

EASY222-DN

DeviceNet-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 08/02
 2. Auflage 2004, Redaktionsdatum 10/04
 3. Auflage 2008, Redaktionsdatum 02/08
 4. Auflage 2010, Redaktionsdatum 09/10
- siehe Änderungsprotokoll im Kapitel „Zu diesem Handbuch“
© 2002by 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 elektro-nischer 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 abschränken.
- 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
	Gerätebezeichnung	8
	Abkürzungen und Symbole	9
	Lesekonventionen	10
<hr/>		
1	Zu EASY222-DN	11
	Systemübersicht	12
	Aufbau des Gerätes	13
	Kommunikationsprofil EASY222-DN	14
	Hardware und Betriebssystemvoraussetzungen	15
	Sachwidriger Einsatz	16
<hr/>		
2	Installation	17
	Anschluss EASY222-DN an das Grundgerät	17
	Versorgungsspannung anschließen	18
	DeviceNet anschließen	19
	– Anschlussbelegung DeviceNet	19
	– Abschlusswiderstände	20
	EMV-gerechte Verdrahtung	20
	Potentialtrennungen	21
	Übertragungsraten – automatische	
	Baudratenerkennung	22
	– Maximale Entfernungen und Busleitungslängen	22

3	Gerät betreiben	23
	Erstes Einschalten	23
	DeviceNet Teilnehmeradresse einstellen	24
	– Adresse mittels easySoft einstellen	26
	– Adresse über die Mastersteuerung einstellen	26
	LED-Statusanzeigen	27
	– Modul-Status-LED	27
	– Network Status LED	28
	Zykluszeit easy-Basisgerät	29
	EDS-Datei	29

4	DeviceNet-Funktionen	31
	Objektmodell	31
	– Identity Object	35
	– DeviceNet Object	37
	– easy Object	38
	DeviceNet-Kommunikationsprofil	41
	– I/O Messages	41
	– Explicit Messages	42

5	Direkter Datenaustausch mit easy/MFD (Polled I/O Connection)	47
	Eingangsdaten: Betriebsart, S1 – S8	49
	Ausgangsdaten: Betriebsart, R1 – R16	51

6 Steuerbefehle für easy600	55
Datum und Uhrzeit, Sommer-/Winterzeit lesen und schreiben	57
Abbilddaten lesen	61
– Generelles zum Arbeiten mit Abbilddaten	61
– Übersicht	61
– Digital-Eingänge, P-Tasten und Bedientasten	62
– Analog-Eingänge: I7 – I8	65
– Zeitrelais, Zählerrelais, Zeitschaltuhren, Analogwertvergleicher	66
– Hilfsrelais (Merker), Digital-Ausgänge, Textanzeige	69
Funktionsbausteine lesen/schreiben	72
– Übersicht	72
– Analogwertvergleicher A1 – A8: Istwerte schreiben (Funktion, Vergleichswerte)	73
– Zählerrelais C1 – C8: Istwert lesen	76
– Zählerrelais C1 – C8: Sollwert schreiben	78
– Zählerrelais C1 – C8: Sollwert lesen	80
– Zeitrelais T1 – T8: Istwert lesen (Zeitbereich, Istwert, Schaltfunktion)	82
– Zeitrelais T1 – T8: Parameter schreiben (Zeitbereich, Sollwert, Schaltfunktion)	86
– Zeitschaltuhr U1 – U4: Istwerte lesen (Kanal, ON-Zeit, OFF-Zeit)	90
– Zeitschaltuhr U1 – U4: Sollwerte schreiben (Kanal, ON-Zeit, OFF-Zeit)	94

7 Steuerbefehle für easy700	99
Datum und Uhrzeit lesen/schreiben	101
Abbilddaten lesen/schreiben	105
– Übersicht	105
– Analogwertvergleich/Schwellwertvergleich:	
A1 – A16	106
– Zähler: C1 – C16	107
– Textbausteine: D1 – D16	108
– Lokale Eingänge: I1 – I16	109
– Lokale Analog-Eingänge: IA1 – IA4	110
– Merker: M1 – M16/N1 – N16 schreiben	112
– Merker: M1 – M16/N1 – N16 lesen	114
– Betriebsstundenzähler: O1 – O4	116
– Lokale P-Tasten: P1 – P4	117
– Lokale Ausgänge: Q1 – Q8	119
– Ein-/Ausgänge von easyLink: R1 – R16/S1 – S8	120
– Zeitglieder: T1 – T16	122
– Jahresschaltuhr: Y1 – Y8	123
– Masterreset: Z1 – Z3	124
– Wochenschaltuhr: $\text{W}1$ – $\text{W}8$	125
Funktionsbaustein-Daten lesen/schreiben	126
– Allgemeine Hinweise	126
– Übersicht	126
– Analogwertvergleich/Schwellwertschalter: A1	
– A16	127
– Zählerrelais: C1 – C16	130
– Betriebsstundenzähler: O1 – O4	133
– Zeitrelais: T1 – T16	135
– Jahresschaltuhr: Y1 – Y8	138
– Wochenschaltuhr: $\text{W}1$ – $\text{W}8$	141
Analyse – Fehlercodes über easyLink	144

8 Steuerbefehle für easy800/MFD	145
Datenaustauschverfahren	145
Versionsgeschichte	147
Datum und Uhrzeit lesen/schreiben	148
– Winter-/Sommerzeit, Zeitumstellung	149
Abbilddaten lesen/schreiben	154
– Übersicht	154
– Lokale Analog-Eingänge: IA1 – IA4	155
– Lokale Diagnose: ID1 – ID16	157
– Lokale Eingänge: IW0	159
– Eingänge der Netzwerkteilnehmer: IW1 – IW8	161
– Merker: M...	162
– Lokale P-Tasten: P1 – P4	165
– Lokaler Analog-Ausgang: QA1	167
– Lokale Ausgänge: QW0/ Ausgänge der Netzwerkteilnehmer: QW1 – QW8	168
– Ein-/Ausgänge von easyLink: RW/SW	170
– Receive-Data Netz: RN1 – RN32/ Send-Data Netz: SN1 – SN32	172
Funktionsbaustein-Daten lesen/schreiben	174
– Allgemeine Hinweise	174
– Übersicht	175
– Analogwertvergleich: A01 – A32	177
– Arithmetikbaustein: AR01 – AR32	179
– Block Compare: BC01 – BC32	181
– Block Transfer: BT01 – BT32	183
– Boolesche Verknüpfung: BV01 – BV32	185
– Zähler: C01 – C32	187
– Frequenzzähler: CF01 – CF04	189
– Schneller Zähler: CH01 – CH04	191
– Inkrementalzähler: CI01 – CI02	193
– Vergleich: CP01 – CP32	195
– Textausgabe-Baustein: D01 – D32	197
– Datenbaustein: DB01 – DB32	200
– PID-Regler: DC01 – DC32	202
– Diagnose DG01...DG16	205
– Signalglättungsfilter: FT01 – FT32	207
– Empfang von Netzdaten: GT01 – GT32	209
– Wochenzeitschaltuhr: HW01 – HW32	211
– Jahreszeitschaltuhr: HY01 – HY32	214

– Bedingter Sprung JC01...JC32	217
– Wertskalierung: LS01 – LS32	219
– Masterreset: MR01 – MR32	221
– Datenmultiplexer MX01...MX32	223
– Zahlenwandler: NC01 – NC32	225
– Betriebsstundenzähler: OT01 – OT04	227
– Impulsausgabe PO01...PO02	229
– Senden von Netzdaten: PT01 – PT32	232
– Pulsweitenmodulation: PW01 – PW02	234
– Uhr synchronisieren: SC01	236
– Serielle Ausgabe SP01...SP32	237
– Schieberegister SR01...SR32	239
– Sollzykluszeit: ST01	242
– Zeitrelais: T01 – T32	244
– Tabellenfunktion TB01...TB32	247
– Wertbegrenzung: VC01 – VC32	249
Analyse – Fehlercodes über easyLink	251
<hr/>	
9 Was ist, wenn...?	253
<hr/>	
Anhang	255
Technische Daten	255
Abmessungen	258
EDS-Datei	259
<hr/>	
Glossar	263
<hr/>	
Stichwortverzeichnis	271

Zu diesem Handbuch

Änderungsprotokoll

Gegenüber der letzten Ausgabe haben sich folgende wesentliche Änderungen ergeben:

Redaktionsdatum	Seite	Stichwort	neu	Änderung	entfällt
10/04	allg.	easy700/800/MFD	✓		
02/08	Kapitel 8	Device-Net-Zugriff auf die geänderten/ neuen Funktionsbausteine der easy800 MFD-CP8/CP10 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 DeviceNet und zur Programmierung einer DeviceNet-Mastersteuerung vorausgesetzt. Außerdem sollten Sie mit der Handhabung des Steuerrelais easy bzw. des Multi-Funktions-Displays 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)
- „Multi-Funktions-Display 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/moeller> → Support als Suchbegriff „EASY222-DN“ ein.

Referenzen

- [1] DeviceNet Specification Volume I
Release 2.0, Errata 1 - 4
April 1, 2001
- [2] DeviceNet Specification Volume II
Release 2.0, Errata 1 - 4
April 1, 2001

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-...

- easy-AB für
 - EASY719-AB...

- easy-AC für
 - EASY6...-AC-RC(X)
 - EASY719AC
 - EASY8...-AC-...

- easy-DC für
 - EASY6...-DC-...
 - EASY719-DC-...
 - EASY8...-DC-...

- easy-DA für
 - EASY719-DA...

- MFD-CP8... für
 - MFD-CP8-ME
 - MFD-CP8-NT
 - MFD-AC-CP8-ME
 - MFD-AC-CP8-NT

- MFD-CP10... für
 - MFD-CP10-ME
 - MFD-CP10-NT

- MFD-...-CP8/CP10 für
 - MFD-CP8-ME
 - MFD-CP8-NT
 - MFD-AC-CP8-ME
 - MFD-AC-CP8-NT
 - MFD-CP10-ME
 - MFD-CP10-NT

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

bcd	Binär Codierter Dezimalcode
CAN	Controller Area Network
dec	Dezimal (Zahlensystem mit der Basis 10)
hex	Hexadezimal (Zahlensystem mit der Basis 16)
len	length (Länge)
MAC ID	Media Access Control Identifier
ODVA	Open DeviceNet Vendor Association
PC	Personal Computer
SELV	Safety Extra Low Voltage „Sicherheitskleinspannung“
UCMM	Unconnected Message Manager

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 Kapitelanfangseiten und leere Seiten am Kapitelende.

► zeigt Handlungsanweisungen an.



Achtung!

warnet vor leichten Sachschäden.



Warnung!

warnet vor schweren Sachschäden und leichten Verletzungen.



Gefahr!

warnet vor schweren Sachschäden und schweren Verletzungen oder Tod.



macht Sie aufmerksam auf interessante Tipps und Zusatzinformationen

1 Zu EASY222-DN

Das Kommunikationsmodul EASY222-DN wurde für Automatisierungsaufgaben mit dem Feldbus DeviceNet entwickelt. EASY222-DN 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 DeviceNet-Gateway EASY222-DN, agiert im Feldbussystem ausschließlich als Slave-Teilnehmer.

Systemübersicht

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

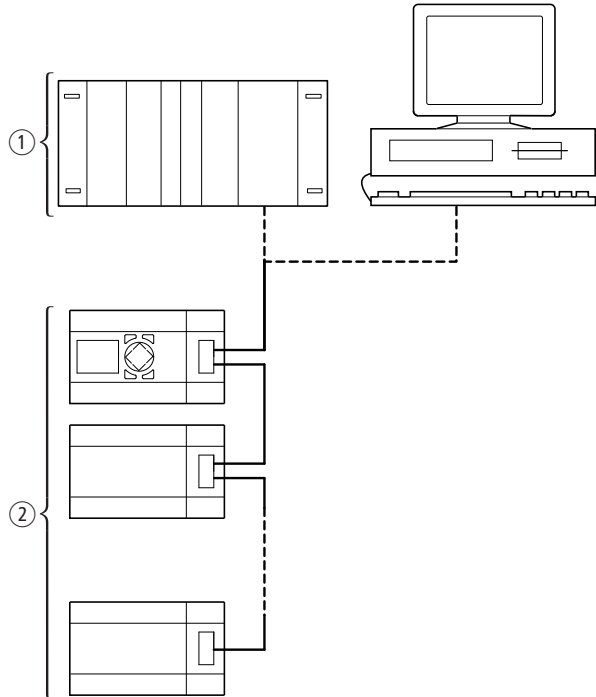


Abbildung 1: Einbindung EASY222-DN im DeviceNet

- ① Masterbereich, SPS (z. B.: SLC 500) oder PC mit CAN-Karte
- ② Slave-Bereich z. B.: Steuerrelais easy/MFD mit DeviceNet-Anschaltung

Aufbau des Gerätes

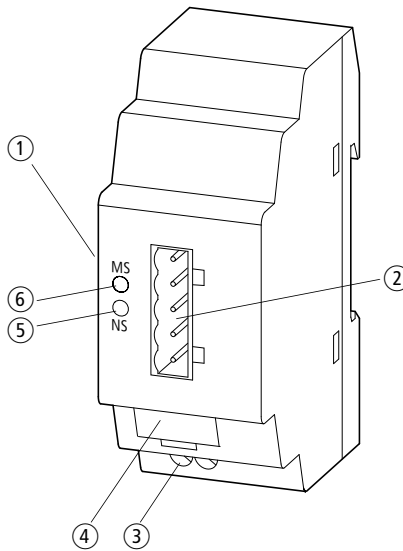


Abbildung 2: Aufbau EASY222-DN

- ① easyLink-Buchse
- ② 5-poliger DeviceNet-Anschluss nach ODVA
- ③ Spannungsversorgung 24 V $\overline{\text{---}}$
- ④ Gerätekennzeichnungsschild
- ⑤ Network-Status-LED NS
- ⑥ Modul-Status-LED MS

**Kommunikationsprofil
EASY222-DN**

- Vordefinierte Master/Slave Verbindungseinstellung
 - Die **I/O Poll**-Verbindung dient der Übertragung von 3 Byte Eingangsdaten (R1 bis R16) und 3 Byte Ausgangsdaten (S1 bis S8) zwischen easy-Grundgerät mit Gateway-Anbindung und DeviceNet-Steuerung.
 - Die **I/O Change of State/Cyclic**-Verbindungen (bestätigt, unbestätigt) dient zur Übertragung der 2 Byte Diagnosedaten vom Steuerrelais easy zur DeviceNet-Steuerung.
 - Der **Explizite Verbindungsaufbau** dient dem Lesen und Schreiben von Funktionsrelais-Parametern im Steuerrelais easy. Weiterhin unterstützt diese Art des Verbindungsaufbaus die Konfiguration, Diagnose und Management Zwecke des Steuerrelais.
- DeviceNet Kommunikation Adapter Profile (Device Type 12), das zusätzlich mit den easy-Anforderungen erweitert wurde
- Group 2 Server
- UCMM-fähiges Gerät
- Dynamischer Aufbau von Expliziten- und I/O Verbindungen möglich
- Device Heartbeat Message
- Device Shutdown Message
- Offline Verbindungseinstellungen

**Hardware und Betriebs-
systemvoraussetzungen**

Das Erweiterungsgerät EASY222-DN arbeitet mit den Basisgeräten easy600, easy700, easy800 und MFD ab den nachfolgenden Betriebssystemversionen zusammen:

Basisgerät		Erweiterungsgerät EASY222-DN	
Geräteversion	BTS-Version	Geräteversion = 01	Geräteversion \geq 02
easy600			
\geq 04	ab 2.4	×	×
easy700			
\geq 01	ab 1.01.xxx	–	×
easy800			
\geq 04	ab 1.10.xxx	–	×
MFD-CP8.../CP10...			
\geq 01	ab 1.10.xxx	–	×
MFD-CP10			
\geq 01	ab 1.00	–	×

Die Geräteversion des entsprechenden Basis- oder Erweiterungsgerätes ist auf der rechten Gehäusseite angegeben. Beispiel: EASY222-DN: 02-206xxxxxxx (02 = 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 147.

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 easy/MFD-Basisgeräte mit Erweiterung.

Anschluss EASY222-DN an das Grundgerät

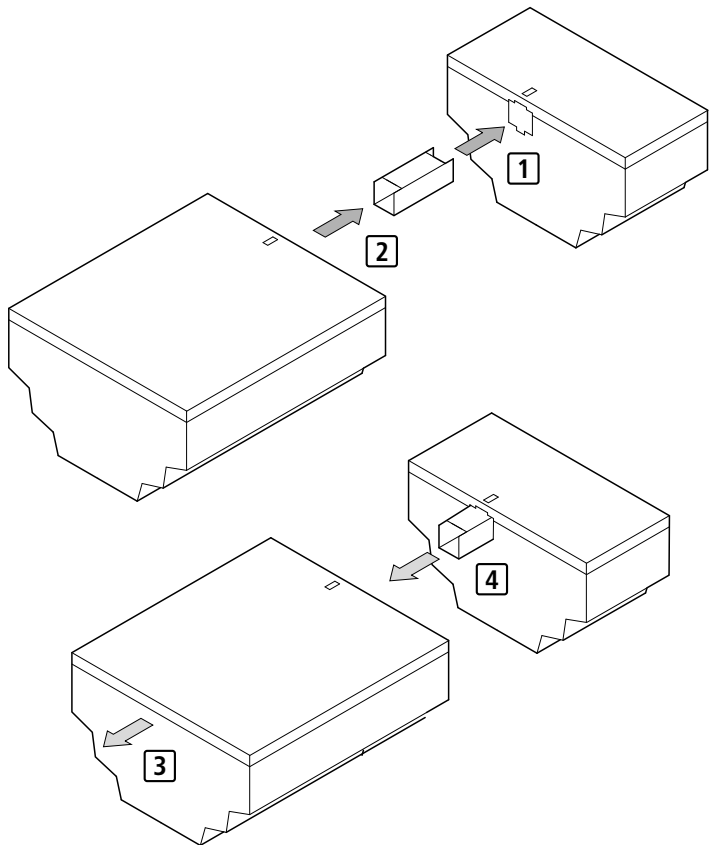


Abbildung 3: EASY222-DN an das Basisgerät montieren

1 + 2 Einbau

3 + 4 Ausbau

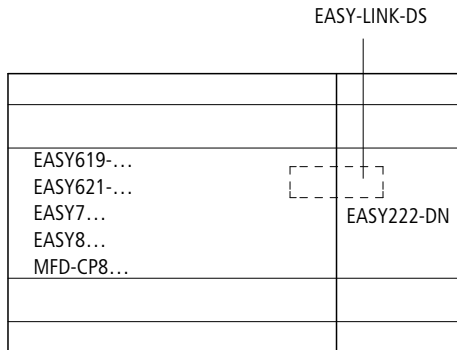


Abbildung 4: Verbindung zwischen Basisgerät und EASY222-DN

Versorgungsspannung anschließen

Das Gerät EASY222-DN wird mit einer 24-V-DC-Versorgungsspannung betrieben (→ Abschnitt „Stromversorgung“, Seite 257).



Gefahr!

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

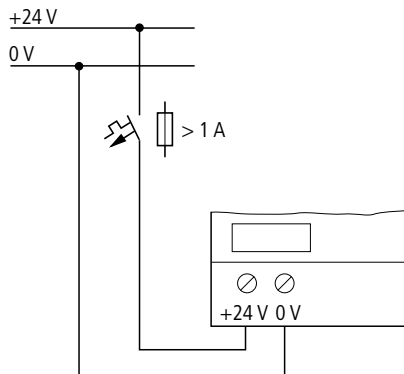


Abbildung 5: Versorgungsspannung EASY222-DN

DeviceNet anschließen

Ein 5-poliger DeviceNet-Stecker verbindet den DeviceNet-Anschluss des Gerätes mit dem Feldbus DeviceNet.

Zum Anschluss nehmen Sie bitte einen speziellen DeviceNet-Stecker und ein spezielles DeviceNet-Kabel. Beides ist von der ODVA genau spezifiziert worden. Die Art des Kabels hat Einfluss auf die verfügbare maximale Busleitungslänge und somit auch auf die Übertragungsrate.

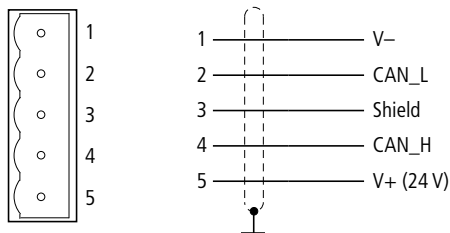
Anschlussbelegung DeviceNet

Abbildung 6: Anschlussbelegung Gerätebuchse

- | | | |
|---|--------|---------|
| 1 | GND | schwarz |
| 2 | CAN_L | blau |
| 3 | Schirm | blank |
| 4 | CAN_H | weiß |
| 5 | 24 V | rot |

Damit eine Kommunikation des EASY222-DN am Feldbus DeviceNet gewährleistet ist, müssen alle Pins des Steckers belegt werden. Dies gilt auch für die 24-V-Busspannung.



Ist die Busspannung nicht vorhanden, wird sich das Gateway auch nicht an der Kommunikation am Bus beteiligen. In diesem Fall ist die Network-Status-LED im OFF-Modus.

Abschlusswiderstände

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

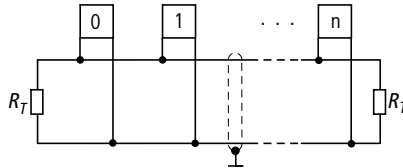


Abbildung 7: Abschlusswiderstände R_T : Anschluss CAN_H und CAN_L

$$R_T = 120\ \Omega$$

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 RJ45 und natürlich durch Auflegen des Schirms wesentlich verringern.

