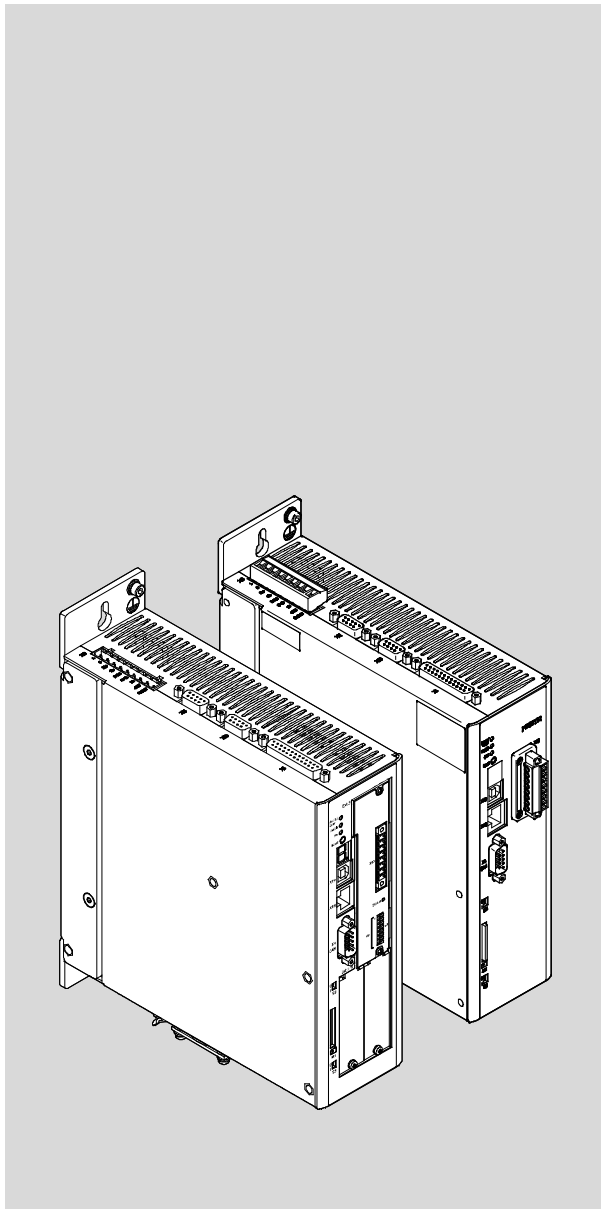


CiA 402 für Motorcontroller

CMMP-AS-...-M3/-M0



FESTO

Beschreibung

Geräteprofil CiA 402

für Motorcontroller
CMMP-AS-...-M3
über Feldbus:

- CANopen
- EtherCAT mit
Interface
CAMC-EC

für Motorcontroller
CMMP-AS-...-M0
über Feldbus:

- CANopen

8022082

1304a

Originalbetriebsanleitung

GDCP-CMMP-M3/-M0-C-CO-DE

CANopen®, CiA®, EthetCAT®, TwinCAT® sind eingetragene Marken der jeweiligen Markeninhaber in bestimmten Ländern.

Kennzeichnung von Gefahren und Hinweise zu deren Vermeidung:



Warnung

Gefahren, die zum Tod oder zu schweren Verletzungen führen können.



Vorsicht

Gefahren, die zu leichten Verletzungen oder zu schwerem Sachschaden führen können.

Weitere Symbole:



Hinweis

Sachschaden oder Funktionsverlust.



Empfehlung, Tipp, Verweis auf andere Dokumentationen.



Notwendiges oder sinnvolles Zubehör.



Information zum umweltschonenden Einsatz.

Textkennzeichnungen:

- Tätigkeiten, die in beliebiger Reihenfolge durchgeführt werden können.
- 1. Tätigkeiten, die in der angegebenen Reihenfolge durchgeführt werden sollen.
- Allgemeine Aufzählungen.

Inhaltsverzeichnis – CMMP-AS-...-M3/-M0

1	Feldbus-Schnittstellen	9
2	CANopen [X4]	10
2.1	Allgemeines zu CANopen	10
2.2	Verkabelung und Steckerbelegung	11
2.2.1	Anschlussbelegungen	11
2.2.2	Verkabelungs-Hinweise	11
2.3	Konfiguration CANopen-Teilnehmer beim CMMP-AS-...-M3	13
2.3.1	Einstellung der Knotennummer mit DIP-Schalter und FCT	14
2.3.2	Einstellung der Übertragungsrate mit DIP-Schalter	15
2.3.3	Aktivierung der CANopen-Kommunikation mit DIP-Schalter	15
2.3.4	Einstellung der physikalischen Einheiten (Faktoren-Gruppe)	15
2.4	Konfiguration CANopen-Teilnehmer beim CMMP-AS-...-M0	16
2.4.1	Einstellung der Knotennummer über DINs und FCT	17
2.4.2	Einstellung der Übertragungsrate über DINs oder FCT	17
2.4.3	Einstellung des Protokolls (Datenprofils) über DINs oder FCT	18
2.4.4	Aktivierung der CANopen-Kommunikation über DINs oder FCT	18
2.4.5	Einstellung der physikalischen Einheiten (Faktoren-Gruppe)	19
2.5	Konfiguration CANopen-Master	19
3	Zugriffsverfahren CANopen	20
3.1	Einleitung	20
3.2	SDO-Zugriff	21
3.2.1	SDO-Sequenzen zum Lesen und Schreiben	22
3.2.2	SDO-Fehlermeldungen	23
3.2.3	Simulation von SDO-Zugriffen	24
3.3	PDO-Message	25
3.3.1	Beschreibung der Objekte	26
3.3.2	Objekte zur PDO-Parametrierung	29
3.3.3	Aktivierung der PDOs	34
3.4	SYNC-Message	35
3.5	EMERGENCY-Message	35
3.5.1	Übersicht	35
3.5.2	Aufbau der EMERGENCY-Message	36
3.5.3	Beschreibung der Objekte	37
3.6	Netzwerkmanagement (NMT-Service)	38
3.7	Bootup	41
3.7.1	Übersicht	41
3.7.2	Aufbau der Bootup-Nachricht	41

3.8	Heartbeat (Error Control Protocol)	42
3.8.1	Übersicht	42
3.8.2	Aufbau der Heartbeat-Nachricht	42
3.8.3	Beschreibung der Objekte	42
3.9	Nodeguarding (Error Control Protocol)	43
3.9.1	Übersicht	43
3.9.2	Aufbau der Nodeguarding-Nachrichten	43
3.9.3	Beschreibung der Objekte	44
3.9.4	Objekt 100Dh: life_time_factor	45
3.9.5	Tabelle der Identifier	45
4	EtherCAT mit CoE	46
4.1	Überblick	46
4.2	EtherCat-Interface CAMC-EC	46
4.3	Einbau des EtherCAT-Interface in den Controller	48
4.4	Steckerbelegung und Kabelspezifikationen	48
4.5	CANopen-Kommunikationsschnittstelle	49
4.5.1	Konfiguration der Kommunikationsschnittstelle	50
4.5.2	Neue und geänderte Objekte unter CoE	52
4.5.3	Nicht unterstützte Objekte unter CoE	58
4.6	Kommunikations-Zustandsmaschine	60
4.6.1	Unterschiede zwischen den Zustandsmaschinen von CANopen und EtherCAT	62
4.7	SDO-Frame	63
4.8	PDO-Frame	64
4.9	Error Control	66
4.10	Emergency Frame	66
4.11	XML-Gerätebeschreibungsdtei	67
4.11.1	Grundsätzlicher Aufbau der Gerätebeschreibungsdtei	67
4.11.2	Receive-PDO-Konfiguration im Knoten RxPDO	69
4.11.3	Transmit-PDO-Konfiguration im Knoten TxPDO	71
4.11.4	Initialisierungskommandos über den Knoten "Mailbox"	71
4.12	Synchronisation (Distributed Clocks)	72
5	Parameter Einstellen	73
5.1	Parametersätze laden und speichern	73
5.2	Kompatibilitäts-Einstellungen	76
5.3	Umrechnungsfaktoren (Factor Group)	79
5.4	Endstufenparameter	89
5.5	Stromregler und Motoranpassung	96
5.6	Drehzahlregler	104
5.7	Lageregler (Position Control Function)	106
5.8	Sollwert-Begrenzung	118

5.9	Geberanpassungen	121
5.10	Inkrementalgeberemulation	125
5.11	Soll-/Istwertaufschaltung	127
5.12	Analoge Eingänge	130
5.13	Digitale Ein- und Ausgänge	132
5.14	Endschalter/Referenzschalter	138
5.15	Sampling von Positionen	141
5.16	Bremsen-Ansteuerung	144
5.17	Geräteinformationen	145
5.18	Fehlermanagement	152
6	Gerätesteuerung (Device Control)	155
6.1	Zustandsdiagramm (State Machine)	155
6.1.1	Übersicht	155
6.1.2	Das Zustandsdiagramm des Motorcontrollers (State Machine)	156
6.1.3	Steuerwort (Controlword)	161
6.1.4	Auslesen des Motorcontrollerzustands	164
6.1.5	Statusworte (Statuswords)	166
6.1.6	Beschreibung der weiteren Objekte	173
7	Betriebsarten	176
7.1	Einstellen der Betriebsart	176
7.1.1	Übersicht	176
7.1.2	Beschreibung der Objekte	176
7.2	Betriebsart Referenzfahrt (Homing Mode)	178
7.2.1	Übersicht	178
7.2.2	Beschreibung der Objekte	179
7.2.3	Referenzfahrt-Abläufe	183
7.2.4	Steuerung der Referenzfahrt	187
7.3	Betriebsart Positionieren (Profile Position Mode)	188
7.3.1	Übersicht	188
7.3.2	Beschreibung der Objekte	189
7.3.3	Funktionsbeschreibung	192
7.4	Synchrone Positionsvorgabe (Interpolated Position Mode)	195
7.4.1	Übersicht	195
7.4.2	Beschreibung der Objekte	195
7.4.3	Funktionsbeschreibung	201
7.5	Betriebsart Drehzahlregelung (Profile Velocity Mode)	203
7.5.1	Übersicht	203
7.5.2	Beschreibung der Objekte	205
7.6	Drehzahl-Rampen	211
7.7	Betriebsart Momentenregelung (Profile Torque Mode)	214

7.7.1	Übersicht	214
7.7.2	Beschreibung der Objekte	215
A	Technischer Anhang	220
A.1	Technische Daten Interface EtherCAT	220
A.1.1	Allgemein	220
A.1.2	Betriebs- und Umweltbedingungen	220
B	Diagnosemeldungen	221
B.1	Erläuterungen zu den Diagnosemeldungen	221
B.2	Errorcodes über CiA 301/402	222
B.2	Diagnosemeldungen mit Hinweisen zur Störungsbeseitigung	225

Hinweise zur vorliegenden Dokumentation

Diese Dokumentation beschreibt das Geräteprofil CiA 402 (DS 402) für die Motorcontroller CMMP-AS-...-M3/-M0 entsprechend Abschnitt "Informationen zur Version" über die Feldbus-Schnittstellen:

- CANopen – Schnittstelle [X4] im Motorcontroller integriert.
- EtherCAT – optionales Interface CAMC-EC im Steckplatz Ext2, nur für CMMP-AS-...-M3.

Damit erhalten Sie ergänzende Informationen zur Steuerung, Diagnose und Parametrierung der Motorcontroller über den Feldbus.

- Beachten Sie unbedingt die generellen Sicherheitsvorschriften zum CMMP-AS-...-M3/-M0.



Die generellen Sicherheitsvorschriften zum CMMP-AS-...-M3/-M0 finden Sie in der Beschreibung Hardware, GDCP-CMMP-AS-M3-HW-... bzw. GDCP-CMMP-AS-M0-HW-..., siehe Tab. 2.

Zielgruppe

Diese Beschreibung wendet sich ausschließlich an ausgebildete Fachleute der Steuerungs- und Automatisierungstechnik, die Erfahrungen mit der Installation, Inbetriebnahme, Programmierung und Diagnose von Positioniersystemen besitzen.

Service

Bitte wenden Sie sich bei technischen Fragen an Ihren regionalen Ansprechpartner von Festo.

Informationen zur Version

Die vorliegende Beschreibung bezieht sich auf folgende Versionen:

Motorcontroller	Version
CMMP-AS-...-M3	Motorcontroller CMMP-AS-...-M3 ab Rev 01
	FCT-PlugIn CMMP-AS ab Version 2.0.x.
CMMP-AS-...-M0	Motorcontroller CMMP-AS-...-M0 ab Rev 01
	FCT-PlugIn CMMP-AS ab Version 2.2.x.

Tab. 1 Versionen



Diese Beschreibung gilt nicht für die älteren Varianten CMMP-AS-.... Benutzen Sie für diese Varianten die zugeordnete CANopen-Beschreibung für die Motorcontroller CMMP-AS.



Hinweis

Prüfen Sie bei neueren Firmware-Ständen, ob hierfür eine neuere Version dieser Beschreibung vorliegt → www.festo.com

Dokumentationen

Weitere Informationen zum Motorcontroller finden Sie in den folgenden Dokumentationen:

Anwenderdokumentation zum Motorcontroller CMMP-AS-...-M3/-M0	
Name, Typ	Inhalt
Beschreibung Hardware, GDCP-CMMP-M3-HW-...	Montage und Installation Motorcontroller CMMP-AS-...- M3 für alle Varianten/Leistungsklassen (1-phasig, 3-phasig), Steckerbelegungen, Fehlermeldungen, Wartung.
Beschreibung Funktionen, GDCP-CMMP-M3-FW-...	Funktionsbeschreibung (Firmware) CMMP-AS-...- M3 , Hinweise zur Inbetriebnahme.
Beschreibung Hardware, GDCP-CMMP-M0-HW-...	Montage und Installation Motorcontroller CMMP-AS-...- M0 für alle Varianten/Leistungsklassen (1-phasig, 3-phasig), Steckerbelegungen, Fehlermeldungen, Wartung.
Beschreibung Funktionen, GDCP-CMMP-M0-FW-...	Funktionsbeschreibung (Firmware) CMMP-AS-...- M0 , Hinweise zur Inbetriebnahme.
Beschreibung FHPP, GDCP-CMMP-M3/-M0-C-HP-...	Steuerung und Parametrierung des Motorcontrollers über das Festo-Profil FHPP. <ul style="list-style-type: none"> – Motorcontroller CMMP-AS-...-M3 mit folgenden Feldbussen: CANopen, PROFINET, PROFIBUS, EtherNet/IP, DeviceNet, EtherCAT. – Motorcontroller CMMP-AS-...-M0 mit Feldbus CANopen.
Beschreibung CiA 402 (DS 402), GDCP-CMMP-M3/-M0-C-CO-...	Steuerung und Parametrierung des Motorcontrollers über das Geräteprofil CiA 402 (DS402) <ul style="list-style-type: none"> – Motorcontroller CMMP-AS-...-M3 mit folgenden Feldbussen: CANopen und EtherCAT. – Motorcontroller CMMP-AS-...-M0 mit Feldbus CANopen.
Beschreibung CAM-Editor, P.BE-CMMP-CAM-SW-...	Kurvenscheiben-Funktionalität (CAM) des Motorcontrollers CMMP-AS-...- M3/-M0 .
Beschreibung Sicherheitsmodul, GDCP-CAMC-G-S1-...	Funktionale Sicherheitstechnik für den Motorcontroller CMMP-AS-...- M3 mit der Sicherheitsfunktion STO.
Beschreibung Sicherheitsmodul, GDCP-CAMC-G-S3-...	Funktionale Sicherheitstechnik für den Motorcontroller CMMP-AS-...- M3 mit den Sicherheitsfunktionen STO, SS1, SS2, SOS, SLS, SSR, SSM, SBC.
Beschreibung Sicherheitsfunktion STO, GDCP-CMMP-AS-M0-S1-...	Funktionale Sicherheitstechnik für den Motorcontroller CMMP-AS-...- M0 mit der integrierten Sicherheitsfunktion STO.
Beschreibung Austausch und Projektkonvertierung GDCP-CMMP-M3/-M0-RP-...	Motorcontroller CMMP-AS-...- M3/-M0 als Ersatzgerät für bisherige Motorcontroller CMMP-AS. Änderungen bei der elektrischen Installation und Beschreibung der Projektkonvertierung.
Hilfe zum FCT-PlugIn CMMP-AS	Oberfläche und Funktionen des PlugIn CMMP-AS für das Festo Configuration Tool. ➔ www.festo.com

Tab. 2 Dokumentationen zum Motorcontroller CMMP-AS-...-M3/-M0

1 Feldbus-Schnittstellen

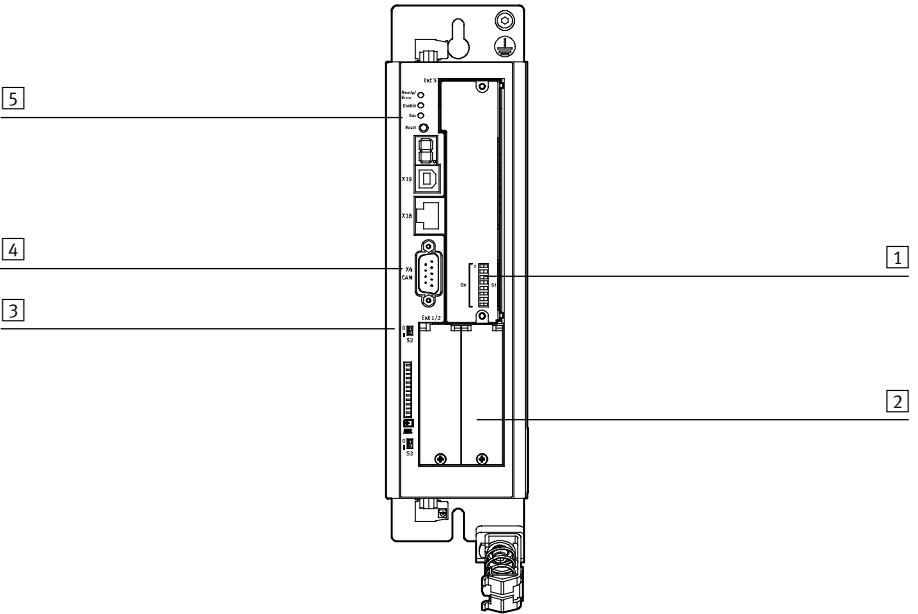
Die Steuerung und Parametrierung über CiA 402 wird beim CMMP-AS-...-M3/-M0 über die Feldbus-Schnittstellen entsprechend Tab. 1.1 unterstützt. Die CANopen-Schnittstelle ist im Motorcontroller integriert, über Interfaces kann der Motorcontroller um weitere Feldbus-Schnittstellen erweitert werden. Der Feldbus wird mit den DIP-Schaltern [S1] konfiguriert.

Feldbus	Schnittstelle	Beschreibung
CANopen	[X4] – integriert	➔ Kapitel 2
EtherCAT	Interface CAMC-EC	➔ Kapitel 4

Tab. 1.1 Feldbus-Schnittstellen für CiA 402

M0

Die Motorcontroller CMMP-AS-...-**M0** haben nur die Feldbusschnittstelle CANopen und keine Steckplätze für Interfaces, Schalter- oder Sicherheitsmodule.



- 1

DIP-Schalter [S1] für Feldbus-Einstellungen auf dem Schalter- oder Sicherheitsmodul in Steckplatz Ext3
- 2

Steckplätze Ext1/Ext2 für Interfaces
- 3

CANopen-Abschlusswiderstand [S2]
- 4

CANopen-Schnittstelle [X4]
- 5

CAN-LED

Fig. 1.1 Motorcontroller CMMP-AS-...-M3/-M0: Ansicht vorne, Beispiel mit Schaltermodul in Ext3

2 CANopen [X4]

2.1 Allgemeines zu CANopen

CANopen ist ein von der Vereinigung „CAN in Automation“ erarbeiteter Standard. In diesem Verbund ist eine Vielzahl von Geräteherstellern organisiert. Dieser Standard hat die bisherigen herstellerspezifischen CAN-Protokolle weitgehend ersetzt. Somit steht dem Endanwender ein herstellerunabhängiges Kommunikations-Interface zur Verfügung.

Von diesem Verbund sind unter anderem folgende Handbücher beziehbar:

CiA Draft Standard 201 ... 207:

In diesen Werken werden die allgemeinen Grundlagen und die Einbettung von CANopen in das OSI-Schichtenmodell behandelt. Die relevanten Punkte dieses Buches werden im vorliegenden CANopen-Handbuch vorgestellt, so dass der Erwerb der DS 201 ... 207 im Allgemeinen nicht notwendig ist.

CiA Draft Standard 301:

In diesem Werk werden der grundsätzliche Aufbau des Objektverzeichnisses eines CANopen-Gerätes und der Zugriff auf dieses beschrieben. Außerdem werden die Aussagen der DS201 ... 207 konkretisiert. Die für die Motorcontrollerfamilien CMMP benötigten Elemente des Objektverzeichnisses und die zugehörigen Zugriffsmethoden sind im vorliegenden Handbuch beschrieben. Der Erwerb der DS 301 ist ratsam aber nicht unbedingt notwendig.

CiA Draft Standard 402:

Dieses Buch befasst sich mit der konkreten Implementation von CANopen in Antriebsregler. Obwohl alle implementierten Objekte auch im vorliegenden CANopen-Handbuch in kurzer Form dokumentiert und beschrieben sind, sollte der Anwender über dieses Werk verfügen.

Bezugsadresse:

CAN in Automation (CiA) International Headquarter
Am Weichselgarten 26
D-91058 Erlangen
Tel.: 09131-601091
Fax: 09131-601092
➔ www.can-cia.de

Der CANopen-Implementierung des Motorcontrollers liegen folgende Standards zugrunde:

1	CiA Draft Standard 301,	Version 4.02,	13. Februar 2002
2	CiA Draft Standard Proposal 402,	Version 2.0,	26. Juli 2002

2.2 Verkabelung und Steckerbelegung

2.2.1 Anschlussbelegungen

Das CAN-Interface ist beim Motorcontroller CMMP-AS-...-M3/-M0 bereits integriert und somit immer verfügbar. Der CAN-Bus-Anschluss ist normgemäß als 9-poliger DSUB-Stecker ausgeführt.

[X4]	Pin Nr.	Bezeichnung	Wert	Beschreibung
	1	–	–	Nicht belegt
	6	CAN-GND	–	Masse
	2	CAN-L	–	Negiertes CAN-Signal (Dominant Low)
	7	CAN-H	–	Positives CAN-Signal (Dominant High)
	3	CAN-GND	–	Masse
	8	–	–	Nicht belegt
	4	–	–	Nicht belegt
	9	–	–	Nicht belegt
	5	CAN-Shield	–	Schirmung

Tab. 2.1 Steckerbelegung CAN-Interface [X4]



CAN-Bus-Verkabelung

Bei der Verkabelung der Motorcontroller über den CAN-Bus sollten Sie unbedingt die nachfolgenden Informationen und Hinweise beachten, um ein stabiles, störungsfreies System zu erhalten.

Bei einer nicht sachgemäßen Verkabelung können während des Betriebs Störungen auf dem CAN-Bus auftreten, die dazu führen, dass der Motorcontroller aus Sicherheitsgründen mit einem Fehler abschaltet.

Terminierung

Bei Bedarf kann ein Abschlusswiderstand (120 Ω) mittels DIP-Schalter S2 = 1 (CAN Term) auf dem Grundgerät zugeschaltet werden.

2.2.2 Verkabelungs-Hinweise

Der CAN-Bus bietet eine einfache und störungssichere Möglichkeit alle Komponenten einer Anlage miteinander zu vernetzen. Voraussetzung dafür ist allerdings, dass alle nachfolgenden Hinweise für die Verkabelung beachtet werden.

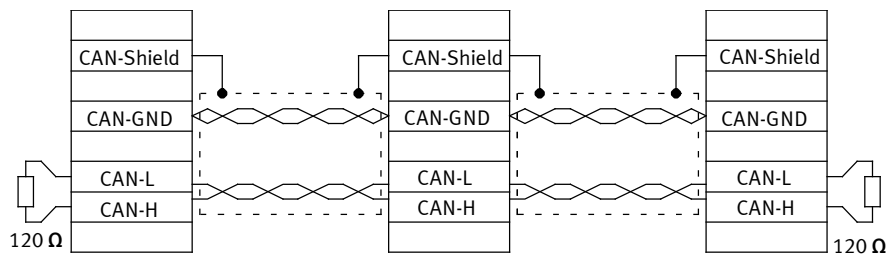


Fig. 2.1 Verkabelungsbeispiel

- Die einzelnen Knoten des Netzwerkes werden grundsätzlich linienförmig miteinander verbunden, so dass das CAN-Kabel von Controller zu Controller durchgeschleift wird (→ Fig. 2.1).
- An beiden Enden des CAN-Kabels muss jeweils genau ein Abschlusswiderstand von $120\ \Omega \pm 5\%$ vorhanden sein. Häufig ist in CAN-Karten oder in einer SPS bereits ein solcher Abschlusswiderstand eingebaut, der entsprechend berücksichtigt werden muss.
- Für die Verkabelung muss ein geschirmtes Kabel mit genau zwei verdrehten Aderpaaren verwendet werden.
Ein verdrehtes Aderpaar wird für den Anschluss von CAN-H und CAN-L verwendet. Die Adern des anderen Paares werden gemeinsam für CAN-GND verwendet. Der Schirm des Kabels wird bei allen Knoten an die CAN-Shield-Anschlüsse geführt. (Eine Tabelle mit den technischen Daten von verwendbaren Kabeln befindet sich am Ende dieses Kapitels.)
- Von der Verwendung von Zwischensteckern bei der CAN-Bus-Verkabelung wird abgeraten. Sollte dies dennoch notwendig sein, ist zu beachten, dass metallische Steckergehäuse verwendet werden, um den Kabelschirm zu verbinden.
- Um die Störeinkopplung so gering wie möglich zu halten, sollten grundsätzlich Motorleitungen nicht parallel zu Signalleitungen verlegt werden. Motorleitungen müssen gemäß der Spezifikation ausgeführt sein. Motorleitungen müssen ordnungsgemäß geschirmt und geerdet sein.
- Für weitere Informationen zum Aufbau einer störungsfreien CAN-Bus-Verkabelung verweisen wir auf die Controller Area Network protocol specification, Version 2.0 der Robert Bosch GmbH, 1991.

Eigenschaft		Wert
Aderpaare	–	2
Aderquerschnitt	[mm ²]	≥ 0,22
Schirmung	–	ja
Schleifenwiderstand	[Ω / m]	< 0,2
Wellenwiderstand	[Ω]	100...120

Tab. 2.2 Technische Daten CAN-Bus-Kabel

2.3 Konfiguration CANopen-Teilnehmer beim CMMP-AS-...-M3

M3

Dieser Abschnitt gilt nur für die Motorcontroller CMMP-AS-...-M3.

Zur Herstellung einer funktionsfähigen CANopen-Anschaltung sind mehrere Schritte erforderlich. Einige dieser Einstellungen sollten bzw. müssen vor der Aktivierung der CANopen-Kommunikation ausgeführt werden. Dieser Abschnitt liefert eine Übersicht über die auf Seiten des Slaves erforderlichen Schritte zur Parametrierung und Konfiguration. Da einige Parameter erst nach Speichern und Reset des Controllers wirksam werden, wird empfohlen, zuerst die Inbetriebnahme mit dem FCT ohne Anschluss an den CANopen-Bus vorzunehmen.



Hinweise zur Inbetriebnahme mit dem Festo Configuration Tool finden Sie in der Hilfe zum gerätespezifischen FCT-PlugIn.

Bei der Projektierung der CANopen-Anschaltung muss der Anwender daher diese Festlegungen treffen. Erst dann sollte die Parametrierung der Feldbus-Anbindung auf beiden Seiten erfolgen. Es wird empfohlen, zuerst die Parametrierung des Slaves durchzuführen. Danach wird der Master konfiguriert.

Folgendes Vorgehen wird empfohlen:

1. Einstellung des Offset der Knotennummer, der Bitrate und Aktivierung der Bus-Kommunikation über DIP-Schalter.



Der Zustand der DIP-Schalter wird bei Power-ON / RESET einmalig gelesen. Änderungen der Schalterstellungen im laufenden Betrieb übernimmt der CMMP-AS erst beim nächsten RESET oder Neustart

2. Parametrierung und Inbetriebnahme mit dem Festo Configuration Tool (FCT).
Insbesondere auf der Seite Anwendungsdaten:

- Steuerschnittstelle CANopen (Register Betriebsartenauswahl)

Außerdem folgende Einstellungen auf der Seite Feldbus:

- Basisadresse der Knotennummer
- Protokoll CANopen DS 402 (Register Betriebsparameter)
- physikalische Einheiten (Register Faktoren-Gruppe)




Beachten Sie, dass die Parametrierung der CANopen-Funktionalität nach einem Reset nur erhalten bleibt, wenn der Parametersatz des Motorcontrollers gesichert wurde. Während die FCT-Gerätesteuerung aktiv ist, wird die CAN-Kommunikation automatisch deaktiviert.

3. Konfiguration des CANopen-Masters → Abschnitte 2.5 und 3.

2.3.1 Einstellung der Knotennummer mit DIP-Schalter und FCT

Jedem Gerät im Netzwerk muss eine eindeutige Knotennummer zugeordnet werden. Die Knotennummer kann über die DIP-Schalter 1 ... 5 am Modul in Steckplatz Ext3 und im Programm FCT eingestellt werden.

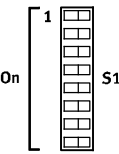


Die resultierende Knotennummer setzt sich zusammen aus der Basisadresse (FCT) und dem Offset (DIP-Schalter).

Zulässige Werte für die Knotennummer liegen im Bereich 1 ... 127.

Einstellung des Offset der Knotennummer mit DIP-Schalter

Die Einstellung der Knotennummer kann mit DIP-Schalter 1 ... 5 vorgenommen werden. Der über DIP-Schalter 1...5 eingestellte Offset der Knotennummer wird im Programm FCT auf der Seite Feldbus im Register Betriebsparameter angezeigt.


DIP-Schalter		Wert		Beispiel	
	1	ON	OFF		Wert
	1	1	0	ON	1
	2	2	0	ON	2
	3	4	0	OFF	0
	4	8	0	ON	8
	5	16	0	ON	16
Summe 1 ... 5 = Offset		1 ... 31 ¹⁾			27

1) Der Wert 0 für den Offset wird in Zusammenhang mit einer Basisadresse 0 als Knotennummer 1 interpretiert.
Eine Knotennummer größer 31 muss mit dem FCT eingestellt werden.

Tab. 2.3 Einstellung des Offset der Knotennummer

Einstellung der Basisadresse der Knotennummer mit FCT

Mit dem Festo-Configuration-Tool (FCT) wird die Knotennummer auf der Seite Feldbus im Register Betriebsparameter als Basisadresse eingestellt.
Default-Einstellung = 0 (das bedeutet Offset = Knotennummer).



Wird gleichzeitig über DIP-Schalter 1...5 und im Programm FCT eine Knotennummer vergeben, ist die resultierende Knotennummer die Summe von Basisadresse und Offset. Ist diese Summe größer als 127, wird der Wert automatisch auf 127 begrenzt.

2.3.2 Einstellung der Übertragungsrate mit DIP-Schalter

Die Übertragungsrate muss mit DIP-Schalter 6 und 7 auf dem Modul in Steckplatz Ext3 vorgenommen werden. Der Zustand der DIP-Schalter wird bei Power-ON/RESET einmalig gelesen. Änderungen der Schalterstellung im laufenden Betrieb übernimmt der CMMP-AS-...-M3 erst beim nächsten RESET.

Übertragungsrate		DIP-Schalter 6	DIP-Schalter 7
125	[Kbit/s]	OFF	OFF
250	[Kbit/s]	ON	OFF
500	[Kbit/s]	OFF	ON
1	[Mbit/s]	ON	ON

Tab. 2.4 Einstellung der Übertragungsrate

2.3.3 Aktivierung der CANopen-Kommunikation mit DIP-Schalter

Nach der Einstellung der Knotennummer und der Übertragungsrate kann die CANopen-Kommunikation aktiviert werden. Bitte denken Sie daran, dass die oben erwähnten Parameter nur geändert werden können, wenn das Protokoll deaktiviert ist.

CANopen-Kommunikation	DIP-Schalter 8
Deaktiviert	OFF
Aktiviert	ON

Tab. 2.5 Aktivierung der CANopen-Kommunikation

Bitte beachten Sie, dass die Aktivierung der CANopen-Kommunikation nur zur Verfügung steht, nachdem der Parametersatz (das FCT-Projekt) gespeichert und ein Reset durchgeführt wurde.



Wenn ein anderes Feldbus-Interface in Ext1 oder Ext2 gesteckt ist (→ Kapitel 1), wird mit DIP-Schalter 8 statt der CANopen-Kommunikation über [X4] der entsprechende Feldbus aktiviert.

2.3.4 Einstellung der physikalischen Einheiten (Faktoren-Gruppe)

Damit ein Feldbus-Master Positions-, Geschwindigkeits- und Beschleunigungsdaten in physikalischen Einheiten (z. B. mm, mm/s, mm/s²) mit dem Motorcontroller austauschen kann, müssen diese über die Faktoren-Gruppe parametrierung werden → Abschnitt 5.3.

Die Parametrierung kann über FCT oder den Feldbus erfolgen.

2.4 Konfiguration CANopen-Teilnehmer beim CMMP-AS-...-M0

M0

Dieser Abschnitt gilt nur für die Motorcontroller CMMP-AS-...-M0.

Zur Herstellung einer funktionsfähigen CANopen-Anschaltung sind mehrere Schritte erforderlich. Einige dieser Einstellungen sollten bzw. müssen vor der Aktivierung der CANopen-Kommunikation ausgeführt werden. Dieser Abschnitt liefert eine Übersicht über die auf Seiten des Slaves erforderlichen Schritte zur Parametrierung und Konfiguration.



Hinweise zur Inbetriebnahme mit dem Festo Configuration Tool finden Sie in der Hilfe zum gerätespezifischen FCT-PlugIn.

Bei der Projektierung der CANopen-Anschaltung muss der Anwender daher diese Festlegungen treffen. Erst dann sollte die Parametrierung der Feldbus-Anbindung auf beiden Seiten erfolgen. Es wird empfohlen, zuerst die Parametrierung des Slaves durchzuführen. Danach wird der Master konfiguriert.

Die Einstellungen der CAN Bus spezifischen Parameter kann auf zwei Wegen durchgeführt werden. Diese Wege sind voneinander getrennt und werden über die Option „Feldbusparametrierung über DINs“ auf der Seite „Anwendungsdaten“ im FCT umgeschaltet.

Im Auslieferungszustand und nach Zurücksetzen auf Werkseinstellungen ist die Option „Feldbusparametrierung über DINs“ aktiv. Eine Parametrierung mit FCT zur Aktivierung des CAN Bus ist somit nicht zwingend notwendig.

Folgende Parameter können über die DINs oder FCT eingestellt werden:

Parameter	Einstellung über	
	DIN	FCT
Knotennummer	0 ... 3 ¹⁾	Seite „Feldbus“, Betriebsparameter.
Übertragungsrate (Bitrate)	12, 13 ¹⁾	Die Aktivierung des CAN Bus wird automatisch durch FCT durchgeführt (abhängig von Gerätesteuerung): – Gerätesteuerung bei FCT → CAN deaktiviert – Gerätesteuerung abgegeben → CAN aktiviert
Aktivierung	8	
Protokoll (Datenprofil)	9 ²⁾	

1) Wird erst bei inaktiver CAN-Kommunikation übernommen

2) Wird erst nach Geräte-RESET übernommen

Tab. 2.6 Übersicht Einstellung der CAN-Parameter über DINs oder FCT

2.4.1 Einstellung der Knotennummer über DINs und FCT

Jedem Gerät im Netzwerk muss eine eindeutige Knotennummer zugeordnet werden.

Die Knotennummer kann über die digitalen Eingänge DIN0 DIN3 **und** im Programm FCT eingestellt werden.



Zulässige Werte für die Knotennummer liegen im Bereich 1 ... 127.

Einstellung des Offset der Knotennummer über DINs

Die Einstellungen der Knotennummer kann mittels Beschaltung der digitalen Eingänge DIN0 DIN3 vorgenommen werden. Der über die digitalen Eingänge eingestellte Offset der Knotennummer wird im Programm FCT auf der Seite „Feldbus“ im Register „Betriebsparameter“ angezeigt.

DINs	Wert		Beispiel	
	High	Low		Wert
0	1	0	High	1
1	2	0	High	2
2	4	0	Low	0
3	8	0	High	8
Summe 0 ... 3 = Knotennummer 0 ... 15				11

Tab. 2.7 Einstellung der Knotennummer

Einstellung der Basisadresse der Knotennummer über FCT

Mit FCT kann die Basisadresse der Knotennummer auf der Seite „Feldbus“ im Register „Betriebsparameter“ eingestellt werden.

Die resultierende Knotennummer ist abhängig von der Option "Feldbusparametrierung über DINs" auf der Seite "Anwendungsdaten". Ist diese Option aktiviert, ermittelt sich die Knotennummer aus der Addition der Basisadresse im FCT mit dem Offset über die digitalen Eingänge DIN0 ... 3.

Wenn die Option deaktiviert ist, entspricht die Basisadresse im FCT der resultierenden Knotennummer.

2.4.2 Einstellung der Übertragungsrate über DINs oder FCT

Die Übertragungsrate kann über die digitalen Eingänge DIN12 und DIN13 **oder** im FCT eingestellt werden.

Einstellung der Übertragungsrate über DINs

Übertragungsrate	DIN12	DIN13
125 [Kbit/s]	Low	Low
250 [Kbit/s]	High	Low
500 [Kbit/s]	Low	High
1 [Mbit/s]	High	High

Tab. 2.8 Einstellung der Übertragungsrate

Einstellung der Übertragungsrate über FCT

Mit FCT kann die Übertragungsrate auf der Seite „Feldbus“ im Register „Betriebsparameter“ eingestellt werden. Zuvor muss auf der Seite „Anwendungsdaten“ die Option „Feldbusparametrierung über DINs“ deaktiviert werden. Nach der Deaktivierung der Option sind DIN12 bzw. DIN13 wieder frei parametrierbar. Optional können sie mit dem FCT aber auch als AIN1 bzw. AIN2 parametriert werden.

2.4.3 Einstellung des Protokolls (Datenprofils) über DINs oder FCT

Über den digitalen Eingang DIN9 **oder** FCT kann das Protokoll (Datenprofil) eingestellt werden.

Einstellung der Protokolls (Datenprofil) über DINs

Protokoll (Datenprofil)	DIN9
CiA 402 (DS 402)	Low
FHPP	High

Tab. 2.9 Aktivierung der Protokolls (Datenprofil)

Einstellung des Protokolls (Datenprofils) über FCT

Mit FCT wird das Protokoll auf der Seite „Feldbus“ im Register „Betriebsparameter“ eingestellt.

2.4.4 Aktivierung der CANopen-Kommunikation über DINs oder FCT

Nach der Einstellung der Knotennummer, der Übertragungsrate und des Protokolls (Datenprofil) kann die CANopen-Kommunikation aktiviert werden.

Aktivierung der CANopen-Kommunikation über DIN

CANopen-Kommunikation	DIN8
Deaktiviert	Low
Aktiviert	High

Tab. 2.10 Aktivierung der CANopen-Kommunikation



Zur Aktivierung per digitalem Eingang ist kein erneuter Gerätereset notwendig. Der CAN Bus wird sofort nach Pegeländerung (Low → High) an DIN8 aktiviert.

Aktivierung der CANopen-Kommunikation über FCT

Die CANopen-Kommunikation wird automatisch durch das FCT aktiviert, wenn die Option „Feldbusparametrierung über DINs“ deaktiviert ist.



Solange die Gerätesteuerung bei FCT liegt, ist der CAN Bus ausgeschaltet.

2.4.5 Einstellung der physikalischen Einheiten (Faktoren-Gruppe)

Damit ein Feldbus-Master Positions-, Geschwindigkeits- und Beschleunigungsdaten in physikalischen Einheiten (z. B. mm, mm/s, mm/s²) mit dem Motorcontroller austauschen kann, müssen diese über die Faktoren-Gruppe parametrierung werden → Abschnitt 5.3.

Die Parametrierung kann über FCT oder den Feldbus erfolgen.

2.5 Konfiguration CANopen-Master

Zur Konfiguration des CANopen-Masters können Sie eine EDS-Datei verwenden.

Die EDS-Datei ist auf der dem Motorcontroller beigelegten CD-ROM enthalten.



Die aktuellsten Versionen finden Sie unter → www.festo.com

EDS-Dateien	Beschreibung
CMMP-AS-...-M3.eds	Motorcontroller CMMP-AS-...- M3 mit Protokoll „CiA402 (DS402)“
CMMP-AS-...-M0.eds	Motorcontroller CMMP-AS-...- M0 mit Protokoll „CiA402 (DS402)“

Tab. 2.11 EDS-Dateien für CANopen

3 Zugriffsverfahren CANopen

3.1 Einleitung

CANopen stellt eine einfache und standardisierte Möglichkeit bereit, auf die Parameter des Motor-controllers (z. B. den maximalen Motorstrom) zuzugreifen. Dazu ist jedem Parameter (CAN-Objekt) eine eindeutige Nummer (Index und Subindex) zugeordnet. Die Gesamtheit aller einstellbaren Parameter-wird als Objektverzeichnis bezeichnet.

Für den Zugriff auf die CAN-Objekte über den CAN-Bus sind im Wesentlichen zwei Methoden verfügbar: Eine bestätigte Zugriffsart, bei der der Motorcontroller jeden Parameterzugriff quittiert (über sog. SDOs) und eine unbestätigte Zugriffsart, bei der keine Quittierung erfolgt (über sog. PDOs).

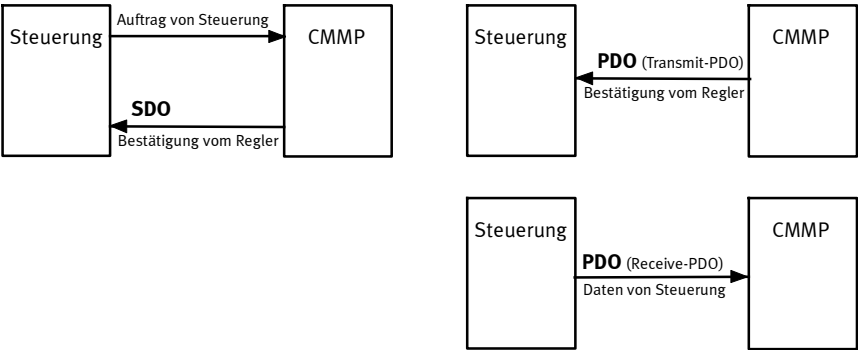


Fig. 3.1 Zugriffsverfahren

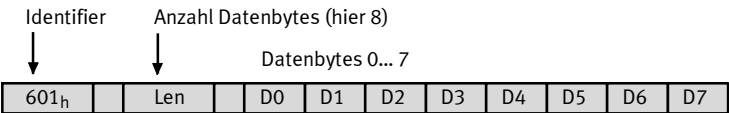
In der Regel wird der Motorcontroller über SDO-Zugriffe sowohl parametrieret als auch gesteuert. Für spezielle Anwendungsfälle sind darüber hinaus noch weitere Arten von Nachrichten (sog. Kommunikations-Objekte) definiert, die entweder vom Motorcontroller oder der übergeordneten Steuerung gesendet werden:

Kommunikations-Objekte		
SDO	Service Data Objekt	Werden zur normalen Parametrierung des Motor-controllers verwendet.
PDO	Process Data Object	Schneller Austausch von Prozessdaten (z. B. Iststdrehzahl) möglich
SYNC	Synchronisation Message	Synchronisierung mehrerer CAN-Knoten
EMCY	Emergency Message	Übermittlung von Fehlermeldungen

Kommunikations-Objekte		
NMT	Network Management	Netzwerkdienst: Es kann z. B. auf alle CAN-Knoten gleichzeitig eingewirkt werden.
HEART-BEAT	Error Control Protocol	Überwachung der Kommunikationsteilnehmer durch regelmäßige Nachrichten.

Tab. 3.1 Kommunikations-Objekte

Jede Nachricht, die auf dem CAN-Bus verschickt wird, enthält eine Art Adresse, mit dessen Hilfe festgestellt werden kann, für welchen Bus-Teilnehmer die Nachricht gedacht ist. Diese Nummer wird als Identifier bezeichnet. Je niedriger der Identifier, desto größer ist die Priorität der Nachricht. Für die oben genannten Kommunikationsobjekte sind jeweils Identifier festgelegt. Die folgende Skizze zeigt den prinzipiellen Aufbau einer CANopen-Nachricht:



3.2 SDO-Zugriff

Über die Service-Data-Objekte (SDO) kann auf das Objektverzeichnis des Motorcontrollers zugegriffen werden. Dieser Zugriff ist besonders einfach und übersichtlich. Es wird daher empfohlen, die Applikation zunächst nur mit SDOs aufzubauen und erst später einige Objektzugriffe auf die zwar schnelleren, aber auch komplizierteren Process Data Objekte (PDOs) umzustellen.

SDO-Zugriffe gehen immer von der übergeordneten Steuerung (Host) aus. Dieser sendet an den Motorcontroller entweder einen Schreibbefehl, um einen Parameter des Objektverzeichnisses zu ändern, oder einen Lesebefehl, um einen Parameter auszulesen. Zu jedem Befehl erhält der Host eine Antwort, die entweder den ausgelesenen Wert enthält oder – im Falle eines Schreibbefehls – als Quittung dient. Damit der Motorcontroller erkennt, dass der Befehl für ihn bestimmt ist, muss der Host den Befehl mit einem bestimmten Identifier senden. Dieser setzt sich aus der Basis 600_h + Knotennummer des betreffenden Motorcontrollers zusammen. Der Motorcontroller antwortet entsprechend mit dem Identifier 580_h + Knotennummer.

Der Aufbau der Befehle bzw. der Antworten hängt vom Datentyp des zu lesenden oder schreibenden Objekts ab, da entweder 1, 2 oder 4 Datenbytes gesendet bzw. empfangen werden müssen. Folgende Datentypen werden unterstützt:

Datentyp	Größe und Vorzeichen	Bereich
UINT8	8-Bit-Wert ohne Vorzeichen	0 ... 255
INT8	8-Bit-Wert mit Vorzeichen	-128 ... 127
UINT16	16-Bit-Wert ohne Vorzeichen	0 ... 65535
INT16	16-Bit-Wert mit Vorzeichen	-32768 ... 32767
UINT32	32-Bit-Wert ohne Vorzeichen	0 ... (2 ³² -1)
INT32	32-Bit-Wert mit Vorzeichen	-(2 ³¹) ... (2 ³² -1)

Tab. 3.2 Unterstützte Datentypen

3.2.1 SDO-Sequenzen zum Lesen und Schreiben

Um Objekte dieser Zahlentypen auszulesen oder zu beschreiben sind die nachfolgend aufgeführten Sequenzen zu verwenden. Die Kommandos, um einen Wert in den Motorcontroller zu schreiben, beginnen je nach Datentyp mit einer unterschiedlichen Kennung. Die Antwortkennung ist hingegen stets die gleiche. Lesebefehle beginnen immer mit der gleichen Kennung und der Motorcontroller antwortet je nach zurückgegebenem Datentyp unterschiedlich. Alle Zahlen sind in hexadezimaler Schreibweise gehalten.

Kennung	8 Bit	16 Bit	32 Bit
Auftragskennung	2F _h	2B _h	23 _h
Antwortkennung	4F _h	4B _h	43 _h
Antwortkennung bei Fehler	–	–	80 _h

Tab. 3.3 SDO – Antwort-/Auftragskennung

BEISPIEL		
UINT8/INT8	Lesen von Obj. 6061_00 _h Rückgabe-Daten: 01 _h	Schreiben von Obj. 1401_02 _h Daten: EF _h
Befehl	40 _h 61 _h 60 _h 00 _h	2F _h 01 _h 14 _h 02 _h EF _h
Antwort:	4F _h 61 _h 60 _h 00 _h 01 _h	60 _h 01 _h 14 _h 02 _h
UINT16/INT16	Lesen von Obj. 6041_00 _h Rückgabe-Daten: 1234 _h	Schreiben von Obj. 6040_00 _h Daten: 03E8 _h
Befehl	40 _h 41 _h 60 _h 00 _h	2B _h 40 _h 60 _h 00 _h E8 _h 03 _h
Antwort:	4B _h 41 _h 60 _h 00 _h 34 _h 12 _h	60 _h 40 _h 60 _h 00 _h
UINT32/INT32	Lesen von Obj. 6093_01 _h Rückgabe-Daten: 12345678 _h	Schreiben von Obj. 6093_01 _h Daten: 12345678 _h
Befehl	40 _h 93 _h 60 _h 01 _h	23 _h 93 _h 60 _h 01 _h 78 _h 56 _h 34 _h 12 _h
Antwort:	43 _h 93 _h 60 _h 01 _h 78 _h 56 _h 34 _h 12 _h	60 _h 93 _h 60 _h 01 _h



Vorsicht
Die Quittierung vom Motorcontroller muss in jedem Fall abgewartet werden!
Erst wenn der Motorcontroller die Anforderung quittiert hat, dürfen weitere Anforderungen gesendet werden.

3.2.2 SDO-Fehlermeldungen

Im Falle eines Fehlers beim Lesen oder Schreiben (z. B. weil der geschriebene Wert zu groß ist), antwortet der Motorcontroller mit einer Fehlermeldung anstelle der Quittierung:

Befehl	23 _h	41 _h	60 _h	00 _h
Antwort:	80 _h	41 _h	60 _h	00 _h	02 _h	00 _h	01 _h	06 _h
	↑				↑	↑	↑	↑
	Fehler-Kennung				Fehlercode (4 Byte)			

Fehlercode F3 F2 F1 F0	Bedeutung
05 03 00 00 _h	Protokollfehler: Toggle Bit wurde nicht geändert
05 04 00 01 _h	Protokollfehler: client/server command specifier ungültig oder unbekannt
06 06 00 00 _h	Zugriff fehlerhaft aufgrund einer Hardware-Problems ¹⁾
06 01 00 00 _h	Zugriffsart wird nicht unterstützt.
06 01 00 01 _h	Lesezugriff auf ein Objekt, dass nur geschrieben werden kann
06 01 00 02 _h	Schreibzugriff auf ein Objekt, dass nur gelesen werden kann
06 02 00 00 _h	Das angesprochene Objekt existiert nicht im Objektverzeichnis
06 04 00 41 _h	Das Objekt darf nicht in ein PDO eingetragen werden (z. B. ro-Objekt in RPDO)
06 04 00 42 _h	Die Länge der in das PDO eingetragenen Objekte überschreitet die PDO-Länge
06 04 00 43 _h	Allgemeiner Parameterfehler
06 04 00 47 _h	Überlauf einer internen Größe/Genereller Fehler
06 07 00 10 _h	Protokollfehler: Länge des Service-Parameters stimmt nicht überein
06 07 00 12 _h	Protokollfehler: Länge des Service-Parameters zu groß
06 07 00 13 _h	Protokollfehler: Länge des Service-Parameters zu klein
06 09 00 11 _h	Der angesprochene Subindex existiert nicht
06 09 00 30 _h	Die Daten überschreiten den Wertebereich des Objekts
06 09 00 31 _h	Die Daten sind zu groß für das Objekt
06 09 00 32 _h	Die Daten sind zu klein für das Objekt
06 09 00 36 _h	Obere Grenze ist kleiner als untere Grenze
08 00 00 20 _h	Daten können nicht übertragen oder gespeichert werden ¹⁾
08 00 00 21 _h	Daten können nicht übertragen oder gespeichert werden, da der Regler lokal arbeitet
08 00 00 22 _h	Daten können nicht übertragen oder gespeichert werden, da sich der Regler dafür nicht im richtigen Zustand befindet ²⁾
08 00 00 23 _h	Es ist kein Object Dictionary vorhanden ³⁾

1) Werden gemäß CiA 301 bei fehlerhaftem Zugriff auf store_parameters/restore_parameters zurückgegeben.

2) „Zustand“ ist hier allgemein zu verstehen: Es kann sich dabei sowohl um die falsche Betriebsart handeln, als auch um ein nicht vorhandenes Technologie-Modul o. ä.

3) Dieser Fehler wird z. B. zurückgegeben, wenn ein anderes Bussystem den Motorcontroller kontrolliert oder der Parameterzugriff nicht erlaubt ist.

3.2.3 Simulation von SDO-Zugriffen

Die Firmware der Motorcontroller bietet die Möglichkeit, SDO-Zugriffe zu simulieren. So können in der Testphase Objekte nach dem Einschreiben über den CAN-Bus über die das CI-Terminal der Parametrierungssoftware gelesen und kontrolliert werden.

Die Syntax der Befehle lautet:

Lesebefehle		Schreibbefehle
↓ Hauptindex (hex)		
↓ Subindex (hex)		
UINT8/INT8		
Befehl	? XXXX SU	= XXXX SU: WW
Antwort:	= XXXX SU: WW	= XXXX SU: WW
UINT16/INT16	↑ 8 Bit Daten (hex)	
Befehl	? XXXX SU	= XXXX SU: WWWW
Antwort:	= XXXX SU: WWWW	= XXXX SU: WWWW
UINT32/INT32	↑ 16 Bit Daten (hex)	
Befehl	? XXXX SU	= XXXX SU:
Antwort:	= XXXX SU: WWWWWWW	= XXXX SU: WWWWWWW
	↑ 32 Bit Daten (hex)	

Beachten Sie, dass die Befehle als Zeichen ohne jegliche Leerzeichen eingegeben werden.

Lesefehler	Schreibfehler
Befehl	? XXXX SU
Antwort:	= XXXX SU: WWWWWWW ¹⁾
	! FFFFFFFF
↑ 32 Bit Fehlercode	↑ 32 Bit Fehlercode
F3 F2 F1 F0 gemäß Kap. 3.2.2	F3 F2 F1 F0 gemäß Kap. 3.2.2

1) Die Antwort ist im Fehlerfall für alle 3 Schreibbefehle (8, 16, 32 Bit) gleich aufgebaut.

Die Befehle werden als Zeichen ohne jegliche Leerzeichen eingegeben.



Vorsicht
Verwenden sie diese Testbefehle niemals in Applikationen!
Der Zugriff dient lediglich zu Testzwecken und ist nicht für eine echtzeitfähige Kommunikation geeignet.
Darüber hinaus kann die Syntax der Testbefehle jederzeit geändert werden.

3.3 **PDO-Message**

Mit Process-Data-Objekten (PDOs) können Daten ereignisgesteuert oder zyklisch übertragen werden. Das PDO überträgt dabei einen oder mehrere vorher festgelegte Parameter. Anders als bei einem SDO erfolgt bei der Übertragung eines PDOs keine Quittierung. Nach der PDO-Aktivierung müssen daher alle Empfänger jederzeit eventuell ankommende PDOs verarbeiten können. Dies bedeutet meistens einen erheblichen Softwareaufwand im Host-Rechner. Diesem Nachteil steht der Vorteil gegenüber, dass der Host-Rechner die durch ein PDO übertragenen Parameter nicht zyklisch abzufragen braucht, was zu einer starken Verminderung der CAN-Busauslastung führt.

BEISPIEL

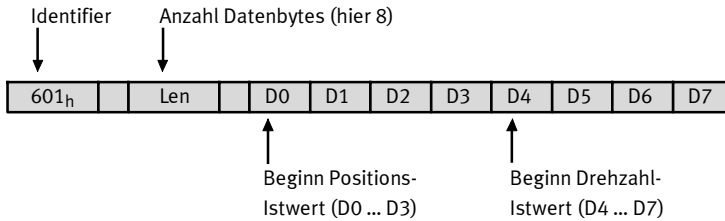
Der Host-Rechner möchte wissen, wann der Motorcontroller eine Positionierung von A nach B abgeschlossen hat.
Bei der Verwendung von SDOs muss er hierzu ständig, beispielsweise jede Millisekunde, das Objekt statusword abfragen, womit er die Buskapazität stark auslastet.
Bei der Verwendung eines PDOs wird der Motorcontroller schon beim Start der Applikation so parametriert, dass er bei jeder Veränderung des Objektes statusword ein PDO absetzt, in dem das Objekt statusword enthalten ist.
Statt ständig nachzufragen, wird dem Host-Rechner somit automatisch eine entsprechende Meldung zugestellt, sobald das Ereignis eingetreten ist.

Folgende Typen von PDOs werden unterschieden:

Typ	Weg	Bemerkung
Transmit-PDO	Motorcontroller → Host	Motorcontroller sendet PDO bei Auftreten eines bestimmten Ereignisses.
Receive-PDO	Host → Motorcontroller	Motorcontroller wertet PDO bei Auftreten eines bestimmten Ereignisses aus.

Tab. 3.4 PDO-Typen

Der Motorcontroller verfügt über vier Transmit- und vier Receive-PDOs.
In die PDOs können nahezu alle Objekte des Objektverzeichnisses eingetragen (gemappt) werden, d. h. das PDO enthält als Daten z. B. den Drehzahl-Istwert, den Positions-Istwert o. ä. Welche Daten übertragen werden, muss dem Motorcontroller vorher mitgeteilt werden, da das PDO lediglich Nutzdaten und keine Information über die Art des Parameters enthält. In der unteren Beispiel würde in den Datenbytes 0 ... 3 des PDOs der Positions-Istwert und in den Bytes 4 ... 7 der Drehzahl-Istwert übertragen.



Auf diese Art können nahezu beliebige Datentelegramme definiert werden. Die folgenden Kapitel beschreiben die dazu nötigen Einstellungen.

3.3.1 Beschreibung der Objekte

Objekt	Bemerkung
COB_ID_used_by_PDO	In dem Objekt COB_ID_used_by_PDO ist der Identifier einzutragen, auf dem das jeweilige PDO gesendet bzw. empfangen werden soll. Ist Bit 31 gesetzt, ist das jeweilige PDO deaktiviert. Dies ist die Voreinstellung für alle PDOs. Die COB-ID darf nur geändert werden, wenn das PDO deaktiviert, d. h. Bit 31 gesetzt ist. Ein anderer Identifier als aktuell im Regler eingestellt darf daher nur geschrieben werden, wenn gleichzeitig Bit 31 gesetzt ist. Das gesetzte Bit 30 beim Lesen des Identifiers zeigt an, dass das Objekt nicht durch ein Remoteframe abgefragt werden kann. Dieses Bit wird beim Schreiben ignoriert und ist beim Lesen immer gesetzt
number_of_mapped_objects	Dieses Objekt gibt an, wie viele Objekte in das entsprechende PDO gemappt werden sollen. Folgende Einschränkungen sind zu beachten: Es können pro PDO maximal 4 Objekte gemappt werden Ein PDO darf über maximal 64 Bit (8 Byte) verfügen.
first_mapped_object ... fourth_mapped_object	Für jedes Objekt, das im PDO enthalten sein soll muss dem Motorcontroller der entsprechende Hauptindex, der Subindex und die Länge mitgeteilt werden. Die Längenangabe muss mit der Längenangabe im Object Dictionary übereinstimmen. Teile eines Objekts können nicht gemappt werden. Die Mapping-Informationen besitzen folgendes Format → Tab. 3.6
transmission_type und inhibit_time	Für jedes PDO kann festgelegt werden, welches Ereignis zum Aus-senden (Transmit-PDO) bzw. Auswerten (Receive-PDO) einer Nachricht führt. → Tab. 3.7

Objekt	Bemerkung
Transmit_mask_high und transmit_mask_low	Wird als transmission_type „Änderung“ gewählt, wird das TPDO immer gesendet, wenn sich mindestens 1 Bit des TPDOs ändert. Häufig wird es aber benötigt, dass das TPDO nur gesendet wird, wenn sich bestimmte Bits geändert haben. Daher kann das TPDO mit einer Maske versehen werden: Nur die Bits des TPDOs, die in der Maske auf „1“ gesetzt sind, werden zur Auswertung, ob sich das PDO geändert hat herangezogen. Da diese Funktion herstellerspezifisch ist, sind als Defaultwert alle Bits der Masken gesetzt.

Tab. 3.5 Beschreibung der Objekte

xxx_mapped_object		
Hauptindex (hex)	[Bit]	16
Subindex (hex)	[Bit]	8
Länge des Objekts (hex)	[Bit]	8

Tab. 3.6 Format der Mapping-Informationen

Zur Vereinfachung des Mappings ist folgendes Vorgehen vorgeschrieben:

1. Die Anzahl der gemappten Objekte wird auf 0 gesetzt.
2. Die Parameter first_mapped_object ... fourth_mapped_object dürfen beschrieben werden (Die Gesamtlänge aller Objekte ist in dieser Zeit nicht relevant).
3. Die Anzahl der gemappten Objekte wird auf einen Wert zwischen 1 ... 4 gesetzt. Die Länge all dieser Objekte darf jetzt 64 Bit nicht überschreiten.

Wert	Bedeutung	Erlaubt bei
01 _h – F0 _h	SYNC-Message Der Zahlenwert gibt an, wie viel SYNC-Nachrichten eingetroffen sein müssen, bevor das PDO – gesendet (T-PDO) bzw. – ausgewertet (R-PDO) wird.	TPDOs RPDOs
FE _h	Zyklisch Das Transfer-PDO wird vom Motorcontroller zyklisch aktualisiert und gesendet. Die Zeitspanne wird durch das Objekt inhibit_time festgelegt. Receive-PDOs werden hingegen unmittelbar nach Empfang ausgewertet.	TPDOs (RPDOs)
FF _h	Änderung Das Transfer-PDO wird gesendet, wenn sich in den Daten des PDOs mindestens 1 Bit geändert hat. Mit inhibit_time kann zusätzlich der minimale Abstand zwischen dem Absenden zweier PDOs in 100 µs-Schritten festgelegt werden.	TPDOs

Tab. 3.7 Übertragungsart

Die Verwendung aller anderen Werte ist nicht zulässig.

BEISPIEL

Folgende Objekte sollen zusammen in einem PDO übertragen werden:

Name des Objekts	Index_Subindex	Bedeutung
statusword	6041 _h _00 _h	Controllersteuerung
modes_of_operation_display	6061 _h _00 _h	Betriebsart
digital_inputs	60FD _h _00 _h	Digitale Eingänge

Es soll das erste Transmit-PDO (TPDO 1) verwendet werden, welches immer gesendet werden soll, wenn sich eines der digitalen Eingänge ändert, allerdings maximal alle 10 ms. Als Identifier für dieses PDO soll 187_h verwendet werden.

1. PDO deaktivieren
Falls das PDO aktiv ist, muss es zuerst deaktiviert werden.
Schreiben des Identifiers mit gesetztem Bit 31 (PDO ist deaktiviert):
→ cob_id_used_by_pdo = C0000187_h
2. Anzahl der Objekte löschen
Damit das Objektmapping geändert werden darf, Anzahl der Objekte auf Null setzen.
→ number_of_mapped_objects = 0
3. Objekte, die gemappt werden sollen, parametrieren
Die oben aufgeführten Objekte müssen jeweils zu einem 32 Bit-Wert zusammengesetzt werden:

Index	Subindex		→ first_mapped_object = 60410010 _h
= 6041 _h	= 00 _h	Länge = 10 _h	
Index	Subindex		→ second_mapped_object = 60610008 _h
= 6061 _h	= 00 _h	Länge = 08 _h	
Index	Subindex		→ third_mapped_object = 60FD0020 _h
= 60FD _h	= 00 _h	Länge = 20 _h	
4. Anzahl der Objekte parametrieren
es sollen 3 Objekte im PDO enthalten sein
→ number_of_mapped_objects = 3_h
5. Übertragungsart parametrieren
Das PDO soll bei Änderung (der digitalen Eingänge) gesendet werden.
Damit nur die Änderung der digitalen Eingänge zum Senden führt, wird das PDO maskiert, so dass nur die 16 Bits des Objekts 60FD_h „durchkommen“.
Das PDO soll höchstens alle 10 ms (100D100 µs) gesendet werden.
→ transmit_mask_high = 00FFFF00_h
→ transmit_mask_low = 00000000_h
→ inhibit_time = 64_h
6. Identifier parametrieren
Das PDO soll mit Identifier 187_h gesendet werden.
Schreiben des neuen Identifier und Aktivieren des PDOs durch Löschen von Bit 31:
→ cob_id_used_by_pdo = 40000187_h



Beachten Sie, dass die Parametrierung der PDOs generell nur geändert werden darf, wenn der Netzwerkstatus (NMT) nicht operational ist. → Kapitel 3.3.3

3.3.2 Objekte zur PDO-Parametrierung

In den Motorcontrollern der CMMP-Reihe sind insgesamt 4 Transmit und 4 Receive-PDOs verfügbar. Die einzelnen Objekte, um diese PDOs zu parametrieren sind jeweils für alle 4 TPDOs und alle 4 RPDOs gleich. Daher ist im Folgenden nur die Parameterbeschreibung des ersten TPDOs explizit aufgeführt. Sie ist sinngemäß auch für die anderen PDOs zu verwenden, die im Anschluss tabellarisch aufgeführt sind:

Index	1800_h
Name	transmit_pdo_parameter_tpdo1
Object Code	RECORD
No. of Elements	3

Sub-Index	01_h
Description	cob_id_used_by_pdo_tpdo1
Data Type	UINT32
Access	rw
PDO Mapping	no
Units	–
Value Range	181 _h ... 1FF _h , Bit 30 und 31 dürfen gesetzt sein
Default Value	C0000181 _h

Sub-Index	02_h
Description	transmission_type_tpdo1
Data Type	UINT8
Access	rw
PDO Mapping	no
Units	–
Value Range	0 ... 8C _h , FE _h , FF _h
Default Value	FF _h

Sub-Index	03_h
Description	inhibit_time_tpdo1
Data Type	UINT16
Access	rw
PDO Mapping	no
Units	100 µs (i.e. 10 = 1ms)
Value Range	–
Default Value	0

Index	1A00_h
Name	transmit_pdo_mapping_tpdo1
Object Code	RECORD
No. of Elements	4

Sub-Index	00_h
Description	number_of_mapped_objects_tpdo1
Data Type	UINT8
Access	rw
PDO Mapping	no
Units	–
Value Range	0 ... 4
Default Value	→ Tabelle

Sub-Index	01_h
Description	first_mapped_object_tpdo1
Data Type	UINT32
Access	rw
PDO Mapping	no
Units	–
Value Range	–
Default Value	→ Tabelle

Sub-Index	02_h
Description	second_mapped_object_tpdo1
Data Type	UINT32
Access	rw
PDO Mapping	no
Units	–
Value Range	–
Default Value	→ Tabelle

Sub-Index	03_h
Description	third_mapped_object_tpdo1
Data Type	UINT32
Access	rw
PDO Mapping	no
Units	–
Value Range	–
Default Value	→ Tabelle

Sub-Index	04_h
Description	fourth_mapped_object_tpdo1
Data Type	UINT32
Access	rw
PDO Mapping	no
Units	–
Value Range	–
Default Value	→ Tabelle



Beachten Sie, dass die Objekt-Gruppen transmit_pdo_parameter_xxx und transmit_pdo_mapping_xxx nur beschrieben werden können, wenn das PDO deaktiviert ist (Bit 31 in cob_id_used_by_pdo_xxx gesetzt)

1. Transmit-PDO

Index	Comment	Type	Acc.	Default Value
1800 _h _00 _h	number of entries	UINT8	ro	03 _h
1800 _h _01 _h	COB-ID used by PDO	UINT32	rw	C0000181 _h
1800 _h _02 _h	transmission type	UINT8	rw	FF _h
1800 _h _03 _h	inhibit time (100 µs)	UINT16	rw	0000 _h
1A00 _h _00 _h	number of mapped objects	UINT8	rw	01 _h
1A00 _h _01 _h	first mapped object	UINT32	rw	60410010 _h
1A00 _h _02 _h	second mapped object	UINT32	rw	00000000 _h
1A00 _h _03 _h	third mapped object	UINT32	rw	00000000 _h
1A00 _h _04 _h	fourth mapped object	UINT32	rw	00000000 _h

2. Transmit-PDO

Index	Comment	Type	Acc.	Default Value
1801 _h _00 _h	number of entries	UINT8	ro	03 _h
1801 _h _01 _h	COB-ID used by PDO	UINT32	rw	C0000281 _h
1801 _h _02 _h	transmission type	UINT8	rw	FF _h
1801 _h _03 _h	inhibit time (100 µs)	UINT16	rw	0000 _h
1A01 _h _00 _h	number of mapped objects	UINT8	rw	02 _h
1A01 _h _01 _h	first mapped object	UINT32	rw	60410010 _h
1A01 _h _02 _h	second mapped object	UINT32	rw	60610008 _h
1A01 _h _03 _h	third mapped object	UINT32	rw	00000000 _h
1A01 _h _04 _h	fourth mapped object	UINT32	rw	00000000 _h

3. Transmit-PDO

Index	Comment	Type	Acc.	Default Value
1802 _h _00 _h	number of entries	UINT8	ro	03 _h
1802 _h _01 _h	COB-ID used by PDO	UINT32	rw	C0000381 _h
1802 _h _02 _h	transmission type	UINT8	rw	FF _h
1802 _h _03 _h	inhibit time (100 µs)	UINT16	rw	0000 _h
1A02 _h _00 _h	number of mapped objects	UINT8	rw	02 _h
1A02 _h _01 _h	first mapped object	UINT32	rw	60410010 _h
1A02 _h _02 _h	second mapped object	UINT32	rw	60640020 _h
1A02 _h _03 _h	third mapped object	UINT32	rw	00000000 _h
1A02 _h _04 _h	fourth mapped object	UINT32	rw	00000000 _h

4. Transmit-PDO

Index	Comment	Type	Acc.	Default Value
1803 _h _00 _h	number of entries	UINT8	ro	03 _h
1803 _h _01 _h	COB-ID used by PDO	UINT32	rw	C0000481 _h
1803 _h _02 _h	transmission type	UINT8	rw	FF _h
1803 _h _03 _h	inhibit time (100 µs)	UINT16	rw	0000 _h
1A03 _h _00 _h	number of mapped objects	UINT8	rw	02 _h
1A03 _h _01 _h	first mapped object	UINT32	rw	60410010 _h
1A03 _h _02 _h	second mapped object	UINT32	rw	606C0020 _h
1A03 _h _03 _h	third mapped object	UINT32	rw	00000000 _h
1A03 _h _04 _h	fourth mapped object	UINT32	rw	00000000 _h

tpdo_1_transmit_mask

Index	Comment	Type	Acc.	Default Value
2014 _h _00 _h	number of entries	UINT8	ro	02 _h
2014 _h _01 _h	tpdo_1_transmit_mask_low	UINT32	rw	FFFFFFFF _h
2014 _h _02 _h	tpdo_1_transmit_mask_high	UINT32	rw	FFFFFFFF _h

tpdo_2_transmit_mask

Index	Comment	Type	Acc.	Default Value
2015 _h _00 _h	number of entries	UINT8	ro	02 _h
2015 _h _01 _h	tpdo_2_transmit_mask_low	UINT32	rw	FFFFFFFF _h
2015 _h _02 _h	tpdo_2_transmit_mask_high	UINT32	rw	FFFFFFFF _h

tpdo_3_transmit_mask

Index	Comment	Type	Acc.	Default Value
2016 _h 00 _h	number of entries	UINT8	ro	02 _h
2016 _h 01 _h	tpdo_3_transmit_mask_low	UINT32	rw	FFFFFFF _h
2016 _h 02 _h	tpdo_3_transmit_mask_high	UINT32	rw	FFFFFFF _h

tpdo_4_transmit_mask

Index	Comment	Type	Acc.	Default Value
2017 _h 00 _h	number of entries	UINT8	ro	02 _h
2017 _h 01 _h	tpdo_4_transmit_mask_low	UINT32	rw	FFFFFFF _h
2017 _h 02 _h	tpdo_4_transmit_mask_high	UINT32	rw	FFFFFFF _h

1. Receive-PDO

Index	Comment	Type	Acc.	Default Value
1400 _h 00 _h	number of entries	UINT8	ro	02 _h
1400 _h 01 _h	COB-ID used by PDO	UINT32	rw	C0000201 _h
1400 _h 02 _h	transmission type	UINT8	rw	FF _h
1600 _h 00 _h	number of mapped objects	UINT8	rw	01 _h
1600 _h 01 _h	first mapped object	UINT32	rw	60400010 _h
1600 _h 02 _h	second mapped object	UINT32	rw	00000000 _h
1600 _h 03 _h	third mapped object	UINT32	rw	00000000 _h
1600 _h 04 _h	fourth mapped object	UINT32	rw	00000000 _h

2. Receive-PDO

Index	Comment	Type	Acc.	Default Value
1401 _h 00 _h	number of entries	UINT8	ro	02 _h
1401 _h 01 _h	COB-ID used by PDO	UINT32	rw	C0000301 _h
1401 _h 02 _h	transmission type	UINT8	rw	FF _h
1601 _h 00 _h	number of mapped objects	UINT8	rw	02 _h
1601 _h 01 _h	first mapped object	UINT32	rw	60400010 _h
1601 _h 02 _h	second mapped object	UINT32	rw	60600008 _h
1601 _h 03 _h	third mapped object	UINT32	rw	00000000 _h
1601 _h 04 _h	fourth mapped object	UINT32	rw	00000000 _h

3. Receive-PDO

Index	Comment	Type	Acc.	Default Value
1402 _h _00 _h	number of entries	UINT8	ro	02 _h
1402 _h _01 _h	COB-ID used by PDO	UINT32	rw	C0000401 _h
1402 _h _02 _h	transmission type	UINT8	rw	FF _h
1602 _h _00 _h	number of mapped objects	UINT8	rw	02 _h
1602 _h _01 _h	first mapped object	UINT32	rw	60400010 _h
1602 _h _02 _h	second mapped object	UINT32	rw	607A0020 _h
1602 _h _03 _h	third mapped object	UINT32	rw	00000000 _h
1602 _h _04 _h	fourth mapped object	UINT32	rw	00000000 _h

4. Receive-PDO

Index	Comment	Type	Acc.	Default Value
1403 _h _00 _h	number of entries	UINT8	ro	02 _h
1403 _h _01 _h	COB-ID used by PDO	UINT32	rw	C0000501 _h
1403 _h _02 _h	transmission type	UINT8	rw	FF _h
1603 _h _00 _h	number of mapped objects	UINT8	rw	02 _h
1603 _h _01 _h	first mapped object	UINT32	rw	60400010 _h
1603 _h _02 _h	second mapped object	UINT32	rw	60FF0020 _h
1603 _h _03 _h	third mapped object	UINT32	rw	00000000 _h
1603 _h _04 _h	fourth mapped object	UINT32	rw	00000000 _h

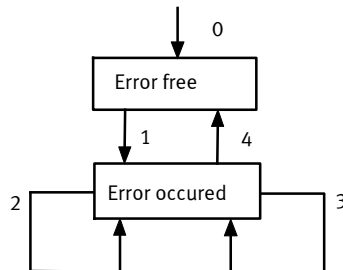
3.3.3 Aktivierung der PDOs

Damit der Motorcontroller PDOs sendet oder empfängt müssen folgende Punkte erfüllt sein:

- Das Objekt `number_of_mapped_objects` muss ungleich Null sein.
- Im Objekt `cob_id_used_for_pdos` muss das Bit 31 gelöscht sein.
- Der Kommunikationsstatus des Motorcontrollers muss operational sein (➔ Kapitel 3.6, Netzwerkmanagement: NMT-Service)

Damit PDOs parametrierbar werden können, müssen folgende Punkte erfüllt sein:

- Der Kommunikationsstatus des Motorcontrollers darf nicht operational sein.



