDO-Transfer aus dem Grundgerät gelesen. Hierzu ist im Handbuch (→ Seite 87) folgender easy-Telegrammaufbau angegeben.

Byte	Bedeutung	Wert
0	Befehl: Lesen	5D
1	Wochentag	00
2	Stunde	00
3	Minute	00
4	Sommer-/Winterzeit	00

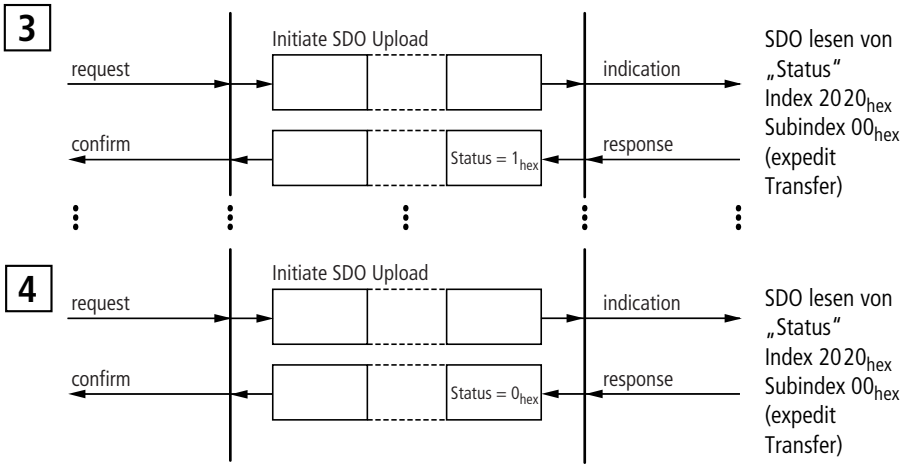
Diese Daten sind mit dem CANopen-Protokoll zu übertragen.



Beschreibung	ID (hex)	CAN-Daten – Byte (hex)							
		0	1	2	3	4	5	6	7
Befehl zum EASY221-CO schreiben									
Initialisiere den SDO-Download mit 7 Datenbyte	602	21	21	20	00	**07**[1)]	00	00	00
Bestätigung des SDO-Block-Transfer	582	60	21	20	00	00	00	00	00
Übermittlung des 1. Blocks mit 7 Datenbyte	602	01	**5D**[2)]	**00**[2)]	**00**[2)]	**00**[2)]	**00**[2)]	**00**[2)]	**00**[2)]
Bestätigung des übermittelten Datenblocks	582	20	00	00	00	00	00	00	00

1) Anzahl der zu übertragenden easy-Daten: easy600 – 7 Byte

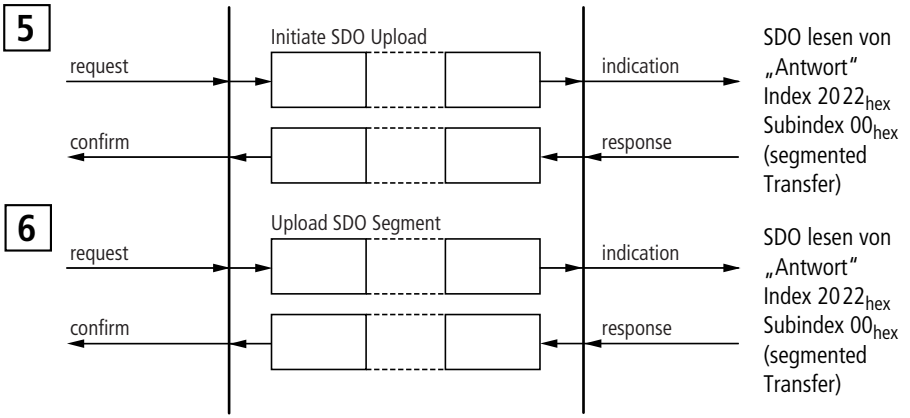
2) Gültige Daten vom easy-Grundgerät



Beschreibung	**ID(hex)**	**CAN-Daten – Byte (hex)**							
Status abfragen		0	1	2	3	4	5	6	7
Initialisiere den SDO-Upload	602	40	20	20	00	00	00	00	00
Übergabe des Statusbyte	582	4F	20	20	00	**01**[3)]	xx	xx	xx
Neuer Versuch		Data 1 wird über Index 2020_{hex} und Subindex 00_{hex} so lange abgefragt, bis der Wert = 00_{hex} ist.							
Initialisiere den SDO-Upload	602	40	20	20	00	00	00	00	00
Übergabe des Statusbyte	582	4F	20	20	00	**00**[3)]	xx	xx	xx

3) Erst mit dem Anzeigen des Wertes 00_{hex} wird sichergestellt, dass die entsprechenden Antwortdaten im Receive-Buffer bereitstehen.

xx = Wert hat keine Bedeutung



Beschreibung	ID(hex)	CAN-Daten – Byte (hex)							
Antwort abrufen		0	1	2	3	4	5	6	7
Initialisiere den SDO-Upload	602	40	22	20	00	00	00	00	00
Bestätigung des SDO-Upload-Block-Transfer mit 7 Byte	582	41	22	20	00	**07**[4]	xx	xx	xx
Abfrage des 1. Datenblocks	602	60	22	20	00	00	00	00	00
Übermittlung der easy-Antwort-Bytes	582	01	**42**[5]	**03**[5]	**17**[5]	**11**[5]	**00**[5]	**00**[5]	**00**[5]

4) Anzahl der zu übertragenden easy-Daten: easy600 – 7 Byte

5) Gültige Daten vom easy-Grundgerät

xx = Wert hat keine Bedeutung

Auswertung der erhaltenen Daten

Byte	Bedeutung	Wert
0	Antwort: Lesen erfolgreich	42
1	Wochentag	03
2	Stunde	17
3	Minute	11
4	Sommer-/Winterzeit	00

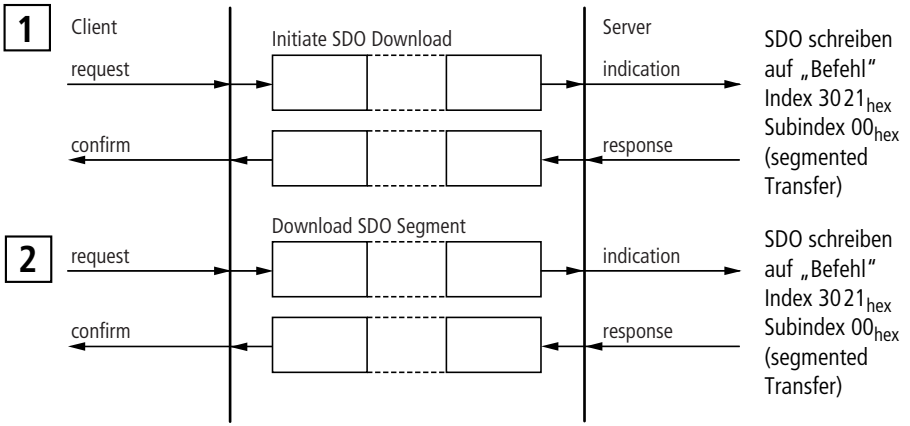
Donnerstag, 17:11 Uhr – Winterzeit

Beispiel für easy800 : Uhrzeit lesen (8 Byte)

Die Uhrzeit wird über den SDO-Transfer aus dem Grundgerät gelesen. Hierzu ist im Handbuch (→ Seite 177) folgender easy-Telegrammaufbau angegeben.

Byte	Bedeutung		Wert
0	Befehl: Lesen		93
1	Len		05
2	Index		00
3	Data 1	Stunde	00
4	Data 2	Minute	00
5	Data 3	Tag	00
6	Data 4	Monat	00
7	Data 5	Jahr	00

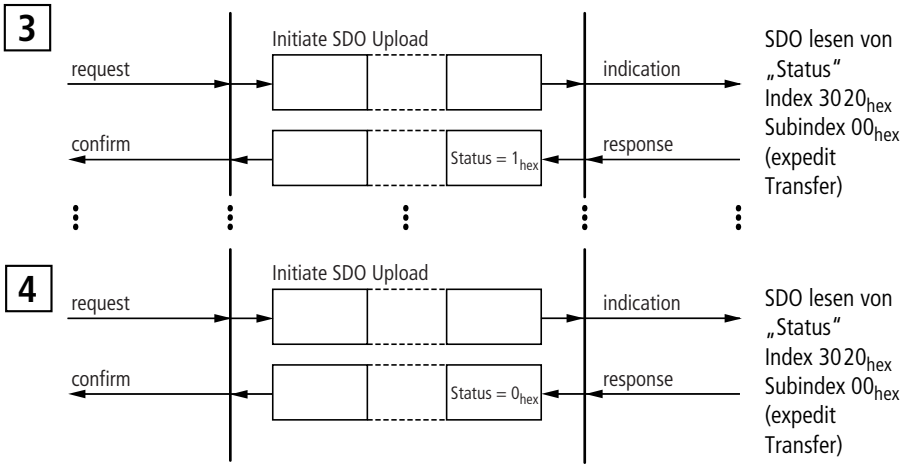
Diese Daten sind mit dem CANopen-Protokoll zu übertragen.



Beschreibung	ID(hex)	CAN-Daten – Byte (hex)							
Befehl zum EASY221-CO schreiben		0	1	2	3	4	5	6	7
Initialisiere den SDO-Download mit 7 Datenbyte	602	21	21	30	00	**08**[1)]	00	00	00
Bestätigung des SDO-Block-Transfer	582	60	21	30	00	00	00	00	00
Übermittlung des 1. Blocks mit 7DatenByte	602	00	**93**[2)]	**05**[2)]	**00**[2)]	**00**[2)]	**00**[2)]	**00**[2)]	**00**[2)]
Bestätigung des übermittelten Datenblocks	582	20	00	00	00	00	00	00	00
Übermittlung des 2. Blocks mit dem 8. Datenbyte	602	1D	**00**[2)]	xx	xx	xx	xx	xx	xx
Bestätigung des übermittelten Datenblocks	582	30	00	00	00	00	00	00	00

1) Anzahl der zu übertragenden easyDaten: easy700/800/MFD – 8 Byte

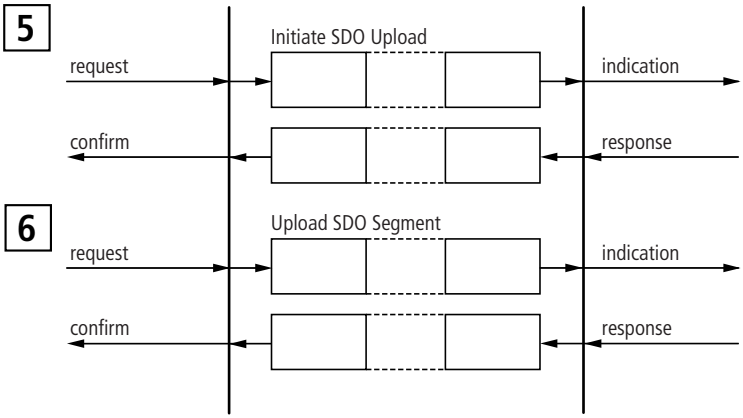
2) Gültige Daten vom easy-Grundgerät



Beschreibung	**ID(hex)**	**CAN-Daten – Byte (hex)**							
Status abfragen		0	1	2	3	4	5	6	7
Initialisiere den SDO-Upload	602	40	20	30	00	00	00	00	00
Übergabe des Statusbyte	582	4F	20	30	00	**01**[3)]	xx	xx	xx
Neuer Versuch		Data 1 wird über Index 3020_{hex} und Subindex 00_{hex} so lange abgefragt, bis der Wert = 00_{hex} ist.							
Initialisiere den SDO-Upload	602	40	20	30	00	00	00	00	00
Übergabe des Statusbyte	582	4F	20	30	00	**00**[3)]	xx	xx	xx

3) Erst mit dem Anzeigen des Wertes 00_{hex} wird sichergestellt, dass die entsprechenden Antwortdaten im Receive-Buffer bereitstehen.

xx = Wert hat keine Bedeutung



SDO lesen von „Antwort" Index 3022_{hex} Subindex 00_{hex} (segmented Transfer)

SDO lesen von „Antwort" Index 3022_{hex} Subindex 00_{hex} (segmented Transfer)

Beschreibung	**ID(hex)**	**CAN-Daten – Byte (hex)**							
Antwort abrufen		0	1	2	3	4	5	6	7
Initialisiere den SDO-Upload	602	40	22	30	00	00	00	00	00
Bestätigung des SDO-Upload-Block-Transfer mit 8 Byte	582	41	22	30	00	**08**[4)]	00	00	00
Abfrage des 1. Datenblocks	602	60	22	30	00	00	00	00	00
Übermittlung der ersten 7 easy-Antwort-Bytes	582	00	**C2**[5)]	**05**[5)]	**00**[5)]	**16**[5)]	**21**[5)]	**01**[5)]	**05**[5)]
Abfrage des 2. Datenblocks	602	70	22	30	00	00	00	00	00
Übermittlung des letzten easy-Antwort-Bytes	582	1D	**03**[5)]	xx	xx	xx	xx	xx	xx

4) Anzahl der zu übertragenden easyDaten: easy700/800/MFD – 8Byte

5) Gültige Daten vom easy-Grundgerät

xx = Wert hat keine Bedeutung

Auswertung der erhaltenen Daten

Byte	Bedeutung		Wert
0	Antwort: Lesen erfolgreich		C2
1	Len		05
2	Index		00
3	Data 1	Stunde	16
4	Data 2	Minute	21
5	Data 3	Tag	01
6	Data 4	Monat	05
7	Data 5	Jahr	03

22:31 Uhr, 01.05.2003

Emergency-Protokoll

Beim EASY221-CO kommt das Write EMCY Protocol, wie in der folgenden Abbildung dargestellt, zur Anwendung. Das Emergency Protokoll ist unbestätigt.

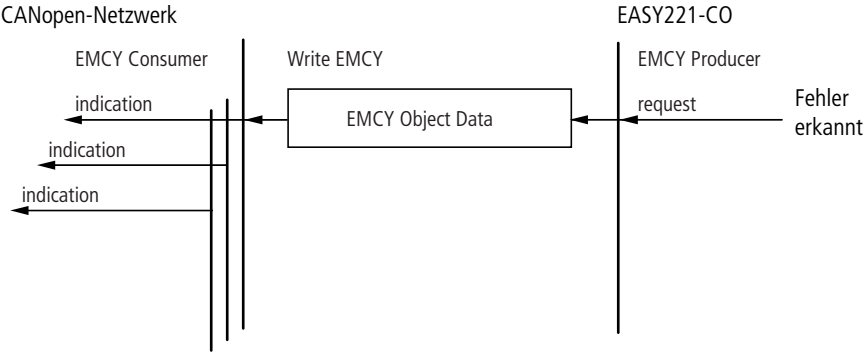


Abbildung 24: Emergency Object Protocol

7 PDO – Direkter Datenaustausch mit easy/MFD

Über den direkten zyklischen Datenaustausch (PDO) kann der CANopen-Master folgende Daten mit dem easy/MFD austauschen:

- beim Schreiben:
  - Setzen bzw. /Rücksetzen der easy/MFD Eingänge
  - Festlegen der Betriebsart RUN/STOP.
- beim Lesen:
  - Abfrage derAusgangszustände des easy/MFD
  - Abfrage der Betriebsart des easy/MFD.

Für den direkten Datenaustausch wird das PDO-Protokoll verwendet. Ausführliche Informationen hierzu finden Sie auf Seite 59. Der direkte Datenaustausch wird über die Objektverzeichnis-Einträge 2011_{hex} (Eingangsdaten) und 2012_{hex} (Ausgangsdaten) realisiert (→ Seite 54).

→ Die Begriffe „Eingangsdaten" und „Ausgangsdaten" werden aus Sicht des CANopen-Masters verwendet.

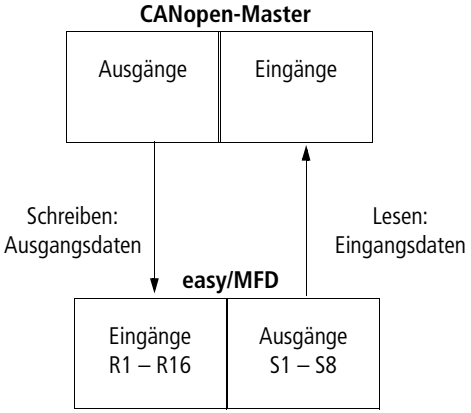


Abbildung 25: Ein- und Ausgangsdaten aus Sicht des CANopen-Masters

Ausgangsdaten (2011_{hex}): Betriebsart, R1 – R16

Die Einträge 2011_{hex} und 2012_{hex} sind map-bar und können über PDOs übertragen werden. Das Objekt 2011_{hex} enthält die Ausgangsdaten des CANopenMasters, die über das Gateway EASY221-CO in die Eingänge (R1 – R16) des easy/MFD geschrieben werden. Die Ausgangsdaten stehen in den Byte 0 bis 2 zur Verfügung und werden in den nachfolgenden Tabellen detailliert beschrieben:

Tabelle 1: Byte 0 bis 2: Ausgangsdaten, Betriebsart

Byte	Bedeutung	Wert
0	Betriebsart festlegen	→ Tabelle 2
1	Setzen/Rücksetzen der easy/MFD-Eingänge R9 bis R16	→ Tabelle 3
2	Setzen/Rücksetzen der easy/MFD-Eingänge R1 bis R8	→ Tabelle 4

Der Master schreibt in die Bytes 0, 1 und 2 folgende Daten:

Tabelle 2: Byte 0: Betriebsart

easy-Betriebsart	Bit							
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Index, um das Basisgerät in den sicheren Zustand zu setzen	0	0	0	0	0	0	0	0
Index zur Übertragung gültiger Daten	**0**	**0**	**0**	**1**	**0**	**1**	**0**	**0**
Kommando RUN	0	0	1	1	0	1	0	0
Kommando STOP	0	1	0	0	0	1	0	0

0 = Zustand „0“, 1 = Zustand „1“

Erläuterung:

Wert $14_{hex} = 00010100_{bin}$:
Dieser Wert muss immer im Byte 0 enthalten sein, wenn Daten über das Gateway EASY221-CO zum Basisgerät easy/MFD geschrieben werden sollen.

Wert $34_{hex} = 00110100_{bin}$:
Dieser Wert setzt den easy/MFD-Status von STOP auf RUN. Er wird nur als Kommando verstanden und lässt somit auch kein zusätzliches Senden von Daten zu. Hierzu muss der Index-Wert 14_{hex} verwendet werden.

Wert $44_{hex} = 01000100_{bin}$:
Dieser Wert setzt den easy/MFD-Status von RUN auf STOP. Auch er wird nur als Kommandobefehl verwendet und unterliegt somit derselben Arbeitsweise, wie auch das Kommando RUN.

Wert $00_{hex} = 00000000_{bin}$:
Wird dieser Wert in das Steuerbyte geschrieben, dann werden die R-Daten durch das Gateway mit Null überschrieben. Diese Funktionalität ist dann interessant, wenn eine Master-Steuerung in den STOP-Modus gebracht wird und als letzte resultierende Maßnahme auf alle gemappten PDOs Nullen versendet, um einen sicheren Zustand zu gewährleisten.

→ Auch wenn die Ein- und Ausgänge eines Steuerrelais direkt einem bestimmten Speicherbereich der Master-SPS zugewiesen werden können, so muss dennoch das richtige Format für die Datenzusammensetzung (z. B.: Eingangsdaten Byte 0 = 14_{hex}) eingehalten werden.

Tabelle 3: Byte 1: Setzen/Rücksetzen der easy/MFD-Eingänge R9 bis R16

easy/MFD-Eingang	Bit 7	6	5	4	3	2	1	0
R9								0/1
R10							0/1	
R11						0/1		
R12					0/1			
R13				0/1				
R14			0/1					
R15		0/1						
R16	0/1							

0 = Zustand „0“, 1 = Zustand „1“

Beispiel:
Wert $19_{hex} = 0001\,1001_{bin}$:
R13, R12 und R9 sollen aktiv werden.

Tabelle 4: Byte 2: Setzen/Rücksetzen der easy/MFD-Eingänge R1 bis R8

easy/MFD-Eingang	Bit 7	6	5	4	3	2	1	0
R1								0/1
R2							0/1	
R3						0/1		
R4					0/1			
R5				0/1				
R6			0/1					
R7		0/1						
R8	0/1							

0 = Zustand „0“, 1 = Zustand „1“

Beispiel:
Wert $2B_{hex} = 0010\ 1011_{bin}$:
R6, R4, R2 und R1 sollen aktiv werden.

→ Werden Steuerbefehle mit Ein- und Ausgangsdaten gleichzeitig benutzt gilt:

- Während der Steuerbefehl ausgeführt wird, bleiben die Eingänge solange auf dem alten Zustand, bis der Steuerbefehl abgearbeitet ist.
- Nach Beenden des Datenaustausch-Steuerbefehl werden die Bytes der Eingänge wieder aktualisiert.

Eingangsdaten (2012_{hex}): Betriebsart, S1 – S8

Die Einträge 2011_{hex} und 2012_{hex} sind map-bar und können über PDOs übertragen werden. Das Objekt 2012_{hex} enthält die Ausgangsdaten des easy/MFD (S-Daten), die über das Gateway EASY221-CO an den CANopen-Master übertragen werden. Die detaillierte Zusammensetzung der Eingangsdaten ist in den folgenden Tabellen angeführt.

Tabelle 5: Eingangsdaten, Betriebsart

Byte	Bedeutung	Wert
0	Betriebsart abfragen	→ Tabelle 6
1	Zustand der easy-Ausgänge S1 bis S8 abfragen	→ Tabelle 7
2	Nicht belegt	00_{hex}

Der Master liest in den Bytes 0, 1 und 2 folgende Daten:

Tabelle 6: Byte 0: Betriebsart

easy-Identifikation	Bit							
	7	6	5	4	3	2	1	0 STOP/RUN
ohne Eingangsverzögerung	0	0	0	1	0	0	0	0/1
mit Eingangsverzögerung	0	0	1	0	0	0	0	0/1

0 = Zustand „0“, 1 = Zustand „1“

Beispiel:
Wert $21_{hex} = 00100001_{bin}$:
easy ist im Status RUN und arbeitet mit Eingangsverzögerung

Tabelle 7: Byte 1: Zustand der easy-Ausgänge S1 bis S8

easy/MFD-Ausgang	Bit							
	7	6	5	4	3	2	1	0
S1								0/1
S2							0/1	
S3						0/1		
S4					0/1			
S5				0/1				
S6			0/1					
S7		0/1						
S8	0/1							

0 = Zustand „0“, 1 = Zustand „1“

Beispiel:
Wert $19_{hex} = 00011001_{bin}$:
S5, S4 und S1 sind aktiv

Byte 2: nicht verwendet

Werden Steuerbefehle mit Ein- und Ausgangsdaten gleichzeitig benutzt gilt:

- Während der Steuerbefehl ausgeführt wird, bleiben die Eingänge solange auf dem alten Zustand, bis der Steuerbefehl abgearbeitet ist.
- Nach Beenden des Datenaustausch-Steuerbefehls werden die Bytes der Eingänge wieder aktualisiert.

Ist der Wert des Koppelmodul-Status ungültig (= 04_{hex}), wird Byte 1 (Daten-Byte) mit dem Wert 00_{hex} auf den Kommunikationsbus übertragen.

8 SDO – Steuerbefehle für easy600

Die OV-Einträge „Status" (2020_{hex}), „Befehl" (2021_{hex}) und „Antwort" (2022_{hex}) stellen die Schnittstelle zum erweiterten Datenaustausch mit easy600 dar.

Mit Steuerbefehlen lässt sich der Datenaustausch für die speziellen Dienste realisieren:

- „Datum und Uhrzeit, Sommer-/Winterzeit lesen und schreiben" (Seite 87)
- „Abbilddaten lesen" (Seite 91)
- „Funktionsbausteine lesen/schreiben" (Seite 102).

Achtung!
Während der Benutzung eines Steuerbefehls behalten die Ein- und Ausgangsdaten ihren zuvor definierten Zustand. Erst nachdem der Datenaustausch des Steuerbefehls beendet wurde, sind die Ein- und Ausgangsdaten aktualisiert.

Warnung!
Sie dürfen nur die für den Befehlscode angegebenen Werte verwenden.
Um unnötige Fehlfunktionen zu vermeiden, überprüfen Sie bitte die zu übertragenden Daten.

Damit die Daten sicher vom Master zum Slave und umgekehrt über CANopen übertragen werden, ist das SDO-CANopen-Protokoll notwendig (→ Seite 60).

Für das Schreiben der verschiedenen Parameter muss sich das Basisgerät im Modus der LCD-Statusanzeige befinden.

Der SDO Download eines Strings auf **Befehl** löst die Übertragung der geschriebenen Daten mittels erweitertem Protokoll zum easy600 aus. Die Antwort vom easy600 kann nach erfolgtem Datenaustausch von **Antwort** per SDO Upload gelesen werden. **Status** gibt dabei den Status der Datenübertragung an. 01_{hex} bedeutet, dass eine Datenübertragung noch nicht beendet wurde. Ein erneutes Schreiben wird dann nicht ausgeführt. Steht in **Status** 00_{hex}, so ist die Datenübertragung erfolgreich abgeschlossen und die Antwort kann von **Antwort** gelesen werden. Steht in **Status** 01_{hex}, so ist bei der Datenübertragung ein Fehler aufgetreten oder es wurden ungültige Daten auf **Befehl** geschrieben. In **Antwort** stehen dann undefinierte Daten.

Datum und Uhrzeit, Sommer-/Winterzeit lesen und schreiben **Telegrammaufbau**

Byte	Bedeutung	Wert (hex), sendet		Bit							
		Master	Slave	7	6	5	4	3	2	1	0
0	Befehl										
	Lesen	5D	–	0	1	0	1	1	1	0	1
	Schreiben	2A	–	0	0	1	0	1	0	1	0
	Antwort										
	Lesen erfolgreich	–	C2/42	1/0	1	0	0	0	0	1	0
	Schreiben erfolgreich	–	C1/41	1/0	1	0	0	0	0	0	1
	Befehl zurückgewiesen	–	C0/40	1/0	1	0	0	0	0	0	0
1	Wochentag										
	Beim Lesen	00	→ Tabelle 8								
	Beim Schreiben	→ Tabelle 8	00								
2	Stunde										
	Beim Lesen	00	→ Tabelle 9								
	Beim Schreiben	→ Tabelle 9	00								
3	Minute										
	Beim Lesen	00	→ Tabelle 10								
	Beim Schreiben	→ Tabelle 10	00								
4	Sommer-/Winterzeit										
	Beim Lesen	00	→ Tabelle 11								
	Beim Schreiben	→ Tabelle 11	00								

Tabelle 8: Byte 1: Wochentag (Wertebereich 00 bis 06)

Wochentag	**Bit**							
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Montag = 00	0	0	0	0	0	0	0	0
Dienstag = 01	0	0	0	0	0	0	0	1
Mittwoch = 02	0	0	0	0	0	0	1	0
Donnerstag = 03	0	0	0	0	0	0	1	1
Freitag = 04	0	0	0	0	0	1	0	0
Samstag = 05	0	0	0	0	0	1	0	1
Sonntag = 06	0	0	0	0	0	1	1	0

Tabelle 9: Byte 2: Stunde (Wertebereich 00 bis 23)

Wert (bcd)	**Wert 10**				**Wert 1**			
	Bit				**Bit**			
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	1
…								
9	0	0	0	0	1	0	0	1
…								
14	0	0	0	1	0	1	0	0
…								
23	0	0	1	0	0	0	1	1

Tabelle 10: Byte 3: Minute (Wertebereich 00 bis 59)

Wert (bcd)	**Wert 10**				**Wert 1**			
	Bit				**Bit**			
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
00	0	0	0	0	0	0	0	0
…								
10	0	0	0	1	0	0	0	0
…								
21	0	0	1	0	0	0	0	1
…								
42	0	1	0	0	0	0	1	0
…								
59	0	1	0	1	1	0	0	1

Tabelle 11: Byte 4: Winter-/Sommerzeit (Wertebereich 00 bis 01)

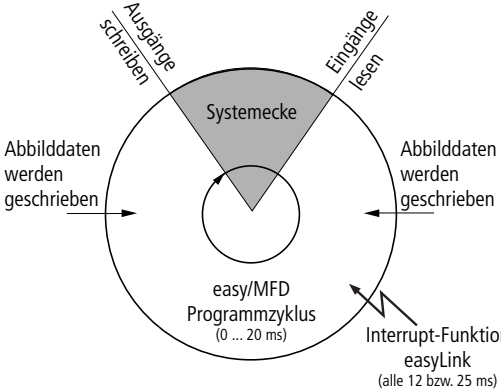
Wert (bcd)	**Wert 10**				**Wert 1**			
	Bit				**Bit**			
Funktion	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Winterzeit = 00	0	0	0	0	0	0	0	0
Sommerzeit = 01	0	0	0	0	0	0	0	1

Beispiel:
Es ist Freitag und die aktuelle Uhrzeit ist 14:36. Außerdem wird die mitteleuropäische Sommerzeit eingestellt.

Byte	Bedeutung	Wert (hex), sendet		Bit							
		Master	Slave	7	6	5	4	3	2	1	0
0	Befehl										
	Schreiben	2A	–	0	0	1	0	1	0	1	0
	Antwort										
	Schreiben erfolgreich	–	C1/41	1/0	1	0	0	0	0	0	1
1	Wochentag	04	00								
2	Stunde (14_{dez})	0E	00								
3	Minute (36_{dez})	24	00								
4	Sommer-/Winterzeit	01	00								

Abbilddaten lesen

Generelles zum Arbeiten mit Abbilddaten



Beim Schreiben auf die Abbilddaten muss beachtet werden, dass ein im easy/MFD-Programm verwendetes Abbild (z. B. Eingänge, Ausgänge,…) ebenfalls durch das eigentliche Programm zyklisch beschrieben wird. Es bleiben lediglich die Abbilddaten unverändert, die im Programm nicht verwendet und somit im Programmzyklus nicht beschrieben werden. Aus dieser Arbeitsweise geht auch hervor, dass ein über den easyLink beschriebenes Abbild, z. B. Ausgangsdaten nur dann an die physikalischen Ausgänge des easy/MFD ausgegeben werden, wenn das Steuerrelais im RUN-Modus arbeitet.

Übersicht

Operanden	Bedeutung	Lesen/ schreiben	Befehl	Seite
I1 – I16, P1 – P4, ESC/OK/DEL/ALT	„Digital-Eingänge, P-Tasten und Bedientasten"	lesen	5C	92
I7 – I8	„Analog-Eingänge: I7 – I8"	lesen	5B	95
T1 – T8, C1 – C8, ⏱1 – ⏱4, A1 – A8	„Zeitrelais, Zählerrelais, Zeitschaltuhren, Analogwertvergleicher"	lesen	5E	96
M1 – M16, Q1 – Q8, D1 – D8	„Hilfsrelais (Merker), Digital-Ausgänge, Textanzeige"	lesen	5F	99

Digital-Eingänge, P-Tasten und Bedientasten

Mit dem folgenden Befehl wird der logische Zustand der digitalen Tastereingänge P1 bis P4 sowie die logischen Zustände der Digital-Eingänge I1 bis I16 ausgelesen.

Der Zustand der P-Tasten wird nur angezeigt, wenn

- eine P-Taste im Schaltplan verwendet ist und
- die Tasten am Gerät aktiviert sind.

Telegrammaufbau

Byte	Bedeutung	Wert (hex), sendet		Bit							
		Master	Slave	7	6	5	4	3	2	1	0
0	Befehl										
	Lesen	5C	–	0	1	0	1	1	1	0	0
	Antwort										
	Lesen erfolgreich	–	C2/42	1/0	1	0	0	0	0	1	0
	Befehl zurückgewiesen	–	C0/40	1/0	1	0	0	0	0	0	0
1	Zustand der Eingänge I1 bis I8	00	→ Tabelle 12								
2	Zustand der Eingänge I9 bis I16	00	→ Tabelle 13								
3	Zustand der Tasten	00	→ Tabelle 14								

Tabelle 12: Byte 1: Status Eingänge I1 bis I8

Wert	**Bit**							
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
I1								0/1
I2							0/1	
I3						0/1		
I4					0/1			
I5				0/1				
I6			0/1					
I7		0/1						
I8	0/1							

Wert 0 = ausgeschaltet, Wert 1 = eingeschaltet

Tabelle 13: Byte 2: Status Eingänge I9 bis I16

Wert	**Bit**							
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
I9								0/1
I10							0/1	
I11						0/1		
I12					0/1			
I13				0/1				
I14			0/1					
I15		0/1						
I16	0/1							

Wert 0 = ausgeschaltet, Wert 1 = eingeschaltet

Tabelle 14: Byte 3: Zustand der Tasten

Bedeutung	Bit							
	7	6	5	4	3	2	1	0
Status P1								0/1
Status P2							0/1	
Status P3						0/1		
Status P4					0/1			
ESC unbetätigt/betätigt				0/1				
OK unbetätigt/betätigt			0/1					
DEL unbetätigt/betätigt		0/1						
ALT unbetätigt/betätigt	0/1							

Beispiel:
Wert $01_{hex} = 00000001_{bin}$:
P1 aktiv – oder Coursertaste > wird gedrückt.

Analog-Eingänge: I7 – I8

Mit dem folgenden Befehl werden die Werte der beiden Analog-Eingänge I7, I8 (nur EASY…-DC-…) gelesen.