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

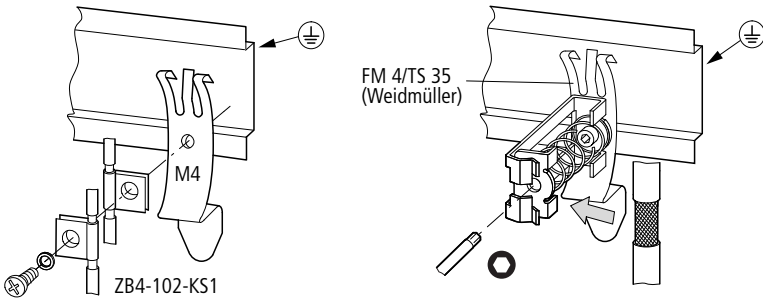


Abbildung 8: Schirmbefestigung auf Hutschiene

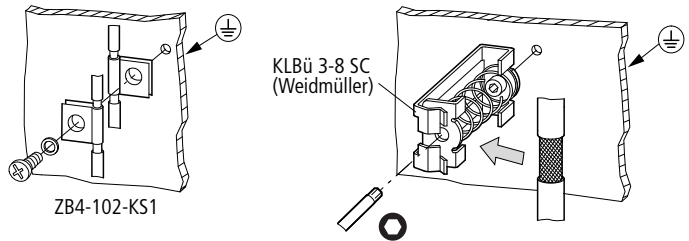


Abbildung 9: Schirmbefestigung auf Montageplatte

Potentialtrennungen

Für die Schnittstellen des EASY222-DN gelten folgende Potentialtrennungen:

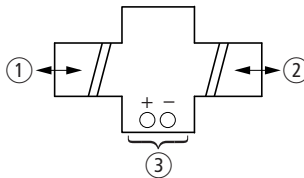


Abbildung 10: Potentialtrennung zwischen Versorgungsspannung und Ausgängen

- ① sichere elektrische Trennung zwischen easyLink und 240 V ~
- ② einfache elektrische Trennung zum DeviceNet-Kommunikationsbus
- ③ Versorgungsspannung 24 V DC

**Übertragungsraten –
automatische
Baudratenerkennung**

Nach dem Einschalten erkennt das Modul EASY222-DN automatisch die im Kommunikationsnetzwerk verwendete Übertragungsrate. Hierzu muss allerdings mindestens ein Teilnehmer im Netzwerk gültige Nachrichten senden. Das Gerät unterstützt die von der ODVA spezifizierten Übertragungsraten von:

- 125 kBit/s,
- 250 kBit/s,
- 500 kBit/s.

Maximale Entfernungen und Busleitungslängen

Die maximale Buslänge ist nicht nur von der Datenrate sondern auch vom verwendeten Kabel abhängig. Zulässig sind folgende Kabel:

- Ein sogenanntes dünnes („Thin Cable“),
- ein dickes („Thick Cable“)
- oder auch ein flaches („Flat Cable“).

Die Anforderungen an die aufgeführten Datenkabel sind von der ODVA spezifiziert.

Baudrate (kBit/s)	maximale Buslänge in m		
	„Thick Cable“	„Thin Cable“	„Flat Cable“
125	500	100	420
250	250	100	200
500	100	100	100

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 EASY222-DN ein.

Die LEDs des EASY222-DN flimmern. Das Gerät befindet sich im Modus zum Erkennen der richtigen Baudrate (→ Abschnitt „Übertragungsraten – automatische Baudratenerkennung“ auf Seite 22). Am Basisgerät wird die GW-Information (intelligenter Teilnehmer angekoppelt) angezeigt.

Basisgerät	Gerätrevision	GW-Anzeige
easy600	04	statisch
easy700	ab 01	blinkend
easy800	04	statisch
	ab 05	blinkend
MFD-CP8...	01	statisch
	ab 02	blinkend
MFD-CP10...	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 „Network Status LED“ auf Seite 28).

Hat EASY222-DN die Werkseinstellungen (Node ID = 127), so müssen Sie die DeviceNet-Teilnehmeradresse einstellen.

DeviceNet Teilnehmer- adresse einstellen

Jeder DeviceNet-Teilnehmer benötigt eine eindeutige Adresse (MAC ID) in der DeviceNet-Struktur. In einer DeviceNet-Struktur können Sie maximal 64 Adressen (0 bis 63) vergeben. Jede MAC ID darf in der gesamten Busstruktur nur einmal vergeben sein.

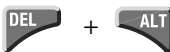
Die DeviceNet-Adresse können Sie bei EASY222-DN auf drei Arten einstellen:

- Mit der integrierten Anzeige und Tastatur am easy-Basisgerät
- 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 EASY222-DN 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.

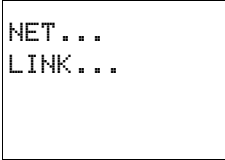
```
PASSWORT...
SYSTEM...
GB D F E I
KONFIGURATOR
```

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

- ▶ Wechseln Sie mit den Cursor-Tasten \wedge oder \vee zu KONFIGURATOR.



► Bestätigen Sie mit OK.

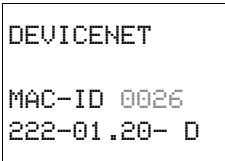


► Wählen Sie bei den Geräten easy800/MFD den Menüpunkt LINK...

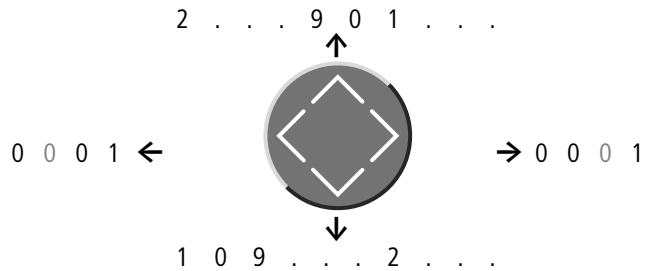


► Bestätigen Sie mit OK.

Es erscheint das Menü DEVICENET.



- Stellen Sie die Adresse mit den Cursor-Tasten ein:
 - Den Wert der aktuellen Ziffer stellen Sie mit ^ oder v 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:

xxx - xx . xx - xx

222-02.10- b

Hardware-Version, Index: b

Software-Version, BTS-Version: 2.1

Geräteerkennung: EASY222-DN

Adresse mittels easySoft einstellen

Bei easySoft, Version 3.1

⟨Menü → Online → Erweiterungsgeräte konfigurieren⟩

Bei easySoft, ab Version 4.01

⟨Menü → Kommunikation → Konfiguration → Erweiterungsgeräte → EASY222-DN⟩.



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



Für Geräte der Geräteversionskennung 01 gilt:

Nachdem Sie die MAC ID über das Basisgerät abgeändert haben, müssen Sie EASY222-DN neu starten. Hierzu schalten Sie die Versorgungsspannung aus und wieder ein. EASY222-DN-Geräte mit einer Versionskennung > 01 übernehmen die Adresse automatisch.

Adresse über die Mastersteuerung einstellen

Eine weitere Möglichkeit zum Einstellen oder Ändern der MAC ID des Gateway besteht über die zur Master-SPS mitgelieferten Konfigurationssoftware. Informationen hierzu entnehmen Sie bitte der SPS beiliegenden Beschreibung.



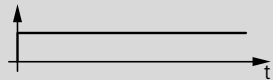

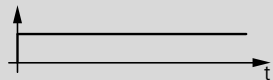

Verschiedene Softwarepakete ermöglichen außerdem das Ändern der MAC ID über das Versenden einer expliziten Nachricht. Hierzu nutzen Sie bitte den entsprechenden Dienst aus dem DeviceNet Object (Abschnitt „DeviceNet Object“, Seite 37).

LED-Statusanzeigen

Das Erweiterungsgerät EASY222-DN besitzt zwei LED-Anzeigeelemente. Diese geben Ihnen eine Unterstützung zur schnellen Problemanalyse. EASY222-DN überwacht sowohl sich selbst, als auch den Kommunikationsbus DeviceNet.


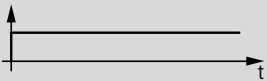


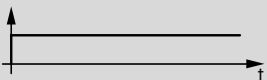

Modul-Status-LED

Die bi-color LED (GRÜN/ROT) zeigt den Status von EASY222-DN an. Hierbei wird überwacht, ob das Gerät überhaupt funktionstüchtig ist und ob es korrekt arbeitet.

AUS	Es liegt keine Versorgungsspannung an EASY222-DN an.	
GRÜN blinkend	EASY222-DN befindet sich im Standby-Status. Die Konfiguration ist falsch, unvollständig oder noch nicht erfolgt.	
GRÜN	EASY222-DN arbeitet normal.	
ROT blinkend	Es liegt eine zu behebbende Störung vor. EASY222-DN muss nicht ausgetauscht werden.	
ROT	Es liegt eine nicht zu behebbende Störung an EASY222-DN vor. EASY222-DN muss ausgetauscht werden.	
GRÜN-ROT blinkend	EASY222-DN befindet sich im Selbsttest-Modus.	

Network Status LED

Die bi-color LED (GRÜN/ROT) zeigt den Status des Kommunikationsbusses DeviceNet an. Hierbei wird überwacht, ob EASY222-DN generell funktionstüchtig ist und ob es korrekt arbeitet.

AUS	EASY222-DN ist nicht online. Entweder es befindet sich im DUP_MAC_ID Test oder es liegt keine Spannungsversorgung am Gerät bzw. am BUS an.	
GRÜN	EASY222-DN ist online und die Verbindung aktiv.	
GRÜN blinkend	EASY222-DN ist online. Es besteht noch keine aktive Kommunikationsverbindung	
ROT blinkend	Mindestens eine I/O-Verbindung befindet sich außerhalb der Zeitüberwachung (Time-out-State).	
ROT	Es liegt ein kritischer Fehler im Netzwerk vor. EASY222-DN hat die Kommunikation abgebrochen.	
GRÜN-ROT blinkend	EASY222-DN hat einen Netzwerkzugangsfehler erkannt und befindet sich nun im Kommunikations-Fehler-Status.	

Zykluszeit easy-Basisgerät Die Kommunikation zwischen easy/MFD-Basisgerät und EASY222-DN über 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 EASY222-DN können Sie in die DeviceNet-Struktur mit Hilfe einer standardisierten EDS-Datei einbinden (Electronic Data Sheet = Elektronisches Datenblatt).

In dieser EDS-Datei sind in erster Linie die Polled I/O Connection, die COS I/O Connection als auch die Cyclic I/O Connection des Gateway definiert. Die Daten bzw. Parameter (easy Object) für die im easy-Grundgerät vorhandenen Funktionen werden in dieser Datei nicht angeführt. Der Zugriff auf diese Funktionen wird über explizite Nachrichten realisiert.

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

<http://www.eaton.com/moeller> → Support → Suchbegriff „EASY222-DN“

Folgen Sie dem "Link" auf dieser Seite.

Die EDS-Datei ist im Anhang nochmals in ausgedruckter Form angeführt (→ Abschnitt „EDS-Datei“, Seite 259).



Hinweis zur EDS-Datei:

Der Identity Object Eintrag - Major Revision gibt den aktuellen Betriebssystemstand des Kommunikationsmoduls EASY222-DN an. Da das Gerät mit einer neueren Betriebssystemversion von der EDS-Beschreibung in diesem Punkt abweichen kann, muss dieser Eintrag entsprechend abgeändert werden, → Abschnitt „Identity Object“ auf Seite 35.

4 DeviceNet-Funktionen

Objektmodell

EASY222-DN basiert auf dem Communications Adapter Profil gemäß der ODVA-Spezifikation (Release V2.0).

Mit dem DeviceNet Objektmodell kann die komplette Funktionalität von EASY222-DN beschrieben werden. Das Objektmodell spiegelt die prinzipielle Kommunikation der Anwendungsschicht wider. Dieses Handbuch geht bei der folgenden Beschreibung nur auf die für Sie relevanten Objekte ein. Das größte Augenmerk wird hierbei auf die herstellerspezifische Klasse „easy Object“ gelegt.

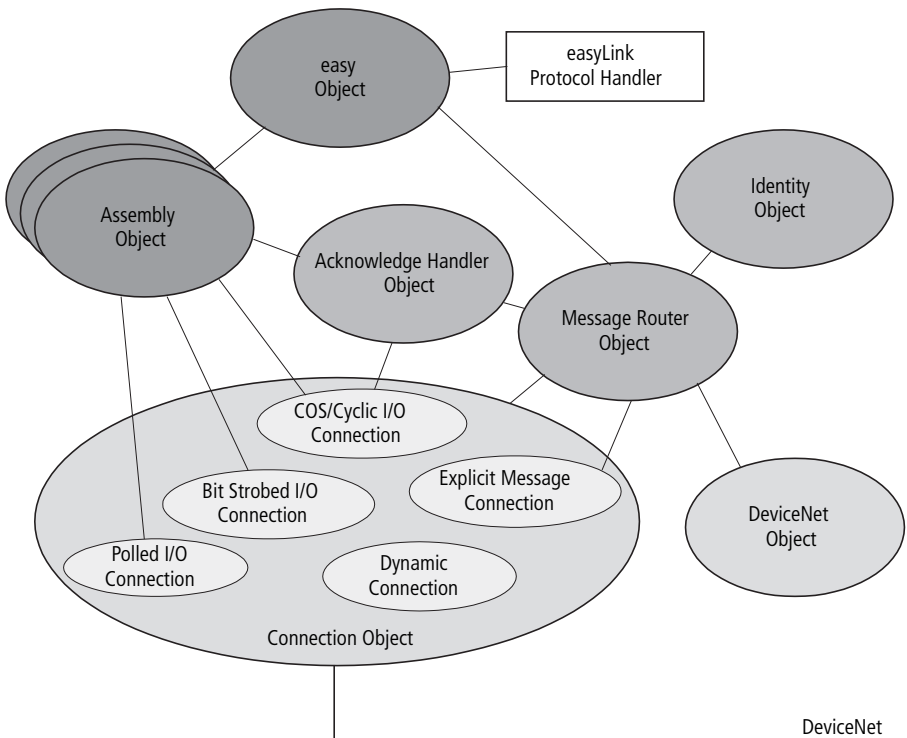


Abbildung 11: DeviceNet-Objekte

Die in der Abbildung dargestellten DeviceNet-Objekte können nochmals zusammengefasst werden in „Managementobjekte“, „Verbindungsobjekte“ und „Herstellerspezifische Objekte“. Ihre Aufgabe wird im Anschluss kurz erläutert.

	Objektadresse		Service Adresse [hex]	Funktion Attribute ID [hex]
	Class ID [hex]	Instance ID [hex]		
① Managementobjekte				
Identity Object	01	01		→ Seite 33
Message Router	02	01		
② Verbindungsobjekte				
DeviceNet Object	03	01		→ Seite 33
Connection Object	05	01 – 04, 04 – 0F		
③ Herstellerspezifische Objekte				
easy Object	64	01		→ Seite 38
Direkter Zugriff: Ein-/Ausgänge, Betriebsart				
Lesen			0E	→ Kapitel 5
Schreiben			10	
Erweiterter Zugriff: Uhrzeit, Abbilddaten, Funktionsbausteine			32	
easy600				→ Kapitel 6
easy700				→ Kapitel 7
easy800/MFD				→ Kapitel 8
Assembly Object	04	64 – 66		

① Managementobjekte

Definieren DeviceNet-spezifische Daten und Funktionen und müssen von jedem DeviceNet-Gerät unterstützt werden:

- Identity Object

Das Identity Object (Class ID 01_{hex}) enthält alle Daten zur eindeutigen Identifizierung eines Knotens innerhalb des Netzwerkes, wie Vendor ID, Device Type und Product Code. Darüber hinaus beinhaltet es den aktuellen Status eines Gerätes, die Seriennummer und den Produktnamen.

Detaillierte Informationen → Seite 35.

- Message Router Object

Das Message Router Object (Class ID 02_{hex}) ermöglicht den Zugriff auf alle Klassen und Instanzen im Gerät über Explicit Messages.

② Verbindungsobjekte

Definieren die über DeviceNet ausgetauschten Nachrichten:

- DeviceNet Object

Das DeviceNet Object (Class ID: 03_{hex}) muss von jedem Gerät unterstützt werden. Es definiert die physikalische Anbindung eines Gerätes an das DeviceNet-Netzwerk. Das heißt, es beinhaltet unter anderem die Geräteadresse (MAC ID) sowie die aktuell eingestellte Baudrate.

Detaillierte Informationen → Seite 37 .

- Connection Object

Das Connection Object (Class ID: 05_{hex}) wird von jedem DeviceNet-Gerät in mindestens einer Instanz unterstützt. Es definiert die Verbindung zu den Daten über I/O Messages oder Explicit Messages, den Pfad und die Länge der zu produzierenden/konsumierenden Daten, den für die Verbindung benutzten CAN-Identifier, die Zeitüberwachung sowie das Verhalten im Fehlerfall.

③ Herstellerspezifische Objekte

Definieren gerätespezifische Daten und Funktionen (Application Objects, Parameter Object, Assembly Object).

- Application Objects – easy Object

Applikationsobjekte (Class ID: 64_{hex}) beschreiben einfache Anwendungen der Automatisierungstechnik. Sie sind entweder in der DeviceNet-Objektbibliothek vordefiniert, oder werden vom Anwender selbst definiert.

Detaillierte Informationen → Seite 38.

- Assembly Objects

Mit dem Assembly Object (Class ID: 04_{hex}) wird dem Anwender die Möglichkeit des Mappings gegeben, d. h. Daten aus Attributen unterschiedlicher Instanzen in verschiedenen Klassen können in einem einzigen Attribut einer Instanz eines Assembly Objects zusammengefasst werden.

Identity Object

Objekt Adresse		Funktion	Zugriff
Class ID	Instance ID	Attribute ID	ServiceCode
01 _{hex}	01 _{hex}	→ Tabelle 1	→ Tabelle 2

Tabelle 1: Attribute IDs der Identity Object Instance

Attribute ID	Zugriff	Name	Beschreibung	Größe [Byte]
1	Lesen	Vendor ID	Die Vendor ID wird von der ODVA vergeben. Für die Eaton GmbH ist dies die 248 _{dez} .	2
2	Lesen	Geräte Typ	Das EASY222-DN gehört zu der Kategorie der Kommunikations-Adapter. Der Wert hierfür beträgt 12 _{dez} .	2
3	Lesen	Produkt Code	Der Produktcode ist von der Firma Eaton mit 650 _{dez} definiert worden. Dieser beschreibt die Modellnummer.	2
4	Lesen	Geräteversion	Beim Auslesen der Geräteversion werden 2 Byte zurückgegeben.	
		Hardwareversion, Betriebssystemversion	Das niederwertige Byte steht für die Hardware-Version und das höherwertige Byte für die Betriebssystemversion.	1 1
5	Lesen	Status	Dieses Attribut beschreibt den kompletten Status des Gerätes.	2
6	Lesen	Seriennummer	Mit diesem Attribut lässt sich die Seriennummer des Gerätes auslesen.	4
7	Lesen	Produktname	Der Produktname EASY222-DN wird als ASCII-Wert (hex) abgelegt.	12
9	Lesen	Konfigurations-Folge-richtigkeit-Wert	Diese Attribut gibt einen Zählwert zurück, der die Anzahl der Modifikationen auf dem nicht flüchtigen Speicher (E2PROM) überwacht.	2
10	Lesen/ Schreiben	Heartbeat Interval	Hier wird ein Intervall zwischen den einzelnen Heartbeat-Nachrichten in Sekunden vorgegeben.	2

Service Code

Die Identity Object Instance und auch die folgenden Instanzen unterstützen die Dienste der folgenden Tabelle.

Tabelle 2: Service Code

Service Code Wert	Service Name	Beschreibung
05 _{hex}	Reset	Ruft die Reset-Funktion des Kommunikationsmoduls EASY222-DN auf.
0E _{hex}	Get_Attribute_Single	Mit diesem Dienst lässt sich der Wert eines ausgewählten Attributes aus dem Kommunikationsmodul lesen.
10 _{hex}	Set_Attribute_Single	Mit diesem Dienst lässt sich der Wert eines ausgewählten Attributes in das Gerät schreiben.

DeviceNet Object

Objekt Adresse		Funktion	Zugriff
Class ID	Instance ID	Attribute ID	ServiceCode
03 _{hex}	01 _{hex}	→ Tabelle 3	→ Tabelle 2

Die DeviceNet Object Instance dient zur Konfiguration des Kommunikationsmoduls EASY222-DN und zur Definition der physikalischen Umgebung. Hierbei werden die gleichen Service-Codes verwendet wie auch beim Identity Object.

Tabelle 3: Attribute IDs der DeviceNet Object Instance

Attribute ID	Zugriff	Name	Beschreibung	Größe [Byte]
1	Lesen/ Schreiben	MAC ID	Die MAC ID ist die Netzwerkadresse eines Netzwerkknotens. Sie kann für EASY222-DN mit diesem Attribut über den Feldbus gelesen und gesetzt werden. Wertebereich: 0 bis 63 _{dez} . (→ Abschnitt „DeviceNet Teilnehmeradresse einstellen“, Seite 24)	1
2	Lesen/ Schreiben	Baud Rate	Mit diesem Attribut lässt sich die Übertragungsrate für die Kommunikation auslesen und gegebenenfalls schreiben Wertebereich: 0 bis 2, 125 bis 500 kbps (→ Abschnitt „Übertragungsraten – automatische Baudratenerkennung“, Seite 22).	1
3	Lesen/ Schreiben	BOI (Bus-Off Interrupt)	Mit diesem Attribut kann das Verhalten bei einem Bus-Off-Ereignis definiert werden (CAN-spezifisch).	1
4	Lesen/ Schreiben	Bus-Off-Counter	Dieser Wert gibt an, wie oft ein Bus-Off-Ereignis aufgetreten ist. Wertebereich 0 bis 255.	1

easy Object

Objekt Adresse		Funktion	Zugriff
Class ID	Instance ID	Attribute ID	ServiceCode
64 _{hex}	01 _{hex}	→ Tabelle 4	→ Tabelle 5

Mit Hilfe des easy Object kann über den Kommunikationsbus DeviceNet auf die Funktionen von easy/MFD zugegriffen werden. Die folgende Tabelle umfasst die von dem Objekt unterstützten Attribute. Hierbei geben die 2 Bytes der Attribute 1 und 2 die Diagnose-Daten des Gerätes an. Mit Attribut 3 können Sie auf die Ausgänge (S1 bis S8) und mit Attribut 4 auf die Eingänge (R1 bis R16) des Grundgerätes zugreifen.

