FAULHABER

Kommunikations-handbuch

MC 5010
MC 5005
MC 5004
MC 5004 P STO
MCS
MC 3001
MC 3603
IMC

CANopen®

Impressum

Version:
7. Auflage, 2.02.2024

Dieses Dokument wurde mit Sorgfalt erstellt.
Die Dr. Fritz Faulhaber GmbH & Co. KG übernimmt jedoch für eventuelle Irrtümer in diesem Dokument und deren Folgen keine Haftung. Ebenso wird keine Haftung für direkte Schäden oder Folgeschäden übernommen, die sich aus einem unsachgemäßen Gebrauch der Geräte ergeben.

Bei der Anwendung der Geräte sind die einschlägigen Vorschriften bezüglich Sicherheitstechnik und Funkentstörung sowie die Vorgaben dieses Dokuments zu beachten.

Änderungen vorbehalten.

Die jeweils aktuelle Version dieses Dokuments finden Sie auf der Internetseite von FAULHABER:
www.faulhaber.com

Inhalt

Inhalt

1 Zu diesem Dokument

1.1 Gültigkeit dieses Dokuments

Dieses Dokument beschreibt:

- Kommunikation mit dem Antrieb über CANopen
- Basisdienste der Kommunikationsstruktur
- Methoden für den Parameterzugriff
- Antrieb aus Kommunikationssicht

Dieses Dokument richtet sich an Softwareentwickler mit CAN-BUS-Erfahrung und an CAN-BUS-Projektingenieure.

Alle Angaben in diesem Dokument beziehen sich auf Standardausführungen der Antriebe. Änderungen aufgrund kundenspezifischer Ausführungen dem entsprechenden Datenblatt entnehmen.

Alle Angaben in diesem Dokument beziehen sich auf die Firmware-Revision M.

1.2 Mitgeltende Dokumente

Für bestimmte Handlungsschritte bei der Inbetriebnahme und Bedienung der FAULHABER Produkte sind zusätzliche Informationen aus folgenden Handbüchern hilfreich:

Handbuch	Beschreibung
Motion Manager 6	Bedienungsanleitung zur FAULHABER Motion Manager PC Software
Schnellstartanleitung	Beschreibung der ersten Schritte zur Inbetriebnahme und Bedienung des FAULHABER Motion Controllers
Antriebsfunktionen	Beschreibung der Betriebsarten und Funktionen des Antriebs
Gerätehandbuch	Anleitung zur Installation und zum Gebrauch des FAULHABER Motion Controllers
CiA 301	CANopen application layer and communication profile
CiA 402	CANopen device profile for drives and motion control

Diese Handbücher können im PDF-Format von der Internetseite www.faulhaber.com/manuals heruntergeladen werden.

1.3 Umgang mit diesem Dokument

- Dokument vor der Konfiguration aufmerksam lesen.
- Dokument während der Lebensdauer des Produkts aufbewahren.
- Dokument dem Bedienpersonal jederzeit zugänglich halten.
- Dokument an jeden nachfolgenden Besitzer oder Benutzer des Produkts weitergeben.

1.4 Abkürzungsverzeichnis

Abkürzung	Bedeutung
Attr.	Attribut
CAN	Controller Area Network
CiA	CAN in Automation e.V.
COB ID	Communication Object Identifier
CS	Command Specifier
EEPROM	Electrically Erasable Programmable Read-Only Memory
EMCY	Emergency
HB	High Byte
HHB	Higher High Byte
HLB	Higher Low Byte
LB	Low Byte
LHB	Lower High Byte
LLB	Lower Low Byte
LSB	Least Significant Byte
LSS	Layer Setting Service
MSB	Most Significant Byte
NMT	CANopen Netzwerkmanagement
OD	Objektverzeichnis
PDO	Prozessdatenobjekt
PP	Profile Position
PV	Profile Velocity
ro	read only
RTR	Remote Request
rw	read-write
RxPDO	Receive-Prozessdatenobjekt (vom Antrieb empfangenes PDO)
SDO	Servicedatenobjekt
PLC	Speicherprogrammierbare Steuerung
Sxx	Datentyp Signed (negative und positive Zahlen) mit Bitgröße xx
SYNC	Synchronisationsobjekt
TxPDO	Transmit-Prozessdatenobjekt (vom Antrieb gesendetes PDO)
Uxx	Datentyp Unsigned (positive Zahlen) mit Bitgröße xx

1.5 Symbole und Kennzeichnungen

HINWEIS!
Gefahr von Sachschäden.

- Maßnahme zur Vermeidung

Hinweise zum Verständnis oder zum Optimieren der Arbeitsabläufe

✓ Voraussetzung zu einer Handlungsaufforderung

1. Erster Schritt einer Handlungsaufforderung
   - Resultat eines Schritts
2. Zweiter Schritt einer Handlungsaufforderung

Resultat einer Handlung

- Einschrittige Handlungsaufforderung

2 Überblick

2.1 Grundaufbau eines CANopen-Geräts

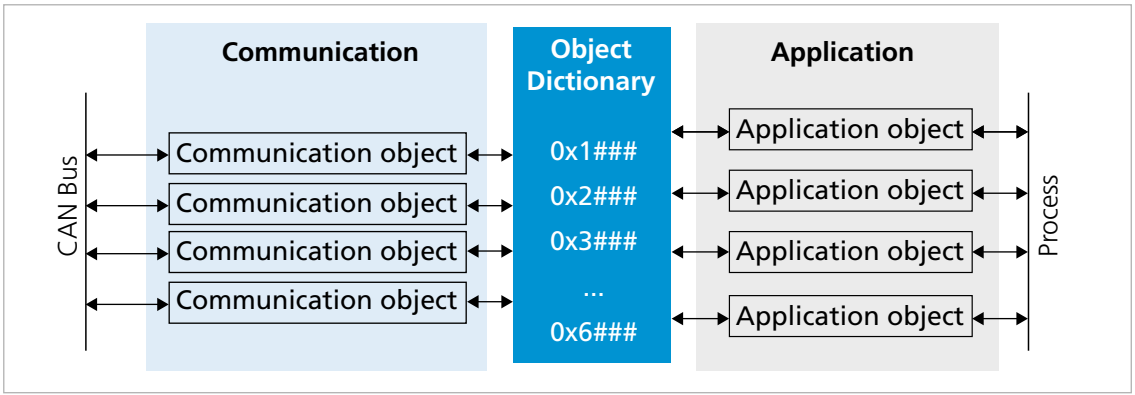

Abb. 1: Grundaufbau eines CANopen-Geräts

Kommunikationsdienste
Der CANopen-Master kommuniziert über das Bussystem und unter Verwendung der Kommunikationsdienste mit dem Objektverzeichnis (siehe Kap. 3.2, S. 16).

Objektverzeichnis
Das Objektverzeichnis enthält Parameter, Soll- und Istwerte eines Antriebs. Das Objektverzeichnis ist das Bindeglied zwischen der Anwendung (Antriebsfunktionen) und den Kommunikationsdiensten. Alle Objekte im Objektverzeichnis sind über eine 16-Bit-Indexnummer (0x1000 bis 0x6FFF) und einen 8-Bit-Subindex (0x00 bis 0xFF) ansprechbar.

Index	Zuordnung der Objekte
0x1000 bis 0x1FFF	Kommunikationsobjekte
0x2000 bis 0x5FFF	Herstellerspezifische Objekte
0x6000 bis 0x6FFF	Objekte des Antriebsprofils nach CiA 402

Die Werte der Parameter können von Kommunikationsseite sowie von der Antriebsseite geändert werden.

Anwendungsteil
Der Anwendungsteil enthält Antriebsfunktionen entsprechend CiA 402. Die Antriebsfunktionen lesen Parameter aus dem Objektverzeichnis, erhalten vom Objektverzeichnis Sollwerte und geben Istwerte zurück. Die Parameter aus dem Objektverzeichnis bestimmen das Antriebsverhalten.

Auf den Anwendungsteil wird in diesem Dokument nicht näher eingegangen. Die Kommunikation mit dem Antrieb und die zugehörigen Betriebsarten sind im separaten Handbuch „Antriebsfunktionen" beschrieben.

2.2 Voraussetzungen für die Kommunikation

Die FAULHABER-Antriebe werden im unkonfigurierten Zustand ausgeliefert. Für den Betrieb in einem CAN-Netzwerk müssen bei der Erstinbetriebnahme eine eindeutige Knotennummer und eine Baudrate eingestellt werden (siehe Kap. 4, S. 38).

Nach dem Einschalten und der Initialisierung befindet sich der Motion Controller zunächst im Zustand **Pre-Operational.** Um Antriebsfunktionen ausführen zu können, muss der Motion Controller in den Zustand **Operational** gebracht werden (siehe Kap. 3.8, S. 29).

1. Controller an eine Spannungsversorgung (mindestens Elektronikversorgung) anschließen.
2. CAN_H, CAN_L, GND mit den entsprechenden Anschlüssen eines host-seitigen CAN-Anschlusses verbinden.
3. Spannung einschalten und über die Konfigurationsanwendung Verbindung herstellen.

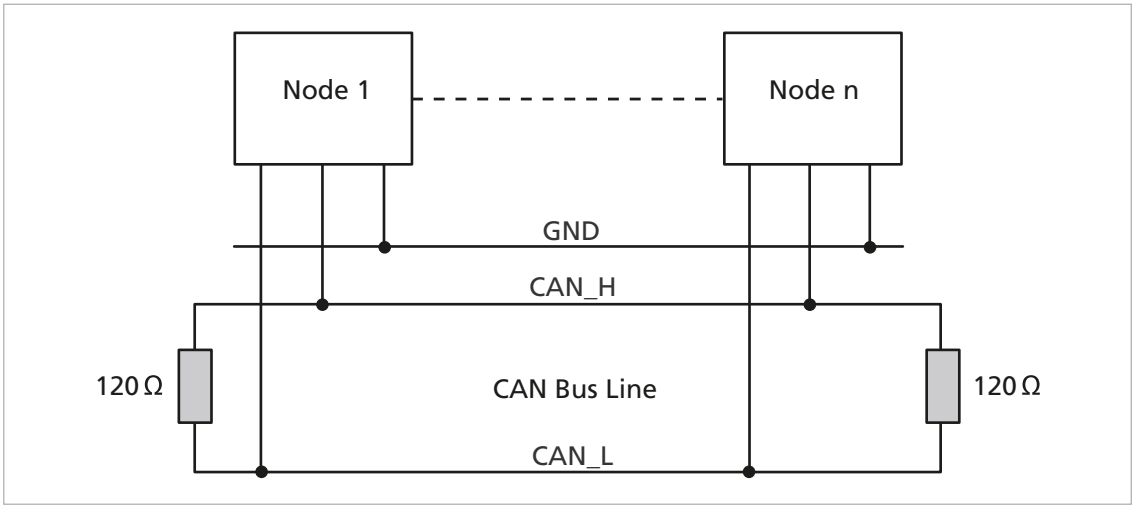

Abb. 2: Anschluss im CANopen-Netzwerk

2.3 FAULHABER Motion Manager

Es wird empfohlen, die erste Inbetriebnahme eines FAULHABER-Antriebs mit der Software „FAULHABER Motion Manager“ durchzuführen.