Byte	Bedeutung	Wert (hex), sendet		Bit							
		Master	Slave	7	6	5	4	3	2	1	0
0	Befehl										
	Lesen	5B	–	0	1	0	1	1	0	1	1
	Antwort										
	Lesen erfolgreich	–	C2/42	1/0	1	0	0	0	0	1	0
	Befehl zurückgewiesen	–	C0/40	1/0	1	0	0	0	0	0	0
1	Analogwert von I7	00	siehe unten								
2	Analogwert von I8	00									

Analog-Eingang I7 und I8 (Byte 1 und Byte 2)
Diese beiden Bytes enthalten den aktuellen Wert der Analog-Eingänge I7 und I8. Ihr Wert liegt zwischen 00 und 99. Dies entspricht einem Spannungspegel von 0 bis 9,9 V an den Eingängen. Die entsprechenden Werte werden hexadezimal zurückgegeben.

Beispiel:

Byte	Wert	Beschreibung
0	42_{hex}	Der auszuführende Lesebefehl wurde erfolgreich ausgeführt. Es folgen Daten.
1	20_{hex}	Am Eingang I7 liegt eine Spannung von 3,2 V an.
2	31_{hex}	Am Eingang I8 liegt eine Spannung von 4,9 V an.

Zeitrelais, Zählerrelais, Zeitschaltuhren, Analogwertvergleicher

Mit dem folgenden Befehl wird der logische Status aller Zeitrelais, Zähler, Zeitschaltuhren und Analogwertvergleichern ausgelesen.

Telegrammaufbau

Byte	Bedeutung	Wert (hex), sendet		Bit							
		Master	Slave	7	6	5	4	3	2	1	0
0	Befehl										
	Lesen	5E	–	0	1	0	1	1	1	1	0
	Antwort										
	Lesen erfolgreich	–	C2/42	1/0	1	0	0	0	0	1	0
	Befehl zurückgewiesen	–	C0/40	1/0	1	0	0	0	0	0	0
1	Zustand Zeitrelais	00	→ Tabelle 15								
2	Zustand Zählerrelais	00	→ Tabelle 16								
3	Zustand Zeitschaltuhr	00	→ Tabelle 17								
4	Zustand Analogwert-vergleicher	00	→ Tabelle 18								

Tabelle 15: Byte 1: Zustand der Zeitrelais

	Bit							
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
T1								0/1
T2							0/1	
T3						0/1		
T4					0/1			
T5				0/1				
T6			0/1					
T7		0/1						
T8	0/1							

Beispiel:
Wert $2B_{hex} = 00101011_{bin}$:
T6, T4, T2 und T1 sind aktiv.

Tabelle 16: Byte 2: Zustand der Zählerrelais

	Bit							
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
C1								0/1
C2							0/1	
C3						0/1		
C4					0/1			
C5				0/1				
C6			0/1					
C7		0/1						
C8	0/1							

Beispiel:
Wert $19_{hex} = 00011001_{bin}$:
C5, C4 und C1 sind aktiv.

Tabelle 17: Byte 3: Zustand der Zeitschaltuhren

	Bit							
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
⏲1								0/1
⏲2							0/1	
⏲3						0/1		
⏲4					0/1			
				0				
			0					
		0						
	0							

Beispiel:
Wert $08_{hex} = 0000\,1000_{bin}$:
W3 ist aktiv.

Tabelle 18: Byte 4: Zustand der Analogwertvergleicher

	Bit							
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
A1								0/1
A2							0/1	
A3						0/1		
A4					0/1			
A5				0/1				
A6			0/1					
A7		0/1						
A8	0/1							

Beispiel:
Wert $84_{hex} = 1000\,1000_{bin}$:
A3 und A8 sind aktiv.

Hilfsrelais (Merker), Digital-Ausgänge, Textanzeige

Mit dem folgenden Befehl wird der logische Status aller Hilfsrelais (Merker) M1 bis M16, Digital-Ausgänge Q1 bis Q8, Textmerker D1 bis D8 ausgelesen.

Telegrammaufbau

Byte	Bedeutung	Wert (hex), sendet		Bit							
		Master	Slave	7	6	5	4	3	2	1	0
0	Befehl										
	Lesen	5F	–	0	1	0	1	1	1	1	1
	Antwort										
	Lesen erfolgreich	–	C2/42	1/0	1	0	0	0	0	1	0
	Befehl zurückgewiesen	–	C0/40	1/0	1	0	0	0	0	0	0
1	Zustand der Hilfsrelais (Merker) M1 bis M8	00	→ Tabelle 19								
2	Zustand der Hilfsrelais (Merker) M9 bis M16	00	→ Tabelle 20								
3	Zustand der Digital-Ausgänge Q1 bis Q8	00	→ Tabelle 21								
4	Zustand der Textmerker D1 bis D8	00	→ Tabelle 22								

Tabelle 19: Byte 1: Zustand der Hilfsrelais (Merker) 1 bis 8

	Bit							
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
M1								0/1
M2							0/1	
M3						0/1		
M4					0/1			
M5				0/1				
M6			0/1					
M7		0/1						
M8	0/1							

Beispiel:
Wert $2B_{hex} = 00101011_{bin}$:
M6, M4, M2 und M1 sind aktiv.

Tabelle 20: Byte 2: Zustand der Hilfsrelais (Merker) 9 bis 16

	Bit							
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
M9								0/1
M10							0/1	
M11						0/1		
M12					0/1			
M13				0/1				
M14			0/1					
M15		0/1						
M16	0/1							

Beispiel:
Wert $19_{hex} = 00011001_{bin}$:
M13, M12 und M9 sind aktiv.

Tabelle 21: Byte 3: Zustand der Digital-Ausgänge Q1 bis Q8

	Bit							
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Q1								0/1
Q2							0/1	
Q3						0/1		
Q4					0/1			
Q5				0/1				
Q6			0/1					
Q7		0/1						
Q8	0/1							

Beispiel:
Wert $A8_{hex} = 10101000_{bin}$:
Q8, Q6 und Q4 sind aktiv.

Tabelle 22: Byte 4: Zustand der Textmerker D1 bis D8

	Bit							
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
D1								0/1
D2							0/1	
D3						0/1		
D4					0/1			
D5				0/1				
D6			0/1					
D7		0/1						
D8	0/1							

Beispiel:
Wert $84_{hex} = 10000100_{bin}$:
D3 und D8 sind aktiv.

Funktionsbausteine lesen/schreiben

Übersicht

Das erste Datenbyte des auf **Befehl** zu schreibenden Strings stellt einen Befehl für easy600 dar und definiert die Bedeutung der restlichen 6 Datenbyte. Die folgenden Tabelle stellt die möglichen Befehle dar.

Operanden	Bedeutung	Befehl	Seite
A1 – A8	„Analogwertvergleicher A1 – A8: Istwerte schreiben (Funktion, Vergleichswerte)"	22_{hex} – 29_{hex}	103
C1 – C8	„Zählerrelais C1 – C8: Istwert lesen"	$3B_{hex}$ – 42_{hex}	106
	„Zählerrelais C1 – C8: Sollwert schreiben"	09_{hex} – 10_{hex}	108
	„Zählerrelais C1 – C8: Sollwert lesen"	43_{hex} – $4A_{hex}$	110
T1 – T8	„Zeitrelais T1 – T8: Istwert lesen (Zeitbereich, Istwert, Schaltfunktion)"	$2B_{hex}$ – 32_{ex}	112
	„Zeitrelais T1 – T8: Parameter schreiben (Zeitbereich, Sollwert, Schaltfunktion)"	01_{hex} – 08_{hex}	116
1 – 4	„Zeitschaltuhr 1 – 4: Istwerte lesen (Kanal, ON-Zeit, OFF-Zeit)"	$4B_{hex}$ – $5A_{hex}$	120
	„Zeitschaltuhr 1 – 4: Sollwerte schreiben (Kanal, ON-Zeit, OFF-Zeit)"	12_{hex} – 21_{hex}	124

Analogwertvergleicher A1 – A8: Istwerte schreiben (Funktion, Vergleichswerte)

Byte	Bedeutung	Wert (hex), sendet		Bit							
		Master	Slave	7	6	5	4	3	2	1	0
0	Befehl										
	A1	22	–	0	0	1	0	0	0	1	0
	A2	23	–	0	0	1	0	0	0	1	1
	A3	24	–	0	0	1	0	0	1	0	0
	A4	25	–	0	0	1	0	0	1	0	1
	A5	26	–	0	0	1	0	0	1	1	0
	A6	27	–	0	0	1	0	0	1	1	1
	A7	28	–	0	0	1	0	1	0	0	0
	A8	29	–	0	0	1	0	1	0	0	1
	Antwort										
	Schreiben erfolgreich	–	C1/41	1/0	1	0	0	0	0	0	1
	Befehl zurückgewiesen	–	C0/40	1/0	1	0	0	0	0	0	0
1	Steuer-Byte	→ Tabelle 23	ungültig								
2	Vergleichswert bei Vergleich mit Konstante	→ Seite 104	00								

→ Halten Sie den Wertebereich ein: Die Vergleichswerte sowie die Funktion sind Bestandteil einer „*.eas-Datei". Werden diese Werte verändert, stimmt die Orginal „*.eas-Datei" mit der in easy6… befindlichen nicht mehr überein.

Beachten Sie diese Eigenschaft beim Hoch- und Runterladen und Vergleich von „easy"-Schaltplänen mit der easySoft. Beim Runterladen vom PC wird der aktuelle Stand der „*.eas" überschrieben. Beim Vergleich sind die Schaltpläne ungleich.

Tabelle 23: Byte 1: Steuer-Byte

Bedeutung	Bit							
	7	6	5	4	3	2	1	0
Vergleich auf „≧"								0
Vergleich auf „≦"								1
I7 zu I8						0	0	
I7 zu Konstante						0	1	
I8 zu Konstante						1	0	
Fest			0	0	0			
Erscheint nicht im Parametermenü		1						
Erscheint im Parametermenü		0						
Bearbeiten	1							

Beispiel:
$82_{hex} = 10000010_{bin}$ bedeutet, dass der ausgewählte Analogwert-Vergleicher im Schaltplan des Grundgerätes aktiv wird, sobald der am Eingang I7 anliegende Analogwert ≧ der vorgegebenen Konstante (→ Byte 2) ist.

Vergleichswert (Byte 2)
Das zweite Byte enthält den Vergleichwert in Form einer Konstanten. Er liegt zwischen 0 bis 99 und entspricht einer Vergleichsspannung von 0,0 bis 9,9 V. Auch diesen Wert müssen Sie hexadezimal angeben.

Beispiel:
Ein Vergleichswert = 20_{hex} entspricht einer Analogspannung von 3,2 V.

Beispiel
Der Analogwertvergleicher A8 besitzt folgende Eigenschaften:

- Vergleich I7 < 4,7 V

Der Master erteilt den Befehl, dass der Vergleichswert auf 4,2 V gesenkt werden soll.

Byte	Bedeutung	Wert (hex)	Bit							
			7	6	5	4	3	2	1	0
0	Befehl: A8	29	0	0	1	0	1	0	0	1
	Antwort: Schreiben erfolgreich	–	0	1	0	0	0	0	0	1
1	Steuer-Byte	→	1	0	0	0	0	0	1	1
2	Vergleichswert bei Vergleich mit Konstante	2A	0	0	1	0	1	0	1	0

Der Slave antwortet mit folgendem Telegramm:

Byte	Bedeutung	Wert (hex)	Bit							
			7	6	5	4	3	2	1	0
0	Antwort: Schreiben erfolgreich	41	0	1	0	0	0	0	0	1
1	Vergleicher	ungültig								
2	Vergleichswert bei Vergleich mit Konstante	00								

Zählerrelais C1 – C8: Istwert lesen

Telegrammaufbau

Byte	Bedeutung	Wert (hex), sendet		Bit							
		Master	Slave	7	6	5	4	3	2	1	0
0	Befehl: Lesen										
	C1	3B	–	0	0	1	1	1	0	1	1
	C2	3C	–	0	0	1	1	1	1	0	0
	C3	3D	–	0	0	1	1	1	1	0	1
	C4	3E	–	0	0	1	1	1	1	1	0
	C5	3F	–	0	0	1	1	1	1	1	1
	C6	40	–	0	1	0	0	0	0	0	0
	C7	41	–	0	1	0	0	0	0	0	1
	C8	42	–	0	1	0	0	0	0	1	0
	Antwort										
	Lesen erfolgreich	–	C2/42	1/0	1	0	0	0	0	1	0
	Befehl zurückgewiesen	–	C0/40	1/0	1	0	0	0	0	0	0
1	Steuer-Byte	00	→ Tabelle 24	x	x	x	x	x	x	x	x
2	Zählerrelais-Istwert (Low Byte)	00	→ Seite 107								
3	Zählerrelais-Istwert (High Byte)	00									

Tabelle 24: Byte 1: Steuer-Byte

Bedeutung	**Bit**							
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Nicht belegt			0	0	0	0	0	0
Erscheint nicht im Parametermenü		1						
Erscheint im Parametermenü		0						
Bearbeiten (wird im Schaltplan abgearbeitet)	1							

Beispiel:
Wert $80_{hex} = 10000000_{bin}$:
Der Istwert des Zählerrelais ist geschrieben und erscheint im Parametermenü.

Istwert (Byte 2 und Byte 3)
Diese geben zusammen den Istwert des Zählerrelais an. Der Istwert kann in einem Wertebereich zwischen 0 und 9999_{dez} liegen. Um den entsprechenden Istwert zu ermitteln, müssen Sie den 16-Bitwert aus Low- und Highwert vom Hexadezimal- in das Dezimalsystem umwandeln.

Beispiel:
High-Wert: 10_{hex}
Low-Wert: DE_{hex}
$10DE_{hex} = 4318_{dez}$

Zählerrelais C1 – C8: Sollwert schreiben

Telegrammaufbau

Byte	Bedeutung	Wert (hex), sendet		Bit							
		Master	Slave	7	6	5	4	3	2	1	0
0	Befehl										
	C1	09	–	1/0	0	0	0	1	0	0	1
	C2	0A	–	1/0	0	0	0	1	0	1	0
	C3	0B	–	1/0	0	0	0	1	0	1	1
	C4	0C	–	1/0	0	0	0	1	1	0	0
	C5	0D	–	1/0	0	0	0	1	1	0	1
	C6	0E	–	1/0	0	0	0	1	1	1	0
	C7	0F	–	1/0	0	0	0	1	1	1	1
	C8	10	–	1/0	0	0	1	0	0	0	0
	Antwort										
	Schreiben erfolgreich	–	C1/41	1/0	1	0	0	0	0	0	1
	Befehl zurückgewiesen	–	C0/40	1/0	1	0	0	0	0	0	0
1	Steuer-Byte	→ Tabelle 25	00								
2	Sollwert (Low Byte)	→ Seite 109	00								
3	Sollwert (High Byte)		00								

Wertebereich der Zählerwerte: 0000 bis 9999

→ Halten Sie den Wertebereich ein.

Der Wert ist Bestandteil einer easySoft-Datei (*.eas). Werden diese Werte verändert, stimmt die Orginal „*.eas-Datei“ mit der in easy6… befindlichen nicht mehr überein.

Beachten Sie diese Eigenschaft beim Hoch- und Runterladen und Vergleich von „easy“-Schaltplänen mit der easySoft.

Beim Runterladen vom PC wird der aktuelle Stand der „*.eas“ überschrieben.

Beim Vergleich sind die Schaltpläne ungleich.

Tabelle 25: Byte 1: Steuer-Byte

Bedeutung	**Bit**							
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Nicht belegt			0	0	0	0	0	0
Erscheint nicht im Parametermenü		1						
Erscheint im Parametermenü		0						
Bearbeiten	1							

Beispiel:
Wert $80_{hex} = 1000000_{bin}$:
Sollwert wird in das gewählte Zählerrelais geschrieben und erscheint im Parametermenü.

Sollwert schreiben (Byte 2 und Byte3)
Diese geben zusammen den Sollwert des Zählerrelais an. Dieser Sollwert kann im Wertebereich zwischen 0 und 9999_{dez} ausgewählt werden. Dazu müssen Sie den gewünschten Dezimalwert in seinen äquivalenten Hexadezimalwert umwandeln und anschließend auf das Low- und das High-Byte aufsplitten.

Beispiel:
Sollwert = $4318_{dez} = 10DE_{hex}$:
Low-Wert: DE_{hex}
High-Wert: 10_{hex}

Zählerrelais C1 – C8: Sollwert lesen

Telegrammaufbau

Byte	Bedeutung	Wert (hex), sendet		Bit							
		Master	Slave	7	6	5	4	3	2	1	0
0	Befehl										
	C1	43	–	0	1	0	0	0	0	1	1
	C2	44	–	0	1	0	0	0	0	1	0
	C3	45	–	0	1	0	0	0	1	0	1
	C4	46	–	0	1	0	0	0	1	1	0
	C5	47	–	0	1	0	0	0	1	1	1
	C6	48	–	0	1	0	0	1	0	0	0
	C7	49	–	0	1	0	0	1	0	0	1
	C8	4A	–	0	1	0	0	1	0	1	0
	Antwort										
	Lesen erfolgreich	–	C2/42	1/0	1	0	0	0	0	1	0
	Befehl zurückgewiesen	–	C0/40	1/0	1	0	0	0	0	0	0
1	Steuer-Byte	00	→ Tabelle 26								
2	Zählerrelais-Sollwert (Low Byte)	00	→ Seite 111								
3	Zählerrelais-Sollwert (High Byte)	00									

Tabelle 26: Byte 1: Steuer-Byte

Bedeutung	**Bit**							
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Nicht belegt			0	0	0	0	0	0
Erscheint nicht im Parametermenü		1						
Erscheint im Parametermenü		0						
Bearbeiten (wird im Schaltplan abgearbeitet)	1							

Beispiel:
Wert $80_{hex} = 10000000_{bin}$:
Der Sollwert des Zählerrelais ist geschrieben und erscheint im Parametermenü.

Sollwert (Byte 2 und Byte 3)
Diese geben zusammen den Sollwert des Zählerrelais an. Der Sollwert kann in einem Wertebereich zwischen 0 und 9999_{dez} liegen. Um den entsprechenden Sollwert zu ermitteln müssen Sie den 16-Bitwert aus Low- und Highwert vom Hexadezimal- in das Dezimalsystem umwandeln.

Beispiel:
High-Wert: 10_{hex}
Low-Wert: DE_{hex}
$10DE_{hex} = 4318_{dez}$

Zeitrelais T1 – T8: Istwert lesen (Zeitbereich, Istwert, Schaltfunktion)

Telegrammaufbau

Byte	Bedeutung	Wert (hex), sendet		Bit							
		Master	Slave	7	6	5	4	3	2	1	0
0	Befehl: Lesen										
	T1	2B	–	0	0	1	0	1	0	1	1
	T2	2C	–	0	0	1	0	1	1	0	0
	T3	2D	–	0	0	1	0	1	1	0	1
	T4	2E	–	0	0	1	0	1	1	1	0
	T5	2F	–	0	0	1	0	1	1	1	1
	T6	30	–	0	0	1	1	0	0	0	0
	T7	31	–	0	0	1	1	0	0	0	1
	T8	32	–	0	0	1	1	0	0	1	0
	Antwort										
	Lesen erfolgreich	–	C2/42	1/0	1	0	0	0	0	1	0
	Befehl zurückgewiesen	–	C0/40	1/0	1	0	0	0	0	0	0
1	Steuer-Byte	00	→ Tabelle 27								
2	Zeit-Istwert (Low Byte)	00	→ Seite 114								
3	Zeit-Istwert (High Byte)	00									
4	zufälliger Wert	00	→ Seite 114								
5 – 6		00	00								

Tabelle 27: Byte 1: Steuer-Byte

Bedeutung	**Bit**							
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Ansprechverzögert						0	0	0
Rückfallverzögert						0	0	1
Ansprechverzögert mit Zufallsschalten						0	1	0
Rückfallverzögert mit Zufallsschalten						0	1	1
Impulsformend						1	0	0
Blinkend						1	0	1
Zeitbasis „s“				0	0			
Zeitbasis „M:S“				0	1			
Zeitbasis „H:M“				1	0			
Nicht belegt			0					
Erscheint im Parametermenü		0						
Erscheint nicht im Parametermenü		1						
Zeitrelais nicht in Bearbeitung vom Betriebssystem	0							
Zeitrelais in Bearbeitung vom Betriebssystem	1							

Istwert (Byte 2 und Byte 3)
Diese geben zusammen den Istwert des Zeitrelais an. Dieser Istwert ist von der ausgewählten Zeitbasis abhängig. Wenn das Steuer-Byte auf „Sekunde" eingestellt ist, stellt der Low-Wert die SEKUNDEN und der High-Wert die MINUTEN dar. Die maximalen Rückgabewerte für jedes Byte sind dabei 0 bis 59_{dez} ($3B_{hex}$). Es ergibt sich folgende Tabelle:

Tabelle 28: Byte 2 bis 3: Zeit-Istwert

Zeitbasis	Low-Wert	High-Wert
Millisekunde	0 bis 59 (10 ms)	0 bis 59 s
Sekunde	0 bis 59 s	0 bis 59 min
Minute	0 bis 59 min	0 bis 59 h

Beispiel:
Low-Wert 11_{hex}: Entspricht 17 s bei einer gewählten Zeitbasis „Sekunde"
High-Wert $2D_{hex}$: Entspricht 45 min bei einer gewählten Zeitbasis „Sekunde"

Zufälliger Wert (Byte 4)
Beim Zeitrelais mit Zufallsschalten, wählt easy eine zufällige Verzögerungszeit zwischen Null und der eingestellten Sollzeit aus. Diese Sollzeit wird hexadezimal in diesem Byte angegeben.

Beispiel

Der Master erteilt den Befehl zum Lesen des Zeitrelais T1:

Byte	Bedeutung	Wert (hex)	Bit							
			7	6	5	4	3	2	1	0
0	Befehl: T1	2B	0	0	1	0	1	0	1	1
1 – 3		00								

Der Slave antwortet mit folgenden Werten:

Byte	Bedeutung	Wert (hex)	Bit							
			7	6	5	4	3	2	1	0
0	Antwort: Lesen erfolgreich	C2	1	1	0	0	0	0	1	0
1	Triggerspule angesteuert, Zeitbasis „M:S", ansprechverzögert, Parameteranzeige +	→	1	0	0	0	1	0	0	0
2	Zeit-Istwert (Low Byte)	10	0	0	0	1	0	0	0	0
3	Zeit-Istwert (High Byte)	0E	0	0	0	0	1	1	1	0

Wert Sollzeit = $0E10_{hex}$ = 3600

3600 s = 60:00 M:S

Zeitrelais T1 – T8: Parameter schreiben (Zeitbereich, Sollwert, Schaltfunktion)

Byte	Bedeutung	Wert (hex), sendet		Bit							
		Master	Slave	7	6	5	4	3	2	1	0
0	Befehl: Schreiben										
	T1	01	–	0	0	0	0	0	0	0	1
	T2	02	–	0	0	0	0	0	0	1	0
	T3	03	–	0	0	0	0	0	0	1	1
	T4	04	–	0	0	0	0	0	1	0	0
	T5	05	–	0	0	0	0	0	1	0	1
	T6	06	–	0	0	0	0	0	1	1	0
	T7	07	–	0	0	0	0	0	1	1	1
	T8	08	–	0	0	0	0	1	0	0	0
	Antwort										
	Schreiben erfolgreich	–	C1/41	1/0	1	0	0	0	0	0	1
	Befehl zurückgewiesen	–	C0/40	1/0	1	0	0	0	0	0	0
1	Steuer-Byte	→ Tabelle 29	ungültig								
2	Sollwert Low-Wert	→ Seite 119	00								
3	Sollwert High-Wert										
4 – 6		00	00								

→ Zeitwerte über 60 s werden in Minuten umgewandelt.
Zeitwerte über 60 Min. werden in Stunden umgewandelt.
Zeitwerte über 24 h werden in Tage umgewandelt.

Der Wertebereich der Zeitwerte und der Sollwert des Zeitrelais sind Bestandteil einer „*.eas-Datei". Werden diese Werte verändert, stimmt die Orginal „*.eas-Datei" mit der in easy6… befindlichen nicht mehr überein.

Beachten Sie diese Eigenschaft beim Hoch- und beim Runterladen und beim Vergleich von „easy"-Schaltplänen mit der easySoft.

Beim Runterladen vom PC wird der aktuelle Stand der „*.eas" überschrieben.

Beim Vergleich sind die Schaltpläne ungleich.

Wertebereich der Zeitwerte

- „S" 00.00 bis 99.99
- „M:S" 00:00 bis 99:59 (M = 00 bis 99, S = 00 bis 59)
- „H:M" 00:00 bis 99:59 (H = 00 bis 99, M = 00 bis 59)

→ Es dürfen nur die Bytes benutzt werden, die für die Zeitbasis gelten.

Tabelle 29: Byte 1: Steuer-Byte

Bedeutung	Bit							
	7	6	5	4	3	2	1	0
Ansprechverzögert						0	0	0
Rückfallverzögert						0	0	1
Ansprechverzögert mit Zufallsschalten						0	1	0
Rückfallverzögert mit Zufallsschalten						0	1	1
Impulsformend						1	0	0
Blinkend						1	0	1
Zeitbasis „S“				0	0			
Zeitbasis „M:S“				0	1			
Zeitbasis „H:M“				1	0			
Nicht belegt			0					
Erscheint nicht im Parametermenü		1						
Erscheint im Parametermenü		0						
Bearbeiten	1							

Beispiel:
Wert $89_{hex} = 10001001_{bin}$
Zeitrelais arbeitet rückfallverzögert mit der Zeitbasis „Sekunde“.

Zeitrelais Sollwert schreiben (Byte 2 und Byte 3)
Byte 2 und Byte 3 geben zusammen den Sollwert des Zeitrelais an. Der Sollwert ist von der ausgewählten Zeitbasis abhängig. Wenn das Steuer-Byte auf Sekunde eingestellt ist, bezieht sich der Low-Wert also auf die Sekunde und der High-Wert auf die nächst höhere Zeitbasis (Minute). Der Maximalwert für jedes Byte ist dabei 0 bis 59_{dez} ($3B_{hex}$). Es ergibt sich folgende Tabelle:

Zeitbasis	Low-Wert	High-Wert
Millisekunden	0 bis 59 (10 ms)	0 bis 59 s
Sekunde	0 bis 59 s	0 bis 59 min
Minute	0 bis 59 min	0 bis 59 h

Beispiel:
Low-Wert 11_{hex}: Entspricht 17 s bei der gewählten Zeitbasis „Sekunde"
High-Wert $2D_{hex}$: Entspricht 45 min bei der gewählten Zeitbasis „Sekunde"

Zeitschaltuhr ⓓ1 – ⓓ4: Istwerte lesen (Kanal, ON-Zeit, OFF-Zeit)

Telegrammaufbau

Byte	Bedeutung	Wert (hex), sendet		Bit							
		Master	Slave	7	6	5	4	3	2	1	0
0	Befehl										
	ⓓ1 Kanal A	4B	–	0	1	0	0	1	0	1	1
	ⓓ1 Kanal B	4C	–	0	1	0	0	1	1	0	0
	ⓓ1 Kanal C	4D	–	0	1	0	0	1	1	0	1
	ⓓ1 Kanal D	4E	–	0	1	0	0	1	1	1	0
	ⓓ2 Kanal A	4F	–	0	1	0	0	1	1	1	1
	ⓓ2 Kanal B	50	–	0	1	0	1	0	0	0	0
	ⓓ2 Kanal C	51	–	0	1	0	1	0	0	0	1
	ⓓ2 Kanal D	52	–	0	1	0	1	0	0	1	0
	ⓓ3 Kanal A	53	–	0	1	0	1	0	0	1	1
	ⓓ3 Kanal B	54	–	0	1	0	1	0	1	0	0
	ⓓ3 Kanal C	55	–	0	1	0	1	0	1	0	1
	ⓓ3 Kanal D	56	–	0	1	0	1	0	1	1	0
	ⓓ4 Kanal A	57	–	0	1	0	1	0	1	1	1
	ⓓ4 Kanal B	58	–	0	1	0	1	1	0	0	0
	ⓓ4 Kanal C	59	–	0	1	0	1	1	0	0	1
	ⓓ4 Kanal D	5A	–	0	1	0	1	1	0	1	0
	Antwort										
	Lesen erfolgreich	–	C2/42	1/0	1	0	0	0	0	1	0
	Befehl zurückgewiesen	–	C0/40	1/0	1	0	0	0	0	0	0
1	Steuer-Byte „Zeitschaltuhr"	00	→ Tabelle 30								
2	Steuer-Byte „Kanal"	00	→ Tabelle 31								
3	Minute (Schaltpunkt ON)	00	→ Seite 123								
4	Stunde (Schaltpunkt ON)	00									
5	Minute (Schaltpunkt OFF)	00									
6	Stunde (Schaltpunkt OFF)	00									

Tabelle 30: Byte 1: Steuer-Byte „Zeitschaltuhr"

Bedeutung	**Bit**							
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Nicht in Bearbeitung	0	0	0	0	0	0	0	0
In Bearbeitung (wird im Schaltplan abgearbeitet)	1	0	0	0	0	0	0	0

Beispiel:
Wert $80_{hex} = 10000000_{bin}$:
Die angesprochene Zeitschaltuhr wird im Schaltplan verwendet.

Steuer-Byte „Kanal"

(Wochentag: beginnend/endend, Parametermenüanzeige)
Jeder Kanal einer Wochenschaltuhr besitzt ein Steuer-Byte in dem Start- und Stoppbedingungen definiert sind. Die genaue Zusammensetzung des Steuer-Bytes ist in der folgenden Tabelle angeführt.

Tabelle 31: Byte 2: Steuer-Byte „Kanal"

Bedeutung	**Bit**							
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Tag ON								
keiner eingestellt						0	0	0
Montag						0	0	1
Dienstag						0	1	0
Mittwoch						0	1	1
Donnerstag						1	0	0
Freitag						1	0	1
Samstag						1	1	0
Sonntag						1	1	1

Bedeutung	Bit							
	7	6	5	4	3	2	1	0
Tag OFF								
keiner eingestellt			0	0	0			
Montag			0	0	1			
Dienstag			0	1	0			
Mittwoch			0	1	1			
Donnerstag			1	0	0			
Freitag			1	0	1			
Samstag			1	1	0			
Sonntag			1	1	1			
Erscheint im Parametermenü								
Nein	1	0						
Ja	0	0						

Beispiel:
Wert $31_{hex} = 00110001_{bin}$:
Der zuvor ausgewählte Kanal X der Wochenzeitschaltuhr Y ist von Montag bis Samstag aktiv.

Schaltzeitpunkte (Byte 3 bis Byte 6)
Die folgenden Bytes beschreiben den genauen Anfangs- und Endschaltzeitpunkt eines Kanals. Die Genauigkeit beträgt hierbei eine Sekunde.

Einschaltzeitpunkt		Ausschaltzeitpunkt	
Byte 3	**Byte 4**	**Byte 5**	**Byte 6**
Minute ON	Stunde ON	Minute OFF	Stunde OFF
00 bis $3B_{hex}$ (00 bis 59_{dez})	00 bis 17_{hex} (00 bis 23_{dez})	00 bis $3B_{hex}$ (00 bis 59_{dez})	00 bis 17_{hex} (00 bis 23_{dez})

→ easy liefert hexadezimale Werte. Eventuell ist es notwendig, die entsprechenden Werte in das dezimale Zahlensystem zu transferieren.

Beispiel:

Byte	Wert	Beschreibung
0	42_{hex}	Der auszuführende Lesebefehl wurde erfolgreich ausgeführt. Es folgen Daten.
1	80_{hex}	Die angesprochene Zeitschaltuhr wird im Schaltplan verwendet.
2	31_{hex} (siehe oben)	Tag: Montag bis Samstag Kanal erscheint im Parametermenü
3	00_{hex}	ON: 19:00
4	13_{hex}	
5	$1E_{hex}$	OFF: 06:30
6	06_{hex}	

Zeitschaltuhr ⏲1 – ⏲4: Sollwerte schreiben (Kanal, ON-Zeit, OFF-Zeit)

Telegrammaufbau

Byte	Bedeutung	Wert (hex), sendet		Bit							
		Master	Slave	7	6	5	4	3	2	1	0
0	Befehl										
	⏲1 Kanal A	12	–	0	0	0	1	0	0	1	0
	⏲1 Kanal B	13	–	0	0	0	1	0	0	1	1
	⏲1 Kanal C	14	–	0	0	0	1	0	1	0	0
	⏲1 Kanal D	15	–	0	0	0	1	0	1	0	1
	⏲2 Kanal A	16	–	0	0	0	1	0	1	1	0
	⏲2 Kanal B	17	–	0	0	0	1	0	1	1	1
	⏲2 Kanal C	18	–	0	0	0	1	1	0	0	0
	⏲2 Kanal D	19	–	0	0	0	1	1	0	0	1
	⏲3 Kanal A	1A	–	0	0	0	1	1	0	1	0
	⏲3 Kanal B	1B	–	0	0	0	1	1	0	1	1
	⏲3 Kanal C	1C	–	0	0	0	1	1	1	0	0
	⏲3 Kanal D	1D	–	0	0	0	1	1	1	0	1
	⏲4 Kanal A	1E	–	0	0	0	1	1	1	1	0
	⏲4 Kanal B	1F	–	0	0	0	1	1	1	1	1
	⏲4 Kanal C	20	–	0	0	1	0	0	0	0	0
	⏲4 Kanal D	21	–	0	0	1	0	0	0	0	1
	Antwort										
	Schreiben erfolgreich	–	C1/41	1/0	1	0	0	0	0	0	1
	Befehl zurückgewiesen	–	C0/40	1/0	1	0	0	0	0	0	0
1	Steuer-Byte (Tag-Beginn/-Ende)	→ Seite 125	00								
2	Minute (Schaltpunkt ON)	→ Seite 127	00								
3	Stunde (Schaltpunkt ON)		00								
4	Minute (Schaltpunkt OFF)		00								
5	Stunde (Schaltpunkt OFF)		00								
6	nicht benutzt										

→ Halten Sie den Wertebereich ein: Die Werte von Minute und Stunde der jeweiligen Schaltpunkte sind Bestandteil einer easySoft-Datei (*.eas). Werden diese Werte verändert, stimmt die Orginal „*.eas-Datei" mit der in easy6… befindlichen nicht mehr überein.