In einer DeviceNet-Konfigurationssoftware (z. B.: RS NetWorx) lassen sich diese Daten direkt auf die entsprechenden Speicherbereiche einer SPS mappen.

Tabelle 4: Attribute IDs der Easy Object Instance

Attribute ID	Zugriff	Name	Beschreibung	Größe [Byte]
1	Lesen	easy-Status	Mit Hilfe dieses Attributes ist es möglich, den Status von easy auszulesen (RUN oder STOP), → Tabelle 6	1
2	Lesen	Koppelmodul-Status	Mit Hilfe dieses Attributes ist es möglich, den Status von easyLink auszulesen, → Tabelle 6	1
3	Lesen	Eingänge – Send Daten	Die Eingangsdaten werden von easy auf den DeviceNet-Bus übertragen. In easy müssen hierzu die Ausgänge S1 bis S8 verwendet werden. Die Zusammensetzung der 3 Byte ist im Abschnitt „Eingangsdaten: Betriebsart, S1 – S8“, Seite 49, detailliert beschrieben.	3

Attribute ID	Zugriff	Name	Beschreibung	Größe [Byte]
4	Lesen/ Schreiben	Ausgänge – Receive Daten	Die Ausgangsdaten werden vom DeviceNet-Bus zu easy übertragen. In easy müssen hierzu die Eingänge R1 bis R16 verwendet werden. Die Zusammensetzung der 3 Byte ist Abschnitt „Ausgangsdaten: Betriebsart, R1 – R16“, Seite 51, detailliert beschrieben.	3
5	Lesen/ Schreiben	vordefinierte Ausgänge	Mit Hilfe diesem Attribute lassen sich die Ausgangsdaten („R“-Daten) vordefinieren, die das Gerät EASY222-DN beim Einschalten aufweist. Die Zusammensetzung der 3 Byte sind im Abschnitt „Ausgangsdaten: Betriebsart, R1 – R16“, Seite 51 detailliert beschrieben.	3

Service Code

Die easy Object Instance unterstützt folgende Dienste.

Tabelle 5: Service Code

Service Code Wert	Service Name	Beschreibung
0E _{hex}	Get_Attribute_Single	Mit diesem Dienst lässt sich der Wert eines ausgewählten Attributes aus dem Kommunikationsmodul lesen.
10 _{hex}	Set_Attribute_Single	Mit diesem Dienst lässt sich der Wert eines ausgewählten Attributes in das Gerät schreiben.
32 _{hex}	Erweiterter Zugriff ¹⁾	Mit diesem Dienst können die zusätzlichen Parameter ¹⁾ des Steuerrelais angesprochen werden:

- 1) Zusätzliche Parameter sind „Uhrzeit“, „Abbilddaten“ und „Funktionsbausteine“. Die Ansprache der Parameter ist easy-spezifisch und wird in den Kapiteln 5 – 7 detailliert beschrieben. Der erweiterte Zugriff erfolgt über eine Explizite Nachrichtenübertragung. Dieses Übertragungsprotokoll ermöglicht den Datenaustausch der Steuerbefehle. Weitere Informationen zum Übertragungsprotokoll finden Sie im Abschnitt „DeviceNet-Kommunikationsprofil“ ab Seite 41.

Change of State I/O Connection

Tabelle 6: Diagnosedaten: 2 Byte

Byte	Bedeutung	Wert	Bedeutung
0	easy-Status (Attribute ID 1)	00 _{hex}	Dieser Wert wird nicht verändert.
1	Koppelmodul-Status (Attribute ID 2)	00 _{hex}	Das Basisgerät ist mit dem Gateway EASY222-DN über den easyLink verbunden (connected).
		04 _{hex}	Das Basisgerät ist nicht eingeschaltet oder nicht mit dem Gateway EASY222-DN über den easyLink verbunden.



Mit Unterbrechen der Kommunikation zwischen dem Grundgerät easy/MFD und dem Erweiterungsgerät EASY222-DN wird der entsprechende Fehler-Code im dritten Datenbyte generiert. Außerdem werden die R/S-Daten des Gateway mit dem Wert 00_{hex} gesendet.

**DeviceNet-
Kommunikationsprofil**

DeviceNet basiert auf einem verbindungsorientierten Kommunikationsmodell. Das bedeutet, dass die Daten immer nur über bestimmte, den Geräten zugeordnete, Verbindungen ausgetauscht werden können.

Die Kommunikation zwischen den Teilnehmern im DeviceNet kann entweder über E/A-Nachrichten (I/O Messages) oder über explizite Nachrichten (Explicit Messages) erfolgen.

I/O Messages

I/O Messages dienen dem Austausch von hochpriorien Prozess- und Applikationsdaten über das Netzwerk. Die Kommunikation zwischen Teilnehmern am DeviceNet erfolgt hierbei nach dem Client/Server-Modell. Das bedeutet: Eine „produzierende“ Applikation überträgt Daten an eine oder mehrere „konsumierende“ Applikationen. Dabei ist es durchaus möglich, dass mehrere Applikationsobjekte in einem einzigen Gerät angesprochen werden.

Die Kommunikation zwischen Geräten über I/O Messages setzt die Einrichtung eines „I/O Messaging Connection Objects“ voraus. Dieses können Sie auf zwei Arten aktivieren:

- Entweder durch ein statisches, bereits im Gerät vorhandenes „I/O-Verbindungsobjekt“ über das „Predefined Master/Slave Connection Set“ oder
- durch ein dynamisch eingerichtetes „I/O-Verbindungsobjekt“. Letzteres können Sie über ein bereits im Gerät existierendes „Explicit Messaging Connection Object“ einrichten.

Explicit Messages

Explizite Nachrichten dienen zur Übertragung niederpriorer Konfigurationsdaten, allgemeiner Managementdaten oder Diagnosedaten zwischen zwei bestimmten Geräten. Es handelt sich hierbei immer um eine Punkt-zu-Punkt-Verbindung in einem Client/Server-System, wobei auf eine Anfrage eines Clients (Request) immer eine Bestätigung des Servers (Response) folgen muss.

Wie bei den I/O Messages muss auch bei der Kommunikation zwischen Geräten über Explicit Messages zunächst ein „Connection Object“ – das „Explicit Messaging Connection Object“ – eingerichtet werden. Dies kann entweder durch Aktivieren eines bereits im Gerät vorhandenen statischen Verbindungsobjekts über das „Predefined Master/Slave Connection Set“ erfolgen, oder dynamisch über den UCMM-Port (Unconnected Message Manager Port) eines Gerätes.

Alle Daten der Funktionsrelais (easy-Grundgeräte) werden mit Hilfe expliziter Nachrichten verarbeitet. Die Master-Steuerung ist damit in der Lage, die Parameter für folgende Funktionen zu schreiben oder auszulesen.

- Uhrzeit
- Abbilddaten
- Funktionsbausteine (Zähler, Zeitglieder, Analogwertvergleicher,...).



Die DeviceNet-Anbindung des easy-Steuerrelais an eine Mastersteuerung SLC 500 erfordert für die Ausführung der Steuerbefehle (Explicit Messages) bestimmte Befehls- und Handshakeroutinen im SPS-Programm.

Der Anwendungshinweis AN2700K17D unterstützt die Steuerbefehle von EASY222-DN. Er stellt Routinen zur Steuerung der notwendigen „Expliziten Nachrichten“ in Form eines Unterprogramms zur Verfügung, sodass die Programmierung durch den Aufruf und die Parametrierung des Unterprogramms ersetzt wird. Die Parametrierung wird über eine Integer-Datei durchgeführt.

Der selbstentpackende Anwendungshinweis AN2700K17D.exe (für die EASY600) steht unter ftp://ftp.moeller.net/AUTOMATION/APPLICATION_Notes/an27k17d.exe zum Herunterladen bereit.

Generelle Arbeitsweise

Nachfolgend soll die generelle Arbeitsweise mit dem EASY222-DN vorgestellt werden. Der azyklische Datenverkehr wird mit Hilfe der expliziten Nachrichten realisiert. Die Funktionsbausteine des easy-Grundgerätes können über den Service Code = 32_{hex} angesprochen werden. Hierbei erfolgt die Differenzierung der unterschiedlichen Funktionen und Parameter über die zugeordnete Attribute ID.

ServiceCode	Objekt Adresse	
	Class ID	Instance ID
32 _{hex}	64 _{hex}	01 _{hex}

Exkurs:

DeviceNet basiert auf dem Standard-CAN-Protokoll und verwendet daher einen 11-Bit-Nachrichtenidentifizier. Dadurch sind $2^{11} = 2048$ Nachrichten (000_{hex} – 7FF_{hex}) unterscheidbar. Da ein DeviceNet-Netzwerk auf maximal 64 Teilnehmer begrenzt ist, genügen 6 Bit zur Kennzeichnung eines Gerätes. Diese wird als MAC-ID (Geräte- oder Knotenadresse) bezeichnet.

Je nach Nutzungsmodell stehen vier Nachrichtengruppen (Message Groups) unterschiedlicher Größe zur Verfügung.

Im Sprachgebrauch von DeviceNet wird der CAN-Identifizier als Connection ID bezeichnet. Diese setzt sich aus der Kennung der Message-Group (Message ID) und der MAC-ID des Gerätes zusammen:

- Als MAC-ID ist sowohl die Quell- als auch die Zieladresse möglich; die Definition ist abhängig von der Message Group und Message ID.
- Mit der Message ID wird die Bedeutung der Nachricht im System festgelegt.

In der Welt des DeviceNet stehen vier Nachrichtengruppen zur Verfügung. Das EASY222-DN nutzt die Nachrichtengruppe 2 (Message Group 2). Diese Gruppe nutzt 512 CAN-Identifizier ($400_{\text{hex}} - 5FF_{\text{hex}}$). Die meisten Message Ids dieser Gruppe sind optional definiert für die Nutzung des „Predefined Master/Slave Connection Sets“. Eine Message ID wird für Netzwerkmanagement genutzt. Die Priorität wird hier in erster Linie durch die Geräteadresse und erst dann durch die Message ID bestimmt. Betrachtet man die Bitposition im Detail, erkennt man, dass ein CAN-Controller mit einer 8-Bit-Maske in der Lage ist, seine Group-2-Messages gezielt herauszufiltern.

Connection ID = CAN-Identifizier										Bedeutung	
10	9	8	7	6	5	4	3	2	1		0
1	0	MAC-ID					Message ID			Message Group 2	
1	0	Source MAC ID					0	0	0	Master's I/O Bit-Strobe Command Message	
1	0	Source MAC ID					0	0	1	Reserved for Master's Use – Use is TBD	
1	0	Destination MAC ID					0	1	0	Master's Change of State or Cyclic Acknowledge Message	
1	0	Source MAC ID					0	1	1	Slave's Explicit/ Unconnected Response Messages	
1	0	Destination MAC ID					1	0	0	Master's Explicit Request Messages	
1	0	Destination MAC ID					1	0	1	Master's I/O Poll Command/Change of State/Cyclic Message	
1	0	Destination MAC ID					1	1	0	Group 2 Only Unconnected Explicit Request Messages	
1	0	Destination MAC ID					1	1	1	Duplicate MAC ID Check Messages	

Quelle: ODVA- DeviceNet Specification Release 2.0, Chapter 7-2

In der folgenden Tabelle ist der Datenverkehr auf dem DeviceNet-Kommunikationsbus aufgezeichnet worden. Der Datenstrom zeigt die Telegramme zum Lesen des Datums und der Uhrzeit im easy700 an (→ Abschnitt „Datum und Uhrzeit lesen/schreiben“ auf Seite 101).

Das Kommunikationsmodul EASY222-DN besitzt die MAC ID = 3. Es ist bei der Datenfolge zu berücksichtigen, dass der Zugriff in fragmentierter Form realisiert ist. Näheres hierzu finden Sie in der ODVA-Spezifikation.

Beschreibung	ID (hex)	Länge	DeviceNet – Byte (hex)							
			0	1	2	3	4	5	6	7
Master sendet eine Anfrage (hex) mit : Byte 2 - Servicecode = 32 Byte 3 - CLASS-ID = 64 Byte 4 - Instance-ID = 01 Byte 5 - Attribute ID = 93 Byte 6 - Len = 05 Byte 7 - Index = 0	41C	8	80	00	32	64	01	93	05	00
DeviceNet-spezifisch easyLink-spezifisch										
Bestätigung des Slaves (Fragmentation Protocol)	41B	3	80	C0	00					
Master sendet restliche easyLink-Byte Byte 2 - Data 1 = 00 Byte 3 - Data 2 = 00 Byte 4 - Data 3 = 00 Byte 5 - Data 4 = 00	41C	6	80	01	00	00	00	00	00	
Bestätigung vom Slaves (Fragmentation Protocol)	41B	3	80	C1	00					
Slave sendet Antwort auf die Anfrage Byte 3 – Antwort = C2 (Lesen erfolgreich) Byte 4 – Len = 05 Byte 5 – Index = 00 Byte 6 – Data 1 = 05	41B	8	80	00	B2	C2	05	00	05	09
Bestätigung vom Masters (Fragmentation Protocol)	41C	3	80	C0	00					
Slave sendet restliche easyLink-Daten: Data 2 = 0D Data 3 = 05 Data 4 = 04	41B	5	80	81	0D	05	04			
Bestätigung vom Masters (Fragmentation Protocol)	41C	3	80	C1	00					

5 Direkter Datenaustausch mit easy/MFD (Polled I/O Connection)

Über den direkten zyklischen Datenaustausch kann der DeviceNet-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 der Ausgangszustände des easy/MFD
 - Abfrage der Betriebsart des easy/MFD.

Um Ein- oder Ausgangsdaten zwischen dem Slave EASY222-DN und einer DeviceNet-Master-Steuerung übertragen zu können, müssen Sie die entsprechenden zyklischen Daten in der jeweiligen Slave-Konfiguration zuweisen (Engl.: to map).



Die Anbindung an die DeviceNet-Steuerungen von Allen Bradley erfolgt über die Zuweisungstabelle im RS-NetWorx-Softwaretool.



Die Begriffe „Eingangsdaten“ und „Ausgangsdaten“ werden aus Sicht des DeviceNet-Masters verwendet.

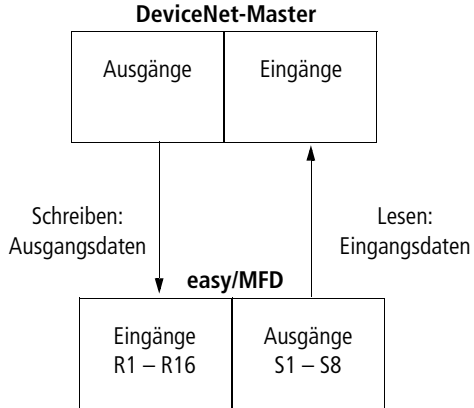


Abbildung 12: Ein- und Ausgangsdaten aus Sicht des DeviceNet-Masters

**Eingangsdaten:
Betriebsart, S1 – S8**

Attribute ID: 3

Der zyklische Datenaustausch zwischen DeviceNet-Master und dem EASY222-DN-Slave wird über die Eingangsdaten Byte 0, 1 und 2 gewährleistet.

Tabelle 7: Byte 0 bis 2: Eingangsdaten, Betriebsart

Byte	Bedeutung	Wert
0	Betriebsart abfragen	→ Tabelle 8
1	Zustand der easy-Ausgänge S1 bis S8 abfragen	→ Tabelle 9
2	Nicht belegt	00 _{hex}

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

Tabelle 8: 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 9: Byte 1: Zustand der easy/MFD-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 $19_{\text{hex}} = 0001\ 1001_{\text{bin}}$:

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 so lange 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.

**Ausgangsdaten:
Betriebsart, R1 – R16****Attribute ID: 4**

Der zyklische Datenaustausch zwischen DeviceNet-Master und dem EASY222-DN-Slave wird über die Ausgangsdaten Byte 0, 1 und 2 gewährleistet.

Tabelle 10: Byte 0 bis 2: Ausgangsdaten, Betriebsart

Byte	Bedeutung	Wert
0	Betriebsart festlegen	→ Tabelle 11
1	Setzen/Rücksetzen der easy/MFD-Eingänge R9 bis R16	→ Tabelle 12
2	Setzen/Rücksetzen der easy/MFD-Eingänge R1 bis R8	→ Tabelle 13

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

Tabelle 11: Byte 0: Betriebsart

easy-Betriebsart	Bit							
	7	6	5	4	3	2	1	0
Index, um das Grundgerät 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 $14_{\text{hex}} = 00010100_{\text{bin}}$:

Dieser Wert muss immer im Byte 0 enthalten sein, wenn Daten über das Gateway EASY222-DN zum Grundgerät easy/MFD geschrieben werden sollen.

Wert $34_{\text{hex}} = 00110100_{\text{bin}}$:

Dieser Wert setzt den easy- 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 $44_{\text{hex}} = 01000100_{\text{bin}}$:

Dieser Wert setzt den easy-Status von RUN auf STOP. Auch er wird nur als Kommandobefehl verwendet und unterliegt somit der selben Arbeitsweise, wie auch das Kommando RUN.

Wert $00_{\text{hex}} = 00000000_{\text{bin}}$:

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 verwendeten I/Os 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 12: Byte 1: Setzen/Rücksetzen der easy/MFD-Eingänge
R9 bis R16

easy/MFD- Ausgang	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} = 00011001_{bin}:

R13, R12 und R9 sollen aktiv werden.

Tabelle 13: 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 so lange auf dem alten Zustand, bis der Steuerbefehl abgearbeitet ist.
- Nach Beenden des Datenaustausch-Steuerbefehl werden die Bytes der Eingänge wieder aktualisiert.

6 Steuerbefehle für easy600

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

- „Datum und Uhrzeit, Sommer-/Winterzeit lesen und schreiben“ (Seite 57)
- „Abbilddaten lesen“ (Seite 61)
- „Funktionsbausteine lesen/schreiben“ (Seite 72).

Hierzu wird in der Mastersteuerung auf das Nachrichtenübertragungsprotokoll der expliziten Nachrichten zurückgegriffen. Die Parameter werden alle über den Service Code 32_{hex} angesprochen. Hierbei erfolgt die Differenzierung der unterschiedliche Parameter über die zugeordnete Attribute ID.

Service Code	Objekt Adresse	
	Class ID	Instance ID
32 _{hex}	64 _{hex}	01 _{hex}



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 DeviceNet übertragen werden, ist ein Datenaustauschverfahren notwendig.



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

Bei der Kommunikation zwischen den Teilnehmern stößt der Master den Datenaustausch mit einem Steuerbefehl an. Der Slave reagiert auf die Anfrage immer mit einer Antwort. Die Antwort gibt Aufschluss darüber, ob der Datenaustausch ausgeführt wurde oder nicht. Konnte der Datenaustausch nicht realisiert werden, wird ein Fehlercode zurückgegeben. Dieser ist von der ODVA genau definiert, → Abschnitt „Referenzen“ auf Seite 8.

Datum und Uhrzeit,
Sommer-/Winterzeit
lesen und schreiben

Telegrammaufbau

Byte		Bedeutung	Wert (hex), sendet		Bit
M	S		Master	Slave	7 6 5 4 3 2 1 0
Attribute ID					
		Lesen	5D	–	0 1 0 1 1 1 0 1
		Schreiben	2A	–	0 0 1 0 1 0 1 0
		0 Antwort			
		Lesen erfolgreich	–	C2	1 1 0 0 0 0 1 0
		Schreiben erfolgreich	–	C1	1 1 0 0 0 0 0 1
		Befehl zurückgewiesen	–	C0	1 1 0 0 0 0 0 0
0	1	Wochentag			
		Beim Lesen	00	→ Tabelle 14	
		Beim Schreiben	→ Tabelle 14	00	
1	2	Stunde			
		Beim Lesen	00	→ Tabelle 15	
		Beim Schreiben	→ Tabelle 15	00	
2	3	Minute			
		Beim Lesen	00	→ Tabelle 16	
		Beim Schreiben	→ Tabelle 16	00	
3	4	Sommer-/Winterzeit			
		Beim Lesen	00	→ Tabelle 17	
		Beim Schreiben	→ Tabelle 17	00	

M = Master

S = Slave

Tabelle 14: Byte 0 (Master) bzw. Byte 1 (Slave):
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 15: Byte 1 (Master) bzw. Byte 2 (Slave):
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 16: Byte 2 (Master) bzw. Byte 3 (Slave):
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 17: Byte 3 (Master) bzw. Byte 4 (Slave):
Winter-/Sommerzeit (Wertebereich 00 bis 01)

Wert (bcd)	Wert 10				Wert 1			
	Bit				Bit			
	7	6	5	4	3	2	1	0
Funktion								
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.