Nach einem Reset befindet sich der Regler im Zustand Error free (den er ggf. sofort wieder verlässt, weil von Anfang an ein Fehler vorhanden ist). Folgende Zustandsübergänge sind möglich:

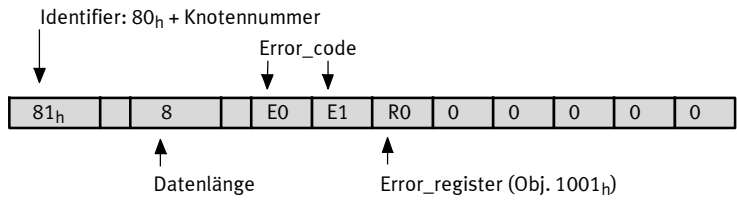
Nr.	Ursache	Bedeutung
0	Initialisierung abgeschlossen	
1	Fehler tritt auf	Es lag kein Fehler vor und ein Fehler tritt auf. Ein EMERGENCY-Telegramm mit dem Fehlercode des aufgetretenen Fehlers wird gesendet.
2	Fehlerquittierung	Eine Fehlerquittierung (→ Kap. 6.1.5) wird versucht, aber nicht alle Ursachen sind behoben.
3	Fehler tritt auf	Es liegt schon ein Fehler vor und ein weiterer Fehler tritt auf. Ein EMERGENCY-Telegramm mit dem Fehlercode des neuen Fehlers wird gesendet.
4	Fehlerquittierung	Eine Fehlerquittierung wird versucht und alle Ursachen sind behoben. Es wird ein EMERGENCY-Telegramm mit dem Fehlercode 0000 gesendet.

Tab. 3.8 Mögliche Zustandsübergänge

3.5.2 Aufbau der EMERGENCY-Message

Der Motorcontroller sendet beim Auftreten eines Fehlers eine EMERGENCY-Message. Der Identifier dieser Nachricht wird aus dem Identifier 80_{h} und der Knotennummer des betroffenen Motorcontrollers zusammengesetzt.

Die EMERGENCY-Message besteht aus acht Datenbytes, wobei in den ersten beiden Bytes ein `error_code` steht, die in folgender Tabelle aufgeführt sind. Im dritten Byte steht ein weiterer Fehlercode (Objekt 1001_{h}). Die restlichen fünf Bytes enthalten Nullen.



error_register (R0)		
Bit	M/O ¹⁾	Bedeutung
0	M	generic error: Fehler liegt an (Oder-Verknüpfung der Bits 1 ... 7)
1	O	current: I ² t-Fehler
2	O	voltage: Spannungsüberwachungsfehler
3	O	temperature: Übertemperatur Motor
4	O	communication error: (overrun, error state)
5	O	–
6	O	reserviert, fix = 0
7	O	reserviert, fix = 0
Werte: 0 = kein Fehler; 1 = Fehler liegt an		

1) M = erforderlich / O = optional

Tab. 3.9 Bitbelegung error_register

Die Fehlercodes sowie Ursache und Maßnahmen finden Sie im Kapitel B „Diagnosemeldungen“.

3.5.3 Beschreibung der Objekte

Objekt 1003_h: pre_defined_error_field

Der jeweilige error_code der Fehlermeldungen wird zusätzlich in einem vierstufigen Fehlerspeicher abgelegt. Dieser ist wie ein Schieberegister strukturiert, so dass immer der zuletzt aufgetretene Fehler im Objekt 1003_h·01_h (standard_error_field_0) abgelegt ist. Durch einen Lesezugriff auf das Objekt 1003_h·00_h (pre_defined_error_field_0) kann festgestellt werden, wie viele Fehlermeldungen zur Zeit im Fehlerspeicher abgelegt sind. Der Fehlerspeicher wird durch das Einschreiben des Wertes 00_h in das Objekt 1003_h·00_h (pre_defined_error_field_0) gelöscht. Um nach einem Fehler die Endstufe des Motorcontrollers wieder aktivieren zu können, muss zusätzlich eine Fehlerquittierung → Kapitel 6.1: Zustandsdiagramm (State Machine) durchgeführt werden.

Index	1003 _h
Name	pre_defined_error_field
Object Code	ARRAY
No. of Elements	4
Data Type	UINT32

Sub-Index	01_h
Description	standard_error_field_0
Access	ro
PDO Mapping	no
Units	–
Value Range	–
Default Value	–

Sub-Index	02_h
Description	standard_error_field_1
Access	ro
PDO Mapping	no
Units	–
Value Range	–
Default Value	–

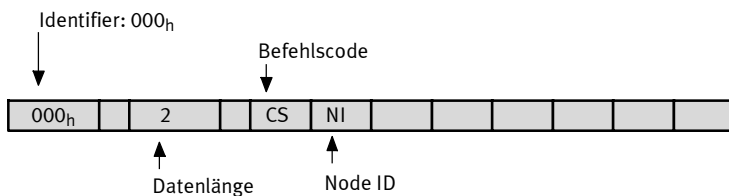
Sub-Index	03_h
Description	standard_error_field_2
Access	ro
PDO Mapping	no
Units	–
Value Range	–
Default Value	–

Sub-Index	04_h
Description	standard_error_field_3
Access	ro
PDO Mapping	no
Units	–
Value Range	–
Default Value	–

3.6 Netzwerkmanagement (NMT-Service)

Alle CANopen-Geräte können über das Netzwerkmanagement angesteuert werden. Hierfür ist der Identifier mit der höchsten Priorität (000_h) reserviert. Mittels NMT können Befehle an einen oder alle Regler gesendet werden. Jeder Befehl besteht aus zwei Bytes, wobei das erste Byte den Befehlscode (command specifier, CS und das zweite Byte die Knotenadresse (node id, NI) des angesprochenen Reglers beinhaltet. Über die Knotenadresse Null können gleichzeitig alle im Netzwerk befindlichen Knoten angesprochen werden. Es ist somit möglich, dass z. B. in allen Geräten gleichzeitig ein Reset ausgelöst wird. Die Regler quittieren die NMT-Befehle nicht. Es kann nur indirekt (z. B. durch die Einschaltmeldung nach einem Reset) auf die erfolgreiche Durchführung geschlossen werden.

Aufbau der NMT-Nachricht:



Für den NMT-Status des CANopen-Knotens sind Zustände in einem Zustandsdiagramm festgelegt. Über das Byte CS in der NMT-Nachricht können Zustandsänderungen ausgelöst werden. Diese sind im Wesentlichen am Ziel-Zustand orientiert.

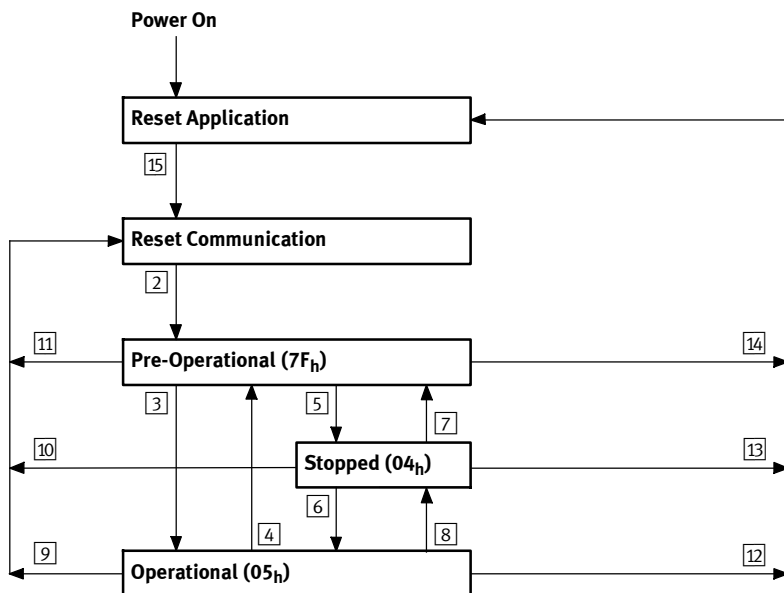


Fig. 3.2 Zustandsdiagramm

Übergang	Bedeutung	CS	Ziel-Zustand	
2	Bootup	--	Pre-Operational	7F _h
3	Start Remote Node	01 _h	Operational	05 _h
4	Enter Pre-Operational	80 _h	Pre-Operational	7F _h
5	Stop Remote Node	02 _h	Stopped	04 _h
6	Start Remote Node	01 _h	Operational	05 _h
7	Enter Pre-Operational	80 _h	Pre-Operational	7F _h
8	Stop Remote Node	02 _h	Stopped	04 _h
9	Reset Communication	82 _h	Reset Communication ¹⁾	
10	Reset Communication	82 _h	Reset Communication ¹⁾	
11	Reset Communication	82 _h	Reset Communication ¹⁾	
12	Reset Application	81 _h	Reset Application ¹⁾	
13	Reset Application	81 _h	Reset Application ¹⁾	
14	Reset Application	81 _h	Reset Application ¹⁾	

1) Endgültiger Zielzustand ist Pre-Operational (7F_h), da die Übergänge 15 und 2 vom Regler automatisch durchgeführt werden.

Tab. 3.10 NMT-State machine

Alle anderen Zustands-Übergänge werden vom Regler selbsttätig ausgeführt, z. B. weil die Initialisierung abgeschlossen ist.

Im Parameter NI muss die Knotennummer des Reglers angegeben werden oder Null, wenn alle im Netzwerk befindlichen Knoten adressiert werden sollen (Broadcast). Je nach NMT-Status können bestimmte Kommunikationsobjekte nicht benutzt werden: So ist es z. B. unbedingt notwendig den NMT-Status auf Operational zu stellen, damit der Regler PDOs sendet.

Name	Bedeutung	SDO	PDO	NMT
Reset Application	Keine Kommunikation. Alle CAN-Objekte werden auf ihre Resetwerte (Applikations-Parametersatz) zurückgesetzt	–	–	–
Reset Communication	Keine Kommunikation Der CAN-Controller wird neu initialisiert.	–	–	–
Initialising	Zustand nach Hardware-Reset. Zurücksetzen des CAN-Knotens, Senden der Bootup-Message	–	–	–
Pre-Operational	Kommunikation über SDOs möglich PDOs nicht aktiv (Kein Senden/Auswerten)	X	–	X
Operational	Kommunikation über SDOs möglich Alle PDOs aktiv (Senden/Auswerten)	X	X	X
Stopped	Keine Kommunikation außer Heartbeating	–	–	X

Tab. 3.11 NMT-State machine



NMT-Telegramme dürfen nicht in einem Burst (unmittelbar hintereinander) gesendet werden!

Zwischen zwei aufeinanderfolgenden NMT-Nachrichten auf dem Bus (auch für verschiedene Knoten!) muss mindestens die doppelte Lagereglerzykluszeit liegen, damit der Regler die NMT-Nachrichten korrekt verarbeitet.



Der NMT Befehl „Reset Application“ wird gegebenenfalls so lange verzögert, bis ein laufender Speichervorgang abgeschlossen ist, da ansonsten der Speichervorgang unvollständig bleiben würde (Defekter Parametersatz).

Die Verzögerung kann im Bereich einiger Sekunden liegen.



Der Kommunikationsstatus muss auf operational eingestellt werden, damit der Regler PDOs sendet und empfängt.

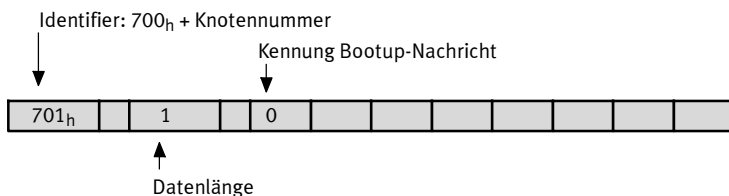
3.7 Bootup

3.7.1 Übersicht

Nach dem Einschalten der Spannungsversorgung oder nach einem Reset, meldet der Regler über eine Bootup-Nachricht, dass die Initialisierungsphase beendet ist. Der Regler ist dann im NMT-Status pre-operational (→ Kapitel 3.6, Netzwerkmanagement (NMT-Service))

3.7.2 Aufbau der Bootup-Nachricht

Die Bootup-Nachricht ist nahezu identisch zur folgenden Heartbeat-Nachricht aufgebaut. Lediglich wird statt des NMT-Status eine Null gesendet.



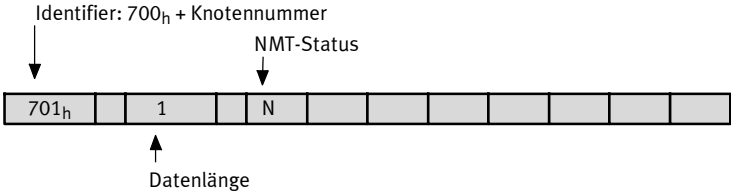
3.8 **Heartbeat (Error Control Protocol)**

3.8.1 **Übersicht**

Zur Überwachung der Kommunikation zwischen Slave (Antrieb) und Master kann das sogenannte Heartbeat-Protokoll aktiviert werden: Hierbei sendet der Antrieb zyklisch Nachrichten an den Master. Der Master kann das zyklische Auftreten dieser Nachrichten überprüfen und entsprechende Maßnahmen einleiten, wenn diese ausbleiben. Da sowohl Heartbeat- als auch Nodeguarding-Telegramme (→ Kap. 3.9) mit dem Identifier 700_h + Knotennummer gesendet werden, können nicht beide Protokolle gleichzeitig aktiv sein. Werden beide Protokolle gleichzeitig aktiviert, ist nur das Heartbeat-Protokoll aktiv.

3.8.2 **Aufbau der Heartbeat-Nachricht**

Das Heartbeat-Telegramm wird mit dem Identifier 700_h + Knotennummer gesendet. Es enthält nur 1 Byte Nutzdaten, den NMT-Status des Reglers (→ Kapitel 3.6, Netzwerkmanagement (NMT-Service)).



N	Bedeutung
04 _h	Stopped
05 _h	Operational
7F _h	Pre-Operational

3.8.3 **Beschreibung der Objekte**

Objekt 1017_h: producer_heartbeat_time

Zur Aktivierung der Heartbeat-Funktionalität kann die Zeit zwischen zwei Heartbeat-Telegrammen über das Object producer_heartbeat_time festgelegt werden.

Index	1017 _h
Name	producer_heartbeat_time
Object Code	VAR
Data Type	UINT16

Access	rw
PDO	no
Units	ms
Value Range	0 ... 65535
Default Value	0

Die `producer_heartbeat_time` kann im Parametersatz gespeichert werden. Startet der Regler mit einer `producer_heartbeat_time` ungleich Null, gilt die Bootup-Nachricht als erstes Heartbeat. Der Regler kann nur als sog. Heartbeat Producer verwendet werden. Das Objekt 1016_h (`consumer_heartbeat_time`) ist daher nur aus Kompatibilitätsgründen implementiert und liefert immer 0 zurück.

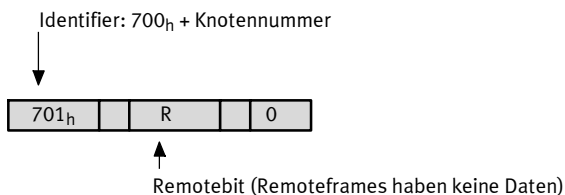
3.9 Nodeguarding (Error Control Protocol)

3.9.1 Übersicht

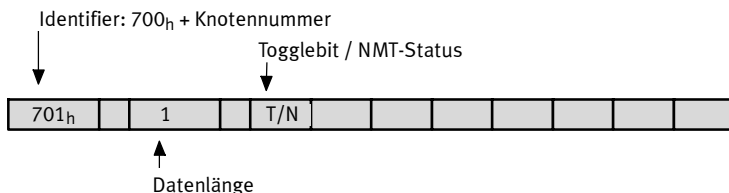
Ebenfalls zur Überwachung der Kommunikation zwischen Slave (Antrieb) und Master kann das sogenannte Nodeguarding-Protokoll verwendet werden. Im Gegensatz zum Heartbeat-Protokoll überwachen sich hierbei Master und Slave gegenseitig: Der Master fragt den Antrieb zyklisch nach seinem NMT-Status. Dabei wird in jeder Antwort des Reglers ein bestimmtes Bit invertiert (getoggelt). Bleiben diese Antworten aus oder antwortet der Regler immer mit dem gleichen Togglebit kann der Master entsprechend reagieren. Ebenso überwacht der Antrieb das regelmäßige Eintreffen der Nodeguarding-Anfragen des Masters: Bleiben die Nachrichten über einen bestimmten Zeitraum aus, löst der Regler Fehler 12-4 aus. Da sowohl Heartbeat- als auch Nodeguarding-Telegramme (→ Kapitel 3.8) mit dem Identifier 700_h + Knotennummer gesendet werden, können nicht beide Protokolle gleichzeitig aktiv sein. Werden beide Protokolle gleichzeitig aktiviert, ist nur das Heartbeat-Protokoll aktiv.

3.9.2 Aufbau der Nodeguarding-Nachrichten

Die Anfrage des Masters muss als sog. Remoteframe mit dem Identifier 700_h + Knotennummer gesendet werden. Bei einem Remoteframe ist zusätzlich ein spezielles Bit im Telegramm gesetzt, das Remotebit. Remoteframes haben grundsätzlich keine Daten.



Die Antwort des Reglers ist analog zur Heartbeat-Nachricht aufgebaut. Sie enthält nur 1 Byte Nutzdaten, das Togglebit und den NMT-Status des Reglers (→ Kapitel 3.6).



Das erste Datenbyte (T/N) ist folgendermaßen aufgebaut:

Bit	Wert	Name	Bedeutung
7	80 _h	toggle_bit	Ändert sich mit jedem Telegramm
0 ... 6	7F _h	nmt_state	04 _h Stopped 05 _h Operational 7F _h Pre-Operational

Die Überwachungszeit für Anfragen des Masters ist parametrierbar. Die Überwachung beginnt mit der ersten empfangenen Remoteabfrage des Masters. Ab diesem Zeitpunkt müssen die Remoteabfragen vor Ablauf der eingestellten Überwachungszeit eintreffen, da anderenfalls Fehler 12-4 ausgelöst wird. Das Togglebit wird durch das NMT-Kommando Reset Communication zurückgesetzt. Es ist daher in der ersten Antwort des Reglers gelöscht.

3.9.3 Beschreibung der Objekte

Objekt 100C_h: guard_time

Zur Aktivierung der Nodeguarding-Überwachung wird die Maximalzeit zwischen zwei Remoteabfragen des Masters parametriert. Diese Zeit wird im Regler aus dem Produkt von guard_time (100C_h) und life_time_factor (100D_h) bestimmt. Es empfiehlt sich daher den life_time_factor mit 1 zu beschreiben und die Zeit dann direkt über die guard_time in Millisekunden vorzugeben.

Index	100C _h
Name	guard_time
Object Code	VAR
Data Type	UINT16

Access	rw
PDO Mapping	no
Units	ms
Value Range	0 ... 65535
Default Value	0

3.9.4 Objekt 100D_h: life_time_factor

Der life_time_factor sollte mit 1 beschrieben werden um die guard_time direkt vorzugeben.

Index	100D_h
Name	life_time_factor
Object Code	VAR
Data Type	UINT8

Access	rw
PDO Mapping	no
Units	–
Value Range	0,1
Default Value	0

3.9.5 Tabelle der Identifier

Die folgende Tabelle gibt eine Übersicht über die verwendeten Identifier:

Objekt-Typ	Identifier (hexadezimal)	Bemerkung
SDO (Host an Controller)	600 _h + Knotennummer	
SDO (Controller an Host)	580 _h + Knotennummer	
TPDO1	180 _h	Standardwerte. Können bei Bedarf geändert werden.
TPDO2	280 _h	
TPDO3	380 _h	
TPDO4	480 _h	
RPDO1	200 _h	
RPDO2	300 _h	
RPDO3	400 _h	
RPDO4	500 _h	
SYNC	080 _h	
EMCY	080 _h + Knotennummer	
HEARTBEAT	700 _h + Knotennummer	
NODEGUARDING	700 _h + Knotennummer	
BOOTUP	700 _h + Knotennummer	
NMT	000 _h	

4 EtherCAT mit CoE

M3

Dieser Abschnitt gilt nur für die Motorcontroller CMMP-AS-...-M3.

4.1 Überblick

Dieser Teil der Dokumentation beschreibt den Anschluss und Konfiguration der Motorcontroller CMMP-AS-...-M3 in einem EtherCAT-Netzwerk. Sie richtet sich an Personen, die bereits mit der Serie der Motorcontroller und CANopen CiA 402 vertraut sind.

Das Feldbussystem EtherCAT bedeutet "Ethernet for Controller and Automation Technology" und wird von der internationalen Organisation EtherCAT Technology Group (ETG) betreut und unterstützt und ist als offene Technologie konzeptioniert, die durch die International Electrotechnical Commission (IEC) genormt ist. EtherCAT ist ein auf Ethernet basierendes Feldbussystem und ist dank hoher Geschwindigkeit, flexibler Topologie (Linie, Baum, Stern) und einfacher Konfiguration wie ein Feldbus zu handhaben. Das EtherCAT-Protokoll wird mit einem speziellen genormten Ethernettyp direkt im Ethernet-Frame gemäß IEEE802.3 transportiert. Broadcast, Multicast und Querkommunikation zwischen den Slaves sind möglich. Beim EtherCAT basiert der Datenaustausch auf einer reinen Hardware-Maschine.

Abkürzung	Bedeutung
ESC	EtherCAT Slave Controller
PDI	Process Data Interface
CoE	CANopen-over-EtherCAT-Protokoll

Tab. 4.1 EtherCAT-spezifische Abkürzungen

4.2 EtherCat-Interface CAMC-EC

Das EtherCAT-Interface CAMC-EC erlaubt die Anbindung des Motorcontrollers CMMP-AS-...-M3 an das Feldbussystem EtherCAT. Die Kommunikation über das EtherCAT-Interface (IEEE 802.3u) erfolgt mit einer EtherCAT-Standard-Verkabelung und ist zwischen dem CMMP-AS-...-M3 ab Revision 01 und der Parametriersoftware FCT ab der Version 2.0 möglich.



Festo unterstützt beim CMMP-AS-...-M3 das CoE-Protokoll (CANopen over EtherCAT).

Kenndaten des EtherCAT-Interface CAMC-EC

Das EtherCAT-Interface besitzt folgende Leistungsmerkmale:

- Mechanisch voll integrierbar in die Motorcontroller der Serie CMMP-AS-...-M3
- EtherCAT entsprechend IEEE-802.3u (100Base-TX) mit 100Mbps (volduplex)
- Stern- und Linientopologie
- Steckverbinder: RJ45
- Potentialgetrennte EtherCAT-Schnittstelle
- Kommunikationszyklus : 1 ms
- Max. 127 Slaves
- EtherCAT-Slave-Implementierung basiert auf dem FPGA ESC20 der Fa. Beckhoff
- Unterstützung des Merkmals "Distributed Clocks" zur zeitlich synchronen Sollwertübernahme
- LED-Anzeigen für Betriebsbereitschaft und Link-Detect

Anschluss- und Anzeigeelemente des EtherCAT-Interface

An der Frontplatte des EtherCAT-Interface sind folgende Elemente angeordnet:

- **LED 1** (Zweifarb-LED) für:
 - EtherCAT-Kommunikation (gelb)
 - "Verbindung aktiv an Port 1" (rot)
 - Run (grün)
- **LED 2** (rot) zur Anzeige "Verbindung aktiv an Port 2"
- zwei RJ45-Buchsen.

Die folgende Bild zeigt die Lage der Buchsen und deren Nummerierung:

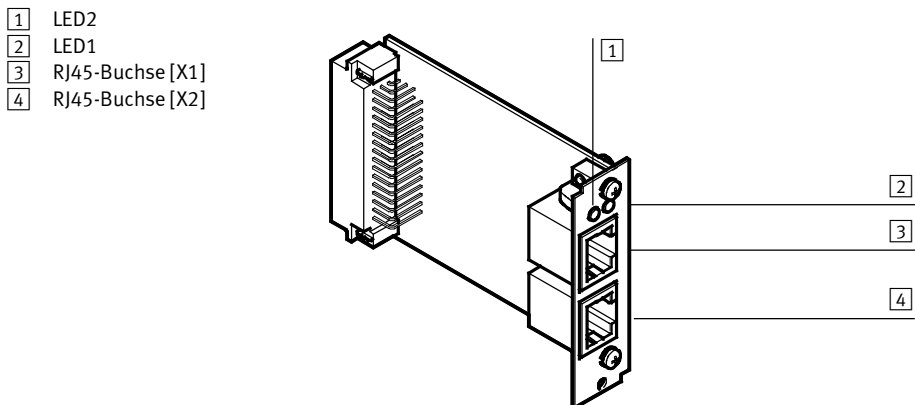



Fig. 4.1 Anschluss- und Anzeigeelemente am EtherCAT-Interface



Das EtherCAT-Interface kann nur im Optionsschacht Ext2 betrieben werden. Der Betrieb weiterer Interfacemodule in dem Optionsschacht Ext1 ist, außer mit CAMC-D-8E8A Interface, dann nicht mehr möglich.

4.3
Einbau des EtherCAT-Interface in den Controller



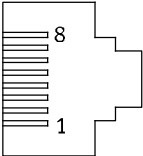
Hinweis
 Beachten Sie vor Montage- und Installationsarbeiten die Sicherheitshinweise in der Beschreibung Hardware GDPC-CMMP-M3-HW-...

Mit einem geeigneten Kreuzschlitz-Schraubendreher wird das Frontblech über dem Einschubschacht Ext2 des Motorcontrollers CMMP-AS-...-M3 abgeschraubt. Das EtherCAT-Interface wird jetzt in den offenen Einschubschacht (Ext2) so eingesteckt, dass die Platine in den seitlichen Führungen des Einschubschachtes läuft. Das Interface wird bis zum Anschlag eingeschoben. Abschließend wird das Interface mit der Kreuzschlitzschraube am Motorcontrollergehäuse angeschraubt.

4.4
Steckerbelegung und Kabelspezifikationen

RJ45-Buchsen	Funktion
[X1] (RJ45-Buchse oben)	Uplink zum Master oder einem vorherigen Teilnehmer einer linienförmigen Verbindung (z. B. mehrere Motorcontroller)
[X2] (RJ45-Buchse unten)	Uplink zum Master, Ende einer linienförmigen Verbindung oder Anschluss weiterer nachgeordneter Teilnehmer

Tab. 4.2 Ausführung der Steckverbinder [X1] und [X2]

	Pin	Spezifikation	
	1	Empfängersignal– (RX–)	Adernpaar 3
	2	Empfängersignal+ (RX+)	Adernpaar 3
	3	Sendesignal– (TX–)	Adernpaar 2
	4	–	Adernpaar 1
	5	–	Adernpaar 1
	6	Sendesignal+ (TX+)	Adernpaar 2
	7	–	Adernpaar 4
	8	–	Adernpaar 4

Tab. 4.3 Belegung der Steckverbinder [X1] und [X2]

Wert	Funktion
EtherCAT-Interface, Signalpegel	0 ... 2,5 VDC
EtherCAT-Interface, Differenzspannung	1,9 ... 2,1 VDC

Tab. 4.4 Spezifikation EthetCAT-Interface

Art und Ausführung des Kabels

Die Verkabelung erfolgt mit geschirmten Twisted-Pair-Kabeln STP, Cat.5. Es werden Stern- und Linien-Topologien unterstützt. Der Netzaufbau muss entsprechend der 5-4-3-Regel erfolgen. Es dürfen maxi-

mal 10 Hubs in Linie verkabelt werden. Das EtherCAT-Interface enthält einen Hub. Die Gesamtkabel-
länge ist auf 100 m begrenzt.



Fehler durch ungeeignete Bus-Kabel

Aufgrund der sehr hohen möglichen Übertragungsraten empfehlen wir ausschließlich die Verwendung von standardisierten Kabeln und Steckverbindern die mindestens der Kategorie 5 (CAT5) nach EN 50173 bzw. ISO/IEC 11801 entsprechen.
Folgen Sie beim Aufbau des EtherCAT-Netzes unbedingt den Ratschlägen der gängigen Literatur bzw. den nachfolgenden Informationen und Hinweisen, um ein stabiles, störungsfreies System zu erhalten. Bei einer nicht sachgemäßen Verkabelung können während des Betriebs Störungen auf dem EtherCAT-Bus auftreten, die dazu führen, dass der Motorcontroller CMMP-AS-...-M3 aus Sicherheitsgründen mit einem Fehler abschaltet.

Bus-Terminierung

Es werden keine externen Buserminierungen benötigt. Das EtherCAT-Technologiemodul überwacht seine beiden Ports und schließt den Bus selbständig ab (Loop-back-Funktion).

4.5 CANopen-Kommunikationsschnittstelle

Die Anwenderprotokolle werden über EtherCAT getunnelt. Für das vom CMMP-AS-...-M3 unterstützte CANopen-over-EtherCAT-Protokoll (CoE) werden für die Kommunikationsschicht die meisten Objekte nach CiA 301 von EtherCAT unterstützt. Hier handelt es sich weitestgehend um Objekte zur Einrichtung der Kommunikation zwischen Master und Slave.

Für das CANopen-Motion-Profil nach CiA 402 werden die meisten Objekte unterstützt, die auch über den normalen CANopen-Feldbus bedient werden können. Grundsätzlich werden folgende Dienste und Objektgruppen von der EtherCAT-CoE-Implementation im Motorcontroller CMMP-AS-...-M3 unterstützt:

Dienste/Objektgruppen		Funktion
SDO	Service Data Object	Werden zur normalen Parametrierung des Motorcontrollers verwendet.
PDO	Process Data Object	Schneller Austausch von Prozessdaten (z. B. Istzahl) möglich.
EMCY	Emergency Message	Übermittlung von Fehlermeldungen.

Tab. 4.5 Unterstützte Dienste und Objektgruppen

Dabei werden die einzelnen Objekte, die über das CoE-Protokoll im Motorcontroller CMMP-AS-...-M3 angesprochen werden können, intern an die bestehende CANopen-Implementierung weitergereicht und dort verarbeitet.

Allerdings wurden unter der CoE-Implementierung unter EtherCAT einige neue CANopen-Objekte hinzugefügt, die für die spezielle Anbindung über CoE notwendig sind. Dieses resultiert aus der geänderten Kommunikationsschnittstelle zwischen dem EtherCAT-Protokoll und dem CANopen-Protokoll. Dort wird ein sogenannter Sync-Manager eingesetzt, um die Übertragung von PDOs und SDOs über die beiden EtherCAT-Transferarten (Mailbox- und Prozessdatenprotokoll) zu steuern.

Dieser Sync Manager und die notwendigen Konfigurationsschritte für den Betrieb des CMMP-AS-...-M3 unter EtherCAT-CoE sind in Kapitel 4.5.1 „Konfiguration der Kommunikationsschnittstelle“ beschrieben. Die zusätzlichen Objekte sind in Kapitel 4.5.2 „Neue und geänderte Objekte unter CoE“ beschrieben. Außerdem werden einige CANopen-Objekte des CMMP-AS-...-M3, die unter einer normalen CANopen-Anbindung verfügbar sind, über eine CoE-Anbindung über EtherCAT nicht unterstützt. Eine Liste der unter CoE nicht unterstützten CANopen-Objekte ist in Kapitel 4.5.3 „Nicht unterstützte Objekte unter CoE“ gegeben.

4.5.1 Konfiguration der Kommunikationsschnittstelle

Wie im vorherigen Kapitel beschrieben, benutzt das EtherCAT-Protokoll zwei verschiedene Transferarten zur Übertragung der Geräte- und Anwenderprotokolle, wie z. B. das vom CMMP-AS-...-M3 verwendete CANopen-over-EtherCAT-Protokoll (CoE). Diese beiden Transferarten sind das Mailbox-Telegrammprotokoll für azyklische Daten und das Prozessdaten-Telegrammprotokoll für die Übertragung von zyklischen Daten.

Für das CoE-Protokoll werden diese beiden Transferarten für die verschiedenen CANopen-Transferarten verwendet. Dabei werden sie wie folgt benutzt:

Telegrammprotokoll	Beschreibung	Verweis
Mailbox	Diese Transferart dient der Übertragung der unter CANopen definierten Service Data Objects (SDOs). Sie werden in EtherCAT in SDO-Frames übertragen.	➔ Kapitel 4.7 "SDO-Frame"
Prozessdaten	Diese Transferart dient der Übertragung der unter CANopen definierten Process Data Objects (PDOs), die zum Austausch von zyklischen Daten benutzt werden. Sie werden in EtherCAT in PDO-Frames übertragen.	➔ Kapitel 4.8 "PDO-Frame"

Tab. 4.6 Telegrammprotokoll – Beschreibung

Grundsätzlich können über diese beiden Transferarten alle PDOs und SDOs genau so benutzt werden, wie sie für das CANopen-Protokoll für den CMMP-AS-...-M3 definiert sind. Allerdings unterscheidet sich die Parametrierung der PDOs und SDOs zum Versenden der Objekte über EtherCAT von den Einstellungen, die unter CANopen gemacht werden müssen. Um die CANopen-Objekte, die über PDO- oder SDO-Transfers zwischen Master und Slave ausgetauscht werden sollen, in das EtherCAT-Protokoll einzubinden, ist unter EtherCAT ein sogenannter Sync-Manager implementiert. Dieser Sync Manager dient dazu, die Daten der zu sendenden PDOs und SDOs in die EtherCAT-Telegramme einzubinden. Zu diesem Zweck stellt der Sync-Manager mehrere Sync-Kanäle zur Verfügung, die jeweils einen CANopen-Datenkanal (Receive SDO, Transmit SDO, Receive PDO oder Transmit PDO) auf das EtherCAT-Telegramm umsetzen können. Das Bild soll die Einbindung des Sync-Managers in das System veranschaulichen:

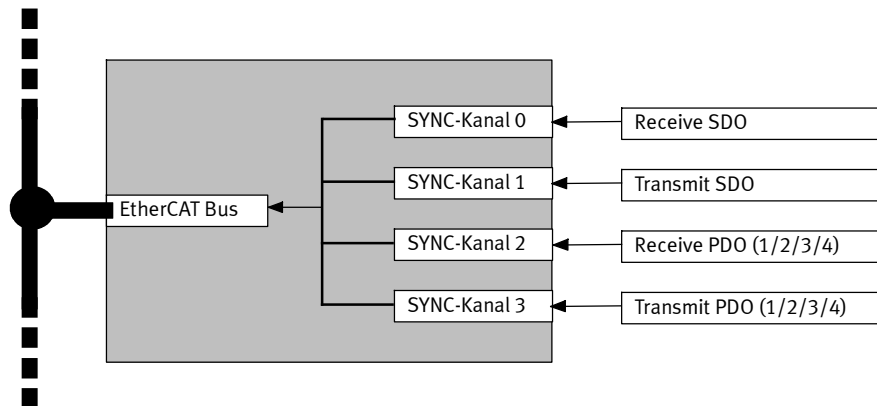


Fig. 4.2 Beispielmapping der SDOs und PDOs auf die Sync-Kanäle

Alle Objekte werden über so genannte Sync-Kanäle verschickt. Die Daten dieser Kanäle werden automatisch in den EtherCAT-Datenstrom eingebunden und übertragen. Die EtherCAT-Implementierung im Motorcontroller CMMP-AS-...-M3 unterstützt vier solcher Sync-Kanäle.

Aus diesem Grund ist gegenüber CANopen ein zusätzliches Mapping der SDOs und PDOs auf die Sync-Kanäle notwendig. Dieses geschieht über die so genannten Sync-Manager-Objekte (Objekte 1C00_h und 1C10_h ... 1C13_h → Kapitel 4.5.2). Diese Objekte sind nachfolgend näher beschrieben.

Die Zuordnung dieser Sync-Kanäle zu den einzelnen Transferarten ist fest und kann vom Anwender nicht geändert werden. Die Belegung ist wie folgt:

- Sync-Kanal 0: Mailbox-Telegrammprotokoll für eingehende SDOs (Master ⇒ Slave)
- Sync-Kanal 1: Mailbox-Telegrammprotokoll für ausgehende SDOs (Master ⇐ Slave)
- Sync-Kanal 2: Prozessdaten-Telegrammprotokoll für eingehende PDOs (Master ⇒ Slave).
Hier ist das Objekt 1C12_h zu beachten.
- Sync-Kanal 3: Prozessdaten-Telegrammprotokoll für ausgehende PDOs (Master ⇐ Slave).
Hier ist das Objekt 1C13_h zu beachten.

Die Parametrierung der einzelnen PDOs wird über die Objekte 1600_h bis 1603_h (Receive PDOs) und 1A00_h bis 1A03_h (Transmit PDOs) eingestellt. Die Parametrierung der PDOs wird dabei wie im Kapitel 3 „Zugriffsverfahren CANopen“ beschrieben durchgeführt.

Grundsätzlich kann die Einstellung der Sync-Kanäle und die Konfiguration der PDOs nur im Zustand „Pre-Operational“ durchgeführt werden.



Unter EtherCAT ist es nicht vorgesehen, die Parametrierung des Slave selbst durchzuführen. Zu diesem Zweck stehen die Gerätebeschreibungsdateien zur Verfügung. In ihnen ist die gesamte Parametrierung, inklusive der PDO Parametrierung vorgegeben und wird vom Master während der Initialisierung so verwendet.

Sämtliche Änderungen der Parametrierung sollten daher nicht per Hand, sondern in den Gerätebeschreibungsdateien erfolgen. Zu diesem Zweck sind die für den Anwender wichtigen Sektionen der Gerätebeschreibungsdateien in Abschnitt 4.11 näher beschrieben.



Die hier beschriebenen Sync-Kanäle entsprechen NICHT den von CANopen bekannten Sync-Telegrammen. CANopen-Sync-Telegramme können weiterhin als SDOs über die unter CoE implementierte SDO-Schnittstelle übertragen werden, beeinflussen aber nicht direkt die oben beschriebenen Sync-Kanäle.

4.5.2 Neue und geänderte Objekte unter CoE

Die folgende Tabelle gibt eine Übersicht über die verwendeten Indizes und Subindizes für die CANopen-kompatiblen Kommunikationsobjekte, die für das Feldbussystem EtherCAT im Bereich von 1000_h bis 1FFF_h eingefügt wurden. Diese ersetzen hauptsächlich die Kommunikationsparameter nach CiA 301.

Objekt	Bedeutung	Erlaubt bei
1000 _h	Device Type	Identifiziert die Gerätesteuerung
1018 _h	Identity Object	Vendor-ID, Product-Code, Revision, Seriennummer
1100 _h	EtherCAT fixed station address	Feste Adresse, die dem Slave während der Initialisierung durch den Master zugewiesen wird
1600 _h	1. RxPDO Mapping	Identifiziert das 1. Receive-PDO
1601 _h	2. RxPDO Mapping	Identifiziert das 2. Receive-PDO
1602 _h	3. RxPDO Mapping	Identifiziert das 3. Receive-PDO
1603 _h	4. RxPDO Mapping	Identifiziert das 4. Receive-PDO
1A00 _h	1. TxPDO Mapping	Identifiziert das 1. Transmit-PDO
1A01 _h	2. TxPDO Mapping	Identifiziert das 2. Transmit-PDO
1A02 _h	3. TxPDO Mapping	Identifiziert das 3. Transmit-PDO
1A03 _h	4. TxPDO Mapping	Identifiziert das 4. Transmit-PDO
1C00 _h	Sync Manager Communication Type	Objekt zur Konfiguration der einzelnen Sync-Kanäle (SDO oder PDO Transfer)
1C10 _h	Sync Manager PDO Mapping for Syncchannel 0	Zuordnung des Sync-Kanal 0 zu einem PDO/SDO (Kanal 0 ist immer reserviert für den Mailbox Receive SDO Transfer)
1C11 _h	Sync Manager PDO Mapping for Syncchannel 1	Zuordnung des Sync-Kanal 1 zu einem PDO/SDO (Kanal 1 ist immer reserviert für den Mailbox Send SDO Transfer)

Objekt	Bedeutung	Erlaubt bei
1C12 _h	Sync Manager PDO Mapping for Syncchannel 2	Zuordnung des Sync-Kanal 2 zu einem PDO (Kanal 2 ist reserviert für Receive PDOs)
1C13 _h	Sync Manager PDO Mapping for Syncchannel 3	Zuordnung des Sync-Kanal 3 zu einem PDO (Kanal 3 ist reserviert für Transmit PDOs)

Tab. 4.7 Neue und geänderte Kommunikationsobjekte

In den nachfolgenden Kapitel werden die Objekte 1C00_h und 1C10_h ... 1C13_h genauer beschrieben, da sie nur unter dem EtherCAT-CoE-Protokoll definiert und implementiert sind und daher im CANopen-Handbuch für den Motorcontroller CMMP-AS-...-M3 nicht dokumentiert sind.



Der Motorcontroller CMMP-AS-...-M3 mit dem EtherCAT-Interface unterstützt vier Receive-PDOs (RxPDO) und vier Transmit-PDOs (TxPDO).

Die Objekte 1008_h, 1009_h und 100A_h werden vom CMMP-AS-...-M3 nicht unterstützt, da keine Klartext-Strings aus dem Motorcontroller gelesen werden können.

Objekt 1100_h - EtherCAT fixed station address

Über dieses Objekt wird dem Slave während der Initialisierungsphase eine eindeutige Adresse zugewiesen. Das Objekt hat die folgende Bedeutung:

Index	1100_h
Name	EtherCAT fixed station address
Object Code	Var
Data Type	uint16
Access	ro
PDO Mapping	no
Value Range	0 ... FFFF _h
Default Value	0

Objekt 1C00_h - Sync Manager Communication Type

Über dieses Objekt kann die Transferart für die verschiedenen Kanäle des EtherCAT-Sync-Managers ausgelesen werden. Da der CMMP-AS-...-M3 unter dem EtherCAT-CoE-Protokoll nur die ersten vier Sync-Kanäle unterstützt, sind die folgenden Objekte nur lesbar (vom Typ "read only").

Dadurch ist die Konfiguration des Sync-Managers für den CMMP-AS-...-M3 fest konfiguriert. Die Objekte haben die folgende Bedeutung:

Index	1C00_h
Name	Sync Manager Communication Type
Object Code	Array
Data Type	uint8

Sub-Index	00_h
Description	Number of used Sync Manager Channels
Access	ro
PDO Mapping	no
Value Range	4
Default Value	4

Sub-Index	01_h
Description	Communication Type Sync Channel 0
Access	ro
PDO Mapping	no
Value Range	2: Mailbox Transmit (Master => Slave)
Default Value	2: Mailbox Transmit (Master => Slave)

Sub-Index	02_h
Description	Communication Type Sync Channel 1
Access	ro
PDO Mapping	no
Value Range	2: Mailbox Transmit (Master <= Slave)
Default Value	2: Mailbox Transmit (Master <= Slave)

Index	03_h
Description	Communication Type Sync Channel 2
Access	ro
PDO Mapping	no
Value Range	0: unused 3: Process Data Output (RxPDO / Master => Slave)
Default Value	3

Sub-Index	04_h
Description	Communication Type Sync Channel 3
Access	ro
PDO Mapping	no
Value Range	0: unused 4: Process Data Input (TxPDO/Master <= Slave)
Default Value	4

Objekt 1C10_h - Sync Manager Channel 0 (Mailbox Receive)

Über dieses Objekt kann ein PDO für den Sync-Kanal 0 konfiguriert werden. Da der Sync-Kanal 0 immer durch das Mailbox-Telegrammprotokoll belegt ist, kann dieses Objekt vom Anwender nicht geändert werden. Das Objekt hat daher immer die folgenden Werte:

Index	1C10_h
Name	Sync Manager Channel 0 (Mailbox Receive)
Object Code	Array
Data Type	uint8

Sub-Index	00_h
Description	Number of assigned PDOs
Access	ro
PDO Mapping	no
Value Range	0 (no PDO assigned to this channel)
Default Value	0 (no PDO assigned to this channel)



Der durch die EtherCAT-Spezifikation für den Subindex 0 dieser Objekte festgelegte Name "Number of assigned PDOs" ist hier irreführend, da die Sync-Manager-Kanäle 0 und 1 immer durch das Mailbox-Telegramm belegt sind. In dieser Telegrammart werden unter EtherCAT-CoE immer SDOs übertragen. Der Subindex 0 dieser beiden Objekte bleibt also unbenutzt.

Objekt 1C11_h - Sync Manager Channel 1 (Mailbox Send)

Über dieses Objekt kann ein PDO für den Sync-Kanal 1 konfiguriert werden. Da der Sync-Kanal 1 immer durch das Mailbox-Telegrammprotokoll belegt ist, kann dieses Objekt vom Anwender nicht geändert werden. Das Objekt hat daher immer die folgenden Werte:

Index	1C11_h
Name	Sync Manager Channel 1 (Mailbox Send)
Object Code	Array
Data Type	uint8

Sub-Index	00_h
Description	Number of assigned PDOs
Access	ro
PDO Mapping	no
Value Range	0 (no PDO assigned to this channel)
Default Value	0 (no PDO assigned to this channel)

Objekt 1C12_h - Sync Manager Channel 2 (Process Data Output)

Über dieses Objekt kann ein PDO für den Sync-Kanal 2 konfiguriert werden. Der Sync-Kanal 2 ist fest für den Empfang von Receive-PDOs (Master ⇒ Slave) vorgesehen. In diesem Objekt muss unter dem Sub-index 0 die Anzahl der PDOs eingestellt werden, die diesem Sync-Kanal zugeordnet sind.

In den Subindizes 1 bis 4 wird anschließend die Objektnummer des PDOs eingetragen, das dem Kanal zugeordnet werden soll. Dabei können hier nur die Objektnummern der vorher konfigurierten Receive-PDOs benutzt werden (Objekt 1600_h ... 1603_h).

In der gegenwärtigen Implementierung erfolgt keine weitere Auswertung der Daten der u.a. Objekte durch die Firmware des Motorcontrollers.

Es wird die CANopen-Konfiguration der PDOs für die Auswertung unter EtherCAT herangezogen.

Index	1C12_h
Name	Sync Manager Channel 2 (Process Data Output)
Object Code	Array
Data Type	uint8

Sub-Index	00_h
Description	Number of assigned PDOs
Access	rw
PDO Mapping	no
Value Range	0: no PDO assigned to this channel 1: one PDO assigned to this channel 2: two PDOs assigned to this channel 3: three PDOs assigned to this channel 4: four PDOs assigned to this channel
Default Value	0 :no PDO assigned to this channel

Sub-Index	01_h
Description	PDO Mapping object Number of assigned RxPDO
Access	rw
PDO Mapping	no
Value Range	1600 _h : first Receive PDO
Default Value	1600 _h : first Receive PDO

Sub-Index	02_h
Description	PDO Mapping object Number of assigned RxPDO
Access	rw
PDO Mapping	no
Value Range	1601 _h : second Receive PDO
Default Value	1601 _h : second Receive PDO

Sub-Index	03_h
Description	PDO Mapping object Number of assigned RxPDO
Access	rw
PDO Mapping	no
Value Range	1602 _h : third Receive PDO
Default Value	1602 _h : third Receive PDO

Sub-Index	04_h
Description	PDO Mapping object Number of assigned RxPDO
Access	rw
PDO Mapping	no
Value Range	1603 _h : fourth Receive PDO
Default Value	1603 _h : fourth Receive PDO

Objekt 1C13_h - Sync Manager Channel 3 (Process Data Input)

Über dieses Objekt kann ein PDO für den Sync-Kanal 3 konfiguriert werden. Der Sync-Kanal 3 ist fest für das Senden von Transmit-PDOs (Master <= Slave) vorgesehen. In diesem Objekt muss unter dem Subindex 0 die Anzahl der PDOs eingestellt werden, die diesem Sync-Kanal zugeordnet sind.

In den Subindizes 1 bis 4 wird anschließend die Objektnummer des PDOs eingetragen, das dem Kanal zugeordnet werden soll. Dabei können hier nur die Objektnummern der vorher konfigurierten Transmit-PDOs benutzt werden (1A00_h bis 1A03_h).

Index	1C13_h
Name	Sync Manager Channel 3 (Process Data Input)
Object Code	Array
Data Type	uint8

Sub-Index	00_h
Description	Number of assigned PDOs
Access	rw
PDO Mapping	no
Value Range	0: no PDO assigned to this channel 1: one PDO assigned to this channel 2: two PDOs assigned to this channel 3: three PDOs assigned to this channel 4: four PDOs assigned to this channel
Default Value	0: no PDO assigned to this channel

Sub-Index	01_h
Description	PDO Mapping object Number of assigned TxPDO
Access	rw
PDO Mapping	no
Value Range	1A00 _h : first Transmit PDO
Default Value	1A00 _h : first Transmit PDO

Sub-Index	02_h
Description	PDO Mapping object Number of assigned TxPDO
Access	rw
PDO Mapping	no
Value Range	1A01 _h : second Transmit PDO
Default Value	1A01 _h : second Transmit PDO

Sub-Index	03_h
Description	PDO Mapping object Number of assigned TxPDO
Access	rw
PDO Mapping	no
Value Range	1A02 _h : third Transmit PDO
Default Value	1A02 _h : third Transmit PDO

Sub-Index	04_h
Description	PDO Mapping object Number of assigned TxPDO
Access	rw
PDO Mapping	no
Value Range	1A03 _h : fourth Transmit PDO
Default Value	1A03 _h : fourth Transmit PDO

4.5.3 Nicht unterstützte Objekte unter CoE

Bei einer Anbindung des CMMP-AS-....M3 unter "CANopen over EtherCAT" werden einige CANopen-Objekte nicht unterstützt, die bei einer Anbindung des CMMP-AS-....M3 über CiA 402 vorhanden sind. Diese Objekte sind in der folgenden Tabelle aufgeführt:

Identifier	Name	Bedeutung
1008 _h	Manufacturer Device Name (String)	Gerätename (Objekt ist nicht vorhanden)
1009 _h	Manufacturer Hardware Version (String)	HW-Version (Objekt ist nicht vorhanden)
100A _h	Manufacturer Software Version (String)	SW-Version (Objekt ist nicht vorhanden)

Identifier	Name	Bedeutung
6089 _h	position_notation_index	Gibt die Anzahl der Nachkommastellen zur Anzeige von Positionswerten in der Steuerung an. Das Objekt ist nur als Datencontainer vorhanden. Es erfolgt keine weitere Auswertung durch die Firmware.
608A _h	position_dimension_index	Gibt die Einheit zur Anzeige von Positionswerten in der Steuerung an. Das Objekt ist nur als Datencontainer vorhanden. Es erfolgt keine weitere Auswertung durch die Firmware.
608B _h	velocity_notation_index	Gibt die Anzahl der Nachkommastellen zur Anzeige von Geschwindigkeitswerten in der Steuerung an. Das Objekt ist nur als Datencontainer vorhanden. Es erfolgt keine weitere Auswertung durch die Firmware.
608C _h	velocity_dimension_index	Gibt die Einheit zur Anzeige von Geschwindigkeitswerten in der Steuerung an. Das Objekt ist nur als Datencontainer vorhanden. Es erfolgt keine weitere Auswertung durch die Firmware.
608D _h	acceleration_notation_index	Gibt die Anzahl der Nachkommastellen zur Anzeige von Beschleunigungswerten in der Steuerung an. Das Objekt ist nur als Datencontainer vorhanden. Es erfolgt keine weitere Auswertung durch die Firmware.
608E _h	acceleration_dimension_index	Gibt die Einheit zur Anzeige von Beschleunigungswerten in der Steuerung an. Das Objekt ist nur als Datencontainer vorhanden. Es erfolgt keine weitere Auswertung durch die Firmware.

Tab. 4.8 Nicht unterstützte Kommunikationsobjekte

4.6 Kommunikations-Zustandsmaschine

Wie in fast allen Feldbusanschlaltungen für Motorcontroller muss der angeschlossene Slave (hier der Motorcontroller CMMP-AS-...-M3) vom Master erst initialisiert werden, bevor er in einer Anwendung durch den Master verwendet werden kann. Zu diesem Zweck ist für die Kommunikation eine Zustandsmaschine (Statemachine) definiert, die einen festen Handlungsablauf für eine solche Initialisierung festlegt.

Solch eine Statemachine ist auch für das EtherCAT-Interface definiert. Dabei dürfen Wechsel zwischen den einzelnen Zuständen der Statemachine nur zwischen bestimmten Zuständen stattfinden und werden immer durch den Master initiiert. Ein Slave darf von sich aus keinen Zustandswechsel vornehmen. Die einzelnen Zustände und die erlaubten Zustandswechsel sind in den folgenden Tabellen und Abbildungen beschrieben.

Zustand	Beschreibung
Power ON	Das Gerät wurde eingeschaltet. Es initialisiert sich selbst und schaltet direkt in den Zustand "Init".
Init	In diesem Zustand wird der EtherCAT-Feldbus durch den Master synchronisiert. Dazu gehört auch das Einrichten der asynchronen Kommunikation zwischen Master und Slave (Mailbox-Telegrammprotokoll). Es findet noch keine direkte Kommunikation zwischen Master und Slave statt. Die Konfiguration startet, gespeicherte Werte werden geladen. Wenn alle Geräte, die an den Bus angeschlossen sind und konfiguriert wurden, wird in den Zustand "Pre-Operational" gewechselt.
Pre-Operational	In diesem Zustand ist die asynchrone Kommunikation zwischen Master und Slave aktiv. Dieser Zustand wird vom Master benutzt, um mögliche zyklische Kommunikation über PDOs einzurichten und notwendige Parametrierungen über die azyklische Kommunikation vorzunehmen. Wenn dieser Zustand fehlerfrei durchlaufen wurde, wechselt der Master in den Zustand "Safe-Operational".
Safe-Operational	Dieser Zustand wird benutzt, um alle Geräte, die an den EtherCAT-Bus angeschlossen sind, in einen sicheren Zustand zu versetzen. Dabei sendet der Slave aktuelle Istwerte an den Master, ignoriert allerdings neue Sollwerte vom Master und benutzt stattdessen sichere Defaultwerte. Wenn dieser Zustand fehlerfrei durchlaufen wurde, wechselt der Master in den Zustand "Operational".
Operational	In diesem Zustand ist sowohl die azyklische, als auch die zyklische Kommunikation aktiv. Master und Slave tauschen Soll- und Istwertdaten aus. In diesem Zustand kann der CMMP-AS-...-M3 über das CoE Protokoll freigegeben und verwendet werden.

Tab. 4.9 Zustände Kommunikations-Zustandsmaschine

Zwischen den einzelnen Zuständen der Kommunikations-Zustandsmaschine sind nur Übergänge gemäß Fig. 4.3 erlaubt:

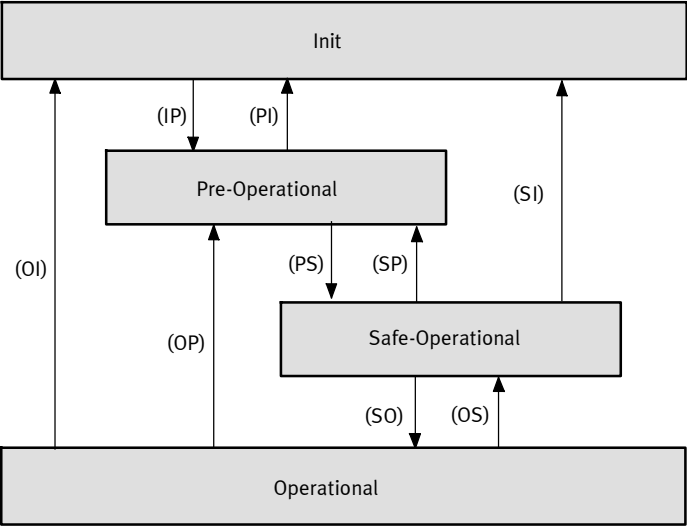


Fig. 4.3 Kommunikations-Zustandsmaschine

In folgender Tabelle sind die Übergänge einzeln beschrieben.

Statusübergang	Status
IP	Start der azyklischen Kommunikation (Mailbox-Telegrammprotokoll)
PI	Stop der azyklischen Kommunikation (Mailbox-Telegrammprotokoll)
PS	Start Inputs Update: Start der zyklischen Kommunikation (Process Data-Telegrammprotokoll). Slave sendet Istwerte an Master. Der Slave ignoriert Sollwerte vom Master und benutzt interne Defaultwerte.
SP	Stop Input Update: Stop der zyklischen Kommunikation (Process Data-Telegrammprotokoll). Der Slave sendet keine Istwerte mehr an den Master.
SO	Start Output Update: Der Slave wertet aktuelle Sollwertvorgaben des Master aus.
OS	Stop Output Update: Der Slave ignoriert die Sollwerte vom Master und benutzt interne Defaultwerte.
OP	Stop Output Update, Stop Input Update: Stop der zyklischen Kommunikation (Process Data-Telegrammprotokoll). Der Slave sendet keine Istwerte mehr an den Master und der Master sendet keine Sollwerte mehr an den Slave.

Statusübergang	Status
SI	Stop Input Update, Stop Mailbox Communication: Stop der zyklischen Kommunikation (Process Data-Telegrammprotokoll) und Stop der azyklischen Kommunikation (Mailbox-Telegrammprotokoll). Der Slave sendet keine Istwerte mehr an den Master und der Master sendet keine Soll- werte mehr an den Slave.
OI	Stop Output Update, Stop Input Update, Stop Mailbox Communication: Stop der zyklischen Kommunikation (Process Data-Telegrammprotokoll) und Stop der azyklischen Kommunikation (Mailbox-Telegrammprotokoll). Der Slave sendet keine Istwerte mehr an den Master und der Master sendet keine Soll- werte mehr an den Slave.

Tab. 4.10 Statusübergänge



In der EtherCAT-Statemachine ist zusätzlich zu den hier aufgeführten Zuständen der Zu-
stand "Bootstrap" spezifiziert. Dieser Zustand für den Motorcontroller CMMP-AS-...-M3
nicht implementiert.

4.6.1 Unterschiede zwischen den Zustandsmaschinen von CANopen und EtherCAT

Beim Betrieb des CMMP-AS-...-M3 über das EtherCAT-CoE-Protokoll wird an Stelle der CANopen-NMT-
Statemachine die EtherCAT-Statemachine verwendet. Diese unterscheidet sich in einigen Punkten von
der CANopen-Statemachine. Diese Unterschiede im Verhalten sind nachfolgend aufgeführt:

- Kein direkter Übergang von Pre-Operational nach Power On
- Kein Stopped-Zustand, sondern direkter Übergang in den INIT-Zustand
- Zusätzlicher Zustand: Safe-Operational

In folgender Tabelle sind die unterschiedlichen Zustände gegenübergestellt:

EtherCAT State	CANopen NMT State
Power ON	Power-On (Initialisierung)
Init	Stopped
Safe-Operational	–
Operational	Operational

Tab. 4.11 Gegenüberstellung der Zustände bei EthetCAT und CANopen

4.7 SDO-Frame

Alle Daten eines SDO-Transfers werden bei CoE über SDO-Frames übertragen. Diese Frames haben den folgenden Aufbau:

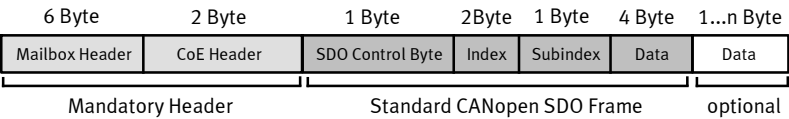


Fig. 4.4 SDO-Frame: Telegrammaufbau

Element	Beschreibung
Mailbox Header	Daten für die Mailbox-Kommunikation (Länge, Adresse und Typ)
CoE Header	Kennung des CoE-Services
SDO Control Byte	Kennung für einen Lese- oder Schreibbefehl
Index	Hauptindex des CANopen-Kommunikationsobjekts
Subindex	Subindex des CANopen-Kommunikationsobjekts
Data	Dateninhalt des CANopen-Kommunikationsobjekts
Data (optional)	Weitere optionale Daten. Diese Option wird vom Motorcontroller CMMP-AS-....M3 nicht unterstützt, da nur Standard-CANopen-Objekte angesprochen werden können. Die maximale Größe dieser Objekte ist 32 Bit.

Tab. 4.12 SDO-Frame: Elemente

Um ein Standard-CANopen-Objekt über einen solchen SDO-Frame zu übertragen, wird der eigentliche CANopen-SDO-Frame in einen EtherCAT-SDO-Frame verpackt und übertragen.

Standard-CANopen-SDO-Frames können verwendet werden für:

- Initialisierung des SDO-Downloads
- Download des SDO-Segments
- Initialisierung des SDO-Uploads
- Upload des SDO-Segments
- Abbruch des SDO-Transfers
- SDO upload expedited request
- SDO upload expedited response
- SDO upload segmented request (max 1 Segment mit 4 Byte Nutzdaten)
- SDO upload segmented response (max 1 Segment mit 4 Byte Nutzdaten)



Alle oben angegebenen Transferarten werden vom Motorcontroller CMMP-AS-....M3 unterstützt.

Da bei Verwendung der CoE-Implementierung des CMMP-AS-....M3 nur die Standard-CANopen-Objekte angesprochen werden können, deren Größe auf 32 Bit (4 Byte) begrenzt ist, werden die Transferarten nur bis zu einer maximalen Datenlänge von 32 Bit (4 Byte) unterstützt.

4.8 PDO-Frame

Die Process Data Objects (PDO) dienen der zyklischen Übertragung von Soll- und Istwertdaten zwischen Master und Slave. Sie müssen vor dem Betrieb des Slave im Zustand "Pre-Operational" durch den Master konfiguriert werden. Anschließend werden sie in PDO-Frames übertragen. Diese PDO-Frames haben den folgenden Aufbau:
Alle Daten eines PDO-Transfers werden bei CoE über PDO-Frames übertragen. Diese Frames haben den folgenden Aufbau:

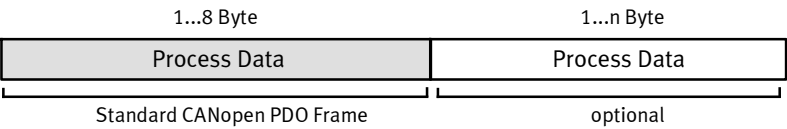



Fig. 4.5 PDO-Frame: Telegrammaufbau

Element	Beschreibung
Process Data	Dateninhalt des PDOs (Process Data Object)
Process Data (optional)	Optionale Dateninhalte weiterer PDOs

Tab. 4.13 PDO-Frame: Elemente

Um ein PDO über das EtherCAT-CoE-Protokoll zu übertragen, müssen die Transmit- und Receive-PDOs zusätzlich zur PDO-Konfiguration (PDO Mapping) einem Übertragungskanal des Sync-Managers zugeordnet werden (→ Kapitel 4.5.1 "Konfiguration der Kommunikationsschnittstelle"). Dabei findet der Datenaustausch von PDOs für den Motorcontroller CMMP-AS-...-M3 ausschließlich über das EtherCAT-Prozessdaten-Telegrammprotokoll statt.

 Die Übertragung von CANopen-Prozessdaten (PDOs) über die azyklische Kommunikation (Mailbox-Telegrammprotokoll) wird vom Motorcontroller CMMP-AS-...-M3 nicht unterstützt.

Da intern im Motorcontroller CMMP-AS-...-M3 alle über das EtherCAT-CoE-Protokoll ausgetauschten Daten direkt an die interne CANopen-Implementierung weitergereicht werden, wird auch das PDO-Mapping wie im Kapitel 3.3 „PDO-Message“ beschrieben realisiert. Das folgende Bild soll diesen Vorgang veranschaulichen:

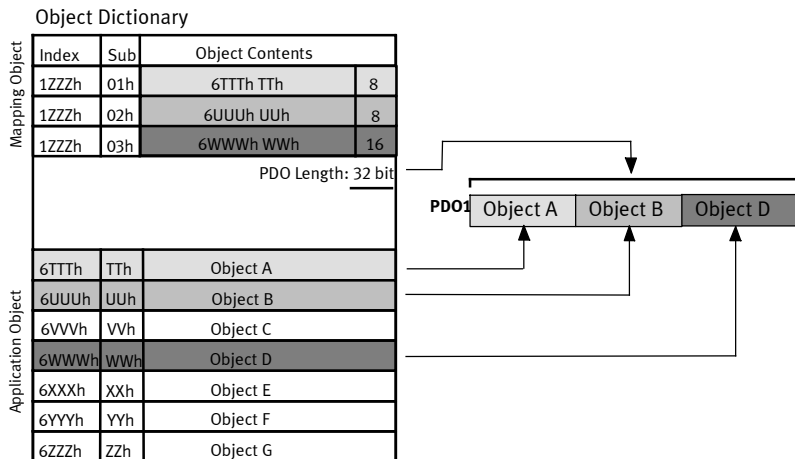


Fig. 4.6 PDO-Mapping

Durch die einfache Weitergabe der über CoE empfangenen Daten an das im CMMP-AS-...-M3 implementierte CANopen-Protokoll können für die zu parametrierenden PDOs neben dem Mapping der CANopen-Objekte auch die für das -Protokoll für den CMMP-AS-...-M3 verfügbaren "Transmission Types" der PDOs verwendet werden.