Beachten Sie diese Eigenschaft beim Hoch- und beim Runterladen und Vergleich von „easy"-Schaltplänen mit der easySoft. Beim Runterladen vom PC wird der aktuelle Stand der „*.eas" überschrieben. Beim Vergleich sind die Schaltpläne ungleich.

Steuer-Byte (Wochentag: beginnend/endend, Parameter-Menüanzeige)
Jeder Kanal einer Wochenschaltuhr besitzt ein Steuer-Byte in dem Start- und Stoppbedingungen definiert sind. Die genaue Zusammensetzung des Steuer-Bytes ist in der folgenden Tabelle angeführt.

Tabelle 32: Byte 1: Steuerbyte

Bedeutung	**Bit**							
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Tag ON								
keiner eingestellt						0	0	0
Montag						0	0	1
Dienstag						0	1	0
Mittwoch						0	1	1
Donnerstag						1	0	0
Freitag						1	0	1
Samstag						1	1	0
Sonntag						1	1	1

Bedeutung	Bit							
	7	6	5	4	3	2	1	0
Tag OFF								
keiner eingestellt			0	0	0			
Montag			0	0	1			
Dienstag			0	1	0			
Mittwoch			0	1	1			
Donnerstag			1	0	0			
Freitag			1	0	1			
Samstag			1	1	0			
Sonntag			1	1	1			
Erscheint im Parametermenü								
Nein	1	0						
Ja	0	0						

Beispiel:
Wert $31_{hex} = 00110001_{bin}$:
Der zuvor ausgewählte Kanal X der Wochenzeitschaltuhr Y ist von Montag bis Samstag aktiv.

Ein- und Ausschaltpunkt schreiben (Byte 2 bis Byte 5)
Die folgenden Bytes beschreiben den genauen Anfangs- und Endschaltzeitpunkt eines Kanals. Die Genauigkeit beträgt hierbei eine Sekunde.

Einschaltzeitpunkt		**Ausschaltzeitpunkt**	
Byte 2	**Byte 3**	**Byte 4**	**Byte 5**
Minute ON	Stunde ON	Minute OFF	Stunde OFF
00 bis $3B_{hex}$ (00 bis 59_{dez})	00 bis 17_{hex} (00 bis 23_{dez})	00 bis $3B_{hex}$ (00 bis 59_{dez})	00 bis 17_{hex} (00 bis 23_{dez})

→ Alle Werte müssen Sie hexadezimal umwandeln und dann entsprechend eintragen.

Beispiel:

Beschreibung	**Befehl/Byte**	**Wert**
Der Kanal A der Uhr 4 besitzt folgende Daten:	Befehl/Byte 0	$1E_{hex}$
Tag: Montag bis Samstag Kanal erscheint im Parametermenü	Byte 1	31_{hex} (siehe oben)
ON: 19:00	Byte 2	00_{hex}
	Byte 3	13_{hex}
OFF: 06:30	Byte 4	$1E_{hex}$
	Byte 5	06_{hex}

Beispiel
Der Master erteilt den Befehl, in den Kanal „C" der 2 folgende Daten zu schreiben:

- Tag: Dienstag (010) bis Samstag (110)
- ON: 10:00
- OFF: 17:30
- Schaltpunkt ON < OFF (0)
- Kanal erscheint nicht im Parametermenü (1)

Byte	Bedeutung	Wert	Bit							
			7	6	5	4	3	2	1	0
0	Befehl: 2 Kanal C	18_{hex}	0	0	0	1	1	0	0	0
1	Wochentag, Parametermenü-Anzeige	$B2_{hex}$	1	0	1	1	0	0	1	0
2	Minute (Schaltpunkt ON)	00_{bcd}	0	0	0	0	0	0	0	0
3	Stunde (Schaltpunkt ON)	10_{bcd}	0	0	0	1	0	0	0	0
4	Minute (Schaltpunkt OFF)	30_{bcd}	0	0	1	1	0	0	0	0
5	Stunde (Schaltpunkt OFF)	17_{bcd}	0	0	0	1	0	1	1	1
6	nicht benutzt									

Der Slave antwortet mit folgendem Telegramm:

Byte	Bedeutung	Wert	Bit							
			7	6	5	4	3	2	1	0
0	Antwort: Schreiben erfolgreich	41_{hex}	0	1	0	0	0	0	0	1
1 – 6		00								

9 SDO – Steuerbefehle für easy700

Die Objektverzeichnis-Einträge „Status" (3020_{hex}), „Befehl" (3021_{hex}) und „Antwort" (3022_{hex}) stellen die Schnittstelle zum erweiterten Datenaustausch mit easy700 am CANopen-Kommunikationsbus dar. Damit können Sie Dienste aus folgenden Bereichen übertragen:

- „Datum und Uhrzeit lesen/schreiben" (Seite 130)
- „Abbilddaten lesen/schreiben" (Seite 134)
- „Funktionsbaustein-Daten lesen/schreiben" (Seite 155).

Damit die Daten sicher vom Master zum Slave und umgekehrt über CANopen übertragen werden, ist das SDO-CANopen-Protokoll notwendig (→ Seite 60).

Achtung!
Während der Benutzung eines Steuerbefehls behalten die Ein- und Ausgangsdaten den Zustand vor Aufruf des Steuerbefehls. Erst wenn der Datenaustausch des Steuerbefehls beendet ist, sind die Ein-/Ausgangsdaten wieder aktuell.

Warnung!
Es dürfen nur die für den Befehlscode angegebenen Werte verwendet werden. Überprüfen Sie die Werte, die Sie schreiben, um Fehlfunktionen zu vermeiden.

Datum und Uhrzeit lesen/schreiben

→ Beachten Sie bitte auch die entsprechende Beschreibung zur Echtzeituhr im easy700-Handbuch (MN05013003Z-DE; frühere Bezeichnung AWB2528-1508D).

Telegrammaufbau

Byte	Bedeutung	Wert (hex), sendet	
		Master	Slave
0	Befehl		
	Lesen	93	–
	Schreiben	B3	–
	Antwort		
	Lesen erfolgreich	–	C2
	Schreiben erfolgreich	–	C1
	Befehl zurückgewiesen	–	C0
1	Len	05	05
2	Index	0 – 2[1)]	0 – 2[1)]
3 – 7	Data 1 – 5	abhängig vom Index, → Tabelle 33	

1) 0 = Uhrzeit/Datum, → Tabelle 33
1 = Sommerzeit, → Tabelle 34
2 = Winterzeit, → Tabelle 35

Tabelle 33: Index 0 – Datum und Zeit der Echtzeituhr

Byte	Inhalt	Operand		Wert (hex)
3	Data 1	Stunde	0 bis 23	0x00 bis 0x17h
4	Data 2	Minute	0 bis 59	0x00 bis 0x3Bh
5	Data 3	Tag	1 bis 28; 29, 30, 31 (abhängig von Monat und Jahr)	0x01 bis 0x1Fh
6	Data 4	Monat	1 bis 12	0x01 bis 0x0Ch
7	Data 5	Jahr	0 bis 99 (entspricht 2000-2099)	0x00 bis 0x63h

Tabelle 34: Index 1 – Sommerzeit

Byte	Inhalt			Wert (hex)
3	Data 1	Area		
			keine	00
			Regel	01
			automatische EU	02
			automatische GB	03
			automatische US	04
für „Area“ = „Regel“:				
4	Data 2	Schaltregel Sommerzeit		→ Tabelle 36
5	Data 3			
6	Data 4			
7	Data 5			

Tabelle 35: Index 2 – Winterzeit
(nur gültig, wenn Area = „Regel" gewählt worden ist)

Byte	Inhalt		Wert (hex)
3	Data 1	Area = Regel	01
4 – 7	Data 2 – 5	Schaltregel Winterzeit	→ Tabelle 36

Bitfeld „Schaltregel"

→ Lesen Sie hierzu bitte die detaillierte Beschreibung im easy700-Handbuch (MN05013003Z-DE; frühere Bezeichnung AWB2528-1508D).

Die nachfolgende Tabelle gibt Ihnen die Vorschrift zum Zusammensetzen der entsprechenden Datenbytes an.

Tabelle 36: Bitfeld Schaltregel

	Data 5								Data 4								Data 3								Data 2											
Bit	**31**	**30**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	Differenz				**Zeit der Umstellung**											**Monat**				**Tag**					**Regel_2**		**Tag**			**Regel_1**			
	0: 0:30h				Minute: 0 bis 59						Stunde: 0 bis 23					0 bis 11				0 bis 30					0: im		0: So			0: am			
	1: 1:00h																								1: nach dem		1: Mo			1: am ersten			
	2: 1:30h																								2: vor dem		2: Di			2: am zweiten			
	3: 2:00h																										3: Mi			3: am dritten			
	4: 2:30h																										4: Do			4: am vierten			
	5: 3:00h																										5: Fr			5: am letzten			
																											6: Sa						

Abbilddaten lesen/ schreiben

→ Beachten Sie bitte auch die entsprechende Beschreibung zu den möglichen Abbilddaten, die im easy700-Handbuch (MN05013003Z-DE; frühere Bezeichnung AWB2528-1508D) oder in der easySoft-Hilfe angeführt sind. Desweiteren gilt der Abschnitt „Generelles zum Arbeiten mit Abbilddaten" auf Seite 91 auch für easy700.

Übersicht

Operanden	Bedeutung	Lesen/ schreiben	Type (hex)	Seite
A1 – A16	„Analogwertvergleicher/Schwellwertvergleicher: A1 – A16"	lesen	8B	135
C1 – C16	„Zähler: C1 – C16"	lesen	EE	136
D1 – D16	„Textbausteine: D1 – D16"	lesen	94	137
I1 – I16	„Lokale Eingänge: I1 – I16"	lesen	84	138
IA1 – IA4	„Lokale Analog-Eingänge: IA1 – IA4"	lesen	8C	139
M1 – M16, N1 – N16	„Merker: M1 – M16/N1 – N16 schreiben"	schreiben	86/87	141
M1 – M16, N1 – N16	„Merker: M1 – M16/N1 – N16 lesen"	lesen	86/87	143
O1 – O4	„Betriebsstundenzähler: O1 – O4"	lesen	EF	145
P1 – P4	„Lokale P-Tasten: P1 – P4"	lesen	8A	146
Q1 – Q8	„Lokale Ausgänge: Q1 – Q8"	lesen	85	148
R1 – R16/ S1 – S8	„Ein-/Ausgänge von easyLink: R1 – R16/ S1 – S8"	lesen	88/89	149
T1 – T16	„Zeitglieder: T1 – T16"	lesen	ED	151
Y1 – Y4	„Jahresschaltuhr: Y1 – Y8"	lesen	91	152
Z1 – Z3	„Masterreset: Z1 – Z3"	lesen	93	153
H1 – H4	„Wochenschaltuhr: 1 – 8"	lesen	90	154

Analogwertvergleicher/Schwellwertvergleicher: A1 – A16

Mit den nachfolgenden Befehlen lesen Sie den logischen Status der einzelnen Analogwertvergleicher A1 bis A16 aus.

Telegrammaufbau

Byte	Bedeutung	Wert (hex), sendet	
		Master	Slave
0	Befehl: Lesen	88	–
	Antwort:		
	Lesen erfolgreich	–	C2
	Befehl zurückgewiesen	–	C0[1)]
1	Len	01	01
2	Type	8B	8C
3	Index	00	00
4	Data 1 (Low Byte)	00	→ Tabelle 37
5	Data 2 (Low Byte)	00	→ Tabelle 37
6 – 7	Data 3 – 4	00	00

1) Mögliche Ursachen → Seite 173

Tabelle 37: Byte 4 bis 5: Data 1 bis 2

Data 1	Bit	7	6	5	4	3	2	1	0
A1									0/1
A2								0/1	
…					…				
A8		0/1							
Data 2	**Bit**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
A9									0/1
A10								0/1	
…					…				
A16		0/1							

Zähler: C1 – C16

Mit den nachfolgenden Befehlen lesen Sie den logischen Status der einzelnen Zähler C1 – C16 aus.

Telegrammaufbau

Byte	Bedeutung	Wert (hex), sendet	
		Master	Slave
0	Befehl: Lesen	88	–
	Antwort:		
	Lesen erfolgreich	–	C2
	Befehl zurückgewiesen	–	C0[1)]
1	Len	01	01
2	Type	EE	EE
3	Index	00	00
4	Data 1 (Low Byte)	00	→ Tabelle 47
5	Data 2 (Low Byte)	00	→ Tabelle 47
6 – 7	Data 3 – 4	00	00

1) Mögliche Ursachen → Seite 173

Tabelle 38: Byte 4 bis 5: Data 1 bis 2

Data 1	Bit	7	6	5	4	3	2	1	0
C1									0/1
C2								0/1	
…					…				
C8		0/1							
Data 2	**Bit**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
C9									0/1
C10								0/1	
…					…				
C16		0/1							

Textbausteine: D1 – D16

Mit den nachfolgenden Befehlen lesen Sie den logischen Status der einzelnen Textbausteine (D-Merker) aus.

Telegrammaufbau

Byte	Bedeutung	Wert (hex), sendet	
		Master	Slave
0	Befehl: Lesen	88	–
	Antwort:		
	Lesen erfolgreich	–	C2
	Befehl zurückgewiesen	–	C0[1)]
1	Len	01	01
2	Type	94	94
3	Index	00	00
4	Data 1 (Low Byte)	00	→ Tabelle 39
5	Data 2 (High Byte)	00	→ Tabelle 39
6 – 7	Data 3 – 4	00	00

1) Mögliche Ursachen → Seite 173

Tabelle 39: Byte 4 bis 5: Data 1 bis 2

Data 1	Bit	7	6	5	4	3	2	1	0
D1									0/1
D2								0/1	
…					…				
D8		0/1							
Data 2	**Bit**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
D9									0/1
D10								0/1	
…					…				
D16		0/1							

Lokale Eingänge: I1 – I16

Mit diesem Befehlsstring können Sie die lokalen Eingänge des Basisgerätes easy700 auslesen. Das entsprechende Eingangswort ist hierbei im Intel-Format abgelegt.

Telegrammaufbau

Byte	Bedeutung	Wert (hex), sendet	
		Master	Slave
0	Befehl: Lesen	88	–
	Antwort:		
	Lesen erfolgreich	–	C2
	Befehl zurückgewiesen	–	C0[1)]
1	Len	02	02
2	Type	84	84
3	Index	00	00
4	Data 1 (Low Byte)	00	→ Tabelle 40
5	Data 2 (High Byte)	00	→ Tabelle 40
6 – 7	Data 3 – 4	00	00

1) Mögliche Ursachen → Seite 173

Tabelle 40: Byte 4 bis 5: Data 1 bis 2

Data 1	Bit	7	6	5	4	3	2	1	0
I1									0/1
I2								0/1	
…					…				
I8		0/1							
Data 2	**Bit**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
I9									0/1
I10								0/1	
…					…				
I16		0/1							

Lokale Analog-Eingänge: IA1 – IA4

Die am Basisgerät easy700 vorhandenen Analog-Eingänge (I7, I8, I11, I12) können Sie direkt über den CANopen auslesen. Hierbei wird der 16-Bitwert im Intelformat übergeben (Low Byte first).

Telegrammaufbau

Byte	Bedeutung	Wert (hex), sendet	
		Master	Slave
0	Befehl: Lesen	88	–
	Antwort:		
	Lesen erfolgreich	–	C2
	Befehl zurückgewiesen	–	C0[1)]
1	Len	02	02
2	Type	8C	8C
3	Index	00 – 03[2)]	00 – 03[2)]
4	Data 1 (Low Byte)	00	→ Tabelle 41
5	Data 2 (High Byte)	00	→ Tabelle 41
6 – 7	Data 3 – 4	00	00

1) Mögliche Ursachen → Seite 173

2) 00 = Analog-Eingang I7
   01 = Analog-Eingang I8
   02 = Analog-Eingang I11
   03 = Analog-Eingang I12

Beispiel:
Es liegt ein Spannungspegel am Analog-Eingang „1" an. Die entsprechenden Telegramme zum Lesen des Analogwertes sehen wie folgt aus:

Tabelle 41: Beispiel-Telegramm zum Lesen des Wertes am Analog-Eingang „1"

Byte	Bedeutung	Wert (hex), sendet	
		Master	Slave
0	Befehl: Lesen	88	–
	Antwort: Lesen erfolgreich	–	C2
1	Len	02	02
2	Type	8C	8C
3	Index	02 1)	02 1)
4	Data 1	00	4B
5	Data 2	00	03
6	Data 3	00	00
7	Data 4	00	00

1) 02 = Analog-Eingang I11

Byte 4 – Data 1 (Low Byte): $4B_{hex}$
Byte 5 – Data 2 (High Byte): 03_{hex}
→ entsprechender 16-Bitwert: $034B_{hex} = 843$

Der Wert „843" entspricht dem 10-Bit-Wert des Analogwandlers. Für den tatsächlichen Analogwert ist folgende Umrechnung notwendig:

$$\frac{10\ V}{1023} \times \text{10-Bitwert} \quad => \quad \frac{10\ V}{1023} \times 843 = 8{,}24\ V$$

Merker: M1 – M16/N1 – N16 schreiben

Telegrammaufbau

Byte	Bedeutung	Wert (hex), sendet	
		Master	Slave
0	Befehl: Schreiben	8C	–
	Antwort:		
	Schreiben erfolgreich	–	C1
	Befehl zurückgewiesen	–	C0[1)]
1	Len	01	01
2	Type[2)]		
	bei Merker M	86	86
	bei Merker N	87	87
3	Index[2)]	00 – 0F	00 – 0F
4	Data 1 (Low Byte)[3)]	00/01	00/01
5 – 7	Data 2 – 4	00	00

1) Mögliche Ursachen → Seite 173

2) Es gibt 16 M-Merker und 16 N-Merker. Die Adressierung der Merker differenziert sich durch „Type" und „Index": Mit „Type" wählen Sie den Merker-Typ „M" bzw. „N". Mit „Index" bestimmen Sie die Merker-Nummer.

3) Der Merker wird gesetzt, wenn ein Wert ungleich Null auf das Datenbyte geschrieben wird. Wird auf das Datenbyte „Data 1" der Wert „0" geschrieben, wird der Merker entsprechend zurückgesetzt.

Beispiel:
Es wird der Merker M13 gesetzt.

Byte	Bedeutung	Wert (hex), sendet	
		Master	Slave
0	Befehl: Schreiben	8C	–
	Antwort:		
	Schreiben erfolgreich	–	C1
	Befehl zurückgewiesen	–	C0[1)]
1	Len	01	01
2	Type		
	Merker M	86	86
3	Index	0C	0C
4	Data 1	01	00
5 – 7	Data 2 – 4	00	00

1) Mögliche Ursachen → Seite 173

Merker: M1 – M16/N1 – N16 lesen

Beim Lesen der Merker wird – im Gegensatz zum Schreiben – der gesamte Merkerdatenbereich eines Merker-Typs (M oder N) ausgelesen.

Telegrammaufbau

Byte	Bedeutung	Wert (hex), sendet	
		Master	Slave
0	Befehl: Lesen	88	–
	Antwort:		
	Lesen erfolgreich	–	C2
	Befehl zurückgewiesen	–	C0[1)]
1	Len	01	01
2	Type		
	Merker M	86	86
	Merker N	87	87
3	Index[2)]	00	00
4	Data 1 (Low Byte)	00	→ Tabelle 42
5	Data 2 (Low Byte)	00	→ Tabelle 42
6 – 7	Data 3 – 4	00	00

1) Mögliche Ursachen → Seite 173

2) Es gibt 16 M-Merker und 16 N-Merker. Die Adressierung der Merker differenziert sich durch „Type" und „Index":
Mit „Type" wählen Sie den Merker-Typ „M" bzw. „N".
Mit „Index" bestimmen Sie die Merker-Nummer.

Tabelle 42: Byte 4 bis 5: Data 1 bis 2

Data 1		Bit	7	6	5	4	3	2	1	0
M	**N**									
M1	N1									0/1
M2	N2								0/1	
…	…					…				
M8	N8		0/1							
Data 2		**Bit**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
M9	N9									0/1
M10	N10								0/1	
…	–					…				
M16	N16		0/1							

Beispiel:
Es werden die „N"-Merker gelesen:

Byte	Bedeutung	Wert (hex), sendet	
		Master	**Slave**
0	Befehl: Lesen	88	–
	Antwort:		
	Lesen erfolgreich	–	C2
	Befehl zurückgewiesen	–	C0[1)]
1	Len	01	01
2	Type		
	Merker N	87	87
3	Index	00	00
4	Data 1 (Low Byte)	00	04
5	Data 2 (Low Byte)	00	84
6 – 7	Data 3 – 4	00	00

1) Mögliche Ursachen → Seite 173

Die Merker N3, N11 und N16 sind gesetzt.

Betriebsstundenzähler: O1 – O4

Mit den nachfolgenden Befehlen lesen Sie den logischen Status der einzelnen Betriebsstundenzähler O1 – O4 aus.

Telegrammaufbau

Byte	Bedeutung	Wert (hex), sendet	
		Master	Slave
0	Befehl: Lesen	88	–
	Antwort:		
	Lesen erfolgreich	–	C2
	Befehl zurückgewiesen	–	C0[1)]
1	Len	01	01
2	Type	EF	EF
3	Index	00	00
4	Data 1 (Low Byte)	00	→ Tabelle 43
5 – 7	Data 2 – 4	00	00

1) Mögliche Ursachen → Seite 173

Tabelle 43: Byte 4 : Data 1

Data 1	Bit 7	6	5	4	3	2	1	0
O1								0/1
O2							0/1	
O3						0/1		
O4					0/1			
…	…	…	…	…				

Lokale P-Tasten: P1 – P4

Die lokalen P-Tasten sind die Display-Courser-Tasten des easy700-Basisgerätes. Die Tasten können Sie sowohl im Modus RUN als auch im Modus STOP abfragen.

→ Achten Sie nur darauf, dass die P-Tasten auch unter dem Menüpunkt „System" (im Basisgerät) aktiviert sind.

Bei den P-Tasten müssen Sie nur ein Byte übergeben.

Telegrammaufbau

Byte	Bedeutung	Wert (hex), sendet	
		Master	Slave
0	Befehl: Lesen	88	–
	Antwort:		
	Lesen erfolgreich	–	C2
	Befehl zurückgewiesen	–	C0[1)]
1	Len	01	01
2	Type	8A	8A
3	Index	00	00
4	Data 1 (Low Byte)	00	→ Tabelle 44
5 – 7	Data 2 – 4	00	00

1) Mögliche Ursachen → Seite 173

Tabelle 44: Byte 4: Data 1

Data 1	**Bit**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
P1									0/1
P2								0/1	
P3							0/1		
P4						0/1			
–					0				
–				0					
–			0						
–		0							

Beispiel:
Data 1 = 2_{hex} → P3 ist aktiv.

Lokale Ausgänge: Q1 – Q8

Die lokalen Ausgänge können Sie über den CANopen-Feldbus direkt lesen.

Telegrammaufbau

Byte	Bedeutung	Wert (hex), sendet	
		Master	Slave
0	Befehl: Lesen	88	–
	Antwort:		
	Lesen erfolgreich	–	C2
	Befehl zurückgewiesen	–	C0[1)]
1	Len	01	01
2	Type	85	85
3	Index	00	00
4	Data 1 (Low Byte)	00	→ Tabelle 45
5 – 7	Data 2 – 4	00	00

1) Mögliche Ursachen → Seite 173

Tabelle 45: Byte 4: Data 1

Data 1	Bit 7	6	5	4	3	2	1	0
Q1								0/1
Q2							0/1	
…				…				
Q8	0/1							

Beispiel:
Data 1 = 52_{hex} → Q2, Q5 und Q7 sind aktiv.

Ein-/Ausgänge von easyLink: R1 – R16/S1 – S8

Mit diesem Dienst können Sie die lokalen R- und S-Daten und die der NET-Teilnehmer (1 – 8) , die über easyLink transferiert werden, nochmals aus dem entsprechenden Abbild der easy700 lesen.

Telegrammaufbau

Byte	Bedeutung	Wert (hex), sendet	
		Master	Slave
0	Befehl: Lesen	88	–
	Antwort:		
	Lesen erfolgreich	–	C2
	Befehl zurückgewiesen	–	C0[1)]
1	Len	01	01
2	Type		
	bei R-Daten	88	88
	bei S-Daten	89	89
3	Index	00	00
4	Data 1 (Low Byte)	00	→ Tabelle 46
5	Data 2 (Low Byte)	00	→ Tabelle 46
6 – 7	Data 3 – 4	00	00

1) Mögliche Ursachen → Seite 173

Tabelle 46: Byte 5 bis 6: Data 1 bis 2

Data 1		**Bit**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RW	**SW**									
R1	S1									0/1
R2	S2								0/1	
…	…					…				
R8	S8		0/1							
Data 2		**Bit**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
R9	–									0/1
R10	–								0/1	
…	–					…				
R16	–		0/1							

Zeitglieder: T1 – T16

Mit den nachfolgenden Befehlen lesen Sie den logischen Status der einzelnen Zeitglied T1-T16 aus.

Telegrammaufbau

Byte	Bedeutung	Wert (hex), sendet	
		Master	Slave
0	Befehl: Lesen	88	–
	Antwort:		
	Lesen erfolgreich	–	C2
	Befehl zurückgewiesen	–	C0[1)]
1	Len	01	01
2	Type	ED	ED
3	Index	00	00
4	Data 1 (Low Byte)	00	→ Tabelle 47
5	Data 2 (Low Byte)	00	→ Tabelle 47
6 – 7	Data 3 – 4	00	00

1) Mögliche Ursachen → Seite 173

Tabelle 47: Byte 4 bis 5: Data 1 bis 2

Data 1	Bit	7	6	5	4	3	2	1	0
T1									0/1
T2								0/1	
…					…				
T8		0/1							
Data 2	**Bit**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
T9									0/1
T10								0/1	
…					…				
T16		0/1							

Jahresschaltuhr: Y1 – Y8

Mit den nachfolgenden Befehlen können Sie den logischen Status der einzelnen Jahresschaltuhren auslesen.

Telegrammaufbau

Byte	Bedeutung	Wert (hex), sendet	
		Master	Slave
0	Befehl: Lesen	88	–
	Antwort:		
	Lesen erfolgreich	–	C2
	Befehl zurückgewiesen	–	C0[1)]
1	Len	01	01
2	Type	91	91
3	Index	00	00
4	Data 1 (Low Byte)	00	→ Tabelle 48
5 – 7	Data 2 – 4	00	00

1) Mögliche Ursachen → Seite 173

Tabelle 48: Byte 4: Data 1

Data 1	Bit 7	6	5	4	3	2	1	0
HY1								0/1
HY2							0/1	
HY3						0/1		
HY4					0/1			
HY5				0				
HY6			0					
HY7		0						
HY8	0							

Beispiel:
Data 1 = 1_{hex} → HY2 ist aktiv

Masterreset: Z1 – Z3

Telegrammaufbau

Byte	Bedeutung	Wert (hex), sendet	
		Master	Slave
0	Befehl: Lesen	88	–
	Antwort:		
	Lesen erfolgreich	–	C2
	Befehl zurückgewiesen	–	C0[1)]
1	Len	01	01
2	Type	93	93
3	Index	00	00
4	Data 1 (Low Byte)	00	→ Tabelle 49
5 – 7	Data 2 – 4	00	00

1) Mögliche Ursachen → Seite 173

Tabelle 49: Byte 4: Data 1

Data 1	Bit	7	6	5	4	3	2	1	0
Z1 für Ausgänge Q									0/1
Z2 für Merker M								0/1	
Z3 für Ausgänge und Merker							0/1		
…		0	0	0	0	0			

Wochenschaltuhr: 1 – 8

Mit den nachfolgenden Befehlen lesen Sie den logischen Status der einzelnen Wochenschaltuhren aus.

Telegrammaufbau

Byte	Bedeutung	Wert (hex), sendet	
		Master	Slave
0	Befehl: Lesen	88	–
	Antwort:		
	Lesen erfolgreich	–	C2
	Befehl zurückgewiesen	–	C0[1)]
1	Len	01	01
2	Type	90	90
3	Index	00	00
4	Data 1 (Low Byte)	00	→ Tabelle 50
5 – 7	Data 2 – 4	00	00

1) Mögliche Ursachen → Seite 173

Tabelle 50: Byte 4: Data 1

Data 1	Bit 7	6	5	4	3	2	1	0
HW1								0/1
HW2							0/1	
HW3						0/1		
HW4					0/1			
HW5				0				
HW6			0					
HW7		0						
HW8	0							

Beispiel:
Data 1 = 2_{hex} → 3 ist aktiv.

Funktionsbaustein-Daten lesen/schreiben

Beachten Sie bitte auch die entsprechende Beschreibung zu den Funktionsbausteinen, die im easy700-Handbuch (MN05013003Z-DE; frühere Bezeichnung AWB2528-1508D) oder in der easySoft-Hilfe angeführt sind.

Allgemeine Hinweise

Grundsätzlich müssen Sie beim Arbeiten mit den Funktionsbausteinen Folgendes beachten:

- Die entsprechenden Daten werden im Intelformat übergeben: das erste Byte ist das Low Byte (Byte 5) und das letzte das High Byte (Byte 8).
- Die Datenlänge beträgt bis zu 4 Byte. Alle Werte müssen Sie im hexadezimalen Zahlenformat übergeben.

Übersicht

Operanden	Bedeutung	Lesen/ schreiben	Type (hex)	Seite
A1 – A16	„Analogwertvergleicher/Schwellwertschalter: A1 – A16“	Lesen/ schreiben	8D	156
C1 – C16	„Zählerrelais: C1 – C16“	Lesen/ schreiben	8F	159
O1 – O4	„Betriebsstundenzähler: O1 – O4“	Lesen/ schreiben	92	162
T1 – T16	„Zeitrelais: T1 – T16“	Lesen/ schreiben	8E	164
Y1 – Y8	„Jahresschaltuhr: Y1 – Y8“	Lesen/ schreiben	A2	167
⏲1 – ⏲8	„Wochenschaltuhr: ⏲1 – ⏲8“	Lesen/ schreiben	A1	170

Analogwertvergleicher/Schwellwertschalter: A1 – A16

Telegrammaufbau

Byte	Bedeutung	Wert (hex), sendet	
		Master	Slave
0	Befehl:		
	Lesen	89	–
	Schreiben	8D	–
	Antwort:		
	Lesen erfolgreich	–	C2
	Schreiben erfolgreich	–	C1
	Befehl zurückgewiesen	–	C0[1)]
1	Type	8D	8D
2	Instanz[2)]	00 – 0F	00 – 0F
3	Index	→ Tabelle 51	
4 – 7	Data 1 – 4	abhängig vom Index, → Tabelle 52	

1) Mögliche Ursachen → Seite 173

2) easy stellt 16 Analogwertvergleicher A1 bis A16 zur Verfügung. Diese können über die Instanz (0 – F) angesprochen werden.

Tabelle 51: Operandenübersicht

Index (hex)	Operand		Lesen	Schreiben
00	Parameter → Tabelle 52		×	
01	Kontrollbyte → Tabelle 53		×	
02	Vergleichswert 1	I1[2)]	×	c[1)]
03	Vergleichswert 2	I2[2)]	×	c[1)]
04	Verstärkungsfaktor für I1 (I1 = F1 × I1)	F1[2)]	×	c[1)]
05	Verstärkungsfaktor für I2 (I2 = F2 × I2)	F2[2)]	×	c[1)]
06	Offset für den Wert I1 (I1 = OS + Istwert an I1)	OS[2)]	×	c[1)]
07	Schalthysterese für den Wert I2	HY[2)]	×	c[1)]

1) Der Wert kann nur geschrieben werden, wenn er im Programm mit einer Konstanten beschaltet ist.