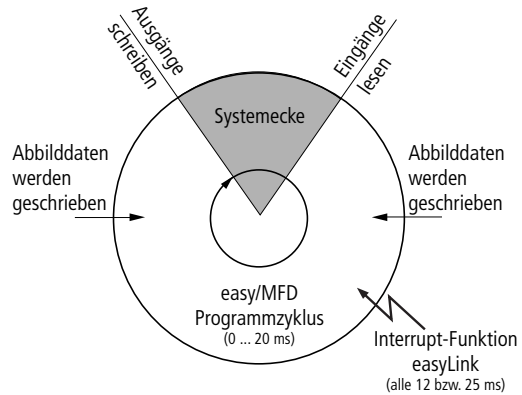
Byte		Bedeutung	Wert (hex), sendet		Bit
M	S		Master	Slave	7 6 5 4 3 2 1 0
Attribute ID					
		Schreiben	2A	–	0 0 1 0 1 0 1 0
	0	Antwort			
		Schreiben erfolgreich	–	C1	1 1 0 0 0 0 0 1
0	1	Wochentag	04	00	
1	2	Stunde (14 _{dez})	0E	00	
2	3	Minute (36 _{dez})	24	00	
3	4	Sommer-/Winterzeit	01	00	

M = Master

S = Slave

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	Attribute ID	Seite
I1 – I16, P1 – P4, ESC/OK/DEL/ALT	„Digital-Eingänge, P-Tasten und Bedientasten“	lesen	5C	62
I7 – I8	„Analog-Eingänge: I7 – I8“	lesen	5B	65
T1 – T8, C1 – C8, Q1 – Q4, A1 – A8	„Zeitrelais, Zählerrelais, Zeitschaltuhren, Analogwertvergleicher“	lesen	5E	66
M1 – M16, Q1 – Q8, D1 – D8	„Hilfsrelais (Merker), Digital-Ausgänge, Textanzeige“	lesen	5F	69

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
M	S		Master	Slave	7 6 5 4 3 2 1 0
		Attribute ID			
		Lesen	5C	–	0 1 0 1 1 1 0 0
	0	Antwort			
		Lesen erfolgreich	–	C2	1 1 0 0 0 0 1 0
		Befehl zurückgewiesen	–	C0	1 1 0 0 0 0 0 0
0	1	Zustand der Eingänge I1 bis I8	00	→ Tabelle 18	
1	2	Zustand der Eingänge I9 bis I16	00	→ Tabelle 19	
2	3	Zustand der Tasten	00	→ Tabelle 20	

M = Master

S = Slave

Tabelle 18: 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 19: 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 20: 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 $01_{\text{hex}} = 00000001_{\text{bin}}$:

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							
M	S		Master	Slave	7	6	5	4	3	2	1	0
		Attribute ID										
		Lesen	5B	–	0 1 0 1 1 0 1 1							
		0 Antwort										
		Lesen erfolgreich	–	C2	1 1 0 0 0 0 1 0							
		Befehl zurückgewiesen	–	C0	1 1 0 0 0 0 0 0							
0	1	Analogwert von I7	00	siehe unten								
1	2	Analogwert von I8	00									

M = Master
S = Slave

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							
M	S		Master	Slave	7	6	5	4	3	2	1	0
		Attribute ID										
		Lesen	5E	–	0 1 0 1 1 1 1 0							
		0 Antwort										
		Lesen erfolgreich	–	C2	1 1 0 0 0 0 1 0							
		Befehl zurückgewiesen	–	C0	1 1 0 0 0 0 0 0							
0	1	Zustand Zeitrelais	00	→ Tabelle 21								
1	2	Zustand Zählerrelais	00	→ Tabelle 22								
2	3	Zustand Zeitschaltuhr	00	→ Tabelle 23								
3	4	Zustand Analogwertvergleicher	00	→ Tabelle 24								

M = Master

S = Slave

Tabelle 21: 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 $2B_{\text{hex}} = 00101011_{\text{bin}}$:

T6, T4, T2 und T1 sind aktiv.

Tabelle 22: 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 $19_{\text{hex}} = 00011001_{\text{bin}}$:

C5, C4 und C1 sind aktiv.

Tabelle 23: Byte 3: Zustand der Zeitschaltuhren

	Bit							
	7	6	5	4	3	2	1	0
W1								0/1
W2							0/1	
W3						0/1		
W4					0/1			
				0				
			0					
		0						
	0							

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

Tabelle 24: 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							
M	S		Master	Slave	7	6	5	4	3	2	1	0
		Attribute ID										
		Lesen	5F	–	0 1 0 1 1 1 1 1							
		0 Antwort										
		Lesen erfolgreich	–	C2	1 1 0 0 0 0 1 0							
		Befehl zurückgewiesen	–	C0	1 1 0 0 0 0 0 0							
0	1	Zustand der Hilfsrelais (Merker) M1 bis M8	00	→ Tabelle 25								
1	2	Zustand der Hilfsrelais (Merker) M9 bis M16	00	→ Tabelle 26								
2	3	Zustand der Digital-Ausgänge Q1 bis Q8	00	→ Tabelle 27								
3	4	Zustand der Textmerker D1 bis D8	00	→ Tabelle 28								

M = Master

S = Slave

Tabelle 25: 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 26: 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 27: 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 $A8_{\text{hex}} = 10101000_{\text{bin}}$:

Q8, Q6 und Q4 sind aktiv.

Tabelle 28: 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 $84_{\text{hex}} = 10000100_{\text{bin}}$:

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 folgende Tabelle stellt die möglichen Befehle dar.

Operanden	Bedeutung	Befehl	Seite
A1 – A8	„Analogwertvergleichler A1 – A8: Istwerte schreiben (Funktion, Vergleichswerte)“	22 _{hex} – 29 _{hex}	73
C1 – C8	„Zählerrelais C1 – C8: Istwert lesen“	3B _{hex} – 42 _{hex}	76
	„Zählerrelais C1 – C8: Sollwert schreiben“	09 _{hex} – 10 _{hex}	78
	„Zählerrelais C1 – C8: Sollwert lesen“	43 _{hex} – 4A _{hex}	80
T1 – T8	„Zeitrelais T1 – T8: Istwert lesen (Zeitbereich, Istwert, Schalfunktion)“	2B _{hex} – 32 _{ex}	82
	„Zeitrelais T1 – T8: Parameter schreiben (Zeitbereich, Sollwert, Schalfunktion)“	01 _{hex} – 08 _{hex}	86
I1 – I4	„Zeitschaltuhr I1 – I4: Istwerte lesen (Kanal, ON-Zeit, OFF-Zeit)“	4B _{hex} – 5A _{hex}	90
	„Zeitschaltuhr I1 – I4: Sollwerte schreiben (Kanal, ON-Zeit, OFF-Zeit)“	12 _{hex} – 21 _{hex}	94

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

Byte		Bedeutung	Wert (hex), sendet		Bit
M	S		Master	Slave	
Attribute ID: Schreiben					
		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
	0	Antwort			
		Schreiben erfolgreich	–	C1	1 1 0 0 0 0 0 1
		Befehl zurückgewiesen	–	C0	1 1 0 0 0 0 0 0
0	1	Steuer-Byte	→ Tabelle 29	00	
1	2	Vergleichswert bei Vergleich mit Konstante	→ Seite 74	00	

M = Master
S = Slave



Halten Sie den Wertebereich ein: Die Vergleichswerte sowie die Funktion sind Bestandteil einer „*.eas-Datei“. Werden diese Werte verändert, stimmt die Original „*.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 29: Byte 0: Steuer-Byte

Bedeutung	Bit							
	7	6	5	4	3	2	1	0
Vergleich auf „ \geq “								0
Vergleich auf „ \leq “								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 \geq der vorgegebenen Konstante (\rightarrow Byte 1) ist.

Vergleichswert (Byte1)

Dieses Byte enthält den Vergleichswert 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 $I7 < 4,7\text{ V}$

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
	Attribute ID: A8	29	0	0	1	0	1	0	0	1
0	Steuer-Byte	→	1	0	0	0	0	0	1	1
1	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	C1	1	1	0	0	0	0	0	1
1	Vergleicher	00								
2	Vergleichswert bei Vergleich mit Konstante	00								

Zählerrelais C1 – C8: Istwert lesen

Telegrammaufbau

Byte		Bedeutung	Wert (hex), sendet		Bit
M	S		Master	Slave	
Attribute ID: 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
	0	Antwort			
		Lesen erfolgreich	–	C2	1 1 0 0 0 0 1 0
		Befehl zurückgewiesen	–	C0	1 1 0 0 0 0 0 0
0	1	Steuer-Byte	00	→ Tabelle 30	x x x x x x x x
1	2	Zählerrelais-Istwert (Low Byte)	00	→ Seite 77	
2	3	Zählerrelais-Istwert (High Byte)	00		

M = Master

S = Slave

Tabelle 30: 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 $80_{\text{hex}} = 10000000_{\text{bin}}$:

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}

$10DE_{\text{hex}} = 4318_{\text{dez}}$

Zählerrelais C1 – C8: Sollwert schreiben

Telegrammaufbau

Byte		Bedeutung	Wert (hex), sendet		Bit
M	S		Master	Slave	7 6 5 4 3 2 1 0
Attribute ID: Schreiben					
		C1	09	–	1 0 0 0 1 0 0 1
		C2	0A	–	1 0 0 0 1 0 1 0
		C3	0B	–	1 0 0 0 1 0 1 1
		C4	0C	–	1 0 0 0 1 1 0 0
		C5	0D	–	1 0 0 0 1 1 0 1
		C6	0E	–	1 0 0 0 1 1 1 0
		C7	0F	–	1 0 0 0 1 1 1 1
		C8	10	–	1 0 0 1 0 0 0 0
	0	Antwort			
		Schreiben erfolgreich	–	C1	1 1 0 0 0 0 0 1
		Befehl zurückgewiesen	–	C0	1 1 0 0 0 0 0 0
0	1	Steuer-Byte	→ Tabelle 31	00	
1	2	Sollwert (Low Byte)	→ Seite 79	00	
2	3	Sollwert (High Byte)		00	

M = Master
S = Slave

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 Original „*.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 31: Byte 0: 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 $80_{\text{hex}} = 1000000_{\text{bin}}$:

Sollwert wird in das gewählte Zählerrelais geschrieben und erscheint im Parametermenü.

Sollwert schreiben (Byte 1 und Byte 2)

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 = $4318_{\text{dez}} = 10DE_{\text{hex}}$:

Low-Wert: DE_{hex}

High-Wert: 10_{hex}

Zählerrelais C1 – C8: Sollwert lesen

Telegrammaufbau

Byte		Bedeutung	Wert (hex), sendet		Bit
M	S		Master	Slave	
Attribute ID: Lesen					
		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
	0	Antwort			
		Lesen erfolgreich	–	C2	1 1 0 0 0 0 1 0
		Befehl zurückgewiesen	–	C0	1 1 0 0 0 0 0 0
0	1	Steuer-Byte	00	→ Tabelle 32	
1	2	Zählerrelais-Sollwert (Low Byte)	00	→ Seite 81	
2	3	Zählerrelais-Sollwert (High Byte)	00		

M = Master

S = Slave

Tabelle 32: 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 $80_{\text{hex}} = 10000000_{\text{bin}}$:

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}

$10DE_{\text{hex}} = 4318_{\text{dez}}$

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

Telegrammaufbau

Byte		Bedeutung	Wert (hex), sendet		Bit
M	S		Master	Slave	
Attribute ID: 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
	0	Antwort			
		Lesen erfolgreich	–	C2	1 1 0 0 0 0 1 0
		Befehl zurückgewiesen	–	C0	1 1 0 0 0 0 0 0
0	1	Steuer-Byte	00	→ Tabelle 33	
1	2	Zeit-Istwert (Low Byte)	00	→ Seite 84	
2	3	Zeit-Istwert (High Byte)	00		
3	4	zufälliger Wert	00	→ Seite 84	
4–5	5–6		00	00	

M = Master

S = Slave

Tabelle 33: 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 34: 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	Attribute ID: 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 = $0E10_{\text{hex}} = 3600$

$3600 \text{ s} = 60:00 \text{ M:S}$

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

Byte		Bedeutung	Wert (hex), sendet		Bit
M	S		Master	Slave	7 6 5 4 3 2 1 0
Attribute ID: 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
	0	Antwort			
		Schreiben erfolgreich	–	C1	1 1 0 0 0 0 0 1
		Befehl zurückgewiesen	–	C0	1 1 0 0 0 0 0 0
0	1	Steuer-Byte	→ Tabelle 35	ungültig	
1	2	Sollwert Low-Wert	→ Seite 89	00	
2	3	Sollwert High-Wert			
3 – 5	4 – 6		00	00	
M	= Master				
S	= Slave				



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 Original „*.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 35: Byte 0: 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 1 und Byte 2)

Byte 1 und Byte 2 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 01 – 04: Istwerte lesen
(Kanal, ON-Zeit, OFF-Zeit)**

Telegrammaufbau

Byte		Bedeutung	Wert (hex), sendet		Bit							
M	S		Master	Slave	7	6	5	4	3	2	1	0
Attribute ID: Lesen												
		01 Kanal A	4B	–	0	1	0	0	1	0	1	1
		01 Kanal B	4C	–	0	1	0	0	1	1	0	0
		01 Kanal C	4D	–	0	1	0	0	1	1	0	1
		01 Kanal D	4E	–	0	1	0	0	1	1	1	0
		02 Kanal A	4F	–	0	1	0	0	1	1	1	1
		02 Kanal B	50	–	0	1	0	1	0	0	0	0
		02 Kanal C	51	–	0	1	0	1	0	0	0	1
		02 Kanal D	52	–	0	1	0	1	0	0	1	0
		03 Kanal A	53	–	0	1	0	1	0	0	1	1
		03 Kanal B	54	–	0	1	0	1	0	1	0	0
		03 Kanal C	55	–	0	1	0	1	0	1	0	1
		03 Kanal D	56	–	0	1	0	1	0	1	1	0
		04 Kanal A	57	–	0	1	0	1	0	1	1	1
		04 Kanal B	58	–	0	1	0	1	1	0	0	0
		04 Kanal C	59	–	0	1	0	1	1	0	0	1
		04 Kanal D	5A	–	0	1	0	1	1	0	1	0
	0	Antwort										
		Lesen erfolgreich	–	C2	1	1	0	0	0	0	1	0
		Befehl zurückgewiesen	–	C0	1	1	0	0	0	0	0	0
0	1	Steuer-Byte „Zeitschaltuhr“	00	→ Tabelle 36								
1	2	Steuer-Byte „Kanal“	00	→ Tabelle 37								
2	3	Minute (Schaltpunkt ON)	00	→ Seite 93								
3	4	Stunde (Schaltpunkt ON)	00									
4	5	Minute (Schaltpunkt OFF)	00									
5	6	Stunde (Schaltpunkt OFF)	00									

M = Master

S = Slave

Tabelle 36: 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 $80_{\text{hex}} = 10000000_{\text{bin}}$:

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 37: 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} = 0011 0001_{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								
M	S		Master	Slave	7	6	5	4	3	2	1	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	
	0	Antwort											
		Schreiben erfolgreich	–	C1	1	1	0	0	0	0	0	1	
		Befehl zurückgewiesen	–	C0	1	1	0	0	0	0	0	0	
0	1	Steuer-Byte (Tag-Beginn/-Ende)	→ Seite 95	00									
1	2	Minute (Schaltpunkt ON)	→ Seite 97	00									
2	3	Stunde (Schaltpunkt ON)		00									
3	4	Minute (Schaltpunkt OFF)		00									
4	5	Stunde (Schaltpunkt OFF)		00									
5	6	nicht benutzt											

M = Master
S = Slave



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 Original „*.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 38: Byte 0: 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} = 0011 0001_{bin}:

Der zuvor ausgewählte Kanal X der Wochenzeitschaltuhr Y ist von Montag bis Samstag aktiv.

Ein- und Ausschaltzeitpunkt 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 1	Byte 2	Byte 3	Byte 4
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:	Attribute ID	1E _{hex}
Tag: Montag bis Samstag Kanal erscheint im Parametermenü	Byte 0	31 _{hex} (siehe oben)
ON: 19:00	Byte 1	00 _{hex}
	Byte 2	13 _{hex}
OFF: 06:30	Byte 3	1E _{hex}
	Byte 4	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	Attribute ID: 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								

7 Steuerbefehle für easy700

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

- „Datum und Uhrzeit lesen/schreiben“ (Seite 101)
- „Abbilddaten lesen/schreiben“ (Seite 105)
- „Funktionsbaustein-Daten lesen/schreiben“ (Seite 126).

Hierzu wird in der Mastersteuerung auf das Nachrichtenübertragungsprotokoll der expliziten Nachrichten zurückgegriffen. Die Parameter werden alle über den Service Code 32_{hex} angesprochen. Hierbei erfolgt die Differenzierung der unterschiedliche Parameter über die zugeordnete Attribute ID.

Service Code	Objekt Adresse	
	Class ID	Instance ID
32 _{hex}	64 _{hex}	01 _{hex}



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 DeviceNet übertragen werden, ist ein Datenaustauschverfahren notwendig.



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

Bei der Kommunikation zwischen den Teilnehmern stößt der Master den Datenaustausch mit einem Steuerbefehl an. Der Slave reagiert auf die Anfrage immer mit einer Antwort. Die Antwort gibt Aufschluss darüber, ob der Datenaustausch ausgeführt wurde oder nicht. Konnte der Datenaustausch nicht realisiert werden, wird ein Fehlercode zurückgegeben. Dieser ist von der ODVA genau definiert, → Abschnitt „Referenzen“ auf Seite 8.

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	
M	S		Master	Slave
Attribute ID				
		Lesen	93	–
		Schreiben	B3	–
		0 Antwort		
		Lesen erfolgreich	–	C2
		Schreiben erfolgreich	–	C1
		Befehl zurückgewiesen	–	C0
0	1	Len	05	05
1	2	Index	0 – 2 ¹)	0 – 2 ¹)
2 – 6	3 – 7	Data 1 – 5	abhängig vom Index, → Tabelle 39	abhängig vom Index, → Tabelle 39

- 1) 0 = Uhrzeit/Datum, → Tabelle 39
 1 = Sommerzeit, → Tabelle 40
 2 = Winterzeit, → Tabelle 41

M = Master

S = Slave

Tabelle 39: Index 0 – Datum und Zeit der Echtzeituhr

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

Tabelle 40: Index 1 – Sommerzeit

Byte		Inhalt	Wert (hex)	
Master	Slave			
2	3	Data 1	Area	
			keine	00
			Regel	01
			automatische EU	02
			automatische GB	03
		automatische US	04	
für „Area“ = „Regel“:				
3	4	Data 2	Schaltregel Sommerzeit	→ Tabelle 42
4	5	Data 3		
5	6	Data 4		
6	7	Data 5		

Tabelle 41: Index 2 – Winterzeit
(nur gültig, wenn Area = „Regel“ gewählt worden ist)

Byte		Inhalt		Wert (hex)
Master	Slave			
2	3	Data 1	Area = Regel	01
3 – 6	4 – 7	Data 2 – 5	Schaltregel Winterzeit	→ Tabelle 42

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 42: 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 61 auch für easy700.

Übersicht

Operanden	Bedeutung	Lesen/ schreiben	Type (hex)	Seite
A1 – A16	„Analogwertvergleicher/Schwellwertvergleicher: A1 – A16“	lesen	8B	106
C1 – C16	„Zähler: C1 – C16“	lesen	EE	107
D1 – D16	„Textbausteine: D1 – D16“	lesen	94	108
I1 – I16	„Lokale Eingänge: I1 – I16“	lesen	84	109
IA1 – IA4	„Lokale Analog-Eingänge: IA1 – IA4“	lesen	8C	110
M1 – M16, N1 – N16	„Merker: M1 – M16/N1 – N16 schreiben“	schreiben	86/87	112
M1 – M16, N1 – N16	„Merker: M1 – M16/N1 – N16 lesen“	lesen	86/87	114
O1 – O4	„Betriebsstundenzähler: O1 – O4“	lesen	EF	116
P1 – P4	„Lokale P-Tasten: P1 – P4“	lesen	8A	117
Q1 – Q8	„Lokale Ausgänge: Q1 – Q8“	lesen	85	119
R1 – R16/ S1 – S8	„Ein-/Ausgänge von easyLink: R1 – R16/S1 – S8“	lesen	88/89	120
T1 – T16	„Zeitglieder: T1 – T16“	lesen	ED	122
Y1 – Y4	„Jahresschaltuhr: Y1 – Y8“	lesen	91	123
Z1 – Z3	„Masterreset: Z1 – Z3“	lesen	93	124
H1 – H4	„Wochenschaltuhr: $\text{H}1 - \text{H}8$ “	lesen	90	125

**Analogwertvergleichler/Schwellwertvergleichler:
A1 – A16**

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

Telegrammaufbau

Byte		Bedeutung	Wert (hex), sendet	
Master	Slave		Master	Slave
		Attribute ID: Lesen	88	–
		Antwort:		
		Lesen erfolgreich	–	C2
		Befehl zurückgewiesen	–	C0 ¹⁾
0	1	Len	01	01
1	2	Type	8B	8B
2	3	Index	00	00
3	4	Data 1 (Low Byte)	00	→ Tabelle 43
4	5	Data 2 (Low Byte)	00	→ Tabelle 43
5 – 6	6 – 7	Data 3 – 4	00	00

1) Mögliche Ursachen → Seite 144

Tabelle 43: Byte 3 bis 4 (Master) bzw. Byte 4 bis 5 (Slave):
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		Master	Slave
		Attribute ID: Lesen	88	–
		Antwort:		
		Lesen erfolgreich	–	C2
		Befehl zurückgewiesen	–	C0 ¹⁾
0	1	Len	01	01
1	2	Type	EE	EE
2	3	Index	00	00
3	4	Data 1 (Low Byte)	00	→ Tabelle 44
4	5	Data 2 (Low Byte)	00	→ Tabelle 44
5 – 6	6 – 7	Data 3 – 4	00	00

1) Mögliche Ursachen → Seite 144

Tabelle 44: Byte 3 bis 4 (Master) bzw. Byte 4 bis 5 (Slave):
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		Master	Slave
		Attribute ID: Lesen	88	–
		Antwort:		
		Lesen erfolgreich	–	C2
		Befehl zurückgewiesen	–	C0 ¹⁾
0	1	Len	01	01
1	2	Type	94	94
2	3	Index	00	00
3	4	Data 1 (Low Byte)	00	→ Tabelle 45
4	5	Data 2 (High Byte)	00	→ Tabelle 45
5 – 6	6 – 7	Data 3 – 4	00	00

1) Mögliche Ursachen → Seite 144

Tabelle 45: Byte 3 bis 4 (Master) bzw. Byte 4 bis 5 (Slave): 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		Master	Slave
		Attribute ID: Lesen	88	–
		Antwort:		
		Lesen erfolgreich	–	C2
		Befehl zurückgewiesen	–	C0 ¹⁾
0	1	Len	02	02
1	2	Type	84	84
2	3	Index	00	00
3	4	Data 1 (Low Byte)	00	→ Tabelle 46
4	5	Data 2 (High Byte)	00	→ Tabelle 46
5 – 6	6 – 7	Data 3 – 4	00	00

1) Mögliche Ursachen → Seite 144

Tabelle 46: Byte 3 bis 4 (Master) bzw. Byte 4 bis 5 (Slave):
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 DeviceNet auslesen. Hierbei wird der 16-Bitwert im Intelformat übergeben (Low Byte first).