Der FAULHABER Motion Manager ermöglicht einen einfachen Zugriff auf die Einstellungen und Parameter der angeschlossenen Motorsteuerungen. Über die grafische Benutzeroberfläche können Konfigurationen ausgelesen, verändert und wieder eingespielt werden. Einzelne Befehle oder komplette Parametersätze und Programmsequenzen können eingegeben und zur Steuerung übertragen werden.

Assistenzfunktionen unterstützen den Bediener bei der Inbetriebnahme von Antriebssteuerungen. Die Assistenzfunktionen sind auf der Benutzeroberfläche so angeordnet wie sie im Normalfall verwendet werden:

- Verbindungsassistent: Unterstützt den Bediener beim Einrichten der Verbindung zur angeschlossenen Steuerung
- Motorassistent: Unterstützt den Bediener beim Anpassen einer externen Steuerung an den angeschlossenen Motor durch Auswahl des jeweiligen FAULHABER Motors
- Reglereinstellungsassistent: Unterstützt den Bediener bei der Optimierung der Reglerparameter.

Die Software kann kostenlos von der FAULHABER Internet-Seite heruntergeladen werden: https://www.faulhaber.com/motionmanager.

Es wird empfohlen, immer die neueste Version des FAULHABER Motion Managers zu verwenden.

Der FAULHABER Motion Manager ist im separaten Handbuch „Motion Manager 6“ beschrieben. Der Inhalt des Handbuchs steht zusätzlich als kontext-sensitive Online-Hilfe des FAULHABER Motion Managers zur Verfügung.

2.4 Parameter speichern und wiederherstellen

Damit geänderte Parameter im OV auch nach erneutem Einschalten des Controllers erhalten bleiben, müssen sie mit dem Save-Befehl dauerhaft in den nicht-flüchtigen Speicher (Anwendungs-EEPROM) gespeichert werden (siehe Kap. 5.1, S. 41). Beim Einschalten des Motors werden die Parameter automatisch aus dem nicht-flüchtigen Speicher in den flüchtigen Speicher (RAM) geladen.

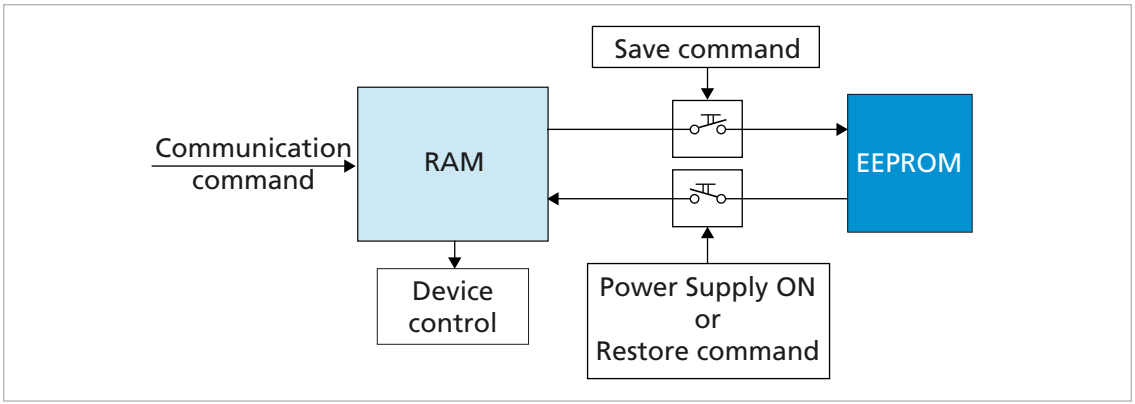

Abb. 3: Parameter speichern und wiederherstellen

Folgende Parameter können mit dem Restore-Befehl geladen werden (siehe Kap. 5.1, S. 41):

- Werkseinstellungen
- Mit dem Save-Befehl gespeicherte Parameter

2.4.1 Parameter speichern

Die aktuellen Parametereinstellungen können komplett oder für einzelne Bereiche im internen EEPROM gespeichert werden (SAVE) (siehe Tab. 17).

- Auf Subindex 01 bis 05 des Objekts 0x1010 die Signatur "save" schreiben (siehe Tab. 18).

2.4.2 Einstellungen wiederherstellen

Abgespeicherte Parameter werden beim Einschalten des Antriebs automatisch geladen.

Werkseinstellungen oder zuletzt gespeicherte Parametereinstellungen können jederzeit komplett oder für einzelne Bereiche aus dem internen EEPROM geladen werden (RESTORE) (siehe Tab. 19).

1. Auf Subindex 01 bis 06 des Objekts 0x1011 die Signatur "load" schreiben (siehe Tab. 20).
   - Nach Restore Factory (01), Restore Communication (02) und Restore Application (03) muss der Antrieb zurückgesetzt werden. Nur dann werden die Parameter aktualisiert.
2. Applikationsparameter (04) sowie Satz 1 und Satz 2 der speziellen Applikationsparameter (05/06) mit dem Reload-Befehl aktualisieren.
   - Der Reload-Befehl überschreibt die zuletzt als Anwenderparameter gespeicherten Werte.

Sollen die aktuell geladenen Werte auch nach einem Restore zur Verfügung stehen, müssen diese mit einem geeigneten Programm (z. B. FAULHABER Motion Manager) auf dem PC gesichert werden.

2.4.3 Parametersatz wechseln

Die Ablage der Applikationsparameter (Motordaten, I/O-Konfiguration, Reglerparameter, Betriebsart etc.) umfasst einen gemeinsamen Basissatz von Parametern (App) und daneben einen Speicherbereich für Parameter, die häufig an unterschiedliche Lastsituationen angepasst werden müssen (App1/App2):

Drehzahlregler und Filter

Index	Subindex	Name	Typ	Attr.	Bedeutung
0x2344	0x01	Gain K_P	U32	rw	Reglerverstärkung [As $1e^{-6}$]
	0x02	Integral Time TN	U16	rw	Reglernachstellzeit [100 µs]
0x2346	0x01	Set Point Velocity Filter Time T_F	U16	rw	Filterzeit T_F [100 µs]
	0x02	Setpoint Filter Enable	U8	rw	0: inactive 1: active
0x2347	0x01	Gain Factor	U8	rw	Gain Faktor (wird bei der Geschwindigkeitsregelung im PP Mode auf K_P angewendet) 0: Verstärkung des Geschwindigkeitsreglers wird im Ziel auf 0 reduziert 128: keine Variable Verstärkung 255: Verstärkung des Geschwindigkeitsreglers wird im Ziel verdoppelt

Positionsregler

Index	Subindex	Name	Typ	Attr.	Bedeutung
0x2348	0x00	Number of Entries	U8	ro	Anzahl Objekteinträge
	0x01	K_V [1/s]	U8	rw	Range: 1-250

Vorsteuerungen

Index	Subindex	Name	Typ	Attr.	Bedeutung
0x2349	0x01	Torque/Force FeedForward Factor	U8	rw	Faktor der Drehmoment- bzw. Kraftvorsteuerung 0: 0% Aufschaltung des Vorsteuerwerts 128: 100% Vorsteuerung
	0x02	Torque/Force FeedForward Delay	U8	rw	Sollwertverzögerung: 0: unverzögerte Aufschaltung 1: Aufschaltung um eine Abtastung verzögert
0x234A	0x01	Velocity Feedforward Factor	U8	rw	Faktor der Drehmoment- bzw. Kraftvorsteuerung 0: 0% Vorsteuerung 128: 100% Vorsteuerung
	0x02	Velocity FeedForward Delay	U8	rw	Sollwertverzögerung: 0: unverzögerte Aufschaltung 1: Aufschaltung um eine Abtastung verzögert

Allgemeine Einstellungen

Index	Subindex	Name	Typ	Attr.	Bedeutung
0x6060	0x00	Modes of Operation	S8	rw	Auswahl der Betriebsart –4: ATC –3: AVC –2: APC –1: Volt Mode 0: Regler nicht aktiviert 1: PP 3: PV 6: Homing 8: CSP 9: CSV 10: CST
0x6081	0x00	Profile Velocity	U32	rw	Profile Velocity [in benutzerdefinierten Einheiten]
0x6083	0x00	Profile Acceleration	U32	rw	Profile Acceleration [1/s^2]
0x6084	0x00	Profile Deceleration	U32	rw	Profile Deceleration [1/s^2]
0x6086	0x00	Motion Profile Type	S16	rw	Bewegungsprofiltyp: 0: Lineares Profil 1: Sin^2 Geschwindigkeit
0x60E0	0x00	Positive Torque Limit Value	U16	rw	Betrag des oberen Begrenzungswerts [in bezogener Darstellung]
0x60E1	0x00	Negative Torque Limit Value	U16	rw	Betrag des unteren Begrenzungswerts [in bezogener Darstellung]

Diese Parameter sind doppelt abgelegt. Im Betrieb kann schnell zwischen diesen unterschiedlichen Voreinstellungen gewechselt werden.

Einen Anwendungssatz anlegen

- Save Application Parameters 1: Auf Subindex 04 des Objekts 0x1010 die Signatur "save" schreiben.
  - ↳ Aktuelle Daten sind als Anwendungsparametersatz 1 gespeichert.
- Save Application Parameters 2: Auf Subindex 05 des Objekts 0x1010 die Signatur "save" schreiben.
  - ↳ Aktuelle Daten sind als Datensatz Anwendungsparametersatz 2 gespeichert.

Einen Anwendungssatz aktivieren

- Reload Application Parameters 1: Auf Subindex 05 des Objekts 0x1011 die Signatur "load" schreiben.
  - ↳ Aktuelle Daten aus dem Anwendungsparametersatz 1 werden direkt aktiviert.
- Reload Application Parameters 2: Auf Subindex 06 des Objekts 0x1011 die Signatur "load" schreiben.
  - ↳ Aktuelle Daten aus dem Anwendungsparametersatz 2 werden direkt aktiviert.

3 CANopen-Protokollbeschreibung

3.1 Einführung

CANopen
CANopen ist ein Standard-Softwareprotokoll. Für die Kommunikation mit CANopen wird eine CAN-Hardwareumgebung benötigt. Innerhalb eines CANopen-Netzwerks können bis zu 127 Knoten adressiert werden. Die maximale Übertragungsgeschwindigkeit beträgt 1 MBit/s.