Der Motorcontroller CMMP-AS-...-M3 unterstützt auch den Transmission Type "Sync Message". Wobei die Sync Message über EtherCAT nicht gesendet zu werden braucht.

Es wird entweder das Eintreffen des Telegramms oder der Hardware-Synchronisationspuls des "Distributed Clocks"-Mechanismus (s.u.) zur Datenübernahme verwendet.

Das EtherCAT-Interface für CMMP-AS-...-M3 unterstützt durch Einsatz des FPGA-Bausteins ESC20 eine Synchronisation über den unter EtherCAT spezifizierten Mechanismus der "Distributed Clocks" (verteilte Uhren). Auf diesen Takt wird der Stromregler des Motorcontrollers CMMP-AS-...-M3 synchronisiert und es erfolgt die Auswertung bzw. das Senden der entsprechend konfigurierten PDOs.

Der Motorcontroller CMMP-AS-...-M3 mit dem EtherCAT-Interface unterstützt die Funktionen:

- Zyklisches PDO-Frame-Telegramm durch das Prozessdaten-Telegrammprotokoll.
- Synchrones PDO-Frame-Telegramm durch das Prozessdaten-Telegrammprotokoll.

Der Motorcontroller CMMP-AS-...-M3 mit EtherCAT-Interface unterstützt vier Receive-PDOs (RxPDO) und vier Transmit-PDOs (TxPDO).

4.9
Error Control

Die EtherCAT-CoE-Implementierung für den Motorcontroller CMMP-AS-...-M3 überwacht folgende Fehlerzustände des EtherCAT-Feldbus:

- FPGA ist nicht bereit bei Start des Systems.
- Es ist ein Busfehler aufgetreten.
- Es ist ein Fehler auf dem Mailbox-Kanal aufgetreten. Folgende Fehler werden hier überwacht:
 - Es wird ein unbekannter Service angefragt.
 - Es soll ein anderes Protokoll als CANopen over EtherCAT (CoE) verwendet werden.
 - Es wird ein unbekannter Sync-Manager angesprochen.

Alle diese Fehler sind als entsprechende Error-Codes für den Motorcontroller CMMP-AS-...-M3 definiert. Tritt einer der oben genannten Fehler auf, wird er über einen “Standard Emergency Frame” an die Steuerung übertragen. Hierzu siehe auch Kapitel 4.10 “Emergency Frame” und Kapitel B “Diagnosemeldungen”.

Der Motorcontroller CMMP-AS-...-M3 mit EtherCAT-Interface unterstützt die Funktion:

- Application Controller übermittelt aufgrund eines Ereignisses eine definierte Fehlermeldungsnummer (Error-Control-Frame-Telegramm vom Regler).

4.10
Emergency Frame

Über den EtherCAT-CoE-Emergency-Frame werden Fehlermeldungen zwischen Master und Slave ausgetauscht. Die CoE-Emergency-Frames dienen dabei direkt der Übertragung der unter CANopen definierten “Emergency Messages”. Dabei werden die CANopen-Telegramme, wie für die SDO- und PDO-Übertragung auch, einfach durch die CoE-Emergency-Frames getunnelt.

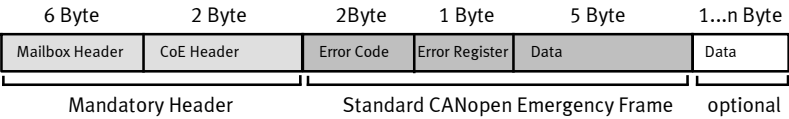


Fig. 4.7
Emergency-Frame: Telegrammaufbau

Element	Beschreibung
Mailbox Header	Daten für die Mailbox-Kommunikation (Länge, Adresse und Typ)
CoE Header	Kennung des CoE-Services
ErrorCode	Error Code der CANopen-EMERGENCY-Message → Kapitel 3.5.2
Error Register	Error Register der CANopen-EMERGENCY-Message → Tab. 3.9
Data	Dateninhalt der CANopen-EMERGENCY-Message
Data (optional)	Weitere optionale Daten. Da in der CoE-Implementation für den Motorcontroller CMMP-AS-...-M3 nur die Standard-CANopen-Emergency-Frames unterstützt werden, wird das "Data (optional)" Feld nicht unterstützt.

Tab. 4.14
Emergency-Frame: Elemente

Da auch hier eine einfache Weitergabe der über CoE empfangenen und gesendeten “Emergency Messages” an das im Motorcontroller implementierte CANopen-Protokoll stattfindet, können alle Fehlermeldungen im Kapitel B nachgeschlagen werden.

4.11 XML-Gerätebeschreibungsdatei

Um EtherCAT-Slave-Geräte einfach an einen EtherCAT-Master anbinden zu können, muss für jedes EtherCAT-Slave-Gerät eine Beschreibungsdatei vorliegen. Diese Beschreibungsdatei ist vergleichbar mit den EDS-Dateien für das CANopen-Feldbussystem oder den GSD-Dateien für Profibus. Im Gegensatz zu diesen ist die EtherCAT-Beschreibungsdatei im XML-Format gehalten, wie es häufig bei Internet- und Webanwendungen benutzt wird und enthält Informationen zu folgenden Merkmalen des EtherCAT-Slave-Gerätes:

- Informationen zum Hersteller des Gerätes
- Name, Typ und Versionsnummer des Gerätes
- Typ und Versionsnummer des zu verwendenden Protokolls für dieses Gerät (z. B. CANopen over Ethernet, ...)
- Parametrierung des Gerätes und Konfiguration der Prozessdaten

In dieser Datei ist die komplette Parametrierung des Slave, inklusive Parametrierung des Sync-Managers und der PDOs, enthalten. Aus diesem Grund kann eine Änderung der Konfiguration des Slave über diese Datei geschehen.

Für den Motorcontroller CMMP-AS-...-M3 hat Festo solch eine Gerätebeschreibungsdatei erstellt. Sie kann von der Homepage von Festo heruntergeladen werden. Um es dem Anwender zu ermöglichen, diese Datei an seine Applikation anzupassen, wird ihr Inhalt hier genauer erklärt.

In der verfügbaren Gerätebeschreibungsdatei werden sowohl das CiA 402-Profil als auch das FHPP-Profil über separat anwählbare Module unterstützt.

4.11.1 Grundsätzlicher Aufbau der Gerätebeschreibungsdatei

Die EtherCAT-Gerätebeschreibungsdatei ist im XML-Format gehalten. Dieses Format hat den Vorteil, dass es mit einem Standard-Texteditor gelesen und editiert werden kann. Eine XML-Datei beschreibt dabei immer eine Baumstruktur. In ihr sind einzelne Zweige durch Knoten definiert. Diese Knoten haben eine Anfangs- und Endmarkierung. Innerhalb eines Knotens können beliebig viele Unterknoten enthalten sein.

BEISPIEL: Grobe Erläuterung des grundsätzlichen Aufbaus einer XML Datei:

```
<EtherCATInfo Version="0.2">
  <Vendor>
    <Id>#x1D</Id>
    <Name>Festo AG</Name>
    <ImageData16x14>424DD60200.....</ImageData16x14>
  </Vendor>
  <Descriptions>
    <Groups>
      <Group SortOrder="1">
        <Type>Festo Electric-Drives</Type>
        <Name LcId="1033">Festo Electric-Drive</Name>
      </Group>
    </Groups>
    <Devices>
      <Device Physics="YY">
      </Device>
    </Devices>
  </Descriptions>
</EteherCATInfo>
```

Für den Aufbau einer XML-Datei müssen folgende kurze Regeln eingehalten werden:

- Jeder Knoten hat einen eindeutigen Namen.
- Jeder Knoten wird geöffnet mit <Knotenname> und geschlossen mit </Knotenname>.

Die Gerätebeschreibungdatei für den Motorcontroller CMMP-AS-...-M3 unter EtherCAT-CoE gliedert sich in folgende Unterpunkte:

Knotenname	Bedeutung	Anpassbar
Vendor	Dieser Knoten enthält den Namen und die ID des Herstellers des Gerätes, zu dem diese Beschreibungsdatei gehört. Zusätzlich ist der Binärcode einer Bitmap mit dem Logo des Herstellers enthalten.	nein
Description	Dieser Unterpunkt enthält die eigentliche Gerätebeschreibung samt Konfiguration und Initialisierung.	teilweise
Group	Dieser Knoten enthält die Zuordnung des Gerätes zu einer Gerätegruppe. Diese Gruppen sind festgelegt und dürfen vom Anwender nicht verändert werden.	nein
Devices	Dieser Unterpunkt enthält die eigentliche Beschreibung des Gerätes.	teilweise

Tab. 4.15 Knoten der Gerätebeschreibungdatei

In der folgenden Tabelle werden ausschließlich die Unterknoten des Knotens "Descriptions" beschrieben, die für die Parametrierung des Motorcontrollers CMMP-AS-...-M3 unter CoE notwendig sind. Alle anderen Knoten sind fest und dürfen vom Anwender nicht verändert werden.

Knotenname	Bedeutung	Anpassbar
RxPDO Fixed=...	Dieser Knoten enthält das PDO-Mapping und die Zuordnung des PDOs zum Sync-Manager für Receive-PDOs.	ja
TxPDO Fixed=...	Dieser Knoten enthält das PDO-Mapping und die Zuordnung des PDOs zum Sync-Manager für Transmit-PDOs.	ja
Mailbox	Unter diesem Knoten können Kommandos definiert werden, die vom Master während des Phasenübergangs von "Pre-Operational" nach "Operational" über SDO-Transfers an den Slave übertragen werden.	ja

Tab. 4.16 Unterknoten des Knotens „Descriptions“

Da für den Anwender zur Anpassung der Gerätebeschreibungsdatei ausschließlich die Knoten aus der Tabelle oberhalb wichtig sind, werden diese in den nachfolgenden Kapiteln detailliert beschrieben. Der restliche Inhalt der Gerätebeschreibungsdatei ist fest und darf vom Anwender nicht geändert werden.

**Wichtig:**

Sollten in der Gerätebeschreibungsdatei Änderungen an anderen Knoten und Inhalten als den Knoten RxPDO, TxPDO und Mailbox vorgenommen werden, kann ein fehlerfreier Betrieb des Gerätes nicht mehr garantiert werden.

4.11.2 Receive-PDO-Konfiguration im Knoten RxPDO

Der Knoten RxPDO dient der Festlegung des Mappings für die Receive-PDOs und deren Zuordnung zu einem Kanal des Sync-Managers. Ein typischer Eintrag in der Gerätebeschreibungsdatei für den Motorcontroller CMMP-AS-...-M3 kann wie folgt aussehen:

```
<RxPDO Fixed="1" Sm="2">
  <Index>#x1600</Index>
  <Name>Outputs</Name>
  <Entry>
    <Index>#x6040</Index>
    <SubIndex>0</SubIndex>
    <BitLen>16</BitLen>
    <Name>Controlword</Name>
    <DataType>UINT</DataType>
  </Entry>
  <Entry>
    <Index>#x6060</Index>
    <SubIndex>0</SubIndex>
    <BitLen>8</BitLen>
    <Name>Mode_Of_Operation</Name>
    <DataType>USINT</DataType>
  </Entry>
</RxPDO>
```

Wie man in obigen Beispiel erkennen kann, wird das gesamte Mapping des Receive-PDOs in einem solchen Eintrag detailliert beschrieben. Dabei gibt der erste große Block die Objektnummer des PDOs und dessen Typ an. Anschließend folgt eine Liste aller CANopen-Objekte, die in das PDO gemappt werden sollen.

In der folgenden Tabelle sind die einzelnen Einträge genauer beschrieben:

Knotenname	Bedeutung	Anpassbar
RxPDO Fixed="1" Sm="2"	Dieser Knoten beschreibt direkt die Beschaffenheit des Receive-PDOs und seiner Zuordnung zum Sync-Manager. Der Eintrag Fixed="1" gibt an, dass das Mapping des Objekts nicht geändert werden kann. Der Eintrag Sm="2" gibt an, dass das PDO dem Sync-Kanal 2 des Sync-Managers zugeordnet werden soll.	nein
Index	Dieser Eintrag enthält die Objektnummer des PDOs. Hier wird das erste Receive-PDO unter der Objektnummer 0x1600 konfiguriert.	ja
Name	Der Name gibt an, ob es sich bei diesem PDO um ein Receive-PDO (Outputs) oder Transmit-PDO (Inputs) handelt. Für ein Receive PDO muss dieser Wert immer auf "Output" gesetzt sein.	nein
Entry	Der Knoten Entry enthält jeweils ein CANopen-Objekt, das in das PDO gemappt werden soll. Ein Entry-Knoten enthält dabei den Index und Subindex des zu mappenden CANopen-Objekts, sowie dessen Name und Datentyp.	ja

Tab. 4.17 Elemente des Knotens „RxPDO“

Die Reihenfolge und das Mapping der einzelnen CANopen-Objekte für das PDO entspricht der Reihenfolge, in der sie über die "Entry"-Einträge in der Gerätebeschreibungsdatei angegeben sind. Die einzelnen Unterpunkte eines "Entry"-Knotens sind in der folgenden Tabelle angegeben:

Knotenname	Bedeutung	Anpassbar
Index	Dieser Eintrag gibt den Index des CANopen-Objekts an, das in das PDO gemappt werden soll.	ja
Subindex	Dieser Eintrag gibt den Subindex des zu mappenden CANopen-Objekts an.	ja
BitLen	Dieser Eintrag gibt die Größe des zu mappenden Objekts in Bit an. Dieser Eintrag muss immer dem Typ des zu mappenden Objekts entsprechen. Erlaubt: 8 Bit / 16 Bit / 32 Bit.	ja
Name	Dieser Eintrag gibt den Namen des zu mappenden Objekts als String an.	ja

Knotenname	Bedeutung	Anpassbar
Data Type	Dieser Eintrag gibt den Datentyp des zu mappenden Objekts an. Dieser kann für die einzelnen CANopen-Objekte der jeweiligen Beschreibung entnommen werden.	ja

Tab. 4.18 Elemente des Knotens „Entry“

4.11.3 Transmit-PDO-Konfiguration im Knoten TxPDO

Der Knoten TxPDO dient der Festlegung des Mappings für die Transmit-PDOs und deren Zuordnung zu einem Kanal des Sync-Managers. Die Konfiguration entspricht dabei der der Receive-PDOs aus Abschnitt 4.11.2 "Receive-PDO-Konfiguration im Knoten RxPDO" mit dem Unterschied, dass der Knoten "Name" des PDOs anstelle von "Outputs" auf "Inputs" gesetzt werden muss.

4.11.4 Initialisierungskommandos über den Knoten "Mailbox"

Der Knoten "Mailbox" in der Gerätebeschreibungdatei dient dem Beschreiben von CANopen-Objekten durch den Master im Slave während der Initialisierungsphase. Die Kommandos und Objekte, die dort beschrieben werden sollen, werden über spezielle Einträge festgelegt. In diesen Einträgen ist der Phasenübergang, bei dem dieser Wert beschrieben werden soll, festgelegt. Weiterhin enthält solch ein Eintrag die Objektnummer (Index und Subindex), sowie den Datenwert, der geschrieben werden soll und einen Kommentar.

Ein typischer Eintrag hat die folgende Form:

```
<InitCmd>
    <Transition>PS</Transition>
    <Index#>x6060</Index>
    <SubIndex>0</SubIndex>
    <Data>03</Data>
    <Comment>velocity mode</Comment>
</InitCmd>
```

In obigem Beispiel wird im Zustandsübergang PS von "Pre-Operational" nach "Safe Operational" die Betriebsart im Objekt "modes_of_operation" auf "Drehzahlregelung" gesetzt. Die einzelnen Unterknoten haben folgende Bedeutung:

Knotenname	Bedeutung	Anpassbar
Transition	Name des Zustandsübergangs, bei dessen Auftreten dieses Kommando ausgeführt werden soll (→ Kapitel 4.6 „Kommunikations-Zustandsmaschine“)	ja
Index	Index des zu schreibenden CANopen-Objekts	ja
Subindex	Subindex des zu schreibenden CANopen-Objekts	ja
Data	Datenwert, der geschrieben werden soll, als hexadezimaler Wert	ja
Comment	Kommentar zu diesem Kommando	ja

Tab. 4.19 Elemente des Knotens „InitCmd“

**Wichtig:**

In einer Gerätebeschreibungsdatei für den Motorcontroller CMMP-AS-...-M3 sind in dieser Sektion einige Einträge bereits vorgegeben. Diese Einträge müssen erhalten bleiben und dürfen vom Anwender nicht geändert werden.

4.12 Synchronisation (Distributed Clocks)

Die zeitliche Synchronisation wird bei EtherCAT über so genannte "verteilte Uhren" (Distributed Clocks) realisiert. Dabei enthält jeder EtherCAT-Slave eine Echtzeituhr, die während der Initialisierungsphase durch den Clock-Master in allen Slaves synchronisiert wird. Anschließend werden die Uhren in allen Slaves im laufenden Betrieb nachgestellt. Der Clock-Master ist der erste Slave im Netzwerk. Dadurch ist im gesamten System eine einheitliche Zeitbasis vorhanden, auf die sich die einzelnen Slaves synchronisieren können. Die unter CANopen für diesen Zweck vorgesehenen Sync-Telegramme entfallen unter CoE.

Das im Motorcontroller CMMP-AS-...-M3 verwendete FPGA ESC20 unterstützt Distributed Clocks. Eine sehr exakte zeitliche Synchronisation kann hiermit durchgeführt werden. Die Zykluszeit des EtherCAT-Frames muss exakt zur Zykluszeit t_p des reglerinternen Interpolators passen. Gegebenfalls muss die Interpolatorzeit über das in der Gerätebeschreibungsdatei enthaltene Objekt angepasst werden.

In der gegenwärtigen Implementierung ist es aber auch möglich ohne Distributed Clocks eine synchrone Übernahme der PDO-Daten und ein Synchronisieren der reglerinternen PLL auf den synchronen Datenrahmen des EtherCAT-Frames zu erreichen. Hierbei nutzt die Firmware das Eintreffen des EtherCAT-Frames als Zeitbasis.

Es gelten die folgenden Einschränkungen:

- Der Master muss die EtherCAT-Frames mit einem sehr geringen Jitter senden können.
- Die Zykluszeit des EtherCAT-Frames muss exakt zur Zykluszeit t_p des reglerinternen Interpolators passen.
- Das Ethernet muss exklusiv für den EtherCAT-Frame zur Verfügung stehen. Andere Telegramme müssen ggf. auf das Raster synchronisiert werden und dürfen nicht den Bus blockieren.

5 Parameter Einstellen

Bevor der Motorcontroller die gewünschte Aufgabe (Momenten-, Drehzahlregelung, Positionierung) ausführen kann, müssen zahlreiche Parameter des Motorcontrollers an den verwendeten Motor und die spezifische Applikation angepasst werden. Dabei sollte in der Reihenfolge der anschließenden Kapitel vorgegangen werden. Im Anschluss an die Einstellung der Parameter wird die Gerätesteuerung und die Nutzung der jeweiligen Betriebsarten erläutert.



Das Display des Motorcontrollers zeigt ein „A“ (Attention) an, wenn der Motorcontroller noch nicht geeignet parametrisiert wurde. Soll der Motorcontroller komplett über CANopen parametrisiert werden, müssen Sie das Objekt 6510_h_C0_h beschreiben, um diese Anzeige zu unterdrücken (→ Seite 150).

Neben den hier ausführlich beschriebenen Parametern sind im Objektverzeichnis des Motorcontrollers weitere Parameter vorhanden, die gemäß CANopen implementiert werden müssen. Sie enthalten aber in der Regel keine Informationen, die beim Aufbau einer Applikation mit einem Motorcontroller CMMP-AS-...-M3/-M0 sinnvoll verwendet werden kann. Bei Bedarf ist in den Spezifikationen von CiA nachzulesen.

5.1 Parametersätze laden und speichern

Übersicht

Der Motorcontroller verfügt über drei Parametersätze:

- **Aktueller Parametersatz**

Dieser Parametersatz befindet sich im flüchtigen Speicher (RAM) des Motorcontrollers. Er kann mit der Parametriersoftware oder über den CAN-Bus beliebig gelesen und beschrieben werden. Beim Einschalten des Motorcontrollers wird der Applikations-Parametersatz in den aktuellen Parametersatz kopiert.

- **Default-Parametersatz**

Dieses ist der vom Hersteller standardmäßig vorgegebene unveränderliche Parametersatz des Motorcontrollers. Durch einen Schreibvorgang in das CANopen-Objekt 1011_h_01_h (restore_all_default_parameters) kann der Default-Parametersatz in den aktuellen Parametersatz kopiert werden. Dieser Kopiervorgang ist nur bei ausgeschalteter Endstufe möglich.

- **Applikations-Parametersatz**

Der aktuelle Parametersatz kann in den nichtflüchtigen Flash-Speicher gesichert werden. Der Speichervorgang wird mit einem Schreibzugriff auf das CANopen-Objekt 1010_h_01_h (save_all_parameters) ausgelöst. Beim Einschalten des Motorcontrollers wird automatisch der Applikations-Parametersatz in den aktuellen Parametersatz kopiert.

Die nachfolgende Grafik veranschaulicht die Zusammenhänge zwischen den einzelnen Parametersätzen.

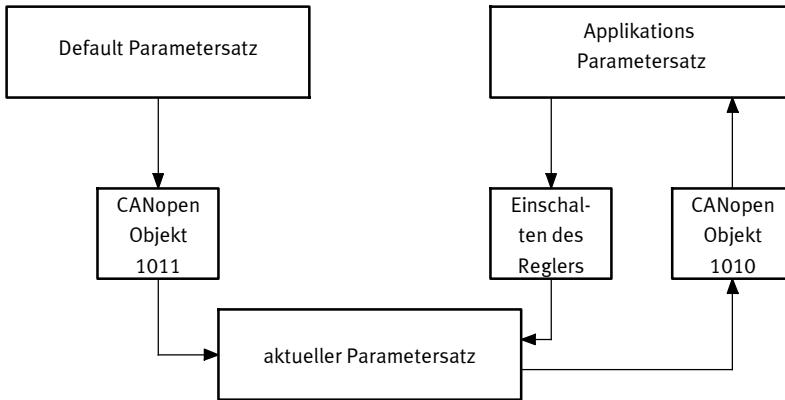


Fig. 5.1 Zusammenhänge Parametersätze

Es sind zwei unterschiedliche Konzepte zur Parametersatzverwaltung denkbar:

1. Der Parametersatz wird mit der Parametrierungssoftware erstellt und komplett in die einzelnen Controller übertragen. Bei diesem Verfahren müssen nur die ausschließlich via CANopen zugänglichen Objekte über den CAN-Bus eingestellt werden. Nachteilig ist hierbei, dass für jede Inbetriebnahme einer neuen Maschine oder im Falle einer Reparatur (Controlleraustausch) die Parametrierungssoftware benötigt wird.
2. Diese Variante basiert auf der Tatsache, dass die meisten applikationsspezifischen Parametersätze nur in wenigen Parametern vom Default-Parametersatz abweichen. Dadurch ist es möglich, dass der aktuelle Parametersatz nach jedem Einschalten der Anlage über den CAN-Bus neu aufgebaut wird. Hierzu wird von der übergeordneten Steuerung zunächst der Default-Parametersatz geladen (Aufruf des CANopen-Objekts 1011h_01h (restore_all_default_parameters)). Danach werden nur die abweichenden Objekte übertragen. Der gesamte Vorgang dauert pro Controller unter 1 Sekunde. Vorteilhaft ist, dass dieses Verfahren auch bei unparametrierten Controllern funktioniert, so dass die Inbetriebnahme von neuen Anlagen oder der Austausch einzelner Controller unproblematisch ist und die Parametrierungssoftware hierfür nicht benötigt wird.



Warnung

Stellen Sie vor dem allerersten Einschalten der Endstufe sicher, dass der Controller wirklich die von Ihnen gewünschten Parameter enthält.

Ein falsch parametrierter Controller kann unkontrolliert drehen und Personen- oder Sachschäden verursachen.

Beschreibung der Objekte**Objekt 1011_h: restore_default_parameters**

Index	1011_h
Name	restore_parameters
Object Code	ARRAY
No. of Elements	1
Data Type	UINT32

Sub-Index	01_h
Description	restore_all_default_parameters
Access	rw
PDO Mapping	no
Units	–
Value Range	64616F6C _h („load“)
Default Value	1 (read access)

Signature	MSB			LSB
ASCII	d	a	o	l
hex	64 _h	61 _h	6F _h	6C _h

Tab. 5.1 Beispiel für ASCII-Text „load“

Das Objekt 1011_h_01_h (restore_all_default_parameters) ermöglicht, den aktuellen Parametersatz in einen definierten Zustand zu versetzen. Hierfür wird der Default-Parametersatz in den aktuellen Parametersatz kopiert. Der Kopiervorgang wird durch einen Schreibzugriff auf dieses Objekt ausgelöst, wobei als Datensatz der String „load“ in hexadezimaler Form zu übergeben ist.

Dieser Befehl darf nur bei deaktivierter Endstufe ausgeführt werden. Andernfalls wird der SDO-Fehler „Daten können nicht übertragen oder gespeichert werden, da sich der Motorcontroller dafür nicht im richtigen Zustand befindet“ erzeugt. Wird die falsche Kennung gesendet, wird der Fehler „Daten können nicht übertragen oder gespeichert werden“ erzeugt. Wird lesend auf das Objekt zugegriffen, wird eine 1 zurückgegeben, um anzuzeigen, dass das Zurücksetzen auf Defaultwerte unterstützt wird.

Die Parameter der CAN-Kommunikation (Knoten-Nr., Baudrate und Betriebsart) sowie zahlreiche Winkelgeber-Einstellungen (die zum Teil einen Reset erfordern um wirksam zu werden) bleiben hierbei unverändert.

Objekt 1010_h: store_parameters

Index	1010_h
Name	store_parameters
Object Code	ARRAY
No. of Elements	1
Data Type	UINT32

Sub-Index	01 _h
Description	save_all_parameters
Access	rw
PDO Mapping	no
Units	–
Value Range	65766173 _h („save“)
Default Value	1

Signature	MSB			LSB
ASCII	e	v	a	s
hex	65 _h	76 _h	61 _h	73 _h

Tab. 5.2 Beispiel für ASCII-Text „save“

Soll der Default-Parametersatz auch in den Applikations-Parametersatz übernommen werden, dann muss außerdem auch das Objekt 1010_h01_h (save_all_parameters) aufgerufen werden. Wird das Objekt über ein SDO geschrieben, ist das Defaultverhalten, dass das SDO sofort beantwortet wird. Die Antwort spiegelt somit nicht das Ende des Speichervorgangs wider. Das Verhalten kann jedoch über das Objekt 6510_hF0_h (compatibility_control) geändert werden.

5.2 **Kompatibilitäts-Einstellungen**

Übersicht

Um einerseits kompatibel zu früheren CANopen-Implementationen (z. B. auch in anderen Gerätefamilien) bleiben zu können und andererseits Änderungen und Korrekturen gegenüber der CiA 402 und der CiA 301 ausführen zu können, wurde das Objekt compatibility_control eingefügt. Im Defaultparametersatz liefert dieses Objekt 0, d. h. Kompatibilität zu früheren Versionen. Für neue Applikationen empfehlen wir, die definierten Bits zu setzen, um so eine möglichst hohe Übereinstimmung mit den genannten Standards zu ermöglichen.

Beschreibung der Objekte**In diesem Kapitel behandelte Objekte**

Index	Objekt	Name	Typ	Attr.
6510_F0 _h	VAR	compatibility_control	UINT16	rw

Objekt 6510_h_F0_h: compatibility_control

Sub-Index	F0 _h
Description	compatibility_control
Data Type	UINT16
Access	rw
PDO Mapping	no
Units	–
Value Range	0 ... 1FF _h , → Tabelle
Default Value	0

Bit	Wert	Name	
0	0001 _h	homing_method_scheme	Das Bit hat die gleiche Bedeutung wie Bit 2 und ist aus Kompatibilitätsgründen vorhanden. Wird Bit 2 gesetzt, wird dieses Bit auch gesetzt und umgekehrt.
1	0002 _h	reserved	Das Bit ist reserviert. Es darf nicht gesetzt werden.
2	0004 _h	homing_method_scheme	Wenn dieses Bit gesetzt ist, sind die Referenzfahrtmethoden 32 ... 35 gemäß CiA 402 nummeriert, anderenfalls ist die Nummerierung kompatibel zu früheren Implementierungen. (→ auch Kap. 7.2.3). Wird dieses Bit gesetzt, wird auch Bit 0 gesetzt und umgekehrt
3	0008 _h	reserved	Das Bit ist reserviert. Es darf nicht gesetzt werden.
4	0010 _h	response_after_save	Wenn dieses Bit gesetzt ist, wird die Antwort auf save_all_parameters erst gesendet, wenn das Speichern abgeschlossen wurde. Dies kann mehrere Sekunden dauern, was ggf. zu einem Timeout in der Steuerung führt. Ist das Bit gelöscht, wird sofort geantwortet, es ist allerdings zu berücksichtigen, dass der Speichervorgang noch nicht abgeschlossen ist.
5	0020 _h	reserved	Das Bit ist reserviert. Es darf nicht gesetzt werden.

Bit	Wert	Name	
6	0040 _h	homing_to_zero	Bisher besteht eine Referenzfahrt unter CANopen nur aus 2 Phasen (Suchfahrt und Kriechfahrt). Der Antrieb fährt anschließend nicht auf die ermittelte Nullposition (die z. B. durch den homing_offset zur gefundenen Referenzposition verschoben sein kann). Wird dieses Bit gesetzt, wird dieses Standardverhalten geändert und der Antrieb schließt der Referenzfahrt eine Fahrt auf Null an. → hierzu Kap. 7.2 Betriebsart Referenzfahrt (Homing Mode)
7	0080 _h	device_control	Wenn dieses Bit gesetzt ist, wird Bit 4 des statusword (voltage_enabled) gemäß CiA 402 v2.0 ausgegeben. Außerdem ist der Zustand FAULT_REACTION_ACTIVE vom Zustand FAULT unterscheidbar. → hierzu Kapitel 6
8	0100 _h	reserved	Das Bit ist reserviert. Es darf nicht gesetzt werden.

5.3 Umrechnungsfaktoren (Factor Group)

Übersicht

Motorcontroller werden in einer Vielzahl von Anwendungsfällen eingesetzt: Als Direktantrieb, mit nachgeschaltetem Getriebe, für Linearantriebe etc. Um für alle diese Anwendungsfälle eine einfache Parametrierung zu ermöglichen, kann der Motorcontroller mit Hilfe der Factor Group so parametriert werden, dass der Nutzer alle Größen wie z. B. die Drehzahl direkt in den gewünschten Einheiten am Abtrieb angeben bzw. auslesen kann (z. B. bei einer Linearachse Positionswerte in Millimeter und Geschwindigkeiten in Millimeter pro Sekunde). Der Motorcontroller rechnet die Eingaben dann mit Hilfe der Factor Group in seine internen Einheiten um. Für jede physikalische Größe (Position, Geschwindigkeit und Beschleunigung) ist ein Umrechnungsfaktor vorhanden, um die Nutzer-Einheiten an die eigene Applikation anzupassen. Die durch die Factor Group eingestellten Einheiten werden allgemein als position_units, speed_units oder acceleration_units bezeichnet. Die folgende Skizze verdeutlicht die Funktion der Factor Group:

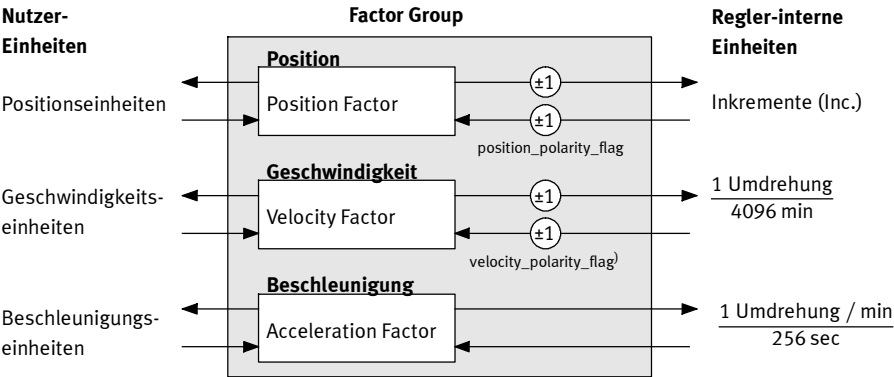


Fig. 5.2 Factor Group

Alle Parameter werden im Motorcontroller grundsätzlich in seinen internen Einheiten gespeichert und erst beim Einschreiben oder Auslesen mit Hilfe der Factor Group umgerechnet. Daher sollte die Factor Group vor der allerersten Parametrierung eingestellt werden und während einer Parametrierung nicht geändert werden. Standardmäßig ist die Factor Group auf folgende Einheiten eingestellt:

Größe	Bezeichnung	Einheit	Erklärung
Länge	Positionseinheiten	Inkrement	65536 Inkremente pro Umdrehung
Geschwindigkeit	Geschwindigkeitseinheiten	min ⁻¹	Umdrehungen pro Minute
Beschleunigung	Beschleunigungseinheiten	(min ⁻¹)/s	Drehzahlerhöhung pro Sekunde

Tab. 5.3 Voreinstellung Factor Group

Beschreibung der Objekte
In diesem Kapitel behandelte Objekte

Index	Objekt	Name	Typ	Attr.
607E _h	VAR	polarity	UINT8	rw
6093 _h	ARRAY	position_factor	UINT32	rw
6094 _h	ARRAY	velocity_encoder_factor	UINT32	rw
6097 _h	ARRAY	acceleration_factor	UINT32	rw

Objekt 6093_h: position_factor

Das Objekt position_factor dient zur Umrechnung aller Längeneinheiten der Applikation von positon_units in die interne Einheit Inkremente (65536 Inkremente entsprechen 1 Umdrehung). Es besteht aus Zähler und Nenner.

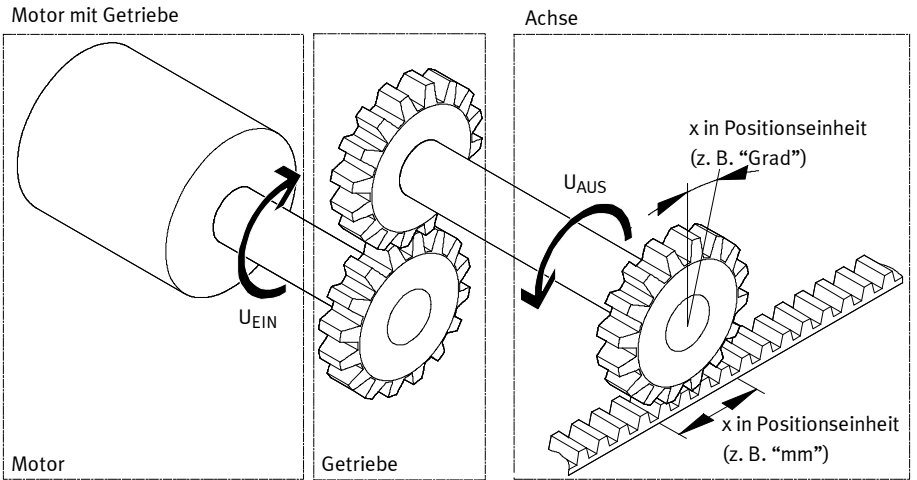


Fig. 5.3 Berechnung der Positionseinheiten

Index	6093 _h
Name	position_factor
Object Code	ARRAY
No. of Elements	2
Data Type	UINT32

Sub-Index	01_h
Description	numerator
Access	rw
PDO Mapping	yes
Units	–
Value Range	–
Default Value	1

Sub-Index	02_h
Description	divisor
Access	rw
PDO Mapping	yes
Units	–
Value Range	–
Default Value	1

In die Berechnungsformel des position_factor gehen folgende Größen ein:

Parameter	Beschreibung
gear_ratio	Getriebeverhältnis zwischen Umdrehungen am Eintrieb (UEIN) und Umdrehungen am Abtrieb (UAUS)
feed_constant	Verhältnis zwischen Umdrehungen am Abtrieb (UAUS) und Bewegung in position_units (z. B. 1 U = 360 Grad)

Tab. 5.4 Parameter Positionsfaktor

Die Berechnung des position_factors erfolgt mit folgender Formel:

$$\text{position_factor} = \frac{\text{numerator}}{\text{divisor}} = \frac{\text{Getriebeübersetzung} * \text{Inkrement/Umdrehung}}{\text{Vorschubkonstante}}$$

Der position_factor muss getrennt nach Zähler und Nenner in den Motorcontroller geschrieben werden. Daher kann es notwendig sein, den Bruch durch geeignete Erweiterung auf ganze Zahlen zu bringen.



Der position_factor darf nicht größer als 2²⁴ sein

BEISPIEL

Zunächst muss die gewünschte Einheit (Spalte 1) und die gewünschten Nachkommastellen (NK) festgelegt, sowie der Getriebefaktor und ggf. die Vorschubkonstante der Applikation ermittelt werden. Diese Vorschubkonstante wird dann in den gewünschten Positions-Einheiten dargestellt (Spalte 2). Letztlich können alle Werte in die Formel eingesetzt und der Bruch berechnet werden:

Ablauf Berechnung Positionsfaktor				
Positionseinheiten	Vorschubkonstante	Getriebe-faktor	Formel	Ergebnis gekürzt
Grad, 1 NK → 1/10 Grad (°/10)	$1 U_{AUS} = \frac{3600}{10}$	$1/1$	$\frac{1}{1} * 65536 \text{ Ink} = \frac{65536}{3600 \frac{0}{10}} \text{ Ink}$	$\frac{\text{num} : 4096}{\text{div} : 225}$

Fig. 5.4 Ablauf Berechnung Positionsfaktor

Beispiele Berechnung Positionsfaktor				
Positionseinheiten ¹⁾	Vorschubkonstante ²⁾	Getriebe-faktor ³⁾	Formel ⁴⁾	Ergebnis gekürzt
Inkreme, 0 NK → Inc.	$1 U_{AUS} = 65536 \text{ Ink}$	$1/1$	$\frac{1}{1} * 65536 \text{ Ink} = \frac{1 \text{ Ink}}{1 \text{ Ink}}$	$\frac{\text{num} : 1}{\text{div} : 1}$
Grad, 1 NK → 1/10 Grad (°/10)	$1 U_{AUS} = \frac{3600}{10}$	$1/1$	$\frac{1}{1} * 65536 \text{ Ink} = \frac{65536}{3600 \frac{0}{10}} \text{ Ink}$	$\frac{\text{num} : 4096}{\text{div} : 225}$
Umdr., 2 NK → 1/100 Umdr. (U/100)	$1 U_{AUS} = 100 \frac{U}{100}$	$1/1$	$\frac{1}{1} * 65536 \text{ Ink} = \frac{65536}{100 \frac{1}{100}} \text{ Ink}$	$\frac{\text{num} : 16384}{\text{div} : 25}$
		$2/3$	$\frac{2}{3} * 65536 \text{ Ink} = \frac{131072}{300 \frac{1}{100}} \text{ Ink}$	$\frac{\text{num} : 32768}{\text{div} : 75}$
mm, 1 NK → 1/10 mm (mm/10)	$1 U_{AUS} = 631,5 \frac{\text{mm}}{10}$	$4/5$	$\frac{4}{5} * 65536 \text{ Ink} = \frac{2621440}{31575 \frac{\text{mm}}{10}} \text{ Ink}$	$\frac{\text{num} : 524288}{\text{div} : 6315}$

- 1) Gewünschte Einheit am Abtrieb
- 2) Positionseinheiten pro Umdrehung am Abtrieb (U_{AUS}). Vorschubkonstante des Antriebs * 10^{-NK} (Nachkommastellen)
- 3) Umdrehungen am Eintrieb pro Umdrehungen am Austrieb (U_{EIN} pro U_{AUS})
- 4) Werte in Formel einsetzen.

Tab. 5.5 Beispiele Berechnung Positionsfaktor

6094_h: velocity_encoder_factor

Das Objekt velocity_encoder_factor dient zur Umrechnung aller Geschwindigkeitswerte der Applikation von speed_units in die interne Einheit Umdrehungen pro 4096 Minuten. Es besteht aus Zähler und Nenner.

Index	6094_h
Name	velocity_encoder_factor
Object Code	ARRAY
No. of Elements	2
Data Type	UINT32

Sub-Index	01_h
Description	numerator
Access	rw
PDO Mapping	yes
Units	–
Value Range	–
Default Value	1000 _h

Sub-Index	02_h
Description	divisor
Access	rw
PDO Mapping	yes
Units	–
Value Range	–
Default Value	1

Die Berechnung des velocity_encoder_factor setzt sich im Prinzip aus zwei Teilen zusammen: Einem Umrechnungsfaktor von internen Längeneinheiten in position_units und einem Umrechnungsfaktor von internen Zeiteinheiten in benutzerdefinierte Zeiteinheiten (z. B. von Sekunden in Minuten). Der erste Teil entspricht der Berechnung des position_factor für den zweiten Teil kommt ein zusätzlicher Faktor zur Berechnung hinzu:

Parameter	Beschreibung
time_factor_v	Verhältnis zwischen interner Zeiteinheit und benutzerdefinierter Zeiteinheit. (z. B. 1 min = $1/4096$ 4096 min)
gear_ratio	Getriebeverhältnis zwischen Umdrehungen am Eintrieb (U_{EIN}) und Umdrehungen am Abtrieb (U_{AUS})
feed_constant	Verhältnis zwischen Umdrehungen am Abtrieb (U_{AUS}) und Bewegung in position_units (z. B. 1 U = 360 Grad)

Tab. 5.6 Parameter Geschwindigkeitsfaktor

Die Berechnung des velocity_encoder_factors erfolgt mit folgender Formel:

$$\text{velocity_encoder_factor} = \frac{\text{numerator}}{\text{divisor}} = \frac{\text{gear_ratio} * \text{time_factor_v}}{\text{feedconstant}}$$

i

Der velocity_encoder_factor darf nichtgrößer als 2²⁴ sein

Wie der position_factor wird auch der velocity_encoder_factor getrennt nach Zähler und Nenner in den Motorcontroller geschrieben werden. Daher kann es notwendig sein, den Bruch durch geeignete Erweiterung auf ganze Zahlen zu bringen.

BEISPIEL

Zunächst muss die gewünschte Einheit (Spalte 1) und die gewünschten Nachkommastellen (NK) festgelegt, sowie der Getriebefaktor und ggf. die Vorschubkonstante der Applikation ermittelt werden. Diese Vorschubkonstante wird dann in den gewünschten Positions-Einheiten dargestellt (Spalte 2). Anschließend wird die gewünschte Zeiteinheit in die Zeiteinheit des Motorcontrollers umgerechnet (Spalte 3).
 Letztlich können alle Werte in die Formel eingesetzt und der Bruch berechnet werden:

Ablauf Berechnung Geschwindigkeitsfaktor						
Geschw.- einheiten	Vorsch.- konst.	Zeitkonstante	Getr.	Formel	Ergebnis gekürzt	
mm/s, 1 NK → 1/10 mm/s (mm/10 s)	63,15 $\frac{\text{mm}}{\text{U}}$ ⇒ 1 U _{AUS} = 631,5 $\frac{\text{mm}}{10}$	1 $\frac{1}{\text{s}}$ = 60 $\frac{1}{\text{min}}$ = 60 * 4096 $\frac{1}{4096 \text{ min}}$	4/5	$\frac{4}{5} * \frac{60 * 4096 \frac{1}{4096 \text{ min}}}{1 \frac{1}{\text{s}}} = \frac{1966080}{4096 \text{ min}}$ $631,5 \frac{\text{mm}}{10} = \frac{6315 \frac{\text{mm}}{10\text{s}}}{10}$	1966080 $\frac{1}{4096 \text{ min}}$	num: 131072 div: 421

Fig. 5.5 Ablauf Berechnung Geschwindigkeitsfaktor

Beispiele Berechnung Geschwindigkeitsfaktor					
Geschw.- einheiten ¹⁾	Vorsch.- konst. ²⁾	Zeitkonstante ³⁾	Getr. 4)	Formel ⁵⁾	Ergebnis gekürzt
U/min, 0 NK → U/min	$1 U_{AUS} =$ $1 U_{AUS}$	$1 \frac{1}{\text{min}} =$ $4096 \frac{1}{4096 \text{ min}}$	1/1	$\frac{1}{1} * \frac{4096 \frac{1}{4096 \text{ min}}}{1 \frac{1}{\text{min}}} = \frac{4096 \frac{1}{4096 \text{ min}}}{1 \frac{1}{\text{min}}}$	num: 4096 div: 1
U/min, 2 NK → 1/100 U/min (U/100 min)	$1 U_{AUS} =$ $100 \frac{U}{100}$	$1 \frac{1}{\text{min}} =$ $4096 \frac{1}{4096 \text{ min}}$	2/3	$\frac{2}{3} * \frac{4096 \frac{1}{4096 \text{ min}}}{1 \frac{1}{\text{min}}} = \frac{8192 \frac{1}{4096 \text{ min}}}{100 \frac{1}{100}} = \frac{8192}{300} \frac{1}{100 \text{ min}}$	num: 2048 div: 75
°/s, 1 NK → 1/10 °/s (°/10 s)	$1 U_{AUS} =$ $3600 \frac{°}{10}$	$1 \frac{1}{s} =$ $60 \frac{1}{\text{min}} =$ $60 * 4096 \frac{1}{4096 \text{ min}}$	1/1	$\frac{1}{1} * \frac{60 * 4096 \frac{1}{4096 \text{ min}}}{1 \frac{1}{s}} = \frac{245760 \frac{1}{4096 \text{ min}}}{3600 \frac{°}{10}} = \frac{245760}{3600} \frac{1}{10 s}$	num: 1024 div: 15
mm/s, 1 NK → 1/10 mm/s (mm/10 s)	$63,15 \frac{\text{mm}}{U}$ ⇒ $1 U_{AUS} =$ $631,5 \frac{\text{mm}}{10}$	$1 \frac{1}{s} =$ $60 \frac{1}{\text{min}} =$ $60 * 4096 \frac{1}{4096 \text{ min}}$	4/5	$\frac{4}{5} * \frac{60 * 4096 \frac{1}{4096 \text{ min}}}{1 \frac{1}{s}} = \frac{1966080 \frac{1}{4096 \text{ min}}}{631,5 \frac{\text{mm}}{10}} = \frac{1966080}{6315} \frac{1}{10 s}$	num: 131072 div: 421

1) Gewünschte Einheit am Abtrieb

2) Positionseinheiten pro Umdrehung am Abtrieb (U_{AUS}). Vorschubkonstante des Antriebs * 10^{-NK} (Nachkommastellen)

3) Zeitfaktor_v: Gewünschte Zeiteinheit pro interne Zeiteinheit

4) Getriebefaktor: U_{EIN} pro U_{AUS}

5) Werte in Formel einsetzen.

Tab. 5.7 Beispiele Berechnung Geschwindigkeitsfaktor

6097_h: acceleration_factor

Das Objekt acceleration_factor dient zur Umrechnung aller Beschleunigungswerte der Applikation von acceleration_units in die interne Einheit Umdrehungen pro Minute pro 256 Sekunden. Es besteht aus Zähler und Nenner.

Index	6097_h
Name	acceleration_factor
Object Code	ARRAY
No. of Elements	2
Data Type	UINT32

Sub-Index	01_h
Description	numerator
Access	rw
PDO Mapping	yes
Units	–
Value Range	–
Default Value	100 _h

Sub-Index	02_h
Description	divisor
Access	rw
PDO Mapping	yes
Units	–
Value Range	–
Default Value	1

Die Berechnung des `acceleration_factor` setzt sich ebenfalls aus zwei Teilen zusammen: Einem Umrechnungsfaktor von internen Längeneinheiten in `position_units` und einem Umrechnungsfaktor von internen Zeiteinheiten zum Quadrat in benutzerdefinierte Zeiteinheiten zum Quadrat (z. B. von Sekunden² in Minuten²). Der erste Teil entspricht der Berechnung des `position_factor` für den zweiten Teil kommt ein zusätzlicher Faktor hinzu:

Parameter	Beschreibung
<code>time_factor_a</code>	Verhältnis zwischen interner Zeiteinheit zum Quadrat und benutzerdefinierter Zeiteinheit zum Quadrat. (z. B. $1 \text{ min}^2 = 1 \text{ min} \times 1 \text{ min} = 60 \text{ s} \times 1 \text{ min} = \frac{60}{256} 256 \text{ min} \times \text{s}$).
<code>gear_ratio</code>	Getriebeverhältnis zwischen Umdrehungen am Eintrieb (U_{EIN}) und Umdrehungen am Abtrieb (U_{AUS}).
<code>feed_constant</code>	Verhältnis zwischen Umdrehungen am Abtrieb (U_{AUS}) und Bewegung in <code>position_units</code> (z. B. $1 \text{ U} = 360 \text{ Grad}$)

Tab. 5.8 Parameter Beschleunigungsfaktor

Die Berechnung des `acceleration_factor` erfolgt mit folgender Formel:

$$\text{acceleration_factor} = \frac{\text{numerator}}{\text{divisor}} = \frac{\text{gear_ratio} * \text{time_factor_a}}{\text{feed_constant}}$$

Auch der `acceleration_factor` wird getrennt nach Zähler und Nenner in den Motorcontroller geschrieben werden, so dass eventuell erweitert werden muss.

BEISPIEL

Zunächst muss die gewünschte Einheit (Spalte 1) und die gewünschten Nachkommastellen (NK) festgelegt, sowie der Getriebefaktor und ggf. die Vorschubkonstante der Applikation ermittelt werden. Diese Vorschubkonstante wird dann in den gewünschten Positionseinheiten dargestellt (Spalte 2). Anschließend wird die gewünschte Zeiteinheit in die Zeiteinheit des Motorcontrollers umgerechnet (Spalte 3). Letztlich können alle Werte in die Formel eingesetzt und der Bruch berechnet werden:

Ablauf Berechnung Beschleunigungsfaktor					
Beschl.- einheiten	Vorsch.- konst.	Zeitkonstante	Getr.	Formel	Ergebnis gekürzt
mm/s ² , 1 NK → 1/10 mm/s ² (mm/10 s ²)	63,15 $\frac{\text{mm}}{\text{U}}$ ⇒ 1 U _{AUS} = 631,5 $\frac{\text{mm}}{10}$	$1 \frac{1}{s^2} =$ $60 \frac{1}{\text{min} * s} =$ $60 * 256 \frac{1}{256 * s}$	4/5	$4 * \frac{60 * 256 \frac{1}{256 \text{ min} * s}}{5 \frac{1}{s^2}} =$ $631,5 \frac{\text{mm}}{10} =$ $122880 \frac{1}{\text{min} * 256 s} =$ $6315 \frac{\text{mm}}{10 s^2}$	num: 8192 div: 421

Beispiele Berechnung Beschleunigungsfaktor					
Beschl.- einheiten ¹⁾	Vorsch.- konst. ²⁾	Zeitkonstante ³⁾	Getr. 4)	Formel ⁵⁾	Ergebnis gekürzt
U/min/s, 0 NK → U/min s	1 U _{AUS} = 1 U _{AUS}	$1 \frac{1}{\text{min} * s} =$ $256 \frac{1}{256 * s}$	1/1	$\frac{1 * \frac{256 \frac{1}{256 \text{ min} * s}}{1 \frac{1}{\text{min} * s}}}{\frac{1}{1}} =$ $256 \frac{1}{256 * s} =$ $1 \frac{1}{s}$	num: 256 div: 1
°/s ² , 1 NK → 1/10 °/s ² (°/10 s ²)	1 U _{AUS} = 3600 $\frac{°}{10}$	$1 \frac{1}{s^2} =$ $60 \frac{1}{\text{min} * s} =$ $60 * 256 \frac{1}{256 * s}$	1/1	$1 * \frac{60 * 256 \frac{1}{256 \text{ min} * s}}{1 \frac{1}{s^2}} =$ $\frac{3600 \frac{°}{10}}{1} =$ $15360 \frac{1}{\text{min} * 256 s} =$ $3600 \frac{°}{10 s^2}$	num: 64 div: 15
U/min ² , 2 NK → 1/100 U/min ² (U/100 min ²)	1 U _{AUS} = 100 $\frac{\text{U}}{100}$	$1 \frac{1}{\text{min}^2} =$ $\frac{1}{60} \frac{1}{s} =$ $\frac{256}{60} \frac{1}{256 * s}$	2/3	$\frac{2 * \frac{256 \frac{1}{256 \text{ min} * s}}{60 \frac{1}{\text{min}^2}}}{\frac{100 \frac{1}{100}}{1}} =$ $\frac{512}{256 s} =$ $18000 \frac{1}{100 \text{ min}^2}$	num: 32 div: 1125

1) Gewünschte Einheit am Abtrieb

2) Positionseinheiten pro Umdrehung am Abtrieb (U_{AUS}). Vorschubkonstante des Antriebs * 10^{-NK} (Nachkommastellen)

3) Zeitfaktor_v: Gewünschte Zeiteinheit pro interne Zeiteinheit

4) Getriebefaktor: U_{EIN} pro U_{AUS}

5) Werte in Formel einsetzen.

Beispiele Berechnung Beschleunigungsfaktor					
Beschl.- einheiten ¹⁾	Vorsch.- konst. ²⁾	Zeitkonstante ³⁾	Getr. 4)	Formel ⁵⁾	Ergebnis gekürzt
mm/s^2 , 1 NK $\rightarrow 1/10 \text{ mm/s}^2$ $(\text{mm}/10 \text{ s}^2)$	63,15 $\frac{\text{mm}}{\text{U}}$ \Rightarrow $1 \text{ U}_{\text{AUS}} =$ 631,5 $\frac{\text{mm}}{10}$	$1 \frac{1}{\text{s}^2} =$ $60 \frac{1}{\text{min} * \text{s}} =$ $60 * 256 \frac{1}{256 * \text{s}}$	4/5	$\frac{4}{5} * \frac{60 * 256 \frac{1}{256 \text{ min} * \text{s}}}{1 \frac{1}{\text{s}^2}} = \frac{122880 \frac{1}{256 \text{ s}}}{631,5 \frac{\text{mm}}{10}} = \frac{6315 \frac{\text{mm}}{10 \text{ s}^2}}$	num: 8192 div: 421

- 1) Gewünschte Einheit am Abtrieb
- 2) Positionseinheiten pro Umdrehung am Abtrieb (U_{AUS}). Vorschubkonstante des Antriebs * 10^{-NK} (Nachkommastellen)
- 3) Zeitfaktor_v: Gewünschte Zeiteinheit pro interne Zeiteinheit
- 4) Getriebefaktor: U_{Ein} pro U_{AUS}
- 5) Werte in Formel einsetzen.

Tab. 5.9 Beispiele Berechnung Beschleunigungsfaktor

Objekt 607E_h: polarity

Das Vorzeichen der Positions- und Geschwindigkeitswerte des Motorcontrollers kann mit dem entsprechenden polarity_flag eingestellt werden. Dieses kann dazu dienen, die Drehrichtung des Motors bei gleichen Sollwerten zu invertieren.

In den meisten Applikationen ist es sinnvoll, das velocity_polarity_flag und das position_polarity_flag auf den gleichen Wert zu setzen.

Das Setzen des polarity_flags beeinflusst nur Parameter beim Lesen und beim Schreiben. Bereits im Motorcontroller vorhandene Parameter werden nicht verändert.

Index	607E_h
Name	polarity
Object Code	VAR
Data Type	UINT8

Access	rw
PDO Mapping	yes
Units	–
Value Range	40 _h , 80 _h , C0 _h
Default Value	0

Bit	Wert	Name	Bedeutung
6	40 _h	velocity_polarity_flag	0: multiply by 1 (default) 1: multiply by -1 (invers)
7	80 _h	position_polarity_flag	0: multiply by 1 (default) 1: multiply by -1 (invers)

5.4 Endstufenparameter

Übersicht

Die Netzspannung wird über eine Vorladeschaltung in die Endstufe eingespeist. Beim Einschalten der Leistungsversorgung wird der Einschaltstrom begrenzt und das Laden überwacht. Nach erfolgter Vorladung des Zwischenkreises wird die Ladeschaltung überbrückt. Dieser Zustand ist Voraussetzung für das Erteilen der Reglerfreigabe. Die gleichgerichtete Netzspannung wird mit den Kondensatoren des Zwischenkreises geglättet. Aus dem Zwischenkreis wird der Motor über die IGBTs gespeist. Die Endstufe enthält eine Reihe von Sicherheitsfunktionen, die zum Teil parametrierbar sind:

- Reglerfreigabelogik (Software- und Hardwarefreigabe)
- Überstromüberwachung
- Überspannungs-/Unterspannungs-Überwachung des Zwischenkreises
- Leistungsteilüberwachung

Beschreibung der Objekte

Index	Objekt	Name	Typ	Attr.
6510 _h	RECORD	Drive_data		

Objekt 6510_h_10_h: enable_logic

Damit die Endstufe des Motorcontrollers aktiviert werden kann, müssen die digitalen Eingänge Endstufenfreigabe und Reglerfreigabe gesetzt sein: Die Endstufenfreigabe wirkt direkt auf die Ansteuersignale der Leistungstransistoren und würde diese auch bei einem defekten Mikroprozessor unterbrechen können. Das Wegnehmen der Endstufenfreigabe bei laufendem Motor bewirkt somit, dass der Motor ungebremst austrudelt bzw. nur durch die eventuell vorhandene Haltebremse gestoppt wird. Die Reglerfreigabe wird vom Mikrokontroller des Motorcontrollers verarbeitet. Je nach Betriebsart reagiert der Motorcontroller nach der Wegnahme dieses Signals unterschiedlich:

- Positionierbetrieb und drehzahl geregelter Betrieb
Der Motor wird nach der Wegnahme des Signals mit einer definierten Bremsrampe abgebremst. Die Endstufe wird erst abgeschaltet, wenn die Motordrehzahl unterhalb 10 min⁻¹ liegt und die eventuell vorhandene Haltebremse angezogen hat.
- Momentengeregelter Betrieb
Die Endstufe wird unmittelbar nach der Wegnahme des Signals abgeschaltet. Gleichzeitig wird eine eventuell vorhandene Haltebremse angezogen. Der Motor trudelt also ungebremst aus bzw. wird nur durch die eventuell vorhandene Haltebremse gestoppt.



Warnung

Lebensgefährliche Spannung!

Beide Signale garantieren nicht, dass der Motor spannungsfrei ist.

Beim Betrieb des Motorcontrollers über den CAN-Bus können die beiden digitalen Eingänge Endstufenfreigabe und Reglerfreigabe gemeinsam auf 24 V gelegt und die Freigabe über den CAN-Bus gesteuert werden. Dazu muss das Objekt 6510_h_10_h (enable_logic) auf zwei gesetzt werden. Aus Sicherheitsgründen erfolgt dies bei der Aktivierung von CANopen (auch nach einem Reset des Motorcontrollers) automatisch.

Index	6510_h
Name	drive_data
Object Code	RECORD
No. of Elements	51

Sub-Index	10_h
Description	enable_logic
Data Type	UINT16
Access	rw
PDO Mapping	no
Units	–
Value Range	0 ... 2
Default Value	0

Wert	Bedeutung
0	Digitale Eingänge Endstufenfreigabe + Reglerfreigabe
1	Digitale Eingänge Endstufenfreigabe + Reglerfreigabe + Parametrierschnittstelle
2	Digitale Eingänge Endstufenfreigabe + Reglerfreigabe + CAN

Objekt 6510_h_30_h: pwm_frequency

Die Schaltverluste der Endstufe sind proportional zur Schaltfrequenz der Leistungstransistoren. Bei den Geräten der CMMP-Familie kann durch Halbieren der normalen PWM-Frequenz mehr Leistung entnommen werden. Dadurch steigt allerdings die durch die Endstufe verursachte Stromwelligkeit. Die Umschaltung ist nur bei ausgeschalteter Endstufe möglich.

Sub-Index	30_h
Description	pwm_frequency
Data Type	UINT16
Access	rw
PDO Mapping	no
Units	–
Value Range	0, 1
Default Value	0

Wert	Bedeutung
0	Normale Endstufenfrequenz
1	Halbe Endstufenfrequenz

Objekt 6510_h_3A_h: enable_enhanced_modulation

Mit dem Objekt enable_enhanced_modulation kann die erweiterte Sinusmodulation aktiviert werden. Sie erlaubt eine bessere Ausnutzung der Zwischenkreisspannung und damit um ca. 14% höhere Drehzahlen. Nachteilig ist in bestimmten Applikationen, dass das Regelverhalten und der Rundlauf des Motors bei sehr kleinen Drehzahlen geringfügig schlechter werden. Der Schreibzugriff ist nur bei ausge-

schalteter Endstufe möglich. Um die Änderung zu übernehmen, muss der Parametersatz gesichert und ein Reset durchgeführt werden.

Sub-Index	3A_h
Description	enable_enhanced_modulation
Data Type	UINT16
Access	rw
PDO Mapping	no
Units	–
Value Range	0, 1
Default Value	0

Wert	Bedeutung
0	Erweiterte Sinusmodulation AUS
1	Erweiterte Sinusmodulation EIN



Die Aktivierung der erweiterten Sinusmodulation wird erst nach einem Reset wirksam. Der Parametersatz muss somit zunächst gespeichert (save_all_parameters) und anschließend ein Reset durchgeführt werden.

Objekt 6510_h 31_h: power_stage_temperature

Die Temperatur der Endstufe kann über das Objekt power_stage_temperature ausgelesen werden. Wenn die im Objekt 6510_h 32_h (max_power_stage_temperature) angegebene Temperatur überschritten wird, schaltet die Endstufe aus und eine Fehlermeldung wird abgesetzt.

Sub-Index	31_h
Description	power_stage_temperature
Data Type	INT16
Access	ro
PDO Mapping	yes
Units	°C
Value Range	–
Default Value	–

Objekt 6510_h_32_h: max_power_stage_temperature

Die Temperatur der Endstufe kann über das Objekt 6510_h_31_h (power_stage_temperature) ausgelesen werden. Wenn die im Objekt max_power_stage_temperature angegebene Temperatur überschritten wird, schaltet die Endstufe aus und eine Fehlermeldung wird abgesetzt.

Sub-Index	32_h
Description	max_power_stage_temperature
Data Type	INT16
Access	ro
PDO Mapping	no
Units	°C
Value Range	100
Default Value	geräteabhängig

Gerätetyp	Wert
CMMP-AS-C2-3A-M3/-M0	100 °C
CMMP-AS-C5-3A-M3/-M0	80 °C
CMMP-AS-C5-11A-P3-M3/-M0	80 °C
CMMP-AS-C10-11A-P3-M3/-M0	80 °C

Objekt 6510_h_33_h: nominal_dc_link_circuit_voltage

Über das Objekt nominal_dc_link_circuit_voltage kann die Gerätenennspannung in Millivolt ausgelesen werden.

Sub-Index	33_h
Description	nominal_dc_link_circuit_voltage
Data Type	UINT32
Access	ro
PDO Mapping	no
Units	mV
Value Range	–
Default Value	geräteabhängig

Gerätetyp	Wert
CMMP-AS-C2-3A-M3/-M0	360000
CMMP-AS-C5-3A-M3/-M0	360000
CMMP-AS-C5-11A-P3-M3/-M0	560000
CMMP-AS-C10-11A-P3-M3/-M0	560000

Objekt 6510_h 34_h: actual_dc_link_circuit_voltage

Über das Objekt actual_dc_link_circuit_voltage kann die aktuelle Spannung des Zwischenkreises in Millivolt ausgelesen werden.

Sub-Index	34_h
Description	actual_dc_link_circuit_voltage
Data Type	UINT32
Access	ro
PDO Mapping	yes
Units	mV
Value Range	–
Default Value	–

Objekt 6510_h 35_h: max_dc_link_circuit_voltage

Das Objekt max_dc_link_circuit_voltage gibt an, ab welcher Zwischenkreisspannung die Endstufe aus Sicherheitsgründen sofort ausgeschaltet und eine Fehlermeldung abgesetzt wird.

Sub-Index	35_h
Description	max_dc_link_circuit_voltage
Data Type	UINT32
Access	ro
PDO Mapping	no
Units	mV
Value Range	–
Default Value	geräteabhängig

Gerätetyp	Wert
CMMP-AS-C2-3A-M3/-M0	460000
CMMP-AS-C5-3A-M3/-M0	460000
CMMP-AS-C5-11A-P3-M3/-M0	800000
CMMP-AS-C10-11A-P3-M3/-M0	800000

Objekt 6510_h_36_h: min_dc_link_circuit_voltage

Der Motorcontroller verfügt über eine Unterspannungsüberwachung. Diese kann über das Objekt 6510_h_37_h (enable_dc_link_undervoltage_error) aktiviert werden. Das Objekt 6510_h_36_h (min_dc_link_circuit_voltage) gibt an, bis zu welcher unteren Zwischenkreisspannung der Motorcontroller arbeiten soll. Unterhalb dieser Spannung wird der Fehler E 02-0 ausgelöst, wenn dieses mit dem nachfolgenden Objekt aktiviert wurde.

Sub-Index	36_h
Description	min_dc_link_circuit_voltage
Data Type	UINT32
Access	rw
PDO Mapping	no
Units	mV
Value Range	0 ... 1000000
Default Value	0

Objekt 6510_h_37_h: enable_dc_link_undervoltage_error

Mit dem Objekt enable_dc_link_undervoltage_error kann die Unterspannungsüberwachung aktiviert werden. Im Objekt 6510_h_36_h (min_dc_link_circuit_voltage) ist anzugeben, bis zu welcher unteren Zwischenkreisspannung der Motorcontroller arbeiten soll.

Sub-Index	37_h
Description	enable_dc_link_undervoltage_error
Data Type	UINT16
Access	rw
PDO Mapping	no
Units	–
Value Range	0, 1
Default Value	0

Wert	Bedeutung
0	Unterspannungsfehler AUS (Reaktion WARNUNG)
1	Unterspannungsfehler EIN (Reaktion REGLERFREIGABE AUS)

Die Aktivierung des Fehlers 02-0 erfolgt durch Änderung der Fehlerreaktion. Reaktionen, die zum Stillsetzen des Antriebs führen, werden als EIN, alle anderen als AUS zurückgegeben. Beim Beschreiben mit 0 wird die Fehlerreaktion WARNUNG gesetzt, beim Beschreiben mit 1 die Fehlerreaktion REGLERFREIGABE AUS.

→ hierzu auch 5.18, Fehlermanagement.

Objekt 6510_h_40_h: nominal_current

Mit dem Objekt nominal_current kann der Gerätenennstrom ausgelesen werden. Es handelt sich gleichzeitig um den oberen Grenzwert, der in das Objekt 6075_h (motorRatedCurrent) eingeschrieben werden kann.

Sub-Index	40_h
Description	nominal_current
Data Type	UINT32
Access	ro
PDO Mapping	no
Units	mA
Value Range	–
Default Value	geräteabhängig

Gerätetyp	Wert
CMMP-AS-C2-3A-M3/-M0	2500
CMMP-AS-C5-3A-M3/-M0	5000
CMMP-AS-C5-11A-P3-M3/-M0	5000
CMMP-AS-C10-11A-P3-M3/-M0	10000



Aufgrund eines Leistungsderating werden abhängig von der Reglerzykluszeit und der Endstufentaktfrequenz gegebenenfalls andere Werte angezeigt.

Objekt 6510_h_41_h: peak_current

Mit dem Objekt peak_current, kann der Gerätespitzenstrom ausgelesen werden. Es handelt sich gleichzeitig um den oberen Grenzwert, der in das Objekt 6073_h (maxCurrent) eingeschrieben werden kann.

Sub-Index	41_h
Description	peak_current
Data Type	UINT32
Access	ro
PDO Mapping	no
Units	mA
Value Range	–
Default Value	geräteabhängig

Gerätetyp	Wert
CMMP-AS-C2-3A-M3/-M0	10000
CMMP-AS-C5-3A-M3/-M0	20000
CMMP-AS-C5-11A-P3-M3/-M0	20000
CMMP-AS-C10-11A-P3-M3/-M0	40000



Die Werte gelten für eine Stromregler-Zykluszeit von 125 µs.



Aufgrund eines Leistungsderating werden abhängig von der Reglerzykluszeit und der Endstufentaktfrequenz gegebenenfalls andere Werte angezeigt.

5.5 Stromregler und Motoranpassung



Vorsicht

Falsche Einstellungen der Stromreglerparameter und der Strombegrenzungen können den Motor und unter Umständen auch den Motorcontroller innerhalb kürzester Zeit zerstören!

Übersicht

Der Parametersatz des Motorcontrollers muss für den angeschlossenen Motor und den verwendeten Kabelsatz angepasst werden. Betroffen sind folgende Parameter:

Parameter	Abhängigkeiten
Nennstrom	Abhängig vom Motor
Überlastbarkeit	Abhängig vom Motor
Polzahl	Abhängig vom Motor
Stromregler	Abhängig vom Motor
Drehsinn	Abhängig vom Motor und der Phasenfolge im Motor- und Winkelgeberkabel
Offsetwinkel	Abhängig vom Motor und der Phasenfolge im Motor- und Winkelgeberkabel

Bitte beachten Sie, dass Drehsinn und Offsetwinkel auch vom verwendeten Kabelsatz abhängen. Die Parametersätze arbeiten daher nur bei identischer Verkabelung.



Vorsicht

Bei verdrehter Phasenfolge im Motor- oder Winkelgeberkabel kann es zu einer Mitkoppelung kommen, so dass die Drehzahl im Motor nicht geregelt werden kann. Der Motor kann unkontrolliert durchdrehen!

Beschreibung der Objekte

Index	Objekt	Name	Typ	Attr.
6075 _h	VAR	motorRatedCurrent	UINT32	rw
6073 _h	VAR	maxCurrent	UINT16	rw
604D _h	VAR	poleNumber	UINT8	rw
6410 _h	RECORD	motorData		rw
60F6 _h	RECORD	torqueControlParameters		rw

Betroffene Objekte aus anderen Kapiteln

Index	Objekt	Name	Typ	Kapitel
2415 _h	RECORD	currentLimitation		5.8 Sollwert-Begrenzung

Objekt 6075_h: motorRatedCurrent

Dieser Wert ist dem Motortypenschild zu entnehmen und wird in der Einheit Milliampere eingegeben. Es wird immer der Effektivwert (RMS) angenommen. Es kann kein Strom vorgegeben werden, der oberhalb des Motorcontroller-Nennstromes (6510_h_40_h: nominalCurrent) liegt.