Telegrammaufbau

Byte		Bedeutung	Wert (hex), sendet	
Master	Slave		Master	Slave
		Attribute ID: Lesen	88	–
		Antwort:		
		Lesen erfolgreich	–	C2
		Befehl zurückgewiesen	–	C0 ¹⁾
0	1	Len	02	02
1	2	Type	8C	8C
2	3	Index	00 – 03 ²⁾	00 – 03 ²⁾
3	4	Data 1 (Low Byte)	00	→ Tabelle 47
4	5	Data 2 (High Byte)	00	→ Tabelle 47
5 – 6	6 – 7	Data 3 – 4	00	00

1) Mögliche Ursachen → Seite 144

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 47: Beispiel-Telegramm zum Lesen des Wertes am Analog-Eingang „1“

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

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

Merker: M1 – M16/N1 – N16 schreiben

Telegrammaufbau

Byte		Bedeutung	Wert (hex), sendet	
Master	Slave		Master	Slave
		Attribute ID: Schreiben	8C	–
		0	Antwort:	
			Schreiben erfolgreich	–
			Befehl zurückgewiesen	C0 ¹⁾
0	1	Len	01	01
1	2	Type ²⁾		
			bei Merker M	86
			bei Merker N	87
2	3	Index ²⁾	00 – 0F	00 – 0F
3	4	Data 1 (Low Byte) ³⁾	00/01	00/01
4 – 6	5 – 7	Data 2 – 4	00	00

- 1) Mögliche Ursachen → Seite 144
- 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		Master	Slave
		Attribute ID: Schreiben	8C	–
	0	Antwort:		
		Schreiben erfolgreich	–	C1
		Befehl zurückgewiesen	–	C0 ¹⁾
0	1	Len	01	01
1	2	Type		
		Merker M	86	86
2	3	Index	0C	0C
3	4	Data 1	01	00
4 – 6	5 – 7	Data 2 – 4	00	00

1) Mögliche Ursachen → Seite 144

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		Master	Slave
		Attribute ID: Lesen	88	–
		Antwort:		
		Lesen erfolgreich	–	C2
		Befehl zurückgewiesen	–	C0 ¹⁾
0	1	Len	01	01
1	2	Type		
		Merker M	86	86
		Merker N	87	87
2	3	Index ²⁾	00	00
3	4	Data 1 (Low Byte)	00	→ Tabelle 48
4	5	Data 2 (Low Byte)	00	→ Tabelle 48
5 – 6	6 – 7	Data 3 – 4	00	00

- 1) Mögliche Ursachen → Seite 144
- 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 48: Byte 3 bis 4 (Master) bzw. Byte 4 bis 5 (Slave):
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		Master	Slave
		Attribute ID: Lesen	88	-
	0	Antwort:		
		Lesen erfolgreich	-	C2
		Befehl zurückgewiesen	-	C0 ¹⁾
0	1	Len	01	01
1	2	Type		
		Merker N	87	87
2	3	Index	00	00
3	4	Data 1 (Low Byte)	00	04
4	5	Data 2 (Low Byte)	00	84
5 – 6	6 – 7	Data 3 – 4	00	00

1) Mögliche Ursachen → Seite 144

Die Merker N3, N11 und N16 sind gesetzt.

Betriebsstundenzähler: 01 – 04

Mit den nachfolgenden Befehlen lesen Sie den logischen Status der einzelnen Betriebsstundenzähler 01 – 04 aus.

Telegrammaufbau

Byte		Bedeutung	Wert (hex), sendet	
Master	Slave		Master	Slave
		Attribute ID: Lesen	88	–
	0	Antwort:		
		Lesen erfolgreich	–	C2
		Befehl zurückgewiesen	–	C0 ¹⁾
0	1	Len	01	01
1	2	Type	EF	EF
2	3	Index	00	00
3	4	Data 1 (Low Byte)	00	→ Tabelle 49
4 – 6	5 – 7	Data 2 – 4	00	00

1) Mögliche Ursachen → Seite 144

Tabelle 49: Byte 3 (Master) bzw. Byte 4 (Slave): Data 1

Data 1	Bit 7	6	5	4	3	2	1	0
01								0/1
02							0/1	
03						0/1		
04					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		Master	Slave
		Attribute ID: Lesen	88	–
	0	Antwort:		
		Lesen erfolgreich	–	C2
		Befehl zurückgewiesen	–	C0 ¹⁾
0	1	Len	01	01
1	2	Type	8A	8A
2	3	Index	00	00
3	4	Data 1 (Low Byte)	00	→ Tabelle 50
4 – 6	5 – 7	Data 2 – 4	00	00

1) Mögliche Ursachen → Seite 144

Tabelle 50: Byte 3 (Master) bzw. Byte 4 (Slave): 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 DeviceNet-Feldbus direkt lesen.

Telegrammaufbau

Byte		Bedeutung	Wert (hex), sendet	
Master	Slave		Master	Slave
		Attribute ID: Lesen	88	–
		Antwort:		
0		Lesen erfolgreich	–	C2
		Befehl zurückgewiesen	–	C0 ¹⁾
0	1	Len	01	01
1	2	Type	85	85
2	3	Index	00	00
3	4	Data 1 (Low Byte)	00	→ Tabelle 51
4 – 6	5 – 7	Data 2 – 4	00	00

1) Mögliche Ursachen → Seite 144

Tabelle 51: 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		Master	Slave
		Attribute ID: Lesen	88	–
	0	Antwort:		
		Lesen erfolgreich	–	C2
		Befehl zurückgewiesen	–	C0 ¹⁾
0	1	Len	01	01
1	2	Type		
		bei R-Daten	88	88
		bei S-Daten	89	89
2	3	Index	00	00
3	4	Data 1 (Low Byte)	00	→ Tabelle 52
4	5	Data 2 (Low Byte)	00	→ Tabelle 52
5 – 6	6 – 7	Data 3 – 4	00	00

1) Mögliche Ursachen → Seite 144

Tabelle 52: Byte 3 – 4 (Master) bzw. Byte 4 bis 5 (Slave):
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		Master	Slave
		Attribute ID: Lesen	88	–
		Antwort:		
		Lesen erfolgreich	–	C2
		Befehl zurückgewiesen	–	C0 ¹⁾
0	1	Len	01	01
1	2	Type	ED	ED
2	3	Index	00	00
3	4	Data 1 (Low Byte)	00	→ Tabelle 53
4	5	Data 2 (Low Byte)	00	→ Tabelle 53
5 – 6	6 – 7	Data 3 – 4	00	00

1) Mögliche Ursachen → Seite 144

Tabelle 53: Byte 3 – 4 (Master) bzw. Byte 4 bis 5 (Slave): 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		Master	Slave
		Attribute ID: Lesen	88	–
	0	Antwort:		
		Lesen erfolgreich	–	C2
		Befehl zurückgewiesen	–	C0 ¹⁾
0	1	Len	01	01
1	2	Type	91	91
2	3	Index	00	00
3	4	Data 1 (Low Byte)	00	→ Tabelle 54
4 – 6	5 – 7	Data 2 – 4	00	00

1) Mögliche Ursachen → Seite 144

Tabelle 54: Byte 3 (Master) bzw. Byte 4 (Slave): 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		Master	Slave
		Attribute ID: Lesen	88	–
	0	Antwort:		
		Lesen erfolgreich	–	C2
		Befehl zurückgewiesen	–	C0 ¹⁾
0	1	Len	01	01
1	2	Type	93	93
2	3	Index	00	00
3	4	Data 1 (Low Byte)	00	→ Tabelle 55
4 – 6	5 – 7	Data 2 – 4	00	00

1) Mögliche Ursachen → Seite 144

Tabelle 55: Byte 3 (Master) bzw. Byte 4 (Slave): 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: 01 – 08

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

Telegrammaufbau

Byte		Bedeutung	Wert (hex), sendet	
Master	Slave		Master	Slave
		Attribute ID: Lesen	88	–
		Antwort:		
		Lesen erfolgreich	–	C2
		Befehl zurückgewiesen	–	C0 ¹⁾
0	1	Len	01	01
1	2	Type	90	90
2	3	Index	00	00
3	4	Data 1 (Low Byte)	00	→ Tabelle 56
4 – 6	5 – 7	Data 2 – 4	00	00

1) Mögliche Ursachen → Seite 144

Tabelle 56: Byte 3 (Master) bzw. Byte 4 (Slave): 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} → 03 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	127
C1 – C16	„Zählerrelais: C1 – C16“	Lesen/schreiben	8F	130
O1 – O4	„Betriebsstundenzähler: O1 – O4“	Lesen/schreiben	92	133
T1 – T16	„Zeitrelais: T1 – T16“	Lesen/schreiben	8E	135
Y1 – Y8	„Jahresschaltuhr: Y1 – Y8“	Lesen/schreiben	A2	138
W1 – W8	„Wochenschaltuhr: W1 – W8“	Lesen/schreiben	A1	141

Analogwertvergleicher/Schwellwertschalter: A1 – A16

Telegrammaufbau

Byte		Bedeutung	Wert (hex), sendet	
Master	Slave		Master	Slave
Attribute ID:				
		Lesen	89	–
		Schreiben	8D	–
		Antwort:		
		Lesen erfolgreich	–	C2
		Schreiben erfolgreich	–	C1
		Befehl zurückgewiesen	–	C0 ¹⁾
0	1	Type	8D	8D
1	2	Instanz ²⁾	00 – 0F	00 – 0F
2	3	Index	→ Tabelle 57	→ Tabelle 57
3 – 6	4 – 7	Data 1 – 4	abhängig vom Index, → Tabelle 58	abhängig vom Index, → Tabelle 58

- 1) Mögliche Ursachen → Seite 144
- 2) easy stellt 16 Analogwertvergleicher A1 bis A16 zur Verfügung. Diese können über die Instanz (0 – F) angesprochen werden.

Tabelle 57: Operandenübersicht

Index (hex)	Operand		Lesen	Schreiben
00	Parameter → Tabelle 58		×	
01	Kontrollbyte → Tabelle 59		×	
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	Schalthyserese 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: $5327_{\text{dez}} = 14CF_{\text{hex}} \rightarrow \text{Data 1} = 0xCF, \text{Data 2} = 0x14$

Tabelle 58: 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 (\geq)														0	1	0	
LE (\leq)														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 $(I1 \times F1) + OS = (I2 \times F2) + HY$ erfüllt ist.

Tabelle 59: 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		Master	Slave
Attribute ID:				
		Lesen	89	–
		Schreiben	8D	–
		0 Antwort:		
		Lesen erfolgreich	–	C2
		Schreiben erfolgreich	–	C1
		Befehl zurückgewiesen	–	C0 ¹⁾
0	1	Type	8F	8F
1	2	Instanz ²⁾	00 – 0F	00 – 0F
2	3	Index	→ Tabelle 60	→ Tabelle 60
3 – 6	4 – 7	Data 1 – 4	abhängig vom Index, → Tabelle 61	abhängig vom Index, → Tabelle 61

- 1) Mögliche Ursachen → Seite 144
- 2) easy stellt 16 Zählerrelais C1 bis C16 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	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 61: 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 62: 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		Master	Slave
		Befehl: Lesen	89	–
	0	Antwort: Lesen erfolgreich	–	C2
0	1	Type	8F	8F
1	2	Instanz	02	02
2	3	Index	02	02
3	4	Data1	00	12
4	5	Data 2	00	03
5	6	Data 3	00	00
6	7	Data 4	00	00

Erklärung:

Data 1 = 12

Data 2 = 03

→ resultierender 16-Bit-Wert = $0312_{\text{hex}} = 786_{\text{dez}}$

Zählerstand = 786

Betriebsstundenzähler: O1 – O4

Telegrammaufbau

Byte		Bedeutung	Wert (hex), sendet	
Master	Slave		Master	Slave
Attribute ID:				
		Lesen	89	–
		Schreiben	8D	–
		Antwort:		
0		Lesen erfolgreich	–	C2
		Schreiben erfolgreich	–	C1
		Befehl zurückgewiesen	–	C0 ¹⁾
0	1	Type	92	92
1	2	Instanz ²⁾	00 – 03	00 – 03
2	3	Index	→ Tabelle 63	→ Tabelle 63
3 – 6	4 – 7	Data 1 – 4	abhängig vom Index, → Tabelle 64	abhängig vom Index, → Tabelle 64

- 1) Mögliche Ursachen → Seite 144
- 2) easy stellt 4 Betriebsstundenzähler O1 bis O4 zur Verfügung. Diese können über die Instanz (0 – 3) angesprochen werden.

Tabelle 63: Operandenübersicht

Index (hex)	Operand		Lesen	Schreiben
00	Parameter → Tabelle 64		×	
01	Kontrollbyte → Tabelle 65		×	
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 64: 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 65: 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		Master	Slave
Attribute ID:				
		Lesen	89	–
		Schreiben	8D	–
		Antwort:		
		Lesen erfolgreich	–	C2
		Schreiben erfolgreich	–	C1
		Befehl zurückgewiesen	–	C0 ¹⁾
0	1	Type	8E	8E
1	2	Instanz ²⁾	00 – 0F	00 – 0F
2	3	Index	→ Tabelle 66	→ Tabelle 66
3 – 6	4 – 7	Data 1 – 4	abhängig vom Index, → Tabelle 67	abhängig vom Index, → Tabelle 67

- 1) Mögliche Ursachen → Seite 144
- 2) easy stellt 16 Zeitrelais T1 bis T16 zur Verfügung. Diese können über die Instanz (0 – F) angesprochen werden.

Tabelle 66: Operandenübersicht

Index (hex)	Operand		Lesen	Schreiben
00	Parameter → Tabelle 67		×	
01	Kontrollbyte → Tabelle 68		×	
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 67: 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 68: 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		Master	Slave
Attribute ID:				
		Lesen	89	–
		Schreiben	8D	–
		Antwort:		
		Lesen erfolgreich	–	C2
		Schreiben erfolgreich	–	C1
		Befehl zurückgewiesen	–	C0 ¹⁾
0	1	Type	A2	A2
1	2	Instanz ²⁾	00 – 07	00 – 07
2	3	Index	→ Tabelle 69	→ Tabelle 69
3 – 6	4 – 7	Data 1 – 4	abhängig vom Index, → Tabelle 70	abhängig vom Index, → Tabelle 70

1) Mögliche Ursachen → Seite 144

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

Tabelle 69: Operandenübersicht

Index (hex)	Operand	Lesen	Schreiben
00	Parameter → Tabelle 70	×	
01	Kontrollbyte → Tabelle 71	×	
	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 70: 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 71: 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: 01 – 08

Telegrammaufbau

Byte		Bedeutung	Wert (hex), sendet	
Master	Slave		Master	Slave
Attribute ID:				
		Lesen	89	–
		Schreiben	8D	–
		Antwort:		
		Lesen erfolgreich	–	C2
		Schreiben erfolgreich	–	C1
		Befehl zurückgewiesen	–	C0 ¹⁾
0	1	Type	A1	A1
1	2	Instanz ²⁾	00 – 07	00 – 07
2	3	Index	→ Tabelle 72	→ Tabelle 72
3 – 6	4 – 7	Data 1 – 4	abhängig vom Index, → Tabelle 73	abhängig vom Index, → Tabelle 73

1) Mögliche Ursachen → Seite 144

2) easy stellt 8 Wochenschaltuhren 01 bis 08 zur Verfügung. Diese können über die Instanz (0 – 7) angesprochen werden.

Tabelle 72: Operandenübersicht

Index (hex)	Operand	Lesen	Schreiben
00	Parameter → Tabelle 73	×	
01	Kontrollbyte → Tabelle 74	×	
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 73: 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 74: 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 75

Tabelle 75: Fehlercodes

Fehlercode	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 Geräte-modus. 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 unbekannt Parameter anzusteuern.
0xF1	Nicht erlaubter Wert

8 Steuerbefehle für easy800/MFD

Datenaustauschverfahren

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

- „Datum und Uhrzeit lesen/schreiben“ (Seite 148)
- „Abbilddaten lesen/schreiben“ (Seite 154)
- „Funktionsbaustein-Daten lesen/schreiben“ (Seite 174).

Hierzu wird in der Mastersteuerung auf das Nachrichtenübertragungsprotokoll der expliziten Nachrichten zurückgegriffen. Die Parameter werden alle über den Service Code 32_{hex} angesprochen. Hierbei erfolgt die Differenzierung der unterschiedliche Parameter über die zugeordnete Attribute ID.

Service Code	Objekt Adresse	
	Class ID	Instance ID
32 _{hex}	64 _{hex}	01 _{hex}



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 DeviceNet übertragen werden, ist ein Datenaustauschverfahren notwendig.



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

Bei der Kommunikation zwischen den Teilnehmern stößt der Master den Datenaustausch mit einem Steuerbefehl an. Der Slave reagiert auf die Anfrage immer mit einer Antwort. Die Antwort gibt Aufschluss darüber, ob der Datenaustausch ausgeführt wurde oder nicht. Konnte der Datenaustausch nicht realisiert werden, wird ein Fehlercode zurückgegeben. Dieser ist von der ODVA genau definiert.

Versionsgeschichte

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

Auswirkung auf easyLink	easy800, Geräteversion			
	ab 01	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		Master	Slave
Attribute ID				
		Lesen	93	–
		Schreiben	B3	–
	0	Antwort		
		Lesen erfolgreich	–	C2
		Schreiben erfolgreich	–	C1
		Befehl zurückgewiesen	–	C0
0	1	Len	05	05
1	2	Index	00	00
2 – 6	3 – 7	Data 1 – 5		
		Beim Lesen	00	→ Tabelle 76
		Beim Schreiben	→ Tabelle 76	00

Tabelle 76: Byte 2 bis 6 (Master) bzw. Byte 3 bis 7 (Slave):
Data 1 bis 5

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

Winter-/Sommerzeit, Zeitumstellung

Telegrammaufbau

Byte		Bedeutung	Wert (hex), sendet	
Master	Slave		Master	Slave
Attribute ID				
		Lesen	93	–
		Schreiben	B3	–
		0	Antwort	
			Lesen erfolgreich	–
			Schreiben erfolgreich	–
			Befehl zurückgewiesen	–
0	1	Len	05	05
1	2	Index		
		01: Sommer-/Winterzeit	→ Tabelle 77	→ Tabelle 77
		02: Winterzeit (nach „Area“ = Regel ¹⁾)	→ Tabelle 78	→ Tabelle 78
2 – 6	3 – 7	Data 1 – 5		
		Beim Lesen	00	abhängig vom Index, → Tabelle 77, 78
		Beim Schreiben	abhängig vom Index, → Tabelle 77, 78	00

1) Detaillierte Einstellmöglichkeit für easy800/MFD ab Version 05

Tabelle 77: Index 01 – Sommer-/Winterzeit-Umschaltung

Byte		Inhalt		Wert (hex)
Master	Slave			
2	3	Data 1	Area	
			keine	00
			manuell	01
			automatische EU	02
			automatische GB	03
			automatische US	04
			Regel ¹⁾	05
für „Area“ = „manuell“:				
3	4	Data 2	Set Sommerzeit Tag 1 bis 28, 29, 30, 31 (abh. von Monat und Jahr)	00 – 3B
4	5	Data 3	Set Sommerzeit Monat (1 – 12)	01 – 1F
5	6	Data 4	Set Winterzeit Tag 1 bis 28, 29, 30, 31 (abh. von Monat und Jahr)	01 – 0C
6	7	Data 5	Set Winterzeit Monat (1 – 12)	00 – 63
für „Area“ = „Regel“ ¹⁾ :				
3 – 6	4 – 7	Data 2 – 5	Schaltregel Sommerzeit	→ Tabelle 79

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

Tabelle 78: Index 02 – Winterzeit
(nur gültig, wenn Area = „Regel“ gewählt worden ist)

Byte		Inhalt		Wert (hex)
Master	Slave			
2	3	Data 1	Area = Regel	01
3 – 6	4 – 7	Data 2 – 5	Schaltregel Winterzeit	→ Tabelle 79

Bitfeld „Schaltregel“



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

Tabelle 79: Bitfeld Schaltregel

Bit	Data 5						Data 4						Data 3						Data 2												
	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
	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		Master	Slave
		Attribute ID: Schreiben	B3	–
	0	Antwort: Schreiben erfolgreich	–	C1
0	1	Len	05	05
1	2	Index	00	00
2	3	Data 1 (Stunde)	0E	00
3	4	Data 2 (Minute)	24	00
4	5	Data 3 (Tag)	17	00
5	6	Data 4 (Monat)	05	00
6	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 61 auch für easy700.