2) In den Datenbytes Data 1 – Data 2 wird ein 16-Bit-Wert übergeben. Dabei ist zu beachten, dass das Low-Byte in Data 1 (Byte 5) und das High-Byte in Data 2 (Byte 8) geführt wird.
Beispiel: $5327_{dez} = 14CF_{hex}$ → Data 1 = 0xCF, Data 2 = 0x14

Tabelle 52: Index 00 – Parameter

Bedeutung	Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Erscheint im Parametermenü																	
ja/nein																	0/1
Vergleich auf																	
FB nicht benutzt														0	0	0	
EQ (=)														0	0	1	
GE (≧)														0	1	0	
LE (≦)														0	1	1	
GT (>)														1	0	0	
LT (<)														1	0	1	
Verwendung als Konstante und somit beschreibbar																	
I1= Konstante													0/1				
F1= Konstante												0/1					
I2= Konstante											0/1						
F2 = Konstante										0/1							
OS = Konstante									0/1								
HY = Konstante								0/1									
Nicht verwendet		0	0	0	0	0	0										

Beispiel:
Data 1 (Byte 4) = 0xA3, Data 2 (Byte 5) = 0x03
→ Resultierender 16-Bit-Wert = $03A3_{hex}$

Bedeutung: HY, OS, F2, F1 sind mit einer Konstanten beschaltet; I1, I2 sind mit einer Variablen z. B. I7, I8 C2…usw. beschaltet; erscheint im Parametermenü;

Der Ausgang des Analogwertvergleichers wird aktiv, solange die Gleichung (I1 × F1) + OS = (I2 × F2) + HY erfüllt ist.

Tabelle 53: Index 01 – Kontrollbyte

	Bit	7	6	5	4	3	2	1	0
FB-Ausgang Data 3		–	–	–	–	–	–	–	Q1[1)]

1) Zustand „1", wenn die Vergleichsbedingung erfüllt ist.

Zählerrelais: C1 – C16

Telegrammaufbau

Byte	Bedeutung	Wert (hex), sendet	
		Master	Slave
0	Befehl:		
	Lesen	89	–
	Schreiben	8D	–
	Antwort:		
	Lesen erfolgreich	–	C2
	Schreiben erfolgreich	–	C1
	Befehl zurückgewiesen	–	C0[1]
1	Type	8F	8F
2	Instanz[2]	00 – 0F	00 – 0F
3	Index	→ Tabelle 54	
4 – 7	Data 1 – 4	abhängig vom Index, → Tabelle 55	

1) Mögliche Ursachen → Seite 173

2) easy stellt 16 Zählerrelais C1 bis C16 zur Verfügung. Diese können über die Instanz (0 – F) angesprochen werden.

Tabelle 54: Operandenübersicht

Index (hex)	Operand		Lesen	Schreiben
00	Parameter → Tabelle 55		×	
01	Kontrollbyte → Tabelle 56		×	
02	Istwert	S1[2]	×	c[1]
03	Zählersollwert 2	S2[2]	×	c[1]

1) Der Wert kann nur geschrieben werden, wenn er im Programm mit einer Konstanten beschaltet ist.

2) In den Datenbytes Data 1 – Data 2 wird ein 16-Bit-Wert übergeben. Dabei ist zu beachten, dass das Low-Byte in Data 1 und das High-Byte in Data 2 geführt wird.

Tabelle 55: Index 00 – Parameter

Bedeutung	Bit	7	6	5	4	3	2	1	0
Erscheint im Parametermenü									
ja/nein									0/1
Zählermodus									
FB nicht benutzt							0	0	
Vor-/Rückwärtszähler (N)							0	1	
Schneller Vor-/Rückwärtszähler (H)							1	0	
Frequenzmesser (F)							1	1	
Verwendung als Konstante und somit beschreibbar									
Zählersollwert S1						0/1			
Nicht benutzte Bits		–	–	–	–				

Beispiel:
Data 1 (Byte 4) = 0x07

Bedeutung:
Die Werte erscheinen im Parametermenü. Der Zähler wird im Modus des Frequenzmessers benutzt. Der Zählersollwert S1 ist nicht mit einer Konstanten beschaltet und somit auch nicht beschreibbar.

Tabelle 56: Index 01 – Kontrollbyte

Data 1	Bit	7	6	5	4	3	2	1	0
FB-Ausgang		–	–	–	–	C[4)]	RE[3)]	D[2)]	Q1[1)]

1) Schaltkontakt
2) Zählrichtungsangabe: Zustand „0" = vorwärts zählen
   Zustand „1" = rückwärts zählen
3) Reset, das Zeitrelais wird zurückgesetzt (Resetspule)
4) Zählerspule, zählt bei jeder positiven Flanke

Beispiel:
Der Istwert von C3 soll gelesen werden:

Byte	Bedeutung	Wert (hex), sendet	
		Master	Slave
0	Befehl: Lesen	89	–
	Antwort: Lesen erfolgreich	–	C2
1	Type	8F	8F
2	Instanz	02	02
3	Index	02	02
4	Data1	00	12
5	Data 2	00	03
6	Data 3	00	00
7	Data 4	00	00

Erklärung:

Data 1 = 12
Data 2 = 03
→ resultierender 16-Bit-Wert = $0312_{hex} = 786_{dez}$

Zählerstand = 786

Betriebsstundenzähler: O1 – O4

Telegrammaufbau

Byte	Bedeutung	Wert (hex), sendet	
		Master	Slave
0	Befehl:		
	Lesen	89	–
	Schreiben	8D	–
	Antwort:		
	Lesen erfolgreich	–	C2
	Schreiben erfolgreich	–	C1
	Befehl zurückgewiesen	–	C0[1]
1	Type	92	92
2	Instanz[2]	00 – 03	00 – 03
3	Index	→ Tabelle 57	
4 – 7	Data 1 – 4	abhängig vom Index, → Tabelle 58	

1) Mögliche Ursachen → Seite 173

2) easy stellt 4 Betriebsstundenzähler O1 bis O4 zur Verfügung. Diese können über die Instanz (0 – 3) angesprochen werden.

Tabelle 57: Operandenübersicht

Index (hex)	Operand		Lesen	Schreiben
00	Parameter → Tabelle 58		×	
01	Kontrollbyte → Tabelle 59		×	
02	Istwert	S1[2]	×	c[1]
03	Zählersollwert 2	S2[2]	×	c[1]

1) Der Wert kann nur geschrieben werden, wenn er im Programm mit einer Konstanten beschaltet ist.

2) In den Datenbytes Data 1 – Data 4 wird ein 32-Bit-Wert übergeben. Dabei ist zu beachten, dass das Low-Byte in Data 1 und das High-Byte in Data 4 geführt wird.

Tabelle 58: Index 00 – Parameter

Bedeutung	**Bit**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Erscheint im Parametermenü									
ja/nein									0/1
Verwendung im Programm									
Sollwert S1								0/1	
Nicht benutzte Bits		–	–	–	–	–	–		

Beispiel:
Data 1 (Byte 4) = 0x01

Bedeutung:
Die Werte erscheinen im Parametermenü.

Tabelle 59: Index 01 – Kontrollbyte

Data 1	**Bit**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
FB-Ausgang		–	–	–	–	–	RE[3)]	EN[2)]	Q1[1)]

1) Schaltkontakt
2) Enable, das Zeitrelais wird gestartet (Triggerspule)
3) Reset, das Zeitrelais wird zurückgesetzt (Resetspule)

Beispiel:
Index 02/03

Übergebene Werte: Data 1 0x21
Data 2 0x23
Data 3 0x40
Data 4 0x00

Resultierender Wert: $00402321_{hex} = 4203297_{dez}$

Zeitrelais: T1 – T16

Telegrammaufbau

Byte	Bedeutung	Wert (hex), sendet	
		Master	Slave
0	Befehl:		
	Lesen	89	–
	Schreiben	8D	–
	Antwort:		
	Lesen erfolgreich	–	C2
	Schreiben erfolgreich	–	C1
	Befehl zurückgewiesen	–	C0[1]
1	Type	8E	8E
2	Instanz[2]	00 – 0F	00 – 0F
3	Index	→ Tabelle 60	
4 – 7	Data 1 – 4	abhängig vom Index, → Tabelle 61	

1) Mögliche Ursachen → Seite 173

2) easy stellt 16 Zeitrelais T1 bis T16 zur Verfügung. Diese können über die Instanz (0 – F) angesprochen werden.

Tabelle 60: Operandenübersicht

Index (hex)	Operand		Lesen	Schreiben
00	Parameter → Tabelle 61		×	
01	Kontrollbyte → Tabelle 62		×	
02	Istwert 1	T	×	c[1]
03	Zeitsollwert 1	S1[2]	×	c[1]
04	Zeitsollwert 2	S2[2]	×	c[1]

1) Der Wert kann nur geschrieben werden, wenn er im Programm mit einer Konstanten beschaltet ist.

2) In den Datenbytes Data 1 – Data 2 wird ein 16-Bit-Wert übergeben. Dabei ist zu beachten, dass das Low-Byte in Data 1 und das High-Byte in Data 2 geführt wird.

Tabelle 61: Index 00 – Parameter

Bedeutung	**Bit**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Erscheint im Parametermenü									
ja/nein									0/1
Timer-Modus									
Ansprechverzögert						0	0	0	
Rückfallverzögert						0	0	1	
Ansprechverzögert mit Zufalls-Sollwert						0	1	0	
Rückfallverzögert mit Zufalls-Sollwert						0	1	1	
Ansprechverzögert und Rückfallverzögert (zwei Zeitsollwerte)						1	0	0	
Ansprechverzögert und Rückfallverzögert, jeweils mit Zufalls-Sollwert (zwei Zeitsollwerte)						1	0	1	
Impulsgeber						1	1	0	
Blink-Relais (zwei Zeitsollwerte)						1	1	1	
Zeitbasis									
FB nicht benutzt				0	0				
Millisekunde: S				0	1				
Sekunde: M:S				1	0				
Minute: H:M				1	1				
Verwendung als Konstante und somit beschreibbar									
Zeitsollwert S1			0/1						
Zeitsollwert S2		0/1							

Beispiel:
Data 1 (Byte 4) = 0xAC

Bedeutung:
Die Werte erscheinen im Parametermenü. Der Timer wird im Modus des Impulsgebers mit der Zeitbasis „Sekunde“ benutzt. Der Zeitsollwert S1 ist mit einer Konstanten beschaltet und der Zeitsollwert S2 ist mit einer Variablen z. B. I7, I8 C2…usw. beschaltet.

Tabelle 62: Index 01 – Kontrollbyte

	Bit	7	6	5	4	3	2	1	0
FB-Ein-/Ausgang Data 3		–	–	–	–	ST[4]	RE[3]	EN[2]	Q1[1]

1) Schaltkontakt
2) Enable, das Zeitrelais wird gestartet (Triggerspule)
3) Reset, das Zeitrelais wird zurückgesetzt (Resetspule)
4) Stopp, das Zeitrelais wird gestoppt (Stoppspule)

Beispiel:
Der Zeitsollwert 1 soll gelesen werden:

Byte	Bedeutung	Wert (hex), sendet	
		Master	Slave
0	Befehl: Lesen	89	–
	Antwort: Lesen erfolgreich	–	C2
1	Type	8E	8E
2	Instanz	00	00
3	Index	03	03
4	Data1	00	4C
5	Data 2	00	06
6	Data 3	00	00
7	Data 4	00	00

Erklärung:

Data 1 = 4C
Data 2 = 06
→ resultierender 16-Bit-Wert = $064C_{hex} = 1612_{dez}$

Bedeutung je nach eingestellter Zeitbasis:

Millisekunde	S	16120 ms	16,120 s
Sekunde	M:S	1620 s	26:52 Minuten
Minute	H:M	1612 min	67:04 Stunden

Jahresschaltuhr: Y1 – Y8

Telegrammaufbau

Byte	Bedeutung	Wert (hex), sendet	
		Master	**Slave**
0	Befehl:		
	Lesen	89	–
	Schreiben	8D	–
	Antwort:		
	Lesen erfolgreich	–	C2
	Schreiben erfolgreich	–	C1
	Befehl zurückgewiesen	–	C0[1)]
1	Type	A2	A2
2	Instanz[2)]	00 – 07	00 – 07
3	Index	→ Tabelle 63	
4 – 7	Data 1 – 4	abhängig vom Index, → Tabelle 64	

1) Mögliche Ursachen → Seite 173

2) easy stellt 8 Jahresschaltuhren Y1 bis Y8 zur Verfügung. Diese können über die Instanz (0 – 7) angesprochen werden.

Tabelle 63: Operandenübersicht

Index (hex)	Operand	Lesen	Schreiben
00	Parameter → Tabelle 64	×	
01	Kontrollbyte → Tabelle 65	×	
	Kanal A	×	c[1)]
11	Zeitpunkt ein	×	c[1)]
12	Zeitpunkt aus	×	c[1)]
	Kanal B	×	c[1)]
21	Zeitpunkt ein	×	c[1)]
22	Zeitpunkt aus	×	c[1)]
	Kanal C	×	c[1)]
31	Zeitpunkt ein	×	c[1)]
32	Zeitpunkt aus	×	c[1)]
	Kanal D	×	c[1)]
41	Zeitpunkt ein	×	c[1)]
42	Zeitpunkt aus	×	c[1)]

1) Der Wert kann nur geschrieben werden, wenn er im Programm mit einer Konstanten beschaltet ist.

2) In den Datenbytes Data 1 – Data 3 werden die Schaltpunkte übergeben.

Tabelle 64: Index 00 – Parameter

Bedeutung	Bit	7	6	5	4	3	2	1	0
Erscheint im Parametermenü									
Kanal A									0/1
Kanal B								0/1	
Kanal C							0/1		
Kanal D						0/1			
Nicht benutzte Bits		–	–	–	–				

Beispiel:
Data 1 (Byte 4) = 0x03 → Die Werte der Jahresschaltuhr von Kanal A und B erscheinen im Parametermenü.

Tabelle 65: Index 01 – Kontrollbyte

Data 1	**Bit**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
FB-Ausgang		–	–	–	–	–	–	–	Q1[1)]

1) Zustand „1", wenn die Zählbedingung erfüllt ist.

Kanal A, Index 11/12

Index 0x11 Kanal A Zeitpunkt des Einschaltens
Index 0x12 Kanal A Zeitpunkt des Ausschaltens
Data 1 (Byte 4) – Tag
Data 2 (Byte 5) – Monat
Data 3 (Byte 6) – Jahr

Beispiel:
Die Jahresschaltuhr Kanal A soll am 21.04.2004 eingeschaltet werden.

Index = 0x11
Data 1 = 0x15
Data 2 = 0x04
Data 3 = 0x04

Die Jahresschaltuhr Kanal B soll am 05.11.2012 ausgeschaltet werden.

Index = 0x22
Data 1 = 0x05
Data 2 = 0x0B
Data 3 = 0x0C

Wochenschaltuhr: 1 – 8

Telegrammaufbau

Byte	Bedeutung	Wert (hex), sendet	
		Master	Slave
0	Befehl:		
	Lesen	89	–
	Schreiben	8D	–
	Antwort:		
	Lesen erfolgreich	–	C2
	Schreiben erfolgreich	–	C1
	Befehl zurückgewiesen	–	C0[1)]
1	Type	A1	A1
2	Instanz[2)]	00 – 07	00 – 07
3	Index	→ Tabelle 66	→ Tabelle 66
4 – 7	Data 1 – 4	abhängig vom Index, → Tabelle 67	

1) Mögliche Ursachen → Seite 173

2) easy stellt 8 Wochenschaltuhren 1 bis 8 zur Verfügung. Diese können über die Instanz (0 – 7) angesprochen werden.

Tabelle 66: Operandenübersicht

Index (hex)	Operand	Lesen	Schreiben
00	Parameter → Tabelle 67	×	
01	Kontrollbyte → Tabelle 68	×	
11	Kanal A Tag an/aus	×	c[1)]
12	Zeit ein	×	c[1)]
13	Zeit aus	×	c[1)]
21	Kanal B Tag an/aus	×	c[1)]
22	Zeit ein	×	c[1)]
23	Zeit aus	×	c[1)]
31	Kanal C Tag an/aus	×	c[1)]
32	Zeit ein	×	c[1)]
33	Zeit aus	×	c[1)]
41	Kanal D Tag an/aus	×	c[1)]
42	Zeit ein	×	c[1)]
43	Zeit aus	×	c[1)]

1) Der Wert kann nur geschrieben werden, wenn er im Programm mit einer Konstanten beschaltet ist.

2) In den Datenbytes Data 1 – Data 4 wird ein 16-Bit-Wert übergeben. Dabei ist zu beachten, dass das Low-Byte in Data 1 und das High-Byte in Data 2 geführt wird.

Tabelle 67: Index 00 – Parameter

Bedeutung	Bit	7	6	5	4	3	2	1	0
Erscheint im Parametermenü									
Kanal A									0/1
Kanal B								0/1	
Kanal C							0/1		
Kanal D						0/1			
Nicht benutzte Bits		–	–	–	–				

Beispiel:
Data 1 (Byte 4) = 0x03

Bedeutung:
Die Werte der Wochenschaltuhr WH… von Kanal A und B erscheinen im Parametermenü.

Tabelle 68: Index 01 – Kontrollbyte

Data 1	Bit	7	6	5	4	3	2	1	0
FB-Ausgang		–	–	–	–	–	–	–	Q1[1)]

1) Zustand „1“, wenn die Zählbedingung erfüllt ist.

Kanal A, Index 11/12/13

Index 0x11 Kanal A Wochentag an/aus
Data 1 (Byte 4) – Wochentag an
Data 2 (Byte 5) – Wochentag aus
0x01 = Sonntag … 0x07 = Samstag

Wenn der Kanal nicht verwendet wird, ist der 16-Bit-Wert gleich 0x00.

Index 0x12 – Zeit ein (2 Byte)
Index 0x13 – Zeit aus (2 Byte)
Data 1 (Byte 4) – Stunde
Data 2 (Byte 5) – Minute

Beispiel: Zeit ein um 13:43 Uhr
Data 1 = 0x0D
Data 2 = 0x2B

Analyse – Fehlercodes über easyLink

Das Basisgerät easy700 liefert im Fall eines nicht korrekt gewählten Betriebsmodus oder eines ungültigen Telegramms einen definierten Fehlercode zurück. Die Übergabe des Fehlercodes ist wie folgt aufgebaut:

Telegrammaufbau

Byte	Bedeutung	Slave sendet (Wert hex)
0	Antwort	
	Befehl zurückgewiesen	C0
1	Type	00
2	Instanz	00
3	Index	00
4	Fehlercode	→ Tabelle 69

Tabelle 69: Fehlercodes

Fehler-code	Beschreibung
0x01	Es wurde ein unbekanntes Telegramm versendet.
0x02	Es wurde ein unbekanntes Objekt versendet.
0x03	Es wurde ein unbekanntes Kommando versendet.
0x04	Es wurde eine ungültige Instanz versendet.
0x05	Es wurde ein ungültiger Parametersatz verwendet.
0x06	Es wurde versucht, eine Variable zu beschreiben die keine Konstante ist.
0x0C	Das Gerät befindet sich in einem ungültigen Gerätemodus. STOP → RUN oder RUN → STOP
0x0D	Es erfolgte ein ungültiger Displayzugriff. Bitte verlassen Sie die Menü-Ebene, sodass die Status-Anzeige im Display angezeigt wird. Das Schreiben der Uhr ist nicht möglich.
0xF0	Es wurde versucht, einen unbekannten Parameter anzusteuern.
0xF1	Nicht erlaubter Wert

10 SDO – Steuerbefehle für easy800/MFD

Die OV-Einträge „Status" (3020_{hex}), „Befehl" (3021_{hex}) und „Antwort" (3022_{hex}) stellen die Schnittstelle zum erweiterten Datenaustausch mit easy800 und MFD am CANopen-Kommunikationsbus dar. Damit können Sie Dienste aus folgenden Bereichen übertragen:

- „Datum und Uhrzeit lesen/schreiben" (Seite 177)
- „Abbilddaten lesen/schreiben" (Seite 182)
- „Funktionsbaustein-Daten lesen/schreiben" (Seite 202).

Damit die Daten sicher vom Master zum Slave und umgekehrt über CANopen übertragen werden, ist das SDO-CANopen-Protokoll notwendig (→ Seite 60).

Achtung!
Während der Benutzung eines Steuerbefehls behalten die Ein- und Ausgangsdaten den Zustand vor Aufruf des Steuerbefehls. Erst wenn der Datenaustausch des Steuerbefehls beendet ist, sind die Ein-/Ausgangsdaten wieder aktuell.

Warnung!
Es dürfen nur die für den Befehlscode angegebenen Werte verwendet werden. Überprüfen Sie die Werte, die Sie schreiben, um Fehlfunktionen zu vermeiden.

Versionsgeschichte

Die nachfolgende Tabelle gibt eine Übersicht über die Änderungen und Neuerungen bei den verschiedenen Geräteversionen des easy800:

Auswirkung auf easyLink	easy800, Geräteversion ab 02	ab 04	ab 05	ab 07
Unterstützung des kompletten PDO-Zugriffs				
R-Daten sind schreibbar	✓	✓	✓	✓
S-Daten sind lesbar	✓	✓	✓	✓
Unterstützung des kompletten SDO-Zugriffs				
Funktionsbausteine	–	A, AR, BC, BT, BV, C, CF, CH, CI, CP, D, DB, DC, FT, GT, HW, HY, LS, MR, NC, OT, PT, PW, SC, ST, T, VC		
	–	–	–	DG, JC, MX, PO, SP, SR, TB
Abbilddaten				
Lesen	–	IW, IA, ID, QW, QA, P, RW, SW, M, MB, MW, MD		
Schreiben	–	QW, QA, M, MB, MW, MD	M, MB, MW, MD	
Uhrenfunktionalität	–	✓	✓	✓
Regel-Option bei der Winter-/ Sommerzeitumstellung	–	–	✓	✓

Datum und Uhrzeit lesen/schreiben

→ Beachten Sie bitte auch die entsprechende Beschreibung zur Echtzeituhr im easy800-Handbuch (MN04902001Z-DE; frühere Bezeichnung AWB2528-1423D).

Telegrammaufbau

Byte	Bedeutung	Wert (hex), sendet	
		Master	Slave
0	Befehl		
	Lesen	93	–
	Schreiben	B3	–
	Antwort		
	Lesen erfolgreich	–	C2
	Schreiben erfolgreich	–	C1
	Befehl zurückgewiesen	–	C0
1	Len	05	05
2	Index	00	00
3 – 7	Data 1 – 5		
	Beim Lesen	00	→ Tabelle 70
	Beim Schreiben	→ Tabelle 70	00

Tabelle 70: Byte 3 bis 7: Data 1 bis 5

Byte	Inhalt		Wert (hex)
3	Data 1	Stunde (0 bis 23)	00 – 17
4	Data 2	Minute (0 bis 59)	00 – 3B
5	Data 3	Tag (1 bis 31; abhängig von Monat und Jahr)	01 – 1F
6	Data 4	Monat (1 bis 12)	01 – 0C
7	Data 5	Jahr (0 – 99, entspricht 2000 – 2099)	00 – 63

Winter-/Sommerzeit, Zeitumstellung

Telegrammaufbau

Byte	Bedeutung	Wert (hex), sendet	
		Master	Slave
0	Befehl		
	Lesen	93	–
	Schreiben	B3	–
	Antwort		
	Lesen erfolgreich	–	C2
	Schreiben erfolgreich	–	C1
	Befehl zurückgewiesen	–	C0
1	Len	05	05
2	Index		
	Sommer-/Winterzeit	01 → Tabelle 71	→ Tabelle 71
	Winterzeit (nach „Area" = Regel")[1)]	02 → Tabelle 72	02 → Tabelle 72
3 – 7	Data 1 – 5		
	Beim Lesen	00	abhängig vom Index, → Tabelle 71, 72
	Beim Schreiben	abhängig vom Index, → Tabelle 71, 72	00

1) Detaillierte Einstellmöglichkeit für easy800/MFD ab Version 05

Tabelle 71: Index 01 – Sommer-/Winterzeit-Umschaltung

Byte	Inhalt		Wert (hex)
3	Data 1	Area	
		keine	00
		manuell	01
		automatische EU	02
		automatische GB	03
		automatische US	04
		Regel[1)]	05
für „Area" = „manuell":			
4	Data 2	Set Sommerzeit Tag 1 bis 28, 29, 30, 31 (abh. von Monat und Jahr)	00 – 3B
5	Data 3	Set Sommerzeit Monat (1 – 12)	01 – 1F
6	Data 4	Set Winterzeit Tag 1 bis 28, 29, 30, 31 (abh. von Monat und Jahr)	01 – 0C
7	Data 5	Set Winterzeit Monat (1 – 12)	00 – 63
für „Area" = „Regel"[1)]:			
4 – 7	Data 2 – 5	Schaltregel Sommerzeit	→ Tabelle 73

1) Detaillierte Einstellmöglichkeit für easy800/MFD ab Version 05.

Tabelle 72: Index 02 – Winterzeit
(nur gültig, wenn Area = „Regel" gewählt worden ist)

Byte	Inhalt		Wert (hex)
3	Data 1	Area = Regel	01
4 – 7	Data 2 – 5	Schaltregel Winterzeit	→ Tabelle 73

Bitfeld „Schaltregel"

Lesen Sie hierzu bitte die Beschreibung im easy800-Handbuch (MN04902001Z-DE; vormals AWB2528-1423D). Die nachfolgende Tabelle gibt Ihnen die Vorschrift zum Zusammensetzen der entsprechenden Datenbytes an.

Tabelle 73: Bitfeld Schaltregel

	Data 5			Data 4			Data 3			Data 2			
Bit	31 30 29	28 27 26	25 24	23 22 21 20 19	18 17 16 15	14 13 12 11 10	9 8	7 6 5 4	3 2 1 0				
	Regel_1	Tag	Regel_2	Tag	Monat	Zeit der Umstellung			Differenz				
	0: am	0: So	0: im	0 bis 30	0 bis 11	Stunde: 0 bis 23		Minute: 0 bis 59	0: 0:30h				
	1: am ersten	1: Mo	1: nach dem						1: 1:00h				
	2: am zweiten	2: Di	2: vor dem						2: 1:30h				
	3: am dritten	3: Mi							3: 2:00h				
	4: am vierten	4: Do							4: 2:30h				
	5: am letzten	5: Fr							5: 3:00h				
		6: Sa											

Beispiel

Die Echtzeituhr der easy800 soll auf Freitag 23.05.2003, 14:36 Uhr, gestellt werden.

Byte	Bedeutung	Wert (hex), sendet	
		Master	Slave
0	Befehl: Schreiben	B3	–
	Antwort: Schreiben erfolgreich	–	C1
1	Len	05	05
2	Index	00	00
3	Data 1 (Stunde)	0E	00
4	Data 2 (Minute)	24	00
5	Data 3 (Tag)	17	00
6	Data 4 (Monat)	05	00
7	Data 5 (Jahr)	03	00

→ Alle Werte müssen hexadezimal übergeben werden.

Abbilddaten lesen/schreiben

→ Beachten Sie bitte auch die entsprechende Beschreibung zu den möglichen Abbilddaten, die im easy800-Handbuch (MN04902001Z-DE; frühere Bezeichnung AWB2528-1423D) oder in der easySoft-Hilfe angeführt sind. Desweiteren gilt der Abschnitt „Generelles zum Arbeiten mit Abbilddaten" auf Seite 91 auch für easy700.

Übersicht

Operanden	Bedeutung	Lesen/ schreiben	Befehl (hex)	Seite
IA1 – IA4	„Lokale Analog-Eingänge: IA1 – IA4"	lesen	02	183
ID1 – ID16	„Lokale Diagnose: ID1 – ID16"	lesen	03	185
IW0	„Lokale Eingänge: IW0"	lesen	01	187
IW1 – IW8	„Eingänge der Netzwerkteilnehmer: IW1 – IW8"	lesen	01	189
M…	„Merker: M…"	lesen/ schreiben	0B – 0E	190
P1 – P4	„Lokale P-Tasten: P1 – P4"	lesen	06	193
QA1	„Lokaler Analog-Ausgang: QA1"	lesen/ schreiben	05	195
QW0, QW1 – QW8	„Lokale Ausgänge: QW0/ Ausgänge der Netzwerkteilnehmer: QW1 – QW8"	lesen/ schreiben	04	196
R1 – R16 S1 – S8	„Ein-/Ausgänge von easyLink: RW/SW"	lesen	07/09	198
RN1 – RN32 SN1 – SN32	„Receive-Data Netz: RN1 – RN32/ Send-Data Netz: SN1 – SN32"	lesen	08/0A	200

Lokale Analog-Eingänge: IA1 – IA4

Die am Basisgerät easy800 bzw. MFD vorhandenen Analogeingänge können Sie direkt über den CANopen auslesen. Hierbei wird der 16-Bitwert im Intelformat übergeben (Low Byte first).

Telegrammaufbau

Byte	Bedeutung	Wert (hex), sendet	
		Master	Slave
0	Befehl: Lesen	91	–
	Antwort:		
	Lesen erfolgreich	–	C2
	Befehl zurückgewiesen	–	C0
1	Len	02	02
2	Type	02	02
3	Index	01 – 04[1)]	01 – 04[1)]
4	Data 1 (Low Byte)	00	siehe Beispiel
5	Data 2 (High Byte)	00	siehe Beispiel
6 – 7	Data 3 – 4	00	00

1) 01 = Analogeingang I7
02 = Analogeingang I8
03 = Analogeingang I11
04 = Analogeingang I12

Beispiel

Es liegt ein Spannungspegel am Analogeingang 1 an. Die entsprechenden Telegramme zum Lesen des Analogwertes sehen wie folgt aus:

Byte	Bedeutung	Wert (hex), sendet	
		Master	Slave
0	Befehl: Lesen	91	–
	Antwort: Lesen erfolgreich	–	C2
1	Len	02	02
2	Type	02	02
3	Index	01[1)]	01[1)]
4	Data 1	00	D9
5	Data 2	00	02
6	Data 3	00	00
7	Data 4	00	00

1) 01 = Analogeingang 1

Byte 4 – Data 1 (Low Byte): $D9_{hex}$

Byte 5 – Data 2 (High Byte): 02_{hex}

→ entsprechender 16-Bitwert: $02D9_{hex}$ = 729 (7,29 V)

Lokale Diagnose: ID1 – ID16

Die lokale Diagnose (ID1 – ID8) gibt den Zustand der einzelnen NET-Teilnehmer an. Die Verbindung zum Remote-Teilnehmer (nur MFD) wird über ID9 angezeigt.

Telegrammaufbau

Byte	Bedeutung	Wert (hex), sendet	
		Master	Slave
0	Befehl: Lesen	91	–
	Antwort:		
	Lesen erfolgreich	–	C2
	Befehl zurückgewiesen	–	C0
1	Len	02	02
2	Type	03	03
3	Index	00	00
4	Data 1 (Low Byte)	00	→ Tabelle 74
5	Data 2 (High Byte)	00	→ Tabelle 74
6 – 7	Data 3 – 4	00	00

Tabelle 74: Byte 4 bis 5: Data 1 bis 2

Data 1	**Bit**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ID1									0/1
ID2								0/1	
…					…				
ID8		0/1							
Data 2	**Bit**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ID9									0/1
–								1	
…					…				
–		1							

0/1= aktiver/inaktiver NET-Teilnehmer, –= nicht belegt

Beispiel

Data 1 = F8, Data 2 = FF → Im easy-Netzwerk NET sind die drei Teilnehmer mit der NET-ID 1, 2, 3 vorhanden.

Lokale Eingänge: IW0

Mit diesem Befehlsstring können Sie die lokalen Eingänge des Basisgerätes easy800 bzw. MFD auslesen. Das entsprechende Eingangswort ist hierbei im Intel-Format abgelegt.

Telegrammaufbau

Byte	Bedeutung	Wert (hex), sendet	
		Master	Slave
0	Befehl: Lesen	91	–
	Antwort:		
	Lesen erfolgreich	–	C2
	Befehl zurückgewiesen	–	C0
1	Len	02	02
2	Type	01	01
3	Index	00	00
4	Data 1 (Low Byte)	00	→ Tabelle 75
5	Data 2 (High Byte)	00	→ Tabelle 75
6 – 7	Data 3 – 4	00	00

Tabelle 75: Byte 4 bis 5: Data 1 bis 2

Data 1	Bit	7	6	5	4	3	2	1	0
I1									0/1
I2								0/1	
…					…				
I8		0/1							
Data 2	**Bit**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
I9									0/1
I10								0/1	
…					…				
I16		0/1							

Beispiel: Lokale Eingänge IW0 lesen

Byte	Bedeutung	Wert (hex), sendet	
		Master	Slave
0	Befehl: Lesen	91	–
	Antwort: Lesen erfolgreich	–	C2
1	Len	02	02
2	Type	01	01
3	Index	00	00
4	Data 1	00	C4
5	Data 2	00	02
6	Data 3	00	00
7	Data 4	00	00

→ Alle Werte müssen hexadezimal übergeben werden.

Durch die Werte Data 1 = C4 und Data 2 = 02 wird angezeigt, dass die Eingänge I8, I7, I3 und I10 mit Zustand „1" beschaltet sind.

Eingänge der Netzwerkteilnehmer: IW1 – IW8

Die easy800 und MFD- Geräte können Sie mit Hilfe des easyNet sehr einfach dezentral erweitern. Mit dem hier angebotenen Dienst ist der lesende Zugriff auf die Eingänge der einzelnen NET-Teilnehmer realisierbar.