Index	6075_h
Name	motorRatedCurrent
Object Code	VAR
Data Type	UINT32

Access	rw
PDO Mapping	yes
Units	mA
Value Range	0 ... nominalCurrent
Default Value	296



Wird das Objekt 6075_h (motorRatedCurrent) mit einem neuen Wert beschrieben, muss in jedem Fall auch das Objekt 6073_h (maxCurrent) neu parametrisiert werden.

Objekt 6073_h: maxCurrent

Servomotoren dürfen in der Regel für einen bestimmten Zeitraum überlastet werden. Mit diesem Objekt wird der höchstzulässige Motorstrom als Faktor eingestellt. Er bezieht sich auf den Motornennstrom (Objekt 6075_h: motorRatedCurrent) und wird in Tausendstel eingestellt. Der Wertebereich wird nach oben durch den maximalen Controllerstrom (Objekt 6510_h_41_h: peakCurrent) begrenzt. Viele Motoren dürfen kurzzeitig um den Faktor 4 überlastet werden. In diesem Fall ist in dieses Objekt der Wert 4000 einzuschreiben.



Das Objekt 6073_h (maxCurrent) darf erst beschrieben werden, wenn zuvor das Objekt 6075_h (motorRatedCurrent) gültig beschrieben wurde.

Index	6073_h
Name	max_current
Object Code	VAR
Data Type	UINT16

Access	rw
PDO Mapping	yes
Units	per thousands of rated current
Value Range	–
Default Value	2023

Objekt 604D_h: pole_number

Die Polzahl des Motors ist dem Motordatenblatt oder der Parametriersoftware zu entnehmen. Die Polzahl ist immer geradzahlig. Oft wird statt der Polzahl die Polpaarzahl angegeben. Die Polzahl entspricht dann der doppelten Polpaarzahl.

Dieses Objekt wird durch restore_default_parameters nicht geändert.

Index	604D_h
Name	pole_number
Object Code	VAR
Data Type	UINT8

Access	rw
PDO Mapping	yes
Units	–
Value Range	2 ... 254
Default Value	4 (nach INIT!)

Objekt 6410_h_03_h: iit_time_motor

Servomotoren dürfen in der Regel für einen bestimmten Zeitraum überlastet werden. Über dieses Objekt wird angegeben, wie lange der angeschlossene Motor mit dem im Objekt 6073_h (max_current) angegebenen Strom bestromt werden darf. Nach Ablauf der I²t-Zeit wird der Strom zum Schutz des Motors automatisch auf den im Objekt 6075_h (motor_rated_current) angegebenen Wert begrenzt. Die Standardeinstellung liegt bei zwei Sekunden und trifft für die meisten Motoren zu.

Index	6410_h
Name	motor_data
Object Code	RECORD
No. of Elements	5

Sub-Index	03_h
Description	iit_time_motor
Data Type	UINT16
Access	rw
PDO Mapping	no
Units	ms
Value Range	0 ... 100000
Default Value	2000

Objekt 6410_h.04_h: iit_ratio_motor

Über das Objekt kann iit_ratio_motor die aktuelle Auslastung der I²t-Begrenzung in Promille ausgelesen werden.

Sub-Index	04_h
Description	iit_ratio_motor
Data Type	UINT16
Access	ro
PDO Mapping	no
Units	promille
Value Range	–
Default Value	–

Objekt 6510_h.38_h: iit_error_enable

Über das Objekt iit_error_enable wird festgelegt, wie sich der Motorcontroller bei Auftreten der I²t-Begrenzung verhält. Entweder wird dieses nur im statusword angezeigt, oder es wird Fehler E 31-0 ausgelöst.

Index	6510_h
Name	drive_data
Object Code	RECORD
No. of Elements	51

Sub-Index	38_h
Description	ii_t_error_enable
Data Type	UINT16
Access	rw
PDO Mapping	no
Units	–
Value Range	0, 1
Default Value	0

Wert	Bedeutung	
0	I ² t-Fehler AUS	(Priorität WARNUNG)
1	I ² t-Fehler EIN	(Priorität REGLERFREIGABE AUS)

Die Aktivierung des Fehlers E 31-0 erfolgt durch Änderung der Fehlerreaktion. Reaktionen, die zum Stillsetzen des Antriebs führen, werden als EIN, alle anderen als AUS zurückgegeben. Beim Beschreiben mit 0 wird die Fehlerreaktion WARNUNG gesetzt, beim Beschreiben mit 1 die Fehlerreaktion REGLERFREIGABE AUS. → Kapitel 5.18, Fehlermanagement.

Objekt 6410_h_10_h: phase_order

In der Phasenfolge (phase_order) werden Verdrehungen zwischen Motorkabel und Winkelgeberkabel berücksichtigt. Sie kann der Parametriersoftware entnommen werden.

Sub-Index	10_h
Description	phase_order
Data Type	INT16
Access	rw
PDO Mapping	yes
Units	–
Value Range	0, 1
Default Value	0

Wert	Bedeutung
0	Rechts
1	Links

Objekt 6410_h_11_h: encoder_offset_angle

Bei den verwendeten Servomotoren befinden sich Dauermagnete auf dem Rotor. Diese erzeugen ein magnetisches Feld, dessen Ausrichtung zum Stator von der Rotorlage abhängt. Für die elektronische Kommutierung muss der Motorcontroller das elektromagnetische Feld des Stators immer im richtigen Winkel zu diesem Permanentmagnetfeld einstellen. Er bestimmt hierzu laufend mit einem Winkelgeber (Resolver etc.) die Rotorlage.

Die Orientierung des Winkelgebers zum Dauermagnetfeld muss in das Objekt encoder_offset_angle eingetragen werden. Mit der Parametriersoftware kann dieser Winkel bestimmt werden. Der mit der

Parametriersoftware bestimmte Winkel liegt im Bereich von $\pm 180^\circ$. Er muss folgendermaßen umgerechnet werden:

$$\text{encoder_offset_angle} = \text{Offsetwinkel des Winkelgebers} * \frac{32767}{180^\circ}$$

Dieses Objekt wird durch `restore_default_parameters` nicht geändert.

Index	6410_h
Name	motor_data
Object Code	RECORD
No. of Elements	5

Sub-Index	11_h
Description	encoder_offset_angle
Data Type	INT16
Access	rw
PDO Mapping	yes
Units	...
Value Range	-32767 ... 32767
Default Value	E000 _h (-45°) (nach Werkseinstellung)

Objekt 6410_h_14_h: motor_temperature_sensor_polarity

Über dieses Objekt kann festgelegt werden, ob ein Öffner oder ein Schließer als digitaler Motortemperatur-Sensor verwendet wird.

Sub-Index	14_h
Description	motor_temperatur_sensor_polarity
Data Type	INT16
Access	rw
PDO Mapping	yes
Units	–
Value Range	0, 1
Default Value	0

Wert	Bedeutung
0	Öffner
1	Schließer

Objekt 6510_h_2E_h: motor_temperature

Mit diesem Objekt kann die aktuelle Motortemperatur ausgelesen werden, falls ein analoger Temperatursensor angeschlossen ist. Anderenfalls ist das Objekt undefiniert.

Index	6510_h
Name	drive_data
Object Code	RECORD
No. of Elements	51

Sub-Index	2E_h
Description	motor_temperature
Data Type	INT16
Access	ro
PDO Mapping	yes
Units	°C
Value Range	–
Default Value	–

Objekt 6510_h_2F_h: max_motor_temperature

Wird die in diesem Objekt definierte Motortemperatur überschritten, erfolgt eine Reaktion gemäß Fehlermanagement (Fehler 03-0, Übertemperatur Motor analog). Ist eine Reaktion parametrierbar, die zum Stillsetzen des Antriebs führt, wird eine Emergency-Message gesendet.

Zur Parametrierung des Fehlermanagements → Kap. 5.18, Fehlermanagement.

Sub-Index	2F_h
Description	max_motor_temperature
Data Type	INT16
Access	rw
PDO Mapping	no
Units	°C
Value Range	20 ... 300
Default Value	100

Objekt 60F6_h: torque_control_parameters

Die Daten des Stromreglers müssen der Parametriersoftware entnommen werden. Hierbei sind folgende Umrechnungen zu beachten:

Die Verstärkung des Stromreglers muss mit 256 multipliziert werden. Bei einer Verstärkung von 1.5 im Menü „Stromregler“ der Parametriersoftware ist in das Objekt torque_control_gain der Wert 384 = 180_h einzuschreiben.

Die Zeitkonstante des Stromreglers ist in der Parametriersoftware in Millisekunden angegeben. Um diese Zeitkonstante in das Objekt torque_control_time übertragen zu können, muss sie zuvor in Mikrosekunden umgerechnet werden. Bei einer angegebenen Zeit von 0.6 Millisekunden ist entsprechend der Wert 600 in das Objekt torque_control_time einzutragen.

Index	60F6_h
Name	torque_control_parameters
Object Code	RECORD
No. of Elements	2

Sub-Index	01_h
Description	torque_control_gain
Data Type	UINT16
Access	rw
PDO Mapping	no
Units	256 = „1“
Value Range	0 ... 32*256
Default Value	3*256 (768)

Sub-Index	02_h
Description	torque_control_time
Data Type	UINT16
Access	rw
PDO Mapping	no
Units	μs
Value Range	104 ... 64401
Default Value	1020

5.6 Drehzahlregler

Übersicht

Der Parametersatz des Motorcontrollers muss für die Applikation angepasst werden. Besonders die Verstärkung ist stark abhängig von eventuell an den Motor angekoppelten Massen. Die Daten müssen bei der Inbetriebnahme der Anlage mit Hilfe der Parametriersoftware optimal bestimmt werden.



Vorsicht

Falsche Einstellungen der Drehzahlreglerparameter können zu starken Schwingungen führen und eventuell Teile der Anlage zerstören!

Beschreibung der Objekte

Index	Objekt	Name	Typ	Attr.
60F9 _h	RECORD	velocity_control_parameters		rw
2073 _h	VAR	velocity_display_filter_time	UINT32	rw

Objekt 60F9_h: velocity_control_parameters

Die Daten des Drehzahlreglers müssen der Parametriersoftware entnommen werden. Hierbei sind folgende Umrechnungen zu beachten:

Die Verstärkung des Drehzahlreglers muss mit 256 multipliziert werden.

Bei einer Verstärkung von 1.5 im Menü „Drehzahlregler“ der Parametriersoftware ist in das Objekt velocity_control_gain der Wert $384 = 180_{\text{h}}$ einzuschreiben.

Die Zeitkonstante des Drehzahlreglers ist in der Parametriersoftware in Millisekunden angegeben. Um diese Zeitkonstante in das Objekt velocity_control_time übertragen zu können, muss sie zuvor in Mikrosekunden umgerechnet werden. Bei einer angegebenen Zeit von 2.0 Millisekunden ist entsprechend der Wert 2000 in das Objekt velocity_control_time einzutragen.

Index	60F9_h
Name	velocity_control_parameter_set
Object Code	RECORD
No. of Elements	3

Sub-Index	01_h
Description	velocity_control_gain
Data Type	UINT16
Access	rw
PDO Mapping	no
Units	256 = Gain 1
Value Range	20 ... 64*256 (16384)
Default Value	256

Sub-Index	02_h
Description	velocity_control_time
Data Type	UINT16
Access	rw
PDO Mapping	no
Units	μs
Value Range	1 ... 32000
Default Value	2000

Sub-Index	04_h
Description	velocity_control_filter_time
Data Type	UINT16
Access	rw
PDO Mapping	no
Units	μs
Value Range	1 ... 32000
Default Value	400

Objekt 2073_h: velocity_display_filter_time

Mit dem Objekt velocity_display_filter_time kann die Filterzeit des Anzeigedrehzahl-Istwertfilters eingestellt werden.

Index	2073_h
Name	velocity_display_filter_time
Object Code	VAR
Data Type	UINT32

Access	rw
PDO Mapping	no
Units	µs
Value Range	1000 ... 50000
Default Value	20000



Bitte beachten Sie, dass das Objekt velocity_actual_value_filtered für den Durchdrehenschutz verwendet wird. Bei sehr großer Filterzeit wird ein Durchdrehfehler erst mit entsprechender Verzögerung erkannt.

5.7 Lageregler (Position Control Function)

Übersicht

In diesem Kapitel sind alle Parameter beschrieben, die für den Lageregler erforderlich sind. Am Eingang des Lagereglers liegt der Lage-Sollwert (position_demand_value) vom Fahrkurven-Generator an. Außerdem wird der Lage-Istwert (position_actual_value) vom Winkelgeber (Resolver, Inkrementalgeber etc.) zugeführt. Das Verhalten des Lagereglers kann durch Parameter beeinflusst werden. Um den Lageregelkreis stabil zu halten, ist eine Begrenzung der Ausgangsgröße (control_effort) möglich. Die Ausgangsgröße wird als Drehzahl-Sollwert dem Drehzahlregler zugeführt. Alle Ein- und Ausgangsgrößen des Lagereglers werden in der Factor Group von den applikationsspezifischen Einheiten in die jeweiligen internen Einheiten des Reglers umgerechnet.

Folgende Unterfunktionen sind in diesem Kapitel definiert:

1. Schleppfehler (Following Error)

Als Schleppfehler wird die Abweichung des Lage-Istwertes (position_actual_value) vom Lage-Sollwert (position_demand_value) bezeichnet. Wenn dieser Schleppfehler für einen bestimmten Zeitraum größer ist als im Schleppfehler-Fenster (following_error_window) angegeben, so wird das Bit 13 following_error im Objekt statusword gesetzt. Der zulässige Zeitraum kann über das Objekt following_error_time_out vorgegeben werden.

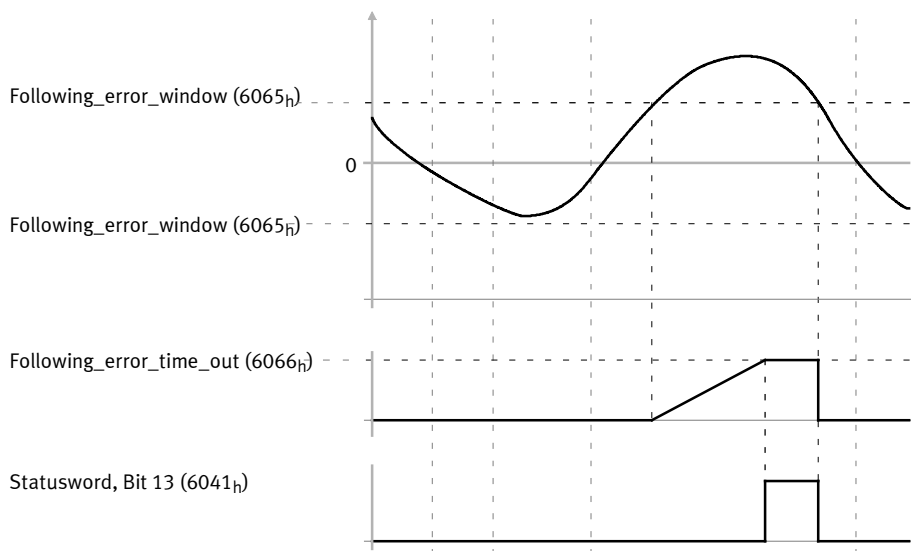


Fig. 5.6 Schleppfehler – Funktionsübersicht

2. Position erreicht (Position Reached)

Diese Funktion bietet die Möglichkeit, ein Positionsfenster um die Zielposition (target_position) herum zu definieren. Wenn sich die Ist-Position des Antriebs für eine bestimmte Zeit – die position_window_time – in diesem Bereich befindet, dann wird das damit verbundene Bit 10 (target_reached) im statusword gesetzt.

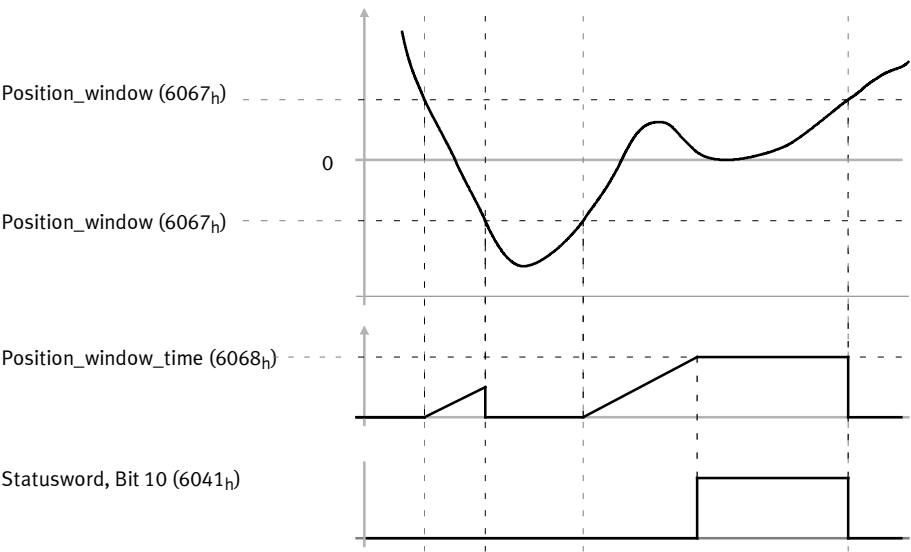


Fig. 5.7
Position erreicht – Funktionsübersicht

Beschreibung der Objekte

In diesem Kapitel behandelte Objekte

Index	Objekt	Name	Typ	Attr.
202D _h	VAR	position_demand_sync_value	INT32	ro
2030 _h	VAR	set_position_absolute	INT32	wo
6062 _h	VAR	position_demand_value	INT32	ro
6063 _h	VAR	position_actual_value_s ¹⁾	INT32	ro
6064 _h	VAR	position_actual_value	INT32	ro
6065 _h	VAR	following_error_window	UINT32	rw
6066 _h	VAR	following_error_time_out	UINT16	rw
6067 _h	VAR	position_window	UINT32	rw
6068 _h	VAR	position_window_time	UINT16	rw
607B _h	ARRAY	position_range_limit	INT32	rw
60F4 _h	VAR	following_error_actual_value	INT32	ro
60FA _h	VAR	control_effort	INT32	ro
60FB _h	RECORD	position_control_parameter_set		rw
6510 _h _20 _h	VAR	position_range_limit_enable	UINT16	rw
6510 _h _22 _h	VAR	position_error_switch_off_limit	UINT32	rw

1) In Inkrementen

Betroffene Objekte aus anderen Kapiteln

Index	Objekt	Name	Typ	Kapitel
607A _h	VAR	target_position	INT32	7.3 Betriebsart Positionieren
607C _h	VAR	home_offset	INT32	7.2 Referenzfahrt
607D _h	VAR	software_position_limit	INT32	7.3 Betriebsart Positionieren
607E _h	VAR	polarity	UINT8	5.3 Umrechnungsfaktoren
6093 _h	VAR	position_factor	UINT32	5.3 Umrechnungsfaktoren
6094 _h	ARRAY	velocity_encoder_factor	UINT32	5.3 Umrechnungsfaktoren
6096 _h	ARRAY	acceleration_factor	UINT32	5.3 Umrechnungsfaktoren
6040 _h	VAR	controlword	INT16	6.1.3 Controlword (Steuerwort)
6041 _h	VAR	statusword	UINT16	6.1.5 Statuswords (Statusworte)

Objekt 60FB_h: position_control_parameter_set

Der Parametersatz des Motorcontrollers muss für die Applikation angepasst werden. Die Daten des Lagereglers müssen bei der Inbetriebnahme der Anlage mit Hilfe der Parametriersoftware optimal bestimmt werden.

**Vorsicht**

Falsche Einstellungen der Lagereglerparameter können zu starken Schwingungen führen und eventuell Teile der Anlage zerstören!

Der Lageregler vergleicht die Soll-Lage mit der Ist-Lage und bildet aus der Differenz unter Berücksichtigung der Verstärkung und eventuell des Integrators eine Korrekturgeschwindigkeit (Objekt 60FA_h: control_effort), die dem Drehzahlregler zugeführt wird.

Der Lageregler ist, gemessen am Strom- und Drehzahlregler, relativ langsam. Der Regler arbeitet daher intern mit Aufschaltungen, so dass die Ausregelarbeit für den Lageregler minimiert wird und der Regler schnell einschwingen kann.

Als Lageregler genügt normalerweise ein Proportional-Glied. Die Verstärkung des Lagereglers muss mit 256 multipliziert werden. Bei einer Verstärkung von 1.5 im Menü „Lageregler“ der Parametriersoftware ist in das Objekt position_control_gain der Wert 384 einzuschreiben.

Normalerweise kommt der Lageregler ohne Integrator aus. Dann ist in das Objekt position_control_time der Wert Null einzuschreiben. Andernfalls muss die Zeitkonstante des Lagereglers in Mikrosekunden umgerechnet werden. Bei einer Zeit von 4.0 Millisekunden ist entsprechend der Wert 4000 in das Objekt position_control_time einzutragen.

Da der Lageregler schon kleinste Lageabweichungen in nennenswerte Korrekturgeschwindigkeiten umsetzt, würde es im Falle einer kurzen Störung (z. B. kurzzeitiges Klemmen der Anlage) zu sehr heftigen Ausregelvorgängen mit sehr großen Korrekturgeschwindigkeiten kommen. Dieses ist zu vermeiden, wenn der Ausgang des Lagereglers über das Objekt position_control_v_max sinnvoll (z. B. 500 min⁻¹) begrenzt wird.

Mit dem Objekt position_error_tolerance_window kann die Größe einer Lageabweichung definiert werden, bis zu der der Lageregler nicht eingreift (Totbereich). Dieses kann zur Stabilisierung eingesetzt werden, wenn z. B. Spiel in der Anlage vorhanden ist.

Index	60FB_h
Name	position_control_parameter_set
Object Code	RECORD
No. of Elements	4

Sub-Index	01_h
Description	position_control_gain
Data Type	UINT16
Access	rw
PDO Mapping	no
Units	256 = „1“
Value Range	0 ... 64*256 (16384)
Default Value	102

Sub-Index	02_h
Description	position_control_time
Data Type	UINT16
Access	ro
PDO Mapping	no
Units	μs
Value Range	0
Default Value	0

Sub-Index	04_h
Description	position_control_v_max
Data Type	UINT32
Access	rw
PDO Mapping	no
Units	speed units
Value Range	0 ... 131072 min ⁻¹
Default Value	500 min ⁻¹

Sub-Index	05_h
Description	position_error_tolerance_window
Data Type	UINT32
Access	rw
PDO Mapping	no
Units	position units
Value Range	1 ... 65536 (1 U)
Default Value	2 (1/32768 U)

Objekt 6062_h: position_demand_value

Über dieses Objekt kann der aktuelle Lage-Sollwert ausgelesen werden. Diese wird vom Fahrkurven-Generator in den Lageregler eingespeist.

Index	6062_h
Name	position_demand_value
Object Code	VAR
No. of Elements	INT32

Access	ro
PDO Mapping	yes
Units	position units
Value Range	–
Default Value	–

Objekt 202D_h: position_demand_sync_value

Über dieses Objekt kann die Soll-Lage des Synchronisationsgeber ausgelesen werden. Diese wird durch das Objekt 2022_h synchronization_encoder_select (➔ Kap. 5.11) definiert. Dieses Objekt wird in benutzerdefinierten Einheiten angegeben.

Index	202D_h
Name	position_demand_sync_value
Object Code	VAR
Data Type	INT32

Access	ro
PDO Mapping	no
Units	position units
Value Range	–
Default Value	–

Objekt 6063_h: position_actual_value_s (Inkremente)

Über dieses Objekt kann die Ist-Lage ausgelesen werden. Diese wird dem Lageregler vom Winkelgeber aus zugeführt. Dieses Objekt wird in Inkrementen angegeben.

Index	6063_h
Name	position_actual_value_s
Object Code	VAR
Data Type	INT32

Access	ro
PDO Mapping	yes
Units	inkrements
Value Range	–
Default Value	–

Objekt 6064_h: position_actual_value (benutzerdefinierte Einheiten)

Über dieses Objekt kann die Ist-Lage ausgelesen werden. Diese wird dem Lageregler vom Winkelgeber aus zugeführt. Dieses Objekt wird in benutzerdefinierten Einheiten angegeben.

Index	6064_h
Name	position_actual_value
Object Code	VAR
Data Type	INT32

Access	ro
PDO Mapping	yes
Units	position units
Value Range	–
Default Value	–

Objekt 6065_h: following_error_window

Das Objekt following_error_window (Schleppfehler-Fenster) definiert um den Lage-Sollwert (position_demand_value) einen symmetrischen Bereich. Wenn sich der Lage-Istwert (position_actual_value) außerhalb des Schleppfehler-Fensters (following_error_window) befindet, dann tritt ein Schleppfehler auf und das Bit 13 im Objekt statusword wird gesetzt. Folgende Ursachen können einen Schleppfehler verursachen:

- der Antrieb ist blockiert
- die Positioniergeschwindigkeit ist zu groß
- die Beschleunigungswerte sind zu groß
- das Objekt following_error_window ist mit einem zu kleinen Wert besetzt
- der Lageregler ist nicht richtig parametrisiert

Index	6065_h
Name	following_error_window
Object Code	VAR
Data Type	UINT32

Access	rw
PDO Mapping	yes
Units	position units
Value Range	–
Default Value	9101 (9101/65536 U = 50°)

Objekt 6066_h: following_error_time_out

Tritt ein Schleppfehler – länger als in diesem Objekt definiert – auf, dann wird das zugehörige Bit 13 following_error im statusword gesetzt.

Index	6066_h
Name	following_error_time_out
Object Code	VAR
Data Type	UINT16

Access	rw
PDO Mapping	yes
Units	ms
Value Range	0 ... 27314
Default Value	0

Objekt 60F4_h: following_error_actual_value

Über dieses Objekt kann der aktuelle Schleppfehler ausgelesen werden. Dieses Objekt wird in benutzerdefinierten Einheiten angegeben.

Index	60F4_h
Name	following_error_actual_value
Object Code	VAR
Data Type	INT32

Access	ro
PDO Mapping	yes
Units	position units
Value Range	–
Default Value	–

Objekt 60FA_h: control_effort

Die Ausgangsgröße des Lagereglers kann über dieses Objekt ausgelesen werden. Dieser Wert wird intern dem Drehzahlregler als Sollwert zugeführt.

Index	60FA_h
Name	control_effort
Object Code	VAR
Data Type	INT32

Access	ro
PDO Mapping	yes
Units	speed units
Value Range	–
Default Value	–

Objekt 6067_h: position_window

Mit dem Objekt **position_window** wird um die Zielposition (target_position) herum ein symmetrischer Bereich definiert. Wenn der Lage-Istwert (position_actual_value) eine bestimmte Zeit innerhalb dieses Bereiches liegt, wird die Zielposition (target_position) als erreicht angesehen.

Index	6067_h
Name	position_window
Object Code	VAR
Data Type	UINT32

Access	rw
PDO Mapping	yes
Units	position units
Value Range	–
Default Value	1820 (1820/65536 U = 10°)

Objekt 6068_h: position_window_time

Wenn sich die Ist-Position des Antriebes innerhalb des Positionierfensters (position_window) befindet und zwar solange, wie in diesem Objekt definiert, dann wird das zugehörige Bit 10 target_reached im statusword gesetzt.

Index	6068_h
Name	position_window_time
Object Code	VAR
Data Type	UINT16

Access	rw
PDO Mapping	yes
Units	ms
Value Range	–
Default Value	0

Objekt 6510_h. 22_h: position_error_switch_off_limit

Im Objekt position_error_switch_off_limit kann die maximal zulässige Abweichung zwischen der Soll- und der Istposition eingetragen werden. Im Gegensatz zur o. g. Schleppfehlermeldung wird bei einer Überschreitung die Endstufe sofort abgeschaltet und ein Fehler ausgelöst. Der Motor trudelt somit ungebremst aus (außer es ist eine Haltebremse vorhanden).

Index	6510_h
Name	drive_data
Object Code	RECORD
No. of Elements	51

Sub-Index	22_h
Description	position_error_switch_off_limit
Data Type	UINT32
Access	rw
PDO Mapping	no
Units	position units
Value Range	0 ... $2^{32}-1$
Default Value	0

Wert	Bedeutung
0	Grenzwert Schleppfehler AUS (Reaktion: KEINE AKTION)
> 0	Grenzwert Schleppfehler EIN (Reaktion: ENDSTUFE SOFORT ABSCHALTEN)

Die Aktivierung des Fehlers 17-0 erfolgt durch Änderung der Fehlerreaktion. Die Reaktion ENDSTUFE SOFORT ABSCHALTEN wird als EIN, alle anderen als AUS zurückgegeben. Beim Beschreiben mit 0 wird

die Fehlerreaktion KEINE AKTION gesetzt, beim Beschreiben mit einem Wert größer 0 die Fehlerreaktion ENDSTUFE SOFORT ABSCHALTEN. → Kapitel 5.18 Fehlermanagement.

Objekt 607B_h: position_range_limit

Die Objektgruppe position_range_limit enthält zwei Unterparameter, die den numerischen Bereich der Positionswerte beschränken. Wenn eine dieser Grenzen überschritten wird, springt der Positionswert automatisch an die jeweils andere Grenze. Dieses ermöglicht die Parametrierung von sog. Rundachsen. Anzugeben sind die Grenzen, die physikalisch der gleichen Position entsprechen sollen, also beispielsweise 0° und 360°.

Damit diese Grenzen wirksam werden, muss über das Objekt 6510_h_20_h (position_range_limit_enable) ein Rundachsmodus ausgewählt werden.

Index	607B_h
Name	position_range_limit
Object Code	ARRAY
No. of Elements	2
Data Type	INT32

Sub-Index	01_h
Description	min_position_range_limit
Access	rw
PDO Mapping	yes
Units	position units
Value Range	–
Default Value	–

Sub-Index	02_h
Description	max_position_range_limit
Access	rw
PDO Mapping	yes
Units	position units
Value Range	–
Default Value	–

Objekt 6510_h_20_h: position_range_limit_enable

Über das Objekt position_range_limit_enable können die durch das Objekt 607B_h definierten Bereichsgrenzen aktiviert werden. Es sind verschiedene Modi möglich:

Wird der Modus „Kürzester Weg“ gewählt, werden Positionierungen immer auf der physikalisch kürzeren Strecke zum Ziel ausgeführt. Der Antrieb passt dazu selber das Vorzeichen der Fahrgeschwindigkeit an. Bei den beiden Modi „Feste Drehrichtung“ erfolgt die Positionierung grundsätzlich nur in die im Modus angegebene Richtung.

Index	6510_h
Name	drive_data
Object Code	RECORD
No. of Elements	51

Sub-Index	20_h
Description	position_range_limit_enable
Data Type	UINT16
Access	rw
PDO Mapping	no
Units	–
Value Range	0 ... 5
Default Value	0

Wert	Bedeutung
0	Aus
1	Kürzester Weg (aus Kompatibilitätsgründen)
2	Kürzester Weg
3	Reserviert
4	Feste Drehrichtung „Positiv“
5	Feste Drehrichtung „Negativ“

Objekt 2030_h: set_position_absolute

Über das Objekt set_position_absolute kann die auslesbare Istposition verschoben werden, ohne dass sich die physikalische Lage ändert. Der Antrieb führt dabei keine Bewegung aus.

Wenn ein absolutes Gebersystem angeschlossen ist, wird die Lageverschiebung im Geber gespeichert, sofern das Gebersystem dies zulässt. Die Lageverschiebung bleibt in diesem Fall also nach einem Reset erhalten. Diese Speicheroperation läuft unabhängig von diesem Objekt im Hintergrund ab. Es werden dabei ebenfalls alle dem Geberspeicher zugehörigen Parameter mit ihren aktuellen Werten gespeichert.

Index	2030_h
Name	set_position_absolute
Object Code	VAR
Data Type	INT32

Access	wo
PDO Mapping	no
Units	position units
Value Range	–
Default Value	–

5.8 Sollwert-Begrenzung

Beschreibung der Objekte

In diesem Kapitel behandelte Objekte

Index	Objekt	Name	Typ	Attr.
2415 _h	RECORD	current_limitation		rw
2416 _h	RECORD	speed_limitation		rw

Objekt 2415_h: current_limitation

Mit der Objektgruppe current_limitation kann in den Betriebsarten profile_position_mode, interpolated_position_mode, homing_mode und velocity_mode der Maximalstrom für den Motor begrenzt werden, wodurch z. B. ein drehmomentbegrenzter Drehzahlbetrieb ermöglicht wird. Über das Objekt limit_current_input_channel wird die Sollwert-Quelle des Begrenzungsmoment vorgegeben. Hier kann zwischen der Vorgabe eines direkten Sollwerts (fester Wert) oder der Vorgabe über einen analogen Eingang gewählt werden. Über das Objekt limit_current wird je nach gewählter Quelle entweder das Begrenzungsmoment (Quelle = fester Wert) oder der Skalierungsfaktor für die Analogeingänge (Quelle = Analogeingang) vorgegeben. Im ersten Fall wird direkt auf den momentproportionalen Strom in mA begrenzt, im zweiten Fall wird der Strom in mA angegeben, der einer anliegenden Spannung von 10 V entsprechen soll.

Index	2415_h
Name	current_limitation
Object Code	RECORD
No. of Elements	2

Sub-Index	01_h
Description	limit_current_input_channel
Data Type	UINT8
Access	rw
PDO Mapping	no
Units	–
Value Range	0 ... 4
Default Value	0

Sub-Index	02_h
Description	limit_current
Data Type	INT32
Access	rw
PDO Mapping	no
Units	mA
Value Range	–
Default Value	0

Wert	Bedeutung
0	Keine Begrenzung
1	AIN0
2	AIN1
3	AIN2
4	Feldbus (Selektor B)

Objekt 2416_h: speed_limitation

Mit der Objektgruppe speed_limitation kann in der Betriebsart profile_torque_mode die Maximaldrehzahl des Motors begrenzt werden, wodurch ein drehzahlbegrenzter Drehmomentbetrieb ermöglicht wird. Über das Objekt limit_speed_input_channel wird die Sollwert-Quelle der Begrenzungsdrehzahl vorgegeben. Hier kann zwischen der Vorgabe eines direkten Sollwerts (fester Wert) oder der Vorgabe über einen analogen Eingang gewählt werden. Über das Objekt limit_speed wird je nach gewählter Quelle entweder die Begrenzungsdrehzahl (fester Wert) oder der Skalierungsfaktor für die Analogeingänge (Quelle = Analogeingang) vorgegeben. Im ersten Fall wird direkt auf die angegebene Drehzahl begrenzt, im zweiten Fall wird die Drehzahl angegeben, der einer anliegenden Spannung von 10 V entsprechen soll.

Index	2416_h
Name	speed_limitation
Object Code	RECORD
No. of Elements	2

Sub-Index	01_h
Description	limit_speed_input_channel
Data Type	UINT8
Access	rw
PDO Mapping	no
Units	–
Value Range	0 ... 4
Default Value	0

Sub-Index	02_h
Description	limit_speed
Data Type	INT32
Access	rw
PDO Mapping	no
Units	speed units
Value Range	–
Default Value	–

Wert	Bedeutung
0	Keine Begrenzung
1	AIN0
2	AIN1
3	AIN2
4	Feldbus (Selektor B)

5.9 Geberanpassungen

Übersicht

Dieses Kapitel beschreibt die Konfiguration des Winkelgebereingangs [X2A], [X2B] und des Inkrementaleingangs [X10].



Vorsicht

Falsche Winkelgeber-Einstellungen können den Antrieb unkontrolliert drehen lassen und eventuell Teile der Anlage zerstören.

Beschreibung der Objekte

In diesem Kapitel behandelte Objekte

Index	Objekt	Name	Typ	Attr.
2024 _h	RECORD	encoder_x2a_data_field		ro
2024 _h _01 _h	VAR	encoder_x2a_resolution	UINT32	ro
2024 _h _02 _h	VAR	encoder_x2a_numerator	INT16	rw
2024 _h _03 _h	VAR	encoder_x2a_divisor	INT16	rw
2025 _h	RECORD	encoder_x10_data_field		ro
2025 _h _01 _h	VAR	encoder_x10_resolution	UINT32	rw
2025 _h _02 _h	VAR	encoder_x10_numerator	INT16	rw
2025 _h _03 _h	VAR	encoder_x10_divisor	INT16	rw
2025 _h _04 _h	VAR	encoder_x10_counter	UINT32	ro
2026 _h	RECORD	encoder_x2b_data_field		ro
2026 _h _01 _h	VAR	encoder_x2b_resolution	UINT32	rw
2026 _h _02 _h	VAR	encoder_x2b_numerator	INT16	rw
2026 _h _03 _h	VAR	encoder_x2b_divisor	INT16	rw
2026 _h _04 _h	VAR	encoder_x2b_counter	UINT32	ro

Objekt 2024_h: encoder_x2a_data_field

Im Record encoder_x2a_data_field sind Parameter zusammengefasst, die für den Betrieb des Winkelgebers am Stecker [X2A] notwendig sind.

Da zahlreiche Winkelgeber-Einstellungen nur nach einem Reset wirksam werden, sollten die Auswahl und die Einstellung der Geber über die Parametriersoftware erfolgen. Unter CANopen lassen sich folgende Einstellungen auslesen bzw. ändern:

Das Objekt encoder_x2a_resolution gibt an, wie viele Inkremente vom Geber pro Umdrehung oder Längeneinheit erzeugt werden. Da am Eingang [X2A] nur Resolver angeschlossen werden können, die immer mit 16 Bit ausgewertet werden, wird hier immer 65536 zurückgegeben. Mit dem Objekt encoder_x2a_numerator und encoder_x2a_divisor kann ein eventuelles Getriebe (auch mit Vorzeichen) zwischen Motorwelle und Geber berücksichtigt werden.

Index	2024_h
Name	encoder_x2a_data_field
Object Code	RECORD
No. of Elements	3

Sub-Index	01_h
Description	encoder_x2a_resolution
Data Type	UINT32
Access	ro
PDO Mapping	no
Units	Inkmente (4 * Strichzahl)
Value Range	–
Default Value	65536

Sub-Index	02_h
Description	encoder_x2a_numerator
Data Type	INT16
Access	rw
PDO Mapping	no
Units	–
Value Range	–32768 ... 32767 (außer 0)
Default Value	1

Sub-Index	03_h
Description	encoder_x2a_divisor
Data Type	INT16
Access	rw
PDO Mapping	no
Units	–
Value Range	1 ... 32767
Default Value	1

Objekt 2026_h: encoder_x2b_data_field

Im Record encoder_x2b_data_field sind Parameter zusammengefasst, die für den Betrieb des Winkelgebers am Stecker [X2B] notwendig sind.

Das Objekt encoder_x2b_resolution gibt an, wie viele Inkmente vom Geber pro Umdrehung erzeugt werden (Bei Inkrementalgebern entspricht dies dem vierfachen der Strichzahl bzw der Perioden pro Umdrehung).

Das Objekt encoder_x2b_counter liefert die aktuell gezählte Inkmentzahl. Es liefert daher Werte zwischen 0 und der eingestellten Inkmentzahl-1.

Mit den Objekten encoder_x2b_numerator und encoder_x2b_divisor kann ein Getriebe zwischen Motorwelle und dem an [X2B] angeschlossenen Geber berücksichtigt werden.

Index	2026_h
Name	encoder_x2b_data_field
Object Code	RECORD
No. of Elements	4

Sub-Index	01_h
Description	encoder_x2b_resolution
Data Type	UINT32
Access	rw
PDO Mapping	no
Units	Inkremente (4 * Strichzahl)
Value Range	abhängig vom benutzten Geber
Default Value	abhängig vom benutzten Geber

Sub-Index	02_h
Description	encoder_x2b_numerator
Data Type	INT16
Access	rw
PDO Mapping	no
Units	–
Value Range	–32768 ... 32767
Default Value	1

Sub-Index	03_h
Description	encoder_x2b_divisor
Data Type	INT16
Access	rw
PDO Mapping	no
Units	–
Value Range	1 ... 32767
Default Value	1

Sub-Index	04_h
Description	encoder_x2b_counter
Data Type	UINT32
Access	ro
PDO Mapping	yes
Units	Inkremente (4 * Strichzahl)
Value Range	0 ... (encoder_x2b_resolution -1)
Default Value	–

Objekt 2025_h: encoder_x10_data_field

Im Record encoder_X10_data_field sind Parameter zusammengefasst, die für den Betrieb des Inkrementaleingangs [X10] notwendig sind. Hier kann wahlweise ein digitaler Inkrementalgeber oder emulierte Inkrementalsignale beispielsweise eines anderen CMMP angeschlossen werden. Die Eingangssignale über [X10] können wahlweise als Sollwert oder als Istwert verwendet werden. Näheres hierzu finden Sie in Kapitel 5.11.

Im Objekt encoder_X10_resolution muss angegeben werden, wie viele Inkremente vom Geber pro Umdrehung des Gebers erzeugt werden. Dies entspricht dem vierfachen der Strichzahl. Das Objekt encoder_X10_counter liefert die aktuell gezählte Inkrementzahl (Zwischen 0 und der eingestellten Inkrementzahl-1).

Mit dem Objekt encoder_X10_numerator und encoder_X10_divisor kann ein eventuelles Getriebe (auch mit Vorzeichen) berücksichtigt werden.

Bei der Verwendung des X10-Signals als Istwert entspräche dies einem Getriebe zwischen dem Motor und dem an [X10] angeschlossenen Istwertgeber, welches am Abtrieb montiert ist. Bei der Verwendung des X10-Signals als Sollwert, können hiermit Getriebeübersetzungen zwischen Master und Slave realisiert werden.

Index	2025_h
Name	encoder_x10_data_field
Object Code	RECORD
No. of Elements	4

Sub-Index	01_h
Description	encoder_x10_resolution
Data Type	UINT32
Access	rw
PDO Mapping	no
Units	Inkremente (4 * Strichzahl)
Value Range	abhängig vom benutzten Geber
Default Value	abhängig vom benutzten Geber

Sub-Index	02_h
Description	encoder_x10_numerator
Data Type	INT16
Access	rw
PDO Mapping	no
Units	–
Value Range	–32768 ... 32767 (außer 0)
Default Value	1

Sub-Index	03_h
Description	encoder_x10_divisor
Data Type	INT16
Access	rw
PDO Mapping	no
Units	–
Value Range	1 ... 32767
Default Value	1

Sub-Index	04_h
Description	encoder_x10_counter
Data Type	UINT32
Access	ro
PDO Mapping	yes
Units	Inkremente (4 * Strichzahl)
Value Range	0 ... (encoder_x10_resolution - 1)
Default Value	–

5.10 Inkrementalgeberemulation

Übersicht

Diese Objekt-Gruppe ermöglicht es, den Inkrementalgeberausgang [X11] zu parametrieren. Somit können Master-Slave-Applikationen, bei denen der Ausgang des Masters [X11] an den Eingang des Slave [X10] angeschlossen ist, hiermit unter CANopen parametrieren werden.

Beschreibung der Objekte

In diesem Kapitel behandelte Objekte

Index	Objekt	Name	Typ	Attr.
2028 _h	VAR	encoder_emulation_resolution	INT32	rw
201A _h	RECORD	encoder_emulation_data		ro
201A _h _01 _h	VAR	encoder_emulation_resolution	INT32	rw
201A _h _02 _h	VAR	encoder_emulation_offset	INT16	rw

Objekt 201A_h: encoder_emulation_data

Der Object-Record encoder_emulation_data kapselt alle Einstellmöglichkeiten für den Inkrementalgeberausgang [X11]:

Über das Objekt encoder_emulation_resolution kann die ausgegebene Inkrementzahl (= vierfache Strichzahl) als Vielfaches von 4 frei eingestellt werden. In einer Master-Slave-Applikation muss diese der encoder_X10_resolution des Slave entsprechen, um ein Verhältnis von 1:1 zu erreichen.

Mit dem Objekt encoder_emulation_offset kann die Position des ausgegebenen Nullimpulses gegenüber der Nulllage des Istwertgebers verschoben werden.

Index	201A_h
Name	encoder_emulation_data
Object Code	RECORD
No. of Elements	2

Sub-Index	01_h
Description	encoder_emulation_resolution
Data Type	INT32
Access	rw
PDO Mapping	no
Units	(4 * Strichzahl)
Value Range	4 * (1 ... 8192)
Default Value	4096

Sub-Index	02_h
Description	encoder_emulation_offset
Data Type	INT16
Access	rw
PDO Mapping	no
Units	32767 = 180°
Value Range	-32768 ... 32767
Default Value	0

Objekt 2028_h: encoder_emulation_resolution

Das Objekt encoder_emulation_resolution ist nur aus Kompatibilitätsgründen vorhanden. Es entspricht dem Objekt 201A_h-01_h.

Index	2028_h
Name	encoder_emulation_resolution
Object Code	VAR
Data Type	INT32

Access	rw
PDO Mapping	no
Units	→ 201A _h -01 _h
Value Range	→ 201A _h -01 _h
Default Value	→ 201A _h -01 _h

5.11 Soll-/Istwertaufschaltung

Übersicht

Mit Hilfe der nachfolgenden Objekte kann die Quelle für den Sollwert und die Quelle für den Istwert geändert werden. Als Standard verwendet der Motorcontroller den Eingang für den Motorgeber [X2A] bzw. [X2B] als Istwert für den Lageregler. Bei Verwendung eines externen Lagegebers, z. B. hinter einem Getriebe, kann der über [X10] eingespeiste Lagewert als Istwert für den Lageregler aufgeschaltet werden. Darüber hinaus ist es möglich über [X10] eingehende Signale (z. B. eines zweiten Controllers) als zusätzlichen Sollwert aufzuschalten, wodurch Synchronbetriebsarten ermöglicht werden.

Beschreibung der Objekte

In diesem Kapitel behandelte Objekte

Index	Objekt	Name	Typ	Attr.
2021 _h	VAR	position_encoder_selection	INT16	rw
2022 _h	VAR	synchronisation_encoder_selection	INT16	rw
2023 _h	VAR	synchronisation_filter_time	UINT32	rw
202F _h	RECORD	synchronisation_selector_data		ro
202F _h _07 _h	VAR	synchronisation_main	UINT16	rw

Objekt 2021_h: position_encoder_selection

Das Objekt **position_encoder_selection** gibt den Gebereingang an, der zur Bestimmung der Istlage (Istwertgeber) verwendet wird. Dieser Wert kann geändert werden, um auf Lageregelung über einen externen (am Abtrieb angeschlossenen) Geber umzuschalten. Dabei kann zwischen [X10] und dem als Kommutiergeber ausgewählten Gebereingang ([X2A]/[X2B]) umgeschaltet werden. Wird einer der Gebereingänge [X2A]/[X2B] als Lageistwertgeber ausgewählt, so muss derjenige verwendet werden, der als Kommutiergeber genutzt wird. Wird der jeweils andere Geber angewählt, wird automatisch auf den Kommutiergeber umgeschaltet.

Index	2021 _h
Name	position_encoder_selection
Object Code	VAR
Data Type	INT16

Access	rw
PDO Mapping	no
Units	–
Value Range	0 ... 2 (→ Tabelle)
Default Value	0

Wert	Bezeichnung
0	[X2A]
1	[X2B]
2	[X10]



Es kann nur zwischen dem Gebereingang [X10] und dem jeweiligen Kommutiergeber [X2A] oder [X2B] als Lageistwertgeber gewählt werden. Die Konfiguration [X2A] als Kommutiergeber und [X2B] als Lageistwertgeber zu nutzen, bzw. umgekehrt, ist nicht möglich.

Objekt 2022_h: synchronisation_encoder_selection

Das Objekt synchronisation_encoder_selection gibt den Gebereingang an, der als Synchronisationsollwert verwendet wird. Je nach Betriebsart entspricht dieses einem Lagesollwert (Profile Position Mode) oder einem Drehzahlsollwert (Profile Velocity Mode).

Als Synchronisationseingang kann nur [X10] verwendet werden. Somit kann zwischen [X10] und keinem Eingang ausgewählt werden. Als Synchronisationsollwert sollte nicht der gleiche Eingang wie für den Istwertgeber gewählt werden.

Index	2022 _h
Name	synchronisation_encoder_selection
Object Code	VAR
Data Type	INT16

Access	rw
PDO Mapping	no
Units	–
Value Range	-1, 2 (→ Tabelle)
Default Value	2

Wert	Bezeichnung
-1	kein Geber / undefiniert
2	[X10]

Objekt 202F_h: synchronisation_selector_data

Über das Objekt synchronisation_main kann die Aufschaltung eines Synchronsollwerts erfolgen. Damit der Synchronsollwert überhaupt berechnet wird, muss Bit 0 gesetzt werden. Bit 1 ermöglicht es die Synchronlage erst durch das Starten eines Positionssatzes aufzuschalten. Zur Zeit ist nur 0 parametrierbar, so dass die Synchronlage immer zugeschaltet ist. Über das Bit 8 kann festgelegt werden, dass die Referenzfahrt ohne Aufschaltung der Synchronlage erfolgen soll, um Master und Slave getrennt referenzieren zu können.

Index	202F_h
Name	synchronisation_selector_data
Object Code	RECORD
No. of Elements	1

Sub-Index	07_h
Description	synchronisation_main
Data Type	UINT16
Access	rw
PDO Mapping	no
Units	–
Value Range	➔ Tabelle
Default Value	–

Bit	Wert	Bedeutung
0	0001 _h	0: Synchronisation inaktiv 1: Synchronisation aktiv
1	0002 _h	„fliegende Säge“ nicht möglich
8	0100 _h	0: Synchronisation während der Referenzfahrt 1: Keine Synchronisation während der Referenzfahrt

Objekt 2023_h: synchronisation_filter_time

Über das Objekt synchronisation_filter_time wird die Filterzeitkonstante eines PT1-Filters festgelegt, mit dem die Synchronisationsdrehzahl geglättet wird. Dies kann insbesondere bei geringen Strichzahlen nötig sein, da hier bereits kleine Änderungen des Eingangswertes hohen Drehzahlen entsprechend. Andererseits ist der Antrieb bei hohen Filterzeiten ggf. nicht mehr in der Lage schnell genug einem dynamischen Eingangssignal zu folgen.

Index	2023_h
Name	synchronisation_filter_time
Object Code	VAR
Data Type	UINT32

Access	rw
PDO Mapping	no
Units	µs
Value Range	10 ... 50000
Default Value	600

5.12 Analoge Eingänge

Übersicht

Die Motorcontroller der Reihe CMMP-AS-...-M3/-M0 verfügen über drei analoge Eingänge, über die dem Motorcontroller beispielsweise Sollwerte vorgegeben werden können. Für alle diese analogen Eingänge bieten die nachfolgenden Objekte die Möglichkeit, die aktuelle Eingangsspannung auszulesen (analog_input_voltage) und einen Offset einzustellen (analog_input_offset).

Beschreibung der Objekte

Index	Objekt	Name	Typ	Attr.
2400 _h	ARRAY	analog_input_voltage	INT16	ro
2401 _h	ARRAY	analog_input_offset	INT32	rw

2400_h: analog_input_voltage (Eingangsspannung)

Die Objektgruppe analog_input_voltage liefert die aktuelle Eingangsspannung des jeweiligen Kanals unter Berücksichtigung des Offsets in Millivolt.

Index	2400_h
Name	analog_input_voltage
Object Code	ARRAY
No. of Elements	3
Data Type	INT16

Sub-Index	01_h
Description	analog_input_voltage_ch_0
Access	ro
PDO Mapping	no
Units	mV
Value Range	–
Default Value	–

Sub-Index	02_h
Description	analog_input_voltage_ch_1
Access	ro
PDO Mapping	no
Units	mV
Value Range	–
Default Value	–

Sub-Index	03_h
Description	analog_input_voltage_ch_2
Access	ro
PDO Mapping	no
Units	mV
Value Range	–
Default Value	–

Objekt 2401_h: analog_input_offset (Offset Analogeingänge)

Über die Objektgruppe **analog_input_offset** kann die Offsetspannung in Millivolt für die jeweiligen Eingänge gesetzt bzw. gelesen werden. Mit Hilfe des Offsets kann eine eventuelle anliegende Gleichspannung ausgeglichen werden. Ein positiver Offset kompensiert dabei eine positive Eingangsspannung.

Index	2401_h
Name	analog_input_offset
Object Code	ARRAY
No. of Elements	3
Data Type	INT32

Sub-Index	01_h
Description	analog_input_offset_ch_0
Access	rw
PDO Mapping	no
Units	mV
Value Range	–10000 ... 10000
Default Value	0

Sub-Index	02_h
Description	analog_input_offset_ch_1
Access	rw
PDO Mapping	no
Units	mV
Value Range	–10000 ... 10000
Default Value	0

Sub-Index	03_h
Description	analog_input_offset_ch_2
Access	rw
PDO Mapping	no
Units	mV
Value Range	–10000 ... 10000
Default Value	0

5.13 Digitale Ein- und Ausgänge

Übersicht

Alle digitalen Eingänge des Motorcontrollers können über den CAN-Bus gelesen und fast alle digitalen Ausgänge können beliebig gesetzt werden. Zudem können den digitalen Ausgängen des Motorcontrollers Statusmeldungen zugeordnet werden.

Beschreibung der Objekte

In diesem Kapitel behandelte Objekte

Index	Objekt	Name	Typ	Attr.
60FD _h	VAR	digital_inputs	UINT32	ro
60FE _h	ARRAY	digital_outputs	UINT32	rw
2420 _h	RECORD	digital_output_state_mapping		ro
2420 _h _01 _h	VAR	dig_out_state_mapp_dout_1	UINT8	rw
2420 _h _02 _h	VAR	dig_out_state_mapp_dout_2	UINT8	rw
2420 _h _03 _h	VAR	dig_out_state_mapp_dout_3	UINT8	rw

Objekt 60FD_h: digital_inputs

Über das Objekt 60FD_h können die digitalen Eingänge ausgelesen werden:

Index	60Fd_h
Name	digital_inputs
Object Code	VAR
Data Type	UINT32

Access	ro
PDO Mapping	yes
Units	–
Value Range	gemäß folgender Tabelle
Default Value	0

Bit	Wert	Bedeutung
0	00000001 _h	Negativer Endschalter
1	00000002 _h	Positiver Endschalter
2	00000004 _h	Referenzschalter
3	00000008 _h	Interlock - (Regler- oder Endstufenfreigabe fehlt)
16 ... 23	00FF0000 _h	Digitale Eingänge des CAMC-D-8E8A
24 ... 27	0F000000 _h	DIN0 ... DIN3
28	10000000 _h	DIN8
29	20000000 _h	DIN9

Objekt 60FE_h: digital_outputs

Über das Objekt 60FE_h können die digitalen Ausgänge angesteuert werden. Hierzu ist im Objekt digital_outputs_mask anzugeben, welche der digitalen Ausgänge angesteuert werden sollen. Über das Objekt digital_outputs_data können die ausgewählten Ausgänge dann beliebig gesetzt werden. Es ist zu beachten, dass bei der Ansteuerung der digitalen Ausgänge eine Verzögerung von bis zu 10 ms auftreten kann. Wann die Ausgänge wirklich gesetzt werden, kann durch Zurücklesen des Objekts 60FE_h festgestellt werden.

Index	60FE_h
Name	digital_outputs
Object Code	ARRAY
No. of Elements	2
Data Type	UINT32

Sub-Index	01_h
Description	digital_outputs_data
Access	rw
PDO Mapping	yes
Units	–
Value Range	–
Default Value	(abhängig vom Zustand der Bremse)

Sub-Index	02_h
Description	digital_outputs_mask
Access	rw
PDO Mapping	yes
Units	–
Value Range	–
Default Value	00000000 _h

Bit	Wert	Bedeutung
0	00000001 _h	1 = Bremse anziehen
16 ... 23	0E000000 _h	Digitale Ausgänge des CAMC-D-8E8A
25 ... 27	0E000000 _h	DOUT1 ... DOUT3

**Vorsicht**

Wenn die Bremsensteuerung über `digital_output_mask` freigegeben ist, wird durch Löschen von Bit 0 in `digital_output_data` die Haltebremse manuell gelüftet! Dies kann bei hängenden Achsen zu einem Absacken der Achse führen.

Objekt 2420_h: digital_output_state_mapping

Über die Objektgruppe `digital_outputs_state_mapping` können verschiedene Statusmeldungen des Motorcontrollers über die digitalen Ausgänge ausgegeben werden.

Für die integrierten digitalen Ausgänge des Motorcontrollers ist hierzu für jeden Ausgang ein eigener Subindex vorhanden. Somit ist für jeden Ausgang ein Byte vorhanden, in das die Funktionsnummer einzutragen ist.

Wenn einem digitalen Ausgang eine derartige Funktion zugeordnet wurde und der Ausgang dann direkt über `digital_outputs` (60FE_h) ein- oder ausgeschaltet wird, wird auch das Objekt `digital_outputs_state_mapping` auf AUS (0) bzw. EIN (12) gesetzt.

Index	2420_h
Name	digital_outputs_state_mapping
Object Code	RECORD
No. of Elements	5

Sub-Index	01_h
Description	dig_out_state_mapp_dout_1
Data Type	UINT8
Access	rw
PDO Mapping	no
Units	–
Value Range	0 ... 44, → Tabelle
Default Value	0

Sub-Index	02_h
Description	dig_out_state_mapp_dout_2
Data Type	UINT8
Access	rw
PDO Mapping	no
Units	–
Value Range	0 ... 44, → Tabelle
Default Value	0

Sub-Index	03_h
Description	dig_out_state_mapp_dout_3
Data Type	UINT8
Access	rw
PDO Mapping	no
Units	–
Value Range	0 ... 44, → Tabelle
Default Value	0

Wert	Bezeichnung
0	Aus (Ausgang ist Low)
1	Position $X_{soll} = X_{ziel}$
2	Position $X_{ist} = X_{ziel}$
3	Reserviert
4	Restwegtrigger aktiv
5	Referenzfahrt aktiv
6	Vergleichsdrehzahl erreicht
7	I ² t-Motor erreicht
8	Schleppfehler
9	Unterspannung Zwischenkreis
10	Feststellbremse gelöst
11	Endstufe eingeschaltet
12	Ein (Ausgang ist High)
13	Sammelfehler aktiv
14	Mindestens eine Sollwertsperrung aktiv
15	Linearmotor identifiziert
16	Referenzposition gültig
17	Sammelstatus: Bereit zur Reglerfreigabe
18	Positionstrigger 1
19	Positionstrigger 2
20	Positionstrigger 3
21	Positionstrigger 4
22 ... 25	Reserviert
26	Alternatives Ziel erreicht

Wert	Bezeichnung
27	Aktiv wenn Positionssatz läuft
28	Vergleichsmoment erreicht
29	Position $x_{soll} = x_{ziel}$ (auch bei Verkettung für mindestens 10 ms)
30	Ack-Signal (activ low) als Handshake zu Start Positionieren
31	Ziel erreicht mit Handshake zum dig. Start, wird nicht gesetzt, solange START auf HIGH-Pegel ist.
32	Kurvenscheibe aktiv
33	CAM-IN-Bewegung läuft
34	CAM-CHANGE, wie CAM-IN aber Wechsel zu einer neuen Kurve
35	CAM-OUT-Bewegung läuft
36	Pegel digitale Endstufenfreigabe, also Pegel an DIN4 (High, wenn DIN4 High)
37	Reserviert
38	CAM aktiv ohne CAM-IN oder CAM-CHANGE Bewegung
39	Geschwindigkeitswert im Fenster für Stillstand
40	Teach Acknowledge
41	Speichervorgang (SAVE!, Save Positions) läuft
42	STO aktiv
43	STO ist angefordert
44	Motion Complete (MC)

Sub-Index	11_h
Description	dig_out_state_mapp_ea88_0_low
Data Type	UINT32
Access	rw
PDO Mapping	no
Units	–
Value Range	0 ... FFFFFFFF _h , → Tabelle
Default Value	0

Bit	Maske	Name	Bezeichnung
0 ... 7	000000FF _h	EA88_0_dout_0_mapping	Funktion für CAMC-D-8E8A 0 DOUT1
8 ... 15	0000FF00 _h	EA88_0_dout_1_mapping	Funktion für CAMC-D-8E8A 0 DOUT2
16 ... 23	00FF0000 _h	EA88_0_dout_2_mapping	Funktion für CAMC-D-8E8A 0 DOUT3
24 ... 31	FF000000 _h	EA88_0_dout_3_mapping	Funktion für CAMC-D-8E8A 0 DOUT4

Sub-Index	12_h
Description	dig_out_state_mapp_ea88_0_high
Data Type	UINT32
Access	rw
PDO Mapping	no
Units	–
Value Range	0 ... FFFFFFFF _h , ➔ Tabelle
Default Value	0

Bit	Maske	Name	Bezeichnung
0 ... 7	000000FF _h	EA88_0_dout_4_mapping	Funktion für CAMC-D-8E8A 0 DOUT5
8 ... 15	0000FF00 _h	EA88_0_dout_5_mapping	Funktion für CAMC-D-8E8A 0 DOUT6
16 ... 23	00FF0000 _h	EA88_0_dout_6_mapping	Funktion für CAMC-D-8E8A 0 DOUT7
24 ... 31	FF000000 _h	EA88_0_dout_7_mapping	Funktion für CAMC-D-8E8A 0 DOUT8

5.14 **Endschalter/Referenzschalter**

Übersicht

Für die Definition der Referenzposition des Motorcontrollers können wahlweise Endschalter (limit switch) oder Referenzschalter (homing switch) verwendet werden. Nähere Informationen zu den möglichen Referenzfahrt-Methoden finden sie im Kapitel 7.2, Betriebsart Referenzfahrt (Homing Mode).

Beschreibung der Objekte

Index	Objekt	Name	Typ	Attr.
6510 _h	RECORD	drive_data		rw

Objekt 6510_h_11h: limit_switch_polarity

Die Polarität der Endschalter kann durch das Objekt 6510_h_11_h (limit_switch_polarity) programmiert werden. Für öffnende Endschalter ist in dieses Objekt eine „0“, bei der Verwendung von schließenden Kontakten ist eine „1“ einzutragen.

Index	6510_h
Name	drive_data
Object Code	RECORD
No. of Elements	51

Sub-Index	11_h
Description	limit_switch_polarity
Data Type	INT16
Access	rw
PDO Mapping	no
Units	–
Value Range	0, 1
Default Value	1

Wert	Bedeutung
0	Öffner
1	Schließer

Objekt 6510_h_12_h: limit_switch_selector

Über das Objekt 6510_h_12_h (limit_switch_selector) kann die Zuordnung der Endschalter (negativ, positiv) vertauscht werden, ohne Änderungen an der Verkabelung vornehmen zu müssen. Um die Zuordnung der Endschalter zu tauschen, ist eine Eins einzutragen.

Sub-Index	12_h
Description	limit_switch_selector
Data Type	INT16
Access	rw
PDO Mapping	no
Units	–
Value Range	0, 1
Default Value	0

Wert	Bedeutung
0	DIN6 = E0 (Endschalter negativ) DIN7 = E1 (Endschalter positiv)
1	DIN6 = E1 (Endschalter positiv) DIN7 = E0 (Endschalter negativ)

Objekt 6510_h_14_h: homing_switch_polarity

Die Polarität des Referenzschalters kann durch das Objekt 6510_h_14_h (homing_switch_polarity) programmiert werden. Für einen öffnenden Referenzschalter ist in dieses Objekt eine Null, bei der Verwendung von schließenden Kontakten ist eine „1“ einzutragen.

Sub-Index	14_h
Description	homing_switch_polarity
Data Type	INT16
Access	rw
PDO Mapping	no
Units	–
Value Range	0, 1
Default Value	1

Wert	Bedeutung
0	Öffner
1	Schließer

Objekt 6510_h_13_h: homing_switch_selector

Das Objekt 6510_h_13_h (homing_switch_selector) legt fest, ob DIN8 oder DIN9 als Referenzschalter verwendet werden soll.

Sub-Index	13_h
Description	homing_switch_selector
Data Type	INT16
Access	rw
PDO Mapping	no
Units	–
Value Range	0, 1
Default Value	0

Wert	Bedeutung
0	DIN9
1	DIN8

Objekt 6510_h_15_h: limit_switch_deceleration

Das Objekt limit_switch_deceleration legt die Beschleunigung fest, mit der gebremst wird, wenn während des normalen Betriebs der Endschalter erreicht wird (Endschalter-Nothalt-Rampe).

Sub-Index	15_h
Description	limit_switch_deceleration
Data Type	INT32
Access	rw
PDO Mapping	no
Units	acceleration units
Value Range	0 ... 3000000 min ⁻¹ /s
Default Value	2000000 min ⁻¹ /s

5.15 Sampling von Positionen

Übersicht

Die CMMP Familie bietet die Möglichkeit den Lageistwert auf der steigenden oder fallenden Flanke eines digitalen Eingangs hin abzuspeichern. Dieser Lagewert kann dann z. B. zur Berechnung innerhalb einer Steuerung ausgelesen werden.

Alle notwendigen Objekte sind in dem Record `sample_data` zusammengefasst: Das Objekt `sample_mode` legt die Art des Samplings fest: Soll nur ein einmaliges Sample-Ereignis aufgezeichnet werden oder soll kontinuierlich gesampelt werden. Über das Objekt `sample_status` kann die Steuerung abfragen, ob ein Sample-Ereignis aufgetreten ist. Dies wird durch ein gesetztes Bit signalisiert, welches ebenfalls im `statusword` angezeigt werden kann, wenn das Objekt `sample_status_mask` entsprechend gesetzt ist.

Das Objekt `sample_control` dient dazu, die Freigabe des Sample-Ereignisses zu steuern und letztlich können über die Objekte `sample_position_rising_edge` und `sample_position_falling_edge` die gesampelten Positionen ausgelesen werden.

Welcher digitale Eingang verwendet wird, lässt sich mit der Parametriersoftware unter Controller – E/A Konfiguration – Digitale Eingänge – Sample-Eingang festlegen.

Beschreibung der Objekte

In diesem Kapitel behandelte Objekte

Index	Objekt	Name	Typ	Attr.
204A _h	RECORD	<code>sample_data</code>		ro
204A _h _01 _h	VAR	<code>sample_mode</code>	UINT16	rw
204A _h _02 _h	VAR	<code>sample_status</code>	UINT8	ro
204A _h _03 _h	VAR	<code>sample_status_mask</code>	UINT8	rw
204A _h _04 _h	VAR	<code>sample_control</code>	UINT8	wo
204A _h _05 _h	VAR	<code>sample_position_rising_edge</code>	INT32	ro
204A _h _06 _h	VAR	<code>sample_position_falling_edge</code>	INT32	ro

Objekt 204A_h: `sample_data`

Index	204A_h
Name	<code>sample_data</code>
Object Code	RECORD
No. of Elements	6

Mit dem folgenden Objekt kann gewählt werden, ob auf jedes Auftreten eines Sample-Events die Position bestimmt werden soll (Kontinuierliches Sampling) oder ob das Sampling nach einem Sample-Ereignis gesperrt werden soll, bis das Sampling erneut freigegeben wird. Beachten Sie hierbei, dass auch bereits ein Prellen beide Flanken auslösen kann!

Sub-Index	01_h
Description	sample_mode
Data Type	UINT16
Access	rw
PDO Mapping	no
Units	–
Value Range	0 ... 1, → Tabelle
Default Value	0

Wert	Bezeichnung
0	Kontinuierliches Sampling
1	Autolock sampling

Das folgende Objekt zeigt ein neues Sample-Ereignis an.

Sub-Index	02_h
Description	sample_status
Data Type	UINT8
Access	ro
PDO Mapping	yes
Units	–
Value Range	0 ... 3, → Tabelle
Default Value	0

Bit	Wert	Name	Beschreibung
0	01 _h	falling_edge_occurred	= 1: Neue Sample-Position (fallende Flanke)
1	02 _h	rising_edge_occurred	= 1: Neue Sample-Position (steigende Flanke)

Mit dem folgenden Objekt können die Bits des Objekts sample_status festgelegt werden, die auch zum Setzen von Bit 15 des statusword führen sollen. Dadurch ist im üblicherweise ohnehin zu übertragenden statusword die Information „Sample-Ereignis aufgetreten“ vorhanden, so dass die Steuerung nur in diesem Fall das Objekt sample_status lesen muss, um ggf. festzustellen welche Flanke aufgetreten ist.

Sub-Index	03_h
Description	sample_status_mask
Data Type	UINT8
Access	rw
PDO Mapping	yes
Units	–
Value Range	0 ... 1, → Tabelle
Default Value	0

Bit	Wert	Name	Beschreibung
0	01 _h	rising_edge_visible	Wenn rising_edge_occured = 1 → Statuswort Bit 15 = 1
1	02 _h	falling_edge_visible	Wenn falling_edge_occured = 1 → Statuswort Bit 15 = 1

Das Setzen des jeweiligen Bits in sample_control setzt zum einen das entsprechende Statusbit in sample_status zurück und schaltet im Falle des „Autolock“-Samplings das Sampling wieder frei.

Sub-Index	04_h
Description	sample_control
Data Type	UINT8
Access	wo
PDO Mapping	yes
Units	–
Value Range	0 ... 1, → Tabelle
Default Value	0

Bit	Wert	Name	Beschreibung
0	01 _h	falling_edge_enable	Sampling bei fallender Flanke
1	02 _h	rising_edge_enable	Sampling bei steigender Flanke

Die folgenden Objekte enthalten die gesampelten Positionen.