CAN-Normung
Die CiA definiert in der CiA 301 folgende Aspekte:

- Kommunikationsstruktur
- Steuer- und Überwachungsfunktionen

Für eine Reihe von Geräteklassen wurden CANopen-Geräteprofile definiert, wie zum Beispiel:

- CiA 402 für Antriebe
- CiA 401 für Ein- und Ausgabegeräte

Aufbau eines CANopen-Telegramms
Ein CANopen-Telegramm besitzt einen 11-Bit-Identifier und kann bis zu 8 Byte Nutzdaten beinhalten.

Tab. 1: Schematischer Aufbau eines CANopen-Telegramms

11-Bit-Identifier	bis zu 8 Byte Nutzdaten							
11 Bit	8 Bit	8 Bit	8 Bit	8 Bit	8 Bit	8 Bit	8 Bit	8 Bit

3.2 Kommunikationsdienste

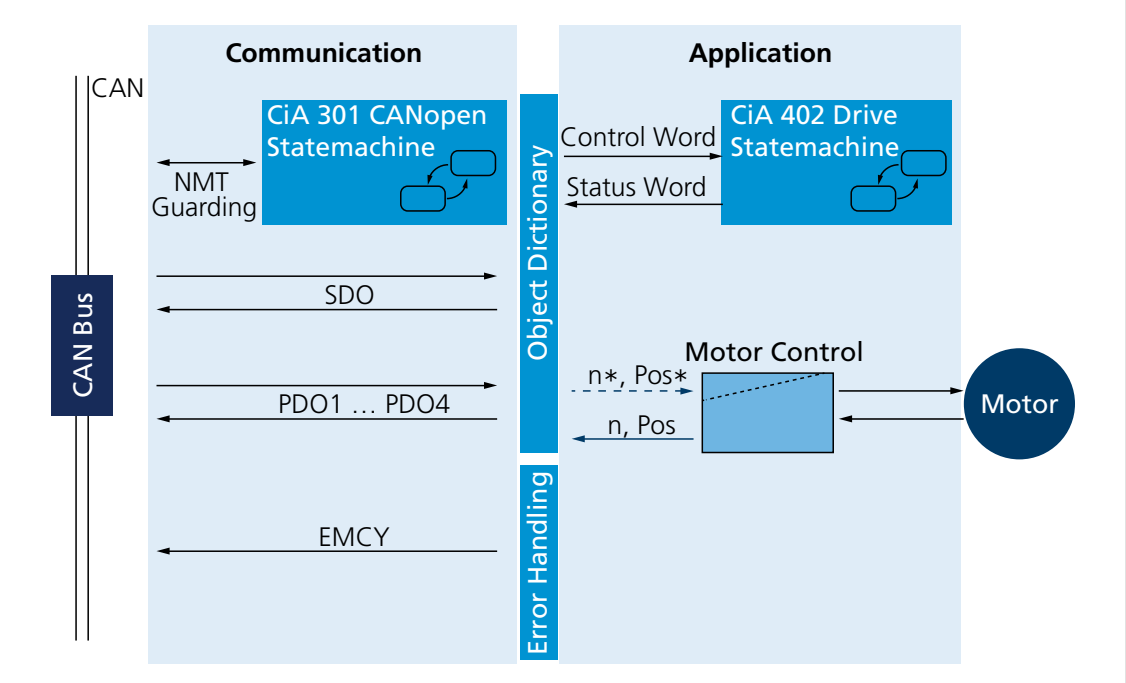

Abb. 4: Kommunikationsdienste des Motion Controllers

Der Kommunikationsteil enthält Kommunikationsdienste entsprechend CiA 301.

Tab. 2: Kommunikationsdienste nach CiA 301

Kommunikationsdienste	Beschreibung
NMT (Netzwerkmanagement)	Aktiviert Knoten und überwacht den aktuellen Zustand eines Knotens (siehe Kap. 3.8, S. 29).
SDO (Servicedatenobjekt)	Über das SDO greift der CANopen-Master auf Parameter in einem Knoten zu. Bei jedem SDO-Zugriff wird genau ein Parameter gelesen oder beschrieben. Ein SDO kann stets nur einen Knoten im Netzwerk ansprechen (siehe Kap. 3.5, S. 23).
PDO (Prozessdatenobjekt)	Über das PDO wird auf Echtzeitdaten zugegriffen. Mit einem PDO kann über eine CAN-Nachricht auf mehrere Antriebsparameter gleichzeitig zugegriffen werden. Die in einem PDO versendeten oder empfangenen Parameter können frei konfiguriert werden (siehe Kap. 3.4, S. 19).
SYNC-Objekt	Über SYNC-Objekte werden verschiedene Anwendungen am CAN-BUS synchronisiert (siehe Kap. 3.7, S. 28).
EMCY (Emergency-Objekt)	Eine Emergency-Nachricht informiert den CANopen-Master über Fehler. Eine CAN-Nachricht übermittelt asynchron den Fehlercode, sodass der Zustand des CANopen-Slaves nicht laufend nach Fehlern abgefragt werden muss (siehe Kap. 3.6, S. 26).

Kommunikationsprofil
Der FAULHABER Motion Controller unterstützt das CANopen-Kommunikationsprofil gemäß CiA 301 V4:

- 4 Sende-PDOs
- 4 Empfangs-PDOs
- 1 Server-SDO
- Emergency-Object
- NMT mit Node-Guarding und Heartbeat
- SYNC-Object

Die Datenbelegung der PDOs ist entsprechend dem “PDO set for servo drive” nach CiA 402 V3 voreingestellt und kann vom Anwender geändert werden (dynamisches PDO-Mapping).

3.3 Identifier-Verteilung

Der Communication Object Identifier (COB-ID) setzt sich aus einer 7-Bit Knotenadresse (Node ID) und einem 4-Bit Funktionscode zusammen.

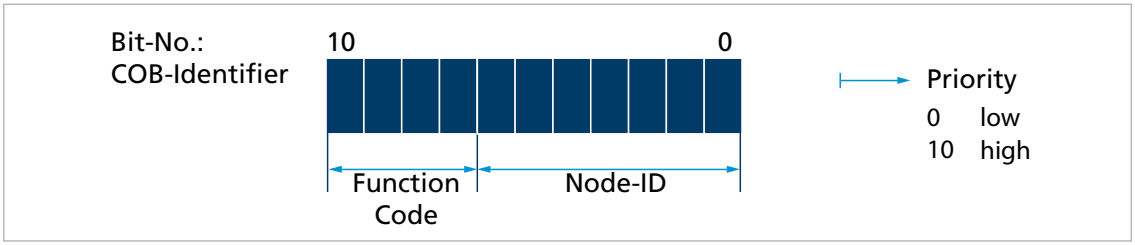

Abb. 5: Identifier-Verteilung

Das Predefined-Connection-Set definiert Standard-Identifier für die wichtigsten Objekte.

Tab. 3: Standard-Identifier

Objekt	Funktionscode (binär)	Resultierende COB-ID	Objektindex für Kommunikationseinstellung
NMT	0000	0	–
SYNC	0001	128 (80h)	1005h
EMERGENCY	0001	129 (81h) bis 255 (FFh)	1014h
PDO1 (tx)	0011	385 (181h) bis 511 (1FFh)	1800h
PDO1 (rx)	0100	513 (201h) bis 639 (27Fh)	1400h
PDO2 (tx)	0101	641 (281h) bis 767 (2FFh)	1801h
PDO2 (rx)	0110	769 (301h) bis 895 (37Fh)	1401h
PDO3 (tx)	0111	897 (381h) bis 1023 (3FFh)	1802h
PDO3 (rx)	1000	1025 (401h) bis 1151 (47Fh)	1402h
PDO4 (tx)	1001	1153 (481h) bis 1279 (4FFh)	1803h
PDO4 (rx)	1010	1281 (501h) bis 1407 (57Fh)	1403h
SDO (tx)	1011	1409 (581h) bis 1535 (5FFh)	1200h
SDO (rx)	1100	1537 (601h) bis 1663 (67Fh)	1200h
NMT Error Control	1110	1793 (701h) bis 1919 (77Fh)	–

Die COB-IDs der PDOs, des SYNC-Objekts und des Emergency-Objekts können über die Kommunikationsparameter im Objektverzeichnis geändert werden. Die COB-ID der SDO-Telegramme kann nicht verändert werden und ergibt sich immer aus der Knotennummer.

Im Auslieferungszustand ist die Knotennummer 1 konfiguriert. Die COB-IDs sind entsprechend voreingestellt:

- RxPDO: 201h, 301h, 401h und 501h
- TxPDO: 181h, 281h, 381h und 481h
- EMCY: 81h
- RxSDO: 581h
- TxSDO: 601h

Bei einem Wechsel von der Knotennummer 255 (unkonfigurierter CANopen-Knoten) nach einer Knotennummer >127 über das LSS-Protokoll werden die von der Knotennummer abhängigen COB-IDs automatisch angepasst (siehe Kap. 4.1.3, S. 39).

3.4 PDO (Prozessdatenobjekt)

PDOs sind CAN-Nachrichten mit bis zu 8 Byte Nutzdaten. PDOs enthalten Prozessdaten zur Steuerung und Überwachung des Geräteverhaltens. Aus Sicht des Antriebs werden PDOs in Empfangs- und Sende-PDOs unterschieden.

- Empfangs-PDO (RxPDO): wird von einem Antrieb empfangen und enthält z. B. Steuerdaten
- Sende-PDO (TxPDO): wird von einem Antrieb gesendet und enthält z. B. Überwachungsdaten

PDOs werden nur ausgewertet oder übertragen, wenn sich das Gerät im NMT-Zustand *Operational* befindet (siehe Kap. 3.8, S. 29).

Die Übertragung von PDOs kann auf unterschiedliche Arten angestoßen werden. Das Verhalten kann für jedes PDO über den Parameter Transmission Type der Kommunikationsparameter im Objektverzeichnis eingestellt werden:

Tab. 4: PDO-Übertragungsarten

Transmission Type	Beschreibung
Ereignisgesteuert	Ereignisgesteuerte RxPDOs werden direkt beim Eintreffen verarbeitet. Ereignisgesteuerte TxPDOs werden versandt, wenn das Statusword des Geräts enthalten ist und sich geändert hat.
Remote Request (RTR)	Daten werden nach einer Anforderungsmeldung gesendet.
Synchronisiert	Daten werden nach Eintreffen eines SYNC-Objekts gesendet (siehe Kap. 3.7, S. 28).