Telegrammaufbau

Byte	Bedeutung	Wert (hex), sendet	
		Master	Slave
0	Befehl: Lesen	91	–
	Antwort:		
	Lesen erfolgreich	–	C2
	Befehl zurückgewiesen	–	C0
1	Len	02	02
2	Type	01	01
3	Index	01 – 08[1)]	01 – 08[1)]
4	Data 1 (Low Byte)	00	→ Tabelle 75 auf Seite 187.
5	Data 2 (High Byte)	00	
6 – 7	Data 3 – 4	00	00

1) enspricht Adresse des Netzteilnehmers

Merker: M…

Byte	Bedeutung	Wert (hex), sendet	
		Master	Slave
0	Befehl		
	Lesen	91	–
	Schreiben	B1	–
	Antwort		
	Lesen erfolgreich	–	C2
	Schreiben erfolgreich	–	C1
	Befehl zurückgewiesen	–	C0
1	Len	→ Tabelle 76	→ Tabelle 76
2	Type		
3	Index		
4 – 7	Data 1 – 4		
	Beim Lesen	00	→ „Beispiel 1: Merker-Bit setzen/rücksetzen" auf Seite 192
	Beim Schreiben	→ „Beispiel 2: Merkerwort schreiben" auf Seite 192	00

Tabelle 76: Byte 1 bis 3: Len, Type, Index

Operand		Len	Type	Index
Merker-Bit	M1 … M96	01_{hex}	$0B_{hex}$	01 bis 60_{hex}
Merker-Byte	MB1 … MB96	01_{hex}	$0C_{hex}$	01 bis 60_{hex}
Merker-Word	MW1 … MW96	02_{hex}	$0D_{hex}$	01 bis 60_{hex}
Merker-Doppelwort	MD1 … MD96	04_{hex}	$0E_{hex}$	01 bis 60_{hex}

Sehen Sie sich hierzu eventuell die Merkeraufteilung im easy800-Handbuch etwas detaillierter an. An dieser Stelle soll lediglich ein kleiner Auszug aus dem genannten Handbuch dargestellt werden, um die prinzipielle Aufteilung einmal kurz darzustellen.

Achtung!
Die Funktionsbausteine und DW-Merker (32Bit-Werte) der easy800/MFD arbeiten mit vorzeichenbehafteten Werten.

Gilt für MD, MW, MB, M	**Links = Größtwertiges Bit, Byte, Wort**			**Rechts = Kleinstwertiges Bit, Byte, Wort**
32 Bit	MD1			
16 Bit	MW2		MW1	
8 Bit	MB4	MB3	MB2	MB1
1 Bit	M32 bis M25	M24 bis M17	M16 bis M9	M8 bis M1
32 Bit	MD2			
16 Bit	MW4		MW3	
8 Bit	MB8	MB7	MB6	MB5
1 Bit	M64 bis M57	M56 bis M49	M48 bis M41	M40 bis M33

Die entsprechenden Merkerwerte werden im Intelformat übergeben; das erste Byte ist dementsprechend das Low Byte (Byte 4) und das letzte demzufolge das High Byte.

Beispiel 1: Merker-Bit setzen/rücksetzen
Das Merker-Bit 62 soll gesetzt bzw. zurückgesetzt werden. Zum Setzen des Merker-Bits schreiben Sie in das unterste Bit von Datenbyte „Data 1" eine „1"; beim Rücksetzen eine „0".

Byte	Bedeutung	Wert (hex), sendet	
		Master	Slave
0	Befehl: Schreiben	B1	–
	Antwort: Schreiben erfolgreich	–	C1
1	Len	01	01
2	Type	0B	0B
3	Index	3E	3E
4	Data 1	01/00[1)]	00
5 – 7	Data 2 – 4	00	00

1) 01 = setzen, 00 = rücksetzen

Beispiel 2: Merkerwort schreiben
In das Merkerwort MW32 soll der Wert 823 geschrieben werden: $823_{dec} = 337_{hex}$ → Data 1 = 37_{hex}, Data 2 = 03_{hex}

Byte	Bedeutung	Wert (hex), sendet	
		Master	Slave
0	Befehl: Schreiben	B1	–
	Antwort: Schreiben erfolgreich	–	C1
1	Len	01	01
2	Type	0D	0D
3	Index	20	20
4	Data 1	37	00
5	Data 2	03	00
6	Data 3	00	00
7	Data 4	00	00

Lokale P-Tasten: P1 – P4

Die lokalen P-Tasten sind die Display-Courser-Tasten des easy800-/MFD-Basisgerätes. Die Tasten können Sie sowohl im Modus RUN als auch im Modus STOP abfragen.

→ Achten Sie nur darauf, dass die P-Tasten auch unter dem Menüpunkt SYSTEM (im Basisgerät) aktiviert sind.

Bei den P-Tasten müssen Sie nur ein Byte übergeben.

Byte	Bedeutung	Wert (hex), sendet	
		Master	Slave
0	Befehl: Lesen	91	–
	Antwort:		
	Lesen erfolgreich	–	C2
	Befehl zurückgewiesen	–	C0
1	Len	02	02
2	Type	06	06
3	Index	00	00
4	Data 1 (Low Byte)	00	→ Tabelle 77
5 – 7	Data 2 – 4	00	00

Tabelle 77: Byte 4: Data

Data 1	Bit 7	6	5	4	3	2	1	0
P1								0/1
P2							0/1	
P3						0/1		
P4					0/1			
–				0				
–			0					
–		0						
–	0							

Lokaler Analog-Ausgang: QA1

Auf den lokalen Analogausgang des Basisgerätes easy800 oder MFD können Sie mit Hilfe des angebotenen Kommandos zugreifen. Beim schreibenden Zugriff auf den Analogausgang (erst ab easy800, Version 04, möglich) wird der Wert allerdings nur nach außen gegeben, wenn das entsprechende Gerät im RUN-Modus ist und das entsprechende Abbild nicht durch das eigentliche Programm beschrieben wird. → Abschnitt „Abbilddaten lesen/schreiben" auf Seite 182.

Byte	Bedeutung	Wert (hex), sendet	
		Master	Slave
0	Befehl		
	Lesen	91	–
	Schreiben[1)]	B1	–
	Antwort		
	Lesen erfolgreich	–	C2
	Schreiben erfolgreich	–	C1
	Befehl zurückgewiesen	–	C0
1	Len	02	02
2	Type	05	05
3	Index	00	00
4 – 5	Data 1 – 2		
	Beim Lesen	00	→ Beispiel
	Beim Schreiben	→ Beispiel	00
6 – 7	Data 3 – 4	00	00

1) Schreiben ist erst ab easy800, Version 04, möglich, → Abschnitt „Versionsgeschichte" auf Seite 176.

Beispiel:
Der Analogausgang soll einen Wert von ca. 5 V ausgeben.

$500 = 01F4_{hex}$ Byte 4 – Data 1 (Low Byte) : $F4_{hex}$
Byte 5 – Data 2 (High Byte): 01_{hex}

Lokale Ausgänge: QW0/ Ausgänge der Netzwerkteilnehmer: QW1 – QW8

Die lokalen Ausgänge können Sie über den CANopen direkt lesen und ab easy800, Version 04, auch beschreiben. Die Ausgänge werden jedoch nur nach außen durchgeschaltet, wenn sich das Gerät im RUN-Modus befindet und der angesprochene Ausgang nicht im Schaltplan verwendet wird.
→ Abschnitt „Abbilddaten lesen/schreiben“ auf Seite 182.

Telegrammaufbau

Byte	Bedeutung	Wert (hex), sendet	
		Master	Slave
0	Befehl		
	Lesen	91	–
	Schreiben[1)]	B1	–
	Antwort		
	Lesen erfolgreich	–	C2
	Schreiben erfolgreich	–	C1
	Befehl zurückgewiesen	–	C0
1	Len	02	02
2	Type	04	04
3	Index[2)]	00/01 – 08	00/01 – 08
4	Data 1		
	Beim Lesen	00	→ Tabelle 74
	Beim Schreiben	→ Tabelle 78	00
5 – 7	Data 2 – 4	00	00

1) Schreiben ist erst ab easy800, Version 04 möglich,
→ Abschnitt „Versionsgeschichte“ auf Seite 176.

2) 00 = Lokaler Ausgang
01 – 08 = Ausgänge der Netzwerkteilnehmer 1 – 8

Tabelle 78: Byte 4: Data

Data 1	Bit 7	6	5	4	3	2	1	0
Q1								0/1
Q2							0/1	
Q3						0/1		
Q4					0/1			
Q5				0				
Q6			0					
Q7		0						
Q8	0							

Ein-/Ausgänge von easyLink: RW/SW

Mit diesem Dienst können Sie die lokalen R- und S-Daten und die der NET-Teilnehmer (1 – 8) , die über easyLink transferiert werden, nochmals aus dem entsprechenden Abbild der easy800/MFD lesen.

Byte	Bedeutung	Wert (hex), sendet	
		Master	Slave
0	Befehl: Lesen	91	–
	Antwort:		
	Lesen erfolgreich	–	C2
	Befehl zurückgewiesen	–	C0
1	Len	02	02
2	Type	bei RW: 07	bei RW: 07
		bei SW: 09	bei SW: 09
3	Index	00/01 – 08[1)]	00/01 – 08[1)]
4	Data 1 (Low Byte)	00	→ Tabelle 79
5	Data 2 (High Byte)	00	→ Tabelle 79
6 – 7	Data 3 – 4	00	00

1) 00 = Lokaler Ein-/Ausgang
01 – 08 = Adresse des Netzwerkteilnehmers (NET-ID 1 – 8)

Tabelle 79: Byte 4 bis 5: Data 1 bis 2

Data 1		**Bit**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RW	**SW**									
R1	S1									0/1
R2	S2								0/1	
R3	S3							0/1		
R4	S4						0/1			
R5	S5					0/1				
R6	S6				0/1					
R7	S7			0/1						
R8	S8		0/1							
Data 2		**Bit**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
R9	–									0/1
R10	–								0/1	
R11	–							0/1		
R12	–						0/1			
R13	–					0/1				
R14	–				0/1					
R15	–			0/1						
R16	–		0/1							

Receive-Data Netz: RN1 – RN32/ Send-Data Netz: SN1 – SN32

Das easyNet bietet die Möglichkeit, eine Punkt-zu-Punkt-Verbindung zwischen den einzelnen NET-Teilnehmern zu realisieren. Der Datenaustausch erfolgt hierbei mittels RN-SN-Daten (siehe easy800-Handbuch).

→ Es können keine RN-SN-Daten des lokalen Gerätes (Index = 0) , an dem sich das EASY204-DP befindet, abgefragt werden. In diesem Fall würde der Befehl mit $0C_{hex}$ zurückgewiesen.

Byte	Bedeutung	Wert (hex), sendet	
		Master	**Slave**
0	Befehl: Lesen	91	–
	Antwort:		
	Lesen erfolgreich	–	C2
	Befehl zurückgewiesen	–	C0
1	Len	04	04
2	Type	bei RN1 – RN32: 08	
		bei SN1 – SN32: 0A	
3	Index	01 – 08[1)]	01 – 08[1)]
4 – 7	Data 1 – 4	00	→ Tabelle 80

1) entspricht NET-ID

Tabelle 80: Byte 4 bis 7: Data 1 bis 4

Data 1	**Bit**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RN1 SN1					…				0/1
…								0/1	
RN8 SN8		0/1							
Data 2	**Bit**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RN9 SN9									0/1
…					…				
RN16 SN16		0/1							
Data 3	**Bit**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RN17 SN17									0/1
…					…				
RN24 SN24		0/1							
Data 4	**Bit**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RN25 SN25									0/1
…					…				
RN32 SN32		0/1							

Funktionsbaustein-Daten lesen/schreiben

→ Beachten Sie bitte auch die entsprechende Beschreibung zu den Funktionsbausteinben, die im easy800-Handbuch angeführt sind.

Allgemeine Hinweise

Grundsätzlich müssen Sie beim Arbeiten mit den Funktionsbausteinen Folgendes beachten:

- Die entsprechenden Daten werden im Intelformat übergeben: das erste Byte ist das Low Byte (Byte 4) und das letzte das High Byte (Byte 7).
- Die Datenlänge beträgt bis zu 4 Byte. Alle Werte müssen Sie im hexadezimalen Zahlenformat übergeben.
- Alle 32-Bit-Werte werden vorzeichenbehaftet behandelt. Wenn Sie 32-Bit-Werte übergeben, achten Sie darauf, dass der entsprechende Wertebereich dem long integer entspricht, also vorzeichenbehaftet ist.
  32-Bit-Wert: –2147483648…0…+2147483647

Übersicht

Operanden	Bedeutung	Lesen/ schreiben	Type (hex)	Seite
A01 – A32	„Analogwertvergleicher: A01 – A32"	lesen/schreiben	11	204
AR01 – AR32	„Arithmetikbaustein: AR01 – AR32"	lesen/schreiben	12	206
BC01 – BC32	„Block Compare: BC01 – BC32"	lesen/schreiben	25	208
BT01 – BT32	„Block Transfer: BT01 – BT32"	lesen/schreiben	26	210
BV01 – BV32	„Boolesche Verknüpfung: BV01 – BV32"	lesen/schreiben	13	212
C01 – C32	„Zähler: C01 – C32"	lesen/schreiben	14	214
CF01 – CF04	„Frequenzzähler: CF01 – CF04"	lesen/schreiben	15	216
CH01 – CH04	„Schneller Zähler: CH01 – CH04"	lesen/schreiben	16	218
CI01 – CI02	„Inkrementalzähler: CI01 – CI02"	lesen/schreiben	17	220

Operanden	Bedeutung	Lesen/ schreiben	Type (hex)	Seite
CP01 – CP32	„Vergleicher: CP01 – CP32"	lesen/schreiben	18	222
D01 – D32	„Textausgabe-Baustein: D01 – D32"	lesen/schreiben	19	224
DB01 – DB32	„Datenbaustein: DB01 – DB32"	lesen/schreiben	1A	227
DC01 – DC32	„PID-Regler: DC01 – DC32"	lesen/schreiben	27	229
DG01 – DG16	„Diagnose DG01…DG16"	lesen	39	232
FT01 – FT32	„Signalglättungsfilter: FT01 – FT32"	lesen/schreiben	28	234
GT01 – GT32	„Empfang von Netzdaten: GT01 – GT32"	lesen	1B	236
HW01 – HW32	„Wochenzeitschaltuhr: HW01 – HW32"	lesen	1C	238
HY01 – HY32	„Jahreszeitschaltuhr: HY01 – HY32"	lesen	1D	241
JC01 – JC32	„Bedingter Sprung JC01…JC32"	lesen	2F	244
LS01 – LS32	„Wertskalierung: LS01 – LS32"	lesen/schreiben	29	246
MR01 – MR32	„Masterreset: MR01 – MR32"	lesen	0F	248
MX01 – MX32	„Datenmultiplexer MX01…MX32"	lesen/schreiben	31	250
NC01 – NC32	„Zahlenwandler: NC01 – NC32"	lesen/schreiben	2A	252
OT01 – OT04	„Betriebsstundenzähler: OT01 – OT04"	lesen/schreiben	1E	254
PO01 – PO02	„Impulsausgabe PO01…PO02"	lesen/schreiben	32	256
PT01 – PT32	„Senden von Netzdaten: PT01 – PT32"	lesen	1F	259
PW01 – PW02	„Pulsweitenmodulation: PW01 – PW02"	lesen/schreiben	2B	261
SC01	„Uhr synchronisieren: SC01"	lesen	20	263
SP01 - SP32	„Serielle Ausgabe SP01…SP32"	lesen	35	264
SR01 - SR32	„Schieberegister SR01…SR32"	lesen/schreiben	33	266
ST01	„Sollzykluszeit: ST01"	lesen/schreiben	2C	269
T01 – T32	„Zeitrelais: T01 – T32"	lesen/schreiben	21	271
TB01 – TB32	„Tabellenfunktion TB01…TB32"	lesen/schreiben	34	274
VC01 – VC32	„Wertbegrenzung: VC01 – VC32"	lesen/schreiben	2D	276

Analogwertvergleicher: A01 – A32

Telegrammaufbau

Byte	Bedeutung	Wert (hex), sendet	
		Master	Slave
0	Befehl:		
	Lesen	92	–
	Schreiben	B2	–
	Antwort:		
	Lesen erfolgreich	–	C2
	Schreiben erfolgreich	–	C1
	Befehl zurückgewiesen	–	C0
1	Type	11	11
2	Instanz	01 – 20	01 – 20
3	Index	→ Tabelle 81	→ Tabelle 81
4 – 7	Data 1 – 4	00	abhängig vom Index, → Tabelle 82, 83

Tabelle 81: Operandenübersicht

Index (hex)	Operand		lesen	schreiben
00	Bit-IO, → Tabelle 82		×	
01	Mode, → Tabelle 83		×	
02	Vergleichswert 1	I1	×	c[1)]
03	Verstärkungsfaktor für I1 (I1 = F1 × Wert)	F1	×	c[1)]
04	Vergleichswert 2	I2	×	c[1)]
05	Verstärkungsfaktor für I2 (I2 = F2 × Wert)	F2	×	c[1)]
06	Offset für den Wert I1	OS	×	c[1)]
07	Schalthysterese für Wert I2 (Wert HY gilt sowohl für die positive als auch negative Hysterese.)	HY	×	c[1)]

1) Der Wert kann nur geschrieben werden, wenn er im Programm mit einer Konstanten beschaltet ist.

→ Die Daten für Index 2 bis 7 werden als 32-Bit-Wert im Intelformat (Data 1 – Low Byte bis Data 4 – High Byte) übergeben.

Tabelle 82: Index 0 – Bit-IO

	Bit	7	6	5	4	3	2	1	0
FB-Ausgang Data 3		–	–	–	–	–	–	CY[1)]	Q1[2)]

1) Zustand „1", wenn der Wertebereich überschritten wird
2) Zustand „1", wenn die Bedingung erfüllt ist (z. B. I1 < I2 bei der Betriebsart LT)

Tabelle 83: Index 1 – Mode

Data 1 (hex)		
00	LT	kleiner (I1 < I2)
01	EQ	gleich (I1 = I2)
02	GT	größer (I1 > I2)

Arithmetikbaustein: AR01 – AR32

Telegrammaufbau

Byte	Bedeutung	Wert (hex), sendet	
		Master	Slave
0	Befehl:		
	Lesen	92	–
	Schreiben	B2	–
	Antwort:		
	Lesen erfolgreich	–	C2
	Schreiben erfolgreich	–	C1
	Befehl zurückgewiesen	–	C0
1	Type	12	12
2	Instanz	01 – 20	01 – 20
3	Index	→ Tabelle 84	→ Tabelle 84
4 – 7	Data 1 – 4		
	beim Lesen	00	abhängig vom Index, → Tabelle 85, 86
	beim Schreiben	abhängig vom Index, → Tabelle 85, 86	00

Tabelle 84: Operandenübersicht

Index (hex)	Operand		lesen	schreiben
00	Bit-IO, → Tabelle 85		×	
01	Mode, → Tabelle 86		×	
02	erster Operand	I1	×	c[1)]
03	zweiter Operand	I2	×	c[1)]
04	Ergebnis	QV	×	

1) Der Wert kann nur geschrieben werden, wenn er im Programm mit einer Konstanten beschaltet ist.

→ Die Daten für Index 2 bis 4 werden als 32-Bit-Wert im Intelformat (Data 1 – Low Byte bis Data 4 – High Byte) übergeben.

Tabelle 85: Index 0 – Bit-IO

	Bit	7	6	5	4	3	2	1
FB-Ausgang Data 3		–	–	–	–	–	ZE[1)]	CY[2)]

1) Zustand „1", wenn der Wert des Bausteinausgangs QV (also das Rechenergebnis) gleich Null ist
2) Zustand „1", wenn der Wertebereich überschritten wird

Tabelle 86: Index 1 – Mode

Data 1 (hex)		
00	ADD	Addieren (I1 + I2 = QV)
01	SUB	Subtrahieren (I1 – I2 = QV)
02	MUL	Multiplizieren (I1 × I2 = QV)
03	DIV	Dividieren (I1 : I2 = QV)

Block Compare: BC01 – BC32

Telegrammaufbau

Byte	Bedeutung	Wert (hex), sendet	
		Master	Slave
0	Befehl:		
	Lesen	92	–
	Schreiben	B2	–
	Antwort:		
	Lesen erfolgreich	–	C2
	Schreiben erfolgreich	–	C1
	Befehl zurückgewiesen	–	C0
1	Type	25	25
2	Instanz	01 – 20	01 – 20
3	Index	→ Tabelle 87	→ Tabelle 87
4 – 7	Data 1 – 4		
	beim Lesen	00	abhängig vom Index, → Tabelle 88, 89
	beim Schreiben	abhängig vom Index, → Tabelle 88, 89	00

Tabelle 87: Operandenübersicht

Index (hex)	Operand		lesen	schreiben
00	Bit-IO, → Tabelle 88		×	
01	Mode, → Tabelle 89		×	
02	Quellbereich 1	I1	×	c[1)]
03	Zielbereich 2	I2	×	c[1)]
04	Anzahl der zu vergleichenden Elemente: 8 (max. 192 Byte)	NO	×	c[1)]

1) Der Wert kann nur geschrieben werden, wenn er im Programm mit einer Konstanten beschaltet ist.

→ Die Daten für Index 2 bis 4 werden als 32-Bit-Wert im Intelformat (Data 1 – Low Byte bis Data 4 – High Byte) übergeben.

Tabelle 88: Index 0 – Bit-IO

	Bit	7	6	5	4	3	2	1	0
FB-Eingang Data 1		–	–	–	–	–	–	–	EN[1)]
FB-Ausgang Data 3		–	–	–	–	EQ[2)]	E3[3)]	E2[4)]	E1[5)]

1) Aktiviert den Baustein bei Zustand „1".

2) Zustand „1", wenn die Datenbereiche gleich sind; Zustand „0", wenn sie ungleich sind

Fehlerausgänge

3) Zustand „1", wenn die Anzahl der Elemente den Quell- oder Zielbereich überschreitet.

4) Zustand „1", wenn sich Quell- und Zielbereich überlappen.

5) Zustand „1", wenn Quell- oder Zielbereich außerhalb des verfügbaren Merkerbereichs liegen (Offsetfehler).

Tabelle 89: Index 1 – Mode

Mode	Data 1 (hex)	Betriebsart
	02	Vergleichen (easy intern Zustandsanzeige Modus Block Compare)

Block Transfer: BT01 – BT32

Telegrammaufbau

Byte	Bedeutung	Wert (hex), sendet	
		Master	Slave
0	Befehl:		
	Lesen	92	–
	Schreiben	B2	–
	Antwort:		
	Lesen erfolgreich	–	C2
	Schreiben erfolgreich	–	C1
	Befehl zurückgewiesen	–	C0
1	Type	26	26
2	Instanz	01 – 20	01 – 20
3	Index	→ Tabelle 90	→ Tabelle 90
4 – 7	Data 1 – 4		
	beim Lesen	00	abhängig vom Index, → Tabelle 91, 92
	beim Schreiben	abhängig vom Index, → Tabelle 91, 92	00

Tabelle 90: Operandenübersicht

Index (hex)	Operand		lesen	schreiben
00	Bit-IO, → Tabelle 91		×	
01	Mode, → Tabelle 92		×	
02	Quellbereich 1	I1	×	c[1)]
03	Zielbereich 2	I2	×	c[1)]
04	Anzahl der zu vergleichenden Elemente: max. 192 Byte	NO	×	c[1)]

1) Der Wert kann nur geschrieben werden, wenn er im Programm mit einer Konstanten beschaltet ist.

→ Die Daten für Index 2 und 3 werden als 32-Bit-Wert im Intelformat (Data 1 – Low Byte…Data 2 – High Byte) übergeben.

Tabelle 91: Index 0 – Bit-IO

	Bit	7	6	5	4	3	2	1	0
FB-Eingang Data 1		–	–	–	–	–	–	–	T[1)]
FB-Ausgang Data 3		–	–	–	–	–	E3[2)]	E2[3)]	E1[4)]

1) Transfer von der an I1 angegeben Quelladresse zu der an I2 angegebenen Zieladresse bei positiver Flanke.

Fehlerausgänge

2) Zustand „1", wenn die Anzahl der Elemente den Quell- oder Zielbereich überschreitet.
3) Zustand „1", wenn sich Quell- und Zielbereich überlappen.
4) Zustand „1", wenn Quell- oder Zielbereich außerhalb des verfügbaren Merkerbereichs liegen (Offsetfehler).

Tabelle 92: Index 1 – Mode

Data 1 (hex)	Betriebsart
00	INI: Initialisiert den Zielbereich mit einem Bytewert, der unter der Quelladrtesse hinterlegt ist.
01	CPY: Kopiert einen Datenblock von einem Quell- zu einem Zielbereich. Über „NO" geben Sie die Datenblockgröße vor.

Boolesche Verknüpfung: BV01 – BV32

Telegrammaufbau

Byte	Bedeutung	Wert (hex), sendet	
		Master	Slave
0	Befehl:		
	Lesen	92	–
	Schreiben	B2	–
	Antwort:		
	Lesen erfolgreich	–	C2
	Schreiben erfolgreich	–	C1
	Befehl zurückgewiesen	–	C0
1	Type	13	13
2	Instanz	01 – 20	01 – 20
3	Index	→ Tabelle 93	→ Tabelle 93
4 – 7	Data 1 – 4		
	beim Lesen	00	abhängig vom Index, → Tabelle 94, 95
	beim Schreiben	abhängig vom Index, → Tabelle 94, 95	00

Tabelle 93: Operandenübersicht

Index (hex)	Operand		lesen	schreiben
00	Bit-IO, → Tabelle 94		×	
01	Mode, → Tabelle 95		×	
02	erster Operand	I1	×	c[1)]
03	zweiter Operand	I2	×	c[1)]
04	Ergebnis der Verknüpfung	QV	×	

1) Der Wert kann nur geschrieben werden, wenn er im Programm mit einer Konstanten beschaltet ist.

→ Die Daten für Index 2 bis 4 werden als 32-Bit-Wert im Intelformat (Data 1 – Low Byte bis Data 4 – High Byte) übergeben.

Tabelle 94: Index 0 – Bit-IO

	Bit	7	6	5	4	3	2	1
FB-Ausgang Data 3		–	–	–	–	–	–	ZE[1)]

1) Zustand „1", wenn der Wert des Bausteinausgangs QV (also das Ergebnis der Verknüpfung) gleich Null ist

Tabelle 95: Index 1 – Mode

Data 1 (hex)		
00	AND	Und-Verknüpfung
01	OR	Oder-Verknüpfung
02	XOR	Exklusiv-Oder-Verknüpfung
03	NET	Invertiert die einzelnen Bit des Wertes von I1. Der invertierte Wert wird vorzeichenbehaftet dezimal dargestellt.

Zähler: C01 – C32

Telegrammaufbau

Byte	Bedeutung	Wert (hex), sendet	
		Master	Slave
0	Befehl:		
	Lesen	92	–
	Schreiben	B2	–
	Antwort:		
	Lesen erfolgreich	–	C2
	Schreiben erfolgreich	–	C1
	Befehl zurückgewiesen	–	C0
1	Type	14	14
2	Instanz	01 – 20	01 – 20
3	Index	→ Tabelle 96	→ Tabelle 96
4 – 7	Data 1 – 4		
	beim Lesen	00	abhängig vom Index, → Tabelle 97
	beim Schreiben	abhängig vom Index, → Tabelle 97	00

Tabelle 96: Operandenübersicht

Index (hex)	Operand		Wert	lesen	schreiben
00	Bit-IO		→ Tabelle 97	×	
01	Mode/Parameter		–	–	–
02	oberer Sollwert	SH	im ganzzahligen Bereich von –2147483648 bis +2147483647	×	c[1]
03	unterer Sollwert	SL		×	c[1]
04	Vorgabe Istwert	SV		×	c[1]
05	Istwert im RUN-Betrieb	QV		×	

1) Der Wert kann nur geschrieben werden, wenn er im Programm mit einer Konstanten beschaltet ist.

→ Die Daten für Index 2 bis 5 werden als 32-Bit-Wert im Intelformat (Data 1 – Low Byte bis Data 4 – High Byte) übergeben.

Tabelle 97: Index 0 – Bit-IO

	Bit	7	6	5	4	3	2	1	0
FB-Eingang Data 1		–	–	–	–	SE[1]	D[2]	C[3]	RE[4]
FB-Ausgang Data 3		–	–	–	–	ZE[5]	CY[6]	FB[7]	OF[8]

1) bei positiver Flanke Vorgabe-Istwert übernehmen
2) Zählrichtungsangabe: Zustand „0" = vorwärts zählen, Zustand „1" = rückwärts zählen
3) Zählspule, zählt bei jeder positiven Flanke
4) Rücksetzen des Istwertes auf Null
5) Zero: Zustand „1", wenn der Wert des Bausteinausganges QV (also der Zählerstand) gleich Null ist
6) Carry: Zustand „1", wenn der Wertebereich überschritten wird.
7) Fall below: Zustand „1", wenn Istwert ≦ unterer Sollwert
8) Overflow: Zustand „1", wenn Istwert ≧ oberer Sollwert

Frequenzzähler: CF01 – CF04

Telegrammaufbau

Byte	Bedeutung	Wert (hex), sendet	
		Master	**Slave**
0	Befehl:		
	Lesen	92	–
	Schreiben	B2	–
	Antwort:		
	Lesen erfolgreich	–	C2
	Schreiben erfolgreich	–	C1
	Befehl zurückgewiesen	–	C0
1	Type	15	15
2	Instanz	01 – 04	01 – 04
3	Index	→ Tabelle 98	→ Tabelle 98
4 – 7	Data 1 – 4		
	beim Lesen	00	abhängig vom Index, → Tabelle 99
	beim Schreiben	abhängig vom Index, → Tabelle 99	00

Tabelle 98: Operandenübersicht

Index (hex)	Operand		lesen	schreiben
00	Bit-IO, → Tabelle 99		×	
01	Mode/Parameter		–	–
02	oberer Sollwert	SH	×	c[1)]
03	unterer Sollwert	SL	×	c[1)]
04	Istwert im RUN-Betrieb	QV	×	

1) Der Wert kann nur geschrieben werden, wenn er im Programm mit einer Konstanten beschaltet ist.

→ Die Daten für Index 2 bis 4 werden als 32-Bit-Wert im Intelformat (Data 1 – Low Byte bis Data 4 – High Byte) übergeben.

Tabelle 99: Index 0 – Bit-IO

	Bit	7	6	5	4	3	2	1	0
FB-Eingang Data 1		–	–	–	–	–	–	–	EN[1)]
FB-Ausgang Data 3		–	–	–	–	–	ZE[2)]	FB[3)]	OF[4)]

1) Freigabe des Zählers
2) Zero: Zustand „1“, wenn der Wert des Bausteinausganges QV (also der Zählerstand) gleich Null ist
3) Fall below: Zustand „1“, wenn Istwert ≦ unterer Sollwert
4) Overflow: Zustand „1“, wenn Istwert ≧ oberer Sollwert.

Schneller Zähler: CH01 – CH04

Telegrammaufbau

Byte	Bedeutung	Wert (hex), sendet	
		Master	**Slave**
0	Befehl:		
	Lesen	92	–
	Schreiben	B2	–
	Antwort:		
	Lesen erfolgreich	–	C2
	Schreiben erfolgreich	–	C1
	Befehl zurückgewiesen	–	C0
1	Type	16	16
2	Instanz	01 – 04	01 – 04
3	Index	→ Tabelle 100	→ Tabelle 100
4 – 7	Data 1 – 4		
	beim Lesen	00	abhängig vom Index, → Tabelle
	beim Schreiben	abhängig vom Index, → Tabelle	00

Tabelle 100: Operandenübersicht

Index (hex)	Operand		Wert	lesen	schreiben
00	Bit-IO		→ Tabelle 101	×	
01	Mode/Parameter		–	–	–
02	oberer Sollwert	SH	im ganzzahligen Bereich von –2147483648 bis +2147483647	×	c[1]
03	unterer Sollwert	SL		×	c[1]
04	Vorgabe Istwert	SV		×	c[1]
05	Istwert im RUN-Betrieb	QV		×	

1) Der Wert kann nur geschrieben werden, wenn er im Programm mit einer Konstanten beschaltet ist.

→ Die Daten für Index 2 bis 5 werden als 32-Bit-Wert im Intelformat (Data 1 – Low Byte bis Data 4 – High Byte) übergeben.

Tabelle 101: Index 0 – Bit-IO

	Bit	7	6	5	4	3	2	1	0
FB-Eingang Data 1		–	–	–	–	EN[1]	SE[2]	D[3]	RE[4]
FB-Ausgang Data 3		–	–	–	–	ZE[5]	CY[6]	FB[7]	OF[8]

1) Freigabe des Zählers
2) bei positiver Flanke Vorgabe-Istwert übernehmen
3) Zählrichtungsangabe: Zustand „0" = vorwärts zählen, Zustand „1" = rückwärts zählen
4) Rücksetzen des Istwertes auf Null
5) Zero: Zustand „1", wenn der Wert des Bausteinausganges QV (also der Zählerstand) gleich Null ist
6) Carry: Zustand „1", wenn der Wertebereich überschritten wird.
7) Fall below: Zustand „1", wenn Istwert ≦ unterer Sollwert
8) Overflow: Zustand „1", wenn Istwert ≧ unterer Sollwert

Inkrementalzähler: CI01 – CI02

Telegrammaufbau

Byte	Bedeutung	Wert (hex), sendet	
		Master	Slave
0	Befehl:		
	Lesen	92	–
	Schreiben	B2	–
	Antwort:		
	Lesen erfolgreich	–	C2
	Schreiben erfolgreich	–	C1
	Befehl zurückgewiesen	–	C0
1	Type	17	17
2	Instanz	01 – 02	01 – 02
3	Index	→ Tabelle 102	→ Tabelle 102
4 – 7	Data 1 – 4		
	beim Lesen	00	abhängig vom Index, → Tabelle 103
	beim Schreiben	abhängig vom Index, → Tabelle 103	00

Tabelle 102: Operandenübersicht

Index (hex)	Operand		Wert	lesen	schreiben
00	Bit-IO		→ Tabelle 103	×	
01	Mode/Parameter		–	–	–
02	oberer Sollwert	SH	im ganzzahligen Bereich von –2147483648 bis +2147483647	×	c[1]
03	unterer Sollwert	SL		×	c[1]
04	Vorgabe Istwert	SV		×	c[1]
05	Istwert im RUN-Betrieb	QV		×	

1) Der Wert kann nur geschrieben werden, wenn er im Programm mit einer Konstanten beschaltet ist.