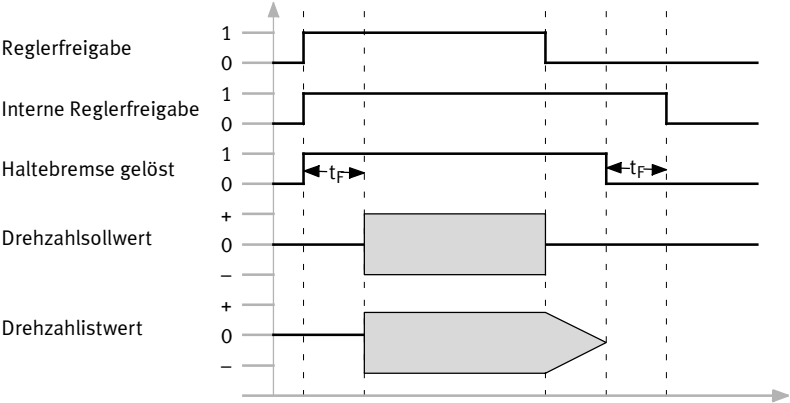
Sub-Index	05_h
Description	sample_position_rising_edge
Data Type	INT32
Access	ro
PDO Mapping	yes
Units	position units
Value Range	–
Default Value	–

Sub-Index	06_h
Description	sample_position_falling_edge
Data Type	INT32
Access	ro
PDO Mapping	yes
Units	position units
Value Range	–
Default Value	–

5.16 Bremsen-Ansteuerung

Übersicht

Mittels der nachfolgenden Objekte kann parametrisiert werden, wie der Motorcontroller eine eventuell im Motor integrierte Haltebremse ansteuert. Die Haltebremse wird immer freigeschaltet, sobald die Reglerfreigabe eingeschaltet wird. Für Haltebremsen mit hoher mechanischer Trägheit kann eine Verzögerungszeit parametrisiert werden, damit die Haltebremse in Eingriff ist, bevor die Endstufe ausgeschaltet wird (Durchsacken vertikaler Achsen). Diese Verzögerung wird durch das Objekt `brake_delay_time` parametrisiert. Wie aus der Skizze zu entnehmen ist, wird bei Einschalten der Reglerfreigabe der Drehzahl-Sollwert erst nach der `brake_delay_time` freigegeben und bei Ausschalten der Reglerfreigabe das Abschalten der Regelung um diese Zeit verzögert.



t_F : Fahrbeginnverzögerung

Fig. 5.8 Funktion der Bremsverzögerung (bei Drehzahlregelung / Positionieren)

Beschreibung der Objekte

Index	Objekt	Name	Typ	Attr.
6510 _h	RECORD	drive_data		rw

Objekt 6510_h_18_h: `brake_delay_time`

Über das Objekt `brake_delay_time` kann die Bremsverzögerungszeit parametrisiert werden.

Index	6510 _h
Name	drive_data
Object Code	RECORD
No. of Elements	51

Sub-Index	18_h
Description	brake_delay_time
Data Type	UINT16
Access	rw
PDO Mapping	no
Units	ms
Value Range	0 ... 32000
Default Value	0

5.17 Geräteinformationen

Index	Objekt	Name	Typ	Attr.
1018 _h	RECORD	identity_object		rw
6510 _h	RECORD	drive_data		rw

Über zahlreiche CAN-Objekte können die verschiedensten Informationen wie Motorcontrollertyp, verwendete Firmware, etc. aus dem Gerät ausgelesen werden.

Beschreibung der Objekte

Objekt 1018_h: identity_object

Über das in der CIA 301 festgelegte identity_object kann der Motorcontroller in einem CANopen-Netzwerk eindeutig identifiziert werden. Zu diesem Zweck kann der Herstellercode (vendor_id), ein eindeutiger Produktcode (product_code), die Revisionsnummer der CANopen-Implementation (revision_number) und die Seriennummer des Geräts (serial_number) ausgelesen werden.

Index	1018_h
Name	identity_object
Object Code	RECORD
No. of Elements	4

Sub-Index	01_h
Description	vendor_id
Data Type	UINT32
Access	ro
PDO Mapping	no
Units	–
Value Range	0000001D
Default Value	0000001D

Sub-Index	02_h
Description	product_code
Data Type	UINT32
Access	ro
PDO Mapping	no
Units	–
Value Range	s. u.
Default Value	s. u.

Wert	Bedeutung
2045 _h	CMMP-AS-C2-3A-M3
2046 _h	CMMP-AS-C5-3A-M3
204A _h	CMMP-AS-C5-11A-P3-M3
204B _h	CMMP-AS-C10-11A-P3-M3
2085 _h	CMMP-AS-C2-3A-M0
2086 _h	CMMP-AS-C5-3A-M0
208A _h	CMMP-AS-C5-11A-P3-M0
208B _h	CMMP-AS-C10-11A-P3-M0

Sub-Index	03_h
Description	revision_number
Data Type	UINT32
Access	ro
PDO Mapping	no
Units	MMMMSSSS _h (M: main version, S: sub version)
Value Range	–
Default Value	–

Sub-Index	04_h
Description	serial_number
Data Type	UINT32
Access	ro
PDO Mapping	no
Units	–
Value Range	–
Default Value	–

Objekt 6510_h_A0_h: drive_serial_number

Über das Objekt drive_serial_number kann die Seriennummer des Reglers gelesen werden. Dieses Objekt dient der Kompatibilität zu früheren Versionen.

Index	6510_h
Name	drive_data
Object Code	RECORD
No. of Elements	51

Sub-Index	A0_h
Description	drive_serial_number
Data Type	UINT32
Access	ro
PDO Mapping	no
Units	–
Value Range	–
Default Value	–

Objekt 6510_h_A1_h: drive_type

Über das Objekt drive_type kann der Gerätetyp des Reglers ausgelesen werden. Dieses Objekt dient der Kompatibilität zu früheren Versionen.

Sub-Index	A1_h
Description	drive_type
Data Type	UINT32
Access	ro
PDO Mapping	no
Units	–
Value Range	→ 1018 _h _02 _h , product_code
Default Value	→ 1018 _h _02 _h , product_code

Objekt 6510_h_A9_h: firmware_main_version

Über das Objekt firmware_main_version kann die Hauptversionsnummer der Firmware (Produktstufe) ausgelesen werden.

Sub-Index	A9_h
Description	firmware_main_version
Data Type	UINT32
Access	ro
PDO Mapping	no
Units	MMMMSSSS _h (M: main version, S: sub version)
Value Range	–
Default Value	–

Objekt 6510_h_AA_h: firmware_custom_version

Über das Objekt firmware_custom_version kann die Versionsnummer der kundenspezifischen Variante der Firmware ausgelesen werden.

Sub-Index	AA_h
Description	firmware_custom_version
Data Type	UINT32
Access	ro
PDO Mapping	no
Units	MMMMSSSS _h (M: main version, S: sub version)
Value Range	–
Default Value	–

Objekt 6510_h_AD_h: km_release

Über die Versionsnummer des km_release können Firmwarestände der gleichen Produktstufe unterschieden werden.

Sub-Index	AD_h
Description	km_release
Data Type	UINT32
Access	ro
PDO Mapping	no
Units	–
Value Range	MMMMSSSS _h (M: main version, S: sub version)
Default Value	–

Objekt 6510_h_AC_h: firmware_type

Über das Objekt firmware_type kann ausgelesen werden, für welche Gerätefamilie und für welchen Winkelgebertyp die geladene Firmware geeignet ist.

Sub-Index	AC_h
Description	firmware_type
Data Type	UINT32
Access	ro
PDO Mapping	no
Units	–
Value Range	00000F2 _h
Default Value	00000F2 _h

Objekt 6510_h_B0_h: cycletime_current_controller

Über das Objekt cycletime_current_controller kann die Zykluszeit des Stromreglers in Mikrosekunden ausgelesen werden.

Sub-Index	B0_h
Description	cycletime_current_controller
Data Type	UINT32
Access	ro
PDO Mapping	no
Units	µs
Value Range	–
Default Value	0000007D _h

Objekt 6510_h_B1_h: cycletime_velocity_controller

Über das Objekt cycletime_velocity_controller kann die Zykluszeit des Drehzahlreglers in Mikrosekunden ausgelesen werden.

Sub-Index	B1_h
Description	cycletime_velocity_controller
Data Type	UINT32
Access	ro
PDO Mapping	no
Units	µs
Value Range	–
Default Value	000000FA _h

Objekt 6510_h_B2_h: cycletime_position_controller

Über das Objekt cycletime_position_controller kann die Zykluszeit des Lagereglers in Mikrosekunden ausgelesen werden.

Sub-Index	B2_h
Description	cycletime_position_controller
Data Type	UINT32
Access	ro
PDO Mapping	no
Units	µs
Value Range	–
Default Value	000001F4 _h

Objekt 6510_h_B3_h: cycletime_trajectory_generator

Über das Objekt cycletime_trajectory_generator kann die Zykluszeit der Positionier-Steuerung in Mikroskunden ausgelesen werden.

Sub-Index	B3_h
Description	cycletime_tracectory_generator
Data Type	UINT32
Access	ro
PDO Mapping	no
Units	µs
Value Range	–
Default Value	000003E8 _h

Objekt 6510_h_C0_h: commissioning_state

Das Objekt commissioning_state wird von der Parametriersoftware beschrieben, wenn bestimmte Parametrierungen durchgeführt worden sind (z. B. des Nennstroms). Nach der Auslieferung und nach restore_default_parameter enthält dieses Objekt eine „0“. In diesem Fall wird auf der 7-Segment-Anzeige des Motorcontrollers ein „A“ angezeigt, um darauf hinzuweisen, dass dieses Gerät noch nicht parametriert wurde. Wenn der Motorcontroller komplett unter CANopen parametriert wird, muss mindestens ein Bit in diesem Objekt gesetzt werden, um die Anzeige „A“ zu unterdrücken. Natürlich ist es bei Bedarf auch möglich, dieses Objekt zu nutzen, um sich den Zustand der Controllerparametrierung zu merken. Beachten Sie in diesem Fall, dass die Parametriersoftware ebenfalls auf dieses Objekt zugreift.

Sub-Index	C0_h
Description	commisioning_state
Data Type	UINT32
Access	rw
PDO Mapping	no
Units	–
Value Range	–
Default Value	0

Wert	Bedeutung
0	Nennstrom gültig
1	Maximalstrom gültig
2	Polzahl des Motors gültig
3	Offsetwinkel / Drehsinn gültig
4	Reserviert
5	Offsetwinkel / Drehsinn Hallgeber gültig
6	Reserviert
7	Absolutlage Gebersystem gültig
8	Stromregler-Parameter gültig
9	Reserviert
10	Physik. Einheiten gültig
11	Drehzahlregler gültig
12	Lageregler gültig
13	Sicherheitsparameter gültig
14	Reserviert
15	Endschalter-Polarität gültig
16 ... 31	Reserviert



Vorsicht

Dieses Objekt enthält keinerlei Informationen darüber, ob der Motorcontroller dem Motor und der Applikation entsprechend richtig parametrierung wurde, sondern nur, ob die genannten Punkte nach der Auslieferung mindestens einmal überhaupt parametrierung wurden.



„A“ in der 7-Segment-Anzeige

Beachten Sie, dass mindestens ein Bit im Objekt `commissioning_state` gesetzt werden muss, um das „A“ auf der 7-Segment-Anzeige Ihres Motorcontrollers zu unterdrücken.

5.18 Fehlermanagement

Übersicht

Die Motorcontroller der CMMP-Familie bieten die Möglichkeit, die Fehlerreaktion einzelner Ereignisse, wie z. B. das Auftreten eines Schleppfehlers, zu ändern. Dadurch reagiert der Motorcontroller unterschiedlich, wenn ein bestimmtes Ereignis eintritt: So kann je nach Einstellung heruntergebremsst werden, die Endstufe sofort ausgeschaltet werden aber auch lediglich eine Warnung auf dem Display angezeigt werden.

Für jedes Ereignis ist herstellenseitig eine Mindestreaktion vorgesehen, die nicht unterschritten werden kann. So lassen sich „kritische“ Fehler wie beispielsweise 60-0 Kurzschluss-Endstufe nicht umparametrieren, da hier eine sofortige Abschaltung notwendig ist, um den Motorcontroller vor einer eventuellen Zerstörung zu schützen.

Wird eine niedrigere Fehlerreaktion als für den jeweiligen Fehler zulässig eingetragen, wird der Wert auf die niedrigst zulässige Fehlerreaktion begrenzt. Eine Liste aller Fehlernummern befindet sich im Kapitel B „Diagnosemeldungen“.

Beschreibung der Objekte
In diesem Kapitel behandelte Objekte

Index	Objekt	Name	Typ	Attr.
2100 _h	RECORD	error_management		ro
2100_01 _h	VAR	error_number	UINT8	rw
2100_02 _h	VAR	error_reaction_code	UINT8	rw
200F _h	VAR	last_warning_code	UINT16	ro

Objekt 2100_h: error_management

Index	2100 _h
Name	error_management
Object Code	RECORD
No. of Elements	2

Im Objekt error_number muss die Hauptfehlernummer angegeben werden, deren Reaktion geändert werden soll. Die Hauptfehlernummer ist in der Regel vor dem Bindestrich angegeben (z. B. Fehler 08-2, Hauptfehlernummer 8). Für mögliche Fehlernummern → hierzu auch Kap. 3.5.

Sub-Index	01_h
Description	error_number
Data Type	UINT8
Access	rw
PDO Mapping	no
Units	–
Value Range	1 ... 96
Default Value	1

Im Objekt `error_reaction_code` kann die Reaktion des Fehlers verändert werden. Wird die herstellereigige Mindestreaktion unterschritten, wird auf diese begrenzt. Die wirklich eingestellte Reaktion kann durch Rücklesen bestimmt werden.

Sub-Index	02_h
Description	error_reaction_code
Data Type	UINT8
Access	rw
PDO Mapping	no
Units	–
Value Range	0, 1, 3, 5, 7, 8
Default Value	abhängig von <code>error_number</code>

Wert	Bedeutung
0	Keine Aktion
1	Eintrag im Puffer
3	Warnung auf der 7-Segment-Anzeige und im Statuswort
5	Reglerfreigabe aus
7	Bremsen mit Maximalstrom
8	Endstufe aus

Objekt 200F_h: `last_warning_code`

Warnungen sind besondere Ereignisse des Antriebs (z. B. ein Schleppfehler), die im Gegensatz zu einem Fehler nicht zum Stillsetzen des Antriebs führen sollen. Warnungen werden auf der 7-Segment-Anzeige des Reglers angezeigt und danach automatisch vom Regler zurückgesetzt.

Die letzte aufgetretene Warnung kann über das folgende Objekt ausgelesen werden: Dabei zeigt Bit 15 an, ob die Warnung aktuell noch aktiv ist.

Index	200F_h
Name	last_warning_code
Object Code	VAR
Data Type	UINT16

Access	ro
PDO Mapping	yes
Units	–
Value Range	–
Default Value	–

Bit	Wert	Bedeutung
0 ... 3	000F _h	Unternummer der Warnung
4 ... 11	0FF0 _h	Hauptnummer der Warnung
15	8000 _h	Warnung ist aktiv

6 Gerätesteuerung (Device Control)

6.1 Zustandsdiagramm (State Machine)

6.1.1 Übersicht

Das nachfolgende Kapitel beschreibt, wie der Motorcontroller unter CANopen gesteuert wird, also wie beispielsweise die Endstufe eingeschaltet oder ein Fehler quittiert wird.

Unter CANopen wird die gesamte Steuerung des Motorcontrollers über zwei Objekte realisiert: Über das controlword kann der Host den Motorcontroller steuern, während der Status des Motorcontrollers im Objekt statusword zurückgelesen werden kann. Zur Erklärung der Controllersteuerung werden die folgenden Begriffe verwandt:

Begriff	Beschreibung
Zustand: (State)	Je nachdem ob beispielsweise die Endstufe eingeschaltet oder ein Fehler aufgetreten ist befindet sich der Motorcontroller in verschiedenen Zuständen. Die unter CANopen definierten Zustände werden im Laufe des Kapitels vorgestellt. Beispiel: SWITCH_ON_DISABLED
Zustandsübergang (State Transition)	Ebenso wie die Zustände ist es unter CANopen ebenfalls definiert, wie man von einem Zustand zu einem anderen gelangt (z. B. um einen Fehler zu quittieren). Zustandsübergänge werden vom Host durch Setzen von Bits im controlword ausgelöst oder intern durch den Motorcontroller, wenn dieser beispielsweise einen Fehler erkennt.
Kommando (Command)	Zum Auslösen von Zustandsübergängen müssen bestimmte Kombinationen von Bits im controlword gesetzt werden. Eine solche Kombination wird als Kommando bezeichnet. Beispiel: Enable Operation
Zustandsdiagramm (State Machine)	Die Zustände und Zustandsübergänge bilden zusammen das Zustandsdiagramm, also die Übersicht über alle Zustände und die von dort möglichen Übergänge.

Tab. 6.1 Begriffe der Controllersteuerung

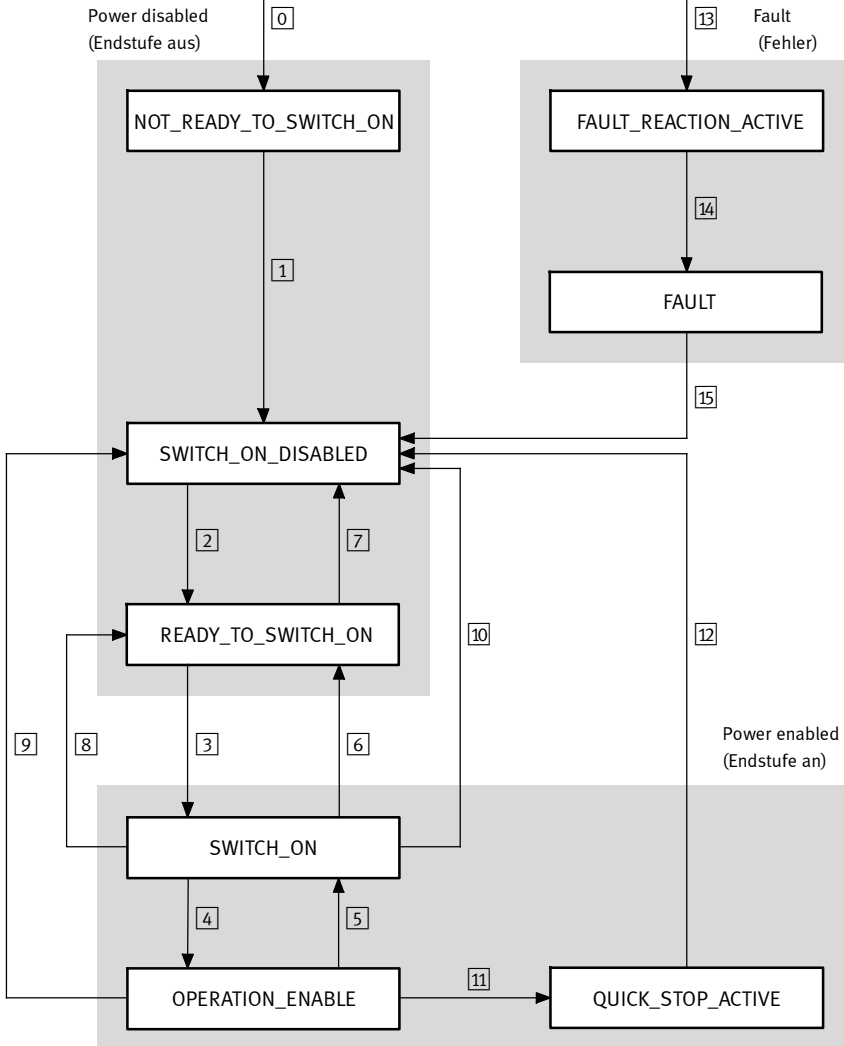
6.1.2 Das Zustandsdiagramm des Motorcontrollers (State Machine)

Fig. 6.1 Zustandsdiagramm des Motorcontrollers

Das Zustandsdiagramm kann grob in drei Bereiche aufgeteilt werden: „Power Disabled“ bedeutet, dass die Endstufe ausgeschaltet ist und „Power Enabled“ dass die Endstufe eingeschaltet ist. Im Bereich „Fault“ sind die zur Fehlerbehandlung notwendigen Zustände zusammengefasst. Die wichtigsten Zustände des Motorcontrollers sind im Diagramm hervorgehoben dargestellt. Nach dem Einschalten initialisiert sich der Motorcontroller und erreicht schließlich den Zustand

SWITCH_ON_DISABLED. In diesem Zustand ist die CAN-Kommunikation voll funktionsfähig und der Motorcontroller kann parametrierbar werden (z. B. die Betriebsart „Drehzahlregelung“ eingestellt werden). Die Endstufe ist ausgeschaltet und die Welle ist somit frei drehbar. Durch die Zustandsübergänge [2], [3], [4] – was im Prinzip der CAN-Reglerfreigabe entspricht – gelangt man in den Zustand OPERATION_ENABLE. In diesem Zustand ist die Endstufe eingeschaltet und der Motor wird gemäß der eingestellten Betriebsart geregelt. Stellen Sie daher vorher unbedingt sicher, dass der Antrieb richtig parametrierbar ist und ein entsprechender Sollwert gleich Null ist.

Der Zustandsübergang [9] entspricht der Wegnahme der Freigabe, d. h. ein noch laufender Motor würde ungeregelt austrudeln.

Tritt ein Fehler auf so wird (egal aus welchem Zustand) letztlich in den Zustand FAULT verzweigt. Je nach Schwere des Fehlers können vorher noch bestimmte Aktionen, wie z. B. eine Notbremsung ausgeführt werden (FAULT_REACTION_ACTIVE).

Um die genannten Zustandsübergänge auszuführen müssen bestimmte Bitkombinationen im controlword (siehe unten) gesetzt werden. Die unteren 4 Bits des controlwords werden gemeinsam ausgewertet, um einen Zustandsübergang auszulösen.

Im Folgenden werden zunächst nur die wichtigsten Zustandsübergänge [2], [3], [4], [9] und [15] erläutert. Eine Tabelle aller möglichen Zustände und Zustandsübergänge findet sich am Ende dieses Kapitels. Die folgende Tabelle enthält in der 1. Spalte den gewünschten Zustandsübergang und in der 2. Spalte die dazu notwendigen Voraussetzungen (Meistens ein Kommando durch den Host, hier mit Rahmen dargestellt). Wie dieses Kommando erzeugt wird, d. h. welche Bits im controlword zu setzen sind, ist in der 3. Spalte ersichtlich (x = nicht relevant).

Nr.	Wird durchgeführt wenn	Bitkombination (controlword)					Aktion
		Bit	3	2	1	0	
[2]	Endstufen- u. Reglerfreig. vorh. + Kommando Shutdown	Shutdown =	x	1	1	0	Keine
[3]	Kommando Switch On	Switch On =	x	1	1	1	Einschalten der Endstufe
[4]	Kommando Enable Operation	Enable Operation =	1	1	1	1	Regelung gemäß eingestellter Betriebsart
[9]	Kommando Disable Voltage	Disable Voltage =	x	x	0	x	Endstufe wird gesperrt. Motor ist frei drehbar.
[15]	Fehler behoben + Kommando Fault Reset	Fault Reset =	Bit 7 = 0 → 1				Fehler quittieren

Tab. 6.2 Wichtigste Zustandsübergänge des Motorcontrollers

BEISPIEL

Nachdem der Motorcontroller parametrierung wurde, soll der Motorcontroller „freigegeben“, d. h. die Endstufe und die Regelung eingeschaltet werden:

1. Der Motorcontroller ist im Zustand SWITCH_ON_DISABLED
2. Der Motorcontroller soll in den Zustand OPERATION_ENABLE
3. Laut Zustandsdiagramm (Fig. 6.1) sind die Übergänge [2], [3] und [4] auszuführen.
4. Aus Tab. 6.2 folgt:

Übergang [2]: controlword = 0006h

Neuer Zustand: READY_TO_SWITCH_ON¹⁾

Übergang [3]: controlword = 0007h

Neuer Zustand: SWITCHED_ON¹⁾

Übergang [4]: controlword = 000Fh

Neuer Zustand: OPERATION_ENABLE¹⁾

Hinweise:

1. Das Beispiel geht davon aus, dass keine weiteren Bits im controlword gesetzt sind (für die Übergänge sind ja nur die Bits 0 ... 3 wichtig).
2. Die Übergänge [3] und [4] können zusammengefasst werden, indem das controlword gleich auf 000Fh gesetzt wird. Für den Zustandsübergang [2] ist das gesetzte Bit 3 nicht relevant.
- 1) Der Host muss warten, bis der Zustand im statusword zurückgelesen werden kann. Dieses wird weiter unten noch ausführlich erläutert.

Zustandsdiagramm: Zustände

In der folgenden Tabelle sind alle Zustände und deren Bedeutung aufgeführt:

Name	Bedeutung
NOT_READY_TO_SWITCH_ON	Der Motorcontroller führt einen Selbsttest durch. Die CAN-Kommunikation arbeitet noch nicht.
SWITCH_ON_DISABLED	Der Motorcontroller hat seinen Selbsttest abgeschlossen. CAN-Kommunikation ist möglich.
READY_TO_SWITCH_ON	Der Motorcontroller wartet bis die digitalen Eingänge „Endstufen-“ und „Reglerfreigabe“ an 24 V liegen. (Reglerfreigabelogik „Digitaler Eingang und CAN“).
SWITCHED_ON ¹⁾	Die Endstufe ist eingeschaltet.
OPERATION_ENABLE ¹⁾	Der Motor liegt an Spannung und wird entsprechend der Betriebsart geregelt.
QUICKSTOP_ACTIVE ¹⁾	Die Quick Stop Function wird ausgeführt (→ quick_stop_option_code). Der Motor liegt an Spannung und wird entsprechend der Quick Stop Function geregelt.
FAULT_REACTION_ACTIVE ¹⁾	Es ist ein Fehler aufgetreten. Bei kritischen Fehlern wird sofort in den Status Fault gewechselt. Ansonsten wird die im fault_reaction_option_code vorgegebene Aktion ausgeführt. Der Motor liegt an Spannung und wird entsprechend der Fault Reaction Function geregelt.
FAULT	Es ist ein Fehler aufgetreten. Der Motor ist spannungsfrei.

1) Die Endstufe ist eingeschaltet.

Zustandsdiagramm: Zustandsübergänge

In der folgenden Tabelle sind alle Zustände und deren Bedeutung aufgeführt:

Nr.	Wird durchgeführt wenn	Bitkombination (controlword)					Aktion
		Bit	3	2	1	0	
0	Eingeschaltet o. Reset erfolgt	interner Übergang					Selbsttest ausführen
1	Selbsttest erfolgreich	interner Übergang					Aktivierung der CAN-Kommunikation
2	Endstufen- u. Reglerfreig. vorh. + Kommando Shutdown	Shutdown	x	1	1	0	–
3	Kommando Switch On	Switch On	x	1	1	1	Einschalten der Endstufe
4	Kommando Enable Operation	Enable Operation	1	1	1	1	Regelung gemäß eingestellter Betriebsart
5	Kommando Disable Operation	Disable Operation	0	1	1	1	Endstufe wird gesperrt. Motor ist frei drehbar.
6	Kommando Shutdown	Shutdown	x	1	1	0	Endstufe wird gesperrt. Motor ist frei drehbar.
7	Kommando Quick Stop	Quick Stop	x	0	1	x	–
8	Kommando Shutdown	Shutdown	x	1	1	0	Endstufe wird gesperrt. Motor ist frei drehbar.
9	Kommando Disable Voltage	Disable Voltage	x	x	0	x	Endstufe wird gesperrt. Motor ist frei drehbar.
10	Kommando Disable Voltage	Disable Voltage	x	x	0	x	Endstufe wird gesperrt. Motor ist frei drehbar.
11	Kommando Quick Stop	Quick Stop	x	0	1	x	Es wird eine Bremsung gemäß quick_stop_option_code eingeleitet.
12	Bremsung beendet o. Kommando Disable Voltage	Disable Voltage	x	x	0	x	Endstufe wird gesperrt. Motor ist frei drehbar.
13	Fehler aufgetreten	interner Übergang					Bei unkritischen Fehlern Reaktion gemäß fault_reaction_option_code. Bei kritischen Fehlern folgt Übergang 14
14	Fehlerbehandlung ist beendet	interner Übergang					Endstufe wird gesperrt. Motor ist frei drehbar.
15	Fehler behoben + Kommando Fault Reset	Fault Reset	Bit 7 = 0 → 1				Fehler quittieren (bei steigender Flanke)



Vorsicht
Endstufe gesperrt ...

... bedeutet, dass die Leistungshalbleiter (Transistoren) nicht mehr angesteuert werden. Wenn dieser Zustand bei einem drehenden Motor eingenommen wird, so trudelt dieser ungebremst aus. Eine eventuell vorhandene mechanische Motorbremse wird hierbei automatisch angezogen.
Das Signal garantiert nicht, dass der Motor wirklich spannungsfrei ist.



Vorsicht
Endstufe freigegeben ...

... bedeutet, dass der Motor entsprechend der gewählten Betriebsart angesteuert und geregelt wird. Eine eventuell vorhandene mechanische Motorbremse wird automatisch gelöst. Bei einem Defekt oder einer Fehlparametrierung (Motorstrom, Polzahl, Resolver-offsetwinkel etc.) kann es zu einem unkontrollierten Verhalten des Antriebes kommen.

6.1.3 Steuerwort (Controlword)

Objekt 6040_h: controlword

Mit dem controlword kann der aktuelle Zustand des Motorcontrollers geändert bzw. direkt eine bestimmte Aktion (z. B. Start der Referenzfahrt) ausgelöst werden. Die Funktion der Bits 4, 5, 6 und 8 hängt von der aktuellen Betriebsart (modes_of_operation) des Motorcontrollers ab, die nach diesem Kapitel erläutert wird.

Index	6040_h
Name	controlword
Object Code	VAR
Data Type	UINT16

Access	rw
PDO Mapping	yes
Units	–
Value Range	–
Default Value	0

Bit	Wert	Funktion
0	0001 _h	Steuerung der Zustandsübergänge. (Diese Bits werden gemeinsam ausgewertet)
1	0002 _h	
2	0004 _h	
3	0008 _h	
4	0010 _h	new_set_point/start_homing_operation/enable_ip_mode
5	0020 _h	change_set_immediatly
6	0040 _h	absolute/relative
7	0080 _h	reset_fault
8	0100 _h	halt
9	0200 _h	reserved – set to 0
10	0400 _h	reserved – set to 0
11	0800 _h	reserved – set to 0
12	1000 _h	reserved – set to 0
13	2000 _h	reserved – set to 0
14	4000 _h	reserved – set to 0
15	8000 _h	reserved – set to 0

Tab. 6.3 Bitbelegung des controlword

Wie bereits umfassend beschrieben können mit den Bits 0 ... 3 Zustandsübergänge ausgeführt werden. Die dazu notwendigen Kommandos sind hier noch einmal in einer Übersicht dargestellt. Das Kommando Fault Reset wird durch einen positiven Flankenwechsel (von 0 nach 1) von Bit 7 erzeugt.

Kommando:	Bit 7	Bit 3	Bit 2	Bit 1	Bit 0
	0008 _h	0008 _h	0004 _h	0002 _h	0001 _h
Shutdown	x	x	1	1	0
Switch On	x	x	1	1	1
Disable Voltage	x	x	x	0	x
Quick Stop	x	x	0	1	x
Disable Operation	x	0	1	1	1
Enable Operation	x	1	1	1	1
Fault Reset	0 → 1	x	x	x	x

Tab. 6.4 Übersicht aller Kommandos (x = nicht relevant)



Da einige Statusänderungen einen gewissen Zeitraum beanspruchen, müssen alle über das controlword ausgelösten Statusänderungen über das statusword zurückgelesen werden. Erst wenn der angeforderte Status auch im statusword gelesen werden kann, darf über das controlword ein weiteres Kommando eingeschrieben werden.

Nachfolgend sind die restlichen Bits des controlwords erläutert. Einige Bits haben dabei je nach Betriebsart (modes_of_operation), d. h. ob der Motorcontroller z. B. drehzahl- oder momentengeregt wird, unterschiedliche Bedeutung:

controlword		
Bit	Funktion	Beschreibung
4	Abhängig von modes_of_operation	
	new_set_point	Im Profile Position Mode: Eine steigende Flanke signalisiert dem Motorcontroller, dass ein neuer Fahrauftrag übernommen werden soll. → dazu unbedingt auch Kapitel 7.3.
	start_homing_operation	Im Homing Mode: Eine steigende Flanke bewirkt, dass die parametrisierte Referenzfahrt gestartet wird. Eine fallende Flanke bricht eine laufende Referenzfahrt vorzeitig ab.
	enable_ip_mode	Im Interpolated Position Mode: Dieses Bit muss gesetzt werden, wenn die Interpolations-Datensätze ausgewertet werden sollen. Es wird durch das Bit ip_mode_active im statusword quittiert. → hierzu unbedingt auch Kapitel 7.4.
5	change_set_immediatly	Nur im Profile Position Mode: Wenn dieses Bit nicht gesetzt ist, so wird bei einem neuen Fahrauftrag zuerst ein eventuell laufender abgearbeitet und erst dann mit dem neuen begonnen. Bei gesetztem Bit wird eine laufende Positionierung sofort abgebrochen und durch den neuen Fahrauftrag ersetzt. → dazu unbedingt auch Kapitel 7.3.
6	relative	Nur im Profile Position Mode: Bei gesetztem Bit bezieht der Motorcontroller die Zielposition (target_position) des aktuellen Fahrauftrages relativ auf die Sollposition (position_demand_value) des Lagereglers.
7	reset_fault	Beim Übergang von Null auf Eins versucht der Motorcontroller die vorhandenen Fehler zu quittieren. Dies gelingt nur, wenn die Ursache für den Fehler behoben wurde.

controlword		
Bit	Funktion	Beschreibung
8	halt	<p>Im Profile Position Mode: Bei gesetztem Bit wird die laufende Positionierung abgebrochen. Gebremst wird hierbei mit der profile_deceleration. Nach Beendigung des Vorgangs wird im statusword das Bit target_reached gesetzt. Das Löschen des Bits hat keine Auswirkung.</p>
		<p>Im Profile Velocity Mode: Bei gesetztem Bit wird die Drehzahl auf Null abgesenkt. Gebremst wird hierbei mit der profile_deceleration. Das Löschen des Bits bewirkt, dass der Motorcontroller wieder beschleunigt.</p>
		<p>Im Profile Torque Mode: Bei gesetztem Bit wird das Drehmoment auf Null abgesenkt. Dies geschieht mit der torque_slope. Das Löschen des Bits bewirkt, dass der Motorcontroller wieder beschleunigt.</p>
		<p>Im Homing Mode: Bei gesetztem Bit wird die laufende Referenzfahrt abgebrochen. Das Löschen des Bits hat keine Auswirkung.</p>

Tab. 6.5 controlword Bit 4 ... 8

6.1.4 Auslesen des Motorcontrollerzustands

Ähnlich wie über die Kombination mehrerer Bits des controlwords verschiedene Zustandsübergänge ausgelöst werden können, kann über die Kombination verschiedener Bits des statusword ausgelesen werden, in welchem Zustand sich der Motorcontroller befindet.

Die folgende Tabelle listet die möglichen Zustände des Zustandsdiagramms sowie die zugehörige Bitkombination auf, mit der sie im statusword angezeigt werden.

Zustand	Bit 6	Bit 5	Bit 3	Bit 2	Bit 1	Bit 0	Maske	Wert
	0040 _h	0020 _h	0008 _h	0004 _h	0002 _h	0001 _h		
Not_Ready_To_Switch_On	0	x	0	0	0	0	004F _h	0000 _h
Switch_On_Disabled	1	x	0	0	0	0	004F _h	0040 _h
Ready_to_Switch_On	0	1	0	0	0	1	006F _h	0021 _h
Switched_On	0	1	0	0	1	1	006F _h	0023 _h
OPERATION_ENABLE	0	1	0	1	1	1	006F _h	0027 _h
QUICK_STOP_ACTIVE	0	0	0	1	1	1	006F _h	0007 _h
Fault_Reaction_Active	0	x	1	1	1	1	004F _h	000F _h
Fault	0	x	1	1	1	1	004F _h	0008 _h
FAULT (gemäß CiA 402) ¹⁾	0	x	1	0	0	0	004F _h	0008 _h

Tab. 6.6 Gerätestatus (x = nicht relevant)

BEISPIEL

Das obige Beispiel zeigt, welche Bits im controlword gesetzt werden müssen, um den Motorcontroller freizugeben. Jetzt soll dabei der neu eingeschriebene Zustand aus dem statusword ausgelesen werden:

Übergang von SWITCH_ON_DISABLED zu OPERATION_ENABLE:

1. Zustandsübergang [2] ins controlword schreiben.
2. Warten, bis der Zustand READY_TO_SWITCH_ON im statusword angezeigt wird.
Übergang 2: controlword = 0006_h
Warten bis (statusword & 006F_h) = 0021_h¹⁾
3. Zustandsübergang [3] und [4] können zusammengefasst ins controlword geschrieben werden.
4. Warten, bis der Zustand OPERATION_ENABLE im statusword angezeigt wird.
Übergang 3+4: controlword = 000F_h
Warten bis (statusword & 006F_h) = 0027_h¹⁾

Hinweis:

Das Beispiel geht davon aus, dass keine weiteren Bits im controlword gesetzt sind (für die Übergänge sind ja nur die Bits 0 ... 3 wichtig).

- 1) Für die Identifizierung der Zustände müssen auch nicht gesetzte Bits ausgewertet werden (siehe Tabelle). Daher muss das statusword entsprechend maskiert werden.

6.1.5 Statusworte (Statuswords)**Objekt 6041_h: statusword**

Index	6041_h
Name	statusword
Object Code	VAR
Data Type	UINT16

Access	ro
PDO Mapping	yes
Units	–
Value Range	–
Default Value	–

Bit	Wert	Funktion
0	0001 _h	Zustand des Motorcontrollers (→ Tab. 6.6). (Diese Bits müssen gemeinsam ausgewertet werden.)
1	0002 _h	
2	0004 _h	
3	0008 _h	
4	0010 _h	voltage_enabled
5	0020 _h	Zustand des Motorcontrollers (→ Tab. 6.6).
6	0040 _h	
7	0080 _h	warning
8	0100 _h	drive_is_moving
9	0200 _h	remote
10	0400 _h	target_reached
11	0800 _h	internal_limit_active
12	1000 _h	set_point_acknowledge/speed_0/homing_attained/ip_mode_active
13	2000 _h	following_error/homing_error
14	4000 _h	manufacturer_statusbit
15	8000 _h	Antrieb referenziert

Tab. 6.7 Bitbelegung im statusword



Alle Bits des statusword sind nicht gepuffert. Sie repräsentieren den aktuellen Gerätestatus.

Neben dem Motorcontrollerstatus werden im statusword diverse Ereignisse angezeigt, d. h. jedem Bit ist ein bestimmtes Ereignis wie z. B. Schleppfehler zugeordnet. Die einzelnen Bits haben dabei folgende Bedeutung:

statusword		
Bit	Funktion	Beschreibung
4	voltage_enabled	Dieses Bit ist gesetzt, wenn die Endstufentransistoren eingeschaltet sind. Wenn im Objekt 6510 _h _FO _h (compatibility_control) Bit 7 gesetzt ist, gilt (→ Kap. 5.2): Dieses Bit ist gesetzt, wenn die Endstufentransistoren eingeschaltet sind.

Tab. 6.8 statusword Bit 4

**Warnung**

Bei einem Defekt kann der Motor trotzdem unter Spannung stehen.

statusword		
Bit	Funktion	Beschreibung
5	quick_stop	Bei gelöschtem Bit führt der Antrieb einen Quick Stop gemäß quick_stop_option_code aus.
7	warning	Dieses Bit zeigt an, dass eine Warnung aktiv ist.
8	drive_is_moving	Dieses Bit wird – unabhängig von modes_of_operation – gesetzt, wenn sich die aktuelle Ist-Drehzahl (velocity_actual_value) des Antriebes außerhalb des zugehörigen Toleranzfenster befindet (velocity_threshold).
9	remote	Dieses Bit zeigt an, dass die Endstufe des Motorcontrollers über das CAN-Netzwerk freigegeben werden kann. Es ist gesetzt, wenn die Reglerfreigabelogik über das Objekt enable_logic entsprechend eingestellt ist.

statusword		
Bit	Funktion	Beschreibung
10	Abhängig von modes_of_operation.	
	target_reached	<p>Im Profile Position Mode: Das Bit wird gesetzt, wenn die aktuelle Zielposition erreicht ist und sich die aktuelle Position (position_actual_value) im parametrisierten Positionsfenster (position_window) befindet. Außerdem wird es gesetzt, wenn der Antrieb bei gesetztem Halt-Bit zum Stillstand kommt. Es wird gelöscht, sobald ein neues Ziel vorgegeben wird.</p>
		<p>Im Profile Velocity Mode Das Bit wird gesetzt, wenn sich die Drehzahl (velocity_actual_value) des Antriebs im Toleranzfenster befindet (velocity_window, velocity_window_time).</p>
11	internal_limit_active	Dieses Bit zeigt an, dass die I ² t-Begrenzung aktiv ist.
12	Abhängig von modes_of_operation.	
	set_point_acknowledge	<p>Im Profile Position Mode Dieses Bit wird gesetzt, wenn der Motorcontroller das gesetzte Bit new_set_point im controlword erkannt hat. Es wird wieder gelöscht, nachdem das Bit new_set_point im controlword auf Null gesetzt wurde. → dazu unbedingt auch Kapitel 7.3</p>
	speed_0	<p>Im Profile Velocity Mode Dieses Bit wird gesetzt, wenn sich die aktuelle Ist-Drehzahl (velocity_actual_value) des Antriebes im zugehörigen Toleranzfenster befindet (velocity_threshold).</p>
	homing_attained	<p>Im Homing Mode: Dieses bit wird gesetzt, wenn die Referenzfahrt ohne Fehler beendet wurde.</p>
	ip_mode_active	<p>Im Interpolated Position Mode: Dieses Bit zeigt an, dass die Interpolation aktiv ist und die Interpolations-Datensätze ausgewertet werden. Es wird gesetzt, wenn dies durch das Bit enable_ip_mode im controlword angefordert wurde. → hierzu unbedingt auch Kapitel 7.4.</p>

statusword		
Bit	Funktion	Beschreibung
13	Abhängig von modes_of_operation.	
	following_error	Im Profile Position Mode: Dieses Bit wird gesetzt, wenn die aktuelle Ist-Position (position_actual_value) von der Soll-Position (position_demand_value) soweit abweicht, dass die Differenz außerhalb des parametrisierten Toleranzfensters liegt (following_error_window, following_error_time_out).
	homing_error	Im Homing Mode: Dieses Bit wird gesetzt, wenn die Referenzfahrt unterbrochen wird (Halt-Bit), beide Endscharter gleichzeitig ansprechen oder die bereits zurückgelegte Endschaltersuchfahrt größer als der vorgegebene Positionierraum ist (min_position_limit, max_position_limit).
14	manufacturer_statusbit	Herstellerspezifisch Die Bedeutung dieses Bits ist konfigurierbar: Es kann gesetzt werden, wenn ein beliebiges Bit des manufacturer_statusword_1 gesetzt bzw. zurückgesetzt wird. ➔ hierzu auch Kap. 6.1.5 Objekt 2000 _h .
15	Antrieb referenziert	Das Bit wird gesetzt, wenn der Regler referenziert ist. Dies ist der Fall, wenn entweder eine Referenzfahrt erfolgreich durchgeführt wurde oder aufgrund des angeschlossenen Gebersystems (z. B. bei einem Absolutwertgeber) keine Referenzfahrt nötig ist.

Tab. 6.9 statusword Bit 5 ... 15

Objekt 2000_h: manufacturer_statuswords

Um weitere Reglerzustände abbilden zu können, die nicht im – häufig zyklisch abgefragten – statusword vorhanden sein müssen, wurde die Objektgruppe manufacturer_statuswords eingeführt.

Index	2000_h
Name	manufacturer_statuswords
Object Code	RECORD
No. of Elements	1

Sub-Index	01 _h
Description	manufacturer_statusword_1
Data Type	UINT32
Access	ro
PDO Mapping	yes
Units	–
Value Range	–
Default Value	–

Bit	Wertigkeit	Name
0	00000001 _h	is_referenced
1	00000002 _h	commutation_valid
2	00000004 _h	ready_for_enable
...		
31	80000000 _h	–

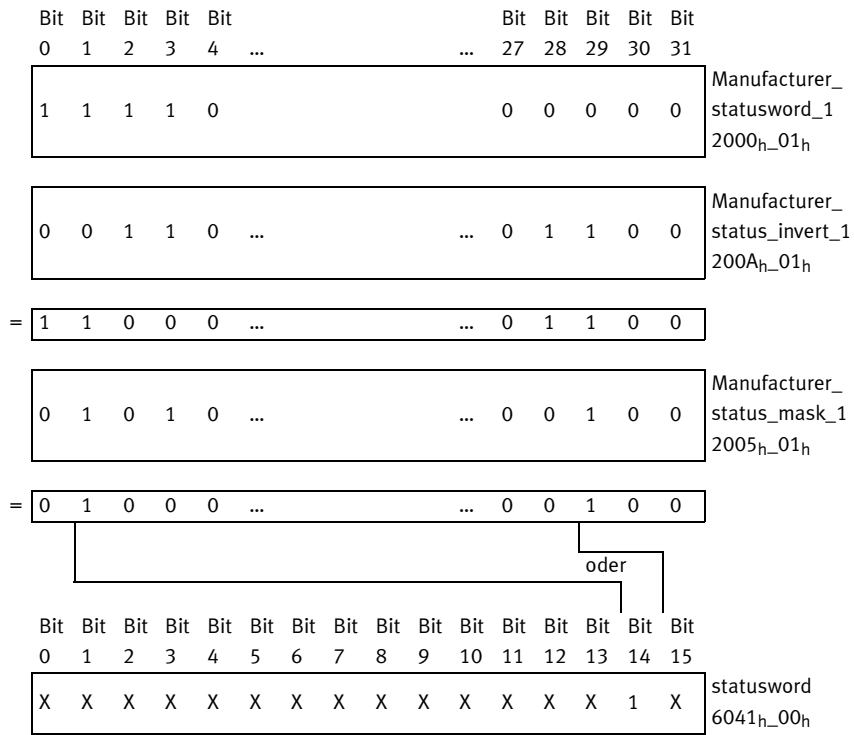
Tab. 6.10 Bitbelegung im manufacturer_statusword_1

Bit	Funktion	Beschreibung
0	is_referenced	Das Bit wird gesetzt, wenn der Regler referenziert ist. Dies ist der Fall, wenn entweder eine Referenzfahrt erfolgreich durchgeführt wurde oder aufgrund des angeschlossenen Gebersystems (z. B. bei einem Absolutwertgeber) keine Referenzfahrt nötig ist.
1	commutation_valid	Das Bit wird gesetzt, wenn die Kommutierinformation gültig ist. Es ist insbesondere bei Gebersystemen ohne Kommutierinformation (z. B. Linearmotoren) hilfreich, weil dort die automatische Kommutierungsfindung einige Zeit in Anspruch nehmen kann. Wird dieses Bit überwacht, kann z. B. ein Timeout der Steuerung bei Freigabe des Reglers verhindert werden.
2	ready_for_enable	Das Bit wird gesetzt, wenn alle Bedingungen vorliegen, um den Regler freizugeben und nur noch die Reglerfreigabe selber fehlt. Folgende Bedingungen müssen vorliegen: <ul style="list-style-type: none"> – Der Antrieb ist fehlerfrei. – Der Zwischenkreis ist geladen. – Die Winkelgeberausertung ist bereit. Es sind keine Prozesse (z.B. serielle Übertragungen) aktiv, die eine Freigabe verhindern. <ul style="list-style-type: none"> – Es ist kein blockierender Prozess aktiv (z. B. die automatische Motorparameter-Identifikation).

Tab. 6.11 Bitbelegung im manufacturer_statusword_1

Mithilfe der Objekte `manufacturer_status_masks` und `manufacturer_status_invert` können ein oder mehrere Bits der `manufacturer_statuswords` in Bit 14 (`manufacturer_statusbit`) des statusword (`6041h`) eingeblendet werden. Alle Bits des `manufacturer_statusword_1` können über das korrespondierende Bit in `manufacturer_status_invert_1` invertiert werden. Somit können auch Bits auf den Zustand „zurückgesetzt“ überwacht werden. Nach der Invertierung werden die Bits maskiert, d. h. nur wenn das korrespondierende Bit in `manufacturer_status_mask_1` gesetzt ist, wird das Bit weiter ausgewertet. Ist nach der Maskierung noch mindestens ein Bit gesetzt, wird auch Bit 14 des statusword gesetzt.

Die folgende Abbildung verdeutlicht dieses beispielhaft:



BEISPIEL

- a) Bit 14 des statusword soll gesetzt werden, wenn der Antrieb referenziert ist.
 Antrieb referenziert ist Bit 0 des manufacturer_statusword_1
 manufacturer_status_invert = 0x00000000
 manufacturer_status_mask = 0x00000001 (Bit 0)
- b) Bit 14 des statusword soll gesetzt werden, wenn der Antrieb keine gültige Kommutierlage hat.
 Gültige Kommutierlage ist Bit 1 des manufacturer_statusword_1.
 Dieses Bit muss invertiert werden, damit es gesetzt wird, wenn die Kommutierinformation ungültig ist:
 manufacturer_status_invert = 0x00000002 (Bit 1)
 manufacturer_status_mask = 0x00000002 (Bit 1)
- c) Bit 14 des statusword soll gesetzt werden, wenn der Antrieb nicht bereit zur Freigabe ist ODER der Antrieb referenziert ist.
 Gültige Kommutierlage ist Bit 2 des manufacturer_statusword_1.
 Antrieb referenziert ist Bit 0. Bit 2 muss invertiert werden, damit es gesetzt wird, wenn der Antrieb nicht bereit zur Freigabe ist:
 manufacturer_status_invert = 0x00000004 (Bit 2)
 manufacturer_status_mask = 0x00000005 (Bit 2, Bit 0)

Objekt 2005_h: manufacturer_status_masks

Mit dieser Objektgruppe wird festgelegt, welche gesetzten Bits der manufacturer_statuswords in das statusword eingeblendet werden. → hierzu auch Kapitel 6.1.5.

Index	2005_h
Name	manufacturer_status_masks
Object Code	RECORD
No. of Elements	1

Sub-Index	01 _h
Description	manufacturer_status_mask_1
Data Type	UINT32
Access	rw
PDO Mapping	yes
Units	–
Value Range	–
Default Value	0x00000000

Objekt 200A_h: manufacturer_status_invert

Mit dieser Objektgruppe wird festgelegt, welche Bits der manufacturer_statuswords invertiert in das statusword eingeblendet werden. → hierzu auch Kapitel 6.1.5.

Index	200A_h
Name	manufacturer_status_invert
Object Code	RECORD
No. of Elements	1

Sub-Index	01 _h
Description	manufacturer_status_invert_1
Data Type	UINT32
Access	rw
PDO Mapping	yes
Units	–
Value Range	–
Default Value	0x00000000

6.1.6 Beschreibung der weiteren Objekte**In diesem Kapitel behandelte Objekte**

Index	Objekt	Name	Typ	Attr.
605B _h	VAR	shutdown_option_code	INT16	rw
605C _h	VAR	disable_operation_option_code	INT16	rw
605A _h	VAR	quick_stop_option_code	INT16	rw
605E _h	VAR	fault_reaction_option_code	INT16	rw

Objekt 605B_h: shutdown_option_code

Mit dem Objekt shutdown_option_code wird vorgegeben, wie sich der Motorcontroller beim Zustandsübergang 8 (von OPERATION ENABLE nach READY TO SWITCH ON) verhält. Das Objekt zeigt das implementierte Verhalten des Motorcontrollers an. Es kann nicht verändert werden.

Index	605B_h
Name	shutdown_option_code
Object Code	VAR
Data Type	INT16

Access	rw
PDO Mapping	no
Units	–
Value Range	0
Default Value	0

Wert	Bedeutung
0	Endstufe wird ausgeschaltet, Motor ist frei drehbar

Objekt 605C_h: disable_operation_option_code

Mit dem Objekt disable_operation_option_code wird vorgegeben, wie sich der Motorcontroller beim Zustandsübergang 5 (von OPERATION ENABLE nach SWITCH ON) verhält. Das Objekt zeigt das implementierte Verhalten des Motorcontrollers an. Es kann nicht verändert werden.

Index	605C_h
Name	disable_operation_option_code
Object Code	VAR
Data Type	INT16

Access	rw
PDO Mapping	no
Units	–
Value Range	-1
Default Value	-1

Wert	Bedeutung
-1	Bremsen mit quickstop_deceleration

Objekt 605A_h: quick_stop_option_code

Mit dem Parameter quick_stop_option_code wird vorgegeben, wie sich der Motorcontroller bei einem Quick Stop verhält. Das Objekt zeigt das implementierte Verhalten des Motorcontrollers an. Es kann nicht verändert werden.

Index	605A_h
Name	quick_stop_option_code
Object Code	VAR
Data Type	INT16

Access	rw
PDO Mapping	no
Units	–
Value Range	2
Default Value	2

Wert	Bedeutung
2	Bremsen mit quickstop_deceleration

Objekt 605E_h: fault_reaction_option_code

Mit dem Objekt fault_reaction_option_code wird vorgegeben, wie sich der Motorcontroller bei einem Fehler (fault) verhält. Da bei der CMMP-Reihe die Fehlerreaktion vom jeweiligen Fehler abhängt, kann dieses Objekt nicht parametrierbar werden und gibt immer 0 zurück. Um die Fehlerreaktion der einzelnen Fehler zu verändern → Kapitel 5.18, Fehlermanagement.

Index	605E_h
Name	fault_reaction_option_code
Object Code	VAR
Data Type	INT16

Access	rw
PDO Mapping	no
Units	–
Value Range	0
Default Value	0

7 Betriebsarten

7.1 Einstellen der Betriebsart

7.1.1 Übersicht

Der Motorcontroller kann in eine Vielzahl von Betriebsarten versetzt werden. Nur einige sind unter CANopen detailliert spezifiziert:

- Momentengeregelter Betrieb
- Drehzahlgeregelter Betrieb
- Referenzfahrt
- Positionierbetrieb
- Synchrone Positionsvorgabe
- profile torque mode
- profile velocity mode
- homing mode
- profile position mode
- interpolated position mode

7.1.2 Beschreibung der Objekte

In diesem Kapitel behandelte Objekte

Index	Objekt	Name	Typ	Attr.
6060 _h	VAR	modes_of_operation	INT8	wo
6061 _h	VAR	modes_of_operation_display	INT8	ro

Objekt 6060_h: modes_of_operation

Mit dem Objekt modes_of_operation wird die Betriebsart des Motorcontrollers eingestellt.

Index	6060_h
Name	modes_of_operation
Object Code	VAR
Data Type	INT8

Access	rw
PDO Mapping	yes
Units	–
Value Range	1, 3, 4, 6, 7
Default Value	–

Wert	Bedeutung
1	Profile Position Mode (Lageregler mit Positionierbetrieb)
3	Profile Velocity Mode (Drehzahlregler mit Sollwertrampe)
4	Profile Torque Mode (Momentenregler mit Sollwertrampe)
6	Homing Mode (Referenzfahrt)
7	Interpolated Position Mode



Die aktuelle Betriebsart kann nur im Objekt `modes_of_operation_display` gelesen werden! Da ein Wechsel der Betriebsart etwas Zeit in Anspruch nehmen kann, muss solange gewartet werden, bis der neu ausgewählte Modus im Objekt `modes_of_operation_display` erscheint.

Objekt 6061_h: `modes_of_operation_display`

Im Objekt `modes_of_operation_display` kann die aktuelle Betriebsart des Motorcontrollers gelesen werden. Wird eine Betriebsart über das Objekt 6060_h eingestellt, werden neben der eigentlichen Betriebsart auch die Sollwert-Aufschaltungen (Sollwert-Selektor) vorgenommen, die für einen Betrieb des Motorcontrollers unter CANopen nötig sind. Dies sind:

Selektor	Profile Velocity Mode	Profile Torque Mode
A	Drehzahl-Sollwert (Feldbus 1)	Drehmoment-Sollwert (Feldbus 1)
B	Ggf. Momentenbegrenzung	Ggf. Drehzahlbegrenzung
C	Drehzahl-Sollwert (Synchrorehz.)	inaktiv

Außerdem wird die Sollwert-Rampe grundsätzlich eingeschaltet. Nur wenn diese Aufschaltungen in der genannten Weise eingestellt sind, wird auch eine der CANopen-Betriebsarten zurückgegeben. Werden diese Einstellungen z. B. mit der Parametriersoftware geändert, wird eine jeweilige „User“-Betriebsart zurückgegeben, um anzuzeigen, dass die Selektoren verändert wurden.

Index	6061_h
Name	<code>modes_of_operation_display</code>
Object Code	VAR
Data Type	INT8

Access	ro
PDO Mapping	yes
Units	–
Value Range	siehe Tabelle
Default Value	3

Wert	Bedeutung
-1	Ungültige Betriebsart oder Betriebsartenwechsel
-11	User Position Mode
-13	User Velocity Mode
-14	User Torque Mode
1	Profile Position Mode (Lageregler mit Positionierbetrieb)
3	Profile Velocity Mode (Drehzahlregler mit Sollwertrampe)
4	Profile Torque Mode (Momentenregler mit Sollwertrampe)
6	Homing Mode (Referenzfahrt)
7	Interpolated Position Mode



Die Betriebsart kann nur über das Objekt `modes_of_operation` gesetzt werden. Da ein Wechsel der Betriebsart etwas Zeit in Anspruch nehmen kann, muss solange gewartet werden, bis der neu ausgewählte Modus im Objekt `modes_of_operation_display` erscheint. Während dieses Zeitraumes kann kurzzeitig „ungültige Betriebsart“ (-1) angezeigt werden.

7.2 Betriebsart Referenzfahrt (Homing Mode)

7.2.1 Übersicht

In diesem Kapitel wird beschrieben, wie der Motorcontroller die Anfangsposition sucht (auch Bezugspunkt, Referenzpunkt oder Nullpunkt genannt). Es gibt verschiedene Methoden diese Position zu bestimmen, wobei entweder die Endschalter am Ende des Positionierbereiches benutzt werden können oder aber ein Referenzschalter (Nullpunkt-Schalter) innerhalb des möglichen Verfahrweges. Um eine möglichst große Reproduzierbarkeit zu erreichen, kann bei einigen Methoden der Nullimpuls des verwendeten Winkelgebers (Resolver, Inkrementalgeber etc.) mit einbezogen werden.

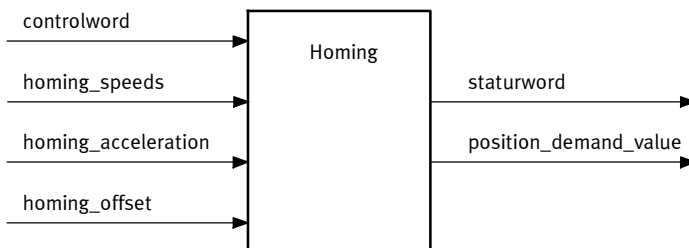


Fig. 7.1 Die Referenzfahrt

Der Benutzer kann die Geschwindigkeit, Beschleunigung und die Art der Referenzfahrt bestimmen. Mit dem Objekt `home_offset` kann die Nullposition des Antriebs an eine beliebige Stelle verschoben werden.

Es gibt zwei Referenzfahrgeschwindigkeiten. Die höhere Suchgeschwindigkeit (speed_during_search_for_switch) wird benutzt, um den Endschalter bzw. den Referenzschalter zu finden. Um dann die Position der betreffenden Schaltflanke exakt bestimmen zu können, wird auf die Kriechgeschwindigkeit (speed_during_search_for_zero) umgeschaltet. Soll der Antrieb nicht neu referenziert werden, sondern lediglich die Position auf einen vorgegebenen Wert gesetzt werden, kann das Objekt 2030_h (set_position_absolute) benutzt werden → Seite 117.



Die Fahrt auf die Nullposition ist unter CANopen in der Regel nicht Bestandteil der Referenzfahrt. Sind dem Motorcontroller alle erforderlichen Größen bekannt (z. B. weil er die Lage des Nullimpulses bereits kennt), wird keine physikalische Bewegung ausgeführt. Dieses Verhalten kann durch das Objekt 6510_h_F0_h (compatibility_control, → Kap. 5.2) geändert werden, so dass immer eine Fahrt auf Null ausgeführt wird.

7.2.2 **Beschreibung der Objekte**

In diesem Kapitel behandelte Objekte

Index	Objekt	Name	Typ	Attr.
607C _h	VAR	home_offset	INT32	rw
6098 _h	VAR	homing_method	INT8	rw
6099 _h	ARRAY	homing_speeds	UINT32	rw
609A _h	VAR	homing_acceleration	UINT32	rw
2045 _h	VAR	homing_timeout	UINT16	rw

Betroffene Objekte aus anderen Kapiteln

Index	Objekt	Name	Typ	Kapitel
6040 _h	VAR	controlword	UINT16	6.1.3 Controlword (Steuerwort)
6041 _h	VAR	statusword	UINT16	6.1.5 Statuswords (Statusworte)

Objekt 607C_h: home_offset

Das Objekt home_offset legt die Verschiebung der Nullposition gegenüber der ermittelten Referenzposition fest.

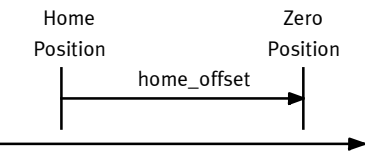


Fig. 7.2 Home Offset

Index	607C_h
Name	home_offset
Object Code	VAR
Data Type	INT32

Access	rw
PDO Mapping	yes
Units	position units
Value Range	–
Default Value	0

Objekt 6098_h: homing_method

Für eine Referenzfahrt werden eine Reihe unterschiedlicher Methoden bereitgestellt. Über das Objekt homing_method kann die für die Applikation benötigte Variante ausgewählt werden. Es gibt vier mögliche Referenzfahrt-Signale: den negativen und positiven Endschalter, den Referenzschalter und den (periodischen) Nullimpuls des Winkelgebers. Außerdem kann der Motorcontroller sich ganz ohne zusätzliches Signal auf den negativen oder positiven Anschlag referenzieren. Wenn über das Objekt homing_method eine Methode zum Referenzieren bestimmt wird, so werden hiermit folgende Einstellungen gemacht:

- Die Referenzquelle (neg./pos. Endschalter, der Referenzschalter, neg. / pos. Anschlag)
- Die Richtung und der Ablauf der Referenzfahrt
- Die Art der Auswertung des Nullimpulses vom verwendeten Winkelgeber

Index	6098_h
Name	homing_method
Object Code	VAR
Data Type	INT8

Access	rw
PDO Mapping	yes
Units	
Value Range	-18, -17, -2, -1, 1, 2, 7, 11, 17, 18, 23, 27, 32, 33, 34, 35
Default Value	17

Wert	Richtung	Ziel	Bezugspunkt für Null
-18	positiv	Anschlag	Anschlag
-17	negativ	Anschlag	Anschlag
-2	positiv	Anschlag	Nullimpuls
-1	negativ	Anschlag	Nullimpuls
1	negativ	Endschalter	Nullimpuls
2	positiv	Endschalter	Nullimpuls
7	positiv	Referenzschalter	Nullimpuls
11	negativ	Referenzschalter	Nullimpuls
17	negativ	Endschalter	Endschalter
18	positiv	Endschalter	Endschalter
23	positiv	Referenzschalter	Referenzschalter
27	negativ	Referenzschalter	Referenzschalter
33	negativ	Nullimpuls	Nullimpuls
34	positiv	Nullimpuls	Nullimpuls
35		Keine Fahrt	Aktuelle Ist-Position

Die homing_method kann nur verstellt werden, wenn die Referenzfahrt nicht aktiv ist. Ansonsten wird eine Fehlermeldung (→ Kapitel 3.5) zurückgegeben.

Der Ablauf der einzelnen Methoden ist in Kapitel 7.2.3 ausführlich erläutert.

Objekt 6099_h: homing_speeds

Dieses Objekt bestimmt die Geschwindigkeiten, die während der Referenzfahrt benutzt werden.

Index	6099_h
Name	homing_speeds
Object Code	ARRAY
No. of Elements	2
Data Type	UINT32

Sub-Index	01_h
Description	speed_during_search_for_switch
Access	rw
PDO Mapping	yes
Units	speed units
Value Range	–
Default Value	100 min ⁻¹

Sub-Index	02_h
Description	speed_during_search_for_zero
Access	rw
PDO Mapping	yes
Units	speed units
Value Range	–
Default Value	10 min ⁻¹



Wird Bit 6 im Objekt compatibility_control, (→ Kap. 5.2) gesetzt, wird nach der Referenzfahrt eine Fahrt auf Null durchgeführt.

Ist dieses Bit gesetzt und das Objekt speed_during_search_for_switch wird beschrieben, wird sowohl die Geschwindigkeit für die Schaltersuche, als auch die Geschwindigkeit für die Fahrt auf Null beschrieben.

Objekt 609A_h: homing_acceleration

Das Objekt homing_acceleration legt die Beschleunigung fest, die während der Referenzfahrt für alle Beschleunigungs- und Bremsvorgänge verwendet wird.

Index	609A_h
Name	homing_acceleration
Object Code	VAR
Data Type	UINT32

Access	rw
PDO Mapping	yes
Units	acceleration units
Value Range	–
Default Value	1000 min ⁻¹ /s

Objekt 2045_h: homing_timeout

Die Referenzfahrt kann auf ihre maximale Ausführungszeit überwacht werden. Dazu kann mit dem Objekt homing_timeout die maximale Ausführungszeit angegeben werden. Wird diese Zeit überschritten, ohne dass die Referenzfahrt beendet wurde, wird der Fehler 11-3 ausgelöst.

Index	2045_h
Name	homing_timeout
Object Code	VAR
Data Type	UINT16

Access	rw
PDO Mapping	no
Units	ms
Value Range	0 (aus), 1 ... 65535
Default Value	60000

7.2.3 Referenzfahrt-Abläufe

Die verschiedenen Referenzfahrt-Methoden sind in den folgenden Abbildungen dargestellt.

Methode 1: Negativer Endschalter mit Nullimpulsauswertung

Bei dieser Methode bewegt sich der Antrieb zunächst relativ schnell in negativer Richtung, bis er den negativen Endschalter erreicht. Dieses wird im Diagramm durch die steigende Flanke dargestellt. Danach fährt der Antrieb langsam zurück und sucht die genaue Position des Endschalters. Die Nullposition bezieht sich auf den ersten Nullimpuls des Winkelgebers in positiver Richtung vom Endschalter.

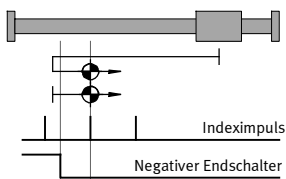


Fig. 7.3 Referenzfahrt auf den negativen Endschalter mit Auswertung des Nullimpulses

Methode 2: Positiver Endschalter mit Nullimpulsauswertung

Bei dieser Methode bewegt sich der Antrieb zunächst relativ schnell in positiver Richtung, bis er den positiven Endschalter erreicht. Dieses wird im Diagramm durch die steigende Flanke dargestellt. Danach fährt der Antrieb langsam zurück und sucht die genaue Position des Endschalters. Die Nullposition bezieht sich auf den ersten Nullimpuls des Winkelgebers in negativer Richtung vom Endschalter.

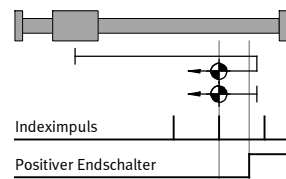


Fig. 7.4 Referenzfahrt auf den positiven Endschalter mit Auswertung des Nullimpulses

Methoden 7 u. 11: Referenzschalter und Nullimpulsauswertung

Diese beiden Methoden nutzen den Referenzschalter, der nur über einen Teil der Strecke aktiv ist. Diese Referenzmethoden bieten sich besonders für Rundachsen-Applikationen an, wo der Referenzschalter einmal pro Umdrehung aktiviert wird.

Bei der Methode 7 bewegt sich der Antrieb zunächst in positiver und bei Methode 11 in negativer Richtung. Abhängig von der Fahrtrichtung bezieht sich die Nullposition auf den ersten Nullimpuls in nega-

tiver oder positiver Richtung vom Referenzschalter. Dieses ist in den beiden folgenden Abbildungen ersichtlich.

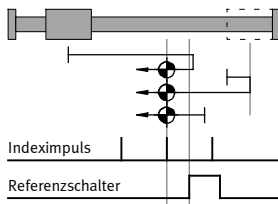


Fig. 7.5 Referenzfahrt auf den Referenzschalter mit Auswertung des Nullimpulses bei positiver Anfangsbewegung



Bei Referenzfahrten auf den Referenzschalter dienen die Endschalter zunächst zur Suchrichtungsumkehr. Wird im Anschluss der gegenüberliegende Endschalter erreicht, wird ein Fehler ausgelöst.

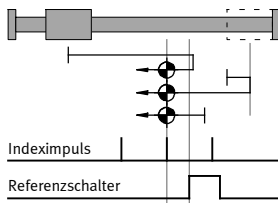


Fig. 7.6 Referenzfahrt auf den Referenzschalter mit Auswertung des Nullimpulses bei negativer Anfangsbewegung

Methode 17: Referenzfahrt auf den negativen Endschalter

Bei dieser Methode bewegt sich der Antrieb zunächst relativ schnell in negativer Richtung, bis er den negativen Endschalter erreicht. Dieses wird im Diagramm durch die steigende Flanke dargestellt. Danach fährt der Antrieb langsam zurück und sucht die genaue Position des Endschalters. Die Nullposition bezieht sich auf die fallende Flanke vom negativen Endschalter.

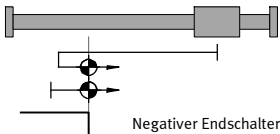


Fig. 7.7 Referenzfahrt auf den negativen Endschalter

Methode 18: Referenzfahrt auf den positiven Endschalter

Bei dieser Methode bewegt sich der Antrieb zunächst relativ schnell in positiver Richtung, bis er den positiven Endschalter erreicht. Dieses wird im Diagramm durch die steigende Flanke dargestellt. Da-

nach fährt der Antrieb langsam zurück und sucht die genaue Position des Endschalters. Die Nullposition bezieht sich auf die fallende Flanke vom positiven Endschalter.