Übersicht

Operanden	Bedeutung	Lesen/ schreiben	Befehl (hex)	Seite
IA1 – IA4	„Lokale Analog-Eingänge: IA1 – IA4“	lesen	02	155
ID1 – ID16	„Lokale Diagnose: ID1 – ID16“	lesen	03	157
IW0	„Lokale Eingänge: IW0“	lesen	01	159
IW1 – IW8	„Eingänge der Netzwerkteilnehmer: IW1 – IW8“	lesen	01	161
M...	„Merker: M...“	lesen/ schreiben	0B – 0E	162
P1 – P4	„Lokale P-Tasten: P1 – P4“	lesen	06	165
QA1	„Lokaler Analog-Ausgang: QA1“	lesen/ schreiben	05	167
QW0, QW1 – QW8	„Lokale Ausgänge: QW0/ Ausgänge der Netz- werkteilnehmer: QW1 – QW8“	lesen/ schreiben	04	168
R1 – R16 S1 – S8	„Ein-/Ausgänge von easyLink: RW/SW“	lesen	07/09	170
RN1 – RN32 SN1 – SN32	„Receive-Data Netz: RN1 – RN32/ Send-Data Netz: SN1 – SN32“	lesen	08/0A	172

Lokale Analog-Eingänge: IA1 – IA4

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

Telegrammaufbau

Byte		Bedeutung	Wert (hex), sendet	
Master	Slave		Master	Slave
		Attribute ID: Lesen	91	–
	0	Antwort: Lesen erfolgreich	–	C2
		Befehl zurückgewiesen	–	C0
0	1	Len	02	02
1	2	Type	02	02
2	3	Index	01 – 04 ¹⁾	01 – 04 ¹⁾
3	4	Data 1 (Low Byte)	00	→ Beispiel auf Seite 156
4	5	Data 2 (High Byte)	00	
5 – 6	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		Master	Slave
		Attribute ID: Lesen	91	–
	0	Antwort: Lesen erfolgreich	–	C2
0	1	Len	02	02
1	2	Type	02	02
2	3	Index	01 ¹⁾	01 ¹⁾
3	4	Data 1	00	D9
4	5	Data 2	00	02
5	6	Data 3	00	00
6	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		Master	Slave
		Attribute ID: Lesen	91	–
	0	Antwort:		
		Lesen erfolgreich	–	C2
		Befehl zurückgewiesen	–	C0
0	1	Len	02	02
1	2	Type	03	03
2	3	Index	00	00
3	4	Data 1 (Low Byte)	00	→ Tabelle 80
4	5	Data 2 (High Byte)	00	→ Tabelle 80
5 – 6	6 – 7	Data 3 – 4	00	00

Tabelle 80: 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: IWO

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		Master	Slave
		Attribute ID: Lesen	91	–
		Antwort:		
		Lesen erfolgreich	–	C2
		Befehl zurückgewiesen	–	C0
0	1	Len	02	02
1	2	Type	01	01
2	3	Index	00	00
3	4	Data 1 (Low Byte)	00	→ Tabelle 81
4	5	Data 2 (High Byte)	00	→ Tabelle 81
5 – 6	6 – 7	Data 3 – 4	00	00

Tabelle 81: Byte 4 bis 5: Data 1 bis 2

Data 1	Bit	7	6	5	4	3	2	1	0
11									0/1
12									0/1
...					...				
18			0/1						
Data 2	Bit	7	6	5	4	3	2	1	0
19									0/1
110									0/1
...					...				
116			0/1						

Beispiel: Lokale Eingänge IWO lesen

Byte		Bedeutung	Wert (hex), sendet	
Master	Slave		Master	Slave
		Attribute ID: Lesen	91	–
	0	Antwort: Lesen erfolgreich	–	C2
0	1	Len	02	02
1	2	Type	01	01
2	3	Index	00	00
3	4	Data 1	00	C4
4	5	Data 2	00	02
5	6	Data 3	00	00
6	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		Master	Slave
		Attribute ID: Lesen	91	–
	0	Antwort:		
		Lesen erfolgreich	–	C2
		Befehl zurückgewiesen	–	C0
0	1	Len	02	02
1	2	Type	01	01
2	3	Index	01 – 08 ¹⁾	01 – 08 ¹⁾
3	4	Data 1 (Low Byte)	00	→ Tabelle 81 auf Seite 159.
4	5	Data 2 (High Byte)	00	
5 – 6	6 – 7	Data 3 – 4	00	00

1) entspricht Adresse des Netzteilnehmers

Merker: M...

Byte		Bedeutung	Wert (hex), sendet	
Master	Slave		Master	Slave
Attribute ID				
		Lesen	91	–
		Schreiben	B1	–
		0	Antwort	
		Lesen erfolgreich	–	C2
		Schreiben erfolgreich	–	C1
		Befehl zurückgewiesen	–	C0
0	1	Len	→ Tabelle 82	→ Tabelle 82
1	2	Type		
2	3	Index		
3 – 6	4 – 7	Data 1 – 4		
		Beim Lesen	00	→ „Beispiel 1: Merker-Bit setzen/rücksetzen“ auf Seite 164
		Beim Schreiben	→ „Beispiel 2: Merkerwort schreiben“ auf Seite 164	00

Tabelle 82: Byte 0 bis 2 (Master) bzw.: Byte 1 bis 3 Slave:
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		Master	Slave
		Attribute ID: Schreiben	B1	–
	0	Antwort: Schreiben erfolgreich	–	C1
0	1	Len	01	01
1	2	Type	0B	0B
2	3	Index	3E	3E
3	4	Data 1	01 ¹⁾	00
4 – 6	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: $823_{dec} = 337_{hex} \rightarrow \text{Data 1} = 37_{hex}, \text{Data 2} = 03_{hex}$

Byte		Bedeutung	Wert (hex), sendet	
Master	Slave		Master	Slave
		Attribute ID: Schreiben	B1	–
	0	Antwort: Schreiben erfolgreich	–	C1
0	1	Len	02	01
1	2	Type	0D	0D
2	3	Index	20	20
3	4	Data 1	37	00
4	5	Data 2	03	00
5	6	Data 3	00	00
6	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		Master	Slave
		Attribute ID: Lesen	91	–
	0	Antwort:		
		Lesen erfolgreich	–	C2
		Befehl zurückgewiesen	–	C0
0	1	Len	02	02
1	2	Type	06	06
2	3	Index	00	00
3	4	Data 1 (Low Byte)	00	→ Tabelle 83
4 – 6	5 – 7	Data 2 – 4	00	00

Tabelle 83: 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							

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

Die lokalen Ausgänge können Sie über den DeviceNet 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 154.

Telegrammaufbau

Byte		Bedeutung	Wert (hex), sendet	
Master	Slave		Master	Slave
Befehl				
		Lesen	91	–
		Schreiben¹⁾	B1	–
	0	Antwort		
		Lesen erfolgreich	–	C2
		Schreiben erfolgreich	–	C1
		Befehl zurückgewiesen	–	C0
0	1	Len	02	02
1	2	Type	04	04
2	3	Index ²⁾	00/01 – 08	00/01 – 08
3	4	Data 1		
		Beim Lesen	00	→ Tabelle 80
		Beim Schreiben	→ Tabelle 84	00
4 – 6	5 – 7	Data 2 – 4	00	00

1) Schreiben ist erst ab easy800, Geräteversion 04, möglich, → Abschnitt „Datum und Uhrzeit lesen/schreiben“ auf Seite 148.

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

Tabelle 84: 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		Master	Slave
		Attribute ID: Lesen	91	–
		Antwort:		
	0	Lesen erfolgreich	–	C2
		Befehl zurückgewiesen	–	C0
0	1	Len	02	02
1	2	Type	bei RW: 07	bei RW: 07
2			bei SW: 09	bei SW: 09
	3	Index	00/01 – 08 ¹⁾	00/01 – 08 ¹⁾
3	4	Data 1 (Low Byte)	00	→ Tabelle 85
4	5	Data 2 (High Byte)	00	→ Tabelle 85
5 – 6	6 – 7	Data 3 – 4	00	00

1) 00 = Lokaler Ein-/Ausgang

01 – 08 = Adresse des Netzwerkteilnehmers (NET-ID 1 – 8)

Tabelle 85: 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		Master	Slave
		Attribute ID: Lesen	91	–
	0	Antwort:		
		Lesen erfolgreich	–	C2
		Befehl zurückgewiesen	–	C0
0	1	Len	04	04
1	2	Type	bei RN1 – RN32: 08 bei SN1 – SN32: 0A	bei RN1 – RN32: 08 bei SN1 – SN32: 0A
2	3	Index	01 – 08 ¹⁾	01 – 08 ¹⁾
3 – 6	4 – 7	Data 1 – 4	00	→ Tabelle 86

1) entspricht NET-ID

Tabelle 86: 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 Funktionsbausteinen, 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: $-2\,147\,483\,648 \dots 0 \dots +2\,147\,483\,647$

Übersicht

Operanden	Bedeutung	lesen/ schreiben	Type (hex)	Seite
A01 – A32	„Analogwertvergleich: A01 – A32“	lesen/schreiben	11	177
AR01 – AR32	„Arithmetikbaustein: AR01 – AR32“	lesen/schreiben	12	179
BC01 – BC32	„Block Compare: BC01 – BC32“	lesen/schreiben	25	181
BT01 – BT32	„Block Transfer: BT01 – BT32“	lesen/schreiben	26	183
BV01 – BV32	„Boolsche Verknüpfung: BV01 – BV32“	lesen/schreiben	13	185
C01 – C32	„Zähler: C01 – C32“	lesen/schreiben	14	187
CF01 – CF04	„Frequenzzähler: CF01 – CF04“	lesen/schreiben	15	189
CH01 – CH04	„Schneller Zähler: CH01 – CH04“	lesen/schreiben	16	191
CI01 – CI02	„Inkrementalzähler: CI01 – CI02“	lesen/schreiben	17	193
CP01 – CP32	„Vergleicher: CP01 – CP32“	lesen/schreiben	18	195
D01 – D32	„Textausgabe-Baustein: D01 – D32“	lesen/schreiben	19	197
DB01 – DB32	„Datenbaustein: DB01 – DB32“	lesen/schreiben	1A	200
DC01 – DC32	„PID-Regler: DC01 – DC32“	lesen/schreiben	27	202
DG01 – DG16	„Diagnose DG01...DG16“	lesen	39	205
FT01 – FT32	„Signalglättungsfilter: FT01 – FT32“	lesen/schreiben	28	207
GT01 – GT32	„Empfang von Netzdaten: GT01 – GT32“	lesen	1B	209
HW01 – HW32	„Wochenzeitschaltuhr: HW01 – HW32“	lesen	1C	211
HY01 – HY32	„Jahreszeitschaltuhr: HY01 – HY32“	lesen	1D	214
JC01 – JC32	„Bedingter Sprung JC01...JC32“	lesen	2F	217
LS01 – LS32	„Wertskalierung: LS01 – LS32“	lesen/schreiben	29	219
MR01 – MR32	„Masterreset: MR01 – MR32“	lesen	0F	221
MX01 – MX32	„Datenmultiplexer MX01...MX32“	lesen/schreiben	31	223
NC01 – NC32	„Zahlenwandler: NC01 – NC32“	lesen/schreiben	2A	225
OT01 – OT04	„Betriebsstundenzähler: OT01 – OT04“	lesen/schreiben	1E	227
PO01 – PO02	„Impulsausgabe PO01...PO02“	lesen/schreiben	32	229
PT01 – PT32	„Senden von Netzdaten: PT01 – PT32“	lesen	1F	232
PW01 – PW02	„Pulsweitenmodulation: PW01 – PW02“	lesen/schreiben	2B	234

Operanden	Bedeutung	lesen/ schreiben	Type (hex)	Seite
SC01	„Uhr synchronisieren: SC01“	lesen	20	236
SP01 - SP32	„Serielle Ausgabe SP01...SP32“	lesen	35	237
SR01 - SR32	„Schieberegister SR01...SR32“	lesen	33	239
ST01	„Sollzykluszeit: ST01“	lesen/schreiben	2C	242
T01 – T32	„Zeitrelais: T01 – T32“	lesen/schreiben	21	244
TB01 – TB32	„Tabellenfunktion TB01...TB32“	lesen/schreiben	34	247
VC01 – VC32	„Wertbegrenzung: VC01 – VC32“	lesen/schreiben	2D	249

Analogwertvergleich: A01 – A32

Telegrammaufbau

Byte		Bedeutung	Wert (hex), sendet	
Master	Slave		Master	Slave
Attribute ID				
		Lesen	92	–
		Schreiben	B2	–
		Antwort:		
		Lesen erfolgreich	–	C2
		Schreiben erfolgreich	–	C1
		Befehl zurückgewiesen	–	C0
0	1	Type	11	11
1	2	Instanz	01 – 20	01 – 20
2	3	Index	→ Tabelle 87	→ Tabelle 87
3 – 6	4 – 7	Data 1 – 4	00	abhängig vom Index, → Tabelle 88, 89

Tabelle 87: Operandenübersicht

Index (hex)	Operand		lesen	schreiben
00	Bit-IO, → Tabelle 88		×	
01	Mode, → Tabelle 89		×	
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 88: 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 89: 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		Master	Slave
Attribute ID				
		Lesen	92	–
		Schreiben	B2	–
		Antwort:		
0		Lesen erfolgreich	–	C2
		Schreiben erfolgreich	–	C1
		Befehl zurückgewiesen	–	C0
0	1	Type	12	12
1	2	Instanz	01 – 20	01 – 20
2	3	Index	→ Tabelle 90	→ Tabelle 90
3 – 6		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	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 91: 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 92: Index 1 – Mode

Data 1 (hex)		
00	ADD	Addieren ($I1 + I2 = QV$)
01	SUB	Subtrahieren ($I1 - I2 = QV$)
02	MUL	Multiplizieren ($I1 \times I2 = QV$)
03	DIV	Dividieren ($I1 : I2 = QV$)

Block Compare: BC01 – BC32

Telegrammaufbau

Byte		Bedeutung	Wert (hex), sendet	
Master	Slave		Master	Slave
Attribute ID				
		Lesen	92	–
		Schreiben	B2	–
		Antwort:		
		Lesen erfolgreich	–	C2
		Schreiben erfolgreich	–	C1
		Befehl zurückgewiesen	–	C0
0	1	Type	25	25
1	2	Instanz	01 – 20	01 – 20
2	3	Index	→ Tabelle 93	→ Tabelle 93
3 – 6	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	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 94: 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 95: 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		Master	Slave
Attribute ID				
		Lesen	92	–
		Schreiben	B2	–
		0 Antwort:		
		Lesen erfolgreich	–	C2
		Schreiben erfolgreich	–	C1
		Befehl zurückgewiesen	–	C0
0	1	Type	26	26
1	2	Instanz	01 – 20	01 – 20
2	3	Index	→ Tabelle 96	→ Tabelle 96
3 – 6	4 – 7	Data 1 – 4		
		beim Lesen	00	abhängig vom Index, → Tabelle 97, 98
		beim Schreiben	abhängig vom Index, → Tabelle 97, 98	00

Tabelle 96: Operandenübersicht

Index (hex)	Operand	lesen	schreiben
00	Bit-IO, → Tabelle 97	×	
01	Mode, → Tabelle 98	×	
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 beschriftet 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 97: 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 98: Index 1 – Mode

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

Boolsche Verknüpfung: BV01 – BV32

Telegrammaufbau

Byte		Bedeutung	Wert (hex), sendet	
Master	Slave		Master	Slave
Attribute ID				
		Lesen	92	–
		Schreiben	B2	–
		Antwort:		
		Lesen erfolgreich	–	C2
		Schreiben erfolgreich	–	C1
		Befehl zurückgewiesen	–	C0
0	1	Type	13	13
1	2	Instanz	01 – 20	01 – 20
2	3	Index	→ Tabelle 99	→ Tabelle 99
3 – 6	4 – 7	Data 1 – 4		
		beim Lesen	00	abhängig vom Index, → Tabelle 100, 101
		beim Schreiben	abhängig vom Index, → Tabelle 100, 101	00

Tabelle 99: Operandenübersicht

Index (hex)	Operand	lesen	schreiben
00	Bit-IO, → Tabelle 100	×	
01	Mode, → Tabelle 101	×	
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 100: 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 101: 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		Master	Slave
Attribute ID				
		Lesen	92	–
		Schreiben	B2	–
		Antwort:		
		Lesen erfolgreich	–	C2
		Schreiben erfolgreich	–	C1
		Befehl zurückgewiesen	–	C0
0	1	Type	14	14
1	2	Instanz	01 – 20	01 – 20
2	3	Index	→ Tabelle 102	→ Tabelle 102
3 – 6	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 –2 147 483 648 bis +2 147 483 647	×	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		–	–	–	–	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 \leq unterer Sollwert
- 8) Overflow: Zustand „1“, wenn Istwert \geq oberer Sollwert

Frequenzzähler: CF01 – CF04

Telegrammaufbau

Byte		Bedeutung	Wert (hex), sendet	
Master	Slave		Master	Slave
		Attribute ID		
		Lesen	92	–
		Schreiben	B2	–
		Antwort:		
0		Lesen erfolgreich	–	C2
		Schreiben erfolgreich	–	C1
		Befehl zurückgewiesen	–	C0
0	1	Type	15	15
1	2	Instanz	01 – 04	01 – 04
2	3	Index	→ Tabelle 104	→ Tabelle 104
3 – 6	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	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 105: 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 \leq unterer Sollwert
- 4) Overflow: Zustand „1“, wenn Istwert \geq oberer Sollwert.

Schneller Zähler: CH01 – CH04

Telegrammaufbau

Byte		Bedeutung	Wert (hex), sendet	
Master	Slave		Master	Slave
Attribute ID				
		Lesen	92	–
		Schreiben	B2	–
		Antwort:		
0		Lesen erfolgreich	–	C2
		Schreiben erfolgreich	–	C1
		Befehl zurückgewiesen	–	C0
0	1	Type	16	16
1	2	Instanz	01 – 04	01 – 04
2	3	Index	→ Tabelle 106	→ Tabelle 106
3 – 6	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	Wert	lesen	schreiben
00	Bit-IO	→ Tabelle 107	×	
01	Mode/Parameter	–	–	–
02	oberer Sollwert SH	im ganzzahligen Bereich von –2 147 483 648 bis +2 147 483 647	×	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 107: 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 \leq unterer Sollwert
- 8) Overflow: Zustand „1“, wenn Istwert \geq unterer Sollwert

Inkrementalzähler: CI01 – CI02

Telegrammaufbau

Byte		Bedeutung	Wert (hex), sendet	
Master	Slave		Master	Slave
		Attribute ID		
		Lesen	92	–
		Schreiben	B2	–
		Antwort:		
0		Lesen erfolgreich	–	C2
		Schreiben erfolgreich	–	C1
		Befehl zurückgewiesen	–	C0
0	1	Type	17	17
1	2	Instanz	01 – 02	01 – 02
2	3	Index	→ Tabelle 108	→ Tabelle 108
3 – 6	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	Wert	lesen	schreiben
00	Bit-IO	→ Tabelle 109	×	
01	Mode/Parameter	–	–	–
02	oberer Sollwert SH	im ganzzahligen Bereich von –2 147 483 648 bis +2 147 483 647	×	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 109: 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 \leq unterer Sollwert
- 7) Overflow: Zustand „1“, wenn Istwert \geq unterer Sollwert

Vergleicher: CP01 – CP32

Telegrammaufbau

Byte		Bedeutung	Wert (hex), sendet	
Master	Slave		Master	Slave
Attribute ID				
		Lesen	92	–
		Schreiben	B2	–
		Antwort:		
		Lesen erfolgreich	–	C2
		Schreiben erfolgreich	–	C1
		Befehl zurückgewiesen	–	C0
0	1	Type	18	18
1	2	Instanz	01 – 20	01 – 20
2	3	Index	→ Tabelle 110	→ Tabelle 110
3 – 6	4 – 7	Data 1 – 4		
		beim Lesen	00	abhängig vom Index, → Tabelle 111
		beim Schreiben	abhängig vom Index, → Tabelle 111	00

Tabelle 110: Operandenübersicht

Index (hex)	Operand	lesen	schreiben
00	Bit-IO, → Tabelle 111	×	
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 111: Index 0 – Bit-IO

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

- 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		Master	Slave
		Attribute ID		
		Lesen	92	–
		Schreiben	B2	–
		Antwort:		
		Lesen erfolgreich	–	C2
		Schreiben erfolgreich	–	C1
		Befehl zurückgewiesen	–	C0
0	1	Type	19	19
1	2	Instanz	01 – 20	01 – 20
2	3	Index	→ Tabelle 112	→ Tabelle 112
3 – 6	4 – 7	Data 1 – 4		
		beim Lesen	00	abhängig vom Index, → Tabelle 113
		beim Schreiben	abhängig vom Index, → Tabelle 113	00

Tabelle 112: Operandenübersicht

Index	Operand	lesen	schreiben
00	Bit-IO, → Tabelle 113	×	
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	×	
27	Skalierung Maximalwert 2	×	