→ Die Daten für Index 2 bis 5 werden als 32-Bit-Wert im Intelformat (Data 1 – Low Byte bis Data 4 – High Byte) übergeben.

Tabelle 103: Index 0 – Bit-IO

	Bit	7	6	5	4	3	2	1	0
FB-Eingang Data 1		–	–	–	–	–	EN[1]	SE[2]	RE[3]
FB-Ausgang Data 3		–	–	–	–	ZE[4]	CY[5]	FB[6]	OF[7]

1) Freigabe des Zählers
2) Bei positiver Flanke Vorgabe-Istwert übernehmen
3) Rücksetzen des Istwertes auf Null
4) Zero: Zustand „1", wenn der Wert des Bausteinausganges QV (also der Zählerstand) gleich Null ist
5) Carry: Zustand „1", wenn der Wertebereich überschritten wird.
6) Fall below: Zustand „1", wenn Istwert ≦ unterer Sollwert
7) Overflow: Zustand „1", wenn Istwert ≧ unterer Sollwert

Vergleicher: CP01 – CP32

Telegrammaufbau

Byte	Bedeutung	Wert (hex), sendet	
		Master	Slave
0	Befehl:		
	Lesen	92	–
	Schreiben	B2	–
	Antwort:		
	Lesen erfolgreich	–	C2
	Schreiben erfolgreich	–	C1
	Befehl zurückgewiesen	–	C0
1	Type	18	18
2	Instanz	01 – 20	01 – 20
3	Index	→ Tabelle 104	→ Tabelle 104
4 – 7	Data 1 – 4		
	beim Lesen	00	abhängig vom Index, → Tabelle 105
	beim Schreiben	abhängig vom Index, → Tabelle 105	00

Tabelle 104: Operandenübersicht

Index (hex)	Operand		lesen	schreiben
00	Bit-IO, → Tabelle 105		×	
01	Mode/Parameter		–	–
02	Vergleichswert	I1	×	c[1)]
03	Vergleichswert	I2	×	c[1)]

1) Der Wert kann nur geschrieben werden, wenn er im Programm mit einer Konstanten beschaltet ist.

→ Die Daten für Index 2 und 3 werden als 32-Bit-Wert im Intelformat (Data 1 – Low Byte bis Data 4 – High Byte) übergeben.

Tabelle 105: Index 0 – Bit-IO

	Bit	7	6	5	4	3	2	1
FB-Ausgang Data 3		–	–	–	–	GT[1)]	EQ[2)]	LT[3)]

1) greater than: Zustand „1", wenn der Wert an I1 größer als der Wert an I2 ist (I1 > I2)

2) equal: Zustand „1", wenn der Wert an I1 gleich dem Wert an I2 ist (I1 = I2)

3) less than: Zustand „1", wenn der Wert an I1 kleiner als der Wert an I2 ist (I1 < I2).

Textausgabe-Baustein: D01 – D32

Telegrammaufbau

Byte	Bedeutung	Wert (hex), sendet	
		Master	Slave
0	Befehl:		
	Lesen	92	–
	Schreiben	B2	–
	Antwort:		
	Lesen erfolgreich	–	C2
	Schreiben erfolgreich	–	C1
	Befehl zurückgewiesen	–	C0
1	Type	19	19
2	Instanz	01 – 20	01 – 20
3	Index	→ Tabelle 106	→ Tabelle 106
4 – 7	Data 1 – 4		
	beim Lesen	00	abhängig vom Index, → Tabelle 107
	beim Schreiben	abhängig vom Index, → Tabelle 107	00

Tabelle 106: Operandenübersicht

Index (hex)	Operand	lesen	schreiben
00	Bit-IO, → Tabelle 107	×	
01	Mode/Parameter	–	–
02	Text Zeile 1, Spalte 1 – 4	×	
03	Text Zeile 1, Spalte 5 – 8	×	
04	Text Zeile 1, Spalte 9 – 12	×	
05	Text Zeile 1, Spalte 13 – 16	×	
06	Text Zeile 2, Spalte 1 – 4	×	
07	Text Zeile 2, Spalte 5 – 8	×	
08	Text Zeile 2, Spalte 9 – 12	×	
09	Text Zeile 2, Spalte 13 – 16	×	
10	Text Zeile 3, Spalte 1 – 4	×	
11	Text Zeile 3, Spalte 5 – 8	×	
12	Text Zeile 3, Spalte 9 – 12	×	
13	Text Zeile 3, Spalte 13 – 16	×	
14	Text Zeile 4, Spalte 1 – 4	×	
15	Text Zeile 4, Spalte 5 – 8	×	
16	Text Zeile 4, Spalte 9 – 12	×	
17	Text Zeile 4, Spalte 13 – 16	×	
18	Variable 1	×	c[1)]
19	Variable 2	×	c[1)]
20	Variable 3	×	c[1)]
21	Variable 4	×	c[1)]
22	Skalierung Minimalwert 1	×	
23	Skalierung Minimalwert 2	×	
24	Skalierung Minimalwert 3	×	
25	Skalierung Minimalwert 4	×	
26	Skalierung Maximalwert 1	×	

Index (hex)	Operand	lesen	schreiben
27	Skalierung Maximalwert 2	×	
28	Skalierung Maximalwert 3	×	
29	Skalierung Maximalwert 4	×	
30	Steuerinformationen Zeile 1	×	
31	Steuerinformationen Zeile 2	×	
32	Steuerinformationen Zeile 3	×	
33	Steuerinformationen Zeile 4	×	

1) Der Wert kann nur geschrieben werden, wenn er im Programm mit einer Konstanten beschaltet ist.

→ Die Variablen 1 bis 4 (Index 18 bis 21) werden als 32-Bit-Wert im Intelformat (Data 1 – Low Byte bis Data 4 – High Byte) übergeben.

Tabelle 107: Index 0 – Bit-IO

	Bit	7	6	5	4	3	2	1	0
FB-Eingang Data 1		–	–	–	–	–	–	–	EN[1)]
FB-Ausgang Data 3		–	–	–	–	–	–	–	Q1[2)]

1) Freigabe des Textbausteins

2) Zustand „1", Textbaustein ist aktiv

Datenbaustein: DB01 – DB32

Telegrammaufbau

Byte	Bedeutung	Wert (hex), sendet	
		Master	**Slave**
0	Befehl:		
	Lesen	92	–
	Schreiben	B2	–
	Antwort:		
	Lesen erfolgreich	–	C2
	Schreiben erfolgreich	–	C1
	Befehl zurückgewiesen	–	C0
1	Type	1A	1A
2	Instanz	01 – 20	01 – 20
3	Index	→ Tabelle 108	→ Tabelle 108
4 – 7	Data 1 – 4		
	beim Lesen	00	abhängig vom Index, → Tabelle 109
	beim Schreiben	abhängig vom Index, → Tabelle 109	00

Tabelle 108: Operandenübersicht

Index (hex)	Operand		lesen	schreiben
00	Bit-IO, → Tabelle 109		×	
01	Mode/Parameter		–	–
02	Eingangswert: Wert, der beim Triggern des Bausteins an den Ausgang QV weiter-gegeben wird.	I1	×	c[1)]
03	Ausgangswert	QV	×	

1) Der Wert kann nur geschrieben werden, wenn er im Programm mit einer Konstanten beschaltet ist.

→ Die Daten für Index 2 und 3 werden als 32-Bit-Wert im Intelformat (Data 1 – Low Byte bis Data 4 – High Byte) übergeben.

Tabelle 109: Index 0 – Bit-IO

	Bit	7	6	5	4	3	2	1	0
FB-Eingang Data 1		–	–	–	–	–	–	–	T[1)]
FB-Ausgang Data 3		–	–	–	–	–	–	–	Q1[2)]

1) Übernahme des an I1 liegenden Wertes bei positiver Flanke.

2) Zustand „1", wenn das Triggersignal den Zustand „1" besitzt.

PID-Regler: DC01 – DC32

Telegrammaufbau

Byte	Bedeutung	Wert (hex), sendet	
		Master	**Slave**
0	Befehl:		
	Lesen	92	–
	Schreiben	B2	–
	Antwort:		
	Lesen erfolgreich	–	C2
	Schreiben erfolgreich	–	C1
	Befehl zurückgewiesen	–	C0
1	Type	27	27
2	Instanz	01 – 20	01 – 20
3	Index	→ Tabelle 110	→ Tabelle 110
4 – 7	Data 1 – 4		
	beim Lesen	00	abhängig vom Index, → Tabelle 111, 112
	beim Schreiben	abhängig vom Index, → Tabelle 111, 112	

Tabelle 110: Operandenübersicht

Index (hex)	Operand		lesen	schreiben
00	Bit-IO, → Tabelle 111		×	
01	Mode, → Tabelle 112		×	
02	Sollwert: –32768 bis +32767	I1	×	c[1)]
03	Istwert: –32768 bis +32767	I2	×	c[1)]
04	Proportionalverstärkung [%], Wertebereich: 0 bis 65535	KP	×	c[1)]
05	Nachstellzeit [0,1 s], Wertebereich: 0 bis 65535	TN	×	c[1)]
06	Vorhaltezeit [0,1 s], Wertebereich: 0 bis 65535	TV	×	c[1)]
07	Abtastzeit = Zeit zwischen den Bausteinaufrufen. Wertebereich: 0.1s bis 6553.5s. Wenn Sie den Wert 0 angeben, wird die Abtastzeit von der Programmzykluszeit bestimmt.	TC	×	c[1)]
08	Handstellgröße, Wertebereich: –4096 bis +4095	MV	×	c[1)]
09	Stellgröße • Betriebsart: UNI, Wertebereich: 0 bis +4095 (12 Bit) • Betriebsart: BIP, Wertebereich: –4096 bis +4095 (13 Bit)	QV	×	

1) Der Wert kann nur geschrieben werden, wenn er im Programm mit einer Konstanten beschaltet ist.

→ Die Daten für Index 2 bis 9 werden als 32-Bit-Wert im Intelformat (Data 1 – Low Byte…Data 2 – High Byte) übergeben.

Tabelle 111: Index 0 – Bit-IO

	Bit	7	6	5	4	3	2	1	0
FB-Eingang Data 1		–	–	–	SE[1)]	ED[2)]	EI[3)]	EP[4)]	EN[5)]
FB-Ausgang Data 3		–	–	–	–	–	–	–	LI[6)]

1) Übernahme der Handstellgröße bei Zustand „1"
2) Aktivieren des D-Teils bei Zustand „1"
3) Aktivieren des I-Teils bei Zustand „1"
4) Aktivieren des P-Teils bei Zustand „1"
5) Aktiviert den Baustein bei Zustand „1".
6) Zustand „1", wenn der Wertebereich der Stellgröße überschritten wurde

Tabelle 112: Index 1 – Mode

Data 1	Betriebsart
UNP unipolar	Die Stellgröße wird als unipolarer 12Bit-Wert ausgegeben. Entsprechender Wertebereich für QV 0 bis 4095.
BIP bipolar	Die Stellgröße wird als bipolarer 13Bit-Wert ausgegeben. Entsprechender Wertebereich für QV –4096 bis 4095

Diagnose DG01…DG16

Telegrammaufbau

Byte	Bedeutung	Wert (hex), sendet	
		Master	Slave
0	Befehl:		
	Lesen	92	–
	Schreiben	B2	–
	Antwort:		
	Lesen erfolgreich	–	C2
	Schreiben erfolgreich	–	C1
	Befehl zurückgewiesen	–	C0
1	Type	39	39
2	Instanz	01 - 10	01 - 10
3	Index	00 - 03	00 - 03
4 - 7	Data 1 - 4 beim Lesen	00	abhängig vom Index, → Tabelle 113, 114

Tabelle 113: Operandenübersicht

Index (hex)	Daten	Data 1, Data 3, Data 4	Data 2	Read/Write
0	Bit-IO	→ Tabelle 114	–	R
2	Diagnose-Register QV	DWORD oder UDINT[1)]		R
3	Ausgangs-Zustände ON	DWORD oder UDINT[1)]		R

1) Wert wird im Intel-Format übergeben: Data 1 enthält Low-Byte, Data 4 enthält High-Byte.

Tabelle 114: Index 0 – Bit-IO

	Bit	7	6	5	4	3	2	1	0
FB-Eingang Data 1		–	–	–	–	–	–	–	EN[1)]
FB-Ausgang Data 3		Q8[2)]	Q7[2)]	Q6[2)]	Q5[2)]	Q4[2)]	Q3[2)]	Q2[2)]	Q1[2)]
FB-Ausgang Data 4		–	–	–	–	–	–	–	QC[3)]

1) Im Zustand „1" wird der Funktionsbaustein aktiviert.

2) Der Zustand „1" wird eingenommen, wenn der ausgewählte Sicherheitsbaustein den gewählten Zustand besitzt.

3) Der Zustand „1" wird eingenommen, wenn einer der Ausgänge Q1 bis Q8 den Zustand „1" eingenommen hat.

Weitere Informationen zu diesem Baustein finden Sie im easy800-Handbuch (MN04902001Z-DE; frühere Bezeichnung AWB2528-1423D) oder in der easySoft-Hilfe.

Signalglättungsfilter: FT01 – FT32

Telegrammaufbau

Byte	Bedeutung	Wert (hex), sendet	
		Master	Slave
0	Befehl:		
	Lesen	92	–
	Schreiben	B2	–
	Antwort:		
	Lesen erfolgreich	–	C2
	Schreiben erfolgreich	–	C1
	Befehl zurückgewiesen	–	C0
1	Type	28	28
2	Instanz	01 – 20	01 – 20
3	Index	→ Tabelle 115	→ Tabelle 115
4 – 7	Data 1 – 4		
	beim Lesen	00	abhängig vom Index, → Tabelle 116
	beim Schreiben	abhängig vom Index, → Tabelle 116	00

Tabelle 115: Operandenübersicht

Index (hex)	Operand		lesen	schreiben
00	Bit-IO, → Tabelle 116		×	
01	Mode/Parameter		–	–
02	Eingangswert, Wertebereich: –32768 bis +32767	I1	×	c[1)]
03	Ausgleichszeit [0,1 s] Wertebereich: 0 bis 65535	TG	×	c[1)]
04	Proportionalverstärkung [%], Wertebereich: 0 bis 65535	KP	×	c[1)]
05	Verzögerter Ausgangswert, Wertebereich: –32768 bis +32767	QV	×	

1) Der Wert kann nur geschrieben werden, wenn er im Programm mit einer Konstanten beschaltet ist.

Tabelle 116: Index 0 – Bit-IO

	Bit	7	6	5	4	3	2	1	0
FB-Ausgang Data 3		–	–	–	–	–	–	–	EN[1)]

1) Aktiviert den Baustein bei Zustand „1“

Empfang von Netzdaten: GT01 – GT32

Telegrammaufbau

Byte	Bedeutung	Wert (hex), sendet	
		Master	Slave
0	Befehl: Lesen	92	–
	Antwort:		
	Lesen erfolgreich	–	C2
	Befehl zurückgewiesen	–	C0
1	Type	1B	1B
2	Instanz	01 – 20	01 – 20
3	Index	→ Tabelle 117	
4 – 7	Data 1 – 4	00	abhängig vom Index, → Tabelle 118, 119

Tabelle 117: Operandenübersicht

Index (hex)	Operand	lesen	schreiben
00	Bit-IO, → Tabelle 118	×	
01	Mode/Parameter, → Tabelle 119	×	–
02	Ausgangswert: Istwert QV aus dem Netzwerk	×	

→ Die Daten für Index 2 werden als 32-Bit-Wert im Intelformat (Data 1 – Low Byte bis Data 4 – High Byte) übergeben.

Tabelle 118: Index 0 – Bit-IO

	Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
FB-Ausgang Data 3		–	–	–	–	–	–	–	Q[1)]

1) Zustand „1", wenn ein neuer Wert anliegt, der vom Netzwerk NET übertragen wird.

Tabelle 119: Index 1 – Mode/Parameter (Bezeichnung des FB PUT, dessen Daten zu empfangen sind)

Mode	**Data 1**	**NET-ID[1)]**	
		0	NET-ID 1
		…	…
		7	NET-ID 8
Parameter	**Data 3**	**Instanz[2)]**	
		0	PT01
		…	…
		31	PT32

1) Nummer des Teilnehmers, der den Wert sendet. Mögliche Teilnehmernummer: 01 bis 08

2) Sendebaustein (z. B. PT 20) des NET-Teilnehmers, der sendet. Mögliche Bausteinnummer: 01 – 32

Wochenzeitschaltuhr: HW01 – HW32

Telegrammaufbau

Byte	Bedeutung	Wert (hex), sendet	
		Master	Slave
0	Befehl: Lesen	92	–
	Antwort:		
	Lesen erfolgreich	–	C2
	Befehl zurückgewiesen	–	C0
1	Type	1C	1C
2	Instanz	01 – 20	01 – 20
3	Index	→ Tabelle 120	
4 – 7	Data 1 – 4	00	abhängig vom Index, → Tabelle 121

Tabelle 120: Operandenübersicht

Index (hex)	Operand	lesen	schreiben
00	Bit-IO → Tabelle 121	×	
01	Mode/Parameter	–	–
02	Parameter → Tabelle 122	×	
	Kanal A		
03	Kanal B		
04	Kanal C		
05	Kanal D		

Tabelle 121: Index 0 – Bit-IO

	Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
FB-Ausgang Data 3		–	–	–	–	–	–	–	Q[1)]

1) Zustand „1", wenn die Einschaltbedingung erfüllt ist.

Die Daten sind in der nachfolgenden Tabelle im Motorolaformat dargestellt, obwohl sie tatsächlich im Intelformat übergeben werden.

Tabelle 122: Index 2 bis 5, Parameterkanäle A bis D

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	Date 2								**Date 1**							
ON	d4	d3	d2	d1	d0	h4	h3	h2	h1	h0	m5	m4	m3	m2	m1	m0
	Wochentag					**Stunde**					**Minute**					

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	Date 4								**Date 3**							
OFF	d4	d3	d2	d1	d0	h4	h3	h2	h1	h0	m5	m4	m3	m2	m1	m0
	Wochentag					**Stunde**					**Minute**					

m5 bis m0: Minute (0 bis 59)
h4 bis h0: Stunde (0 bis 23)
d5 bis d0: Wochentag (0 = Sonntag bis 6 = Samstag)

Beispiel

Es sollen die Parameter Kanal A der Wochenschaltuhr HW19 gelesen werden.

Byte	Bedeutung	Wert (hex), sendet	
		Master	Slave
0	Befehl: Lesen	92	–
	Antwort: Lesen erfolgreich	–	C2
1	Type	1C	1C
2	Instanz	13	13
3	Index	02	02
4	Data 1	00	62
5	Data 2	00	0B
6	Data 3	00	7B
7	Data 4	00	25

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	Date 2 = $0B_{hex}$								**Date 1 = 62_{hex}**							
ON	0	0	0	0	1	0	1	1	0	1	1	0	0	0	1	0
	Wochentag					**Stunde**					**Minute**					

Einschaltzeitpunkt:
Wochentag = 01_{hex}…Montag
Stunde = $0D_{hex}$…13 Uhr
Minute = 22_{hex}…34 Minuten

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	Date 4 = 25_{hex}								**Date 3 = $7B_{hex}$**							
OFF	0	0	1	0	0	1	0	1	0	1	1	1	1	0	1	1
	Wochentag					**Stunde**					**Minute**					

Ausschaltzeitpunkt:
Wochentag = 04_{hex}…Donnerstag
Stunde = 15_{hex}…21 Uhr
Minute = 59_{hex}…34 Minuten

Jahreszeitschaltuhr: HY01 – HY32

Telegrammaufbau

Byte	Bedeutung	Wert (hex), sendet	
		Master	Slave
0	Befehl: Lesen	92	–
	Antwort:		
	Lesen erfolgreich	–	C2
	Befehl zurückgewiesen	–	C0
1	Type	1D	1D
2	Instanz	01 – 20	01 – 20
3	Index	→ Tabelle 123	
4 – 7	Data 1 – 4	00	abhängig vom Index, → Tabelle 124

Tabelle 123: Operandenübersicht

Index (hex)	Operand	lesen	schreiben
00	Bit-IO → Tabelle 124	×	
01	Mode/Parameter	–	–
02	Parameter → Tabelle 125	×	
	Kanal A		
03	Kanal B		
04	Kanal C		
05	Kanal D		

Tabelle 124: Index 0 – Bit-IO

	Bit	7	6	5	4	3	2	1	0
FB-Ausgang Data 3		–	–	–	–	–	–	–	Q[1)]

1) Zustand „1", wenn die Einschaltbedingung erfüllt ist.

Die Daten sind in der nachfolgenden Tabelle im Motorolaformat dargestellt, obwohl sie tatsächlich im Intelformat übergeben werden.

Tabelle 125: Index 2 bis 5, Parameterkanäle A bis D

Bit	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
	Date 2								**Date 1**							
ON	y6	y5	y4	y3	y2	y1	y0	m3	m2	m1	m0	d4	d3	d2	d1	d0
	Jahr								**Monat**				**Tag**			

Bit	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
	Date 4								**Date 3**							
OFF	y6	y5	y4	y3	y2	y1	y0	m3	m2	m1	m0	d4	d3	d2	d1	d0
	Jahr								**Monat**				**Tag**			

d4 … d0: Tag (1…31), m3 … m0: Monat (1…12), y6 … y0: Jahr (0: 2000…99: 2099)

Beispiel

Es sollen die Parameter Kanal A der Jahresschaltuhr HY14 geschrieben werden.

Index 2 – 5, Parameterkanäle A – D

Bit	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
	Date 2								**Date 1**							
ON	0	0	0	0	0	1	1	0	1	1	0	0	1	1	1	0
	Jahr								**Monat**				**Tag**			

Einschaltzeitpunkt:
Tag = 14 = $0E_{hex}$ = 0000 1110b
Monat = 6 (Juni) = 06_{hex} = 0000 0110b
Jahr = 2003 = 03_{hex} = 0000 0011b

Index 2 – 5, Parameterkanäle A – D

Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	Date 2								**Date 1**							
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	Date 4								**Date 3**							
OFF	y6	y5	y4	y3	y2	y1	y0	m3	m2	m1	m0	d4	d3	d2	d1	d0
	Jahr							**Monat**				**Tag**				

Ausschaltzeitpunkt:
Tag = 3 = 03_{hex} = 0000 0011b
Monat = 10 (Oktober) = $0A_{hex}$ = 0000 1010b
Jahr = 2012 = $0C_{hex}$ = 0000 1100b

Resultierendes Telegramm:

Byte	**Bedeutung**	**Wert (hex), sendet**	
		Master	**Slave**
0	Befehl: Schreiben	B2	–
	Antwort: Schreiben erfolgreich	–	C1
1	Type	1D	1D
2	Instanz	0E	0E
3	Index	02	02
4	Data 1	8E	00
5	Data 2	06	00
6	Data 3	43	00
7	Data 4	19	00

Bedingter Sprung JC01…JC32

Telegrammaufbau

Byte	Bedeutung	Wert (hex), sendet	
		Master	Slave
0	Befehl:		
	Lesen	92	–
	Schreiben	B2	–
	Antwort:		
	Lesen erfolgreich	–	C2
	Schreiben erfolgreich	–	C1
	Befehl zurückgewiesen	–	C0
1	Type	2F	2F
2	Instanz	01 - 20	01 - 20
3	Index	00	00
4 - 7	Data 1 - 4		
	beim Lesen	00	abhängig vom Index, → Tabelle 126, 127
	beim Schreiben	abhängig vom Index, → Tabelle 126, 127	00

Tabelle 126: Operandenübersicht

Index (hex)	Daten	Data 1	Data 2	Data 3	Data 4	Read/Write
0	Bit-IO	→ Tabelle 127	–	→ Tabelle 127	–	R

Tabelle 127: Index 0 – Bit-IO

	Bit	7	6	5	4	3	2	1	0
FB-Eingang Data 1		–	–	–	–	–	–	–	EN[1)]
FB-Ausgang Data 3		–	–	–	–	–	–	–	E1[2)]

1) Im Zustand „1" wird im Programm auf die zugehörige Sprungmarke verzweigt.

2) Zustand „1" wird eingenommen, wenn die zugehörige Sprungmarke nicht gefunden wurde.

→ Weitere Informationen zu diesem Baustein finden Sie im easy800-Handbuch (MN04902001Z-DE; frühere Bezeichnung AWB2528-1423D) oder in der easySoft-Hilfe.

Wertskalierung: LS01 – LS32

Telegrammaufbau

Byte	Bedeutung	Wert (hex), sendet	
		Master	Slave
0	Befehl:		
	Lesen	92	–
	Schreiben	B2	–
	Antwort:		
	Lesen erfolgreich	–	C2
	Schreiben erfolgreich	–	C1
	Befehl zurückgewiesen	–	C0
1	Type	29	29
2	Instanz	01 – 20	01 – 20
3	Index	→ Tabelle 128	→ Tabelle 128
4 – 7	Data 1 – 4		
	beim Lesen	00	abhängig vom Index, → Tabelle 129
	beim Schreiben	abhängig vom Index, → Tabelle 129	

Tabelle 128: Operandenübersicht

Index (hex)	Operand		lesen	schreiben
00	Bit-IO, → Tabelle 129		×	
01	Mode/Parameter		–	–
02	Eingangswert, Wertebereich: 32 Bit	I1	×	c[1)]
03	Stützpunkt 1, X-Koordinate, Wertebereich: 32 Bit	X1	×	c[1)]
04	Stützpunkt 1, Y-Koordinate, Wertebereich: 32 Bit	Y1	×	c[1)]
05	Stützpunkt 2, X-Koordinate, Wertebereich: 32 Bit	X2	×	c[1)]
06	Stützpunkt 2, Y-Koordinate, Wertebereich: 32 Bit	Y2	×	c[1)]
07	Ausgangswert: beinhaltet den skalierten Eingangswert	QV	×	

1) Der Wert kann nur geschrieben werden, wenn er im Programm mit einer Konstanten beschaltet ist.

Tabelle 129: Index 0 – Bit-IO

	Bit	7	6	5	4	3	2	1	0
FB-Ausgang Data 3		–	–	–	–	–	–	–	EN[1)]

1) Aktiviert den Baustein bei Zustand „1“

Masterreset: MR01 – MR32

Telegrammaufbau

Byte	Bedeutung	Wert (hex), sendet	
		Master	Slave
0	Befehl: Lesen	92	–
	Antwort:		
	Lesen erfolgreich	–	C2
	Befehl zurückgewiesen	–	C0
1	Type	0F	0F
2	Instanz	01 – 20	01 – 20
3	Index		
	Bit-IO	00	00
	Mode	01	01
4 – 7	Data 1 – 4	00	abhängig vom Index, → Tabelle 130, 131

Tabelle 130: Index 0 – Bit-IO

	Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
FB-Eingang Data 1		–	–	–	–	–	–	–	T[1)]
FB-Ausgang Data 3		–	–	–	–	–	–	–	Q1[2)]

1) Triggerspule. Wird die Spule getriggert (erhält eine positive Flanke), wird der entsprechende Reset durchgeführt.

2) Zustand „1", wenn die Triggerspule MR…T den Zustand „1" besitzt.

Tabelle 131: Index 1 – Mode

Data 1 (hex)		
00	Q	die Ausgänge Q…, *Q…, S…, *S…, *SN…, QA01 werden auf den Zustand „0" zurückgesetzt. * entsprechend der NET-ID
01	M	der Merkerbereich MD01 bis MD48 wird auf Zustand „0" zurückgesetzt
02	ALL	wirkt auf Q und M.

Datenmultiplexer MX01…MX32

Telegrammaufbau

Byte	Bedeutung	Wert (hex), sendet	
		Master	Slave
0	Befehl:		
	Lesen	92	–
	Schreiben	B2	–
	Antwort:		
	Lesen erfolgreich	–	C2
	Schreiben erfolgreich	–	C1
	Befehl zurückgewiesen	–	C0
1	Type	31	31
2	Instanz	01 - 20	01 - 20
3	Index	00 - 0B	00 - 0B
4 - 7	Data 1 - 4		
	beim Lesen	00	abhängig vom Index, → Tabelle 132, 133
	beim Schreiben	abhängig vom Index, → Tabelle 132, 133	00

Tabelle 132: Operandenübersicht

Index (hex)	Daten	Data 1 Data 3	Data 2 Data 4	Read/Write
0	Bit-IO	→ Tabelle 133	–	R
2	Kanalauswahl: 0 bis 7	DWORD oder UDINT[1)]		R/W[2)]
3	Eingangswert Kanal 1	DWORD oder UDINT[1)]		R/W[2)]
4	Eingangswert Kanal 2	DWORD oder UDINT[1)]		R/W[2)]
5	Eingangswert Kanal 3	DWORD oder UDINT[1)]		R/W[2)]
6	Eingangswert Kanal 4	DWORD oder UDINT[1)]		R/W[2)]
7	Eingangswert Kanal 5	DWORD oder UDINT[1)]		R/W[2)]
8	Eingangswert Kanal 6	DWORD oder UDINT[1)]		R/W[2)]
9	Eingangswert Kanal 7	DWORD oder UDINT[1)]		R/W[2)]
A	Eingangswert Kanal 8	DWORD oder UDINT[1)]		R/W[2)]
B	Ausgangswert QV	DWORD oder UDINT[1)]		R

1) Wert wird im Intel-Format übergeben: Data 1 enthält Low-Byte, Data 4 enthält High-Byte.

2) Der Wert kann nur beschrieben werden, wenn er im Programm mit einer Konstanten beschaltet ist.

Tabelle 133: Index 0 – Bit-IO

	Bit	7	6	5	4	3	2	1	0
FB-Eingang Data 1		–	–	–	–	–	–	–	EN[1)]
FB-Ausgang Data 3		–	–	–	–	–	–	–	E1[2)]

1) Im Zustand „1" wird der gewählte Eingangswert in den Ausgangswert eingetragen.

2) Zustand „1" wird eingenommen, wenn die Kanalauswahl ungültig ist.

→ Weitere Informationen zu diesem Baustein finden Sie im easy800-Handbuch (MN04902001Z-DE; frühere Bezeichnung AWB2528-1423D) oder in der easySoft-Hilfe.

Zahlenwandler: NC01 – NC32

Telegrammaufbau

Byte	Bedeutung	Wert (hex), sendet	
		Master	Slave
0	Befehl:		
	Lesen	92	–
	Schreiben	B2	–
	Antwort:		
	Lesen erfolgreich	–	C2
	Schreiben erfolgreich	–	C1
	Befehl zurückgewiesen	–	C0
1	Type	2A	2A
2	Instanz	01 – 20	01 – 20
3	Index	→ Tabelle 134	→ Tabelle 134
4 – 7	Data 1 – 4		
	beim Lesen	00	abhängig vom Index, → Tabelle 135, 136
	beim Schreiben	abhängig vom Index, → Tabelle 135, 136	00

Tabelle 134: Operandenübersicht

Index (hex)	Operand	lesen	schreiben
00	Bit-IO, → Tabelle 135	×	
01	Mode, → Tabelle 136	×	
02	Eingangswert: I1 Operand der gewandelt werden soll	×	c[1)]
03	Ausgangswert: QV beinhaltet das Wandlungsergebnis	×	

1) Der Wert kann nur geschrieben werden, wenn er im Programm mit einer Konstanten beschaltet ist.

→ Die Daten für Index 2 und 3 werden als 32-Bit-Wert im Intelformat (Data 1 – Low Byte…Data 2 – High Byte) übergeben.

Tabelle 135: Index 0 – Bit-IO

	Bit	7	6	5	4	3	2	1	0
FB-Ausgang Data 3		–	–	–	–	–	–	–	EN[1)]

1) Aktiviert den Baustein bei Zustand „1“

Tabelle 136: Index 1 – Mode

Data 1 (hex)		
00	BCD	Wandelt einen BCD-codierten Dezimalwert in einen ganzzahligen Wert um.
01	BIN	Wandelt einen ganzzahligen Wert in einen BCD-codierten Dezimalwert um.

Betriebsstundenzähler: OT01 – OT04

→ Weitere Informationen finden Sie im S40-Anwendungshinweis AN27K21d.exe „EASY800/MFD-DP-Hantierungsbausteine für PS416 und PS4-341.