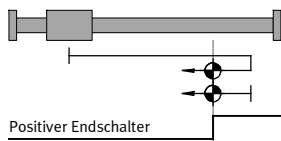


Fig. 7.8 Referenzfahrt auf den positiven Endschalter

Methoden 23 und 27: Referenzfahrt auf den Referenzschalter

Diese beiden Methoden nutzen den Referenzschalter, der nur über einen Teil der Strecke aktiv ist. Diese Referenzmethode bietet sich besonders für Rundachsen-Applikationen an, wo der Referenzschalter einmal pro Umdrehung aktiviert wird.

Bei der Methode 23 bewegt sich der Antrieb zunächst in positiver und bei Methode 27 in negativer Richtung. Die Nullposition bezieht sich auf die Flanke vom Referenzschalter. Dieses ist in den beiden folgenden Abbildungen ersichtlich.

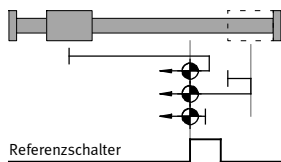


Fig. 7.9 Referenzfahrt auf den Referenzschalter bei positiver Anfangsbewegung



Bei Referenzfahrten auf den Referenzschalter dienen die Endschalter zunächst zur Suchrichtungsumkehr. Wird im Anschluss der gegenüberliegende Endschalter erreicht, wird ein Fehler ausgelöst.

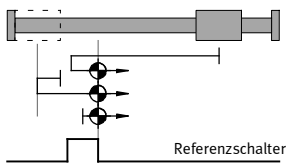


Fig. 7.10 Referenzfahrt auf den Referenzschalter bei negativer Anfangsbewegung

Methode –1: negativer Anschlag mit Nullimpulsauswertung

Bei dieser Methode bewegt sich der Antrieb in negativer Richtung, bis er den Anschlag erreicht. Hierbei steigt das I^2t -Integral des Motors auf maximal 90%. Der Anschlag muss mechanisch so dimensioniert sein, dass er bei dem parametrisierten Maximalstrom keinen Schaden nimmt. Die Nullposition bezieht sich auf den ersten Nullimpuls des Winkelgebers in positiver Richtung vom Anschlag.

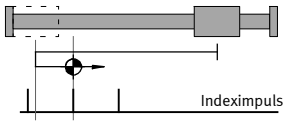


Fig. 7.11 Referenzfahrt auf den negativen Anschlag mit Auswertung des Nullimpulses

Methode –2: positiver Anschlag mit Nullimpulsauswertung

Bei dieser Methode bewegt sich der Antrieb in positiver Richtung, bis er den Anschlag erreicht. Hierbei steigt das I^2t -Integral des Motors auf maximal 90%. Der Anschlag muss mechanisch so dimensioniert sein, dass er bei dem parametrisierten Maximalstrom keinen Schaden nimmt. Die Nullposition bezieht sich auf den ersten Nullimpuls des Winkelgebers in negativer Richtung vom Anschlag.

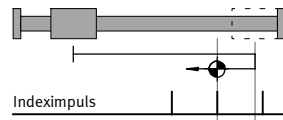


Fig. 7.12 Referenzfahrt auf den positiven Anschlag mit Auswertung des Nullimpulses

Methode –17: Referenzfahrt auf den negativen Anschlag

Bei dieser Methode bewegt sich der Antrieb in negativer Richtung, bis er den Anschlag erreicht. Hierbei steigt das I^2t -Integral des Motors auf maximal 90%. Der Anschlag muss mechanisch so dimensioniert sein, dass er bei dem parametrisierten Maximalstrom keinen Schaden nimmt. Die Nullposition bezieht sich direkt auf den Anschlag.

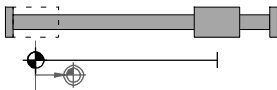


Fig. 7.13 Referenzfahrt auf den negativen Anschlag

Methode –18: Referenzfahrt auf den positiven Anschlag

Bei dieser Methode bewegt sich der Antrieb in positiver Richtung, bis er den Anschlag erreicht. Hierbei steigt das I^2t -Integral des Motors auf maximal 90%. Der Anschlag muss mechanisch so dimensioniert sein, dass er bei dem parametrisierten Maximalstrom keinen Schaden nimmt. Die Nullposition bezieht sich direkt auf den Anschlag.

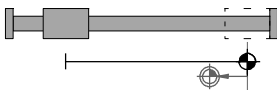


Fig. 7.14 Referenzfahrt auf den positiven Anschlag

Methoden 33: Referenzfahrt in negative Richtung auf den Nullimpuls

Bei der Methoden 33 ist die Richtung der Referenzfahrt negativ. Die Nullposition bezieht sich auf den ersten Nullimpuls vom Winkelgeber in Suchrichtung.

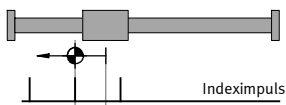


Fig. 7.15 Referenzfahrt in negative Richtung auf den Nullimpuls

Methoden 34: Referenzfahrt in positive Richtung auf den Nullimpuls

Bei der Methoden 34 ist die Richtung der Referenzfahrt positiv. Die Nullposition bezieht sich auf den ersten Nullimpuls vom Winkelgeber in Suchrichtung.

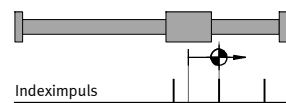


Fig. 7.16 Referenzfahrt in positive Richtung auf den Nullimpuls

Methode 35: Referenzfahrt auf die aktuelle Position

Bei der Methode 35 wird die Nullposition auf die aktuelle Position bezogen.

Soll der Antrieb nicht neu referenziert werden, sondern lediglich die Position auf einen vorgegebenen Wert gesetzt werden, kann das Objekt 2030_h (set_position_absolute) benutzt werden. → hierzu Seite 117.

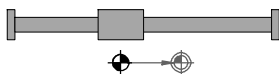


Fig. 7.17 Referenzfahrt auf aktuelle Position

7.2.4 Steuerung der Referenzfahrt

Die Referenzfahrt wird durch das controlword / statusword gesteuert und überwacht. Das Starten erfolgt durch Setzen des Bit 4 im controlword. Der erfolgreiche Abschluss der Fahrt wird durch ein gesetztes Bit 12 im Objekt statusword angezeigt. Ein gesetztes Bit 13 im Objekt statusword zeigt an, dass während der Referenzfahrt ein Fehler aufgetreten ist. Die Fehlerursache kann über die Objekte error_register und pre_defined_error_field bestimmt werden.

Bit 4	Bedeutung
1	Referenzfahrt ist nicht aktiv
0 → 1	Referenzfahrt starten
1	Referenzfahrt ist aktiv
1 → 0	Referenzfahrt unterbrechen

Tab. 7.1 Beschreibung der Bits im controlword

Bit 13	Bit 12	Bedeutung
0	0	Referenzfahrt ist noch nicht fertig
0	1	Referenzfahrt erfolgreich durchgeführt
1	0	Referenzfahrt nicht erfolgreich durchgeführt
1	1	verbotener Zustand

Tab. 7.2 Beschreibung der Bits im statusword

7.3 Betriebsart Positionieren (Profile Position Mode)

7.3.1 Übersicht

Die Struktur dieser Betriebsart wird in Fig. 7.18 ersichtlich:

Die Zielposition (target_position) wird dem Fahrkurven-Generator übergeben. Dieser erzeugt einen Lage-Sollwert (position_demand_value) für den Lageregler, der in dem Kapitel Lageregler beschrieben wird (Position Control Function, Kapitel 6). Diese zwei Funktionsblöcke können unabhängig voneinander eingestellt werden.

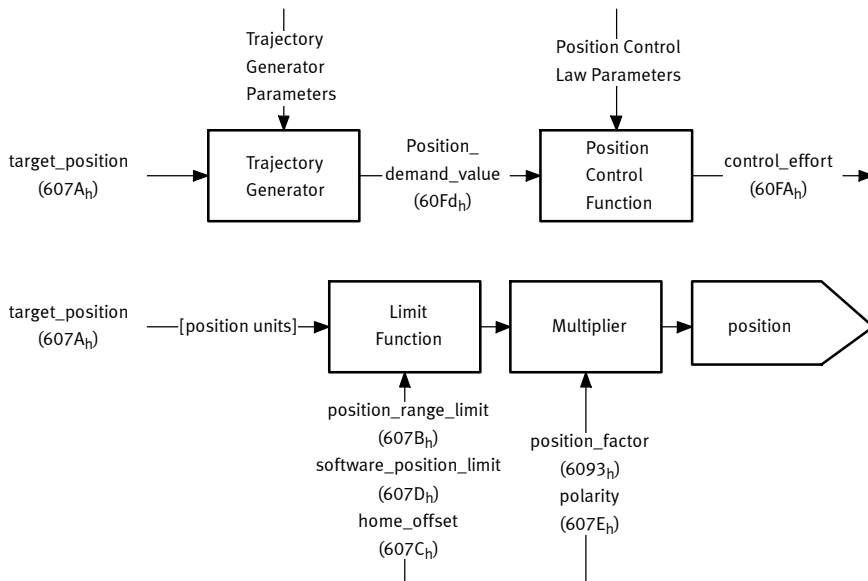


Fig. 7.18 Fahrkurven-Generator und Lageregler

Alle Eingangsgrößen des Fahrkurven-Generators werden mit den Größen der Factor-Group (➔ Kap. 5.3) in die internen Einheiten des Reglers umgerechnet. Die internen Größen werden hier mit einem Sternchen gekennzeichnet und werden vom Anwender in der Regel nicht benötigt.

7.3.2 Beschreibung der Objekte

In diesem Kapitel behandelte Objekte

Index	Objekt	Name	Typ	Attr.
607A _h	VAR	target_position	INT32	rw
6081 _h	VAR	profile_velocity	UINT32	rw
6082 _h	VAR	end_velocity	UINT32	rw
6083 _h	VAR	profile_acceleration	UINT32	rw
6084 _h	VAR	profile_deceleration	UINT32	rw
6085 _h	VAR	quick_stop_deceleration	UINT32	rw
6086 _h	VAR	motion_profile_type	INT16	rw

Betroffene Objekte aus anderen Kapiteln

Index	Objekt	Name	Typ	Kapitel
6040 _h	VAR	controlword	INT16	6 Gerätesteuerung
6041 _h	VAR	statusword	UINT16	6 Gerätesteuerung
605A _h	VAR	quick_stop_option_code	INT16	6 Gerätesteuerung
607E _h	VAR	polarity	UINT8	5.3 Umrechnungsfaktoren
6093 _h	ARRAY	position_factor	UINT32	5.3 Umrechnungsfaktoren
6094 _h	ARRAY	velocity_encoder_factor	UINT32	5.3 Umrechnungsfaktoren
6097 _h	ARRAY	acceleration_factor	UINT32	5.3 Umrechnungsfaktoren

Objekt 607A_h: target_position

Das Objekt target_position (Zielposition) bestimmt, an welche Position der Motorcontroller fahren soll. Dabei muss die aktuelle Einstellung der Geschwindigkeit, der Beschleunigung, der Bremsverzögerung und die Art des Fahrprofils (motion_profile_type) etc. berücksichtigt werden. Die Zielposition (target_position) wird entweder als absolute oder relative Angabe interpretiert (controlword, Bit 6).

Index	607A _h
Name	target_position
Object Code	VAR
Data Type	INT32

Access	rw
PDO Mapping	yes
Units	position units
Value Range	–
Default Value	0

Objekt 6081_h: profile_velocity

Das Objekt profile_velocity gibt die Geschwindigkeit an, die normalerweise während einer Positionierung am Ende der Beschleunigungsrampe erreicht wird. Das Objekt profile_velocity wird in speed units angegeben.

Index	6081_h
Name	profile_velocity
Object Code	VAR
Data Type	UINT32

Access	rw
PDO Mapping	yes
Units	speed units
Value Range	–
Default Value	1000

Objekt 6082_h: end_velocity

Das Objekt end_velocity (Endgeschwindigkeit) definiert die Geschwindigkeit, die der Antrieb haben muss, wenn er die Zielposition (target_position) erreicht. Normalerweise ist dieses Objekt auf Null zu setzen, damit der Motorcontroller beim Erreichen der Zielposition (target_position) stoppt. Für lückenlose Positionierungen kann eine von Null abweichende Geschwindigkeit vorgegeben werden. Das Objekt end_velocity wird in denselben Einheiten wie das Objekt profile_velocity angegeben.

Index	6082_h
Name	end_velocity
Object Code	VAR
Data Type	UINT32

Access	rw
PDO Mapping	yes
Units	speed units
Value Range	–
Default Value	0

Objekt 6083_h: profile_acceleration

Das Objekt profile_acceleration gibt die Beschleunigung an, mit der auf den Sollwert beschleunigt. Es wird in benutzerdefinierten Beschleunigungseinheiten (acceleration units) angegeben (→ Kapitel 5.3 Umrechnungsfaktoren (Factor Group)).

Index	6083_h
Name	profile_acceleration
Object Code	VAR
Data Type	UINT32

Access	rw
PDO Mapping	yes
Units	acceleration units
Value Range	–
Default Value	10000 min ⁻¹ /s

Objekt 6084_h: profile_deceleration

Das Objekt profile_deceleration gibt die Beschleunigung an, mit der gebremst wird. Es wird in benutzerdefinierten Beschleunigungseinheiten (acceleration units) angegeben (➔ Kapitel 5.3 Umrechnungsfaktoren (Factor Group)).

Index	6084_h
Name	profile_deceleration
Object Code	VAR
Data Type	UINT32

Access	rw
PDO Mapping	yes
Units	acceleration units
Value Range	–
Default Value	10000 min ⁻¹ /s

Objekt 6085_h: quick_stop_deceleration

Das Objekt quick_stop_deceleration gibt an, mit welcher Bremsverzögerung der Motor stoppt, wenn ein Quick Stop ausgeführt wird (➔ Kapitel 6). Das Objekt quick_stop_deceleration wird in derselben Einheit wie das Objekt profile_deceleration angegeben.

Index	6085_h
Name	quick_stop_deceleration
Object Code	VAR
Data Type	UINT32

Access	rw
PDO Mapping	yes
Units	acceleration units
Value Range	–
Default Value	14100 min ⁻¹ /s

Objekt 6086_h: motion_profile_type

Das Objekt motion_profile_type wird verwendet, um die Art des Positionierprofils auszuwählen.

Index	6086_h
Name	motion_profile_type
Object Code	VAR
Data Type	INT16

Access	rw
PDO Mapping	yes
Units	–
Value Range	0, 2
Default Value	0

Wert	Kurvenform
0	Lineare Rampe
2	Ruckfreie Rampe

7.3.3 Funktionsbeschreibung

Es gibt zwei Möglichkeiten eine Zielposition an den Motorcontroller zu übergeben:

Einfacher Fahrauftrag

Wenn der Motorcontroller eine Zielposition erreicht hat, signalisiert er dies dem Host mit dem Bit target_reached (Bit 10 im Objekt statusword). In dieser Betriebsart stoppt der Motorcontroller, wenn er das Ziel erreicht hat.

Folge von Fahraufträgen

Nachdem der Motorcontroller ein Ziel erreicht hat, beginnt er sofort das nächste Ziel anzufahren. Dieser Übergang kann fließend erfolgen, ohne dass der Motorcontroller zwischendurch zum Stillstand kommt. Diese beiden Methoden werden durch die Bits new_set_point und change_set_immediatly in dem Objekt controlword und set_point_acknowledge in dem Objekt statusword kontrolliert. Diese Bits stehen in einem Frage-Antwort-Verhältnis zueinander. Hierdurch wird es möglich, einen Fahrauftrag vorzubereiten, während ein anderer noch läuft.

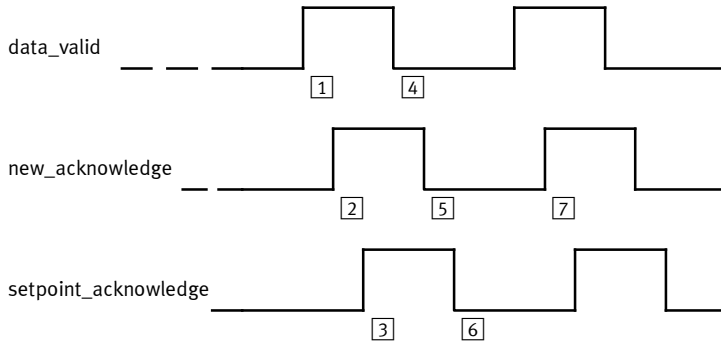


Fig. 7.19 Fahrauftrag-Übertragung von einem Host

In Fig. 7.19 können Sie sehen, wie der Host und der Motorcontroller über den CAN-Bus miteinander kommunizieren:

Zuerst werden die Positionierdaten (Zielposition, Fahrgeschwindigkeit, Endgeschwindigkeit und die Beschleunigung) an den Motorcontroller übertragen. Wenn der Positionierdatensatz vollständig eingeschrieben ist [1], kann der Host die Positionierung starten, indem er das Bit `new_set_point` im `controlword` auf „1“ setzt [2]. Nachdem der Motorcontroller die neuen Daten erkannt und in seinen Puffer übernommen hat, meldet er dies dem Host durch das Setzen des Bits `set_point_acknowledge` im `statusword` [3].

Daraufhin kann der Host beginnen, einen neuen Positionierdatensatz in den Motorcontroller einzuschreiben [4] und das Bit `new_set_point` wieder zu löschen [5]. Erst wenn der Motorcontroller einen neuen Fahrauftrag akzeptieren kann [6], signalisiert er dies durch eine „0“ im `set_point_acknowledge`-Bit. Vorher darf vom Host keine neue Positionierung gestartet werden [7].

In Fig. 7.20 wird eine neue Positionierung erst gestartet, nachdem die vorherige vollständig abgeschlossen wurde. Der Host wertet hierzu das Bit `target_reached` im Objekt `statusword` aus.

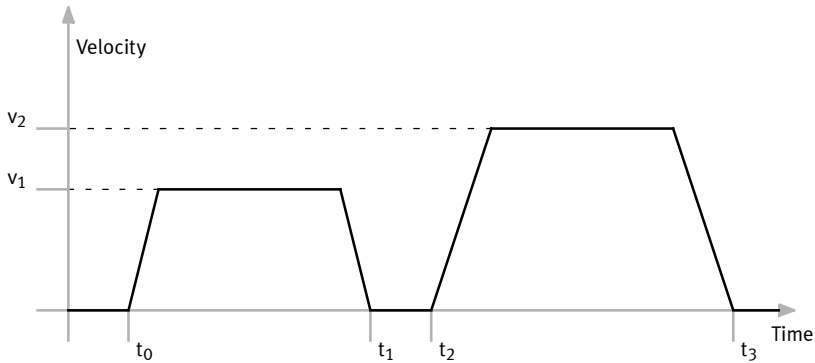


Fig. 7.20 Einfacher Fahrauftrag

In Fig. 7.21 wird eine neue Positionierung bereits gestartet, während sich die Vorherige noch in Bearbeitung befindet. Der Host übergibt hierzu dem Motorcontroller das nachfolgende Ziel schon dann, wenn dieser mit dem Löschen des Bits `set_point_acknowledge` signalisiert, dass er den Puffer gelesen und die zugehörige Positionierung gestartet hat. Die Positionierungen werden auf diese Weise nahtlos aneinander gereiht. Damit der Motorcontroller zwischen den einzelnen Positionierungen nicht jedes Mal kurzzeitig auf Null abbremsst, sollte für diese Betriebsart das Objekt `end_velocity` mit dem gleichen Wert wie das Objekt `profile_velocity` beschrieben werden.

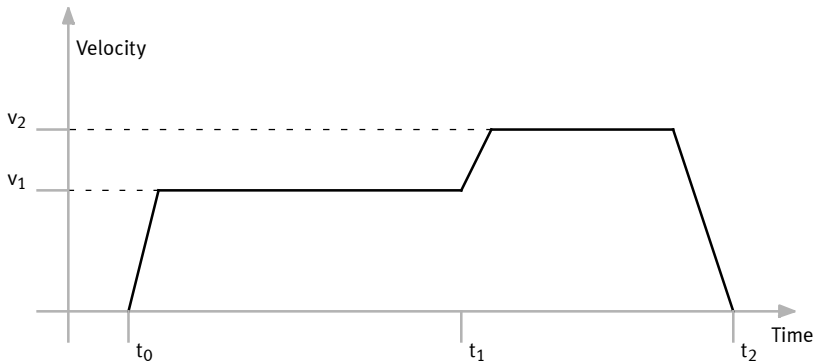


Fig. 7.21 Lückenlose Folge von Fahraufträgen

Wenn im `controlword` neben dem Bit `new_set_point` auch das Bit `change_set_immediately` auf „1“ gesetzt wird, weist der Host den Motorcontroller damit an, sofort den neuen Fahrauftrag zu beginnen. Ein bereits in Bearbeitung befindlicher Fahrauftrag wird in diesem Fall abgebrochen.

7.4 **Synchrone Positionsvorgabe (Interpolated Position Mode)**

7.4.1 **Übersicht**

Der Interpolated Position Mode (IP) ermöglicht die Vorgabe von Lagesollwerten in einer mehrachsigen Anwendung des Motorcontrollers. Dazu werden in einem festen Zeitraster (Synchronisations-Intervall) Synchronisations-Telegramme (SYNC) und Lagesollwerte von einer übergeordneten Steuerung vorgegeben. Da in der Regel das Intervall größer als ein Lagereglerzyklus ist, interpoliert der Motorcontroller selbständig die Datenwerte zwischen zwei vorgegebenen Positionswerten, wie in der folgenden Grafik skizziert.

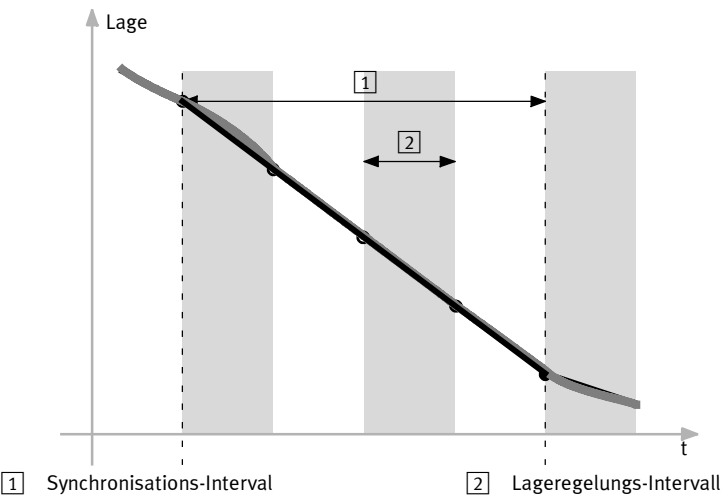


Fig. 7.22 Fahrauftrag Lineare Interpolation zwischen zwei Datenwerten

Im Folgenden sind zunächst die für den interpolated position mode benötigten Objekte beschrieben. In einer anschließenden Funktionsbeschreibung wird umfassend auf die Aktivierung und die Reihenfolge der Parametrierung eingegangen.

7.4.2 **Beschreibung der Objekte**

In diesem Kapitel behandelte Objekte

Index	Objekt	Name	Typ	Attr.
60C0 _h	VAR	interpolation_submode_select	INT16	rw
60C1 _h	REC	interpolation_data_record		rw
60C2 _h	REC	interpolation_time_period		rw
60C3 _h	ARRAY	interpolation_sync_definition	UINT8	rw
60C4 _h	REC	interpolation_data_configuration		rw

Betroffene Objekte aus anderen Kapiteln

Index	Objekt	Name	Typ	Kapitel
6040 _h	VAR	controlword	INT16	6 Gerätesteuerung
6041 _h	VAR	statusword	UINT16	6 Gerätesteuerung
6093 _h	ARRAY	position_factor	UINT32	5.3 Umrechnungsfaktoren
6094 _h	ARRAY	velocity_encoder_factor	UINT32	5.3 Umrechnungsfaktoren
6097 _h	ARRAY	acceleration_factor	UINT32	5.3 Umrechnungsfaktoren

Objekt 60C0_h: interpolation_submode_select

Über das Objekt interpolation_submode_select wird der Typ der Interpolation festgelegt. Zur Zeit ist nur die herstellerspezifische Variante „Lineare Interpolation ohne Puffer“ verfügbar.

Index	60C0_h
Name	interpolation_submode_select
Object Code	VAR
Data Type	INT16

Access	rw
PDO Mapping	yes
Units	–
Value Range	-2
Default Value	-2

Wert	Interpolationstyp
-2	Lineare Interpolation ohne Puffer

Objekt 60C1_h: interpolation_data_record

Der Objekt-Record interpolation_data_record repräsentiert den eigentlichen Datensatz. Er besteht aus einem Eintrag für den Lagewert (ip_data_position) und einem Steuerwort (ip_data_controlword), welches angibt, ob der Lagewert absolut oder relativ zu interpretieren ist. Die Angabe des Steuerworts ist optional. Wird er nicht angegeben, wird der Lagewert als absolut interpretiert. Soll das Steuerwort mit angegeben werden, muss aus Gründen der Datenkonsistenz zuerst Subindex 2 (ip_data_controlword) und anschließend Subindex 1 (ip_data_position) geschrieben werden, da intern die Datenübernahme mit Schreibzugriff auf ip_data_position ausgelöst wird.

Index	60C1_h
Name	interpolation_data_record
Object Code	RECORD
No. of Elements	2

Sub-Index	01_h
Description	ip_data_position
Data Type	INT32
Access	rw
PDO Mapping	yes
Units	position units
Value Range	–
Default Value	–

Sub-Index	02_h
Description	ip_data_controlword
Data Type	UINT8
Access	rw
PDO Mapping	yes
Units	–
Value Range	0, 1
Default Value	0

Wert	ip_data_controlword
0	Absolute Position
1	Relative Entfernung



Die interne Datenübernahme erfolgt bei Schreibzugriff auf Subindex 1. Soll außerdem Subindex 2 verwendet werden, muss dieser vor Subindex 1 beschrieben werden.

Objekt 60C2_h: interpolation_time_period

Über den Objekt-Record interpolation_time_period kann das Synchronisations-Intervall eingestellt werden. Über ip_time_index wird die Einheit (ms oder 1/10 ms) des Intervalls festgelegt, welches über ip_time_units parametrisiert wird. Zur Synchronisation wird die komplette Reglerkaskade (Strom-, Drehzahl- und Lageregler) auf den externen Takt aufsynchroisiert. Die Änderung des Synchronisationsintervalls wird daher nur nach einem Reset wirksam. Soll das Interpolationsintervall über den CAN-Bus geändert werden, muss daher der Parametersatz gesichert (→ Kapitel 5.1) und ein Reset ausgeführt werden (→ Kapitel 6), damit das neue Synchronisations-Intervall wirksam wird. Das Synchronisations-Intervall muss exakt eingehalten werden.

Index	60C2_h
Name	interpolation_time_period
Object Code	RECORD
No. of Elements	2

Sub-Index	01_h
Description	ip_time_units
Data Type	UINT8
Access	rw
PDO Mapping	yes
Units	gemäß ip_time_index
Value Range	ip_time_index = -3: 1, 2 ... 9, 10 ip_time_index = -4: 10, 20 ... 90, 100
Default Value	--

Sub-Index	02_h
Description	ip_time_index
Data Type	INT8
Access	rw
PDO Mapping	yes
Units	–
Value Range	-3, -4
Default Value	-3

Wert	ip_time_units wird angegeben in
-3	10 ⁻³ Sekunden (ms)
-4	10 ⁻⁴ Sekunden (0,1 ms)



Die Änderung des Synchronisationsintervalls wird nur nach einem Reset wirksam. Soll das Interpolationsintervall über den CAN-Bus geändert werden, muss der Parametersatz gesichert und ein Reset ausgeführt werden.

Objekt 60C3_h: interpolation_sync_definition

Über das Objekt interpolation_sync_definition wird die Art (synchronize_on_group) und die Anzahl (ip_sync_every_n_event) von Synchronisations-Telegrammen pro Synchronisations-Intervall vorgegeben. Für die CMMP-Reihe kann nur das Standard-SYNC-Telegramm und 1 SYNC pro Intervall eingestellt werden.

Index	60C3_h
Name	interpolation_sync_definition
Object Code	ARRAY
No. of Elements	2
Data Type	UINT8

Sub-Index	01_h
Description	synchronize_on_group
Access	rw
PDO Mapping	yes
Units	–
Value Range	0
Default Value	0

Wert	Bedeutung
0	Standard SYNC-Telegramm verwenden

Sub-Index	02_h
Description	ip_sync_every_n_event
Access	rw
PDO Mapping	yes
Units	–
Value Range	1
Default Value	1

Objekt 60C4_h: interpolation_data_configuration

Über den Objekt-Record interpolation_data_configuration kann die Art (buffer_organisation) und Größe (max_buffer_size, actual_buffer_size) eines eventuell vorhandenen Puffers sowie der Zugriff auf diesen (buffer_position, buffer_clear) konfiguriert werden. Über das Objekt size_of_data_record kann die Größe eines Puffer-Elements ausgelesen werden. Obwohl bei der Interpolationsart „Lineare Interpolation ohne Puffer“ kein Puffer zur Verfügung steht, muss der Zugriff über das Objekt buffer_clear allerdings auch in diesem Fall freigegeben werden.

Index	60C4_h
Name	interpolation_data_configuration
Object Code	RECORD
No. of Elements	6

Sub-Index	01_h
Description	max_buffer_size
Data Type	UINT32
Access	ro
PDO Mapping	no
Units	–
Value Range	0
Default Value	0

Sub-Index	02_h
Description	actual_size
Data Type	UINT32
Access	rw
PDO Mapping	yes
Units	–
Value Range	0 ... max_buffer_size
Default Value	0

Sub-Index	03_h
Description	buffer_organisation
Data Type	UINT8
Access	rw
PDO Mapping	yes
Units	–
Value Range	0
Default Value	0

Wert	Bedeutung
0	FIFO

Sub-Index	04_h
Description	buffer_position
Data Type	UINT16
Access	rw
PDO Mapping	yes
Units	–
Value Range	0
Default Value	0

Sub-Index	05_h
Description	size_of_data_record
Data Type	UINT8
Access	wo
PDO Mapping	yes
Units	–
Value Range	2
Default Value	2

Sub-Index	06_h
Description	buffer_clear
Data Type	UINT8
Access	wo
PDO Mapping	yes
Units	–
Value Range	0, 1
Default Value	0

Wert	Bedeutung
0	Puffer löschen/Zugriff auf 60C1 _h nicht erlaubt
1	Zugriff auf 60C1 _h freigegeben

7.4.3 Funktionsbeschreibung

Vorbereitende Parametrierung

Bevor der Motorcontroller in die Betriebsart interpolated position mode geschaltet werden kann, müssen diverse Einstellungen vorgenommen werden: Dazu zählen die Einstellung des Interpolations-Intervalls (interpolation_time_period), also der Zeit zwischen zwei SYNC-Telegrammen, der Interpolationstyp (interpolation_submode_select) und die Art der Synchronisation (interpolation_sync_definition). Zusätzlich muss der Zugriff auf den Positionspuffer über das Objekt buffer_clear freigegeben werden.

BEISPIEL				
Aufgabe	CAN-Objekt/COB			
Interpolationsart	-2	60C0 _h , interpolation_submode_select	=	-2
Zeiteinheit	0,1 ms	60C2 _h _02 _h , interpolation_time_index	=	-4
Zeitintervall	4 ms	60C2 _h _01 _h , interpolation_time_units	=	40
Parameter sichern		1010 _h _01 _h , save_all_parameters		
Reset ausführen		NMT reset node		
Warten auf Bootup		Bootup-Nachricht		
Puffer-Freigabe	1	60C4 _h _06 _h , buffer_clear	=	1
SYNC erzeugen		SYNC (Raster 4 ms)		

Aktivierung des Interpolated Position Mode und Aufsynchronisation

Der IP wird über das Objekt modes_of_operation (6060_h) aktiviert. Ab diesem Zeitpunkt versucht der Motorcontroller sich auf das externe Zeitraster, welches durch die SYNC-Telegrammen vorgegeben wird, aufzusynchronisieren. Konnte sich der Motorcontroller erfolgreich aufsynchronisieren, meldet er die Betriebsart interpolated position mode im Objekt modes_of_operation_display (6061_h). Während der Aufsynchronisation meldet der Motorcontroller ungültige Betriebsart (-1) zurück. Werden nach der erfolgten Aufsynchronisation die SYNC-Telegramme nicht im richtigen Zeitraster gesendet, wechselt der Motorcontroller zurück in die ungültige Betriebsart.

Ist die Betriebsart eingenommen, kann die Übertragung von Positionsdaten an den Antrieb beginnen. Sinnvollerweise liest dazu die übergeordnete Steuerung zunächst die aktuelle Istposition aus dem Regler aus und schreibt diese zyklisch als neuen Sollwert (interpolation_data_record) in den Motorcontroller. Über Handshake-Bits des controlword und des statusword wird die Übernahme der Daten durch den Motorcontroller aktiviert. Durch Setzen des Bits enable_ip_mode im controlword zeigt der Host an, dass mit der Auswertung der Lagedaten begonnen werden soll. Erst wenn der Motorcontroller über das Statusbit ip_mode_selected im statusword dieses quittiert, werden die Datensätze ausgewertet. Im Einzelnen ergibt sich daher folgende Zuordnung und der folgende Ablauf:

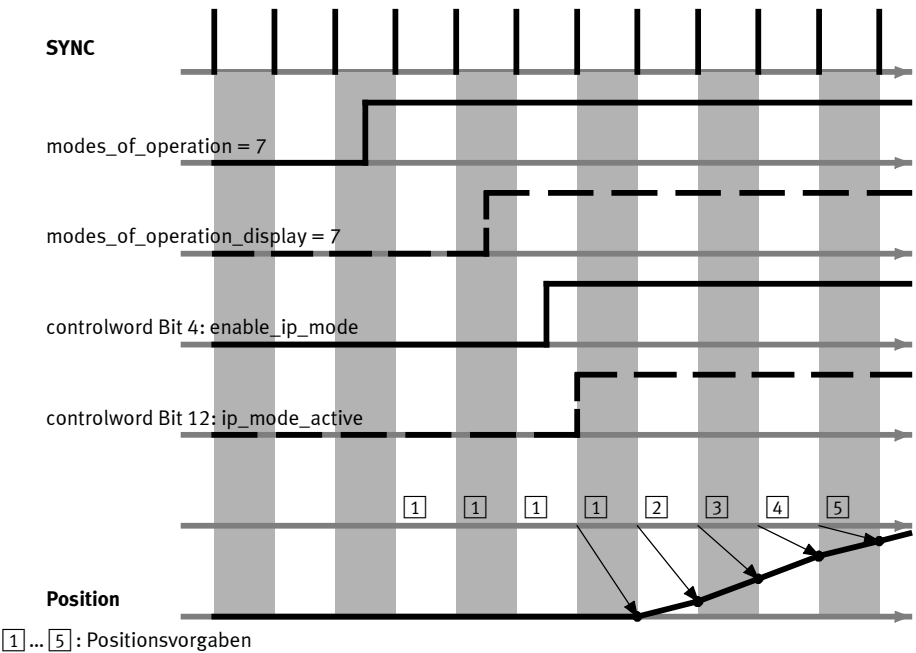


Fig. 7.23 Aufsynchrisation und Datenfreigabe

Ereignis	CAN-Objekt		
SYNC-Nachrichten erzeugen			
Anforderung der Betriebsart ip:	6060 _h , modes_of_operation	=	07
Warten bis Betriebsart eingenommen	6061 _h , modes_of_operation_display	=	07
Auslesen der akt. Istposition	6064 _h , position_actual_value		
Zurückschreiben als aktuelle Sollposition	60C1 _h .01 _h , ip_data_position		
Start der Interpolation	6040 _h , controlword, enable_ip_mode		
Quittierung durch Motorcontroller	6041 _h , statusword, ip_mode_active		
Ändern der aktuellen Sollposition gemäß Trajektorie	60C1 _h .01 _h , ip_data_position		

Nach Beendigung des synchronen Fahrvorgangs kann durch Löschen des Bits `enable_ip_mode` die weitere Auswertung von Lagewerten verhindert werden.

Anschließend kann gegebenenfalls in eine andere Betriebsart umgeschaltet werden.

Unterbrechung der Interpolation im Fehlerfall

Wird eine laufende Interpolation (`ip_mode_active` gesetzt) durch das Auftreten eines Controllerfehlers unterbrochen, verhält sich der Antrieb zunächst so, wie für den jeweiligen Fehler spezifiziert (z. B. Wegnahme der Reglerfreigabe und Wechsel in den Zustand `SWITCH_ON_DISABLED`).

Die Interpolation kann dann nur durch eine erneute Aufsynchronisation fortgesetzt werden, da der Motorcontroller wieder in den Zustand `OPERATION_ENABLE` gebracht werden muss, wodurch das Bit `ip_mode_active` gelöscht wird.

7.5 Betriebsart Drehzahlregelung (Profile Velocity Mode)

7.5.1 Übersicht

Der drehzahlgeregelte Betrieb (Profile Velocity Mode) beinhaltet die folgenden Unterfunktionen:

- Sollwert-Erzeugung durch den Rampen-Generator
 - Drehzahlerfassung über den Winkelgeber durch Differentiation
 - Drehzahlregelung mit geeigneten Eingabe- und Ausgabesignalen
 - Begrenzung des Drehmomenten-Sollwertes (`torque_demand_value`)
 - Überwachung der Ist-Geschwindigkeit (`velocity_actual_value`) mit der Fenster-Funktion/Schwelle
- Die Bedeutung der folgenden Parameter ist im Kapitel Positionieren (Profile Position Mode) beschrieben: `profile_acceleration`, `profile_deceleration`, `quick_stop`.

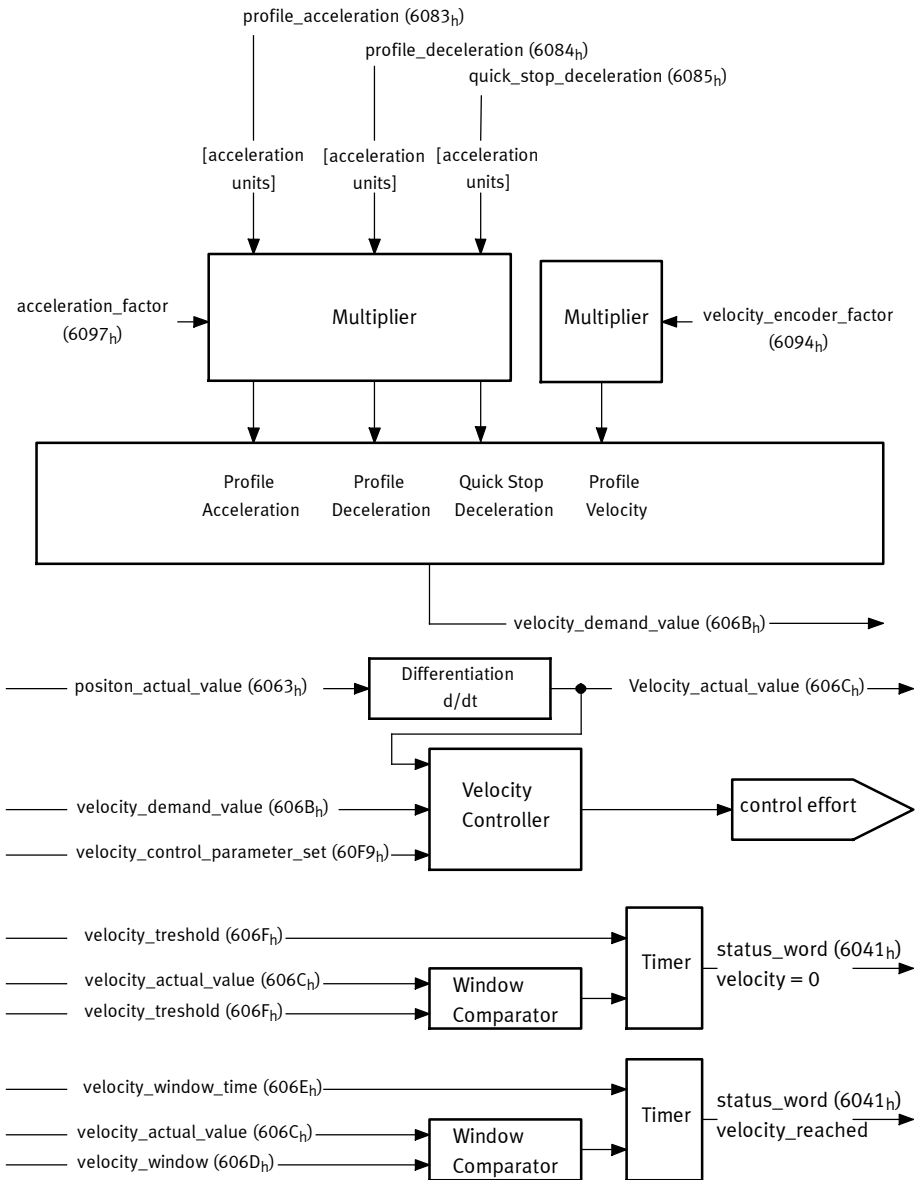


Fig. 7.24 Struktur des drehzahlgeregelten Betriebs (Profile Velocity Mode)

7.5.2 Beschreibung der Objekte

In diesem Kapitel behandelte Objekte

Index	Objekt	Name	Typ	Attr.
6069 _h	VAR	velocity_sensor_actual_value	INT32	ro
606A _h	VAR	sensor_selection_code	INT16	rw
606B _h	VAR	velocity_demand_value	INT32	ro
202E _h	VAR	velocity_demand_sync_value	INT32	ro
606C _h	VAR	velocity_actual_value	INT32	ro
606D _h	VAR	velocity_window	UINT16	rw
606E _h	VAR	velocity_window_time	UINT16	rw
606F _h	VAR	velocity_threshold	UINT16	rw
6080 _h	VAR	max_motor_speed	UINT32	rw
60FF _h	VAR	target_velocity	INT32	rw

Betroffene Objekte aus anderen Kapiteln

Index	Objekt	Name	Typ	Kapitel
6040 _h	VAR	controlword	INT16	6 Gerätesteuerung
6041 _h	VAR	statusword	UINT16	6 Gerätesteuerung
6063 _h	VAR	position_actual_value*	INT32	5.7 Lageregler
6071 _h	VAR	target_torque	INT16	7.7 Momentenregler
6072 _h	VAR	max_torque_value	UINT16	7.7 Momentenregler
607E _h	VAR	polarity	UINT8	5.3 Umrechnungsfaktoren
6083 _h	VAR	profile_acceleration	UINT32	7.3 Positionieren
6084 _h	VAR	profile_deceleration	UINT32	7.3 Positionieren
6085 _h	VAR	quick_stop_deceleration	UINT32	7.3 Positionieren
6086 _h	VAR	motion_profile_type	INT16	7.3 Positionieren
6094 _h	ARRAY	velocity_encoder_factor	UINT32	5.3 Umrechnungsfaktoren

Objekt 6069_h: velocity_sensor_actual_value

Mit dem Objekt velocity_sensor_actual_value kann der Wert eines möglichen Geschwindigkeitsgebers in internen Einheiten ausgelesen werden. Bei der CMMP Familie kann kein separater Drehzahlgeber angeschlossen werden. Zur Bestimmung des Drehzahl-Istwertes sollte daher grundsätzlich das Objekt 606C_h verwendet werden.

Index	6069_h
Name	velocity_sensor_actual_value
Object Code	VAR
Data Type	INT32

Access	ro
PDO Mapping	yes
Units	U/4096 min
Value Range	–
Default Value	–

Objekt 606A_h: sensor_selection_code

Mit diesem Objekt kann der Geschwindigkeitssensor ausgewählt werden. Zur Zeit ist kein separater Geschwindigkeitssensor vorgesehen. Deshalb ist nur der standardmäßige Winkelgeber anwählbar.

Index	606A_h
Name	sensor_selection_code
Object Code	VAR
Data Type	INT16

Access	rw
PDO Mapping	yes
Units	–
Value Range	0
Default Value	0

Objekt 606B_h: velocity_demand_value

Mit diesem Objekt kann der aktuelle Drehzahlsollwert des Drehzahlreglers ausgelesen werden. Auf diesen wirkt der Sollwert vom Rampen-Generator bzw. des Fahrkurven-Generators. Bei aktiviertem Lagereger wird außerdem dessen Korrekturgeschwindigkeit addiert.

Index	606B_h
Name	velocity_demand_value
Object Code	VAR
Data Type	INT32

Access	ro
PDO Mapping	yes
Units	speed units
Value Range	–
Default Value	–

Objekt 202E_h: velocity_demand_sync_value

Über dieses Objekt kann die Soll-Drehzahl des Synchronisationsgeber ausgelesen werden. Diese wird durch das Objekt 2022_h synchronization_encoder_select (Kap. 5.11) definiert. Dieses Objekt wird in benutzerdefinierten Einheiten angegeben.

Index	202E_h
Name	velocity_demand_sync_value
Object Code	VAR
Data Type	INT32

Access	ro
PDO Mapping	no
Units	velocity units
Value Range	–
Default Value	–

Objekt 606C_h: velocity_actual_value

Über das Objekt velocity_actual_value kann der Drehzahl-Istwert ausgelesen werden.

Index	606C_h
Name	velocity_actual_value
Object Code	VAR
Data Type	INT32

Access	ro
PDO Mapping	yes
Units	speed units
Value Range	–
Default Value	–

Objekt 2074_h: velocity_actual_value_filtered

Über das Objekt velocity_actual_value_filtered kann ein gefilterter Drehzahl-Istwert ausgelesen werden, der allerdings nur zu Anzeigezwecken verwendet werden sollte.

Im Gegensatz zu velocity_actual_value wird velocity_actual_value_filtered nicht zur Regelung, wohl aber für den Durchdreheschutz des Reglers verwendet. Die Filterzeitkonstante kann über das Objekt 2073_h (velocity_display_filter_time) eingestellt werden. ➔ Objekt 2073_h: velocity_display_filter_time

Index	2074 _h
Name	velocity_actual_value_filtered
Object Code	VAR
Data Type	INT32

Access	ro
PDO Mapping	yes
Units	speed units
Value Range	–
Default Value	–

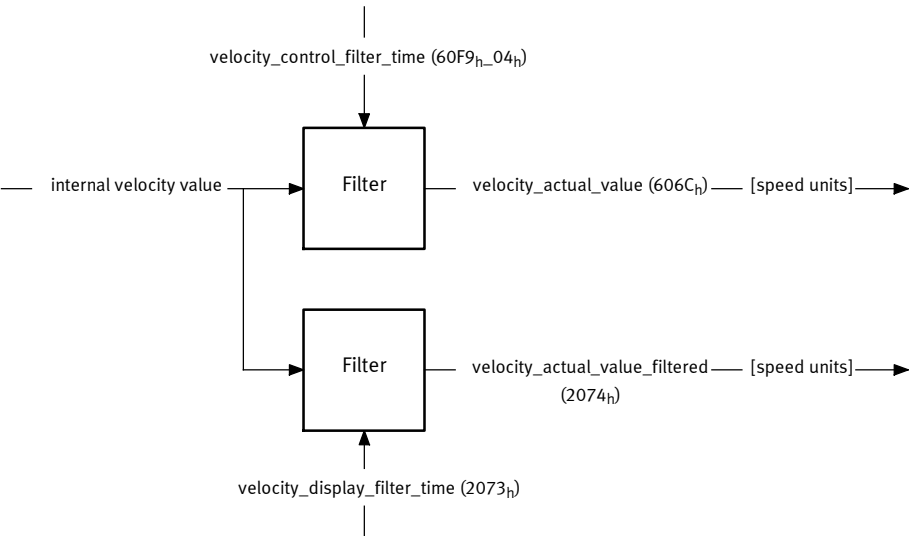


Fig. 7.25 Ermittlung von velocity_actual_value und velocity_actual_value_filtered

Objekt 606D_h: velocity_window

Das Objekt velocity_window dient zur Einstellung des Fensterkomparators. Dieser vergleicht den Drehzahl-Istwert mit der vorgegebenen Endgeschwindigkeit (Objekt 60FF_h: target_velocity). Ist die Differenz eine bestimmte Zeitdauer kleiner als hier angegeben, so wird das Bit 10 target_reached im Objekt statusword gesetzt. → auch: Objekt 606E_h (velocity_window_time).

Index	606D_h
Name	velocity_window
Object Code	VAR
Data Type	UINT16

Access	rw
PDO Mapping	yes
Units	speed units
Value Range	0 ... 65536 min ⁻¹
Default Value	4 min ⁻¹

Objekt 606E_h: velocity_window_time

Das Objekt velocity_window_time dient neben dem Objekt 606D_h: velocity_window der Einstellung des Fensterkomparators. Die Drehzahl muss die hier spezifizierte Zeit innerhalb des velocity_window liegen, damit das Bit 10 target_reached im Objekt statusword gesetzt wird.

Index	606E_h
Name	velocity_window_time
Object Code	VAR
Data Type	UINT16

Access	rw
PDO Mapping	yes
Units	ms
Value Range	0 ... 4999
Default Value	0

Objekt 606F_h: velocity_threshold

Das Objekt velocity_threshold gibt an, ab welchem Drehzahl-Istwert der Antrieb als stehend angesehen wird. Wenn der Antrieb den hier vorgegebenen Drehzahlwert für einen bestimmten Zeitraum überschreitet, wird im statusword das Bit 12 (velocity = 0) gelöscht. Der Zeitraum wird durch das Objekt velocity_threshold_time bestimmt.

Index	606F_h
Name	velocity_threshold
Object Code	VAR
Data Type	UINT16

Access	rw
PDO Mapping	yes
Units	speed units
Value Range	0 ... 65536 min ⁻¹
Default Value	10

Objekt 6070_h: velocity_threshold_time

Das Objekt velocity_threshold_time gibt an, wie lange der Antrieb den vorgegebenen Drehzahlwert überschreiten darf, bevor im statusword das Bit 12 (velocity = 0) gelöscht wird.

Index	6070_h
Name	velocity_threshold_time
Object Code	VAR
Data Type	UINT16

Access	rw
PDO Mapping	yes
Units	ms
Value Range	0 ... 4999
Default Value	0

Objekt 6080_h: max_motor_speed

Das Objekt max_motor_speed gibt die höchste erlaubte Drehzahl für den Motor in min⁻¹. Das Objekt wird benutzt, um den Motor zu schützen und kann dem Motordatenblatt entnommen werden. Der Drehzahl-Sollwert wird auf diesen Wert begrenzt.

Index	6080_h
Name	max_motor_speed
Object Code	VAR
Data Type	UINT16

Access	rw
PDO Mapping	yes
Units	min ⁻¹
Value Range	0 ... 32768 min ⁻¹
Default Value	32768 min ⁻¹

Objekt 60FF_h: target_velocity

Das Objekt target_velocity ist die Sollwertvorgabe für den Rampen-Generator.

Index	60FF _h
Name	target_velocity
Object Code	VAR
Data Type	INT32

Access	rw
PDO Mapping	yes
Units	speed units
Value Range	–
Default Value	–

7.6 Drehzahl-Rampen

Wird als modes_of_operation - profile_velocity_mode gewählt, wird grundsätzlich auch die Sollwertrampe aktiviert. Somit ist es möglich über die Objekte profile_acceleration und profile_deceleration eine sprungförmige Sollwertänderung auf eine bestimmte Drehzahländerungen pro Zeit zu begrenzen. Der Regler ermöglicht es, nicht nur unterschiedliche Beschleunigungen für Bremsen und Beschleunigungen anzugeben, sondern noch zusätzlich nach positiver und negativer Drehzahl zu unterscheiden. Die folgende Abbildung verdeutlicht dieses Verhalten:

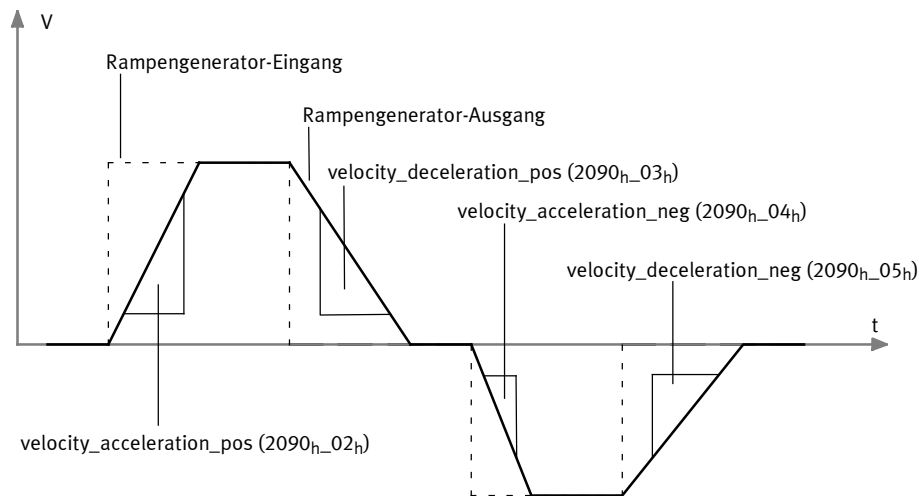


Fig. 7.26 Drehzahlrampen

Um diese 4 Beschleunigungen einzeln parametrieren zu können, ist die Objektgruppe velocity_ramps vorhanden. Es ist zu beachten, dass die Objekte profile_acceleration und profile_deceleration die glei-

chen internen Beschleunigungen verändern, wie die velocity_ramps. Wird die profile_acceleration geschrieben, werden gemeinsam velocity_acceleration_pos und velocity_acceleration_neg geändert, wird die profile_deceleration geschrieben, werden gemeinsam velocity_acceleration_pos und velocity_acceleration_neg geändert. Mit dem Objekt velocity_ramps_enable lässt sich festlegen, ob die Sollwerte über den Rampengenerator geführt werden, oder nicht.

Index	2090_h
Name	velocity_ramps
Object Code	RECORD
No. of Elements	5

Sub-Index	01_h
Description	velocity_ramps_enable
Data Type	UINT8
Access	rw
PDO Mapping	no
Units	–
Value Range	0: Sollwert NICHT über den Rampengenerator 1: Sollwert über den Rampengenerator
Default Value	1

Sub-Index	02_h
Description	velocity_acceleration_pos
Data Type	INT32
Access	rw
PDO Mapping	no
Units	acceleration units
Value Range	–
Default Value	14 100 min ⁻¹ /s

Sub-Index	03_h
Description	velocity_deceleration_pos
Data Type	INT32
Access	rw
PDO Mapping	no
Units	acceleration units
Value Range	–
Default Value	14 100 min ⁻¹ /s

Sub-Index	04_h
Description	velocity_acceleration_neg
Data Type	INT32
Access	rw
PDO Mapping	no
Units	acceleration units
Value Range	–
Default Value	14 100 min ⁻¹ /s

Sub-Index	05_h
Description	velocity_deceleration_neg
Data Type	INT32
Access	rw
PDO Mapping	no
Units	acceleration units
Value Range	–
Default Value	14 100 min ⁻¹ /s

7.7 Betriebsart Momentenregelung (Profile Torque Mode)

7.7.1 Übersicht

Dieses Kapitel beschreibt den drehmomentengeregelten Betrieb. Diese Betriebsart erlaubt es, dass dem Motorcontroller ein externer Momenten-Sollwert `target_torque` vorgegeben wird, welcher durch den integrierten Rampen-Generator geglättet werden kann. Somit ist es möglich, dass dieser Motorcontroller auch für Bahnsteuerungen eingesetzt werden kann, bei denen sowohl der Lageregler als auch der Drehzahlregler auf einen externen Rechner verlagert sind.

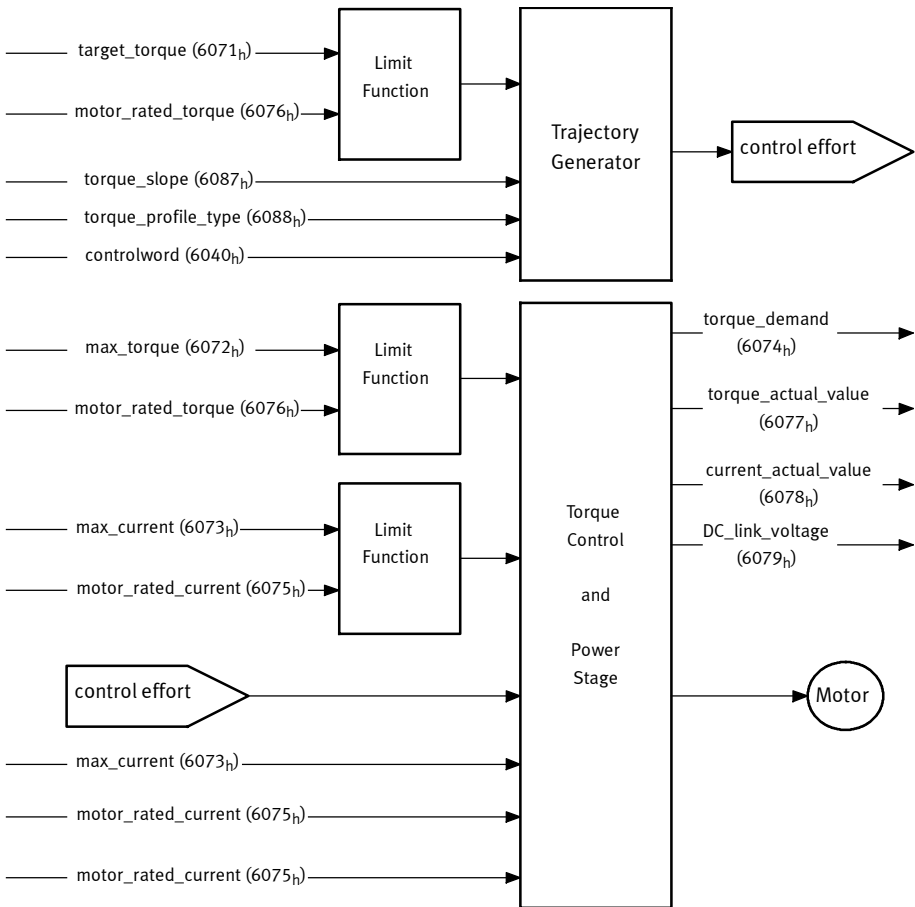


Fig. 7.27 Struktur des drehmomentengeregelten Betriebs

Für den Rampengenerator müssen die Parameter Rampensteilheit `torque_slope` und Rampenform `torque_profile_type` vorgegeben werden.

Wenn im `controlword` das Bit 8 halt gesetzt wird, senkt der Rampen-Generator das Drehmoment bis auf Null ab. Entsprechend erhöht er es wieder auf das Sollmoment `target_torque`, wenn das Bit 8 wieder gelöscht wird. In beiden Fällen berücksichtigt der Rampen-Generator die Rampensteilheit `torque_slope` und die Rampenform `torque_profile_type`.

Alle Definitionen innerhalb dieses Dokumentes beziehen sich auf drehbare Motoren. Wenn lineare Motoren benutzt werden, müssen sich alle „Drehmoment“-Objekte statt dessen auf eine „Kraft“ beziehen. Der Einfachheit halber sind die Objekte nicht doppelt vertreten und ihre Namen sollten nicht verändert werden.

Die Betriebsarten Positionierbetrieb (Profile Position Mode) und Drehzahlregler (Profile Velocity Mode) benötigen für ihre Funktion den Momentenregler. Deshalb ist es immer notwendig, diesen zu parametrieren.

7.7.2 Beschreibung der Objekte

In diesem Kapitel behandelte Objekte

Index	Objekt	Name	Typ	Attr.
6071 _h	VAR	<code>target_torque</code>	INT16	rw
6072 _h	VAR	<code>max_torque</code>	UINT16	rw
6074 _h	VAR	<code>torque_demand_value</code>	INT16	ro
6076 _h	VAR	<code>motorRated_torque</code>	UINT32	rw
6077 _h	VAR	<code>torque_actual_value</code>	INT16	ro
6078 _h	VAR	<code>current_actual_value</code>	INT16	ro
6079 _h	VAR	<code>DC_link_circuit_voltage</code>	UINT32	ro
6087 _h	VAR	<code>torque_slope</code>	UINT32	rw
6088 _h	VAR	<code>torque_profile_type</code>	INT16	rw
60F7 _h	RECORD	<code>power_stage_parameters</code>		rw
60F6 _h	RECORD	<code>torque_control_parameters</code>		rw

Betroffene Objekte aus anderen Kapiteln

Index	Objekt	Name	Typ	Kapitel
6040 _h	VAR	<code>controlword</code>	INT16	6 Gerätesteuerung (Device Control)
60F9 _h	RECORD	<code>motor_parameters</code>		5.5 Stromregler und Motoranpassung
6075 _h	VAR	<code>motorRated_current</code>	UINT32	5.5 Stromregler und Motoranpassung
6073 _h	VAR	<code>max_current</code>	UINT16	5.5 Stromregler und Motoranpassung

Objekt 6071_h: target_torque

Dieser Parameter ist im drehmomentengeregelten Betrieb (Profile Torque Mode) der Eingabewert für den Drehmomentenregler. Er wird in Tausendsteln des Nennmomentes (Objekt 6076_h) angegeben.

Index	6071_h
Name	target_torque
Object Code	VAR
Data Type	INT16

Access	rw
PDO Mapping	yes
Units	motorRatedTorque/1000
Value Range	-32768 ... 32768
Default Value	0

Objekt 6072_h: max_torque

Dieser Wert stellt das höchstzulässige Drehmoment des Motors dar. Es wird in Tausendsteln des Nennmomentes (Objekt 6076_h) angegeben. Wenn zum Beispiel kurzzeitig eine zweifache Überlastung des Motors zulässig ist, so ist hier der Wert 2000 einzutragen.



Das Objekt 6072_h: max_torque korrespondiert mit dem Objekt 6073_h: max_current und darf erst beschrieben werden, wenn zuvor das Objekt 6075_h: motorRatedCurrent mit einem gültigen Wert beschrieben wurde.

Index	6072_h
Name	max_torque
Object Code	VAR
Data Type	UINT16

Access	rw
PDO Mapping	yes
Units	motorRatedTorque/1000
Value Range	-1000 ... 65536
Default Value	2023

Objekt 6074_h: torque_demand_value

Über dieses Objekt kann das aktuelle Sollmoment in Tausendstel des Nennmoments (6076_h) ausgelesen werden. Berücksichtigt sind hierbei die internen Begrenzungen des Reglers (Stromgrenzwerte und I²t-Überwachung).

Index	6074_h
Name	torque_demand_value
Object Code	VAR
Data Type	INT16

Access	ro
PDO Mapping	yes
Units	motorRatedTorque/1000
Value Range	--
Default Value	--

Objekt 6076_h: motorRatedTorque

Dieses Objekt gibt das Nennmoment des Motors an. Dieses kann dem Typenschild des Motors entnommen werden. Es ist in der Einheit 0,001 Nm einzugeben.

Index	6076_h
Name	motorRatedTorque
Object Code	VAR
Data Type	UINT32

Access	rw
PDO Mapping	yes
Units	0,001 mNm
Value Range	–
Default Value	296

Objekt 6077_h: torque_actual_value

Über dieses Objekt kann der Drehmomenten-Istwert des Motors in Tausendstel des Nennmomentes (Objekt 6076_h) ausgelesen werden.

Index	6077_h
Name	torque_actual_value
Object Code	VAR
Data Type	INT16

Access	ro
PDO Mapping	yes
Units	motorRatedTorque/1000
Value Range	–
Default Value	–

Objekt 6078_h: current_actual_value

Über dieses Objekt kann der Strom-Istwert des Motors in Tausendstel des Nennstromes (Objekt 6075_h) ausgelesen werden.

Index	6078_h
Name	current_actual_value
Object Code	VAR
Data Type	INT16

Access	ro
PDO Mapping	yes
Units	motorRatedCurrent/1000
Value Range	–
Default Value	–

Objekt 6079_h: dc_link_circuit_voltage

Über dieses Objekt kann die Zwischenkreisspannung des Reglers ausgelesen werden. Die Spannung wird in der Einheit Millivolt angegeben.

Index	6079_h
Name	dc_link_circuit_voltage
Object Code	VAR
Data Type	UINT32

Access	ro
PDO Mapping	yes
Units	mV
Value Range	–
Default Value	–

Objekt 6087_h: torque_slope

Dieser Parameter beschreibt die Änderungsgeschwindigkeit der Sollwertrampe. Diese ist in Tausendstel vom Nennmoment pro Sekunde anzugeben. Beispielsweise wird der Drehmomenten-Sollwert target_torque von 0 Nm auf den Wert motorRated_torque erhöht. Wenn der Ausgangswert der zwischengeschalteten Drehmomentenrampe diesen Wert in einer Sekunde erreichen soll, dann ist in diesem Objekt der Wert 1000 einzuschreiben.

Index	6087_h
Name	torque_slope
Object Code	VAR
Data Type	UINT32

Access	rw
PDO Mapping	yes
Units	motorRated_torque/1000 s
Value Range	–
Default Value	0E310F94 _h

Objekt 6088_h: torque_profile_type

Mit dem Objekt torque_profile_type wird vorgegeben, mit welcher Kurvenform ein Sollwertsprung ausgeführt wird. Zur Zeit ist in diesem Regler nur die lineare Rampe implementiert, so dass dieses Objekt nur mit dem Wert 0 beschrieben werden kann.

Index	6088_h
Name	torque_profile_type
Object Code	VAR
Data Type	INT16

Access	rw
PDO Mapping	yes
Units	–
Value Range	0
Default Value	0

Wert	Bedeutung
0	Lineare Rampe

A Technischer Anhang

A.1 Technische Daten Interface EtherCAT

M3

Dieser Abschnitt gilt nur für die Motorcontroller CMMP-AS-...-**M3**.

A.1.1 Allgemein

Mechanisch		
Länge / Breite / Höhe	[mm]	112,6 x 87,2 x 28,3
Gewicht	[g]	55
Steckplatz		Steckplatz Ext2
Werkstoff-Hinweis		RoHS-konform

Tab. A.1 Technische Daten: Mechanisch

Elektrisch		
Signalpegel	[VDC]	0 ... 2,5
Differenzspannung	[VDC]	1,9 ... 2,1

Tab. A.2 Technische Daten: Elektrisch

A.1.2 Betriebs- und Umweltbedingungen

Transport		
Temperaturbereich	[°C]	0 ... +50
Luftfeuchtigkeit, bei max. 40 °C Umgebungstemperatur, nicht betauend	[%]	0 ... 90

Tab. A.3 Technische Daten: Transport

Lagerung		
Lagertemperatur	[°C]	-25 ... +75
Luftfeuchtigkeit, bei max. 40 °C Umgebungstemperatur, nicht betauend	[%]	0 ... 90
Zulässige Höhe (über NN)	[m]	< 1000

Tab. A.4 Technische Daten: Lagerung

B Diagnosemeldungen

Wenn ein Fehler auftritt, zeigt der Motorcontroller CMMP-AS-...-M3/-M0 eine Diagnosemeldung zyklisch in der 7-Segment-Anzeige an. Eine Fehlermeldung setzt sich aus einem E (für Error), einem Hauptindex und ein Subindex zusammen, z. B.: - **E 0 1 0** -.

Warnungen haben die gleiche Nummer wie eine Fehlermeldung. Im Unterschied dazu erscheint aber eine Warnung durch einen vorangestellten und nachgestellten Mittelbalken, z. B.: - **1 7 0** -.

B.1 Erläuterungen zu den Diagnosemeldungen

Die Bedeutung und ihre Maßnahmen der Diagnosemeldungen sind in der folgenden Tabelle zusammengefasst:

Begriffe	Bedeutung
Nr.	Hauptindex (Fehlergruppe) und Subindex der Diagnosemeldung. Anzeige im Display, in FCT bzw. im Diagnosespeicher über FHPP.
Code	Die Spalte Code enthält den Errorcode (Hex) über CiA 301.
Meldung	Meldung die im FCT angezeigt wird.
Ursache	Mögliche Ursachen für die Meldung.
Maßnahme	Maßnahme durch den Anwender.
Reaktion	Die Spalte Reaktion enthält die Fehlerreaktion (Defaulteinstellung, teilweise konfigurierbar): <ul style="list-style-type: none"> – PS off (Endstufe abschalten), – MCStop (Schnellhalt mit maximalem Strom), – QStop (Schnellhalt mit parametrierter Rampe), – Warn (Warnung), – Ignore (Keine Meldung, nur Eintrag in Diagnosespeicher), – NoLog (Keine Meldung und kein Eintrag in Diagnosespeicher).

Tab. B.1 Erläuterungen den Diagnosemeldungen

Unter Abschnitt B.2 finden Sie die Errorcodes nach CiA301/402 mit Zuordnung zu den Fehlernummern der Diagnosemeldungen.

Eine vollständige Liste der Diagnosemeldungen entsprechend der Firmwarestände zum Zeitpunkt der Drucklegung dieses Dokuments finden Sie unter Abschnitt B.3.