Index	Operand	lesen	schreiben
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 113: 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		Master	Slave
Attribute ID				
		Lesen	92	–
		Schreiben	B2	–
		Antwort:		
0		Lesen erfolgreich	–	C2
		Schreiben erfolgreich	–	C1
		Befehl zurückgewiesen	–	C0
0	1	Type	1A	1A
1	2	Instanz	01 – 20	01 – 20
2	3	Index	→ Tabelle 114	→ Tabelle 114
3 – 6		Data 1 – 4		
		beim Lesen	00	abhängig vom Index, → Tabelle 115
		beim Schreiben	abhängig vom Index, → Tabelle 115	00

Tabelle 114: Operandenübersicht

Index (hex)	Operand	lesen	schreiben
00	Bit-IO, → Tabelle 115	×	
01	Mode/Parameter	–	–
02	Eingangswert: Wert, I1 der beim Triggern des Bausteins an den Ausgang QV weitergegeben wird.	×	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 115: 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		Master	Slave
Attribute ID				
		Lesen	92	–
		Schreiben	B2	–
	0	Antwort:		
		Lesen erfolgreich	–	C2
		Schreiben erfolgreich	–	C1
		Befehl zurückgewiesen	–	C0
0		1	Type	27
1	2	Instanz	01 – 20	01 – 20
2	3	Index	→ Tabelle 116	→ Tabelle 116
3 – 6	4 – 7	Data 1 – 4		
		beim Lesen	00	abhängig vom Index, → Tabelle 117, 118
		beim Schreiben	abhängig vom Index, → Tabelle 117, 118	

Tabelle 116: Operandenübersicht

Index (hex)	Operand		lesen	schreiben
00	Bit-IO, → Tabelle 117		×	
01	Mode, → Tabelle 118		×	
02	Sollwert: -32 768 bis +32 767	I1	×	c ¹⁾
03	Istwert: -32 768 bis +32 767	I2	×	c ¹⁾
04	Proportionalverstärkung [%], Wertebereich: 0 bis 65 535	KP	×	c ¹⁾
05	Nachstellzeit [0,1 s], Wertebereich: 0 bis 65 535	TN	×	c ¹⁾
06	Vorhaltezeit [0,1 s], Wertebereich: 0 bis 65 535	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 117: 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 118: 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		Master	Slave
		Attribute ID Befehl: Lesen	92	–
		Attribute ID Befehl: Schreiben	B2	–
	0	Antwort:		
		Lesen erfolgreich	–	C2
		Schreiben erfolgreich	–	C1
		Befehl zurückgewiesen	–	C0
0	1	Type	39	39
1	2	Instanz	01 - 10	01 - 10
2	3	Index	00 - 03	00 - 03
3 - 6	4 - 7	Data 1 - 4 beim Lesen	00	abhängig vom Index, → Tabelle 119, 120

Tabelle 119: Operandenübersicht

Index (hex)	Daten	Data 1, Data 3, Data 4	Data 2	Read/Write
0	Bit-IO	→ Tabelle 120	–	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 120: 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		Master	Slave
		Attribute ID		
		Lesen	92	–
		Schreiben	B2	–
		Antwort:		
0		Lesen erfolgreich	–	C2
		Schreiben erfolgreich	–	C1
		Befehl zurückgewiesen	–	C0
0	1	Type	28	28
1	2	Instanz	01 – 20	01 – 20
2	3	Index	→ Tabelle 121	→ Tabelle 121
3 – 6	4 – 7	Data 1 – 4		
		beim Lesen	00	abhängig vom Index, → Tabelle 122
		beim Schreiben	abhängig vom Index, → Tabelle 122	00

Tabelle 121: Operandenübersicht

Index (hex)	Operand		lesen	schreiben
00	Bit-IO, → Tabelle 122		×	
01	Mode/Parameter		–	–
02	Eingangswert, Wertebereich: –32 768 bis +32 767	I1	×	c ¹⁾
03	Ausgleichszeit [0,1 s] Wertebereich: 0 bis 65 535	TG	×	c ¹⁾
04	Proportionalverstärkung [%], Wertebereich: 0 bis 65 535	KP	×	c ¹⁾
05	Verzögerter Ausgangswert, Wertebereich: –32 768 bis +32 767	QV	×	

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

Tabelle 122: 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		Master	Slave
		Attribute ID: Lesen	92	–
	0	Antwort:		
		Lesen erfolgreich	–	C2
		Befehl zurückgewiesen	–	C0
0	1	Type	1B	1B
1	2	Instanz	01 – 20	01 – 20
2	3	Index	→ Tabelle 123	→ Tabelle 123
3 – 6	4 – 7	Data 1 – 4	00	abhängig vom Index, → Tabelle 124, 125

Tabelle 123: Operandenübersicht

Index (hex)	Operand	lesen	schreiben
00	Bit-IO, → Tabelle 124	×	
01	Mode/Parameter, → Tabelle 125	×	–
02	Ausgangswert: Istwert QV aus dem Netzwerk	×	



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

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 ein neuer Wert anliegt, der vom Netzwerk NET übertragen wird.

Tabelle 125: 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		Master	Slave
		Attribute ID: Lesen	92	–
	0	Antwort: Lesen erfolgreich	–	C2
		Befehl zurückgewiesen	–	C0
0	1	Type	1C	1C
1	2	Instanz	01 – 20	01 – 20
2	3	Index	→ Tabelle 126	→ Tabelle 126
3 – 6	4 – 7	Data 1 – 4	00	abhängig vom Index, → Tabelle 127

Tabelle 126: Operandenübersicht

Index (hex)	Operand	lesen	schreiben
00	Bit-IO → Tabelle 127	×	
01	Mode/Parameter	–	–
02	Parameter → Tabelle 128	×	
	Kanal A		
03	Kanal B		
04	Kanal C		
05	Kanal D		

Tabelle 127: 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 128: 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	Attribute ID: 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		Master	Slave
		Attribute ID: Lesen	92	–
	0	Antwort: Lesen erfolgreich	–	C2
		Befehl zurückgewiesen	–	C0
0	1	Type	1D	1D
1	2	Instanz	01 – 20	01 – 20
2	3	Index	→ Tabelle 129	→ Tabelle 129
3 – 6	4 – 7	Data 1 – 4	00	abhängig vom Index, → Tabelle 130

Tabelle 129: Operandenübersicht

Index (hex)	Operand	lesen	schreiben
00	Bit-IO → Tabelle 130	×	
01	Mode/Parameter	–	–
02	Parameter → Tabelle 131	×	
	Kanal A		
03	Kanal B		
04	Kanal C		
05	Kanal D		

Tabelle 130: 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 131: 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 1110_{bin}

Monat = 6 (Juni) = 06_{hex} = 0000 0110_{bin}

Jahr = 2003 = 03_{hex} = 0000 0011_{bin}

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 0011_{bin}

Monat = 10 (Oktober) = 0A_{hex} = 0000 1010_{bin}

Jahr = 2012 = 0C_{hex} = 0000 1100_{bin}

Resultierendes Telegramm:

Byte	Bedeutung	Wert (hex), sendet	
		Master	Slave
0	Attribute ID: 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		Master	Slave
		Attribute ID Befehl: Lesen	92	–
		Attribute ID Befehl: Schreiben	B2	–
	0	Antwort:		
		Lesen erfolgreich	–	C2
		Schreiben erfolgreich	–	C1
		Befehl zurückgewiesen	–	C0
0	1	Type	2F	2F
1	2	Instanz	01 - 20	01 - 20
2	3	Index	00	00
3 - 6	4 - 7	Data 1 - 4 beim Lesen	00	abhängig vom Index, → Tabelle 132, 133
		Date 1 - 4 beim Schreiben	abhängig vom Index, → Tabelle 132, 133	00

Tabelle 132: Operandenübersicht

Index (hex)	Daten	Data 1	Data 2	Data 3	Data 4	Read/Write
0	Bit-IO	→ Tabelle 133	–	→ Tabelle 133	–	R

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 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		Master	Slave
Attribute ID				
		Lesen	92	–
		Schreiben	B2	–
		Antwort:		
		Lesen erfolgreich	–	C2
		Schreiben erfolgreich	–	C1
		Befehl zurückgewiesen	–	C0
0	1	Type	29	29
1	2	Instanz	01 – 20	01 – 20
2	3	Index	→ Tabelle 134	→ Tabelle 134
3 – 6	4 – 7	Data 1 – 4		
		beim Lesen	00	abhängig vom Index, → Tabelle 135
		beim Schreiben	abhängig vom Index, → Tabelle 135	

Tabelle 134: Operandenübersicht

Index (hex)	Operand	lesen	schreiben
00	Bit-IO, → Tabelle 135	×	
01	Mode/Parameter	–	–
02	Eingangswert, Wertebereich: 32 Bit	I1	×
03	Stützpunkt 1, X-Koordinate, Wertebereich: 32 Bit	X1	×
04	Stützpunkt 1, Y-Koordinate, Wertebereich: 32 Bit	Y1	×
05	Stützpunkt 2, X-Koordinate, Wertebereich: 32 Bit	X2	×
06	Stützpunkt 2, Y-Koordinate, Wertebereich: 32 Bit	Y2	×
07	Ausgangswert: beinhaltet den skalierten Eingangswert	QV	×

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

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“

Masterreset: MR01 – MR32**Telegrammaufbau**

Byte		Bedeutung	Wert (hex), sendet	
Master	Slave		Master	Slave
		Attribute ID: Lesen	92	–
	0	Antwort:		
		Lesen erfolgreich	–	C2
		Befehl zurückgewiesen	–	C0
0	1	Type	0F	0F
1	2	Instanz	01 – 20	01 – 20
2	3	Index		
		Bit-IO	00	00
		Mode	01	01
3 – 6	4 – 7	Data 1 – 4	00	abhängig vom Index, → Tabelle 136, 137

Tabelle 136: 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 137: 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		Master	Slave
		Attribute ID Befehl: Lesen	92	–
		Attribute ID Befehl: Schreiben	B2	–
	0	Antwort:		
		Lesen erfolgreich	–	C2
		Schreiben erfolgreich	–	C1
		Befehl zurückgewiesen	–	C0
0	1	Type	31	31
1	2	Instanz	01 - 20	01 - 20
2	3	Index	00 - 0B	00 - 0B
3 - 6	4 - 7	Data 1 - 4 beim Lesen	00	abhängig vom Index, → Tabelle 138, 139
		Date 1 - 4 beim Schreiben	abhängig vom Index, → Tabelle 138, 139	00

Tabelle 138: Operandenübersicht

Index (hex)	Daten	Data 1 Data 3	Data 2 Data 4	Read/Write
0	Bit-IO	→ Tabelle 139	–	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 139: 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.

Zahlenwandler: NC01 – NC32

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		Master	Slave
Attribute ID				
		Lesen	92	–
		Schreiben	B2	–
	0	Antwort:		
		Lesen erfolgreich	–	C2
		Schreiben erfolgreich	–	C1
		Befehl zurückgewiesen	–	C0
0	1	Type	2A	2A
1	2	Instanz	01 – 20	01 – 20
2	3	Index	→ Tabelle 140	→ Tabelle 140
3 – 6	4 – 7	Data 1 – 4		
		beim Lesen	00	abhängig vom Index, → Tabelle 141, 142
		beim Schreiben	abhängig vom Index, → Tabelle 141, 142	00

Tabelle 140: Operandenübersicht

Index (hex)	Operand	lesen	schreiben
00	Bit-IO, → Tabelle 141	×	
01	Mode, → Tabelle 142	×	
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 141: 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 142: 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		Master	Slave
Attribute ID				
		Lesen	92	–
		Schreiben	B2	–
	0	Antwort:		
		Lesen erfolgreich	–	C2
		Schreiben erfolgreich	–	C1
		Befehl zurückgewiesen	–	C0
0	1	Type	1E	1E
1	2	Instanz	01 – 04	01 – 04
2	3	Index	→ Tabelle 143	→ Tabelle 143
3 – 6	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	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 144: 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 P001...P002

Telegrammaufbau

Byte		Bedeutung	Wert (hex), sendet	
Master	Slave		Master	Slave
		Attribute ID Befehl: Lesen	92	–
		Attribute ID Befehl: Schreiben	B2	–
	0	Antwort:		
		Lesen erfolgreich	–	C2
		Schreiben erfolgreich	–	C1
		Befehl zurückgewiesen	–	C0
0	1	Type	32	32
1	2	Instanz	01 - 02	01 - 02
2	3	Index	00 - 0A	00 - 0A
3 - 6	4 - 7	Data 1 - 4 beim Lesen	00	abhängig vom Index, → Tabelle 145, 146
		Date 1 - 4 beim Schreiben	abhängig vom Index, → Tabelle 145, 146	00

Tabelle 145: Operandenübersicht

Index (hex)	Daten	Data 1 Data 3	Data 2 Data 4	Read/ Write
0	Bit-IO	→ Tabelle 146	–	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	Schrittzahl 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 146: 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		Master	Slave
		Attribute ID Befehl: Lesen	92	–
		Attribute ID Befehl: Schreiben	B2	–
	0	Antwort:		
		Lesen erfolgreich	–	C2
		Schreiben erfolgreich	–	C1
		Befehl zurückgewiesen	–	C0
0	1	Type	1F	1F
1	2	Instanz	01 - 20	01 - 20
2	3	Index	00 - 02	00 - 02
3 - 6	4 - 7	Data 1 - 4 beim Lesen	00	abhängig vom Index, → Tabelle 147, 148
		Date 1 - 4 beim Schreiben	abhängig vom Index, → Tabelle 147, 148	00

Tabelle 147: Operandenübersicht

Index (hex)	Daten	Data 1	Data 2	Data 3	Data 4	Read/Write
0	Bit-IO	→ Tabelle 148	–	→ Tabelle 148	–	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 148: 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		Master	Slave
Attribute ID				
		Lesen	92	–
		Schreiben	B2	–
		Antwort:		
0		Lesen erfolgreich	–	C2
		Schreiben erfolgreich	–	C1
		Befehl zurückgewiesen	–	C0
0	1	Type	2B	2B
1	2	Instanz	01 – 02	01 – 02
2	3	Index	→ Tabelle 149	→ Tabelle 149
3 – 6	4 – 7	Data 1 – 4		
		beim Lesen	00	abhängig vom Index, → Tabelle 150
		beim Schreiben	abhängig vom Index, → Tabelle 150	00

Tabelle 149: Operandenübersicht

Index (hex)	Operand		lesen	schreiben
00	Bit-IO, → Tabelle 150		×	
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 beschriftet ist.

Tabelle 150: 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		Master	Slave
		Attribute ID: Lesen	92	–
		Antwort:		
0		Lesen erfolgreich	–	C2
		Befehl zurückgewiesen	–	C0
0	1	Type	20	20
1	2	Instanz	01	01
2	3	Index	→ Tabelle 151	→ Tabelle 151
3 – 6	4 – 7	Data 1 – 4	00	abhängig vom Index, → Tabelle 152

Tabelle 151: Operandenübersicht

Index (hex)	Operand	lesen	schreiben
00	Bit-IO, → Tabelle 152	×	
01	Mode/Parameter	–	–

Tabelle 152: Index 0 – Bit-IO

	Bit	7	6	5	4	3	2	1	0
FB-Eingang Data 1		–	–	–	–	–	–	–	$\bar{T}^{1)}$
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	Master	Slave
		Attribute ID Befehl: Lesen	92	–
		Attribute ID Befehl: Schreiben	B2	–
	0	Antwort:		
		Lesen erfolgreich	–	C2
		Schreiben erfolgreich	–	C1
		Befehl zurückgewiesen	–	C0
0	1	Type	35	35
1	2	Instanz	01 - 20	01 - 20
2	3	Index	00	00
3 - 6	4 - 7	Data 1 - 4 beim Lesen	00	abhängig vom Index, → Tabelle 153, 154

Tabelle 153: Operandenübersicht

Index (hex)	Daten	Data 1	Data 2	Data 3	Data 4	Read/Write
0	Bit-IO	→ Tabelle 154	–	→ Tabelle 154	–	R

Tabelle 154: 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**Telegrammaufbau**

Byte		Bedeutung	Wert (hex), sendet	
Master	Slave		Master	Slave
		Attribute ID Befehl: Lesen	92	–
		Attribute ID Befehl: Schreiben	B2	–
	0	Antwort:		
		Lesen erfolgreich	–	C2
		Schreiben erfolgreich	–	C1
		Befehl zurückgewiesen	–	C0
0	1	Type	33	33
1	2	Instanz	01 - 20	01 - 20
2	3	Index	00 - 0B	00 - 0B
3 - 6	4 - 7	Data 1 - 4 beim Lesen	00	abhängig vom Index, → Tabelle 155, 156
		Date 1 - 4 beim Schreiben	abhängig vom Index, → Tabelle 155, 156	00

Tabelle 155: Operandenübersicht

Index (hex)	Daten	Data 1	Data 2 Data 4	Data 3	Read/ Write
0	Bit-IO	→ Tabelle 156	–	→ Tabelle 156	R
1	Mode	→ Tabelle 157	–	–	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 156: 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 157: : 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



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		Master	Slave
Attribute ID:				
		Lesen	92	–
		Schreiben	B2	–
0		Antwort:		
		Lesen erfolgreich	–	C2
		Schreiben erfolgreich	–	C1
		Befehl zurückgewiesen	–	C0
0	1	Type	2C	2C
1	2	Instanz	01	01
2	3	Index	→ Tabelle 158	→ Tabelle 158
3 – 6	4 – 7	Data 1 – 4		
		beim Lesen	00	abhängig vom Index, → Tabelle 159
		beim Schreiben	abhängig vom Index, → Tabelle 159	00

Tabelle 158: Operandenübersicht

Index (hex)	Operand	lesen	schreiben
00	Bit-IO, → Tabelle 159	×	
01	Mode/Parameter	–	–
02	Zykluszeit in ms, 11 Wertebereich: 0 – 1000	×	c ¹⁾

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

Tabelle 159: 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		Master	Slave
Attribute ID				
		Lesen	92	–
		Schreiben	B2	–
	0	Antwort:		
		Lesen erfolgreich	–	C2
		Schreiben erfolgreich	–	C1
		Befehl zurückgewiesen	–	C0
0		1	Type	21
1	2	Instanz	01 – 20	01 – 20
2	3	Index	→ Tabelle 160	→ Tabelle 160
3 – 6	4 – 7	Data 1 – 4		
		beim Lesen	00	abhängig vom Index, → Tabelle 161, 162
		beim Schreiben	abhängig vom Index, → Tabelle 161, 162	

Tabelle 160: Operandenübersicht

Index (hex)	Operand	lesen	schreiben
00	Bit-IO, → Tabelle 161	×	
01	Mode/Parameter, → Tabelle 162	×	
02	Sollwert 1: I1 Zeitsollwert 1	×	c ¹⁾
03	Sollwert 2: I2 Zeitsollwert 2 (bei Zeitrelais mit 2 Sollwerten)	×	c ¹⁾
04	Istwert: QV Abgelaufene Istzeit im RUN-Betrieb	×	

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 161: 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 162: 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		Master	Slave
		Attribute ID Befehl: Lesen	92	–
		Attribute ID Befehl: Schreiben	B2	–
	0	Antwort:		
		Lesen erfolgreich	–	C2
		Schreiben erfolgreich	–	C1
		Befehl zurückgewiesen	–	C0
0	1	Type	34	34
1	2	Instanz	01 - 20	01 - 20
2	3	Index	00 - 04	00 - 04
3 - 6	4 - 7	Data 1 - 4 beim Lesen	00	abhängig vom Index, → Tabelle 163, 164
		Date 1 - 4 beim Schreiben	abhängig vom Index, → Tabelle 163, 164	00

Tabelle 163: Operandenübersicht

Index (hex)	Daten	Data 1 Data 3	Data 2 Data 4	Read/ Write
0	Bit-IO	→ Tabelle 164	–	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 beschriftet ist.

Tabelle 164: 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		Master	Slave
Attribute ID				
		Lesen	92	–
		Schreiben	B2	–
		Antwort:		
		Lesen erfolgreich	–	C2
		Schreiben erfolgreich	–	C1
		Befehl zurückgewiesen	–	C0
0	1	Type	2D	2D
1	2	Instanz	01 – 20	01 – 20
2	3	Index	→ Tabelle 165	→ Tabelle 165
3 – 6	4 – 7	Data 1 – 4		
		beim Lesen	00	abhängig vom Index, → Tabelle 166
		beim Schreiben	abhängig vom Index, → Tabelle 166	00

Tabelle 165: Operandenübersicht

Index (hex)	Operand	lesen	schreiben
00	Bit-IO, → Tabelle 166	×	
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 166: 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 167
5 – 7	Data 2 – 4	

Tabelle 167: Fehlercodes

Fehlercode	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.

9 Was ist, wenn ...?

Modul Status LED MS	Mögliche Ursache	Abhilfemaßnahme
Aus	EASY222-DN ist nicht mit Spannung versorgt.	Spannungsversorgung einschalten.
Grün	EASY222-DN befindet sich im Normalbetrieb.	Keine
Blinkt Grün	EASY222-DN ist nicht konfiguriert.	Überprüfen, ob die richtige MAC ID eingestellt wurde.
Blinkt Rot	Ungültige Konfiguration	Konfigurationseinrichtung überprüfen.
Rot	Modul weist eine nicht behebbare Störung auf.	EASY222-DN ersetzen.