3.4.1 PDO-Konfiguration

- Maximal 4 Parameter können in einem PDO gemappt werden.
- Über die Objekte 0x1600 bis 0x1603 und 0x1A00 bis 0x1A03 kann die Datenbelegung der PDOs geändert werden. Die dafür notwendige Mapping-Prozedur ist in der CiA 301 beschrieben. Zur Durchführung der Mapping-Prozedur ist ein geeignetes Tool notwendig (z. B. FAULHABER Motion Manager oder Konfigurationswerkzeug der verwendeten SPS-Steuerung).
- Über die Objekte 0x1400 bis 0x1403 bzw. 0x1800 bis 0x1803 können Transmission Type und COB-ID der PDOs geändert werden.
- Über den Parameter Transmission Type kann das Verhalten eines PDO eingestellt werden:

Tab. 5: Transmission-Types eines PDOs

Transmission Type	Bedeutung
0	synchron, azyklisch PDO wird einmal nach einem SYNC-Objekt gesendet bzw. ausgeführt, wenn sich der Inhalt des PDOs geändert hat (siehe Kap. 3.7, S. 28).
1 bis 240	synchron, zyklisch PDO wird nach jedem SYNC-Objekt gesendet (siehe Kap. 3.7, S. 28). Der Wert ist dabei gleich der Anzahl der SYNC-Objekte, welche bis zum erneuten Senden der PDO empfangen worden sein müssen (1 = PDO wird mit jedem SYNC-Objekt gesendet)
252	Nur bei TxPDOs: asynchron ▪ Auf ein SYNC-Signal wird der Inhalt des TxPDO gespeichert ▪ Auf Anforderung (RTR) wird das TxPDO an den Master versandt
253	Nur bei TxPDOs: asynchron Auf Anforderung (RTR) wird das TxPDO an den Master versandt
255	asynchron (ereignisgesteuert)

3.4.2 PDO-Mapping in der Standardkonfiguration (Auslieferungszustand)

RxPDO1: Controlword

11-Bit-Identifier	2 Byte Nutzdaten	
0x200 (512d) + Node ID	LB	HB

Das RxPDO1 enthält das 16-Bit-Controlword nach CiA DSP402. Das Controlword steuert die Statemachine der Antriebseinheit und verweist auf den Objektindex 0x6040 im Objektverzeichnis. Die Bitaufteilung ist in der Dokumentation der Antriebsfunktionen beschrieben.

TxPDO1: Statusword

11-Bit-Identifier	2 Byte Nutzdaten	
0x180 (384d) + Node ID	LB	HB

Das TxPDO1 enthält das 16-Bit-Statusword nach CiA 402. Das Statusword zeigt den Zustand der Antriebseinheit an und verweist auf den Objektindex 0x6041 im Objektverzeichnis. Die Bitaufteilung ist in der Dokumentation der Antriebsfunktionen beschrieben.

RxPDO2: Controlword, Target Position (PP)

11-Bit-Identifier	6 Byte Nutzdaten					
0x300 (768d) + Node ID	LB	HB	LLB	LHB	HLB	HHB

Das RxPDO2 enthält das 16-Bit-Controlword und den 32-Bit-Wert der Zielposition (Objekt 0x607A) für den Profile Position Mode (PP)

TxPDO2: Statusword, Position Actual Value

11-Bit-Identifier	6 Byte Nutzdaten					
0x280 (640d) + Node ID	LB	HB	LLB	LHB	HLB	HHB

Das TxPDO2 enthält das 16-Bit-Statusword und den 32-Bit-Wert der Istposition (Objekt 0x6064).

RxPDO3: Controlword, Target Velocity (PV)

11-Bit-Identifier	6 Byte Nutzdaten					
0x400 (1024d) + Node ID	LB	HB	LLB	LHB	HLB	HHB

Das RxPDO3 enthält das 16-Bit-Controlword und den 32-Bit-Wert der Solldrehzahl (Objekt 0x60FF) für den Profile Velocity Mode (PV).

TxPDO3: Statusword, Velocity Actual Value

11-Bit-Identifier	6 Byte Nutzdaten					
0x380 (896d) + Node ID	LB	HB	LLB	LHB	HLB	HHB

Das TxPDO3 enthält das 16-Bit-Statusword und den 32-Bit-Wert der Istdrehzahl (Objekt 0x606C).

RxPDO4: Controlword, Target Torque

11-Bit-Identifier	4 Byte Nutzdaten					
0x400 (1024d) + Node ID	LB	HB	LLB	LHB	HLB	HHB

Das RxPDO4 enthält das 16-Bit-Controlword und den 16-Bit Wert des Soll-Drehmoments (Objekt 0x6071) für den Cyclic Torque Modus (CST).

TxPDO4: Statusword, Torque Actual Value

11-Bit-Identifier	4 Byte Nutzdaten					
0x380 (896d) + Node ID	LB	HB	LLB	LHB	HLB	HHB

Das TxPDO4 enthält das 16-Bit-Statusword und den 16-Bit Wert des Ist-Drehmoments (Objekt 0x6077) für den Cyclic Torque Modus (CST)

3.4.3 Behandlung von Mapping-Fehlern

Wenn die von CiA 301 vorgegebene Mapping-Prozedur nicht eingehalten wird, wird einer der folgenden SDO-Fehler zurückgegeben:

Tab. 6: SDO-Fehler bei fehlerhafter Mapping-Prozedur

SDO-Fehler	Bedeutung	Ursache
0x06090030	Allgemeiner Wertebereichfehler	Der Mapping-Parameter ist nicht in der vorgegebenen Mapping-Prozedur beschrieben.
0x06020000	Objekt nicht im Objektverzeichnis vorhanden	Der Wert für die Anzahl gemappter Objekte ist größer als die Anzahl der gültigen Einträge in den jeweiligen Subindexen der Mapping-Parameter-Objekte.

Ist die Anzahl gemappter Objekte 0, wird das PDO intern als ungültig markiert und nicht bedient.

Weitere Mapping-Fehler sind in der SDO-Fehlertabelle beschrieben (siehe Kap. 3.5.2, S. 25).

3.4.4 Dummy Mappings

RxPDOs können so konfiguriert werden, dass mehr als ein Teilnehmer darauf reagiert. In diesem Fall kann es gewünscht sein, nur einen Teil der im PDO enthaltenen Daten in einem der Geräte auszuwerten.

Für lokal nicht genutzte Daten kann ein Dummy Mapping auf einen der unterstützten Datentypen in die Mappingtabelle des PDOs eingetragen werden:

Index	Typ
0x0002	S8
0x0003	S16
0x0004	S32
0x0005	U8
0x0006	U16
0x0007	U32

Beispiel
In einem RxPDO sind die Sollpositionen für zwei Achsen enthalten.

Mapping für den Knoten, der auf die erste Sollposition reagieren soll:

- 0x160x.00 = 2
- 0x160x.01 = 0x607A0020
- 0x160x.02 = 0x00040020

Mapping für den Knoten, der auf die zweite Sollposition reagieren soll:

- 0x160x.00 = 2
- 0x160x.01 = 0x00040020
- 0x160x.02 = 0x607A0020

3.5 SDO (Servicedatenobjekt)

Das SDO liest und beschreibt Parameter im OV (Objektverzeichnis). Über den 16-Bit-Index und den 8-Bit-Subindex greift das SDO auf das Objektverzeichnis zu. Der Motion Controller stellt auf Anforderung des Clients (PC,SPS (Speicherprogrammierbare Steuerung)) Daten zur Verfügung (Upload) bzw. empfängt Daten vom Client (Download).

Tab. 7: Allgemeine Strukturierung der SDO-Nutzdaten

Byte0	Byte 1 bis 2	Byte 3	Byte 4 bis 7
Command-Specifier	16-Bit-Index	8-Bit-Subindex	4 Byte-Parameter-Data

Tab. 8: Einteilung der SDO-Übertragungsarten

Übertragungsart	Byteanzahl	Verwendungszweck
Expedited Transfer	maximal 4 Byte	Lesen und Beschreiben einzelner numerischer Parameter
Segmented Transfer	mehr als 4 Byte	Lesen von Text-Parametern (z. B. Gerätename, Firmware-Version) und Übertragung von Datenblöcken (z. B. Trace-Puffer)

In diesem Dokument ist nur der Expedited Transfer beschrieben. Der Segmented-Transfer ist in der CiA 301 beschrieben.

3.5.1 Expedited Transfer

SDO-Nachrichten sind immer 8 Byte groß.

Lesen von OV-Einträgen (Client-to-Server, Upload-Request)

11-Bit-Identifier	8 Byte Nutzdaten							
0x600 (1536d) + Node ID	0x40	Index LB	Index HB	Subindex	0	0	0	0

Server-to-Client, Upload-Response

11-Bit-Identifier	8 Byte Nutzdaten							
0x580 (1408d) + Node ID	CS(0x4x)	Index LB	Index HB	Subindex	LLB (D0)	LHB (D1)	HLB (D2)	HHB (D3)

Der Command-Specifier CS(0x4x) gibt die Anzahl der gültigen Datenbytes in D0 bis D3 und die Transferkennung an. Der Command-Specifier ist wie folgt codiert:

- CS = 0x4F, 1 Datenbyte in D0
- CS = 0x4B, 2 Datenbytes in D0 bis D1
- CS = 0x47, 3 Datenbytes in D0 bis D2
- CS = 0x43, 4 Datenbytes in D0 bis D3

Schreiben von OV-Einträgen (Client-to-Server, Download-Request)

11-Bit-Identifier	8 Byte Nutzdaten							
0x600 (1536d) + Node ID	CS(0x2x)	Index LB	Index HB	Subindex	LLB (D0)	LHB (D1)	HLB (D2)	HHB (D3)

Der Command-Specifier CS(0x2x) gibt die Anzahl der gültigen Datenbytes in D0 bis D3 und die Transferkennung an. Der Command-Specifier ist wie folgt codiert:

- CS = 0x2F, 1 Datenbyte in D0
- CS = 0x2B, 2 Datenbytes in D0 bis D1
- CS = 0x27, 3 Datenbytes in D0 bis D2
- CS = 0x23, 4 Datenbytes in D0 bis D3
- CS = 0x22, keine Angabe der Anzahl Datenbytes

Server-to-Client, Download-Response

11-Bit-Identifier	8 Byte Nutzdaten							
0x580 (1407d) + Node ID	0x60	Index LB	Index HB	Subindex	0	0	0	0

Abbruch bei SDO-Fehlern

SDO-Abort Client-to-Server

11-Bit-Identifier	8 Byte Nutzdaten							
0x600 (1536d) + Node ID	0x80	Index LB	Index HB	Subindex	ERROR 0	ERROR 1	ERROR 2	ERROR 3

SDO-Abort Server-to-Client

11-Bit-Identifier	8 Byte Nutzdaten							
0x580 (1536d) + Node ID	0x80	Index LB	Index HB	Subindex	ERROR 0	ERROR 1	ERROR 2	ERROR 3

3.5.2 SDO-Fehlerbeschreibung

Kann das SDO-Protokoll auf einer Seite nicht weiter verarbeitet werden, wird ein SDO-Abort-Telegramm versendet (siehe Kap. 3.5.1, S. 23). Die Fehlerarten sind wie folgt codiert:

- Error0: Zusätzlicher Fehlercode HB
- Error1: Zusätzlicher Fehlercode LB
- Error2: Fehlercode
- Error3: Fehlerklasse

Fehler-klasse	Fehler-code	Zusatz-code	Beschreibung
0x05	0x03	0x0000	Toggle-Bit nicht geändert
0x05	0x04	0x0001	SDO-Command-Specifier ungültig oder unbekannt
0x06	0x01	0x0000	Zugriff auf dieses Objekt wird nicht unterstützt
0x06	0x01	0x0001	Versuch, einen Write-Only-Parameter zu lesen
0x06	0x01	0x0002	Versuch, auf einen Read-Only-Parameter zu schreiben
0x06	0x02	0x0000	Objekt nicht im Objektverzeichnis vorhanden
0x06	0x04	0x0041	Objekt kann nicht in PDO gemappt werden
0x06	0x04	0x0042	Anzahl und/oder Länge der gemappten Objekte würde PDO-Länge überschreiten
0x06	0x04	0x0043	Allgemeine Parameter-Inkompatibilität
0x06	0x04	0x0047	Allgemeiner interner Inkompatibilitätfehler im Gerät
0x06	0x07	0x0010	Datentyp oder Parameterlänge stimmen nicht überein oder sind unbekannt
0x06	0x07	0x0012	Datentypen stimmen nicht überein, Parameterlänge zu groß
0x06	0x07	0x0013	Datentypen stimmen nicht überein, Parameterlänge zu klein
0x06	0x09	0x0011	Subindex nicht vorhanden
0x06	0x09	0x0030	Allgemeiner Wertebereichfehler
0x06	0x09	0x0031	Wertebereichfehler: Parameterwert zu groß
0x06	0x09	0x0032	Wertebereichfehler: Parameterwert zu klein
0x06	0x09	0x0036	Wertebereichfehler: Maximumwert kleiner als Minimumwert
0x08	0x00	0x0000	Allgemeiner SDO-Fehler
0x08	0x00	0x0020	Zugriff nicht möglich
0x08	0x00	0x0022	Zugriff bei aktuellem Gerätestatus nicht möglich

3.6 Emergency-Objekt (Fehlermeldung)

Das Emergency-Objekt informiert asynchron andere Busteilnehmer über Fehler und muss nicht abgefragt werden. Das Emergency-Object ist immer 8 Byte groß:

11-Bit-Identifier	8 Byte Nutzdaten							
0x80 (128d) + Node ID	Error0(LB)	Error1(HB)	Error-Reg	FE0 (LB)	FE1 (HB)	0	0	0

Belegung der Nutzdaten:

- Error0(LB)/Error1(HB): 16-Bit-Error-Code
- Error-Reg: Error-Register (Inhalt von Objekt 0x1001, siehe Kap. 5.1, S. 41)
- FE0(LB)/FE1(HB): 16-Bit FAULHABER Fehlerregister (Inhalt von Objekt 0x2320, siehe Tab. 12)
- Bytes 5 bis 7: unbenutzt (0)

Das Error Register kennzeichnet die Fehlerart. Die einzelnen Fehlerarten sind bitcodiert und den jeweiligen Error Codes zugeordnet. Über das Objekt 0x1001 kann der letzte Wert des Error Registers abgefragt werden.

Tab. 9 listet alle Fehler auf, die über Emergency-Nachrichten gemeldet werden, sofern der entsprechende Fehler in der Emergency-Mask für das FAULHABER Fehlerregister gesetzt ist (Tab. 13).

Tab. 9: Emergency-Error-Codes

Emergency-Nachricht		FAULHABER-Fehlerregister 0x2320			Error Register 0x1001	
Error Code	Bezeichnung	Error Mask 0x2321	Bit	Bezeichnung	Bit	Bezeichnung
0x0000	No error (wird verschickt, wenn ein Fehler nicht mehr vorliegt bzw. bestätigt wurde)	–	–	–	–	–
–	–	–	–		0	Generic error (wird gesetzt, wenn eines der Fehlerbits 1 bis 7 gesetzt wird)
0x3210	Overvoltage	0x0004	2	OverVoltageError	2	Spannungsfehler
0x3220	Undervoltage	0x0008	3	UnderVoltageError	2	Spannungsfehler
0x43F0	Temperature Warning	0x0010	4	TempWarning	1	Strom-Fehler[a)]
0x4310	Temperature Error	0x0020	5	TempError	3	Temperatur-Fehler
0x5410	Output Stages	0x0080	7	IntHWError	7	Herstellerspezifischer Fehler
0x5530	EEPROM Fault	0x0400	10	MemError	–	–
0x6100	Software Error	0x1000	12	CalcError	7	Herstellerspezifischer Fehler

Emergency-Nachricht		FAULHABER-Fehlerregister 0x2320			Error Register 0x1001	
Error Code	Bezeichnung	Error Mask 0x2321	Bit	Bezeichnung	Bit	Bezeichnung
0x7200	Measurement Circuit: Current Measurement	0x0200	9	CurrentMeasError	7	Herstellerspezifischer Fehler
0x7300	Sensor Fault (Encoder)	0x0040	6	EncoderError	7	Herstellerspezifischer Fehler
0x7400	Computation Circuit: Module Fault	0x0100	8	ModuleError	7	Herstellerspezifischer Fehler
0x8110 0x8130 0x8140 0x8310	CAN Overrun CAN Guarding Failed CAN Recovered From Bus-Off RS232 overrun	0x0800	11	ComError	4	Kommunikations-Fehler
0x84F0	Deviation Error (Velocity Controller)	0x0001	0	SpeedDeviationError	5	Antriebsspezifischer Fehler
0x84FF	Max Speed Error	0x2000	13	DynamicError	7	Herstellerspezifischer Fehler
0x8611	FollowingError (Position Controller)	0x0002	1	FollowingError	5	Antriebsspezifischer Fehler

a) *Der Stromregler hält den Motorstrom immer unter der eingestellten Grenze. Das Überstromfehler-Bit wird bei Überschreiten der Warnungstemperatur gesetzt und der zulässige Motorstrom wird vom Spitzenstrom-Wert auf den Dauerstrom-Wert reduziert.*

Beispiel:
Eine Emergency-Nachricht mit der Nutzdatenbelegung in Tab. 10 wird in folgendem Fall versendet:

- In der Error Mask 0x2321 ist unter Subindex 1 (Emergency Mask) Bit 1 (Schleppfehler) gesetzt.
- Der in Objekt 0x6065.00 eingestellte Korridor für die Regelabweichung des Positionsreglers wurde für einen längeren Zeitraum überschritten, als der in Objekt 0x6066.00 eingestellte Wert für die Fehlerverzögerungszeit (siehe Dokumentation der Antriebsfunktionen).

Tab. 10: Beispielhafte Nutzdatenbelegung einer Emergency-Nachricht

8 Byte Nutzdaten							
0x11	0x86	0x20	0x02	0x00	0x00	0x00	0x00

3.7 SYNC-Object

Das SYNC-Objekt ist eine Nachricht ohne Nutzdaten. Das SYNC-Objekt wird zum Triggern synchroner PDOs und zum gleichzeitigen Starten von Prozessen auf verschiedenen Geräten verwendet.

Der Identifier des SYNC-Objekts wird im Objektverzeichnis unter Index 0x1005 eingestellt (standardmäßig 0x80).

11-Bit-Identifier	0 Byte Nutzdaten
0x80	keine Nutzdaten

Damit ein SYNC-Objekt ein PDO triggert, muss der Transmission Type des zu triggernden PDOs entsprechend eingestellt werden (siehe Tab. 5).

3.7.1 Triggern Synchroner PDOs

Synchrone RxPDO: Der mit dem PDO übertragene Befehl wird erst nach Erhalt eines SYNC Objekts ausgeführt. Der Transmission Type 1 bis 240 eines RxPDO ist identisch mit dem Transmission Type 0.

Synchrone TxPDO: Sofort nach Erhalt eines SYNC-Objekts werden die synchronen TxPDOs mit den aktuellen Daten verschickt.

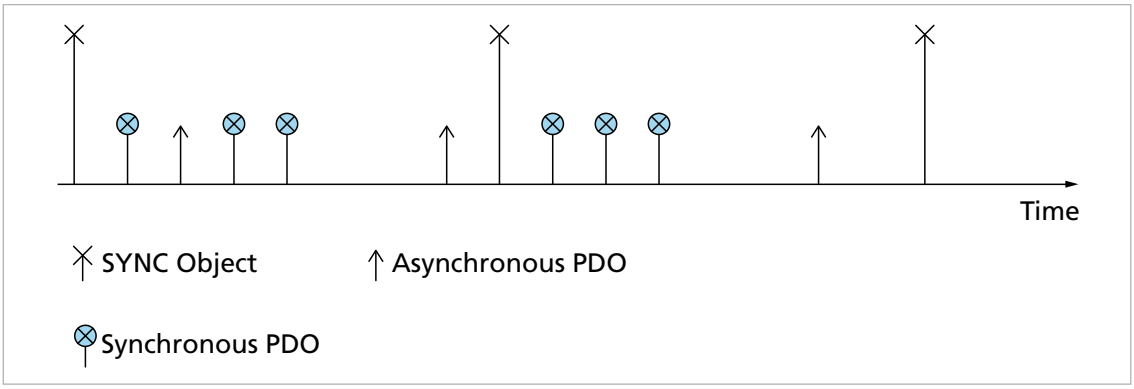

Abb. 6: Schaubild TxPDO mit SYNC

Mit den Übertragungsarten 1-240 können Knoten auch gruppiert werden.

3.8 NMT (Netzwerkmanagement)

Das Netzwerkmanagementobjekt steuert die CiA 301 Statemachine des CANopen-Geräts und überwacht Netzwerk-Knoten.

Nach dem Einschalten und der Initialisierung befindet sich der Motion Controller automatisch im Zustand *Pre-Operational*. Im Zustand *Pre-Operational* kann nur mit NMT-Nachrichten und über SDOs mit dem Gerät kommuniziert werden.