B.2 Errorcodes über CiA 301/402

Diagnosemeldungen			
Code	Nr.	Meldung	Reaktion
2311h	31-1	I²t-Servoregler	konfigurierbar
2312h	31-0	I²t-Motor	konfigurierbar
2313h	31-2	I²t-PFC	konfigurierbar
2314h	31-3	I²t-Bremswiderstand	konfigurierbar
2320h	06-0	Kurzschluss Endstufe	PS off
	06-1	Überstrom Brems-Chopper	PS off
3210h	07-0	Überspannung im Zwischenkreis	PS off
3220h	02-0	Unterspannung Zwischenkreis	konfigurierbar
3280h	32-0	Ladezeit Zwischenkreis überschritten	konfigurierbar
3281h	32-1	Unterspannung für aktive PFC	konfigurierbar
3282h	32-5	Überlast Brems-Chopper. Zwischenkreis konnte nicht entladen werden.	konfigurierbar
3283h	32-6	Entladezeit Zwischenkreis überschritten	konfigurierbar
3284h	32-7	Leistungsversorgung fehlt für Reglerfreigabe	konfigurierbar
3285h	32-8	Ausfall Leistungsversorgung bei Reglerfreigabe	QStop
3286h	32-9	Phasenausfall	QStop
4210h	04-0	Übertemperatur Leistungsteil	konfigurierbar
4280h	04-1	Übertemperatur Zwischenkreis	konfigurierbar
4310h	03-0	Übertemperatur Motor analog	QStop
	03-1	Übertemperatur Motor digital	konfigurierbar
	03-2	Übertemperatur Motor analog: Drahtbruch	konfigurierbar
	03-3	Übertemperatur Motor analog: Kurzschluss	konfigurierbar
5080h	90-0	Fehlende Hardwarekomponente (SRAM)	PS off
	90-2	Fehler beim Booten FPGA	PS off
	90-3	Fehler bei Start SD-ADUs	PS off
	90-4	Synchronisationsfehler SD-ADU nach Start	PS off
	90-5	SD-ADU nicht synchron	PS off
	90-6	IRQ0 (Stromregler): Trigger-Fehler	PS off
	90-9	DEBUG-Firmware geladen	PS off
5114h	05-0	Ausfall interne Spannung 1	PS off
5115h	05-1	Ausfall interne Spannung 2	PS off
5116h	05-2	Ausfall Treiberversorgung	PS off
5280h	21-0	Fehler 1 Strommessung U	PS off
5281h	21-1	Fehler 1 Strommessung V	PS off
5282h	21-2	Fehler 2 Strommessung U	PS off
5283h	21-3	Fehler 2 Strommessung V	PS off
5410h	05-3	Unterspannung dig. I/O	PS off
	05-4	Überstrom dig. I/O	PS off
5580h	26-0	Fehlender User-Parametersatz	PS off

Diagnosemeldungen			
Code	Nr.	Meldung	Reaktion
5581h	26-1	Checksummenfehler	PS off
5582h	26-2	Flash: Fehler beim Schreiben	PS off
5583h	26-3	Flash: Fehler beim Löschen	PS off
5584h	26-4	Flash: Fehler im internen Flash	PS off
5585h	26-5	Fehlende Kalibrierdaten	PS off
5586h	26-6	Fehlende User-Positionsdatensätze	PS off
6000h	91-0	Interner Initialisierungsfehler	PS off
6080h	25-0	Ungültiger Gerätetyp	PS off
6081h	25-1	Gerätetyp nicht unterstützt	PS off
6082h	25-2	HW-Revision nicht unterstützt	PS off
6083h	25-3	Gerätfunktion beschränkt!	PS off
6180h	01-0	Stack overflow	PS off
6181h	16-0	Programmausführung fehlerhaft	PS off
6182h	16-1	Illegaler Interrupt	PS off
6183h	16-3	Unerwarteter Zustand	PS off
6185h	15-0	Division durch 0	PS off
6186h	15-1	Bereichsüberschreitung	PS off
6187h	16-2	Initialisierungsfehler	PS off
6320h	36-0	Parameter wurde limitiert	konfigurierbar
	36-1	Parameter wurde nicht akzeptiert	konfigurierbar
6380h	30-0	Interner Umrechnungsfehler	PS off
7380h	08-0	Winkelgeberfehler Resolver	konfigurierbar
7382h	08-2	Fehler Spursignale Z0 Inkrementalgeber	konfigurierbar
7383h	08-3	Fehler Spursignale Z1 Inkrementalgeber	konfigurierbar
7384h	08-4	Fehler Spursignale digitaler Inkrementalgeber [X2B]	konfigurierbar
7385h	08-5	Fehler Hallgebersignale Inkrementalgeber	konfigurierbar
7386h	08-6	Kommunikationsfehler Winkelgeber	konfigurierbar
7387h	08-7	Signalamplitude Inkrementalspuren fehlerhaft [X10]	konfigurierbar
7388h	08-8	Interner Winkelgeberfehler	konfigurierbar
7389h	08-9	Winkelgeber an [X2B] wird nicht unterstützt	konfigurierbar
73A1h	09-0	Alter Winkelgeber-Parametersatz	konfigurierbar
73A2h	09-1	Winkelgeber-Parametersatz kann nicht dekodiert werden	konfigurierbar
73A3h	09-2	Unbekannte Version Winkelgeber-Parametersatz	konfigurierbar
73A4h	09-3	Defekte Datenstruktur Winkelgeber-Parametersatz	konfigurierbar
73A5h	09-7	Schreibgeschütztes EEPROM Winkelgeber	konfigurierbar
73A6h	09-9	EEPROM Winkelgeber zu klein	konfigurierbar
8081h	43-0	Endschalter: Negativer Sollwert gesperrt	konfigurierbar
8082h	43-1	Endschalter: Positiver Sollwert gesperrt	konfigurierbar
8083h	43-2	Endschalter: Positionierung unterdrückt	konfigurierbar
8120h	12-1	CAN: Kommunikationsfehler, Bus AUS	konfigurierbar
8180h	12-0	CAN: Knotennummer doppelt	konfigurierbar

Diagnosemeldungen			
Code	Nr.	Meldung	Reaktion
8181h	12-2	CAN: Kommunikationsfehler beim Senden	konfigurierbar
8182h	12-3	CAN: Kommunikationsfehler beim Empfangen	konfigurierbar
8480h	35-0	Durchdrehschutz Linearmotor	konfigurierbar
8611h	17-0	Schleppfehlerüberwachung	konfigurierbar
	17-1	Geberdifferenzüberwachung	konfigurierbar
	27-0	Warnschwelle Schleppfehler	konfigurierbar
8612h	40-0	Negativer SW-Endschalter erreicht	konfigurierbar
	40-1	Positiver SW-Endschalter erreicht	konfigurierbar
	40-2	Zielposition hinter negativem SW-Endschalter	konfigurierbar
	40-3	Zielposition hinter positivem SW-Endschalter	konfigurierbar
8680h	42-0	Positionierung: Fehlende Anschlusspositionierung: Stopp	konfigurierbar
8681h	42-1	Positionierung: Drehrichtungsumkehr nicht erlaubt: Stopp	konfigurierbar
8682h	42-2	Positionierung: Drehrichtungsumkehr nach Halt nicht erlaubt	konfigurierbar
8780h	34-0	Keine Synchronisation über Feldbus	konfigurierbar
8781h	34-1	Synchronisationsfehler Feldbus	konfigurierbar
8A80h	11-0	Fehler beim Starten der Referenzfahrt	konfigurierbar
8A81h	11-1	Fehler während der Referenzfahrt	konfigurierbar
8A82h	11-2	Referenzfahrt: kein gültiger Nullimpuls	konfigurierbar
8A83h	11-3	Referenzfahrt: Zeitüberschreitung	konfigurierbar
8A84h	11-4	Referenzfahrt: falscher / ungültiger Endschalter	konfigurierbar
8A85h	11-5	Referenzfahrt: I²t / Schleppfehler	konfigurierbar
8A86h	11-6	Referenzfahrt: Ende der Suchstrecke	konfigurierbar
8A87h	33-0	Schleppfehler Encoderemulation	konfigurierbar
F080h	80-0	Überlauf Stromregler IRQ	PS off
F081h	80-1	Überlauf Drehzahlregler IRQ	PS off
F082h	80-2	Überlauf Lageregler IRQ	PS off
F083h	80-3	Überlauf Interpolator IRQ	PS off
F084h	81-4	Überlauf Low-Level IRQ	PS off
F085h	81-5	Überlauf MDC IRQ	PS off
FF01h	28-0	Betriebsstundenzähler fehlt	konfigurierbar
FF02h	28-1	Betriebsstundenzähler: Schreibfehler	konfigurierbar

B.3 Diagnosemeldungen mit Hinweisen zur Störungsbeseitigung

Fehlergruppe 00		Ungültige Meldung oder Information	
Nr.	Code	Meldung	Reaktion
00-0	-	Ungültiger Fehler	
		Ursache	Information: Ein ungültiger Fehlereintrag (korruptiert) wurde im Diagnosespeicher mit dieser Fehlernummer markiert. Der Eintrag der Systemzeit wird auf 0 gesetzt.
		Maßnahme	–
00-1	-	Ungültiger Fehler entdeckt und korrigiert	
		Ursache	Information: Ein ungültiger Fehlereintrag (korruptiert) wurde im Diagnosespeicher entdeckt und korrigiert. In der Zusatz-Information steht die ursprüngliche Fehlernummer. Der Eintrag der Systemzeit enthält die Adresse der korruptierten Fehlernummer.
		Maßnahme	–
00-2	-	Fehler gelöscht	
		Ursache	Information: Aktive Fehler wurden quitiert.
		Maßnahme	–

Fehlergruppe 01		Stack overflow	
Nr.	Code	Meldung	Reaktion
01-0	6180h	Stack overflow	
		Ursache	– Falsche Firmware? – Sporadische hohe Rechenlast durch zu kleine Zykluszeit und spezielle rechenintensive Prozesse (Parametersatz speichern etc.).
		Maßnahme	• Eine freigegebene Firmware laden. • Rechenlast vermindern. • Kontakt zum Technischen Support aufnehmen.

Fehlergruppe 02		Zwischenkreis	
Nr.	Code	Meldung	Reaktion
02-0	3220h	Unterspannung Zwischenkreis	
		konfigurierbar	
		Ursache	Zwischenkreisspannung sinkt unter die parametrisierte Schwelle (→ Zusatzinformation). Fehlerpriorität zu hoch eingestellt?
		Maßnahme	<ul style="list-style-type: none">• Schnellentladung aufgrund abgeschalteter Netzversorgung.• Leistungsversorgung prüfen.• Zwischenkreise koppeln, sofern technisch zulässig.• Zwischenkreisspannung prüfen (messen).• Unterspannungsüberwachung (Schwellwert) prüfen.
		Zusatzinfo	Zusatzinfo in PNU 203/213: Obere 16 Bit: Zustandsnummer interne Statemachine Untere 16 Bit: Zwischenkreisspannung (interne Skalierung ca. 17,1 digit/V).

Fehlergruppe 03		Übertemperatur Motor	
Nr.	Code	Meldung	Reaktion
03-0	4310h	Übertemperatur Motor analog	
		Ursache	Motor überlastet, Temperatur zu hoch. <ul style="list-style-type: none">– Motor zu heiß?– Falscher Sensor?– Sensor defekt?– Kabelbruch?
		Maßnahme	<ul style="list-style-type: none">• Parametrierung prüfen (Stromregler, Stromgrenzwerte).• Parametrierung des Sensors oder der Sensorkennlinie prüfen. Falls Fehler auch bei überbrücktem Sensor vorhanden: Gerät defekt.
03-1	4310h	Übertemperatur Motor digital	
		Ursache	<ul style="list-style-type: none">– Motor überlastet, Temperatur zu hoch.– Passender Sensor oder Sensorkennlinie parametrisiert?– Sensor defekt?
		Maßnahme	<ul style="list-style-type: none">• Parametrierung prüfen (Stromregler, Stromgrenzwerte).• Parametrierung des Sensors oder der Sensorkennlinie prüfen. Falls Fehler auch bei überbrücktem Sensor vorhanden: Gerät defekt.
03-2	4310h	Übertemperatur Motor analog: Drahtbruch	
		Ursache	Gemessener Widerstandswert liegt oberhalb der Schwelle für die Drahtbruchererkennung.
		Maßnahme	<ul style="list-style-type: none">• Anschlussleitungen Temperatursensor auf Drahtbruch prüfen.• Parametrierung (Schwellwert) der Drahtbruchererkennung prüfen.

Fehlergruppe 03		Übertemperatur Motor	
Nr.	Code	Meldung	Reaktion
03-3	4310h	Übertemperatur Motor analog: Kurzschluss	
		konfigurierbar	
		Ursache	Gemessener Widerstandswert liegt unterhalb der Schwelle für die Kurzschlusserkennung.
		Maßnahme	<ul style="list-style-type: none"> • Anschlussleitungen Temperatursensor auf Drahtbruch prüfen. • Parametrierung (Schwellwert) der Kurzschlusserkennung prüfen.

Fehlergruppe 04		Übertemperatur Leistungsteil/Zwischenkreis	
Nr.	Code	Meldung	Reaktion
04-0	4210h	Übertemperatur Leistungsteil	
		konfigurierbar	
		Ursache	Gerät ist überhitzt <ul style="list-style-type: none"> – Temperaturanzeige plausibel? – Gerätelüfter defekt? – Gerät überlastet?
		Maßnahme	<ul style="list-style-type: none"> • Einbaubedingungen prüfen, Filter der Schaltschrank-Lüfter verschmutzt? • Antriebsauslegung prüfen (wegen möglicher Überlastung im Dauerbetrieb).
04-1	4280h	Übertemperatur Zwischenkreis	
		konfigurierbar	
		Ursache	Gerät ist überhitzt <ul style="list-style-type: none"> – Temperaturanzeige plausibel? – Gerätelüfter defekt? – Gerät überlastet?
		Maßnahme	<ul style="list-style-type: none"> • Einbaubedingungen prüfen, Filter der Schaltschrank-Lüfter verschmutzt? • Antriebsauslegung prüfen (wegen möglicher Überlastung im Dauerbetrieb).

Fehlergruppe 05		Interne Spannungsversorgung	
Nr.	Code	Meldung	Reaktion
05-0	5114h	Ausfall interne Spannung 1	
		PS off	
		Ursache	Überwachung der internen Spannungsversorgung hat eine Unterspannung erkannt. Entweder ein interner Defekt oder eine Überlastung / Kurzschluss durch angeschlossene Peripherie.
		Maßnahme	<ul style="list-style-type: none"> • Digitale Ausgänge und Bremsausgang auf Kurzschluss bzw. spezifizierte Belastung prüfen. • Gerät von der gesamten Peripherie trennen und prüfen, ob der Fehler nach Reset immer noch vorliegt. Wenn ja, dann liegt ein interner Defekt vor → Reparatur durch den Hersteller.

Fehlergruppe 05		Interne Spannungsversorgung		
Nr.	Code	Meldung	Reaktion	
05-1	5115h	Ausfall interne Spannung 2		PS off
		Ursache	Überwachung der internen Spannungsversorgung hat eine Unterspannung erkannt. Entweder ein interner Defekt oder eine Überlastung / Kurzschluss durch angeschlossene Peripherie.	
		Maßnahme	<ul style="list-style-type: none">• Digitale Ausgänge und Bremsausgang auf Kurzschluss bzw. spezifizierte Belastung prüfen.• Gerät von der gesamten Peripherie trennen und prüfen, ob der Fehler nach Reset immer noch vorliegt. Wenn ja, dann liegt ein interner Defekt vor → Reparatur durch den Hersteller.	
05-2	5116h	Ausfall Treiberversorgung		PS off
		Ursache	Überwachung der internen Spannungsversorgung hat eine Unterspannung erkannt. Entweder ein interner Defekt oder eine Überlastung / Kurzschluss durch angeschlossene Peripherie.	
		Maßnahme	<ul style="list-style-type: none">• Digitale Ausgänge und Bremsausgang auf Kurzschluss bzw. spezifizierte Belastung prüfen.• Gerät von der gesamten Peripherie trennen und prüfen, ob der Fehler nach Reset immer noch vorliegt. Wenn ja, dann liegt ein interner Defekt vor → Reparatur durch den Hersteller.	
05-3	5410h	Unterspannung dig. I/O		PS off
		Ursache	Überlastung der I/Os? Peripherie defekt?	
		Maßnahme	<ul style="list-style-type: none">• Angeschlossene Peripherie auf Kurzschluss bzw. spezifizierte Belastung prüfen.• Anschluss der Bremse prüfen (falsch angeschlossen?).	
05-4	5410h	Überstrom dig. I/O		PS off
		Ursache	Überlastung der I/Os? Peripherie defekt?	
		Maßnahme	<ul style="list-style-type: none">• Angeschlossene Peripherie auf Kurzschluss bzw. spezifizierte Belastung prüfen.• Anschluss der Bremse prüfen (falsch angeschlossen?).	
05-5	-	Ausfall Spannung Interface Ext1/Ext2		PS off
		Ursache	Defekt auf dem eingesteckten Interface.	
		Maßnahme	• Austausch Interface → Reparatur durch den Hersteller.	
05-6	-	Ausfall Spannung [X10], [X11]		PS off
		Ursache	Überlastung durch angeschlossene Peripherie.	
		Maßnahme	<ul style="list-style-type: none">• Pin-Belegung der angeschlossenen Peripherie prüfen.• Kurzschluß?	
05-7	-	Ausfall interne Spannung Sicherheitsmodul		PS off
		Ursache	Defekt auf dem Sicherheitsmodul.	
		Maßnahme	• Interner Defekt → Reparatur durch den Hersteller.	

Fehlergruppe 05		Interne Spannungsversorgung	
Nr.	Code	Meldung	Reaktion
05-8	-	Ausfall interne Spannung 3	
		Ursache	Defekt im Motorcontroller.
		Maßnahme	• Interner Defekt → Reparatur durch den Hersteller.
05-9	-	Geberversorgung fehlerhaft	
		Ursache	Rückmessung der Geberspannung nicht in Ordnung.
		Maßnahme	• Interner Defekt → Reparatur durch den Hersteller.

Fehlergruppe 06		Überstrom	
Nr.	Code	Meldung	Reaktion
06-0	2320h	Kurzschluss Endstufe	
		PS off	
		Ursache	<ul style="list-style-type: none"> – Motor defekt, z. B. Windungskurzschluss durch Überhitzung des Motors oder Schluss motorintern gegen PE. – Kurzschluss im Kabel oder den Verbindungssteckern, d.h. Kurzschluss der Motorphasen gegeneinander oder gegen Schirm/PE. – Endstufe defekt (Kurzschluss). – Fehlparametrierung des Stromreglers.
		Maßnahme	Abhängig vom Zustand der Anlage → Zusatzinformation Fall a) bis f).
06-1	2320h	Zusatzinfo	<p>Maßnahmen:</p> <p>a) Fehler nur bei aktivem Brems-Chopper: Externen Bremswiderstand auf Kurzschluss oder zu kleinen Widerstandswert prüfen. Beschaltung des Brems-Chopper-Ausgang am Motorcontroller prüfen (Brücke etc.).</p> <p>b) Fehlermeldung unmittelbar bei Zuschalten der Leistungsversorgung: interner Kurzschluss in der Endstufe (Kurzschluss einer kompletten Halbbrücke). Der Motorcontroller kann nicht mehr an die Leistungsversorgung angeschlossen werden, es fallen die internen (und ggf. die externen) Sicherungen aus. Reparatur durch Hersteller erforderlich.</p> <p>c) Fehlermeldung Kurzschluss erst bei Erteilen der Endstufen- bzw. Reglerfreigabe.</p> <p>d) Lösen des Motorsteckers [X6] direkt am Motorcontroller. Tritt der Fehler immer noch auf, liegt ein Defekt im Motorcontroller vor. Reparatur durch Hersteller erforderlich.</p> <p>e) Tritt der Fehler nur bei angeschlossenem Motorkabel auf: Motor und Kabel auf Kurzschlüsse prüfen, z. B. mit einem Multimeter.</p> <p>f) Parametrierung des Stromreglers prüfen. Ein falsch parametrierter Stromregler kann durch Schwingen Ströme bis zur Kurzschluss-Grenze erzeugen, in der Regel durch hochfrequentens Pfeifen deutlich wahrnehmbar. Verifikation ggf. mit dem Trace im FCT (Wirkstrom-Istwert).</p>
		Überstrom Brems-Chopper	
		PS off	
		Ursache	Überstrom am Brems-Chopper-Ausgang.
		Maßnahme	<ul style="list-style-type: none"> • Externen Bremswiderstand auf Kurzschluss oder zu kleinen Widerstandswert prüfen. • Beschaltung des Brems-Chopper-Ausgangs am Motorcontroller prüfen (Brücken etc.).

Fehlergruppe 07		Überspannung im Zwischenkreis	
Nr.	Code	Meldung	Reaktion
07-0	3210h	Überspannung im Zwischenkreis	
		PS off	
		Ursache	<p>Bremswiderstand wird überlastet, zu hohe Bremsenergie, die nicht schnell genug abgebaut werden kann.</p> <ul style="list-style-type: none"> – Widerstand falsch dimensioniert? – Widerstand nicht richtig angeschlossen? – Auslegung (Applikation) prüfen.
		Maßnahme	<ul style="list-style-type: none"> • Auslegung des Bremswiderstands prüfen, Widerstandswert ggf. zu groß. • Anschluss zum Bremswiderstand prüfen (intern/extern).

Fehlergruppe 08		Winkelgeberfehler	
Nr.	Code	Meldung	Reaktion
08-0	7380h	Winkelgeberfehler Resolver	
		konfigurierbar	
		Ursache	Signalamplitude Resolver fehlerhaft.
		Maßnahme	Schrittweises Vorgehen → Zusatzinformation Fall a) bis c).
		Zusatzinfo	<p>a) Falls möglich Test mit einem anderen (fehlerfreien) Resolver (auch die Anschlussleitung tauschen). Tritt der Fehler immer noch auf, liegt ein Defekt im Motorcontroller vor. Reparatur durch Hersteller erforderlich.</p> <p>b) Tritt der Fehler nur mit einem speziellen Resolver und dessen Anschlussleitung auf: Resolver signale prüfen (Träger und SIN/COS-Signale), siehe Spezifikation. Wird die Signalspezifikation nicht eingehalten, ist der Resolver zu tauschen.</p> <p>c) Tritt der Fehler immer wieder sporadisch auf, ist die Schirmanbindung zu untersuchen oder zu prüfen ob der Resolver grundsätzlich ein zu kleines Übertragungsverhältnis hat (Normresolver: $A = 0,5$).</p>

Fehlergruppe 08		Winkelgeberfehler	
Nr.	Code	Meldung	Reaktion
08-1	-	Drehsinn inkrementelle Lageerfassung ungleich	
		konfigurierbar	
		Ursache	Nur Geber mit serieller Positionsübertragung kombiniert mit einer analogen SIN/COS-Signalspur: Drehsinn von geberinterner Positionsbestimmung und inkrementeller Auswertung des analogen Spursystems im Motorcontroller ist vertauscht → Zusatzinformation.
		Maßnahme	Tauschen der folgenden Signale an der Winkelgeberschnittstelle [X2B] (Änderung der Adern im Anschlussstecker erforderlich), ggf. Datenblatt des Winkelgebers beachten: – SIN- / COS-Spur tauschen. – Tauschen der SIN+ / SIN- bzw. COS+ / COS- Signale.
		Zusatzinfo	Der Geber zählt intern z. B. im Uhrzeigersinn positiv während die inkrementelle Auswertung bei gleicher mechanischer Drehung in negativer Richtung zählt. Bei der ersten Bewegung um über 30° mechanisch wird die Vertauschung der Drehrichtung erkannt und der Fehler ausgelöst.
08-2	7382h	Fehler Spursignale Z0 Inkrementalgeber	
		konfigurierbar	
		Ursache	Signalamplitude der Z0-Spur an [X2B] fehlerhaft. – Winkelgeber angeschlossen? – Winkelgeberkabel defekt? – Winkelgeber defekt?
		Maßnahme	Konfiguration Winkelgeberinterface prüfen: a) Z0-Auswertung aktiviert aber es sind keine Spursignale angeschlossen oder vorhanden → Zusatzinformation. b) Gebersignale gestört? c) Test mit anderem Geber. → Tab. B.2, Seite 269.
		Zusatzinfo	Z. B. bei EnDat 2.2 oder EnDat 2.1 ohne Analogspur. Heidenhain-Geber: Bestellbezeichnungen EnDat 22 und EnDat 21. Bei diesen Gebern sind keine Inkrementalsignale vorhanden, auch wenn die Leitungen angeschlossen sind.

Fehlergruppe 08		Winkelgeberfehler	
Nr.	Code	Meldung	Reaktion
08-3	7383h	Fehler Spursignale Z1 Inkrementalgeber	
		konfigurierbar	
		Ursache	Signalamplitude der Z1-Spur an X2B fehlerhaft. – Winkelgeber angeschlossen? – Winkelgeberkabel defekt? – Winkelgeber defekt?
		Maßnahme	Konfiguration Winkelgeberinterface prüfen: a) Z1-Auswertung aktiviert aber nicht angeschlossen. b) Gebersignale gestört? c) Test mit anderem Geber. ➔ Tab. B.2, Seite 269.
08-4	7384h	Fehler Spursignale digitaler Inkrementalgeber [X2B]	
		konfigurierbar	
		Ursache	A, B, oder N-Spursignale an [X2B] fehlerhaft. – Winkelgeber angeschlossen? – Winkelgeberkabel defekt? – Winkelgeber defekt?
		Maßnahme	Konfiguration Winkelgeberinterface prüfen. a) Gebersignale gestört? b) Test mit anderem Geber. ➔ Tab. B.2, Seite 269.
08-5	7385h	Fehler Hallgebersignale Inkrementalgeber	
		konfigurierbar	
		Ursache	Hallgeber-Signale eines dig. Ink. an [X2B] fehlerhaft. – Winkelgeber angeschlossen? – Winkelgeberkabel defekt? – Winkelgeber defekt?
		Maßnahme	Konfiguration Winkelgeberinterface prüfen. a) Gebersignale gestört? b) Test mit anderem Geber. ➔ Tab. B.2, Seite 269.

Fehlergruppe 08		Winkelgeberfehler	
Nr.	Code	Meldung	Reaktion
08-6	7386h	Kommunikationsfehler Winkelgeber	
		konfigurierbar	
		Ursache	Kommunikation zu seriellen Winkelgebern gestört (EnDat-Geber, HIPERFACE-Geber, BiSS-Geber). – Winkelgeber angeschlossen? – Winkelgeberkabel defekt? – Winkelgeber defekt?
		Maßnahme	Konfiguration Winkelgeberinterface prüfen, Vorgehen entsprechend a) bis c): a) Serieller Geber parametriert aber nicht angeschlossen? Falsches seriellcs Protokoll ausgewählt? b) Gebersignale gestört? c) Test mit anderem Geber. → Tab. B.2, Seite 269.
08-7	7387h	Signalamplitude Inkrementalspuren fehlerhaft [X10]	
		konfigurierbar	
		Ursache	A, B, oder N-Spursignale an [X10] fehlerhaft. – Winkelgeber angeschlossen? – Winkelgeberkabel defekt? – Winkelgeber defekt?
		Maßnahme	Konfiguration Winkelgeberinterface prüfen. a) Gebersignale gestört? b) Test mit anderem Geber. → Tab. B.2, Seite 269.
08-8	7388h	Interner Winkelgeberfehler	
		konfigurierbar	
		Ursache	Interne Überwachung des Winkelgebers [X2B] hat einen Fehler erkannt und über die serielle Kommunikation an den Regler weitergeleitet. – Nachlassende Beleuchtungsstärke bei optischen Gebern? – Drehzahlüberschreitung? – Winkelgeber defekt?
		Maßnahme	Tritt der Fehler nachhaltig auf, ist der Geber defekt. → Geber wechseln.

Fehlergruppe 08		Winkelgeberfehler	
Nr.	Code	Meldung	Reaktion
08-9	7389h	Winkelgeber an [X2B] wird nicht unterstützt	
		konfigurierbar	
		Ursache	<p>Winkelgebertyp an [X2B] gelesen, der nicht unterstützt wird oder in der gewünschten Betriebsart nicht verwendet werden kann.</p> <ul style="list-style-type: none"> – Falscher oder ungeeigneter Protokolltyp gewählt? – Firmware unterstützt die angeschlossene Gebervariante nicht?
		Maßnahme	<p>Je nach Zusatzinformation der Fehlermeldung → Zusatzinformation:</p> <ul style="list-style-type: none"> • Geeignete Firmware laden. • Konfiguration der Geberauswertung prüfen / korrigieren. • Geeigneten Gebertyp anschließen.
		Zusatzinfo	<p>Zusatzinfo (PNU 203/213):</p> <p>0001: HIPERFACE: Gebertyp wird von der FW nicht unterstützt → anderen Gebertyp verwenden oder ggf. neuere Firmware laden.</p> <p>0002: EnDat: Der Adressraum, in dem Geberparameter liegen müssten, gibt es bei dem angeschlossenen EnDat-Geber nicht → Gebertyp prüfen.</p> <p>0003: EnDat: Gebertyp wird von der FW nicht unterstützt → anderen Gebertyp verwenden oder ggf. neuere Firmware laden.</p> <p>0004: EnDat: Gebertypenschild kann aus dem angeschlossenen Geber nicht ausgelesen werden. → Geber wechseln oder ggf. neuere Firmware laden.</p> <p>0005: EnDat: EnDat 2.2-Interface parametrieren, angeschlossener Geber unterstützt aber nur EnDat2.1. → Gebertyp wechseln oder auf EnDat 2.1 umparametrieren.</p> <p>0006: EnDat: EnDat2.1-Interface mit analoger Spurauswertung parametrieren aber laut Typenschild unterstützt der angeschlossene Geber keine Spursignale. → Geber wechseln oder Z0-Spursignalauswertung abschalten.</p> <p>0007: Codelängenmesssystem mit EnDat2.1 angeschlossen aber als rein serieller Geber parametrieren. Aufgrund der langen Antwortzeiten dieses Systems ist eine rein serielle Auswertung nicht möglich. Geber muss mit analoger Spursignalauswertung betrieben werden → Analoge Z0-Spursignalauswertung zuschalten.</p>

Fehlergruppe 09		Fehler im Winkelgeber-Parametersatz	
Nr.	Code	Meldung	Reaktion
09-0	73A1h	Alter Winkelgeber-Parametersatz	
		Ursache	Warnung: Im EEPROM des angeschlossenen Gebers wurde ein Geberparametersatz in einem alten Format gefunden. Dieser wurde jetzt konvertiert und neu gespeichert.
		Maßnahme	Soweit keine Aktivität. Die Warnung sollte beim erneuten Einschalten der 24 V nicht mehr auftauchen.
09-1	73A2h	Winkelgeber-Parametersatz kann nicht dekodiert werden	
		Ursache	Daten im EEPROM des Winkelgebers konnten nicht vollständig gelesen werden, bzw. der Zugriff wurde teilweise abgewehrt.
		Maßnahme	Im EEPROM des Gebers sind Daten (Kommunikationsobjekte) hinterlegt, die von der geladenen Firmware nicht unterstützt werden. Die entsprechenden Daten werden dann verworfen. <ul style="list-style-type: none"> Durch Schreiben der Geberdaten in den Geber kann der Parametersatz an die aktuelle Firmware angepasst werden. Alternativ geeignete (neuere) Firmware laden.
09-2	73A3h	Unbekannte Version Winkelgeber-Parametersatz	
		Ursache	Im EEPROM gespeicherte Daten nicht kompatibel zur aktuellen Version. Es ist eine Datenstruktur gefunden worden, die die geladene Firmware nicht decodieren kann.
		Maßnahme	<ul style="list-style-type: none"> Geberparameter erneut speichern um den Parametersatz im Geber zu löschen und gegen einen lesbaren Satz zu tauschen (allerdings werden dann die Daten im Geber irreversibel gelöscht). Alternativ geeignete (neuere) Firmware laden.
09-3	73A4h	Defekte Datenstruktur Winkelgeber-Parametersatz	
		Ursache	Daten im EEPROM passen nicht zur hinterlegten Datenstruktur. Datenstruktur wurde als gültig erkannt, ist aber eventuell korrupt.
		Maßnahme	<ul style="list-style-type: none"> Geberparameter erneut speichern um den Parametersatz im Geber zu löschen und gegen einen lesbaren Satz zu tauschen. Tritt der Fehler danach immer noch auf, ist eventuell der Geber defekt. Testweise Geber tauschen.

Fehlergruppe 09		Fehler im Winkelgeber-Parametersatz	
Nr.	Code	Meldung	Reaktion
09-4	-	EEPROM-Daten: Kundenspezifische Konfiguration fehlerhaft	
		konfigurierbar	
		Ursache	Nur bei speziellen Motoren: Die Plausibilitätsprüfung liefert einen Fehler, z. B. weil der Motor repariert oder getauscht wurde.
Maßnahme	<ul style="list-style-type: none">• Wenn Motor repariert: Neu referenzieren und Speichern im Winkelgeber, danach (!) speichern im Motorcontroller.• Wenn Motor getauscht: Controller neu parametrieren, danach wieder neu referenzieren und Speichern im Winkelgeber, danach (!) speichern im Motorcontroller.		
09-7	73A5h	Schreibgeschütztes EEPROM Winkelgeber	
		konfigurierbar	
		Ursache	Kein Speichern von Daten im EEPROM des Winkelgebers möglich. Tritt bei Hiperface-Gebern auf.
Maßnahme	Ein Datenfeld des Geber EEPROMs ist schreibgeschützt (z. B. nach Betrieb an Motorcontroller eines anderen Herstellers). Keine Lösung möglich, Geberspeicher muss über entsprechendes Parametrieretool (Hersteller) entsperrt werden.		
09-9	73A6h	EEPROM Winkelgeber zu klein	
		konfigurierbar	
		Ursache	Es können nicht alle Daten im EEPROM des Winkelgebers gespeichert werden.
Maßnahme	<ul style="list-style-type: none">• Anzahl der Datensätze für das Speichern reduzieren. Bitte lesen Sie die Dokumentation oder nehmen Sie Kontakt zum Technischen Support auf.		

Fehlergruppe 10		Überdrehzahl	
Nr.	Code	Meldung	Reaktion
10-0	-	Überdrehzahl (Durchdreherschutz)	
		konfigurierbar	
		Ursache	<ul style="list-style-type: none">– Motor hat durchgedreht weil der Kommutierwinkeloffset falsch ist.– Motor ist korrekt parametrier, aber Grenzwert für Durchdreh-schutz ist zu klein eingestellt.
		Maßnahme	<ul style="list-style-type: none">• Kommutierwinkeloffset prüfen.• Parametrierung des Grenzwertes prüfen.

Fehlergruppe 11		Fehler Referenzfahrt	
Nr.	Code	Meldung	Reaktion
11-0	8A80h	Fehler beim Starten der Referenzfahrt	
		Ursache	Reglerfreigabe fehlt.
		Maßnahme	Ein Start der Referenzfahrt ist nur bei aktiver Reglerfreigabe möglich. <ul style="list-style-type: none">• Bedingung bzw. Ablauf prüfen.
11-1	8A81h	Fehler während der Referenzfahrt	
		Ursache	Referenzfahrt wurde unterbrochen, z. B. durch: <ul style="list-style-type: none">– Wegnahme der Reglerfreigabe.– Referenzschalter liegt hinter dem Endschalter.– Externes Stop-Signal (Abbruch einer Phase der Referenzfahrt).
		Maßnahme	<ul style="list-style-type: none">• Ablauf der Referenzfahrt prüfen.• Anordnung der Schalter prüfen.• Stop-Eingang während der Referenzfahrt ggf. verriegeln falls unerwünscht.
11-2	8A82h	Referenzfahrt: kein gültiger Nullimpuls	
		Ursache	Erforderlicher Nullimpuls bei der Referenzfahrt fehlt.
		Maßnahme	<ul style="list-style-type: none">• Nullimpulssignal überprüfen.• Winkelgebereinstellungen überprüfen.
11-3	8A83h	Referenzfahrt: Zeitüberschreitung	
		Ursache	Die maximal für die Referenzfahrt parametrisierte Zeit wurde erreicht, noch bevor die Referenzfahrt beendet wurde.
		Maßnahme	<ul style="list-style-type: none">• Parametrierung der Zeit prüfen.
11-4	8A84h	Referenzfahrt: falscher / ungültiger Endschalter	
		Ursache	<ul style="list-style-type: none">– Zugehöriger Endschalter nicht angeschlossen.– Endschalter vertauscht?– Kein Referenzschalter zwischen den beiden Endschaltern gefunden.– Referenzschalter liegt auf Endschalter.– Methode "Aktuelle Position mit Nullimpuls": Endschalter im Bereich des Nullimpulses aktiv (nicht zulässig).– Beide Endschalter gleichzeitig aktiv.
		Maßnahme	<ul style="list-style-type: none">• Prüfung, ob die Endschalter in der richtigen Fahrtrichtung angeschlossen sind oder ob die Endschalter auf die vorgesehenen Eingänge wirken.• Referenzschalter angeschlossen?• Anordnung Referenzschalter prüfen.• Endschalter verschieben, so dass er nicht im Bereich des Nullimpulses liegt.• Parametrierung Endschalter (Öffner/Schließer) prüfen.

Fehlergruppe 11		Fehler Referenzfahrt			
Nr.	Code	Meldung	Reaktion		
11-5	8A85h	Referenzfahrt: I²t / Schleppfehler		konfigurierbar	
		Ursache	<ul style="list-style-type: none">– Beschleunigungsrampen ungeeignet parametrier.– Richtungswechsel durch vorzeitig ausgelösten Schleppfehler, Parametrierung des Schleppfehlers prüfen.– Zwischen den Endanschlägen keinen Referenzschalter erreicht.– Methode Nullimpuls: Endanschlag erreicht (hier nicht zulässig).		
		Maßnahme	<ul style="list-style-type: none">• Beschleunigungsrampen flacher parametrieren.• Anschluss eines Referenzschalters prüfen.• Methode für Applikation geeignet?		
11-6	8A86h	Referenzfahrt: Ende der Suchstrecke		konfigurierbar	
		Ursache	Die für die Referenzfahrt maximal zulässige Strecke ist abgefahren, ohne dass der Bezugspunkt oder das Ziel der Referenzfahrt erreicht wurde.		
		Maßnahme	Störung bei der Erkennung des Schalters. <ul style="list-style-type: none">• Schalter für Referenzfahrt defekt?		
11-7	-	Referenzfahrt: Fehler Geberdifferenzüberwachung		konfigurierbar	
		Ursache	Abweichung zwischen Lageistwert und Kommutierlage zu groß. Externer Winkelgeber nicht angeschlossen bzw. defekt?		
		Maßnahme	<ul style="list-style-type: none">• Abweichung schwankt z.B. aufgrund von Getriebeispiel, ggf. Abschaltschwelle vergrößern.• Anschluss des Istwertgebers prüfen.		

Fehlergruppe 12		CAN-Fehler		
Nr.	Code	Meldung	Reaktion	
12-0	8180h	CAN: Knotennummer doppelt		konfigurierbar
		Ursache	Doppelt vergebene Knotennummer.	
		Maßnahme	• Konfiguration der Teilnehmer am CAN-Bus prüfen.	
12-1	8120h	CAN: Kommunikationsfehler, Bus AUS		konfigurierbar
		Ursache	Der CAN-Chip hat die Kommunikation aufgrund von Kommunikationsfehlern abgeschaltet (BUS OFF).	
		Maßnahme	• Verkabelung prüfen: Kabelspezifikation eingehalten, Kabelbruch, maximale Kabellänge überschritten, Abschlusswiderstände korrekt, Kabelschirm geerdet, alle Signale aufgelegt? • Gerät ggf. testweise tauschen. Wenn ein anderes Gerät bei gleicher Verkabelung fehlerfrei arbeitet, Gerät zur Prüfung zum Hersteller einschicken.	

Fehlergruppe 12		CAN-Fehler		
Nr.	Code	Meldung	Reaktion	
12-2	8181h	CAN: Kommunikationsfehler beim Senden		konfigurierbar
		Ursache	Beim Senden von Nachrichten sind die Signale gestört. Hochlauf des Gerätes so schnell, dass beim Senden der Boot-Up Nachricht noch kein weiterer Knoten am Bus erkannt wird.	
		Maßnahme	<ul style="list-style-type: none">Verkabelung prüfen: Kabelspezifikation eingehalten, Kabelbruch, maximale Kabellänge überschritten, Abschlusswiderstände korrekt, Kabelschirm geerdet, alle Signale aufgelegt?Gerät ggf. testweise tauschen. Wenn ein anderes Gerät bei gleicher Verkabelung fehlerfrei arbeitet, Gerät zur Prüfung zum Hersteller einschicken.	
12-3	8182h	CAN: Kommunikationsfehler beim Empfangen		konfigurierbar
		Ursache	Beim Empfangen von Nachrichten sind die Signale gestört.	
		Maßnahme	<ul style="list-style-type: none">Verkabelung prüfen: Kabelspezifikation eingehalten, Kabelbruch, maximale Kabellänge überschritten, Abschlusswiderstände korrekt, Kabelschirm geerdet, alle Signale aufgelegt?Gerät ggf. testweise tauschen. Wenn ein anderes Gerät bei gleicher Verkabelung fehlerfrei arbeitet, Gerät zur Prüfung zum Hersteller einschicken.	
12-4	-	CAN: Node Guarding		konfigurierbar
		Ursache	Kein Node Guarding Telegramm innerhalb der parametrierten Zeit empfangen. Signale gestört?	
		Maßnahme	<ul style="list-style-type: none">Zykluszeit der Remoteframes mit der Steuerung abgleichen.Prüfen: Ausfall der Steuerung?	
12-5	-	CAN: RPDO zu kurz		konfigurierbar
		Ursache	Ein empfangenes RPDO enthält nicht die parametrierte Anzahl von Bytes.	
		Maßnahme	Anzahl der parametrierten Bytes entspricht nicht der Anzahl der empfangenen Bytes. <ul style="list-style-type: none">Parametrierung prüfen und korrigieren.	
12-9	-	CAN: Protokollfehler		konfigurierbar
		Ursache	Fehlerhaftes Busprotokoll.	
		Maßnahme	<ul style="list-style-type: none">Parametrierung des ausgewählten CAN-Busprotokolls prüfen.	

Fehlergruppe 13		Timeout CAN-Bus	
Nr.	Code	Meldung	Reaktion
13-0	-	Timeout CAN-Bus	
		konfigurierbar	
		Ursache	Fehlermeldung aus herstellerspezifischem Protokoll.
		Maßnahme	<ul style="list-style-type: none">CAN-Parametrierung prüfen.

Fehlergruppe 14		Fehler Identifizierung		
Nr.	Code	Meldung	Reaktion	
14-0	-	Unzureichende Versorgung für Identifizierung		PS off
		Ursache	Stromregler-Parameter können nicht bestimmt werden (unzureichende Versorgung).	
		Maßnahme	Die zur Verfügung stehende Zwischenkreisspannung ist für die Durchführung der Messung zu gering.	
14-1	-	Identifizierung Stromregler: Messzyklus unzureichend		PS off
		Ursache	Für angeschlossenen Motor zu wenig oder zu viele Messzyklen erforderlich.	
		Maßnahme	Die automatische Parameterbestimmung liefert eine Zeitkonstante, die außerhalb des parametrierbaren Wertebereichs liegt. <ul style="list-style-type: none">Die Parameter müssen manuell optimiert werden.	
14-2	-	Endstufenfreigabe konnte nicht erteilt werden		PS off
		Ursache	Die Erteilung der Endstufenfreigabe ist nicht erfolgt.	
		Maßnahme	<ul style="list-style-type: none">Anschluss von DIN4 prüfen.	
14-3	-	Endstufe wurde vorzeitig abgeschaltet		PS off
		Ursache	Die Endstufenfreigabe wurde bei laufender Identifizierung abgeschaltet.	
		Maßnahme	<ul style="list-style-type: none">Ablaufsteuerung prüfen.	
14-5	-	Nullimpuls konnte nicht gefunden werden		PS off
		Ursache	Der Nullimpuls konnte nach Ausführung der maximal zulässigen Anzahl elektrischer Umdrehungen nicht gefunden werden.	
		Maßnahme	<ul style="list-style-type: none">Nullimpulssignal prüfen.Winkelgeber korrekt parametriert?	
14-6	-	Hall-Signale ungültig		PS off
		Ursache	Hall-Signale fehlerhaft oder ungültig. Die Impulsfolge bzw. Segmentierung der Hallsignale ist ungeeignet.	
		Maßnahme	<ul style="list-style-type: none">Anschluss prüfen.Anhand Datenblatt prüfen, ob der Geber 3 Hallsignale mit 1205 oder 605 Segmenten aufweist, ggf. Kontakt zum Technischen Support aufnehmen.	
14-7	-	Identifizierung nicht möglich		PS off
		Ursache	Winkelgeber steht still.	
		Maßnahme	<ul style="list-style-type: none">Ausreichende Zwischenkreisspannung sicherstellen.Geberkabel mit dem richtigen Motor verbunden?Motor blockiert, z. B. Haltebremse löst nicht?	

Fehlergruppe 14		Fehler Identifizierung	
Nr.	Code	Meldung	Reaktion
14-8	-	Ungültige Polpaarzahl	
		PS off	
		Ursache	Die berechnete Polpaarzahl liegt außerhalb des parametrierbaren Bereiches.
		Maßnahme	<ul style="list-style-type: none">• Resultat mit den Angaben aus dem Datenblatt des Motors vergleichen.• Parametrierte Strichzahl prüfen.

Fehlergruppe 15		Ungültige Operation		
Nr.	Code	Meldung	Reaktion	
15-0	6185h	Division durch 0		PS off
		Ursache	Interner Firmwarefehler. Division durch 0 bei Verwendung der Mathe-Library.	
		Maßnahme	<ul style="list-style-type: none">• Werkseinstellungen laden.• Firmware prüfen, ob eine freigegebene Firmware geladen ist.	
15-1	6186h	Bereichsüberschreitung		PS off
		Ursache	Interner Firmwarefehler. Overflow bei Verwendung der Mathe-Library.	
		Maßnahme	<ul style="list-style-type: none">• Werkseinstellungen laden.• Firmware prüfen, ob eine freigegebene Firmware geladen ist.	
15-2	-	Zahlenunterlauf		PS off
		Ursache	Interner Firmwarefehler. Interne Korrekturgrößen konnten nicht berechnet werden.	
		Maßnahme	<ul style="list-style-type: none">• Einstellung der Factor Group auf extreme Werte prüfen und ggf. ändern.	

Fehlergruppe 16		Interner Fehler	
Nr.	Code	Meldung	Reaktion
16-0	6181h	Programmausführung fehlerhaft	
		PS off	
		Ursache	Interner Firmwarefehler. Fehler bei der Programmausführung. Illegales CPU-Kommando im Programmablauf gefunden.
		Maßnahme	<ul style="list-style-type: none">• Im Wiederholungsfall Firmware erneut laden. Tritt der Fehler wiederholt auf, ist die Hardware defekt.
16-1	6182h	Illegaler Interrupt	
		PS off	
		Ursache	Fehler bei der Programmausführung. Es wurde ein nicht benutzter IRQ-Vektor von der CPU genutzt.
		Maßnahme	<ul style="list-style-type: none">• Im Wiederholungsfall Firmware erneut laden. Tritt der Fehler wiederholt auf, ist die Hardware defekt.

Fehlergruppe 16		Interner Fehler	
Nr.	Code	Meldung	Reaktion
16-2	6187h	Initialisierungsfehler	
		PS off	
		Ursache	Fehler beim Initialisieren der Default-Parameter.
		Maßnahme	<ul style="list-style-type: none">Im Wiederholungsfall Firmware erneut laden. Tritt der Fehler wiederholt auf, ist die Hardware defekt.
16-3	6183h	Unerwarteter Zustand	
		PS off	
		Ursache	Fehler bei CPU-internen Peripheriezugriffen oder Fehler im Programmablauf (illegale Verzweigung in Case-Strukturen).
		Maßnahme	<ul style="list-style-type: none">Im Wiederholungsfall Firmware erneut laden. Tritt der Fehler wiederholt auf, ist die Hardware defekt.

Fehlergruppe 17		Überschreitung Schleppfehler		
Nr.	Code	Meldung	Reaktion	
17-0	8611h	Schleppfehlerüberwachung		konfigurierbar
		Ursache	Vergleichsschwelle zum Grenzwert des Schleppfehlers überschritten.	
		Maßnahme	<ul style="list-style-type: none">• Fehlerfenster vergrößern.• Beschleunigung kleiner parametrieren.• Motor überlastet (Strombegrenzung aus der I²t Überwachung aktiv?).	
17-1	8611h	Geberdifferenzüberwachung		konfigurierbar
		Ursache	Abweichung zwischen Lageistwert und Kommutierlage zu groß. Externer Winkelgeber nicht angeschlossen bzw. defekt?	
		Maßnahme	<ul style="list-style-type: none">• Abweichung schwankt z. B. aufgrund von Getriebeispiel, ggf. Abschaltschwelle vergrößern.• Anschluss des Istwertgebers prüfen.	

Fehlergruppe 18		Warnschwellen Temperatur	
Nr.	Code	Meldung	Reaktion
18-0	-	Analoge Motortemperatur	
		konfigurierbar	
		Ursache	Temperatur Motor (analog) größer als 5° unter T_max.
		Maßnahme	<ul style="list-style-type: none">• Stromregler- bzw. Drehzahlreglerparametrierung prüfen.• Motor dauerhaft überlastet?

Fehlergruppe 21		Fehler Strommessung	
Nr.	Code	Meldung	Reaktion
21-0	5280h	Fehler 1 Strommessung U	
		PS off	
		Ursache	Offset Strommessung 1 Phase U zu groß. Der Regler führt bei jeder Reglerfreigabe einen Offsetabgleich der Strommessung durch. Zu große Toleranzen führen zu einem Fehler.
		Maßnahme	Tritt der Fehler wiederholt auf, ist die Hardware defekt.
21-1	5281h	Fehler 1 Strommessung V	
		PS off	
		Ursache	Offset Strommessung 1 Phase V zu groß.
		Maßnahme	Tritt der Fehler wiederholt auf, ist die Hardware defekt.
21-2	5282h	Fehler 2 Strommessung U	
		PS off	
		Ursache	Offset Strommessung 2 Phase U zu groß.
		Maßnahme	Tritt der Fehler wiederholt auf, ist die Hardware defekt.
21-3	5283h	Fehler 2 Strommessung V	
		PS off	
		Ursache	Offset Strommessung 2 Phase V zu groß.
		Maßnahme	Tritt der Fehler wiederholt auf, ist die Hardware defekt.

Fehlergruppe 22		Fehler PROFIBUS (nur CMMP-AS-...-M3)		
Nr.	Code	Meldung	Reaktion	
22-0	-	PROFIBUS: Fehlerhafte Initialisierung		konfigurierbar
		Ursache	Fehlerhafte Initialisierung des PROFIBUS Interface. Interface defekt?	
		Maßnahme	• Interface tauschen. Ggf. Reparatur durch den Hersteller möglich.	
22-2	-	Kommunikationsfehler PROFIBUS		konfigurierbar
		Ursache	Störungen bei der Kommunikation.	
		Maßnahme	• Eingestellte Slave-Adresse prüfen. • Busabschluss prüfen. • Verkabelung prüfen.	
22-3	-	PROFIBUS: ungültige Slave-Adresse		konfigurierbar
		Ursache	Kommunikation wurde mit der Slave-Adresse 126 gestartet.	
		Maßnahme	• Auswahl einer anderen Slave-Adresse.	
22-4	-	PROFIBUS: Fehler im Wertebereich		konfigurierbar
		Ursache	Bei Umrechnung mit Factor Group wurde der Wertebereich überschritten. Mathematischer Fehler in der Umrechnung der physikalischen Einheiten.	
		Maßnahme	Wertebereich der Daten und der physikalischen Einheiten passen nicht zueinander. • Prüfen und korrigieren.	

Fehlergruppe 25		Fehler Gerätetyp/-funktion	
Nr.	Code	Meldung	Reaktion
25-0	6080h	Ungültiger Gerätetyp	
		Ursache	Gerätecodierung nicht erkannt oder ungültig.
		Maßnahme	Fehler kann nicht selbst behoben werden. <ul style="list-style-type: none">Motorcontroller zum Hersteller einschicken.
25-1	6081h	Gerätetyp nicht unterstützt	
		Ursache	Gerätecodierung ungültig, wird von geladener Firmware nicht unterstützt.
		Maßnahme	<ul style="list-style-type: none">Aktuelle Firmware laden.Falls keine neuere Firmware verfügbar ist kann es sich um einen Hardware-Defekt handeln. Motorcontroller zum Hersteller einschicken.
25-2	6082h	HW-Revision nicht unterstützt	
		Ursache	Die Hardware-Revision des Controllers wird von der geladenen Firmware nicht unterstützt.
		Maßnahme	<ul style="list-style-type: none">Firmware-Version prüfen, ggf. Firmware-Update auf eine neuere Firmware-Version durchführen.
25-3	6083h	Gerätfunktion beschränkt!	
		Ursache	Gerät ist für diese Funktion nicht freigeschaltet.
		Maßnahme	Gerät ist für die gewünschte Funktionalität nicht freigeschaltet und muss ggf. vom Hersteller freigeschaltet werden. Dazu muss Gerät eingeschickt werden.
25-4	-	Ungültiger Leistungsteiltyp	
		Ursache	<ul style="list-style-type: none">Leistungsteilbereich im EEPROM ist unprogrammiert.Leistungsteil wird von der Firmware nicht unterstützt.
		Maßnahme	<ul style="list-style-type: none">Geeignete Firmware laden.

Fehlergruppe 26		Interner Datenfehler		
Nr.	Code	Meldung	Reaktion	
26-0	5580h	Fehlender User-Parametersatz		PS off
		Ursache	Kein gültiger User-Parametersatz im Flash.	
		Maßnahme	<ul style="list-style-type: none">Werkseinstellungen laden. Steht der Fehler weiter an, ist eventuell die Hardware defekt.	
26-1	5581h	Checksummenfehler		PS off
		Ursache	Checksummenfehler eines Parametersatzes.	
		Maßnahme	<ul style="list-style-type: none">Werkseinstellungen laden. Steht der Fehler weiter an, ist eventuell die Hardware defekt.	
26-2	5582h	Flash: Fehler beim Schreiben		PS off
		Ursache	Fehler beim Schreiben des internen Flash.	
		Maßnahme	<ul style="list-style-type: none">Letzte Operation erneut ausführen. Tritt der Fehler wiederholt auf, ist eventuell die Hardware defekt.	

Fehlergruppe 26		Interner Datenfehler	
Nr.	Code	Meldung	Reaktion
26-3	5583h	Flash: Fehler beim Löschen	
		PS off	
		Ursache	Fehler beim Löschen des internen Flash.
		Maßnahme	<ul style="list-style-type: none">• Letzte Operation erneut ausführen. Tritt der Fehler wiederholt auf, ist eventuell die Hardware defekt.
26-4	5584h	Flash: Fehler im internen Flash	
		PS off	
		Ursache	Default-Parametersatz ist korruptiert / Datenfehler im FLASH-Bereich in dem der Default-Parametersatz liegt.
		Maßnahme	<ul style="list-style-type: none">• Firmware erneut laden. Tritt der Fehler wiederholt auf, ist eventuell die Hardware defekt.
26-5	5585h	Fehlende Kalibrierdaten	
		PS off	
		Ursache	Werkseitige Kalibrierparameter unvollständig / korruptiert.
		Maßnahme	Fehler kann nicht selbst behoben werden.
26-6	5586h	Fehlende User-Positionsdatensätze	
		PS off	
		Ursache	Positionsdatensätze unvollständig oder korruptiert.
		Maßnahme	<ul style="list-style-type: none">• Werkseinstellungen laden oder• aktuelle Parameter erneut sichern, damit die Positionsdaten erneut geschrieben werden.
26-7	-	Fehler in den Datentabellen (CAM)	
		PS off	
		Ursache	Daten für die Kurvenscheibe korruptiert.
		Maßnahme	<ul style="list-style-type: none">• Werkseinstellungen laden.• Parametersatz ggf. erneut laden. Steht der Fehler weiter an, Kontakt zum Technischen Support aufnehmen.

Fehlergruppe 27		Warnschwelle Schleppfehler	
Nr.	Code	Meldung	Reaktion
27-0	8611h	Warnschwelle Schleppfehler	
		konfigurierbar	
		Ursache	<ul style="list-style-type: none">– Motor überlastet? Dimensionierung prüfen.– Beschleunigungs oder Bremsrampen sind zu steil eingestellt.– Motor blockiert? Kommutierwinkel korrekt?
		Maßnahme	<ul style="list-style-type: none">• Parametrierung der Motordaten prüfen.• Parametrierung des Schleppfehlers prüfen.

Fehlergruppe 28		Fehler Betriebsstundenzähler	
Nr.	Code	Meldung	Reaktion
28-0	FF01h	Betriebsstundenzähler fehlt	
		Ursache	Im Parameterblock konnte kein Datensatz für einen Betriebsstundenzähler gefunden werden. Es wurde ein neuer Betriebsstundenzähler angelegt. Tritt bei Erstinbetriebnahme oder einem Prozessorwechsel auf.
		Maßnahme	Nur Warnung, keine weiteren Maßnahmen erforderlich.
28-1	FF02h	Betriebsstundenzähler: Schreibfehler	
		Ursache	Der Datenblock in dem sich der Betriebsstundenzähler befindet konnte nicht geschrieben werden. Ursache unbekannt, eventuell Probleme mit der Hardware.
		Maßnahme	Nur Warnung, keine weiteren Maßnahmen erforderlich. Bei wiederholtem Auftreten ist eventuell die Hardware defekt.
28-2	FF03h	Betriebsstundenzähler korrigiert	
		Ursache	Der Betriebsstundenzähler besitzt eine Sicherheitskopie. Wird die 24V-Versorgung des Reglers genau in dem Moment abgeschaltet wenn der Betriebstundenzähler aktualisiert wird, wird der beschriebene Datensatz eventuell korumpiert. In diesem Fall restauriert der Regler beim Wiedereinschalten den Betriebsstundenzähler aus der intakten Sicherheitskopie.
		Maßnahme	Nur Warnung, keine weiteren Maßnahmen erforderlich.
28-3	FF04h	Betriebsstundenzähler konvertiert	
		Ursache	Es wurde eine Firmware geladen, bei der der Betriebstundenzähler ein anderes Datenformat hat. Beim erstmaligen Einschalten wird der alte Datensatz des Betriebsstundenzählers in das neue Format konvertiert.
		Maßnahme	Nur Warnung, keine weiteren Maßnahmen erforderlich.

Fehlergruppe 29		MMC/SD-Karte	
Nr.	Code	Meldung	Reaktion
29-0	-	MMC/SD-Karte nicht vorhanden	
		konfigurierbar	
		Ursache	Dieser Fehler wird in folgenden Fällen ausgelöst: <ul style="list-style-type: none">– wenn eine Aktion auf der Speicherkarte durchgeführt werden soll (DCO-Datei laden bzw. erstellen, FW-Download), aber keine Speicherkarte eingesteckt ist.– Der DIP-Schalter S3 auf ON steht aber nach dem Reset/ Neustart keine Karte gesteckt ist.
		Maßnahme	Geeignete Speicherkarte in den Slot stecken. Nur wenn ausdrücklich erwünscht!

Fehlergruppe 29		MMC/SD-Karte	
Nr.	Code	Meldung	Reaktion
29-1	-	MMC/SD-Karte: Initialisierungsfehler	
		konfigurierbar	
		Ursache	<p>Dieser Fehler wird in folgenden Fällen ausgelöst:</p> <ul style="list-style-type: none"> – Die Speicherkarte konnte nicht initialisiert werden. Ggf. nicht unterstützter Kartentyp! – Nicht unterstütztes Dateisystem. – Fehler im Zusammenhang mit dem Shared Memory.
		Maßnahme	<ul style="list-style-type: none"> • Verwendeten Kartentyp prüfen. • Speicherkarte an einen PC anschließen und neu formatieren.
29-2	-	MMC/SD-Karte: Fehler Parametersatz	
		konfigurierbar	
		Ursache	<p>Dieser Fehler wird in folgenden Fällen ausgelöst:</p> <ul style="list-style-type: none"> – Ein Lade- bzw. Speichervorgang läuft bereits, aber ein neuer Lade- bzw. Speichervorgang wird angefordert. DCO-Datei » Servo – Die zu ladende DCO-Datei wurde nicht gefunden. – Die zu ladende DCO-Datei ist nicht für das Gerät geeignet. – Die zu ladende DCO-Datei ist fehlerhaft. – Servo » DCO-Datei – Die Speicherkarte ist schreibgeschützt. – Sonstiger Fehler beim Speichern des Parametersatzes als DCO-Datei. – Fehler bei der Erstellung der Datei „INFO.TXT“.
		Maßnahme	<ul style="list-style-type: none"> • Lade- bzw. Speichervorgang nach einer Wartezeit von 5 Sekunden neu ausführen. • Speicherkarte an einen PC anschließen und die enthaltenen Dateien prüfen. • Schreibschutz von der Speicherkarte entfernen.
29-3	-	MMC/SD-Karte voll	
		konfigurierbar	
		Ursache	<ul style="list-style-type: none"> – Dieser Fehler wird ausgelöst, falls beim Speichern der DCO-Datei oder der Datei INFO.TXT festgestellt wird, dass die Speicherkarte schon voll ist. – Der maximale Datei-Index (99) existiert bereits. D.h., alle Datei-Indizes sind belegt. Es kann kein Dateiname vergeben werden!
		Maßnahme	<ul style="list-style-type: none"> • Andere Speicherkarte einsetzen. • Dateinamen ändern.

Fehlergruppe 29		MMC/SD-Karte	
Nr.	Code	Meldung	Reaktion
29-4	-	MMC/SD-Karte: Firmware-Download	
		konfigurierbar	
		Ursache	<p>Dieser Fehler wird in folgenden Fällen ausgelöst:</p> <ul style="list-style-type: none"> – keine FW-Datei auf der Speicherkarte. – Die FW-Datei ist nicht für das Gerät geeignet. – Sonstiger Fehler beim FW-Download, z. B. Checksummenfehler bei einem SRecord, Fehler beim Flashen, etc.
		Maßnahme	<ul style="list-style-type: none"> • Speicherkarte an PC anschließen und Firmwaredatei übertragen.

Fehlergruppe 30		Interner Umrechnungsfehler	
Nr.	Code	Meldung	Reaktion
30-0	6380h	Interner Umrechnungsfehler	
		PS off	
		Ursache	<p>Bereichsüberschreitung bei internen Skalierungsfaktoren aufgetreten, die von den parametrisierten Reglerzykluszeiten abhängen.</p>
		Maßnahme	<ul style="list-style-type: none"> • Prüfen ob extrem kleine oder extrem große Zykluszeiten parametrisiert wurden.

Fehlergruppe 31		I ² t-Fehler	
Nr.	Code	Meldung	Reaktion
31-0	2312h	I²t-Motor	
		konfigurierbar	
		Ursache	<p>I²t-Überwachung des Motors hat angesprochen.</p> <ul style="list-style-type: none"> – Motor/Mechanik blockiert oder schwergängig. – Motor unterdimensioniert?
		Maßnahme	<ul style="list-style-type: none"> • Leistungsdimensionierung Antriebspaket prüfen.
31-1	2311h	I²t-Servoregler	
		konfigurierbar	
		Ursache	<p>Die I²t-Überwachung spricht häufig an.</p> <ul style="list-style-type: none"> – Motorcontroller unterdimensioniert? – Mechanik schwergängig?
		Maßnahme	<ul style="list-style-type: none"> • Projektierung des Motorcontrollers prüfen, • ggf. Leistungsstärkeren Typ einsetzen. • Mechanik prüfen.
31-2	2313h	I²t-PFC	
		konfigurierbar	
		Ursache	<p>Leistungsbemessung der PFC überschritten.</p>
		Maßnahme	<ul style="list-style-type: none"> • Betrieb ohne PFC parametrieren (FCT).
31-3	2314h	I²t-Bremswiderstand	
		konfigurierbar	
		Ursache	<p>– Überlastung des internen Bremswiderstandes.</p>
		Maßnahme	<ul style="list-style-type: none"> • Externen Bremswiderstand verwenden. • Widerstandswert reduzieren oder Widerstand mit höherer Impulsbelastung einsetzen.

Fehlergruppe 32		Fehler Zwischenkreis	
Nr.	Code	Meldung	Reaktion
32-0	3280h	Ladezeit Zwischenkreis überschritten	
		konfigurierbar	
		Ursache	Nach Anlegen der Netzspannung konnte der Zwischenkreis nicht geladen werden. – Eventuell Sicherung defekt oder – interner Bremswiderstand defekt oder – im Betrieb mit externem Widerstand dieser nicht angeschlossen.
		Maßnahme	<ul style="list-style-type: none">• Anschaltung des externen Bremswiderstandes prüfen.• Alternativ prüfen ob die Brücke für den internen Bremswiderstand gesetzt ist. Ist die Anschaltung korrekt ist vermutlich der interne Bremswiderstand oder die eingebaute Sicherung defekt. Eine Reparatur vor Ort ist nicht möglich.
32-1	3281h	Unterspannung für aktive PFC	
		konfigurierbar	
		Ursache	Die PFC kann erst ab einer Zwischenkreisspannung von ca. 130 V DC überhaupt aktiviert werden.
		Maßnahme	<ul style="list-style-type: none">• Leistungsversorgung prüfen.
32-5	3282h	Überlast Brems-Chopper. Zwischenkreis konnte nicht entladen werden.	
		konfigurierbar	
		Ursache	Die Auslastung des Brems-Choppers bei Beginn der Schnellentladung lag bereits im Bereich oberhalb 100%. Die Schnellentladung hat den Brems-Chopper an die maximale Belastungsgrenze gebracht und wurde verhindert/abgebrochen.
		Maßnahme	Keine Maßnahme erforderlich.
32-6	3283h	Entladezeit Zwischenkreis überschritten	
		konfigurierbar	
		Ursache	Zwischenkreis konnte nicht schnellentladen werden. Eventuell ist der interne Bremswiderstand defekt oder im Betrieb mit externem Widerstand ist dieser nicht angeschlossen.
		Maßnahme	<ul style="list-style-type: none">• Anschaltung des externen Bremswiderstandes prüfen.• Alternativ prüfen ob die Brücke für den internen Bremswiderstand gesetzt ist. Ist der interne Widerstand gewählt und die Brücke korrekt gesetzt, ist vermutlich der interne Bremswiderstand defekt.

Fehlergruppe 32		Fehler Zwischenkreis		
Nr.	Code	Meldung	Reaktion	
32-7	3284h	Leistungsversorgung fehlt für Reglerfreigabe		konfigurierbar
		Ursache	Reglerfreigabe wurde erteilt, als der Zwischenkreis sich nach angelegter Netzspannung noch in der Aufladephase befand und das Netzrelais noch nicht angezogen war. Der Antrieb kann in dieser Phase nicht freigegeben werden, da der Antrieb noch nicht hart an das Netz angeschaltet ist (Netzrelais).	
		Maßnahme	• In der Applikation prüfen ob Netzversorgung und Reglerfreigabe entsprechend kurz hintereinander erteilt werden.	
32-8	3285h	Ausfall Leistungsversorgung bei Reglerfreigabe		QStop
		Ursache	Unterbrechungen / Netzausfall der Leistungsversorgung während die Reglerfreigabe aktiviert war.	
		Maßnahme	• Leistungsversorgung prüfen.	
32-9	3286h	Phasenausfall		QStop
		Ursache	Ausfall einer oder mehrerer Phasen (nur bei dreiphasiger Speisung).	
		Maßnahme	• Leistungsversorgung prüfen.	

Fehlergruppe 33		Schleppfehler Encoderemulation	
Nr.	Code	Meldung	Reaktion
33-0	8A87h	Schleppfehler Encoderemulation	
		konfigurierbar	
		Ursache	Die Grenzfrequenz der Encoderemulation wurde überschritten (siehe Handbuch) und der emulierte Winkel an [X11] konnte nicht mehr folgen. Kann auftreten, wenn sehr hohe Strichzahlen für [X11] programmiert sind und der Antrieb hohe Drehzahlen erreicht.
		Maßnahme	<ul style="list-style-type: none">• Prüfen ob die parametrisierte Strichzahl eventuell zu hoch für die abzubildende Drehzahl ist.• Gegebenenfalls Strichzahl reduzieren.

Fehlergruppe 34		Fehler Synchronisation Feldbus	
Nr.	Code	Meldung	Reaktion
34-0	8780h	Keine Synchronisation über Feldbus	
		konfigurierbar	
		Ursache	Bei aktivieren des Interpolated-Position-Mode konnte der Regler nicht auf den Feldbus aufsynchronisiert werden. – Eventuell sind die Synchronisationsnachrichten vom Master ausgefallen oder – das IPO-Intervall ist nicht korrekt auf das Synchronisationsintervall des Feldbusses eingestellt.
		Maßnahme	• Einstellungen der Reglerzykluszeiten prüfen.

Fehlergruppe 34		Fehler Synchronisation Feldbus	
Nr.	Code	Meldung	Reaktion
34-1	8781h	Synchronisationsfehler Feldbus	konfigurierbar
		Ursache	<ul style="list-style-type: none"> – Die Synchronisation über Feldbusnachrichten im laufenden Betrieb (Interpolated-Position-Mode) ist ausgefallen. – Synchronisationsnachrichten vom Master ausgefallen? – Synchronisationsintervall (IPO-Intervall) zu klein/zu groß parametrisiert?
		Maßnahme	<ul style="list-style-type: none"> • Einstellungen der Reglerzykluszeiten prüfen.

Fehlergruppe 35		Linearmotor	
Nr.	Code	Meldung	Reaktion
35-0	8480h	Durchdreherschutz Linearmotor	konfigurierbar
		Ursache	Gebersignale sind gestört. Der Motor dreht eventuell durch weil die Kommutierlage sich durch die gestörten Gebersignale verstellt hat.
		Maßnahme	<ul style="list-style-type: none"> • Installation auf EMV-Empfehlungen prüfen. • Bei Linearmotoren mit induktiven/optischen Gebern mit getrennt montiertem Massband und Messkopf den mechanischen Abstand kontrollieren. • Bei Linearmotoren mit induktiven Gebern sicherstellen, dass das Magnetfeld der Magneten oder der Motorwicklung nicht in den Messkopf streut (dieser Effekt tritt dann meist bei hohen Beschleunigungen = hohem Motorstrom auf).