Network Status LED NS	Mögliche Ursache	Abhilfemaßnahme
Aus	<ul style="list-style-type: none"> • EASY222-DN ist nicht mit Spannung versorgt oder • der Kanal ist für die Kommunikation gesperrt, <ul style="list-style-type: none"> – da der Bus ausgeschaltet wurde, – die Netzspannung verloren ging oder – der Kanal mit Absicht gesperrt wurde. 	<ul style="list-style-type: none"> • EASY222-DN einschalten, • Kanal mit Netzspannung versorgen und • sicherstellen, dass der Kanal aktiviert ist.
Grün	Der Kanal ist aktiviert, jedoch erfolgt keine Kommunikation.	Master-Steuerung auf Kommunikation hin überprüfen.
Blinkt Grün	Normalbetrieb	Keine
Blinkt Rot	Die Kommunikation ist gestört oder EASY222-DN ist unter Umständen defekt.	Modul zurücksetzen. Kommt es weiterhin zu Störungen, EASY222-DN austauschen.
Rot	Kommunikation ist gestört.	Überprüfen der Master-Steuerung.

Anhang

Technische Daten		
Allgemeines		
Normen und Bestimmungen		EN 61000-6-1; EN 61000-6-2; EN 61000-6-3; EN 61000-6-4, 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 ₂ S 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		
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		
eindrätig minimal bis maximal	mm ²	0,2 bis 4
	AWG	22 bis 12
feindrätig 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
DeviceNet		
Anschluss Gerät		5-polig, Buchse
Potentialtrennung		Bus zu Spannungsversorgung (einfach) Bus und Stromversorgung zu easy-Basisgerät (sichere Trennung)
Funktion		DeviceNet-Slave
Schnittstelle		DeviceNet (CAN)
Busprotokoll		DeviceNet
Baudraten, automatische Suche bis	kBd	500
Busabschlusswiderstände		Separat am Bus anbringbar
Busadressen, über easy-Basisgerät mit Display oder easySoft adressierbar		0 bis 63
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

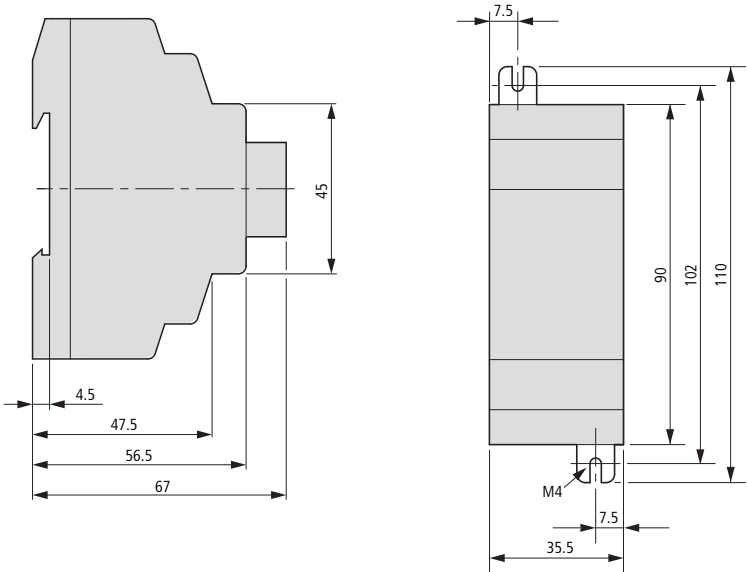


Abbildung 13: Abmessung EASY222-DN, Maße in mm

EDS-Datei

```

$*****
$ Moeller GmbH
$ Geraet: EASY222-DN
$ Version: V1.0
$ Datum: 27.05.02
$ Autor: Ronny Happ
$ Beschreibung: EDS-Datei fuer easy-DeveceNet-Slave-Modul
$ Aenderungen:
$
$ Copyright (c) 2002 by Moeller GmbH
$*****

```

[File]

```

$ File Description Section:
$ For more information about the meaning of each entry, please check
$ DeviceNet Specification Volume II Chapter 4-3.5.1

```

```

DescText      = "Moeller DeviceNet Coupler easy 222-DN";
CreateDate    = 27-05-2002;
CreateTime    = 17:00:00;
ModDate       = 25-06-2002;
ModTime       = 11:00:00;
Revision      = 1.0;

```

[Device]

```

$ Device Description Section:
$ For more information about the meaning of each entry, please check
$ DeviceNet Specification Volume II Chapter 4-3.5.2

```

```

VendCode      = 248;           $ Identity Object - Vendor ID
ProdType      = 12;           $ Identity Object - Device Type
ProdCode      = 650;         $ Identity Object - Product Code
MajRev        = 1;           $ Identity Object - Major Revision
MinRev        = 1;           $ Identity Object - Minor Revision

```

```

                                $ Identity Object - Product Name
ProdName      = "EASY 222-DN";
VendName      = "Moeller ElectroniX";
ProdTypeStr   = "Generic";
Catalog       = "Moeller HPL Bestell-Nr. 233540";

[IO_Info]
$ I/O Characteristics Section:
$ For more information about the meaning of each entry, please check
$ DeviceNet Specification Volume II Chapter 4-3.5.3

Default       = 0x000D;                $ Cyclic, Change of State and Poll

PollInfo      =
0x000D,      $ Poll (ok to combine with Cyclic or COS)
2,           $ Default Input = Input2
1           $ Default Output = Output1

COSInfo       =
0x000D,      $ COS (ok to combine with Poll)
1           $ Default Input = Input1
2;          $ Default Output = Output2

CyclicInfo    =
0x000D,      $ Cyclic (ok to combine with Poll)
1,           $ Default Input = Input1
2;          $ Default Output = Output2

$ Input Connections
Input1        =
2,           $ 2 bytes are transfered
16,          $ all bits are significant
0x0004,      $ COS only
"Diagnostic Data from easy", $ Name
6, "20 04 24 64 30 03",    $ Assembly Object Instance 100,
                                $ Attribute 3
";          $ Help

```



```

Input2      =
            3,                $ 3 bytes are transferred
            24,               $ all bits are significant
            0x0001,          $ Poll only
            "Input Data from easy", $ Name
            6, "20 04 24 65 30 03", $ Assembly Object Instance 101,
                                   $ Attribute 3
            "";               $ Help
$ Output Connections
Output1     =
            3,                $ 3 byte is transferred
            24,               $ all bits are significant
            0x0001,          $ Poll and COS
            "Output Data to easy", $ Name
            6, "20 04 24 66 30 03", $ Assembly Object Instance 102,
                                   $ Attribute 3
            "";               $ Help
Output2     =
            0,                $ 0 byte is transferred
            0,                $ all bits are significant
            0x0004,          $ Poll and COS
            "Acknowledge Handler", $ Name
            6, "20 2B 24 01 30 00", $ Acknowledge Handler
            "Acknowledge Handler"; $ Help

[ParamClass]
$ Parameter Class Section:
$ For more information about the meaning of each entry, please check
$ DeviceNet Specification Volume II Chapter 4-3.5.4 and Chapter 6-14.1

MaxInst     = 0;                $ no parameters are supported
Descriptor   = 0;                $
CfgAssembly = 0;                $ not used here

```

[Params]

\$ Parameter Section:

\$ For more information about the meaning of each entry, please check
\$ DeviceNet Specification Volume II Chapter 4-3.5.5 and Chapter 6-14.2

[EnumPar]

\$ Parameter Enumerated String Section:

\$ For more information about the meaning of each entry, please check
\$ DeviceNet Specification Volume II Chapter 4-3.5.6

[Groups]

\$ Parameter Groups Section:

\$ Not used here
\$ For more information about the meaning of each entry, please check
\$ DeviceNet Specification Volume II Chapter 4-3.5.7

\$ End of File

**Hinweis zur EDS-Datei:**

Der Identity Object Eintrag - Major Revision gibt den aktuellen Betriebssystemstand des Kommunikationsmoduls EASY222-DN an. Da das Gerät mit einer neueren Betriebssystemversion von der EDS-Beschreibung in diesem Punkt abweichen kann, muss dieser Eintrag entsprechend abgeändert werden, → Abschnitt „Identity Object“ auf Seite 35.

Glossar

Das Glossar bezieht sich auf Themen rund um DeviceNet.

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.
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.
Automatisierungsgerät	Gerät zur Steuerung mit Ein- und Ausgängen, das an einen technischen Prozess angeschlossen wird. Speicherprogrammierbare Steuerungen (SPS) sind eine spezielle Gruppe von Automatisierungsgeräten.
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).
Blitzschutz	Alle Maßnahmen, die dazu dienen, ein System vor Schäden durch Überspannungen zu schützen, die von Blitzen hervorgerufen werden können.
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
Code	Format der Datenübertragung
COS I/O Connection	Bei COS (Change Of State) I/O Connections werden ereignisgesteuerte Verbindungen aufgebaut. Das heißt, dass die DeviceNet-Geräte von sich aus Nachrichten generieren, sobald eine Zustandsänderung vorliegt. 2 Byte Diagnosedaten des easy-Steuerrelais Koppelmodul Status

CPU	Abkürzung für englisch "Central Processing Unit". Zentrale Einheit zur Datenverarbeitung, das Kernstück eines Rechners.
Cyclic I/O Connection	Bei einer Cyclic I/O Connection erfolgt die Auslösung von Nachrichten zeitgesteuert mittels Zeitgeber.
Device Heartbeat Message	Mit Device Heartbeat Message kann ein DeviceNet-Gerät in einem konfigurierten Zeitintervall seinen eigenen Status bekannt geben. Konfiguriert werden diese Nachrichten im Identity Object.
Device Shut Down Message	Muss sich ein Gerät aufgrund interner Fehler oder Zustände abschalten, kann es sich mit der Device Shut Down Message definiert bei der Steuerung abmelden.
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.“.
Dual Code	Natürlicher Binärcode, häufig verwendeter Code bei absoluten Messsystemen.
EDS	Electronical Data Sheet: In dieser EDS-Datei sind in erster Linie die Polled I/O Connection, die COS I/O Connection als auch die Cyclic I/O Connection des Gateway definiert. Die Daten bzw. Parameter (easy Object) für die im easy-Grundgerät vorhandenen Funktionen werden in dieser Datei nicht angeführt. Der Zugriff auf die entsprechenden Funktionen wird über explizite Nachrichten realisiert ist.
EEPROM	Abkürzung für englisch "Electrically Erasable Programmable Read-Only Memory": Elektrisch löschbarer, programmierbarer und lesbarer Speicher.
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“, elektro-statische Entladung.
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).
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.
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.

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.
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.
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.
LSB	Abkürzung für englisch "Least Significant Bit". Bit mit dem niedrigsten Stellenwert.
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.
Offline Connection Set	Das Offline Connection Set ermöglicht die Kommunikation mit einem Gerät, das sich aufgrund einer doppelten Adresse im Communication-Fault-Zustand befindet, aber nicht im Bus-Off ist. Dieses Gerät ist normalerweise nicht mehr über das Netzwerk ansprechbar, sondern muss von Hand ausgeschaltet oder durch Ein- und Ausschalten wieder initialisiert werden. Mit Hilfe des Offline Connection Sets kann ein solches Gerät über das Netzwerk angesprochen werden.
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 DeviceNet-Masters.

Polled I/O Connection	<p>Mit einer Polled I/O Connection wird eine klassische Master/ Slave-Beziehung zwischen einer Steuerung und einem DeviceNet-Gerät aufgebaut. Bei einer Polled I/O Connection handelt es sich um eine Punkt-zu-Punkt-Verbindung zwischen zwei Teilnehmern am Feldbus. Der Master (Client) sendet mit einem Poll-Request eine Anfrage an den Slave (Server) und dieser antwortet mit einer Poll-Response.</p> <ul style="list-style-type: none"> • 3 Byte Ausgangsdaten S1 bis S8 easy/MFD-Ausgangsbereich, RUN/STOP (Eingänge am DeviceNet-Master) • 3 Byte Eingangsdaten R1 bis R16 easy/MFD-Eingangsbereich, RUN/STOP (Ausgänge vom DeviceNet-Master)
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.
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.
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").

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.
Topologie	Geometrischer Aufbau eines Netzes bzw. Anordnung der Schaltungen.
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.
UCMM	Das DeviceNet-Gateway bietet die Möglichkeit, über den UCMM-Port (Unconnected Message Manager Port) dynamische Verbindungsobjekte einzurichten.
unidirektional	In einer Richtung arbeitend.

Stichwortverzeichnis

A	Abbilddaten	
	Allgemeine Informationen	61
	Übersicht easy600	61
	Übersicht easy700	105
	Übersicht easy800/MFD	154
	Abmessungen	258
	Abschlusswiderstände	20
	Adressbereich	24
	Adresse einstellen mit easySoft	26
	ALT-Taste, Zustand lesen	64
	Analog-Ausgang	
	easy800/MFD (Zustand lesen)	167
	Analog-Eingänge	
	easy600 (Zustand lesen)	65
	easy700 (Zustand lesen)	110
	easy800/MFD (Zustand lesen)	155
	Analogwertvergleicher	
	easy600 (Istwerte schreiben)	73
	easy700	127
	easy700 (Zustand lesen)	106
	easy800/MFD	177
	Anschlussbelegung DeviceNet	19
	Application Objects	34
	Applikationsspezifische Objekte	34
	Arithmetikbaustein	
	easy800/MFD	179
	Assembly Objects	34
	Aufbau des Gerätes	13
	Ausgänge S1 bis S8 lesen	50
	Ausgänge von easyLink	
	easy700 (Zustand lesen)	120
	easy800/MFD (Zustand lesen)	170
	Ausgänge, lokale und Netzwerkteilnehmer	
	easy800/MFD (Zustand lesen)	168
	Ausgangsdaten, Definition	48

	Ausschaltzeitpunkt	
	lesen	93
	schreiben	97
<hr/>		
B	Baudratenerkennung, automatisch	22
	Bedientasten	
	easy600 (Zustand lesen)	62
	Bedingter Sprung	
	easy800/MFD	217
	Betriebsart schreiben	51
	Betriebsmodus, ungültig	144, 251
	Betriebsstundenzähler	
	easy700	133
	easy800/MFD	227
	Betriebssystemvoraussetzungen	15
	Bitfeld	103
	Block Compare	
	easy800/MFD	181
	Block Transfer	
	easy800/MFD	183
	Boolsche Verknüpfung	
	easy800/MFD	185
	Busleitungslängen	22
<hr/>		
C	Connection ID	44
	COS I/O Connection	264
	Cyclic I/O Connection	265
<hr/>		
D	Datenaustausch, PDO	47
	Datenbaustein	
	easy800/MFD	200
	Datenmultiplexer	
	easy800/MFD	223
	Datum lesen/schreiben	
	easy600	57
	easy700	101, 148
	DEL-Taste, Zustand lesen	64

DeviceNet	
anschließen	19
Anschlussbelegung	19
Object	33
Diagnose	
easy800/MFD	205
Diagnose, lokale	
easy800/MFD (Abbilddaten)	157
Diagnose, Remote-Teilnehmer	
easyMFD (Abbilddaten)	157
Digital-Ausgänge	
easy600 (Zustand lesen)	69
easy700 (Zustand lesen)	119
Digital-Ausgänge, lokale und Netzwerkteilnehmer	
easy800/MFD (Zustand lesen)	168
Digital-Eingänge	
easy600 (Zustand lesen)	62
easy700 (Zustand lesen)	109
easy800/MFD (Zustand lesen)	159
Direkter Datenaustausch	47

E	easy Object	34, 38
	EDS-Datei	29
	Eingänge R1 bis R8 setzen/rücksetzen	54
	Eingänge von easyLink	
	easy700 (Zustand lesen)	120
	easy800/MFD (Zustand lesen)	170
	Eingänge, Netzwerkteilnehmer	
	easy800/MFD (Zustand lesen)	161
	Eingangsdaten, Definition	48
	Einschalten, erstes	23
	Einschaltzeitpunkt	
	lesen	93
	schreiben	97
	Empfangsdaten, Netzwerkteilnehmer	
	easy800/MFD	209
	easy800/MFD (Zustand lesen)	172
	ESC-Taste, Zustand lesen	64
	Explicit Messages	42

<hr/>	
F	Fehlercodes, über easyLink
	easy700 144
	easy800/MFD 251
	Frequenzzähler
	easy800/MFD 189
	Funktionsbausteine, Übersicht
	easy600 72
	easy700 126
	easy800/MFD 175
<hr/>	
G	Geräteadresse 43
<hr/>	
H	Hardware-Voraussetzungen 15
	Heartbeat Message 265
	Hilfsrelais
	easy600 (Zustand lesen) 69
<hr/>	
I	Identity Object 33
	Impulsausgabe
	easy800/MFD 229
	Inkrementalzähler
	easy800/MFD 193
<hr/>	
J	Jahresschaltuhr
	easy700 138
	easy700 (Zustand lesen) 123
	Jahreszeitschaltuhr
	easy800/MFD 214
<hr/>	
K	Knotenadresse 43
	Kommunikationsprofil 14

L	LED-Statusanzeigen27, 253	
	Lokale Ausgänge	
	easy700 (Zustand lesen) 119	
	Lokale Eingänge	
	easy700 (Zustand lesen) 109	
	easy800/MFD (Zustand lesen) 159	
	Lokaler Analog-Ausgang	
	easy800/MFD (Zustand lesen) 167	
<hr/>		
M	MAC-ID43, 44	
	Masterreset	
	easy700 124	
	easy800/MFD221	
	Merker	
	easy600 (Zustand lesen)69	
	easy700, lesen 114	
	easy700, schreiben 112	
	easy800/MFD(Zustand lesen) 162	
	Message ID44	
	Message Router Object33	
	Message-Group44	
	Minute	
	easy60059	
	lesen93	
	schreiben97	
	Modul Status LED27, 253	
	MS-LED27, 253	
<hr/>		
N	Network Status LED28, 253	
	Netzdaten empfangen	
	easy800/MFD209	
	Netzdaten senden	
	easy800/MFD232	
	Netzwerkteilnehmer, Zustand der Eingänge lesen	
	161	
	NS-LED28, 253	

O	Offline Connection Set	268
	OK-Taste, Zustand lesen	64
<hr/>		
P	PDO	47
	PID-Regler	
	easy800/MFD	202
	Polled I/O Connection	269
	Potentialtrennungen	21
	P-Tasten	
	easy600(Zustand lesen)	62
	easy700 (Zustand lesen)	117
	easy800 (Zustand lesen)	165
	Pulsweitenmodulation	
	easy800/MFD	234
<hr/>		
R	Reaktionszeit Basisgerät	29
	Receive-Data, Netzwerkteilnehmer	
	easy800/MFD (Zustand lesen)	172
	Rücksetzen, easy/MFD-Ein-/Ausgänge	51
<hr/>		
S	Schaltregel	103
	Schaltzeitpunkte	
	lesen	93
	schreiben	97
	Schieberegister	
	easy800/MFD	239
	Schneller Zähler	
	easy800/MFD	191
	Schwellertschalter	
	easy700	127
	Schwellwertvergleich	
	easy700 (Zustand lesen)	106
	SDO	
	Steuerbefehle für easy600	55
	Steuerbefehle für easy700	99
	Steuerbefehle für easy800	145
	Send-Data, Netzwerkteilnehmer	
	easy800/MFD (Zustand lesen)	172

Sendedaten, Netzwerkteilnehmer	
easy800/MFD	232
easy800/MFD (Zustand lesen)	172
Serielle Ausgabe	
easy800/MFD	237
Setzen, easy/MFD-Ein-/Ausgänge	51
Shut Down Message	265
Signalglättungsfilter	
easy800/MFD	207
Sollzykluszeit	
easy800/MFD	242
Sommerzeit	
easy600	57
easy700	102
easy800/MFD	149
Steuerbefehle	
easy600	55
easy700	99
easy800	145
Stunde	
easy600	58
lesen	93
schreiben	97
Systemübersicht	12

T Tabellenfunktion	
easy800/MFD	247
Tastereingänge P1 bis P4	
easy600	62
Teilnehmeradresse einstellen	24
Telegramm, ungültig	144, 251
Textanzeige	
easy600 (Zustand lesen)	69
Textausgabe-Baustein	
easy800/MFD	197
Textbausteine	
easy700 (Zustand lesen)	108
Textmerker	71

U	Übertragungsraten	22
	UCMM	270
	Uhr synchronisieren	
	easy800/MFD	236
	Uhrzeit lesen/schreiben	
	easy600	57
	easy700	101, 148
	ungültiger Betriebsmodus	144, 251
	ungültiges Telegramm	144, 251
<hr/>		
V	Verbindungsobjekte	33
	Vergleicher	
	easy800/MFD	195
	Vergleichswert schreiben (Analogwert-Vergleicher) .	74
	Versionsgeschichte, easy800	148
	Versorgungsspannung	18
	Verzögerungszeit, zufälliger Wert	84
<hr/>		
W	Wertbegrenzung	
	easy800/MFD	249
	Wertskalierung	
	easy800/MFD	219
	Winterzeit	
	easy600	57
	easy700	103
	easy800/MFD	149
	Wochenschaltuhr	
	easy700	125, 141
	Wochentag	
	easy600	58
	Wochenzeitschaltuhr	
	easy800/MFD	211

Z	Zahlenwandler	
	easy800/MFD	225
	Zähler	
	easy700 (Zustand lesen)	107
	easy800/MFD	187
	Zählerrelais	
	easy600 (Istwert lesen)	76
	easy600 (Sollwert lesen)	80
	easy600 (Sollwert schreiben)	78
	easy700	130
	Zeitglieder	
	easy700 (Zustand lesen)	122
	Zeitrelais	
	easy600 (Istwert lesen)	82
	easy600 (Parameter schreiben)	86
	easy700	135
	easy800/MFD	244
	Zeitschaltuhr	
	easy600 (Istwerte lesen)	90
	easy600 (Sollwerte schreiben)	94
	Zeitumstellung (easy800/MFD)	149
	Zustand	
	Ausgänge S1 bis S8 lesen	50
	Betriebsart lesen	49
	Betriebsart schreiben	51
	Eingänge R1 bis R8 schreiben	54
	Zyklischer Datenaustausch	47
	Zykluszeit	29