Telegrammaufbau

Byte	Bedeutung	Wert (hex), sendet	
		Master	Slave
0	Befehl:		
	Lesen	92	–
	Schreiben	B2	–
	Antwort:		
	Lesen erfolgreich	–	C2
	Schreiben erfolgreich	–	C1
	Befehl zurückgewiesen	–	C0
1	Type	1E	1E
2	Instanz	01 – 04	01 – 04
3	Index	→ Tabelle 137	→ Tabelle 137
4 – 7	Data 1 – 4		
	beim Lesen	00	abhängig vom Index, → Tabelle 138
	beim Schreiben	abhängig vom Index, → Tabelle 138	00

Tabelle 137: Operandenübersicht

Index (hex)	Operand		lesen	schreiben
00	Bit-IO, → Tabelle 138		×	
01	Mode/Parameter		–	–
02	oberer Schwellwert	I1	×	c[1)]
03	Istwert des Betriebsstundenzählers	QV	×	

1) Der Wert kann nur geschrieben werden, wenn er im Programm mit einer Konstanten beschaltet ist.

Tabelle 138: Index 0 – Bit-IO

	Bit	7	6	5	4	3	2	1	0
FB-Eingang Data 1		–	–	–	–	–	–	RE[1)]	EN[2)]
FB-Ausgang Data 3		–	–	–	–	–	–	–	Q1[3)]

1) Resetspule: Zustand „1" setzt den Zähler-Istwert auf Null zurück.
2) Freigabespule
3) Zustand „1", wenn Sollwert erreicht wurde (größer/gleich)

→ Die Daten für Index 2 und 3 werden als 32-Bit-Wert im Intelformat (Data 1 – Low Byte bis Data 4 – High Byte) übergeben.

Impulsausgabe PO01…PO02

Telegrammaufbau

Byte	Bedeutung	Wert (hex), sendet	
		Master	Slave
0	Befehl:		
	Lesen	92	–
	Schreiben	B2	–
	Antwort:		
	Lesen erfolgreich	–	C2
	Schreiben erfolgreich	–	C1
	Befehl zurückgewiesen	–	C0
1	Type	32	32
2	Instanz	01 - 02	01 - 02
3	Index	00 - 0A	00 - 0A
4 - 7	Data 1 - 4		
	beim Lesen	00	abhängig vom Index, → Tabelle 139, 140
	beim Schreiben	abhängig vom Index, → Tabelle 139, 140	00

Tabelle 139: Operandenübersicht

Index (hex)	Daten	Data 1 Data 3	Data 2 Data 4	Read/ Write
0	Bit-IO	→ Tabelle 140	–	R
2	Impulsanzahl im Fahrbetrieb I1: 0 bis 2147483647	DWORD oder UDINT[1)]		R/W[2)]
3	Startfrequenz FS: 0 bis 5000 Hz	DWORD oder UDINT[1)]		R/W[2)]
4	Betriebsfrequenz FO: 0 bis 5000 Hz	DWORD oder UDINT[1)]		R/W[2)]
5	Frequenzänderung im Hochlauf RF: 0 bis 65535 mHz	DWORD oder UDINT[1)]		R/W[2)]
6	Frequenzänderung beim Bremsen BF: 0 bis 65535 mHz	DWORD oder UDINT[1)]		R/W[2)]
7	Schrittanzahl im Tippbetrieb P1: 0 bis 65535	DWORD oder UDINT[1)]		R/W[2)]
8	Frequenz im Tippbetrieb PF: 0 bis 5000 Hz	DWORD oder UDINT[1)]		R/W[2)]
9	Aktuelle Schrittzahl QV	DWORD oder UDINT[1)]		R
A	Aktuelle Frequenz QF	DWORD oder UDINT[1)]		R

1) Wert wird im Intel-Format übergeben: Data 1 enthält Low-Byte, Data 4 enthält High-Byte.

2) Der Wert kann nur beschrieben werden, wenn er im Programm mit einer Konstanten beschaltet ist.

Tabelle 140: Index 0 – Bit-IO

	Bit	7	6	5	4	3	2	1	0
FB-Eingang Data 1		–	–	–	–	TP[1)]	BR[2)]	ST[3)]	EN[4)]
FB-Ausgang Data 3		–	–	–	–	–	–	E1[5)]	AC[6)]

1) Bei Erhalt einer positiven Flanke wird der Tippbetrieb gestartet.
2) Bei Erhalt einer positiven Flanke wird der Fahrauftrag abgebrochen.
3) Bei Erhalt einer positiven Flanke wird der Fahrauftrag gestartet.
4) Im Zustand „1" wird der Funktionsbaustein aktiviert.
5) Zustand „1" wird eingenommen, wenn die Parametereingabe ungültig ist.
6) Zustand „1" wird eingenommen, wenn ein Fahrauftrag aktiv ist.

→ Weitere Informationen zu diesem Baustein finden Sie im easy800-Handbuch (MN04902001Z-DE; frühere Bezeichnung AWB2528-1423D) oder in der easySoft-Hilfe.

Senden von Netzdaten: PT01 – PT32

Telegrammaufbau

Byte	Bedeutung	Wert (hex), sendet	
		Master	**Slave**
0	Befehl:		
	Lesen	92	–
	Schreiben	B2	–
	Antwort:		
	Lesen erfolgreich	–	C2
	Schreiben erfolgreich	–	C1
	Befehl zurückgewiesen	–	C0
1	Type	1F	1F
2	Instanz	01 - 20	01 - 20
3	Index	00 - 02	00 - 02
4 - 7	Data 1 - 4		
	beim Lesen	00	abhängig vom Index, → Tabelle 141, 142
	beim Schreiben	abhängig vom Index, → Tabelle 141, 142	00

Tabelle 141: Operandenübersicht

Index (hex)	Daten	Data 1	Data 2	Data 3	Data 4	Read/ Write
0	Bit-IO	→ Tabelle 142	–	→ Tabelle 142	–	R
2	Sollwert QV für das Netzwerk	DWORD oder UDINT[1)]				R

1) Wert wird im Intel-Format übergeben: Data 1 enthält Low-Byte, Data 4 enthält High-Byte.

Tabelle 142: Index 0 – Bit-IO

	Bit	7	6	5	4	3	2	1	0
FB-Eingang Data 1		–	–	–	–	–	–	–	T[1)]
FB-Ausgang Data 3		–	–	–	–	–	E1[2)]	AC[3)]	Q1[4)]

1) Triggerspule. Wird die Spule getriggert (erhält eine positive Flanke), wird der entsprechende Wert dem NET zur Verfügung gestellt.
2) Zustand „1" wird eingenommen, wenn der Sendeauftrag wegen eines Fehlers abgebrochen wurde.
3) Zustand „1" wird eingenommen, wenn die Triggerspule getriggert wird. Zustand „0" wird eingenommen, wenn der Sendeauftrag erfolgreich durchgeführt wurde oder wegen eines Fehlers abgebrochen wurde.
4) Zustand „1", wenn der Zustand der Triggerspule ebenfalls „1" ist.

→ Weitere Informationen zu diesem Baustein finden Sie im easy800-Handbuch (MN04902001Z-DE; frühere Bezeichnung AWB2528-1423D) oder in der easySoft-Hilfe.

Pulsweitenmodulation: PW01 – PW02

Telegrammaufbau

Byte	Bedeutung	Wert (hex), sendet	
		Master	**Slave**
0	Befehl:		
	Lesen	92	–
	Schreiben	B2	–
	Antwort:		
	Lesen erfolgreich	–	C2
	Schreiben erfolgreich	–	C1
	Befehl zurückgewiesen	–	C0
1	Type	2B	2B
2	Instanz	01 – 02	01 – 02
3	Index	→ Tabelle 143	→ Tabelle 143
4 – 7	Data 1 – 4		
	beim Lesen	00	abhängig vom Index, → Tabelle 144
	beim Schreiben	abhängig vom Index, → Tabelle 144	00

Tabelle 143: Operandenübersicht

Index (hex)	Operand		lesen	schreiben
00	Bit-IO, → Tabelle 144		×	
01	Mode/Parameter		–	–
02	Stellgröße, Wertebereich: 0 bis 4095 (12 Bit)	SV	×	c[1)]
03	Periodendauer [ms], Wertebereich: 0 bis 65535	PD	×	c[1)]
04	Mindesteinschaltdauer [ms], Wertebereich: 0 bis 65535	ME	×	c[1)]

1) Der Wert kann nur geschrieben werden, wenn er im Programm mit einer Konstanten beschaltet ist.

Tabelle 144: Index 0 – Bit-IO

	Bit	7	6	5	4	3	2	1	0
FB-Eingang Data 1		–	–	–	–	–	–	–	EN[1)]
FB-Ausgang Data 3		–	–	–	–	–	–	–	E1[2)]

1) Aktiviert den Baustein bei Zustand „1".

2) Zustand „1", wenn die Mindesteinschaltdauer oder die Mindestausschaltdauer unterschritten wird

Uhr synchronisieren: SC01

Telegrammaufbau

Byte	Bedeutung	Wert (hex), sendet	
		Master	Slave
0	Befehl: Lesen	92	–
	Antwort:		
	Lesen erfolgreich	–	C2
	Befehl zurückgewiesen	–	C0
1	Type	20	20
2	Instanz	01	01
3	Index	→ Tabelle 145	
4 – 7	Data 1 – 4	00	abhängig vom Index, → Tabelle 146

Tabelle 145: Operandenübersicht

Index (hex)	Operand	lesen	schreiben
00	Bit-IO, → Tabelle 146	×	
01	Mode/Parameter	–	–

Tabelle 146: Index 0 – Bit-IO

	Bit	7	6	5	4	3	2	1	0
FB-Eingang Data 1		–	–	–	–	–	–	–	T[1]
FB-Ausgang Data 3		–	–	–	–	–	–	–	Q1[2]

1) Triggerspule. Wird die Spule getriggert (erhält eine positive Flanke), so wird automatisch das aktuelle Datum, der Wochentag und die Uhrzeit des sendenden Teilnehmers in das Netzwerk NET gestellt.

2) Zustand „1", wenn der Zustand der Triggerspule SC01T_ ebenfalls „1" ist.

Serielle Ausgabe SP01…SP32

Telegrammaufbau

Byte	Bedeutung	Wert (hex), sendet	
		Master	Slave
0	Befehl:		
	Lesen	92	–
	Schreiben	B2	–
	Antwort:		
	Lesen erfolgreich	–	C2
	Schreiben erfolgreich	–	C1
	Befehl zurückgewiesen	–	C0
1	Type	35	35
2	Instanz	01 - 20	01 - 20
3	Index	00	00
4 - 7	Data 1 - 4 beim Lesen	00	abhängig vom Index, → Tabelle 147, 148

Tabelle 147: Operandenübersicht

Index (hex)	Daten	Data 1	Data 2	Data 3	Data 4	Read/Write
0	Bit-IO	→ Tabelle 148	–	→ Tabelle 148	–	R

Tabelle 148: Index 0 – Bit-IO

	Bit	7	6	5	4	3	2	1	0
FB-Eingang Data 1		–	–	–	–	–	–	T[1)]	EN[2)]
FB-Ausgang Data 3		–	–	–	–	–	–	E1[3)]	AC[4)]

1) Bei Erhalt einer positiven Flanke wird der Sendevorgang ausgelöst.
2) Im Zustand „1" wird der Funktionsbaustein aktiviert.
3) Der Zustand „1" wird eingenommen, wenn beim Sendevorgang ein Fehler aufgetreten ist.
4) Der Zustand „1" wird eingenommen, wenn der Sendevorgang aktiv ist.

Schieberegister SR01…SR32

→ Weitere Informationen zu diesem Baustein finden Sie im easy800-Handbuch (MN04902001Z-DE; frühere Bezeichnung AWB2528-1423D) oder in der easySoft-Hilfe.

Telegrammaufbau

Byte	Bedeutung	Wert (hex), sendet	
		Master	**Slave**
0	Befehl:		
	Lesen	92	–
	Schreiben	B2	–
	Antwort:		
	Lesen erfolgreich	–	C2
	Schreiben erfolgreich	–	C1
	Befehl zurückgewiesen	–	C0
1	Type	33	33
2	Instanz	01 - 20	01 - 20
3	Index	00 - 0B	00 - 0B
4 - 7	Data 1 - 4		
	beim Lesen	00	abhängig vom Index, → Tabelle 149, 150
	beim Schreiben	abhängig vom Index, → Tabelle 149, 150	00

Tabelle 149: Operandenübersicht

Index (hex)	Daten	Data 1	Data 2 Data 4	Data 3	Read/ Write
0	Bit-IO	→ Tabelle 150	–	→ Tabelle 150	R
1	Mode	→ Tabelle 151	–	–	R
2	Dateneingang vorwärts I1	DWORD oder UDINT[1)]			R/W[2)]
3	Dateneingang rückwärts I2	DWORD oder UDINT[1)]			R/W[2)]
4	Datenausgang 1 (D1)	DWORD oder UDINT[1)]			R
5	Datenausgang 2 (D2)	DWORD oder UDINT[1)]			R
6	Datenausgang 3 (D3)	DWORD oder UDINT[1)]			R
7	Datenausgang 4 (D4)	DWORD oder UDINT[1)]			R
8	Datenausgang 5 (D5)	DWORD oder UDINT[1)]			R
9	Datenausgang 6 (D6)	DWORD oder UDINT[1)]			R
A	Datenausgang 7 (D7)	DWORD oder UDINT[1)]			R
B	Datenausgang 8 (D8)	DWORD oder UDINT[1)]			R

1) Wert wird im Intel-Format übergeben: Data 1 enthält Low-Byte, Data 4 enthält High-Byte.

2) Der Wert kann nur beschrieben werden, wenn er im Programm mit einer Konstanten beschaltet ist.

Tabelle 150: Index 0 – Bit-IO

	Bit	7	6	5	4	3	2	1	0
FB-Eingang Data 1		–	–	BD[1]	FD[2]	RE[3]	BP[4]	FP[5]	EN[6]
FB-Ausgang Data 3		Q8[7]	Q8[7]	Q6[7]	Q5[7]	Q4[7]	Q3[7]	Q2[7]	Q1[7]

1) Eingangs-Bitwert für die Rückwärts-Schiebeoperation im Mode BIT.
2) Eingangs-Bitwert für die Vorwärts-Schiebeoperation im Mode BIT.
3) Im Zustand „1" wird der Funktionsbaustein zurückgesetzt.
4) Bei Erhalt einer positiven Flanke wird im Mode BIT der Wert von BD in das letzte Register-Feld Q8 eingetragen und die ursprünglichen Inhalte der Registerfelder werden um ein Feld in Richtung niedrigerer Feldnummern verschoben. Bei Erhalt einer positiven Flanke wird im Mode DW der Wert von I2 in das letzte Register-Feld D8 eingetragen und die ursprünglichen Inhalte der Registerfelder werden um ein Feld in Richtung niedrigerer Feldnummern verschoben.
5) Bei Erhalt einer positiven Flanke wird im Mode BIT der Wert von FD in das erste Register-Feld Q1 eingetragen und die ursprünglichen Inhalte der Registerfelder werden um ein Feld in Richtung höherer Feldnummern verschoben. Bei Erhalt einer positiven Flanke wird im Mode DW der Wert von I1 in das erste Register-Feld D1 eingetragen und die ursprünglichen Inhalte der Registerfelder werden um ein Feld in Richtung höherer Feldnummern verschoben.
6) Im Zustand „1" wird der Funktionsbaustein aktiviert.
7) Zustand der acht Felder des Bit-Schieberegisters.

Tabelle 151: : Index 1 – Mode

Data 1 (hex)		
00	BIT	Betriebsart: Bit schieben
01	DW	Betriebsart: Doppelwort schieben

→ Weitere Informationen zu diesem Baustein finden Sie im easy800-Handbuch (MN04902001Z-DE; frühere Bezeichnung AWB2528-1423D) oder in der easySoft-Hilfe.

Sollzykluszeit: ST01

Telegrammaufbau

Byte	Bedeutung	Wert (hex), sendet	
		Master	**Slave**
0	Befehl:		
	Lesen	92	–
	Schreiben	B2	–
	Antwort:		
	Lesen erfolgreich	–	C2
	Schreiben erfolgreich	–	C1
	Befehl zurückgewiesen	–	C0
1	Type	2C	2C
2	Instanz	01	01
3	Index	→ Tabelle 152	→ Tabelle 152
4 – 7	Data 1 – 4		
	beim Lesen	00	abhängig vom Index, → Tabelle 153
	beim Schreiben	abhängig vom Index, → Tabelle 153	00

Tabelle 152: Operandenübersicht

Index (hex)	Operand	lesen	schreiben
00	Bit-IO, → Tabelle 153	×	
01	Mode/Parameter	–	–
02	Zykluszeit in ms, I1 Wertebereich: 0 – 1000	×	c[1)]

1) Der Wert kann nur geschrieben werden, wenn er im Programm mit einer Konstanten beschaltet ist.

Tabelle 153: Index 0 – Bit-IO

	Bit	7	6	5	4	3	2	1	0
FB-Ausgang Data 3		–	–	–	–	–	–	–	EN[1)]

1) Aktiviert den Baustein bei Zustand „1“

Zeitrelais: T01 – T32

Telegrammaufbau

Byte	Bedeutung	Wert (hex), sendet	
		Master	Slave
0	Befehl:		
	Lesen	92	–
	Schreiben	B2	–
	Antwort:		
	Lesen erfolgreich	–	C2
	Schreiben erfolgreich	–	C1
	Befehl zurückgewiesen	–	C0
1	Type	21	21
2	Instanz	01 – 20	01 – 20
3	Index	→ Tabelle 154	→ Tabelle 154
4 – 7	Data 1 – 4		
	beim Lesen	00	abhängig vom Index, → Tabelle 155, 156
	beim Schreiben	abhängig vom Index, → Tabelle 155, 156	

Tabelle 154: Operandenübersicht

Index (hex)	Operand		lesen	schreiben
00	Bit-IO, → Tabelle 155		×	
01	Mode/Parameter, → Tabelle 156		×	
02	Sollwert 1: Zeitsollwert 1	I1	×	c[1]
03	Sollwert 2: Zeitsollwert 2 (bei Zeitrelais mit 2 Sollwerten)	I2	×	c[1]
04	Istwert: Abgelaufene Istzeit im RUN-Betrieb	QV	×	

1) Der Wert kann nur geschrieben werden, wenn er im Programm mit einer Konstanten beschaltet ist.

→ Die Daten für Index 2 bis 4 werden als 32-Bit-Wert im Intelformat (Data 1 – Low Byte bis Data 4 – High Byte) übergeben.

Tabelle 155: Index 0 – Bit-IO

	Bit	7	6	5	4	3	2	1	0
FB-Eingang Data 1		–	–	–	–	–	ST[1]	EN[2]	RE[3]
FB-Ausgang Data 3		–	–	–	–	–	–	–	Q1[4]

1) Stopp, das Zeitrelais wird gestoppt (Stoppspule)
2) Enable, das Zeitrelais wird gestartet (Triggerspule)
3) Reset, das Zeitrelais wird zurückgesetzt (Resetspule)
4) Schaltkontakt

Tabelle 156: Index 1 – Mode/Parameter

Mode	**Data 1**	**Betriebsart**
	0	Ansprechverzögert
	1	Ansprechverzögert mit Zufalls-Sollwert
	2	Rückfallverzögert
	3	Rückfallverzögert mit Zufalls-Sollwert
	4	Ansprechverzögert und Rückfallverzögert (zwei Zeitsollwerte)
	5	Ansprechverzögert und Rückfallverzögert, jeweils mit Zufalls-Sollwert (zwei Zeitsollwerte)
	6	Impulsgeber
	7	Blink-Relais (zwei Zeitsollwerte)
	8	Rückfallverzögert, retriggerbar (easy600 Mode)
	9	Rückfallverzögert mit Zufalls-Sollwert, retriggerbar (easy600 Mode)
Parameter	**Data 3**	**Betriebsart**
	0	S (Millisekunden)
	1	M:S (Sekunden)
	2	H:M (Minuten)

Tabellenfunktion TB01…TB32

Telegrammaufbau

Byte	Bedeutung	Wert (hex), sendet	
		Master	Slave
0	Befehl:		
	Lesen	92	–
	Schreiben	B2	–
	Antwort:		
	Lesen erfolgreich	–	C2
	Schreiben erfolgreich	–	C1
	Befehl zurückgewiesen	–	C0
1	Type	34	34
2	Instanz	01 - 20	01 - 20
3	Index	00 - 04	00 - 04
4 - 7	Data 1 - 4		
	beim Lesen	00	abhängig vom Index, → Tabelle 157, 158
	beim Schreiben	abhängig vom Index, → Tabelle 157, 158	00

Tabelle 157: Operandenübersicht

Index (hex)	Daten	Data 1 Data 3	Data 2 Data 4	Read/ Write
0	Bit-IO	→ Tabelle 158	–	R
2	Eingangswert I1 für Tabelle von TB…	DWORD oder UDINT[1)]		R/W[2)]
3	Ausgangswert QV aus Tabelle von TB…	DWORD oder UDINT[1)]		R
4	Anzahl Einträge QN in Tabelle von TB…	DWORD oder UDINT[1)]		R

1) Wert wird im Intel-Format übergeben: Data 1 enthält Low-Byte, Data 4 enthält High-Byte.

2) Der Wert kann nur beschrieben werden, wenn er im Programm mit einer Konstanten beschaltet ist.

Tabelle 158: Index 0 – Bit-IO

	Bit	7	6	5	4	3	2	1	0
FB-Eingang Data 1		–	–	–	RE[1)]	RL[2)]	RF[3)]	WP[4)]	EN[5)]
FB-Ausgang Data 3		–	–	–	–	–	–	TF[6)]	TE[7)]

1) Bei Erhalt einer positiven Flanke werden alle Einträge aus der Tabelle entfernt. Die Anzahl der Tabelleneinträge QN wird auf „0" gesetzt.
2) Bei Erhalt einer positiven Flanke wird der neuste Eintrag in der Tabelle am Ausgang QV ausgegeben und aus der Tabelle entfernt. Die Anzahl der Tabelleneinträge QN wird um eins erniedrigt.
3) Bei Erhalt einer positiven Flanke wird der älteste Eintrag in der Tabelle am Ausgang QV ausgegeben und aus der Tabelle entfernt. Die Anzahl der Tabelleneinträge QN wird um eins erniedrigt.
4) Bei Erhalt einer positiven Flanke wird der Wert von I1 in die Tabelle übernommen und die Anzahl der Tabelleneinträge QN um eins erhöht, solange die maximale Anzahl von Einträgen nicht überschritten wird. In diesem Fall wird der Wert von I1 am Ausgang QV ausgegeben.
5) Im Zustand „1" wird der Funktionsbaustein aktiviert.
6) Der Zustand „1" wird eingenommen, wenn die Tabelle voll ist.
7) Der Zustand „1" wird eingenommen, wenn die Tabelle leer ist.

Weitere Informationen zu diesem Baustein finden Sie im easy800-Handbuch (MN04902001Z-DE; frühere Bezeichnung AWB2528-1423D) oder in der easySoft-Hilfe.

Wertbegrenzung: VC01 – VC32

Telegrammaufbau

Byte	Bedeutung	Wert (hex), sendet	
		Master	Slave
0	Befehl:		
	Lesen	92	–
	Schreiben	B2	–
	Antwort:		
	Lesen erfolgreich	–	C2
	Schreiben erfolgreich	–	C1
	Befehl zurückgewiesen	–	C0
1	Type	2D	2D
2	Instanz	01 – 20	01 – 20
3	Index	→ Tabelle 159	→ Tabelle 159
4 – 7	Data 1 – 4		
	beim Lesen	00	abhängig vom Index, → Tabelle 160
	beim Schreiben	abhängig vom Index, → Tabelle 160	00

Tabelle 159: Operandenübersicht

Index (hex)	Operand		lesen	schreiben
00	Bit-IO, → Tabelle 160		×	
01	Mode/Parameter		–	–
02	Eingangswert	I1	×	c[1)]
03	Oberer Grenzwert	SH	×	c[1)]
04	Unterer Grenzwert	SL	×	c[1)]
05	Ausgangswert: gibt innerhalb der gesetzten Grenzen den Wert aus, der am Eingang I1 anliegt.	QV	×	

1) Der Wert kann nur geschrieben werden, wenn er im Programm mit einer Konstanten beschaltet ist.

Tabelle 160: Index 0 – Bit-IO

	Bit	7	6	5	4	3	2	1	0
FB-Ausgang Data 3		–	–	–	–	–	–	–	EN[1)]

1) Aktiviert den Baustein bei Zustand „1“

Analyse – Fehlercodes über easyLink

Das Basisgerät easy800/MFD liefert im Fall eines nicht korrekt gewählten Betriebsmodus oder eines ungültigen Telegramms einen definierten Fehlercode zurück. Die Übergabe des Fehlercodes ist wie folgt aufgebaut:

Telegrammaufbau

Byte	Bedeutung	Slave sendet (Wert hex)
0	Antwort	
	Befehl zurückgewiesen	C0
1	Type	
2	Instanz	
3	Index	
4	Fehlercode	→ Tabelle 161
5 – 7	Data 2 – 4	

Tabelle 161: Fehlercodes

Fehler-code	Beschreibung
0x00	kein Fehler
0x03	formaler Fehler in der Antwort bezüglich Type, Instanz oder Index
0x04	keine Kommunikation möglich (Timeout)
0x05	DP-Modul hat nur 0xC0 gesendet (Easy800 Basic II, MFD Version I).
0x45	der durch Type und Index ausgewählte Wert darf nicht beschrieben werden (Bit-IO, Mode/Parameter oder Ausgangswert).
0x46	der durch Type und Index ausgewählte Wert ist nicht mit einer Konstanten beschaltet.
0x9E	Zugriff auf die FB-Daten nicht möglich (Programm-Download aktiv).
0x9F	Type ist ungültig (kein definierter FB, abhängig auch von der Version des angesprochenen Gerätes).
0xA0	durch Type und Instanz ausgewählter FB existiert nicht im Programm.
0xA1	Index bezogen auf den angegebenen FB-Type ist ungültig.

11 Was ist, wenn…?

RUN-LED

Zustand der RUN-LED	Mögliche Ursache	Abhilfemaßnahme
AUS	EASY221-CO ist entweder ausgeschaltet oder befindet sich im Reset-Zustand.	EASY221-CO einschalten und mit Netzspannung versorgen.
Flimmernd	Die Auto Baudrate Erkennung läuft gerade (flimmert abwechselnd mit der Error-LED).	Master-Steuerung bzw. den Bus auf Kommunikation hin überprüfen.
Einmal aufblitzen	Das Gerät befindet sich im Zustand STOPPED.	Zustand des NMT (Netzwerkmanagement) ändern, siehe Kapitel 4.3
Blinkend	Das Gerät befindet sich im Zustand PRE-OPERATIONAL.	
EIN	Das Gerät befindet sich im Zustand OPERATIONAL.	

Error-LED

Zustand der Error-LED	Mögliche Ursache	Abhilfemaßnahme
AUS	EASY221-CO arbeitet korrekt. Ist die RUN LED ebenfalls aus, so ist EASY221-CO entweder ausgeschaltet oder befindet sich im Reset-Zustand.	Spannungsversorgung einschalten.
Einmal aufblitzen	Mindestens ein Fehlerzähler des CANopen-Controllers hat das „Warning Limit" erreicht oder überschritten. Es sind zu viele Fehler auf dem CANopen-Bus aufgetreten.	Überprüfen des Busses auf eventuelle äußere Störungen EMV-Probleme – ist der Schirm richtig aufgelegt? Ist die Baudrate der anderer Teil-nehmer richtig eingestellt?
Flimmernd	Die Auto Baudrate Erkennung läuft gerade (flimmert abwechselnd mit der RUN-LED).	Master-Steuerung bzw. den Bus auf Kommunikation hin überprüfen.
Zweimal aufblitzen	Ein Schutzereignis durch „Guard Event" oder ein Lebenszeichenereignis durch „Heartbeat Event" ist aufgetreten.	Konfigurationseinrichtung überprüfen.
EIN	Der CANopen-Controller hat in den BUS-OFF Status gewechselt.	Überprüfen ob die richtige NODE ID eingestellt wurde.

Anhang

Technische Daten

Allgemeines		
Normen und Bestimmungen		EN 61000-6-2; EN 61000-6-3; EN 61000-6-4; EN 61000-6-5, EN 61000-6-6; IEC 60068-2-27, IEC 50178
Abmessungen (B × H × T)	mm	35,5 × 90 × 56,5
Gewicht	g	150
Montage		Hutschiene DIN 50022, 35 mm Schraubmontage mit Gerätefüße ZB4-101-GF1 (Zubehör)
Klimatische Umgebungstemperaturen (Kälte nach IEC 60068-2-1, Wärme nach IEC 60068-2-2)		
Betriebsumgebungstemperatur waagerechter/ senkrechter Einbau	°C	–25 bis +55
Betauung		Betauung durch geeignete Maßnahmen verhindern
Lager-/Transporttemperatur	°C	–40 bis +70
Relative Luftfeuchte (IEC 60068-2-30), keine Betauung	%	5 bis 95
Luftdruck (Betrieb)	hPa	795 bis 1080
Korrosionsunempfindlichkeit (IEC 60068-2-42, IEC 60068-2-43)		SO_2 10 cm^3/m^3, 4 Tage H_2S 1 cm^3/m^3, 4 Tage
Mechanische Umgebungsbedingungen		
Verschmutzungsgrad		2
Schutzart (EN 50178, IEC 60529, VBG4)		IP20
Schwingungen (IEC 60068-2-6)		
konstante Amplitude 0,15 mm	Hz	10 bis 57
konstante Beschleunigung 2 g	Hz	57 bis 150
Schocken (IEC 60068-2-27) Halbsinus 15 g/11ms	Schocks	18

Kippfallen (IEC 60068-2-31) Fallhöhe	mm	50
Freier Fall, verpackt (IEC 60068-2-32)	m	1
Elektromagnetische Verträglichkeit (EMV)		
Elektrostatische Entladung (ESD) (IEC/EN 61000-4-2, Schärfegrad 3)		
Luftentladung	kV	8
Kontaktentladung	kV	6
Elektromagnetische Felder (IEC/EN 61000-4-3)	V/m	10
Funkentstörung (EN 55011, EN 55022) Grenzwertklasse		B
Burst Impulse (IEC/EN 61000-4-4, Schärfegrad 3)		
Versorgungsleitungen	kV	2
Signalleitungen	kV	2
Energiereiche Impulse (Surge) easy-AC (IEC/EN 61000-4-5), Versorgungsleitung symmetrisch	kV	1
Energiereiche Impulse (Surge) easy-DC (IEC/EN 61 000-4-5, Schärfegrad 2), Versorgungsleitung symmetrisch	kV	0,5
Einströmung (IEC/EN 61000-4-6)	V	10
Isolationsfestigkeit		
Bemessung der Luft- und Kriechstrecken		EN 50178, UL508, CSA C22.2 No. 142
Isolationsfestigkeit		EN 50178
Werkzeug und Anschlussquerschnitte		
Leiterquerschnitte		
eindrähtig minimal bis maximal	mm^2	0,2 bis 4
	AWG	22 bis 12
feindrähtig mit Aderendhülse minimal bis maximal	mm^2	0,2 bis 2,5
	AWG	22 bis 12
Schlitzschraubendreherbreite	mm	3,5 × 0,8
Anzugsdrehmoment	Nm	0,5

Stromversorgung		
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	200
Spannungseinbrüche (IEC/EN 61131-2)	ms	10
Verlustleistung bei 24 V DC, typisch	W	4,8
LED-Anzeigen		
LED-MS (Modul-Status)	Farbe	Grün/rot
LED-NS (Network-Status)	Farbe	Grün/rot
CANopen		
Anschluss Gerät		8-polige-RJ-45-Buchse
Potentialtrennung		Bus zu Spannungsversorgung (einfach) Bus und Stromversorgung zu easy-Basisgerät (sichere Trennung)
Funktion		CANopen-Slave
Schnittstelle		CANopen (CAN)
Busprotokoll		CANopen
Baudraten, automatische Suche bis	kBd	1000
Busabschlusswiderstände		Separat am Bus anbringbar
Busadressen, über easy-Basisgerät mit Display oder easySoft adressierbar		1 bis 127
Dienste		
Modul Eingänge		alle Daten S1 bis S8 (easy600)
Modul Ausgänge		alle Daten R1 bis R16 (easy600)
Modul Steuerbefehle		Read/Write Uhrzeit, Tag, Sommer/Winterzeit Alle Parameter der easy-Funktionen

Abmessungen

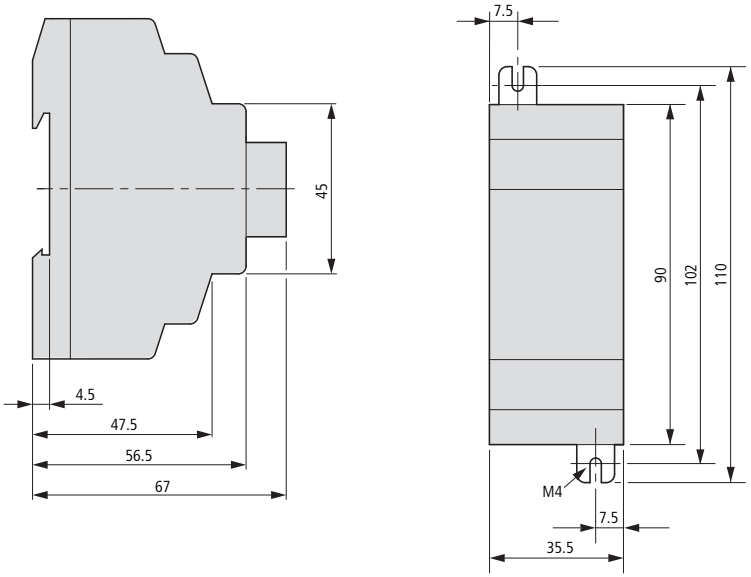


Abbildung 26: Abmessung EASY221-CO, Maße in mm

Glossar

Das Glossar bezieht sich auf Themen rund um CANopen.