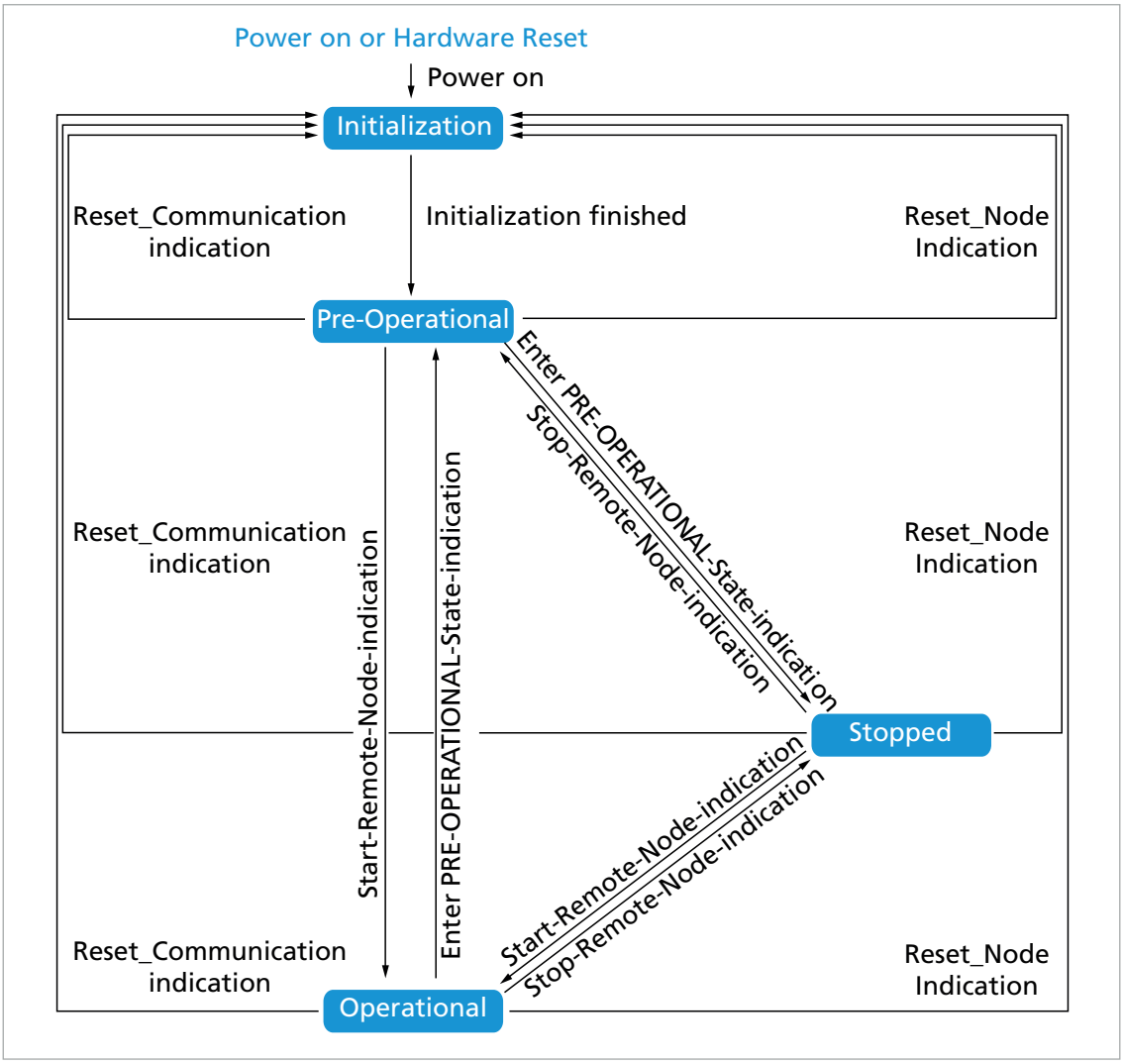

Abb. 7: CiA 301 Statemachine

Tab. 11: NMT-Zustandsänderungen

Statusübergang	CS	Bedeutung
Power on	–	Der Initialisierungs-Status wird beim Einschalten selbsttätig erreicht.
Initialization finished	–	Nach der Initialisierung befindet sich das Gerät automatisch im Pre-Operational-Status und eine Boot-Up-Nachricht wird abgeschickt.
Start Remote-Node indication	0x01 (1d)	Startet das Gerät und gibt die Übertragung von PDOs frei.
Enter Pre-Operational-State indication	0x80 (128d)	Stoppt die PDO-Übertragung, SDO sind weiter aktiv.

Statusübergang	CS	Bedeutung
Stop Remote-Node indication	0x02 (2d)	Antrieb geht in den Stopped-Zustand, SDO und PDO sind abgeschaltet.
Reset Node indication	0x81 (129d)	Führt einen Reset durch. Alle Objekte werden auf Power-On-Standards zurückgesetzt.
Reset Communication indication	0x82 (130d)	Führt einen Reset der Kommunikationsfunktionen durch.

Die FAULHABER Motion Controller sind mit Standardkonfiguration für alle Objekte ausgestattet. Nach abgeschlossener Inbetriebnahme können die anwendungsspezifischen Einstellungen direkt im Gerät gespeichert werden. Im Regelfall ist daher keine weitere Parametrierung beim Systemstart notwendig.

Starten eines CANopen Knotens
Start Remote-Node:

11-Bit-Identifier	2 Byte Nutzdaten	
0x000	0x01	Node ID

Mit einer CAN-Nachricht kann auch ein gesamtes Netzwerk gestartet werden:

Start All Remote-Nodes:

11-Bit-Identifier	2 Byte Nutzdaten	
0x000	0x01	0x00

Nach Start des Knotens oder des gesamten Netzwerks befindet sich das Gerät im Zustand *Operational*. Das Gerät kann jetzt über PDOs bedient werden.

Im Zustand *Stopped* befindet sich das Gerät im Fehlerzustand und kann nicht mehr über SDO oder PDO bedient werden. Die Kommunikation mit dem Gerät ist hier nur mit NMT-Nachrichten möglich.

Eine NMT-Nachricht besteht immer aus 2 Byte auf dem Identifier 0x000.

NMT-Nachricht

11-Bit-Identifier	2 Byte Nutzdaten	
0x000	CS	Node ID

Belegung der Nutzdaten:

- CS: Command-Specifier (siehe Tab. 11)
- Node ID: Knotenadresse (0 = alle Knoten)

Bei schweren Kommunikationsfehlern geht der Motion Controller standardmäßig in den NMT-Zustand *Pre-Operational*. Im Objekt 0x1029 kann ein anderes Verhalten eingestellt werden.

3.8.1 Boot-Up

Der Motion Controller sendet unmittelbar nach der Initialisierungsphase eine Boot-up-Nachricht. Eine Boot-up-Nachricht signalisiert das Ende der Initialisierungsphase einer neu eingeschalteten Baugruppe. Eine Boot-up-Nachricht ist eine CAN-Nachricht mit einem Datenbyte (Byte 0 = 0x00) auf dem Identifier der Node-Guarding-Nachricht (0x700 + Node ID).

11-Bit-Identifier	1 Byte Nutzdaten
0x700 (1792d) + Node ID	0x00

3.8.2 Überwachungsfunktionen

Nur eine Überwachungsfunktion, Node-Guarding oder Heartbeat, kann verwendet werden.

3.8.2.1 Node-Guarding

Das Node-Guarding-Objekt fragt den momentanen Zustand des Geräts ab. Dazu setzt der Master ein Remote-Frame mit einer Anforderung auf dem Guarding-Identifier des zu überwachenden Knotens. Der zu überwachende Knoten antwortet mit der Guarding-Nachricht, die den aktuellen Status des Knotens und ein Toggle-Bit enthält.

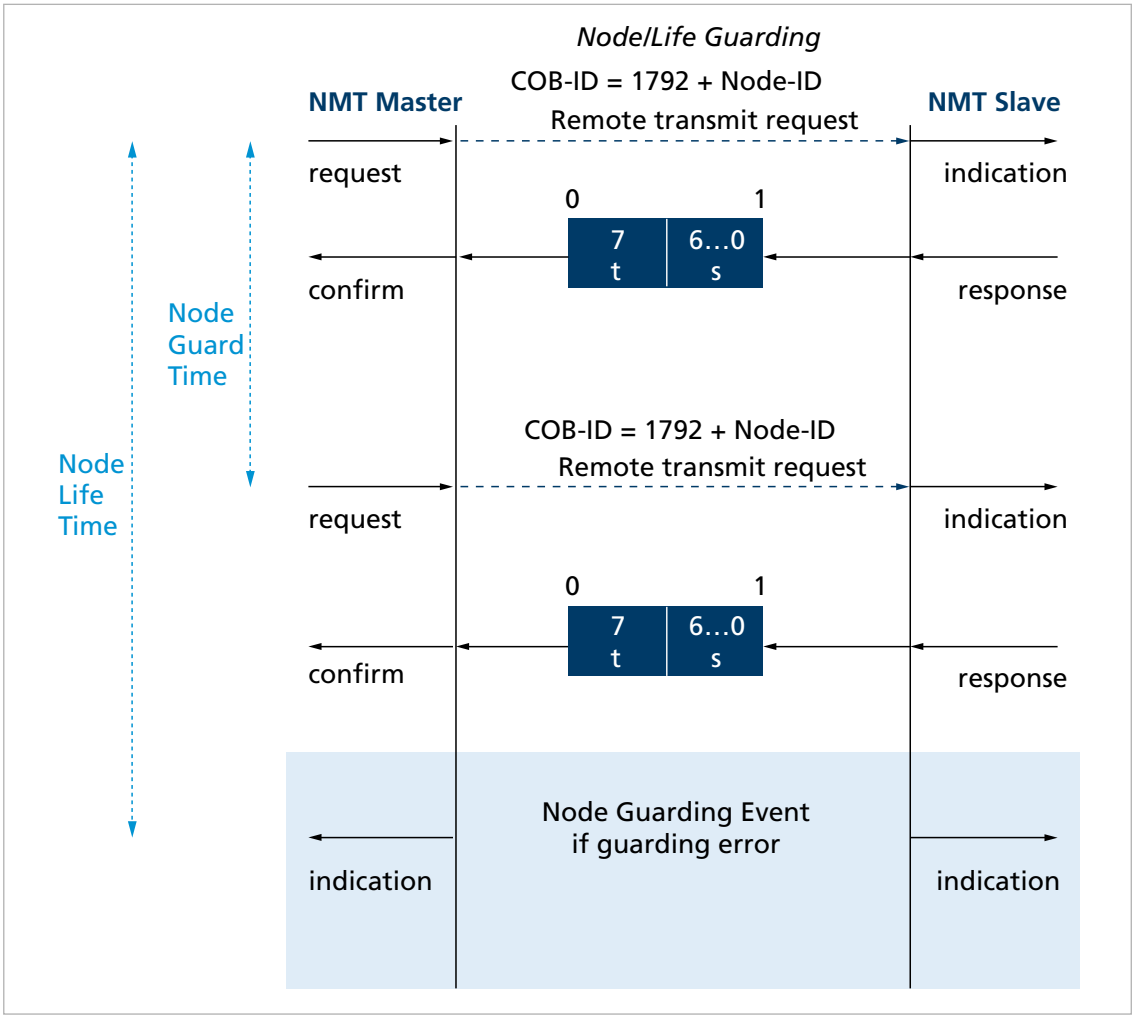

Abb. 8: Schaubild Node-Guarding-Protokoll

t: Toggle Bit
Anfänglich 0, wechselt in jedem Guarding-Telegramm seinen Wert
s: Status
s = 0x04 (4d): Stopped
s = 0x05 (5d): Operational
s = 0x7F (127d): Pre-Operational

Wenn eine Node-Life-Time > 0 eingestellt ist (Objekte 0x100C und 0x100D) und innerhalb der angegebenen Life-Time keine Node-Guarding-Abfrage des Masters eintrifft, wird ein Node-Guarding-Fehler gesetzt. Über das FAULHABER Fehlerregister (Objekt 0x2321) wird

die Reaktion auf einen Node-Guarding-Fehler eingestellt (siehe Tab. 14). Standardmäßig wird die Emergency-Nachricht 0x8130 versendet.