Fehlergruppe 35		Linearmotor	
Nr.	Code	Meldung	Reaktion
35-5	-	Fehler bei der Kommutierlagebestimmung	
		konfigurierbar	
		Ursache	<p>Rotorlage konnte nicht eindeutig identifiziert werden.</p> <ul style="list-style-type: none">– Das gewählte Verfahren ist möglicherweise ungeeignet.– Eventuell der gewählte Motorstrom für die Identifizierung nicht passend eingestellt.
		Maßnahme	<ul style="list-style-type: none">• Methode der Kommutierlagebestimmung prüfen → Zusatzinformation.
Zusatzinfo	<p>Hinweise zur Kommutierlagebestimmung:</p> <p>a) Das Ausrichteverfahren ist ungeeignet für festgebremste oder schwergängige Antriebe oder Antriebe die niederfrequent schwingfähig sind.</p> <p>b) Das Mikroschrittverfahren ist für eisenlose und eisenbehaftete Motoren geeignet. Da nur sehr kleine Bewegungen durchgeführt werden arbeitet es auch wenn der Antrieb auf elastischen Anschlägen steht oder festgebremst aber noch etwas elastisch bewegbar ist. Aufgrund der hohen Anregungsfrequenz ist das Verfahren jedoch bei schlecht gedämpften Antrieben sehr anfällig für Schwingungen. In diesem Fall kann versucht werden, den Anregungsstrom (%) zu reduzieren.</p> <p>c) Das Sättigungsverfahren nutzt lokale Sättigungserscheinungen im Eisen des Motors. Empfohlen für festgebremste Antriebe. Eisenlose Antrieb sind prinzipiell für diese Methode ungeeignet. Bewegt sich der (eisenbehaftete) Antrieb bei der Kommutierlagefindung zu stark, kann das Messergebnis verfälscht sein. In diesem Fall den Anregungsstrom reduzieren. Im umgekehrten Fall bewegt sich der Antrieb nicht, der Anregungsstrom ist aber eventuell nicht stark genug und damit die Sättigung nicht ausgeprägt genug.</p>		

Fehlergruppe 36		Parameterfehler	
Nr.	Code	Meldung	Reaktion
36-0	6320h	Parameter wurde limitiert	
		konfigurierbar	
		Ursache	Es wurde versucht ein Wert zu schreiben, der außerhalb der zulässigen Grenzen liegt und deshalb limitiert wurde.
		Maßnahme	• Benutzerparametersatz kontrollieren.
36-1	6320h	Parameter wurde nicht akzeptiert	
		konfigurierbar	
		Ursache	Es wurde versucht ein Objekt zu schreiben, welches nur lesbar ist oder im aktuellen Zustand (z. B. bei aktiver Reglerfreigabe) nicht beschreibbar ist.
		Maßnahme	• Benutzerparametersatz kontrollieren.

Fehlergruppe 40		Software-Endschalter		
Nr.	Code	Meldung	Reaktion	
40-0	8612h	Negativer SW-Endschalter erreicht		konfigurierbar
		Ursache	Der Lagesollwert hat den negativen Software-Endschalter erreicht bzw. überschritten.	
		Maßnahme	<ul style="list-style-type: none">• Zieldaten prüfen.• Positionierbereich prüfen.	
40-1	8612h	Positiver SW-Endschalter erreicht		konfigurierbar
		Ursache	Der Lagesollwert hat den positiven Software-Endschalter erreicht bzw. überschritten.	
		Maßnahme	<ul style="list-style-type: none">• Zieldaten prüfen.• Positionierbereich prüfen.	
40-2	8612h	Zielposition hinter negativem SW-Endschalter		konfigurierbar
		Ursache	Der Start einer Positionierung wurde unterdrückt, da das Ziel hinter dem negativen Software-Endschalter liegt.	
		Maßnahme	<ul style="list-style-type: none">• Zieldaten prüfen.• Positionierbereich prüfen.	
40-3	8612h	Zielposition hinter positivem SW-Endschalter		konfigurierbar
		Ursache	Der Start einer Positionierung wurde unterdrückt, da das Ziel hinter dem positiven Software-Endschalter liegt.	
		Maßnahme	<ul style="list-style-type: none">• Zieldaten prüfen.• Positionierbereich prüfen.	

Fehlergruppe 41		Satzweiserschaltung: Synchronisationsfehler	
Nr.	Code	Meldung	Reaktion
41-0	-	Satzweiserschaltung: Synchronisationsfehler	
		konfigurierbar	
		Ursache	Start eines Aufsynchronisierens ohne vorigem Sampling-Puls.
		Maßnahme	• Parametrierung der Vorhalt-Strecke prüfen.

Fehlergruppe 42		Fehler Positionierung	
Nr.	Code	Meldung	Reaktion
42-0	8680h	Positionierung: Fehlende Anschlusspositionierung: Stopp	
		konfigurierbar	
		Ursache	Das Ziel der Positionierung kann durch die Optionen der Positionierung bzw. der Randbedingungen nicht erreicht werden.
		Maßnahme	• Parametrierung der betreffenden Positionssätze prüfen.
42-1	8681h	Positionierung: Drehrichtungsumkehr nicht erlaubt: Stopp	
		konfigurierbar	
		Ursache	Das Ziel der Positionierung kann durch die Optionen der Positionierung bzw. der Randbedingungen nicht erreicht werden.
		Maßnahme	• Parametrierung der betreffenden Positionssätze prüfen.

Fehlergruppe 42		Fehler Positionierung	
Nr.	Code	Meldung	Reaktion
42-2	8682h	Positionierung: Drehrichtungsumkehr nach Halt nicht erlaubt	
		Ursache	Das Ziel der Positionierung kann durch die Optionen der Positionierung bzw. der Randbedingungen nicht erreicht werden.
		Maßnahme	• Parametrierung der betreffenden Positionssätze prüfen.
42-3	-	Start Positionierung verworfen: falsche Betriebsart	
		Ursache	Eine Umschaltung der Betriebsart durch den Positionssatz war nicht möglich.
		Maßnahme	• Parametrierung der betreffenden Positionssätze prüfen.
42-4	-	Start Positionierung verworfen: Referenzfahrt erforderlich	
		Ursache	Es wurde ein normaler Positionssatz gestartet, obwohl der Antrieb vor dem Start eine gültige Referenzposition benötigt.
		Maßnahme	• Neue Referenzfahrt durchführen.
42-5	-	Modulo Positionierung: Drehrichtung nicht erlaubt	
		Ursache	<ul style="list-style-type: none">– Das Ziel der Positionierung kann durch die Optionen der Positionierung bzw. der Randbedingungen nicht erreicht werden.– Die berechnete Drehrichtung ist gemäß dem eingestellten Modus für die Modulo Positionierung nicht erlaubt.
		Maßnahme	• Gewählten Modus prüfen.
42-9	-	Fehler beim Starten der Positionierung	
		Ursache	<ul style="list-style-type: none">– Beschleunigungsgrenzwert überschritten.– Positionssatz gesperrt.
		Maßnahme	• Parametrierung und Ablaufsteuerung prüfen, ggf. korrigieren.

Fehlergruppe 43		Fehler Hardware-Endschalter	
Nr.	Code	Meldung	Reaktion
43-0	8081h	Endschalter: Negativer Sollwert gesperrt	
		Ursache	Negativer Hardware-Endschalter erreicht.
		Maßnahme	• Parametrierung, Verdrahtung und Endschalter prüfen.
43-1	8082h	Endschalter: Positiver Sollwert gesperrt	
		Ursache	Positiver Hardware-Endschalter erreicht.
		Maßnahme	• Parametrierung, Verdrahtung und Endschalter prüfen.
43-2	8083h	Endschalter: Positionierung unterdrückt	
		Ursache	– Der Antrieb hat den vorgesehenen Bewegungsraum verlassen. – Technischer Defekt in der Anlage?
		Maßnahme	• Vorgesehenen Bewegungsraum prüfen.

Fehlergruppe 44		Fehler Kurvenscheibe	
Nr.	Code	Meldung	Reaktion
44-0	-	Fehler in den Kurvenscheibentabellen	
		Ursache	Zu startende Kurvenscheibe nicht vorhanden.
		Maßnahme	<ul style="list-style-type: none"> • Übergebene Kurvenscheiben-Nr. prüfen. • Parametrierung korrigieren. • Programmierung korrigieren.
44-1	-	Kurvenscheibe: allgemeiner Fehler Referenzierung	
		Ursache	– Start einer Kurvenscheibe, aber der Antrieb noch nicht referenziert ist.
		Maßnahme	• Referenzfahrt ausführen.
		Ursache	– Start einer Referenzfahrt bei aktiver Kurvenscheibe.
		Maßnahme	• Kurvenscheibe deaktivieren. Dann ggf. Kurvenscheibe neu starten.

Fehlergruppe 47		Timeout Einrichtbetrieb	
Nr.	Code	Meldung	Reaktion
47-0	-	Fehler Einrichtbetrieb: Timeout abgelaufen	
		Ursache	Die für den Einrichtbetrieb erforderliche Drehzahl wurde nicht rechtzeitig unterschritten.
		Maßnahme	Verarbeitung der Anforderung auf Steuerungsseite prüfen.

Fehlergruppe 48		Referenzfahrt erforderlich	
Nr.	Code	Meldung	Reaktion
48-0	-	Referenzfahrt erforderlich	
		Ursache	Es wird versucht, in der Betriebsart Drehzahl- bzw. Momentenregelung umzuschalten bzw. in einer dieser Betriebsarten die Reglerfreigabe zu erteilen, obwohl der Antrieb hierfür eine gültige Referenzposition benötigt.
		Maßnahme	• Referenzfahrt ausführen.

Fehlergruppe 50		Fehler CAN	
Nr.	Code	Meldung	Reaktion
50-0	-	Zu viele synchrone PDOs	
		konfigurierbar	
		Ursache	Es sind mehr PDOs aktiviert, als im zugrunde liegenden SYNC-Intervall abgearbeitet werden können. Diese Meldung tritt auch auf, wenn nur ein PDO synchron übertragen werden soll, aber eine hohe Anzahl weiterer PDOs mit anderem transmission type aktiviert sind.
		Maßnahme	<ul style="list-style-type: none"> • Aktivierung der PDOs prüfen. Falls eine geeignete Konfiguration vorliegt, kann die Warnung über das Fehlermanagement unterdrückt werden. • Synchronisationsintervall verlängern.
50-1	-	SDO-Fehler aufgetreten	
		konfigurierbar	
		Ursache	Ein SDO-Transfer hat einen SDO-Abort verursacht. <ul style="list-style-type: none"> – Daten überschreiten den Wertebereich. – Zugriff auf ein nicht existierendes Objekt.
		Maßnahme	<ul style="list-style-type: none"> • Gesendetes Kommando prüfen.

Fehlergruppe 51		Fehler Sicherheitsmodul (nur CMMP-AS-....-M3)	
Nr.	Code	Meldung	Reaktion
51-0	-	Kein / unbekanntes Sicherheitsmodul (Fehler ist nicht quittierbar)	
		PS off	
		Ursache	– Kein Sicherheitsmodul erkannt bzw. unbekannter Modultyp.
		Maßnahme	<ul style="list-style-type: none"> • Für die Firmware und Hardware geeignetes Sicherheits- oder Schaltermodul einbauen. • Eine für das Sicherheits- oder Schaltermodul geeignete Firmware laden, vgl. Typenbezeichnung auf dem Modul.
		Ursache	– Interner Spannungsfehler des Sicherheitsmoduls oder Schaltermoduls.
		Maßnahme	<ul style="list-style-type: none"> • Modul vermutlich defekt. Falls möglich mit einem anderen Modul tauschen.
51-2	-	Sicherheitsmodul: Ungleiches Modultyp (Fehler ist nicht quittierbar)	
		PS off	
		Ursache	Typ oder Revision des Moduls passt nicht zur Projektierung.
		Maßnahme	<ul style="list-style-type: none"> • Beim Modultauch: Modultyp noch nicht projiziert. Aktuell eingebautes Sicherheits- oder Schaltermodul als akzeptiert übernehmen.

Fehlergruppe 51		Fehler Sicherheitsmodul (nur CMMP-AS-...-M3)	
Nr.	Code	Meldung	Reaktion
51-3	-	Sicherheitsmodul: Ungleiche Modulversion (Fehler ist nicht quittierbar)	
		Ursache	Typ oder Revision des Moduls wird nicht unterstützt.
		Maßnahme	<ul style="list-style-type: none"> Für die Firmware und Hardware geeignetes Sicherheits- oder Schaltermodul einbauen. Eine für das Modul geeignete Firmware laden, vergleiche Typenbezeichnung auf dem Modul.

Fehlergruppe 51		Fehler Sicherheitsfunktion (nur CMMP-AS-...-M0)	
Nr.	Code	Meldung	Reaktion
51-0	-	Sicherheitsfunktion: Treiberfunktion fehlerhaft (Fehler ist nicht quittierbar)	
		Ursache	Interner Spannungsfehler der STO-Schaltung.
		Maßnahme	<ul style="list-style-type: none"> Sicherheitsschaltung defekt. Keine Maßnahmen möglich, bitte kontaktieren Sie Festo. Falls möglich durch einen anderen Motorcontroller tauschen.

Fehlergruppe 52		Fehler Sicherheitsmodul (nur CMMP-AS-...-M3)	
Nr.	Code	Meldung	Reaktion
52-1	-	Sicherheitsmodul: Diskrepanzzeit abgelaufen	
		Ursache	– Steuereingänge STO-A und STO-B werden nicht gleichzeitig betätigt.
		Maßnahme	<ul style="list-style-type: none"> Diskrepanzzeit prüfen.
		Ursache	– Steuereingänge STO-A und STO-B sind nicht gleichsinnig beschaltet.
		Maßnahme	<ul style="list-style-type: none"> Diskrepanzzeit prüfen.
52-2	-	Sicherheitsmodul: Ausfall Treiberversorgung bei aktiver PWM-Ansteuerung	
		Ursache	Diese Fehlermeldung tritt bei ab Werk gelieferten Geräten nicht auf. Sie kann auftreten bei Verwendung einer kundenspezifischen Gerätefirmware.
		Maßnahme	<ul style="list-style-type: none"> Der sichere Zustand wurde bei freigegebener Leistungsstufe angefordert. Einbindung in die sicherheitsgerichtete Anschaltung prüfen.

Fehlergruppe 52		Fehler Sicherheitsfunktion (nur CMMP-AS-...-M0)		
Nr.	Code	Meldung	Reaktion	
52-1	-	Sicherheitsfunktion: Diskrepanzzeit abgelaufen		PS off
		Ursache	– Steuereingänge STO-A und STO-B werden nicht gleichzeitig betätigt.	
		Maßnahme	• Diskrepanzzeit prüfen.	
		Ursache	– Steuereingänge STO-A und STO-B sind nicht gleichsinnig beschaltet.	
		Maßnahme	• Diskrepanzzeit prüfen.	
52-2	-	Sicherheitsfunktion: Ausfall Treiberversorgung bei aktiver PWM-Ansteuerung		PS off
		Ursache	Diese Fehlermeldung tritt bei ab Werk gelieferten Geräten nicht auf. Sie kann auftreten bei Verwendung einer kundenspezifischen Gerätefirmware.	
		Maßnahme	• Der sichere Zustand wurde bei freigegebener Leistungsendstufe angefordert. Einbindung in die sicherheitsgerichtete Anschaltung prüfen.	

Fehlergruppe 62		Fehler EtherCAT (nur CMMP-AS-...-M3)	
Nr.	Code	Meldung	Reaktion
62-0	-	EtherCAT: Allgemeiner Busfehler	
		Ursache	Kein EtherCAT Bus vorhanden.
		Maßnahme	<ul style="list-style-type: none">• Den EtherCAT Master einschalten.• Verkabelung prüfen.
62-1	-	EtherCAT: Initialisierungsfehler	
		Ursache	Fehler in der Hardware.
		Maßnahme	<ul style="list-style-type: none">• Interface austauschen und zur Prüfung an den Hersteller einschicken.
62-2	-	EtherCAT: Protokollfehler	
		Ursache	Es wird kein CAN over EtherCAT verwendet.
		Maßnahme	<ul style="list-style-type: none">• Falsches Protokoll.• EtherCAT Bus Verkabelung gestört.
62-3	-	EtherCAT: Ungültige RPDO-Länge	
		Ursache	Sync Manager 2 Puffer Größe zu groß.
		Maßnahme	<ul style="list-style-type: none">• Prüfen Sie die RPDO Konfiguration des Motorcontrollers und der Steuerung.
62-4	-	EtherCAT: Ungültige TPDO-Länge	
		Ursache	Sync Manager 3 Puffer Größe zu groß.
		Maßnahme	<ul style="list-style-type: none">• Prüfen Sie die TPDO Konfiguration des Motorcontrollers und der Steuerung.

Fehlergruppe 62		Fehler EtherCAT (nur CMMP-AS-...-M3)	
Nr.	Code	Meldung	Reaktion
62-5	-	EtherCAT: Zyklische Datenübertragung fehlerhaft	
		konfigurierbar	
		Ursache	Sicherheitsabschaltung durch Ausfall der zyklischen Datenübertragung.
		Maßnahme	<ul style="list-style-type: none">• Prüfen Sie die Konfiguration des Masters. Die synchrone Übertragung ist nicht stabil.

Fehlergruppe 63		Fehler EtherCAT (nur CMMP-AS-...-M3)		
Nr.	Code	Meldung	Reaktion	
63-0	-	EtherCAT: Interface defekt		konfigurierbar
		Ursache	Fehler in der Hardware.	
		Maßnahme	• Interface austauschen und zur Prüfung an den Hersteller einschicken.	
63-1	-	EtherCAT: Ungültige Daten		konfigurierbar
		Ursache	Fehlerhafter Telegrammtyp.	
		Maßnahme	• Verkabelung prüfen.	
63-2	-	EtherCAT: TPDO-Daten wurden nicht gelesen		konfigurierbar
		Ursache	Puffer zum Versenden der Daten voll.	
		Maßnahme	Die Daten werden schneller gesendet als der Motorcontroller sie verarbeiten kann. • Reduzieren Sie die Zykluszeit auf dem EtherCAT Bus.	
63-3	-	EtherCAT: Keine Distributed Clocks aktiv		konfigurierbar
		Ursache	Warnung: Firmware synchronisiert auf das Telegramm nicht auf das Distributed clocks System. Beim Starten des EtherCAT wurde kein Hardware SYNC (Distributed Clocks) gefunden. Die Firmware synchronisiert sich nun auf den EtherCAT Frame.	
		Maßnahme	• Ggf. Prüfen ob der Master das Merkmal Distributed Clocks unterstützt. • Andernfalls: Sicherstellen, dass die EtherCAT Frames nicht durch andere Frames gestört werden, falls der Interpolated Position Mode verwendet werden soll.	
63-4	-	EtherCAT: Fehlen einer SYNC-Nachricht im IPO-Zyklus		konfigurierbar
		Ursache	Es wird nicht im Zeitraster des IPO Telegramme verschickt.	
		Maßnahme	• Zuständigen Teilnehmer für Distributed Clocks prüfen.	

Fehlergruppe 64		Fehler DeviceNet (nur CMM-AS-...-M3)	
Nr.	Code	Meldung	Reaktion
64-0	-	DeviceNet: MAC ID doppelt	
		Ursache	Der Duplicate MAC-ID Check hat zwei Knoten mit der gleichen MAC-ID gefunden.
		Maßnahme	<ul style="list-style-type: none"> • Ändern sie die MAC-ID eines Knotens auf einen nicht verwendeten Wert.
64-1	-	DeviceNet: Busspannung fehlt	
		Ursache	Das DeviceNet-Interface wird nicht mit 24 V DC versorgt.
		Maßnahme	<ul style="list-style-type: none"> • Zusätzlich zum Motorcontroller auch das DeviceNet-Interface an 24 V DC anschließen.
64-2	-	DeviceNet: Empfangspuffer übergelaufen	
		Ursache	Zu viele Nachrichten innerhalb kurzer Zeit erhalten.
		Maßnahme	<ul style="list-style-type: none"> • Reduzieren Sie die Scanrate.
64-3	-	DeviceNet: Sendepuffer übergelaufen	
		Ursache	Nicht genügend freier Platz auf dem CAN-Bus, um Nachrichten zu senden.
		Maßnahme	<ul style="list-style-type: none"> • Erhöhen Sie die Baudrate. • reduzieren Sie die Anzahl von Knoten. • reduzieren Sie die Scanrate.
64-4	-	DeviceNet: IO-Nachricht nicht gesendet	
		Ursache	Fehler beim Senden von I/O-Daten.
		Maßnahme	<ul style="list-style-type: none"> • Prüfen Sie, ob das Netzwerk ordnungsgemäß verbunden und nicht gestört ist.
64-5	-	DeviceNet: Bus Off	
		Ursache	Der CAN-Regler ist BUS OFF.
		Maßnahme	<ul style="list-style-type: none"> • Prüfen Sie, ob das Netzwerk ordnungsgemäß verbunden und nicht gestört ist.
64-6	-	DeviceNet: CAN-Controller meldet Überlauf	
		Ursache	Der CAN-Regler hat einen Überlauf.
		Maßnahme	<ul style="list-style-type: none"> • Erhöhen Sie die Baudrate. • reduzieren sie die Anzahl der Knoten. • reduzieren Sie die Scanrate.

Fehlergruppe 65		Fehler DeviceNet (nur CMM-AS-...-M3)	
Nr.	Code	Meldung	Reaktion
65-0	-	DeviceNet aktiviert, aber kein Interface	
		Ursache	Die DeviceNet-Kommunikation ist im Parametersatz des Motorcontrollers aktiviert, es ist jedoch kein Interface verfügbar.
		Maßnahme	<ul style="list-style-type: none"> • Deaktivieren Sie die DeviceNet-Kommunikation. • schließen Sie ein Interface an.

Fehlergruppe 65		Fehler DeviceNet (nur CMMP-AS-...-M3)	
Nr.	Code	Meldung	Reaktion
65-1	-	DeviceNet: Timeout IO-Verbindung	
		Ursache	Unterbrechen einer I/O-Verbindung.
		Maßnahme	<ul style="list-style-type: none"> Innerhalb der erwarteten Zeit wurde keine I/O-Nachricht erhalten.

Fehlergruppe 68		Fehler EtherNet/IP (nur CMMP-AS-...-M3)	
Nr.	Code	Meldung	Reaktion
68-0	-	EtherNet/IP: Schwere Fehler	
		Ursache	Es ist ein schwerer interner Fehler aufgetreten. Dies kann z. B. durch ein defektes Interface ausgelöst werden.
		Maßnahme	<ul style="list-style-type: none"> Versuchen Sie den Fehler zu quittieren. Führen Sie einen Reset durch. Tauschen Sie das Interface aus. Falls der Fehler weiterhin besteht, kontaktieren Sie den Technischen Support.
68-1	-	EtherNet/IP: Allgemeiner Kommunikationsfehler	
		Ursache	Es wurde ein schwerer Fehler im EtherNet/IP Interface festgestellt.
		Maßnahme	<ul style="list-style-type: none"> Versuchen Sie den Fehler zu quittieren. Führen Sie einen Reset durch. Tauschen Sie das Interface aus. Falls der Fehler weiterhin besteht, kontaktieren Sie den Technischen Support.
68-2	-	EtherNet/IP: Verbindung wurde geschlossen	
		Ursache	Die Verbindung wurde über die Steuerung geschlossen.
		Maßnahme	Es muss eine neue Verbindung zur Steuerung aufgebaut werden.
68-3	-	EtherNet/IP: Verbindungsabbruch	
		Ursache	Während des Betriebs ist ein Verbindungsabbruch aufgetreten.
		Maßnahme	<ul style="list-style-type: none"> Überprüfen Sie die Verkabelung zwischen Motorcontroller und Steuerung. Bauen Sie eine neue Verbindung zur Steuerung auf.
68-6	-	EtherNet/IP: Doppelte Netzwerkadresse vorhanden	
		Ursache	Im Netzwerk befindet sich mindestens ein Gerät mit der gleichen IP-Adresse.
		Maßnahme	<ul style="list-style-type: none"> Verwenden Sie eindeutige IP-Adressen für alle Geräte im Netzwerk.

Fehlergruppe 69		Fehler EtherNet/IP (nur CMMP-AS-...-M3)	
Nr.	Code	Meldung	Reaktion
69-0	-	EtherNet/IP: Leichter Fehler	
		Ursache	Es wurde ein leichter Fehler im EtherNet/IP Interface festgestellt.
		Maßnahme	<ul style="list-style-type: none">• Versuchen Sie den Fehler zu quittieren.• Führen Sie einen Reset durch.
69-1	-	EtherNet/IP: Falsche IP-Konfiguration	
		Ursache	Es wurde eine falsche IP-Konfiguration festgestellt.
		Maßnahme	<ul style="list-style-type: none">• Korrigieren Sie die IP-Konfiguration.
69-2	-	EtherNet/IP: Feldbus-Interface nicht gefunden	
		Ursache	Im Einschubschacht befindet sich kein EtherNet/IP-Interface.
		Maßnahme	<ul style="list-style-type: none">• Bitte überprüfen Sie, ob ein EtherNet/IP-Interface im Einschubschacht Ext2 steckt.
69-3	-	EtherNet/IP: Interface Version nicht unterstützt	
		Ursache	Im Einschubschacht befindet sich ein EtherNet/IP-Interface mit inkompatibler Version.
		Maßnahme	<ul style="list-style-type: none">• Bitte führen Sie ein Firmware-Update auf die aktuellste Motorcontroller-Firmware durch.

Fehlergruppe 70		FHPP-Protokoll		
Nr.	Code	Meldung	Reaktion	
70-1	-	FHPP: Mathe-Fehler		konfigurierbar
		Ursache	Über-/Unterlauf oder Teilung durch Null während der Berechnung zyklischer Daten.	
		Maßnahme	<ul style="list-style-type: none">• Prüfen sie die zyklischen Daten.• Prüfen Sie die Factor Group.	
70-2	-	FHPP: Factor Group unzulässig		konfigurierbar
		Ursache	Berechnung der Factor Group führt zu ungültigen Werten.	
		Maßnahme	<ul style="list-style-type: none">• Prüfen Sie die Factor Group.	
70-3	-	FHPP: Unzulässiger Betriebsart-Wechsel		konfigurierbar
		Ursache	Wechseln vom aktuellen zum gewünschten Betriebsmodus ist nicht gestattet. <ul style="list-style-type: none">– Fehler tritt auf wenn die OPM-Bits im Status S5 'Reaction to fault' oder S4 'Operation enabled' geändert werden.– Ausnahme: Im Status SA1 'Ready' ist der Wechsel zwischen 'Record select' und 'Direct Mode' zulässig.	
		Maßnahme	<ul style="list-style-type: none">• Prüfen Sie Ihre Anwendung. Es kann sein, dass nicht jeder Wechsel zulässig ist.	

Fehlergruppe 71		Fehler FHPP-Protokoll		
Nr.	Code	Meldung	Reaktion	
71-1	-	FHPP: Ungültiges Empfangstelegramm		konfigurierbar
		Ursache	Es werden von der Steuerung zu wenig Daten übertragen (Datenlänge zu klein).	
		Maßnahme	<ul style="list-style-type: none">• Prüfen der in der Steuerung parametrisierten Datenlänge für das Empfangstelegramm des Controllers.• Prüfen der konfigurierten Datenlänge im FHPP+ Editor vom FCT.	
71-2	-	FHPP: Ungültiges Antworttelegramm		konfigurierbar
		Ursache	Es sollen vom Motorcontroller zu viele Daten zur Steuerung übertragen werden (Datenlänge zu groß).	
		Maßnahme	<ul style="list-style-type: none">• Prüfen der in der Steuerung parametrisierten Datenlänge für das Empfangstelegramm des Controllers.• Prüfen der konfigurierten Datenlänge im FHPP+ Editor vom FCT.	

Fehlergruppe 72		Fehler PROFINET (nur CMMP-AS-...-M3)	
Nr.	Code	Meldung	Reaktion
72-0	-	PROFINET: Fehlerhafte Initialisierung	
		Ursache	Interface enthält vermutlich eine nicht kompatible Stack-Version oder ist defekt.
		Maßnahme	• Interface tauschen.
72-1	-	PROFINET: Busfehler	
		Ursache	Keine Kommunikation möglich (z.B. Leitung abgezogen).
		Maßnahme	• Überprüfen der Verkabelung • PROFINET-Kommunikation neu starten.
72-3	-	PROFINET: Ungültige IP-Konfiguration	
		Ursache	Es wurde eine ungültige IP-Konfiguration in das Interface eingetragen. Mit dieser kann das Interface nicht starten.
		Maßnahme	• Parametrieren Sie über FCT eine zulässige IP-Konfiguration.
72-4	-	PROFINET: Ungültige Geräte name	
		Ursache	Es wurde ein PROFINET-Geräte name vergeben, mit dem der Controller nicht am PROFINET kommunizieren kann (Zeichen-Vorgabe aus PROFINET Norm).
		Maßnahme	• Parametrieren Sie über FCT einen zulässigen PROFINET-Geräte name.
72-5	-	PROFINET: Interface defekt	
		Ursache	Interface CAMC-F-PN defekt.
		Maßnahme	• Interface tauschen.
72-6	-	PROFINET: Ungültige/nicht unterstützte Indication	
		Ursache	Vom PROFINET-Interface kam eine Meldung die vom Motorcontroller nicht unterstützt wird.
		Maßnahme	• Bitte nehmen Sie Kontakt zum Technischen Support auf.

Fehlergruppe 73		Fehler PROFIenergy (nur CMMP-AS-...-M3)	
Nr.	Code	Meldung	Reaktion
73-0	-	PROFIenergy: Zustand nicht möglich	
		konfigurierbar	
		Ursache	Es wurde versucht in einer Verfahrbewegung den Controller in den Energiesparzustand zu versetzen. Dies ist nur im Stillstand möglich. Der Antrieb nimmt den Zustand nicht ein und verfährt weiterhin.
		Maßnahme	–

Fehlergruppe 80		Überlauf IRQ	
Nr.	Code	Meldung	Reaktion
80-0	F080h	Überlauf Stromregler IRQ	
		PS off	
		Ursache	Berechnung der Prozeßdaten konnte nicht in dem eingestellten Strom-/Drehzahl-/Lage-Interpolatorzyklus ausgeführt werden.
		Maßnahme	• Bitte nehmen Sie Kontakt zum Technischen Support auf.
80-1	F081h	Überlauf Drehzahlregler IRQ	
		PS off	
		Ursache	Berechnung der Prozeßdaten konnte nicht in dem eingestellten Strom-/Drehzahl-/Lage-Interpolatorzyklus ausgeführt werden.
		Maßnahme	• Bitte nehmen Sie Kontakt zum Technischen Support auf.
80-2	F082h	Überlauf Lageregler IRQ	
		PS off	
		Ursache	Berechnung der Prozeßdaten konnte nicht in dem eingestellten Strom-/Drehzahl-/Lage-Interpolatorzyklus ausgeführt werden.
		Maßnahme	• Bitte nehmen Sie Kontakt zum Technischen Support auf.
80-3	F083h	Überlauf Interpolator IRQ	
		PS off	
		Ursache	Berechnung der Prozeßdaten konnte nicht in dem eingestellten Strom-/Drehzahl-/Lage-Interpolatorzyklus ausgeführt werden.
		Maßnahme	• Bitte nehmen Sie Kontakt zum Technischen Support auf.

Fehlergruppe 81		Überlauf IRQ	
Nr.	Code	Meldung	Reaktion
81-4	F084h	Überlauf Low-Level IRQ	
		PS off	
		Ursache	Berechnung der Prozeßdaten konnte nicht in dem eingestellten Strom-/Drehzahl-/Lage-Interpolatorzyklus ausgeführt werden.
		Maßnahme	• Bitte nehmen Sie Kontakt zum Technischen Support auf.
81-5	F085h	Überlauf MDC IRQ	
		PS off	
		Ursache	Berechnung der Prozeßdaten konnte nicht in dem eingestellten Strom-/Drehzahl-/Lage-Interpolatorzyklus ausgeführt werden.
		Maßnahme	• Bitte nehmen Sie Kontakt zum Technischen Support auf.

Fehlergruppe 82		Ablaufsteuerung	
Nr.	Code	Meldung	Reaktion
82-0	-	Ablaufsteuerung	
		Ursache	Überlauf IRQ4 (10 ms Low-Level IRQ).
		Maßnahme	<ul style="list-style-type: none"> Interne Ablaufsteuerung: Prozess wurde abgebrochen. Nur zur Information - Keine Maßnahmen erforderlich.
82-1	-	Mehrfach gestarteter KO-Schreibzugriff	
		Ursache	Es werden Parameter im zyklischen und azyklischen Betrieb konkurrierend verwendet.
		Maßnahme	<ul style="list-style-type: none"> Es darf nur eine Parametrierschnittstelle verwendet werden (USB oder Ethernet).

Fehlergruppe 83		Fehler Interface (nur CMMP-AS-...-M3)	
Nr.	Code	Meldung	Reaktion
83-0	-	Ungültiges Optionsmodul	
		Ursache	<ul style="list-style-type: none"> Das gesteckte Interface konnte nicht erkannt werden. die geladene Firmware nicht bekannt. Ein unterstütztes Interface ist eventuell auf dem falschen Steckplatz (z. B. SERCOS 2, EtherCAT).
		Maßnahme	<ul style="list-style-type: none"> Firmware prüfen ob Interface unterstützt wird. Wenn ja: Interface prüfen, ob es auf dem richtigen Platz sitzt und korrekt gesteckt ist. Interface und/oder Firmware tauschen.
83-1	-	Nicht unterstütztes Optionsmodul	
		Ursache	Das gesteckte Interface konnte erkannt werden, wird aber von der geladenen Firmware nicht unterstützt.
		Maßnahme	<ul style="list-style-type: none"> Firmware prüfen ob Interface unterstützt wird. Ggf. Firmware tauschen.
83-2	-	Optionsmodul: HW-Revision nicht unterstützt	
		Ursache	Das gesteckte Interface konnte erkannt werden und auch prinzipiell unterstützt. In diesem Fall jedoch nicht die aktuelle Hardwareversion (weil sie zu alt ist).
		Maßnahme	<ul style="list-style-type: none"> Das Interface muss getauscht werden. Hier ggf. Kontakt zum technischen Support aufnehmen.

Fehlergruppe 84		Bedingungen für Reglerfreigabe nicht erfüllt	
Nr.	Code	Meldung	Reaktion
84-0	-	Bedingungen für Reglerfreigabe nicht erfüllt	
		Warn	
		Ursache	<p>Eine oder mehrere Bedingungen zur Reglerfreigabe sind nicht erfüllt. Dazu gehören:</p> <ul style="list-style-type: none"> – DIN4 (Endstufenfreigabe) ist aus. – DIN5 (Reglerfreigabe) ist aus. – Zwischenkreis noch nicht geladen. – Geber ist noch nicht betriebsbereit. – Winkelgeber-Identifikation ist noch aktiv. – Automatische Stromregler-Identifikation ist noch aktiv. – Geberdaten sind ungültig. – Statuswechsel der Sicherheitsfunktion noch nicht abgeschlossen. – FW- oder DCO-Download über Ethernet (TFTP) aktiv. – DCO-Download auf Speicherkarte noch aktiv. – FW-Download über Ethernet aktiv.
		Maßnahme	<ul style="list-style-type: none"> • Zustand digitale Eingänge prüfen. • Encoderleitungen prüfen. • automatische Identifikation abwarten. • Fertigstellung des FW- bzw. DCO Downloads abwarten.

Fehlergruppe 90		Interner Fehler	
Nr.	Code	Meldung	Reaktion
90-0	5080h	Fehlende Hardwarekomponente (SRAM)	
		PS off	
		Ursache	<p>Externes SRAM nicht erkannt / nicht ausreichend. Hardware-Fehler (SRAM-Bauteil oder Platine defekt).</p>
		Maßnahme	<ul style="list-style-type: none"> • Bitte nehmen Sie Kontakt zum Technischen Support auf.
90-2	5080h	Fehler beim Booten FPGA	
		PS off	
		Ursache	<p>Kein Booten des FPGA (Hardware) möglich. Das FPGA wird nach Start des Gerätes seriell gebootet, konnte aber in diesem Fall nicht mit Daten geladen werden oder es hat einen Checksummenfehler zurückgemeldet.</p>
		Maßnahme	<ul style="list-style-type: none"> • Gerät erneut einschalten (24 V). Wenn der Fehler wiederholt auftritt, ist die Hardware defekt.
90-3	5080h	Fehler bei Start SD-ADUs	
		PS off	
		Ursache	<p>Kein Start SD-ADUs (Hardware) möglich. Einer oder mehrere SD-ADUs liefern keine seriellen Daten.</p>
		Maßnahme	<ul style="list-style-type: none"> • Gerät erneut einschalten (24 V). Wenn der Fehler wiederholt auftritt, ist die Hardware defekt.

Fehlergruppe 90		Interner Fehler	
Nr.	Code	Meldung	Reaktion
90-4	5080h	Synchronisationsfehler SD-ADU nach Start	
		PS off	
		Ursache	SD-ADU (Hardware) nach Start nicht synchron. Im Betrieb laufen die SD-ADUs für die Resolersignale streng synchron weiter, nachdem sie einmalig synchron gestartet wurden. Bereits in der Startphase konnten die SD-ADUs nicht gleichzeitig angestartet werden.
		Maßnahme	<ul style="list-style-type: none"> Gerät erneut einschalten (24 V). Wenn der Fehler wiederholt auftritt, ist die Hardware defekt.
90-5	5080h	SD-ADU nicht synchron	
		PS off	
		Ursache	SD-ADU (Hardware) nach Start nicht synchron. Im Betrieb laufen die SD-ADUs für die Resolersignale streng synchron weiter, nachdem sie einmalig synchron gestartet wurden. Das wird im Betrieb laufend überprüft und ggf. ein Fehler ausgelöst.
		Maßnahme	<ul style="list-style-type: none"> Möglicherweise eine massive EMV-Einkopplung. Gerät erneut einschalten (24 V). Wenn der Fehler wiederholt auftritt, ist die Hardware defekt.
90-6	5080h	IRQ0 (Stromregler): Trigger-Fehler	
		PS off	
		Ursache	Endstufe triggert nicht den SW-IRQ der dann den Stromregler bedient. Ist höchstwahrscheinlich ein Hardware-Fehler auf der Platine oder im Prozessor.
		Maßnahme	<ul style="list-style-type: none"> Gerät erneut einschalten (24 V). Wenn der Fehler wiederholt auftritt, ist die Hardware defekt.
90-9	5080h	DEBUG-Firmware geladen	
		PS off	
		Ursache	Eine für den Debugger compilierte Entwicklungsversion wurde regulär geladen.
		Maßnahme	<ul style="list-style-type: none"> Firmware-Version prüfen, ggf. Update der Firmware.

Fehlergruppe 91		Initialisierungsfehler	
Nr.	Code	Meldung	Reaktion
91-0	6000h	Interner Initialisierungsfehler	
		PS off	
		Ursache	Internes SRAM zu klein für die compilierte Firmware. Kann nur bei Entwicklungsversionen auftreten.
		Maßnahme	<ul style="list-style-type: none"> Firmware-Version prüfen, ggf. Update der Firmware.
91-1	-	Speicher-Fehler beim Kopieren	
		PS off	
		Ursache	Firmwareteile wurden beim Start nicht korrekt vom externen FLASH ins interne RAM kopiert.
		Maßnahme	<ul style="list-style-type: none"> Gerät erneut einschalten (24 V). Wenn der Fehler nachhaltig auftritt, Firmware-Version prüfen, ggf. Update der Firmware.

Fehlergruppe 91		Initialisierungsfehler	
Nr.	Code	Meldung	Reaktion
91-2	-	Fehler beim Auslesen der Controller-/Leistungsteilcodierung	
		Ursache	Das ID-EEPROM im Controller oder dem Leistungsteil konnte entweder gar nicht erst angesprochen werden oder hat keine konsistenten Daten.
		Maßnahme	<ul style="list-style-type: none"> Gerät erneut einschalten (24 V). Wenn der Fehler nachhaltig auftritt, ist die HW defekt. Keine Reparatur möglich.
91-3	-	SW-Initialisierungsfehler	
		Ursache	Eine der folgenden Komponenten fehlt oder konnte nicht initialisiert werden: <ul style="list-style-type: none"> a) Shared Memory nicht vorhanden bzw. fehlerhaft. b) Treiberbibliothek nicht vorhanden bzw. fehlerhaft.
		Maßnahme	<ul style="list-style-type: none"> Firmware-Version prüfen, ggf. Update.

Hinweise zu den Maßnahmen bei den Fehlermeldungen 08-2 ... 08-7

Maßnahme	Hinweise
<ul style="list-style-type: none"> Prüfen ob Gebersignale gestört sind. 	<ul style="list-style-type: none"> Verkabelung prüfen, z. B. eine oder mehrere Phasen der Spursignale unterbrochen oder kurzgeschlossen? Installation auf EMV-Empfehlungen prüfen (Kabelschirm beidseitig aufgelegt?). Nur bei Inkrementalgebern: Bei TTL single ended Signalen (HALL-Signale sind immer TTL single ended Signale): Prüfen, ob ggf. ein zu hoher Spannungsabfall auf der GND-Leitung auftritt, in diesem Fall = Signalreferenz. Prüfen, ob ggf. ein zu hoher Spannungsabfall auf der GND-Leitung auftritt, in diesem Fall = Signalreferenz. Pegel der Versorgungsspannung am Geber prüfen. Ausreichend? Falls nicht Kabelquerschnitt anpassen (nicht benutzte Leitungen parallel schalten) oder Spannungsrückführung (SENSE+ und SENSE-) verwenden.
<ul style="list-style-type: none"> Test mit anderen Gebern. 	<ul style="list-style-type: none"> Tritt der Fehler bei korrekter Konfiguration immer noch auf, Test mit einem anderen (fehlerfreien) Geber (auch die Anschlussleitung tauschen). Tritt der Fehler dann immer noch auf, liegt ein Defekt im Motorcontroller vor. Reparatur durch Hersteller erforderlich.

Tab. B.2 Hinweise zu Fehlermeldungen 08-2 ... 08-7

Stichwortverzeichnis

A			
acceleration_factor	85	buffer_organisation	200
actual_dc_link_circuit_voltage	93	buffer_position	200
actual_size	200		
Aktuelle Zwischenkreisspannung	93	C	
analog_input_offset	131	cob_id_sync	35
analog_input_offset_ch_0	131	cob_id_used_by_pdo	29
analog_input_offset_ch_1	131	commisioning_state	150
analog_input_offset_ch_2	132	control_effort	114
analog_input_voltage	130	controlword	161
analog_input_voltage_ch_0	130	– Bitbelegung	157, 160, 162
analog_input_voltage_ch_1	130	– Kommandos	162
analog_input_voltage_ch_2	131	– Objektbeschreibung	161
Analoge Eingänge		Controlword für Interpolationsdaten	197
– Eingangsspannung Kanal 0	130	current_actual_value	218
– Eingangsspannung Kanal 1	130	current_limitation	118
– Eingangsspannung Kanal 2	131	cycletime_current_controller	149
– Eingangsspannungen	130	cycletime_position_controller	149
– Offsetspannung Kanal 0	131	cycletime_tracectory_generator	150
– Offsetspannung Kanal 1	131	cycletime_velocity_controller	149
– Offsetspannung Kanal 2	132		
– Offsetspannungen	131	D	
Anschlag	185, 186	dc_link_circuit_voltage	218
Anschlussbelegung CAN	11	Default-Parameter laden	75
Anzahl gemappter Objekte	30	Device Control	155
Auswahl der Istwert Lage	127	dig_out_state_mapp_dout_1	134
Auswahl der Synchronisationsquelle	128	dig_out_state_mapp_dout_2	135
		dig_out_state_mapp_dout_3	135
		dig_out_state_mapp_ea88_0_high	137
		dig_out_state_mapp_ea88_0_low	136
B		digital_inputs	132
Beschleunigung		digital_outputs	133
– Brems- (Positionieren)	191	digital_outputs_data	133
– Schnellstop- (Positionieren)	191	digital_outputs_mask	133
Betriebsart	176, 177	digital_outputs_state_mapping	134
– Ändern der	176	Digitale Ausgänge	133
– Einstellen der	176	– Mapping	134
– Lesen der	177	– Mapping von CAMC-EA	136, 137
– Referenzfahrt	178	– Mapping von DOUT1	134
Betriebsart Drehzahlregelung	203	– Mapping von DOUT2	135
Betriebsart Momentenregelung	214	– Mapping von DOUT3	135
brake_delay_time	145	– Maske	133
Bremsverzögerungszeit	145	– Zustände	133
buffer_clear	201		

Digitale Eingänge	132	encoder_x2a_divisor	122
disable_operation_option_code	174	encoder_x2a_numerator	122
divisor		encoder_x2a_resolution	122
– acceleration_factor	86	encoder_x2b_counter	123
– position_factor	81	encoder_x2b_data_field	123
– velocity_encoder_factor	83	encoder_x2b_divisor	123
Drehzahl-Istwert	207	encoder_x2b_numerator	123
Drehzahlbegrenzter Momentenbetrieb	119	encoder_x2b_resolution	123
Drehzahlbegrenzung	119	end_velocity	190
– Quelle	119	Endschalter	138, 183, 184
– Skalierung	120	– Nothalt-Rampe	140
– Sollwert	119	– Polarität	138
Drehzahlregelung	203	Endstufenparameter	89
– Max. Motordrehzahl	210	– Freigabelogik	90
– Sollgeschwindigkeit	211	– Gerätenennspannung	92
– Stillstandsschwelle	209	– Gerätenennstrom	95
– Stillstandsschwellenzeit	210	– max. Zwischenkreisspannung	93
– Zielfenster	209	– Maximale Temperatur	92
– Zielfensterzeit	209	– Maximalstrom	95
– Zielgeschwindigkeit	211	– min. Zwischenkreisspannung	94
Drehzahlregler		– PWM-Frequenz	90
– Filterzeitkonstante	105	– Zwischenkreisspannung	93
– Parameter	105	error_management	152
– Verstärkung	105	error_register	37
– Zeitkonstante	105	Erweiterte Sinusmodulation	91
drive_data	90, 99, 115, 138, 144		
E		F	
Einstellen der Betriebsart	176	Factor Group	79
EMERGENCY-Message	36	– acceleration_factor	85
enable_dc_link_undervoltage_error	94	– polarity	88
enable_enhanced_modulation	91	– position_factor	80
enable_logic	90	– velocity_encoder_factor	83
encoder_emulation_data	126	Fahrkurven-Generator	188
encoder_emulation_offset	126	fault_reaction_option_code	175
encoder_emulation_resolution	126	Fehlermanagement	152
encoder_offset_angle	101	Filterzeitkonstante Synchrondrehzahl	129
encoder_x10_counter	125	firmware_custom_version	148
encoder_x10_data_field	124	firmware_main_version	147
encoder_x10_divisor	125	first_mapped_object	30
encoder_x10_numerator	124	Following_Error	106
encoder_x10_resolution	124	following_error_actuel_value	113
encoder_x2a_data_field	122	following_error_time_out	113
		following_error_window	113

fourth_mapped_object	31
Freigabelogik	90

G

Gerätenennspannung	92
Gerätenennstrom	95
Gerätesteuerung	155
Geschwindigkeit	
– bei der Referenzfahrt	181
– beim Positionieren	190
Grenzwert Schleppfehler	115

H

Herstellercode	145
Hinweise zur Dokumentation	7
home_offset	180
Homing Mode	178
– home_offset	180
– homing_acceleration	182
– homing_method	180
– homing_speeds	181
homing_acceleration	182
homing_method	180
homing_speeds	181
homing_switch_polarity	139
homing_switch_selector	140
homing_timeout	182

I

I2t-Auslastung	99
I2t-Zeit	99
Identifizier für PDO	29
Identifizierung des Geräts	145
identity_object	145
iit_error_enable	100
iit_ratio_motor	99
iit_time_motor	99
iit-Fehler auslösen	100
inhibit_time	29
Inkrementalgeberemulation	
– Auflösung	126
– Offset	126
interpolation_data_configuration	199

interpolation_data_record	196
interpolation_submode_select	196
interpolation_sync_definition	198
interpolation_time_period	197
Interpolations-Daten	196
Interpolations-Typ	196
ip_data_controlword	197
ip_data_position	197
ip_sync_every_n_event	199
ip_time_index	198
ip_time_units	198
Istwert	
– Lage in Inkrementen (position_actual_value_s)	112
– Lage in position_units (position_actual_value)	112
– Moment (torque_actual_value)	217

K

Korrekturgeschwindigkeit	110
--------------------------------	-----

L

Lage-Istwert (Inkrementen)	112
Lage-Istwert (position units)	112
Lageregler	106
– Ausgang des	114
– Parameter	110
– Totbereich	110
– Verstärkung	110
– Zeitkonstante	110
Lageregler-Parameter	110
Lagereglerausgang	114
Lagereglerverstärkung	110
Lagereglerzeitkonstante	110
Lagewert Interpolation	197
limit_current	118, 119
limit_current_input_channel	118
limit_speed_input_channel	119
limit_switch_deceleration	140
limit_switch_polarity	138

M

Mappingparameter für PDOs	30
max_buffer_size	199
max_current	98
max_dc_link_circuit_voltage	93
max_motor_speed	210
max_position_range_limit	116
max_power_stage_temperature	92
max_torque	216
Maximale Endstufentemperatur	92
Maximale Motordrehzahl	210
Maximale Zwischenkreisspannung	93
Maximales Moment	216
Maximalstrom	95
min_dc_link_circuit_voltage	94
min_position_range_limit	116
Minimale Zwischenkreisspannung	94
modes_of_operation	176
modes_of_operation_display	177
Momenten-Istwert	217
Momentenbegrenzter Drehzahlbetrieb	118
Momentenbegrenzung	118
– Quelle	118
– Skalierung	119
– Sollwert	118
Momentenregeln	214
Momentenregelung	
– Max. Moment	216
– Momenten-Istwert	217
– Nennmoment	217
– Sollmoment	216
– Sollwertprofil	219
– Stromsollwert	217
– Zielmoment	216
motion_profile_type	192
motor_data	99, 101
motorRatedCurrent	97
motorRatedTorque	217
motorTemperatureSensorPolarity	101
Motornennstrom	97
Motorparameter	
– I2t-Zeit	99
– Nennstrom	97

– Pol(paar)zahl	98
– Resolveroffsetwinkel	101
Motorspitzenstrom	98

N

Nennmoment des Motors	217
Nennstrom des Motors	97
Neue Position anfahren	193
nominalCurrent	95
nominal_dc_link_circuit_voltage	92
Not Ready to Switch On	159
Nullpunkt-Offset	180
number_of_mapped_objects	30
numerator	88
– acceleration_factor	86
numerator	
– position_factor	81
– velocity_encoder_factor	83

O**Objekte**

– Objekt 1001h	36
– Objekt 1003h	37
– Objekt 1003h_01h	38
– Objekt 1003h_02h	38
– Objekt 1003h_03h	38
– Objekt 1003h_04h	38
– Objekt 1005h	35
– Objekt 1010h	75
– Objekt 1010h_01h	76
– Objekt 1011h	75
– Objekt 1011h_01h	75
– Objekt 1018h	145
– Objekt 1018h_01h	145
– Objekt 1018h_02h	146
– Objekt 1018h_03h	146
– Objekt 1018h_04h	146
– Objekt 1100h	53
– Objekt 1402h	34
– Objekt 1403h	34
– Objekt 1602h	34
– Objekt 1603h	34
– Objekt 1800h	29, 31

– Objekt 1800h_01h	29	– Objekt 2022h	128
– Objekt 1800h_02h	29	– Objekt 2023h	129
– Objekt 1800h_03h	29	– Objekt 2024h	122
– Objekt 1801h	31	– Objekt 2024h_01h	122
– Objekt 1802h	32	– Objekt 2024h_02h	122
– Objekt 1803h	32	– Objekt 2024h_03h	122
– Objekt 1A00h	30, 31	– Objekt 2025h	124
– Objekt 1A00h_00h	30	– Objekt 2025h_01h	124
– Objekt 1A00h_01h	30	– Objekt 2025h_02h	124
– Objekt 1A00h_02h	30	– Objekt 2025h_03h	125
– Objekt 1A00h_03h	30	– Objekt 2025h_04h	125
– Objekt 1A00h_04h	31	– Objekt 2026h	123
– Objekt 1A01h	31	– Objekt 2026h_01h	123
– Objekt 1A02h	32	– Objekt 2026h_02h	123
– Objekt 1A03h	32	– Objekt 2026h_03h	123
– Objekt 1C00h	53	– Objekt 2026h_04h	123
– Objekt 1C00h_00h	54	– Objekt 2028h	126
– Objekt 1C00h_01h	54	– Objekt 202Dh	111
– Objekt 1C00h_02h	54	– Objekt 202Eh	207
– Objekt 1C00h_03h	54	– Objekt 202Fh	129
– Objekt 1C00h_04h	54	– Objekt 202Fh_07h	129
– Objekt 1C10h	55	– Objekt 2045h	182
– Objekt 1C11h	55	– Objekt 204Ah	141
– Objekt 1C12h	56	– Objekt 204Ah_01h	142
– Objekt 1C12h_00h	56	– Objekt 204Ah_02h	142
– Objekt 1C12h_01h	56	– Objekt 204Ah_03h	142
– Objekt 1C12h_02h	56	– Objekt 204Ah_04h	143
– Objekt 1C12h_03h	57	– Objekt 204Ah_05h	143
– Objekt 1C12h_04h	57	– Objekt 204Ah_06h	143
– Objekt 1C13h	57	– Objekt 2090h	212
– Objekt 1C13h_00h	57	– Objekt 2090h_01h	212
– Objekt 1C13h_01h	58	– Objekt 2090h_02h	212
– Objekt 1C13h_02h	58	– Objekt 2090h_03h	212
– Objekt 1C13h_03h	58	– Objekt 2090h_04h	213
– Objekt 1C13h_04h	58	– Objekt 2090h_05h	213
– Objekt 2014h	32	– Objekt 2100h	152
– Objekt 2015h	32	– Objekt 2400h	130
– Objekt 2016h	33	– Objekt 2400h_01h	130
– Objekt 2017h	33	– Objekt 2400h_02h	130
– Objekt 201Ah	126	– Objekt 2400h_03h	131
– Objekt 201Ah_01h	126	– Objekt 2401h	131
– Objekt 201Ah_02h	126	– Objekt 2401h_01h	131
– Objekt 2021h	127	– Objekt 2401h_02h	131

– Objekt 2401h_03h	132	– Objekt 6077h	217
– Objekt 2415h	118	– Objekt 6078h	218
– Objekt 2415h_01h	118	– Objekt 6079h	218
– Objekt 2415h_02h	118	– Objekt 607Ah	189
– Objekt 2416h	119	– Objekt 607Bh	116
– Objekt 2416h_01h	119	– Objekt 607Bh_01h	116
– Objekt 2416h_02h	119	– Objekt 607Bh_02h	116
– Objekt 2420h	134	– Objekt 607Ch	180
– Objekt 2420h_01h	134	– Objekt 607Eh	88
– Objekt 2420h_02h	135	– Objekt 6080h	210
– Objekt 2420h_03h	135	– Objekt 6081h	190
– Objekt 2420h_11h	136	– Objekt 6082h	190
– Objekt 2420h_12h	137	– Objekt 6083h	190
– Objekt 6040h	161	– Objekt 6084h	191
– Objekt 6041h	166	– Objekt 6085h	191
– Objekt 604Dh	98	– Objekt 6086h	192
– Objekt 605Ah	174	– Objekt 6087h	219
– Objekt 605Bh	173	– Objekt 6088h	219
– Objekt 605Ch	174	– Objekt 608Ah	59
– Objekt 605Eh	175	– Objekt 608Bh	59
– Objekt 6060h	176	– Objekt 608Ch	59
– Objekt 6061h	177	– Objekt 608Dh	59
– Objekt 6062h	111	– Objekt 608Eh	59
– Objekt 6063h	112	– Objekt 6093h	80
– Objekt 6064h	112	– Objekt 6093h_01h	81
– Objekt 6065h	113	– Objekt 6093h_02h	81
– Objekt 6066h	113	– Objekt 6094h	83
– Objekt 6067h	114	– Objekt 6094h_01h	83
– Objekt 6068h	115	– Objekt 6094h_02h	83
– Objekt 6069h	206	– Objekt 6097h	85
– Objekt 606Ah	206	– Objekt 6097h_01h	86
– Objekt 606Bh	206	– Objekt 6097h_02h	86
– Objekt 606Ch	207	– Objekt 6098h	180
– Objekt 606Dh	209	– Objekt 6099h	181
– Objekt 606Eh	209	– Objekt 6099h_01h	181
– Objekt 606Fh	209	– Objekt 6099h_02h	182
– Objekt 6070h	210	– Objekt 609Ah	182
– Objekt 6071h	216	– Objekt 60C0h	196
– Objekt 6072h	216	– Objekt 60C1h	196
– Objekt 6073h	98	– Objekt 60C1h_01h	197
– Objekt 6074h	217	– Objekt 60C1h_02h	197
– Objekt 6075h	97	– Objekt 60C2h	197
– Objekt 6076h	217	– Objekt 60C2h_01h	198

– Objekt 60C2h_02h	198	– Objekt 6510h_20h	117
– Objekt 60C3h	198	– Objekt 6510h_22h	115
– Objekt 60C3h_01h	199	– Objekt 6510h_30h	90
– Objekt 60C3h_02h	199	– Objekt 6510h_31h	91
– Objekt 60C4h	199	– Objekt 6510h_32h	92
– Objekt 60C4h_01h	199	– Objekt 6510h_33h	92
– Objekt 60C4h_02h	200	– Objekt 6510h_34h	93
– Objekt 60C4h_03h	200	– Objekt 6510h_35h	93
– Objekt 60C4h_04h	200	– Objekt 6510h_36h	94
– Objekt 60C4h_05h	200	– Objekt 6510h_37h	94
– Objekt 60C4h_06h	201	– Objekt 6510h_38h	100
– Objekt 60F4h	113	– Objekt 6510h_3Ah	91
– Objekt 60F6h	102	– Objekt 6510h_40h	95
– Objekt 60F6h_01h	103	– Objekt 6510h_41h	95
– Objekt 60F6h_02h	103	– Objekt 6510h_A9h	147
– Objekt 60F9h	104	– Objekt 6510h_AAh	148
– Objekt 60F9h_01h	105	– Objekt 6510h_B0h	149
– Objekt 60F9h_02h	105	– Objekt 6510h_B1h	149
– Objekt 60F9h_04h	105	– Objekt 6510h_B2h	149
– Objekt 60FAh	114	– Objekt 6510h_B3h	150
– Objekt 60FBh	109	– Objekt 6510h_C0h	150
– Objekt 60FBh_01h	110	Offset des Winkelgebers	101
– Objekt 60FBh_02h	110		
– Objekt 60FBh_04h	110	P	
– Objekt 60FBh_05h	110	Parameter einstellen	73
– Objekt 60FDh	132	Parametersatz sichern	76
– Objekt 60FEh	133	Parametersätze	
– Objekt 60FEh_01h	133	– Defaultwerte laden	75
– Objekt 60FEh_02h	133	– Laden und speichern	73
– Objekt 60FFh	211	– Parametersatz sichern	75
– Objekt 6410h	98	Parametriestatus	150
– Objekt 6410h_03h	99	PDO	25
– Objekt 6410h_04h	99	– 1. eingetragenes Objekt	30
– Objekt 6410h_10h	100	– 2. eingetragenes Objekt	30
– Objekt 6410h_11h	101	– 3. eingetragenes Objekt	30
– Objekt 6410h_14h	101	– 4. eingetragenes Objekt	31
– Objekt 6510h	90	– RPDO3	
– Objekt 6510h_10h	90	1. eingetragenes Objekt	34
– Objekt 6510h_11h	138	2. eingetragenes Objekt	34
– Objekt 6510h_13h	140	3. eingetragenes Objekt	34
– Objekt 6510h_14h	139	4. eingetragenes Objekt	34
– Objekt 6510h_15h	140	Anzahl eingetragener Objekte	34
– Objekt 6510h_18h	145	COB-ID used by PDO	34

first mapped object	34	1. eingetragenes Objekt	31
fourth mapped object	34	2. eingetragenes Objekt	31
Identifier	34	3. eingetragenes Objekt	31
number of mapped objects	34	4. eingetragenes Objekt	31
second mapped object	34	Anzahl eingetragener Objekte	31
third mapped object	34	COB-ID used by PDO	31
transmission type	34	first mapped object	31
Übertragungstyp	34	fourth mapped object	31
– RPDO4		Identifier	31
1. eingetragenes Objekt	34	inhibit time	31
2. eingetragenes Objekt	34	number of mapped objects	31
3. eingetragenes Objekt	34	second mapped object	31
4. eingetragenes Objekt	34	Sperrzeit	31
Anzahl eingetragener Objekte	34	third mapped object	31
COB-ID used by PDO	34	transmission type	31
first mapped object	34	Übertragungsmaske	32
fourth mapped object	34	Übertragungstyp	31
Identifier	34	– TPDO3	
number of mapped objects	34	1. eingetragenes Objekt	32
second mapped object	34	2. eingetragenes Objekt	32
third mapped object	34	3. eingetragenes Objekt	32
transmission type	34	4. eingetragenes Objekt	32
Übertragungstyp	34	Anzahl eingetragener Objekte	32
– TPDO1		COB-ID used by PDO	32
1. eingetragenes Objekt	31	first mapped object	32
2. eingetragenes Objekt	31	fourth mapped object	32
3. eingetragenes Objekt	31	Identifier	32
4. eingetragenes Objekt	31	inhibit time	32
Anzahl eingetragener Objekte	31	number of mapped objects	32
COB-ID used by PDO	31	second mapped object	32
first mapped object	31	Sperrzeit	32
fourth mapped object	31	third mapped object	32
Identifier	31	transmission type	32
inhibit time	31	Übertragungsmaske	33
number of mapped objects	31	Übertragungstyp	32
second mapped object	31	– TPDO4	
Sperrzeit	31	1. eingetragenes Objekt	32
third mapped object	31	2. eingetragenes Objekt	32
transmission type	31	3. eingetragenes Objekt	32
Übertragungsmaske	32	4. eingetragenes Objekt	32
Übertragungstyp	31	Anzahl eingetragener Objekte	32
– TPDO2		COB-ID used by PDO	32

first mapped object	32	– Zielposition	189
fourth mapped object	32	Positionierprofil	
Identifier	32	– Lineares	192
inhibit time	32	– Ruckfreies	192
number of mapped objects	32	– Sinus2	192
second mapped object	32	Positionierung starten	193
Sperrzeit	32	Positionswert Interpolation	197
third mapped object	32	power_stage_temperature	91
transmission type	32	pre_defined_error_field	37
Übertragungsmaske	33	product_code	146
Übertragungstyp	32	Produktcode	146
PDO-Message	25	Profile Position Mode	
peak_current	95	– end_velocity	190
phase_order	100	– motion_profile_type	192
Polarität Motortemperatursensor	101	– profile_acceleration	190
pole_number	98	– profile_deceleration	191
Polpaarzahl	98	– profile_velocity	190
Polzahl	98	– quick_stop_deceleration	191
position_actual_value	112	– target_position	189
position_actual_value_s	112	Profile Torque Mode	214
position_control_gain	110	– current_actual_value	218
position_control_parameter_set	110	– dc_link_circuit_voltage	218
position_control_time	110	– max_torque	216
position_control_v_max	110	– motor_rated_torque	217
position_demand_sync_value	111	– target_torque	216
position_demand_value	111	– torque_actual_value	217
position_encoder_selection	127	– torque_demand_value	217
position_error_switch_off_limit	115	– torque_profile_type	219
position_error_tolerance_window	110	– torque_slope	219
position_factor	80	Profile Velocity Mode	203
position_range_limit	116	– max_motor_speed	210
position_range_limit_enable	117	– sensor_selection_code	206
position_reached	107	– target_velocity	211
position_window	114	– velocity_actual_value	207
position_window_time	115	– velocity_demand_value	206
position-control-function	106	– velocity_sensor	206
Positionier-Bremsbeschleunigung	191	– velocity_threshold	209
Positionier-Geschwindigkeit	190	– velocity_threshold_time	210
Positionieren	193	– velocity_window	209
– Bremsbeschleunigung	191	– velocity_window_time	209
– Geschwindigkeit beim	190	profile_acceleration	190
– Handshake	193	profile_deceleration	191
– Schnellstop-Beschleunigung	191	profile_velocity	190

pwm_frequency	90	sample_mode	142
PWM-Frequenz	90	sample_position_falling_edge	143
Q		sample_position_rising_edge	143
quick_stop_deceleration	191	sample_status	142
quick_stop_option_code	174	sample_status_mask	142
R		SAMPLE-Eingang als Referenzschalter	140
R-PDO 3	34	Sampling-Position	
R-PDO4	34	– Fallende Flanke	143
Ready to Switch On	159	– Steigende Flanke	143
Receive_PDO_3	34	save_all_parameters	76
Receive_PDO_4	34	Schleppfehler	106
Referenzfahrt	178	– Fehlerfenster	113
– Steuerung der	187	– Grenzwert-Überschreitung	115
– Timeout	182	– Timeoutzeit	113
Referenzfahrt-Methode	181	Schleppfehler aktueller Wert	113
Referenzfahrt-Methoden	183	Schleppfehler-Timeoutzeit	113
Referenzfahrten		Schleppfehlerfenster	113
– Geschwindigkeiten	181	Schnellstop-Beschleunigung	191
– Kriechgeschwindigkeit	182	SDO	21
– Methode	181	SDO-Fehlermeldungen	23
– Nullpunkt-Offset	180	SDO-Message	20
– Suchgeschwindigkeit	181	second_mapped_object	30
Referenzschalter	138, 140	sensor_selection_code	206
– Polarität	139	serial_number	146
Regler-Freigabelogik	90	Service	7
Reglerfehler	36	shutdown_option_code	173
resolver_offset_angle	101	size_of_data_record	200
Resolveroffsetwinkel	101	Skalierungsfaktoren	79
restore_all_default_parameters	75	– Positionsfaktor	81
restore_parameters	75	– Vorzeichenwahl	88
revision_number	146	Sollgeschwindigkeit für Drehzahlregelung ..	211
Revisionsnummer CANopen	146	Sollmoment (Momentenregelung)	216
S		Sollwert	
Sample		– Moment	216
– Modus	142	– Strom	217
– Status	142	– Synchrondrehzahl (velocity units)	207
– Statusmaske	142	speed_during_search_for_switch	181
– Steuerung	143	speed_during_search_for_zero	182
sample_control	143	speed_limitation	119
sample_data	141	Spitzenstrom	
		– Motor	98
		– Motorcontroller	95
		standard_error_field_0	38

standard_error_field_1	38
standard_error_field_2	38
standard_error_field_3	38
START-Eingang als Referenzschalter	140
State	
– Not Ready to Switch On	159
– Ready to Switch On	159
– Switch On Disabled	159
– Switched On	159
statusword	
– Bitbelegung	166
– Objektbeschreibung	166
Steuerung des Reglers	155
Stillstandschwelle bei Drehzahlregelung ...	209
Stillstandsschwellenzeit bei Drehzahlregelung ...	210
store_parameters	75
Strombegrenzung	118
Stromregler	
– Parameter	103
– Verstärkung	103
– Zeitkonstante	103
Stromsollwert	217
Switch On Disabled	159
SYNC	35
SYNC-Message	35
Synchrondrehzahl (velocity units)	207
synchronisation_encoder_selection	128
synchronisation_filter_time	129
synchronisation_main	129
synchronisation_selector_data	129
synchronize_on_group	199

T

T-PDO 1	31
T-PDO 2	31
T-PDO 3	32
T-PDO 4	32
target_position	189
target_torque	216
target_velocity	211
Technische Daten Interface CANopen	220
third_mapped_object	30

torque_actual_value	217
torque_control_gain	103
torque_control_parameters	103
torque_control_time	103
torque_demand_value	217
torque_profile_type	219
torque_slope	219
tpdo_1_transmit_mask	32
tpdo_2_transmit_mask	32
tpdo_3_transmit_mask	33
tpdo_4_transmit_mask	33
transfer_PDO_1	31
transfer_PDO_2	31
transfer_PDO_3	32
transfer_PDO_4	32
transmission_type	29
transmit_pdo_mapping	30
transmit_pdo_parameter	29

Ü

Überschreitung Grenzwert Schleppfehler ...	115
Übertragungsart	29
Übertragungsparameter für PDOs	29
Umrechnungsfaktoren	79
– Positionsfaktor	81
– Vorzeichenwahl	88
Unterspannungsüberwachung aktivieren	94
Unterspannungsüberwachung deaktivieren ..	94

V

velocity_acceleration_neg	213
velocity_acceleration_pos	212
velocity_actual_value	207
velocity_control_filter_time	105
velocity_control_gain	105
velocity_control_parameter_set	105
velocity_control_time	105
velocity_deceleration_neg	213
velocity_deceleration_pos	212
velocity_demand_sync_value	207
velocity_demand_value	206
velocity_encoder_factor	83
velocity_ramps	212

velocity_ramps_enable	212	– Zähler	123
velocity_sensor_actual_value	206		
velocity_threshold	209	Z	
velocity_threshold_time	210	Zeitkonstante des Stromreglers	103
velocity_window	209	Zielfenster	
velocity_window_time	209	– Positionsfenster	114
vendor_id	145	– Zeit	115
Verhalten bei Kommando		Zielfenster bei Drehzahlregelung	209
– disable operation	174	Zielfensterzeit	115
– quick stop	174	Zielfensterzeit bei Drehzahlregelung	209
– shutdown	173	Zielgeschwindigkeit für Drehzahlregelung ..	211
Version	7	Zielgruppe	7
Versionsnummer der Firmware	147	Zielmoment (Momentenregelung)	216
Versionsnummer der kundenspez. Variante ..	148	Zielposition	189
Verstärkung des Stromreglers	103	Zielpositionsfenster	114
		Zulässiges Moment	216
W		Zustand	
Winkelgeberoffset	101	– Not Ready to Switch On	159
		– Ready to Switch On	159
X		– Switch On Disabled	159
X10		– Switched On	159
– Abtrieb	125	Zwischenkreisspannung	
– Antrieb	124	– aktuelle	93
– Auflösung	124	– maximale	93
– Zähler	125	– minimale	94
X2A		Zwischenkreisüberwachung	93, 94
– Abtrieb	122	Zykluszeit	
– Antrieb	122	– Drehzahlregler	149
– Auflösung	122	– Lageregler	149
X2B		– Positioniersteuerung	150
– Abtrieb	123	– Stromregler	149
– Antrieb	123	Zykluszeit PDOs	29
– Auflösung	123		

Copyright:
Festo AG & Co. KG
Postfach
D-73726 Esslingen

Phone:
+49 711 347 0

Fax:
+49 711 347 2144

e-mail:
service_international@festo.com

Weitergabe sowie Vervielfältigung dieses Dokuments, Verwertung und Mitteilung seines Inhalts sind verboten, soweit nicht ausdrücklich gestattet. Zuwiderhandlungen verpflichten zu Schadenersatz. Alle Rechte sind für den Fall der Patent-, Gebrauchsmuster- oder Geschmacksmustereintragung vorbehalten.

Internet:
www.festo.com

Original: de
Version: 1304a