Abschlusswiderstand	Widerstand am Anfang und am Ende einer Bus-Leitung, der störende Signalreflexionen verhindert und zur Leitungsanpassung bei Busleitungen dient. Abschlusswiderstände müssen immer die letzte Einheit am Ende eines Bussegments sein.
Access Type	Zugriffsberechtigung auf ein Objekt.
Acknowledge	Quittung des Empfängers für ein empfangenes Signal.
Adresse	Nummer zur Kennzeichnung z. B. eines Speicherplatzes, eines Systems oder eines Moduls innerhalb eines Netzwerks.
Adressierung	Zuweisung bzw. Einstellung einer Adresse, z. B. für ein Modul in einem Netzwerk.
aktives Metallteil	Leiter oder leitfähiges Bauteil, das im Betrieb unter Spannung steht.
analog	Wert – z. B. einer Spannung – der sich stufenlos proportional verhält. Bei analogen Signalen kann der Wert des Signals innerhalb bestimmter Grenzen jeden beliebigen Wert annehmen.
Arbitrierung	Bei CANopen benutztes Buszuleitungsverfahren.
Autobaud-Erkennung	Automatische Erkennung der Kommunikationsgeschwindigkeit an einem Bussystem, sobald sich mindestens zwei Teilnehmer unterhalten oder ein Teilnehmer Nachrichten auf den Kommunikationsbus sendet.
Automatisierungsgerät	Gerät zur Steuerung mit Ein- und Ausgängen, das an einen technischen Prozess angeschlossen wird. Speicherprogrammierbare Steuerungen (SPS) sind eine spezielle Gruppe von Automatisierungsgeräten.

Basic-CAN	Konzept für die Implementierung eines CAN-Controllers. Zwischenspeicherung aller CAN-Nachrichten in einem Sende- und einem Empfangspuffer, d. h. hohe Anforderungen an den Host Controller, der alle Nachrichten auswerten muss.
Baud	Maß für die Übertragungsgeschwindigkeit von Daten. Ein Baud entspricht der Übertragung von einem Bit pro Sekunde (Bit/s).
Baud-Rate	Maßeinheit für die Geschwindigkeit bei der Datenübertragung in Bit/s.
Betriebsmittel, elektrische	Alle Gegenstände, die für die Erzeugung, Umwandlung, Übertragung, Verteilung und Anwendung von elektrischer Energie eingesetzt werden, z. B. Leitungen, Kabel, Maschinen, Steuergeräte.
Bezugserde	Potenzial des Erdreichs im Bereich von Erdungseinrichtungen. Kann im Gegensatz zur „Erde", deren Potenzial immer Null ist, ein von Null verschiedenes Potenzial haben.
Bezugspotenzial	Potenzial, von dem aus die Spannungen aller angeschlossenen Stromkreise betrachtet und/oder gemessen werden.
bidirektional	In beiden Richtungen arbeitend.
Bit	Abkürzung für englisch "binary digit". Kleinste Informationseinheit eines Binär-Systems, deren Wertigkeit 1 oder 0 sein kann (Ja/Nein-Entscheidung).
Bit-Stuffing	Im CAN genutztes Codierverfahren, bei dem nach einer Folge von fünf Bits gleicher Polarität ein sogenanntes "Stuffbit" mit entgegengesetzter Polarität in die laufende Nachricht eingefügt wird.
Blitzschutz	Alle Maßnahmen, die dazu dienen, ein System vor Schäden durch Überspannungen zu schützen, die von Blitzen hervorgerufen werden können.
Bridge	Die Bridge verbindet das CANopen-Netzwerk mit den Elektronikmodulen, die die Teilnehmer des Netzwerkes darstellen.

Bus	Sammelleitungssystem für den Datenaustausch, z. B. zwischen CPU, Speicher und I/O-Ebene. Ein Bus kann aus mehreren parallelen Leitungen für Datenübertragung, Adressierung, Steuerung und Stromversorgung bestehen.
Buslinie	Kleinste mit einem Bus verbundene Einheit bestehend aus einer SPS, einem Kopplungselement für Module an den Bus und einem Modul.
Bussystem	Die Gesamtheit aller Einheiten, die über einen Bus miteinander kommunizieren.
Buszykluszeit	Zeitintervall, in dem ein Master alle Slaves bzw. Teilnehmer in einem Bussystem bedient, d. h. deren Ausgänge schreibt und Eingänge liest.
Byte	Folge von 8 Bits
CAL	CAN Application Layer. Vom CiA als CiA-DS-201 bis -207 standardisiertes Schicht-7-Protokoll.
CAN	Controller Area Network
CAN 2.0A	11 Bit Identifer
CAN 2.0B	29 Bit Identifer
CAN Highspeed	Bis zu 1 MBd, Normal 500 kBd
CAN Knoten	Im CAN werden die Netzwerkteilnehmer auch als CAN-Knoten bezeichnet.
CAN Lowspeed	maximal 250 kBd
CAN Transceiver	Im CAN werden die Controller über eine ISO/DIS 11898 spezifizierte Anschaltung mit dem Busmedium verbunden. Diese Anschaltung wird im allgemeinen nicht diskret aufgebaut, sondern in Form eines CAN Transceiver Bausteines eingesetzt
CANopen	Auf CAL aufbauende Profilfamilie für den schnellen Datenaustausch. Das Kommunikationsprofil ist vom CiA als CiA-DS-301 genormt.
Change of State	bei CAN: Geber sendet automatisch seine Daten, sobald die Position verändert wird.
CiA	CiA e. V./CAN in Automation. Internationale CAN-Hersteller- und Anwender-Organisation.

CiA DS	CAN in Automation Draft Standard, Kommunikationsprofil
CiA DSP	CAN in Automation Draft Standard Proposal, Standardentwurfsvorschlag
CMS	CAN Based Message Specification. Einer der Dienste des Application Layer im CAN Reference Model.
COB	Communication Object/CAN Message. Eine Nachricht im CAN-Netzwerk. Alle über CAN zu versendenden Daten werden in COBs transportiert.
COB-ID	COB-Identifier. Identifiziert ein COB eindeutig im gesamten CAN-Netzwerk. Der COB-ID bestimmt die Priorität des COB für die Buszuteilung.
Code	Format der Datenübertragung
Communication Profile	hier: CANopen-Kommunikationsprofil. Beschreibung im CiA Draft Standard CiA-DS-301.
Const	Es handelt sich um ein konstantes Objekt. Der Wert kann nur gelesen werden und ändert sich nicht während des Betriebs. Beispiel: Device Software Version.
CPU	Abkürzung für englisch "Central Processing Unit". Zentrale Einheit zur Datenverarbeitung, das Kernstück eines Rechners.
CRC	Cyclic Redundancy Check: Im CAN verwendetes Datensicherungsverfahren mit geringer Restfehlerwahrscheinlichkeit, das auch in anderen Bereichen der Datenübertragung verwendet wird.
CSA-Zulassung	Kanadische Zulassung (Canadian Standards Association)
CSMA	Carrier Sense Multiple Access. Bei CAN verwendetes Buszugriffsverfahren, bei dem jeder Netzwerkteilnehmer selbstständig auf den Bus zugreifen kann, sobald dieser frei ist.
Data Frame, Datentelegramm	CAN-Nachricht, die zur Datenübertragung von einem Sender zu mehreren Empfängern gesendet wird.
Datenanforderungstelegramm	CAN-Nachricht, die von einem Netzwerkteilnehmer zur Datenanforderung an einen anderen Teilnehmer gesendet wird.

DBT	Distributor. Einer der Dienste des Application Layer im CAN Reference Model. Er dient zur Konfiguration der Layer im CAN Reference Model. Die Zuordnung von COB-IDs zu den vom CMS benutzten COBs ist Aufgabe des DBT.
DBT-Master	Spezieller CAN-Knoten, der die Aufgabe und Verwaltung der COB-IDs in einem CAL oder CANopen-Netzwerk übernimmt.
DBT-Slave	Alle CAN-Knoten, die sich von einem DBT-Master COB-IDs zuteilen lassen.
Device Profile	hier: CANopen-Geräteprofil. Beschrieben in den CiA Draft Standards CiA-DS-401 ff.
digital	Wert – z. B. einer Spannung – der innerhalb einer endlichen Menge nur bestimmte Zustände annehmen kann, meist definiert als „0“ und „1“.
DIN	Abkürzung für „Deutsches Institut für Normungen e. V.“.
Download	Herunterladen von Konfigurationsdaten, Parametern oder Programmen in einen CAN-Knoten.
Dual Code	Natürlicher Binärcode, häufig verwendeter Code bei absoluten Messsystemen.
EDS	Electronical Data Sheet: Datei mit der gerätespezifischen Parameterbeschreibung (wird vom Hersteller eines CANopen oder DeviceNet-Gerätes bereitgestellt)
EEPROM	Abkürzung für englisch “Electrically Erasable Programmable Read-Only Memory“: Elektrisch löschbarer, programmierbarer und lesbarer Speicher.
EIA	Abkürzung für englisch “Electronic Industries Association“, Vereinigung von Unternehmen der elektronischen Industrie in den USA.
EMV	Abkürzung für „Elektromagnetische Verträglichkeit“. Die Fähigkeit eines elektrischen Betriebsmittels, in einer bestimmten Umgebung fehlerfrei zu funktionieren, ohne negativen Einfluss auf die Umgebung zu haben.
EN	Abkürzung für „Europäische Norm“.

Erde	In der Elektrotechnik die Bezeichnung für leitfähiges Erdreich, dessen elektrisches Potenzial an jedem Punkt gleich Null ist. In der Umgebung von Erdungseinrichtungen kann das elektrische Potenzial der Erde ungleich Null sein, dann spricht man von „Bezugserde“.
erden	Verbinden eines elektrisch leitfähigen Teils über eine Erdungseinrichtung mit dem Erder.
Erder	Eine oder mehrere Komponenten, die mit dem Erdreich direkten und guten Kontakt haben.
ESD	Abkürzung für englisch “Electro Static Discharge“, elektrostatische Entladung.
Fault Mode	Legt fest, wie ein Ausgang im Fehlerfall reagieren soll. Ist das Bit für den Ausgang gesetzt, nimmt der Ausgang den Wert an, der in seinem Fault-State-Parameter steht.
Feldbus	Datennetz auf der Sensor-/Aktorebene. Ein Feldbus verbindet die Geräte in der Feldebene. Kennzeichnend für einen Feldbus sind hohe Übertragungssicherheit und Echtzeitverhalten.
Feldeinspeisung	Einspeisung der Spannung zur Versorgung der Feldgeräte sowie der Signalspannung.
galvanische Kopplung	Eine galvanische Kopplung tritt generell auf, wenn zwei Stromkreise eine gemeinsame Leitung benutzen. Typische Störquellen sind z. B. anlaufende Motoren, statische Entladungen, getaktete Geräte und ein unterschiedliches Potenzial der Gehäuse von Komponenten und der gemeinsamen Stromversorgung.
GND	Abkürzung für englisch “GROUND“, deutsch Masse (Potenzial 0).
Geräteprofil	hier: CANopen-Geräteprofil. Beschrieben in den CiA Draft Standards CiA-DS-401 ff.
Guard Identifier	Identifier für das Guarding-Protokoll. Mit diesem Identifier wird die Knotenüberwachung durchgeführt. Dazu wird der NMT-Master die zu überwachenden NMT-Slaves mit einem Datenanforderungstelegramm auffordern, ihren aktuellen Zustand zu melden.

Guard Time	Knotenüberwachungszeit. Parametrierbare Zeit, die zur Überwachung der CAN-Knoten herangezogen wird. Nach dieser Guard Time wird der NMT-Master den jeweiligen NMT-Slave über ein Datenanforderungstelegramm mit Guard Identifier auffordern, seinen aktuellen Zustand zu melden.
Guarding	Knotenüberwachung. Wird mit Hilfe des Guarding Protokolls durchgeführt.
hexadezimal	Zahlensystem mit der Basis 16. Gezählt wird von 0 bis 9 und weiter mit den Buchstaben A, B, C, D, E und F.
I/O	Abkürzung für englisch "Input/Output", Eingabe/Ausgabe.
Identifier	Nachrichtenkennung. Bei Standard CAN werden 11-Bit- und bei Extended CAN 29-Bit-Identifier verwendet.
Impedanz	Scheinwiderstand, den ein Bauelement oder eine Schaltung aus mehreren Bauelementen für einen Wechselstrom einer bestimmten Frequenz besitzt.
impedanzarme Verbindung	Verbindung mit geringem Wechselstromwiderstand.
inaktive Metallteile	Nicht berührbare leitfähige Elemente, die von den aktiven Metallteilen durch eine Isolierung elektrisch getrennt sind, im Fehlerfall jedoch Spannung annehmen können.
Index	Der Index und (bei Arrays sowie Records) der Subindex geben die CANopen-konforme Adresse eines Objekts an. Diese Adresse ist ein Index im Objektverzeichnis. Bei einfachen Variablen wird nur ein Index angegeben. Array-Strukturen haben Subindizes, die der Schreibweise als Nachkommazahl an den Index angehängt werden. Beispiel: [1800,01] = Index 1800, Subindex 1.
induktive Kopplung	Eine induktive (magnetische) Kopplung tritt zwischen zwei stromdurchflossenen Leitern auf. Die durch die Ströme hervorgerufene magnetische Wirkung induziert eine Störspannung. Typische Störquellen sind z. B. Transformatoren, Motoren, parallel laufende Netzkabel und HF-Signalkabel.
Inhibit Time	Zeitraum, innerhalb dessen ein PDO nicht nochmals gesendet werden darf. Dies dient der Vermeidung zu hoher Netzwerklasten.

kapazitive Kopplung	Eine kapazitive (elektrische) Kopplung tritt zwischen Leitern auf, die sich auf unterschiedlichen Potenzialen befinden. Typische Störquellen sind z. B. parallel verlaufende Signalkabel, Schütze und statische Entladungen.
Knoten	Netzwerkteilnehmer.
Kodierelement	Zweiteiliges Element zur eindeutigen Zuordnung von Elektronik- und Basismodul.
kommandofähige Module	Kommandofähige Module sind Module mit internem Speichersatz, die in der Lage sind, bestimmte Befehle (z. B. Ersatzwerte auszugeben) auszuführen.
Konfigurieren	Systematisches Anordnen der I/O-Module einer Station.
kurzschlussfest	Eigenschaft von elektrischen Betriebsmitteln. Ein kurzschlussfestes Betriebsmittel hält den thermischen und dynamischen Belastungen, die an seinem Installationsort aufgrund eines Kurzschlusses auftreten können, stand.
Life Time	Lebensdauer/Knotenüberwachungszeit. Parametrierbare Zeit, die zur Überwachung der CAN-Knoten herangezogen wird. Innerhalb der Life-Time erwartet der zu überwachende CAN-Knoten mindestens eine Guarding-Message.
LMT	Layer Management. Einer der Dienste im CAN Application Layer (CAL) im CAN Reference Model. Beschrieben im CiA Draft Standard CiA-DS-205. Er beinhaltet sogenannte schichtenspezifische Managementfunktionen. Das sind hier insbesondere die Modulnamen und Modul-ID sowie die Timing-Parameter der physikalischen Übertragungsschicht, damit also Baudrate der CAN-Knoten.
LMT-Master	Im LMT-Modell hat dieser CAN-Knoten die Aufgabe, die LMT-Parameter der anderen CAN-Knoten zu konfigurieren.
LMT-Slave	CAN-Knoten, der im LMT-Modell mit dem LMT-Master in Form einer Master-Slave-Beziehung kommuniziert.
LSB	Abkürzung für englisch "Least Significant Bit". Bit mit dem niedrigsten Stellenwert.

Mapping	Gesamtheit der Verbindungsdaten, also der Zuordnungen von Netzwerkvariablen an PDOs. Ein PDO kann eine oder mehrere Netzwerkvariablen übertragen (siehe CiA DS-301). Die Zuordnung von Variablen zu PDOs ist über Mapping-Tabellen definiert. Diese sind über das Objektverzeichnis adressierbar.
Masse	Gesamtheit aller untereinander verbundenen inaktiven Teile eines Betriebsmittels, die auch im Fehlerfall keine Berührungsspannung annehmen.
Masseband	Flexibler Leiter, meist geflochten, der die inaktiven Teile eines Betriebsmittels verbindet, z. B. die Tür eines Schaltschranks mit dem Schaltschrankkorpus.
Master	Station oder Teilnehmer in einem Bussystem, die/der die Kommunikation zwischen den anderen Teilnehmern des Bussystems steuert.
Master-Slave Mode	Betriebsart, bei der eine Station oder ein Teilnehmer im System als Master die Kommunikation über den Bus leitet.
Mode	englisch, deutsch Betriebsart (Modus).
Modulbus	Der Modulbus ist der interne Bus einer XI/ON-Station. Über ihn kommunizieren die XI/ON-Module mit dem Gateway. Er ist unabhängig vom Feldbus.
MSB	Abkürzung für englisch "Most Significant Bit". Bit mit dem höchsten Stellenwert.
Multimaster Mode	Betriebsart, bei der alle Stationen oder Teilnehmer im System gleichberechtigt über den Bus kommunizieren können.
Namur	„Normen-Arbeitsgemeinschaft für Mess- und Regeltechnik". Namur-Initiatoren sind Sonderausführungen der Zweidrahtinitiatoren. Aufgrund der besonderen Konstruktion – niedriger Innenwiderstand, wenige Bauteile, kurze Bauform – zeichnen sich Namur-Initiatoren durch eine hohe Stör- und Betriebssicherheit aus.
NMT	Network-Management. Einer der Dienste des Application Layer im CAN Reference Model. Er dient zum Initialisieren, Konfigurieren und zur Fehlerbehandlung im CAN-Netzwerk.

NV-Speicher	Non volotile: Elektronischer, nichtflüchtiger Speicher für elektronische Zähler, zur Werterhaltung bei Unterbrechung der Versorgungsspannung
Object Dictionary	Objekt-Verzeichnis. In dem Objekt-Verzeichnis sind alle über das Netzwerk zugänglichen Objekte in einer festgelegten Reihenfolge abgelegt. Der Zugriff auf diese Objekte erfolgt über einen 16-Bit-Index.
Operational	Aktiver Zustand eines CANopen-Knotens. In diesem Zustand kann der Knoten je nach Typ und Konfiguration PDOs senden und empfangen. SDO-Kommunikation ist weiterhin möglich.
Overhead	Systemverwaltungszeit, die bei jedem Übertragungszyklus einmal im System benötigt wird.
Parametrieren	Festlegen von Parametern der einzelnen Busteilnehmer bzw. ihrer Module in der Konfigurationssoftware des CANopen-Masters.
PDO	Process Data Object. Objekt für den Datenaustausch zwischen verschiedenen CAN-Teilnehmern.
Pollmodus	Slave sendet Daten erst nach Aufforderung durch den Busmaster.
Potenzialausgleich	Die Angleichung der elektrischen Niveaus der Körper elektrischer Betriebsmittel und fremder, leitfähiger Körper durch eine elektrische Verbindung.
potenzialfrei	Galvanische Trennung der Bezugspotenziale von Steuer- und Laststromkreisen bei I/O-Modulen.
potenzialgebunden	Elektrische Verbindung der Bezugspotenziale von Steuer- und Laststromkreisen bei I/O-Modulen.
Pre-Operational	Zustand eines CANopen-Knotens wie EASY221-CO nach Spannungswiederkehr und Vollendung der automatischen Initialisierung. Der Knoten kann per SDO angesprochen werden. Ferner kann von diesem Zustand aus in den Zustand "Operational" geschaltet werden.

Prioritäten	Die Identifier der CAN-Nachrichten bestimmen gleichzeitig die Priorität bei der Busvergabe. Damit ist es möglich, den Nachrichten entsprechend ihrer Wichtigkeit einen schnellen Buszugriff zu ermöglichen.
Reaktionszeit	In einem Bussystem das Zeitintervall zwischen dem Absenden eines Leseauftrags und dem Erhalt einer Antwort. Innerhalb eines Eingabemoduls das Zeitintervall von der Signaländerung am Eingang des Moduls bis zur Ausgabe derselben an das Bussystem.
Repeater	Verstärker für die über einen Bus übertragenen Signale.
RO	Read Only. Dieses Objekt kann nur gelesen werden.
RW	Read Write. Dieses Objekt kann gelesen und geschrieben werden.
RWR	Read Write Read. Dieses Objekt kann gelesen und geschrieben werden. Bei der Übertragung über PDOs (als Netzwerkvariable) kann es aber nur gelesen werden.
RWW	Read Write Write. Dieses Objekt kann gelesen und geschrieben werden. Bei der Übertragung über PDOs (als Netzwerkvariable) kann es aber nur geschrieben werden. Dies entspricht zum Beispiel einem Digital Output, der normalerweise geschrieben wird, aber zusätzlich (über SDO) erlaubt, den zuletzt geschriebenen Wert zurückzulesen.
Schirm	Bezeichnung für die leitfähige Hülle von Leitungen, Gehäusen und Schränken.
Schirmung	Gesamtheit der Maßnahmen und Betriebsmittel, die zur Verbindung von Anlagenteilen mit dem Schirm dienen.
Schutzleiter	Ein für den Schutz gegen gefährliche Körperströme notwendiger Leiter, dargestellt durch das Kürzel PE (Abkürzung für englisch "Protective Earth").
SDO	Service Data Object. Objekt für eine Punkt-zu-Punkt-Kommunikation mit Zugriff auf das Object Dictionary eines CAN-Teilnehmers.

SDO-Manager	CANopen-Manager/-Master, der per SDO auf die Geräte zugreifen kann. Von diesen können in komplexen oder großen Anlagen mehrere vorhanden sein (z. B. zur Aufgabenteilung).
seriell	Bezeichnung für eine Art der Informationsübertragung, bei der die Daten nacheinander – Bit für Bit – über eine Leitung übertragen werden.
Slave	Station oder Teilnehmer in einem Bussystem, die/der dem Master untergeordnet ist/sind.
SPS	Abkürzung für Speicherprogrammierbare Steuerung.
Station	Funktionseinheit oder Baugruppe, bestehend aus mehreren Elementen.
Störaussendung (EMV)	Prüfverfahren nach EN 61000-6-4
Störfestigkeit (EMV)	Prüfverfahren nach EN 61000-6-2
Strahlungskopplung	Eine Strahlungskopplung tritt auf, wenn eine elektromagnetische Welle auf eine Leiterstruktur trifft. Durch das Auftreffen der Welle werden Ströme und Spannungen induziert. Typische Störquellen sind z. B. Funkenstrecken (Zündkerzen, Kollektoren von Elektromotoren) und Sender (z. B. Funkgeräte), die nahe bei der entsprechenden Leiterstruktur betrieben werden.
Subindex	Siehe Index.
Sync	Das SYNC-Objekt ist eine Botschaft, die von einer Station periodisch gesendet wird. Mit ihrer Hilfe können Gerätedaten mit einem zeitlich definierten Abstand übertragen werden. PDOs, die hierauf reagieren sollen, tragen das Attribut TransmissionType synchron (siehe Transmission Type).
Topologie	Geometrischer Aufbau eines Netzes bzw. Anordnung der Schaltungen.
Transmission Type	Sendeverhalten eines PDO.

UART	Abkürzung für englisch "Universal Asynchronous Receiver/ Transmitter", deutsch universeller asynchroner Empfänger/ Sender. Ein „UART" ist ein Logikschaltkreis, der zur Umwandlung einer asynchronen seriellen Datenfolge in eine bitparallele Datenfolge oder umgekehrt eingesetzt wird.
unidirektional	In einer Richtung arbeitend.
WO	Write Only. Dieses Objekt kann nurgeschrieben werden.

Stichwortverzeichnis

A
Abbilddaten
Allgemeine Informationen 91
Übersicht easy600 91
Übersicht easy700 134
Übersicht easy800/MFD 182
Abmessungen 286
Abschlusswiderstände 22
Adressbereich 26
Adresse einstellen
am Basisgerät 26
mit easySoft 28
über spezielle Konfigurationstools 29
ALT-Taste, Zustand lesen 94
Analog-Ausgang
easy800/MFD (Zustand lesen) 195
Analog-Eingänge
easy600 (Zustand lesen) 95
easy700 (Zustand lesen) 139
easy800/MFD (Zustand lesen) 183
Analogwertvergleicher
easy600 (Istwerte schreiben) 103
easy700 156
easy700 (Zustand lesen) 135
easy800/MFD 204
Anschließen
CANopen 21
EASY221-CO an Basisgerät 19
Versorgungsspannung 20
Anschlussbelegung CANopen 21
Arithmetikbaustein
easy800/MFD 206
Aufbau des Gerätes 15
Ausgänge S1 bis S8, Zustand lesen 82
Ausgänge von easyLink
easy700 (Zustand lesen) 149
easy800/MFD (Zustand lesen) 198

Ausgänge, lokale und Netzwerkteilnehmer
easy800/MFD (Zustand lesen) 196
Ausgangsdaten, Definition 77
Ausschaltzeitpunkt
lesen 123
schreiben 127

B
Basisgerät anschließen 19
Baudratenerkennung, automatisch 24
Bedientasten
easy600 (Zustand lesen) 92
Bedingter Sprung
easy800/MFD 244
Betriebsart
lesen 82
schreiben 78
Betriebsmodus, ungültig 173, 278
Betriebsstundenzähler
easy700 162
easy800/MFD 254
Betriebssystem-Voraussetzungen 17
Bitfeld 132
Block Compare, easy800/MFD 208
Block Transfer, easy800/MFD 210
Boolesche Verknüpfung
easy800/MFD 212

C
CANopen
Anschlussbelegung 21

D
Datenaustausch, PDO 77
Datenbaustein, easy800/MFD 227
Datenmultiplexer, easy800/MFD 250
Datentypen 8
Datum lesen/schreiben
easy600 87
easy700 ..... 130, 177
DEL-Taste, Zustand lesen 94

Diagnose lokal, easy800/MFD (Abbilddaten) 185
Diagnose Remote-Teilnehmer, easy800/MFD (Abbilddaten) 185
Diagnose, easy800/MFD 232
Digital-Ausgänge
easy600 (Zustand lesen) 99
easy700 (Zustand lesen) 148
easy800/MFD (Zustand lesen) 196
Digital-Eingänge
easy600 (Zustand lesen) 92
easy700 (Zustand lesen) 138
easy800/MFD (Zustand lesen) 187
Direkter Datenaustausch 77

E
EDS-Datei 31
Eingänge R1 bis R8 setzen/rücksetzen 80
Eingänge von easyLink
easy700 (Zustand lesen) 149
easy800/MFD (Zustand lesen) 198
Eingänge, Netzwerkteilnehmer
easy800/MFD (Zustand lesen) 189
Eingangsdaten, Definition 77
Einschalten, erstes 25
Einschaltzeitpunkt
lesen 123
schreiben 127
Emergency 55
-Objekt 37
-Protokoll 75
Empfang-PDO 34
Empfangsdaten, Netzwerkteilnehmer
easy800/MFD 200, 236
EMV 22
Error-LED 29, 282
Erstes Einschalten 25
ESC-Taste, Zustand lesen 94
Event Timed PDOs 35

F
Fehlercodes, über easyLink
easy700 173
easy800/MFD 278
Fehlermeldung 55
EASY221-CO an Netzwerk 37
Frequenzzähler
easy800/MFDr 216
Funktionsbausteine, Übersicht
easy600 102
easy700 155
easy800/MFD 202

G
Geräteprofil 44

H
Hardware-Voraussetzungen 17
Heartbeat-Producer 43
Herstellerspezifische Objekte 54
Hilfsrelais
easy600 (Zustand lesen) 99

I
Impulsausgabe
easy800/MFD 256
Inbetriebnahme 25
Inhibit Time 35
Initialisation 39
Inkrementalzähler
easy800/MFD 220

J
Jahresschaltuhr
easy700 167
easy700 (Zustand lesen) 152
Jahreszeitschaltuhr
easy800/MFD 241

K
Knotennummer einstellen 44
Knotenüberwachung 42
Kommunikationsobjekte 33
Kommunikationsparameter 45
Kommunikationsstatus 43

L
Layer Setting Service 44
LED-Anzeigen 29
Lokale Ausgänge
easy700 (Zustand lesen) 148
Lokale Eingänge
easy700 (Zustand lesen) 138
easy800/MFD (Zustand lesen) 187
Lokaler Analog-Ausgang
easy800/MFD (Zustand lesen) 195

M
Masterreset
easy700 153
easy800/MFD 248
Merker
easy600 (Zustand lesen) 99
easy800/MFD(Zustand lesen) 190
Minute
easy600 89
lesen 123
schreiben 127
Multiplexed PDOs 35

N
Netzdaten empfangen
easy800/MFD 236
Netzdaten senden
easy800/MFD 259
Netzwerkmanagement, CANopen 38
Netzwerkteilnehmer, Zustand der Eingänge lesen 189
NMT Service 41
Node-Guarding 42

O
Objektverzeichnis 45
OK-Taste, Zustand lesen 94
Operational 39

P
PDO 77
-Mapping 34, 36
-Protokoll 59
PID-Regler
easy800/MFD 229
Potentialtrennungen 23
Pre-Operational 39
Prepared 40
Protokolle, CANopen 59
Prozessdatenobjekte 34
P-Tasten
easy600(Zustand lesen) 92
easy700 (Zustand lesen) 146
easy800 (Zustand lesen) 193
Pulsweitenmodulation
easy800/MFD 261

R
Reaktionszeit Basisgerät 31
Receive-Data, Netzwerkteilnehmer
easy800/MFD (Zustand lesen) 200
Rücksetzen, easy/MFD-Ein-/Ausgänge 78
RUN-LED 30, 281

S
Sachwidriger Einsatz 18
Schaltregel 132
Schaltzeitpunkte
lesen 123
schreiben 127
Schieberegister
easy800/MFD 266
Schneller Zähler
easy800/MFD 218
Schwellwertschalter
easy700 156

Schwellwertvergleicher
easy700 (Zustand lesen) 135
SDO
Steuerbefehle für easy600 85
Steuerbefehle für easy700 129
Steuerbefehle für easy800 175
Send-Data, Netzwerkteilnehmer
easy800/MFD (Zustand lesen) 200
Sendedaten, Netzwerkteilnehmer
easy800/MFD 259
easy800/MFD (Zustand lesen) 200
Sende-PDO 34
Serielle Ausgabe
easy800/MFD 264
Servicedatenobjekte 33
Setzen, easy/MFD-Ein-/Ausgänge 78
Signalglättungsfilter
easy800/MFD 234
Sollzykluszeit
easy800/MFD 269
Sommerzeit
easy600 87
easy700 131
easy800/MFD 178
Speichern, Objektverzeichniseinträge 43
Statusanzeige (LED) 29
Steuerbefehle
easy600 85
easy700 129
easy800 175
Stunde
easy600 88
lesen 123
schreiben 127
synchrone Übertragung 37
Synchronisations-Objekt 37
Systemdienste 37
Systemübersicht, CANopen 14

T
Tabellenfunktion
easy800/MFD 274
Tastereingänge P1 bis P4
easy600 92
Teilnehmeradresse einstellen 26
Telegramm, ungültig 173, 278
Textanzeige
easy600 (Zustand lesen) 99
Textausgabe-Baustein
easy800/MFD 224
Textbausteine
easy700 (Zustand lesen) 137
Textmerker 101
Time Stamp-Objekt 37

U
Übertragungsarten, der PDOs 35
Übertragungsraten 24
Uhr synchronisieren
easy800/MFD 263
Uhrzeit lesen/schreiben
easy600 87
easy700 ..... 130, 177
ungültiger Betriebsmodus 173, 278
ungültiges Telegramm 173, 278

V
Verdrahtung, EMV-gerecht 22
Vergleicher
easy800/MFD 222
Vergleichswert schreiben (Analogwert-Vergleicher) . 104
Versionsgeschichte, easy800 176
Versorgungsspannung anschließen 20
Verzögerungszeit, zufälliger Wert 114

W Wertbegrenzung
easy800/MFD 276
Wertskalierung
easy800/MFD 246
Wiederherstellen, Objektverzeichniseinträge 43
Winterzeit
easy600 87
easy700 132
easy800/MFD 178
Wochenschaltuhr
easy700 .....154, 170
easy800/MFD 238
Wochentag
esy600 88

Z Zahlenwandler
easy800/MFD 252
Zähler
easy700 (Zustand lesen) 136
easy800/MFD 214
Zählerrelais
easy600 (Istwert lesen) 106
easy600 (Sollwert lesen) 110
easy600 (Sollwert schreiben) 108
easy700 159
Zeitglieder
easy700 (Zustand lesen) 151
Zeitreferenz 37
Zeitrelais
easy600 (Istwert lesen) 112
easy600 (Parameter schreiben) 116
easy700 164
easy800/MFD 271
Zeitschaltuhr
easy600 (Istwerte lesen) 120
easy600 (Sollwerte schreiben) 124
Zeitumstellung (easy800/MFD) 178

Zustand
Ausgänge S1 bis S8 lesen 82
Eingänge R1 bis R8 schreiben 80
Zyklischer Datenaustausch 77
Zykluszeit 31