3.8.2.2 Heartbeat

Der Motion Controller kann sowohl als Heartbeat-Producer wie auch als Heartbeat-Consumer eingestellt werden.

- **Heartbeat-Producer:** Der Motion Controller setzt zyklisch eine Nachricht ab, die von ein oder mehreren Heartbeat-Consumern im Netzwerk empfangen wird.
- **Heartbeat-Consumer:** Der Motion Controller reagiert mit dem im FAULHABER Fehlerregister (Objekt 0x2320) eingestellten Verhalten, wenn innerhalb der Heartbeat-Consumer-Time keine Heartbeat-Nachricht des zu überwachenden Heartbeat-Producers eintrifft (siehe Tab. 12).

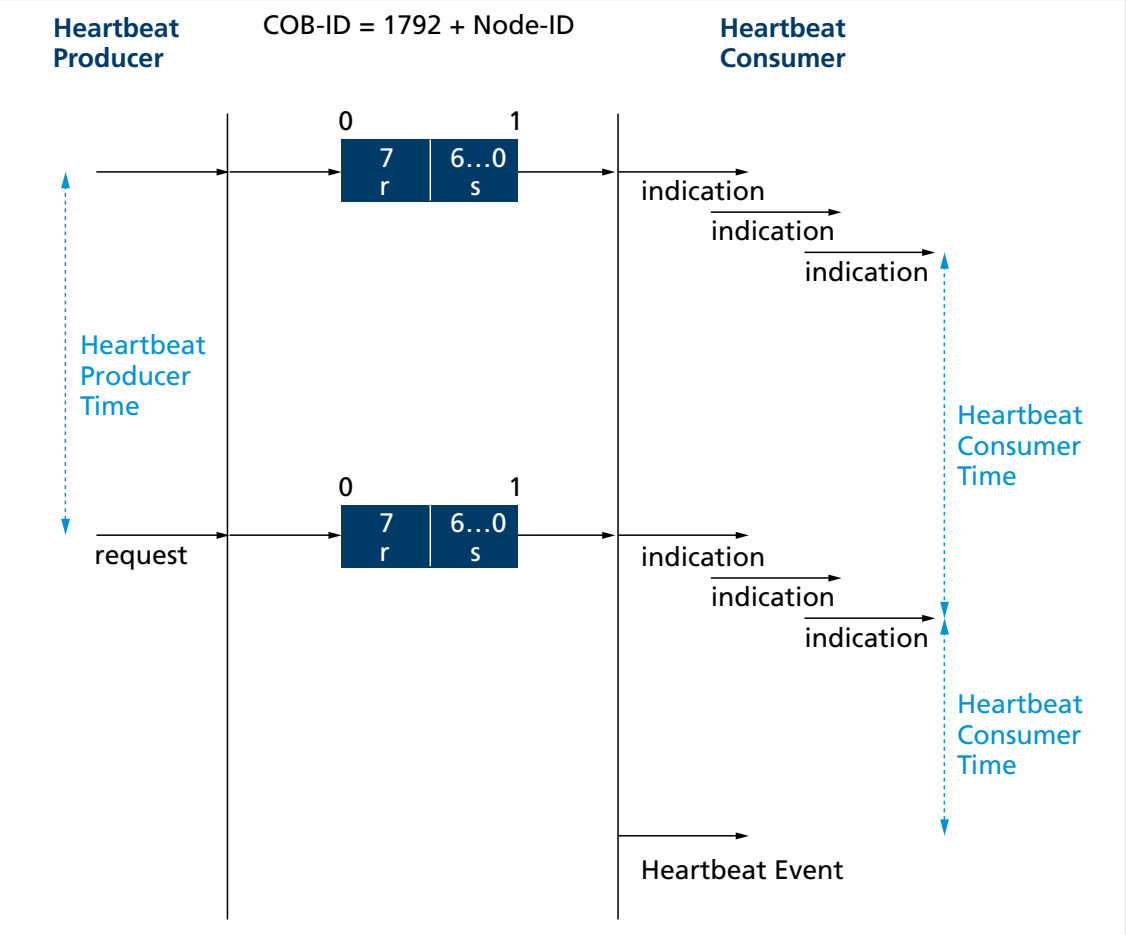

Abb. 9: Schaubild Heartbeat-Protokoll

r: Reserved
Immer 0
s: Status
s = 0x00 (0d): Boot-Up
s = 0x04 (4d): Stopped
s = 0x05 (5d): Operational
s = 0x7F (127d): Pre-Operational

3.8.3 Einstellung der Überwachungsfunktionen

- Nur einer der beiden Überwachungsmechanismen (Node-Guarding, Heartbeat) kann aktiviert sein.
- Bei einer Producer-Heartbeat-Time > 0 (Objekt 0x1017) arbeitet der Motion Controller als Heartbeat-Producer. Der Motion Controller sendet im Zeitintervall der Producer-Heartbeat-Time eine Heartbeat-Nachricht. Die Node-Guarding-Time wird auf 0 gesetzt (siehe Kap. 3.8.2.1, S. 32).
- Ist Heartbeat aktiviert, entspricht die Boot-up-Nachricht nach dem Einschalten der ersten Heartbeat-Nachricht. Weitere Heartbeats folgen im Abstand der Producer-Heartbeat-Time.
- Ist zusätzlich zur Producer-Heartbeat-Time eine Heartbeat-Consumer-Time > 0 eingestellt (Objekt 0x1016.01), arbeitet der Motion Controller als Heartbeat-Consumer. Die Einstellungen des Heartbeat-Producers sind unwirksam. Die Node ID des zu überwachenden Masters und die Heartbeat-Consumer-Time wird in das Objekt 0x1016 eingetragen.
- Die Heartbeat-Consumer-Time muss immer größer sein als die Producer-Heartbeat-Time des Masters.
- Falls der Motion Controller innerhalb der eingestellten Heartbeat-Consumer-Time keine Heartbeat-Nachricht des Masters erhält, wird ein Heartbeat-Event ausgelöst. Die Reaktion auf ein Heartbeat-Event wird über die Error-Mask des FAULHABER Fehlerregisters (Objekt 0x2321) eingestellt (siehe Tab. 12). Standardmäßig wird die Emergency-Nachricht 0x8130 versendet.
- Beim Versuch eine Node-Guarding-Zeit einzustellen, während der Heartbeat-Producer aktiviert ist, wird der SDO-Fehler 0x08000020 (Zugriff nicht möglich) versendet.

3.9 Einträge im Objektverzeichnis

Das Objektverzeichnis verwaltet die Konfigurationsparameter. Das Objektverzeichnis ist in drei Bereiche unterteilt. Jedes Objekt kann über seinen Index und Subindex referenziert werden (SDO-Protokoll).

- Kommunikationsparameter (Index 0x1000 bis 0x1FFF, enthält Kommunikationsobjekte nach CiA 301, siehe Kap. 5.1, S. 41)
- Herstellerspezifischer Bereich (Index 0x2000 bis 0x5FFF, enthält herstellerspezifische Objekte, siehe Kap. 5.2, S. 49)
- Standardisierte Geräteprofile (0x6000 bis 0x9FFF, enthält die vom Motion Controller unterstützten Objekte, siehe Dokumentation der Antriebsfunktionen)

3.10 Fehlerbehandlung

3.10.1 CAN-Fehler

CAN-Overrun (object lost)
Falls Nachrichten verloren gehen, versendet der Controller die Emergency-Nachricht 0x8110. Im Error-Register wird das Bit 4 (Communication-Error) und im FAULHABER Fehlerregister das Bit 7 (CAN-Overrun) gesetzt. Die Emergency-Nachricht wird zeitverzögert versendet. Der Fehler wird nicht durch die Emergency-Nachricht (0x000) zurückgenommen. Die entsprechenden Bits im Error-Register und im FAULHABER Fehlerregister werden nicht gelöscht.

CAN in Error-Passive-Mode
Falls das CAN-Modul des Antriebs im Zustand *Error-Passive* ist, wird die Emergency-Nachricht 0x8120 versendet. Im Error-Register wird das Bit 4 (Communication-Error) und im FAULHABER Fehlerregister das Bit 6 (CAN in Error-Passive-Mode) gesetzt. Die Emergency-Nachricht (0x000) wird versendet und der Fehler wird zurückgenommen, wenn der Antrieb wieder in den Zustand *Error-Active* geht.

Recovered from Bus-Off
Falls das CAN-Modul des Antriebs im Zustand *Bus-Off* ist und eine gültige Nachricht empfängt, wird die Emergency-Nachricht 0x8140 versendet. Die Emergency-Nachricht meldet, dass der Zustand *Bus-Off* verlassen wurde. Im Error-Register wird das Bit 4 (Communication-Error) und im FAULHABER Fehlerregister das Bit 9 (Recovered from Bus-Off) gesetzt. Der Fehler wird nicht zurückgenommen. Die entsprechenden Bits im Error-Register und im FAULHABER Fehlerregister werden nicht gelöscht.

Die Fehler CAN-Overrun und Recovered from bus off sind schwere Kommunikationsfehler. Die entsprechenden Bits im Error-Register und im FAULHABER Fehlerregister können nur mit einem Neustart des Motion Controllers zurückgesetzt werden. Weitere schwere Kommunikationsfehler sind:

- Node-Guarding-Timeouts
- Heartbeat-Timeouts

3.10.2 Gerätefehler

Tab. 12: FAULHABER Fehlerregister (0x2320)

Index	Subindex	Name	Typ	Attr.	Standardwert	Bedeutung
0x2320	0x00	Fault Register	U16	ro	–	FAULHABER Fehlerregister

Das FAULHABER Fehlerregister enthält bitcodiert die zuletzt aufgetretenen Fehler. Die Fehler können durch Selektion der gewünschten Fehlerarten über das Objekt Error Mask (0x2321) maskiert werden.

Tab. 13: Fehlercodierung

Error-Bit	Fehlermeldung	Beschreibung
0x0001	SpeedDeviationError	Geschwindigkeitsabweichung zu groß
0x0002	FollowingError	Schleppfehler
0x0004	OverVoltageError	Überspannung detektiert
0x0008	UnderVoltageError	Unterspannung detektiert
0x0010	TempWarning	Temperatur überschritten, bei der eine Warnung ausgegeben wird
0x0020	TempError	Temperatur überschritten, bei der eine Fehlermeldung ausgegeben wird
0x0040	EncoderError	Fehler am Encoder detektiert
0x0080	IntHWError	Interner Hardwarefehler
0x0100	ModuleError	Fehler am externen Modul
0x0200	CurrentMeasError	Strommessfehler
0x0400	MemError	Speicherfehler (EEPROM)
0x0800	ComError	Kommunikationsfehler
0x1000	CalcError	Interner Softwarefehler
0x2000	DynamicError	Aktuelle Geschwindigkeit ist größer als die eingestellte Maximalgeschwindigkeit des Motors.
0x4000	–	Nicht verwendet, Wert = 0
0x8000	–	Nicht verwendet, Wert = 0

Jeder dieser Fehler entspricht auch einem Emergency Error Code. (siehe Kap. 3.6, S. 26).

Die Error Mask beschreibt die Behandlung interner Fehler entsprechend der Fehlercodierung (siehe Tab. 13).

Tab. 14: Error Mask (0x2321)

Index	Subindex	Name	Typ	Attr.	Standardwert	Bedeutung
0x2321	0x00	Number of Entries	U8	ro	6	Anzahl Objekteinträge
	0x01	Emergency Mask	U16	rw	0xFFFF	Fehler, für die eine Fehlermeldung verschickt werden
	0x02	Fault Mask	U16	rw	0x0000	Fehler, für die die Zustandsmaschine des Antriebs in den Zustand *Fault Reaction Active* geht
	0x03	Error Out Mask	U16	rw	0x0000	Fehler, für die der Fehler-Ausgangspin gesetzt wird
	0x04	Disable Voltage Mask	U16	ro	0x4024	Fehler, die den Antrieb abschalten (nicht konfigurierbar)
	0x05	Disable Voltage User Mask	U16	rw	0x0000	Fehler, die den Antrieb abschalten (konfigurierbar)
	0x06	Quick Stop Mask	U16	rw	0x0000	Fehler, für die die Zustandsmaschine des Antriebs in den Zustand *Quick Stop Active* geht

Beispiele:

- Beim Setzen der Fault Mask (Subindex 2) von Objekt 0x2321 auf 0x0001 wird der Antrieb bei Überstrom ausgeschaltet und dessen Zustandsmaschine wird in den Zustand *Fault Reaction Active* versetzt.
- Wenn der Subindex 3 von Objekt 0x2321 auf 0 gesetzt ist, zeigt der Fehlerausgang (Fault-Pin) keine Fehler an. Wenn der Subindex 3 von Objekt 0x2321 auf 0xFFFF gesetzt ist, zeigt der Fehlerausgang (Fault-Pin) alle Fehler an.

4 Kommunikationseinstellungen

Die FAULHABER-Antriebe werden standardmäßig mit eingestellter Knotennummer 1 und mit eingestellter automatischer Baudratenerkennung (AutoBaud) ausgeliefert.

Im Netzwerkbetrieb sollte die verwendete Netzwerk-Übertragungsrate fest eingestellt werden.

4.1 Einstellung über das CAN-Netzwerk

Für die Einstellung über das CAN-Netzwerk wird der FAULHABER Motion Manager oder ein anderes Konfigurationstool, das das LSS-Protokoll (Layer Setting Service and Protocol) nach CiA 305 unterstützt, benötigt.

Der FAULHABER Motion Manager muss auf einem PC mit unterstütztem CAN-Interface installiert sein.

Es gibt zwei Möglichkeiten zur Einstellung der Kommunikationsparameter:

- Ein einzelner Antrieb ist an der CAN-Schnittstelle des Konfigurations-Tools angeschlossen:
  Über den „LSS Switch Mode Global“ wird der Antrieb ohne weitere Angaben in den Konfigurationsmodus versetzt, um Knotennummer und Baudrate einzustellen.
- Der zu konfigurierende Antrieb ist innerhalb eines Netzwerks über die CAN-Schnittstelle an das Konfigurations-Tool angeschlossen:
  Über den „LSS Switch Mode Selective“ wird der gewünschte Antrieb durch Eingabe der LSS-Adresse (Vendor-ID, Productcode, Revision Number, Seriennummer) adressiert und in den Konfigurationsmodus versetzt, um Knotennummer und Baudrate einzustellen.

  Die FAULHABER-Antriebe der Serie MC V3.0 benötigen folgende Eingaben:

  - Vendor-ID: 327
  - Produktcode: 48
  - Revision Number: 1.0
  - Seriennummer: Siehe Produkt-Aufkleber

Das LSS-Protokoll unterstützt neben der Einstellung von Knotennummer und Baudrate auch das Auslesen der LSS-Adressen von angeschlossenen Einheiten und das Auslesen der eingestellten Node ID.

Zur LSS-Kommunikation werden die Identifier 0x7E5 (vom Master) und 0x7E4 (vom Slave) verwendet.

Der Motion Controller speichert nach der Konfiguration die eingestellten Parameter in das EEPROM. Sie sind nach Aus- und Einschalten wieder verfügbar.

Für eine detaillierte Beschreibung des LSS-Protokolls wird auf das Dokument CiA 305 verwiesen.

4.1.1 Einstellung der Knotennummer

- Knotennummern 1 bis 127 können eingestellt werden.
- Die Node ID 255 (0xFF) kennzeichnet den Knoten als nicht konfiguriert. Der Knoten befindet sich nach dem Einschalten im LSS-Init-Status, bis eine gültige Knotennummer übermittelt wird. Nachdem eine gültige Knotennummer an den Knoten übermittelt wurde, wird die NMT-Initialisierung fortgesetzt.

4.1.2 Einstellung der Baudrate

- Bei aktivierter automatischer Baudratenerkennung (AutoBaud) kann der Antrieb in ein Netzwerk mit beliebiger Übertragungsrate gemäß Tab. 15 eingesetzt werden. Nach spätestens 24 Telegrammen (3 pro Baudrate) auf der Busleitung ist die Baudrate des Netzwerks detektiert. Der Antrieb stellt sich entsprechend der Baudrate des Netzwerks ein.
- Wenn die automatische Baudratenerkennung aktiv ist, können Telegramme solange nicht verarbeitet werden, bis die Baudrate detektiert wurde. Das Hochfahren des Systems dauert bei automatischer Baudratenerkennung entsprechend länger.
- Eine feste Baudrate gemäß Tab. 15 kann durch Angabe des Index 0 bis 8 eingestellt werden.

Tab. 15: Bit-Timing-Parameter

Baudrate	Index
1 000 kBit/s	00
800 kBit/s	01
500 kBit/s	02
250 kBit/s	03
125 kBit/s	04
50 kBit/s	06
20 kBit/s	07
10 kBit/s	08
AutoBaud	09

4.1.3 Automatische Einstellung der COB-IDs

Bei einem Wechsel von der Knotennummer 255 (unkonfigurierter CANopen-Knoten) nach einer validen Knotennummer werden die COB-IDs für die Receive- und Transmit-PDOs (RxPDO, TxPDO) und für Emergency (EMCY) automatisch auf ihre Standardwerte eingestellt (siehe Kap. 5.1, S. 41, Objekte 0x1014.00, 0x1400.01, 0x1401.01, 0x1402.01, 0x1403.01, 0x1800.01, 0x1801.01, 0x1802.01, 0x1803.01).

Die Konfiguration muss mit dem Save-Befehl gespeichert werden.

4.2 Einstellung der Knotennummer über das Objektverzeichnis

Alternativ zu der LSS-Methode über das CAN-Netzwerk kann die Knotennummer auch über jede am Antrieb verfügbare Schnittstelle (CAN, USB, RS232) eingestellt werden.

Die Einstellung erfolgt durch Beschreiben des Objekts 0x2400.03 im Objektverzeichnis:

Tab. 16: CAN Baudrate Index und Knotennummer

Index	Subindex	Name	Typ	Attr.	Standardwert	Bedeutung
0x2400	0x00	Number of Entries	U8	ro	8	Anzahl Objekteinträge
	0x01	CAN Rate	U8	rw	9	Index der CAN-Baudrate gemäß Tab. 15
	0x03	Node ID	U8	rw	1	Knotennummer

Über das Objekt 0x2400.01 kann die aktuelle Einstellung der Baudrate (AutoBaud oder feste Baudrate) ausgelesen werden.

Eine Änderung der Knotennummer über das Objekt 0x2400.03 wird mit der letzten Knotennummer quittiert. Erst nach einem Save-Befehl für Anwendungsparameter und einem anschließenden Reset-Befehl wird die geänderte Knotennummer übernommen.

5 Parameterbeschreibung

5.1 Kommunikationsobjekte nach CiA 301

Device Type

Index	Subindex	Name	Typ	Attr.	Standardwert	Bedeutung
0x1000	0x00	Device Type	U32	ro	0x00420192	Angabe des Gerätetyps

Enthält Informationen zum Gerätetyp, aufgeteilt in zwei 16-Bit-Feldern:

- Byte MSB (Most Significant Byte): Additional Information = 0x42 (Servo drive, type specific PDO mapping)
- Byte LSB (Least Significant Byte): Device Profile Number = 0x192 (402d)

Error Register

Index	Subindex	Name	Typ	Attr.	Standardwert	Bedeutung
0x1001	0x00	Error Register	U8	ro	ja	Fehlerregister

Das Error Register beinhaltet bitcodiert die zuletzt aufgetretenen Fehlerarten.

Dieser Parameter kann in ein PDO gemappt werden.

Predefined Error Field (Fehlerspeicher)

Index	Subindex	Name	Typ	Attr.	Standardwert	Bedeutung
0x1003	0x00	Number of Errors	U8	rw	–	Anzahl gespeicherter Fehler
	0x01–0x08	Standard Error Field	U32	ro	–	Zuletzt aufgetretene Fehlercodes

Der Fehlerspeicher enthält die Codierung der zuletzt aufgetretenen Fehler.

- Byte MSB: Error Register
- Byte LSB: Error Code

Die Bedeutung der Fehlercodes ist in Kap. 3.6, S. 26 beschrieben.

Durch Schreiben einer 0 auf Subindex 0 wird der Fehlerspeicher gelöscht.

COB-ID SYNC

Index	Subindex	Name	Typ	Attr.	Standardwert	Bedeutung
0x1005	0x00	COB ID SYNC	U32	rw	0x80	CAN-Objekt-Identifier des SYNC-Objekts

Manufacturer Device Name

Index	Subindex	Name	Typ	Attr.	Standardwert	Bedeutung
0x1008	0x00	Manufacturer Device Name	Vis-String	const	–	Gerätename

Zum Auslesen des Manufacturer Device Namens muss das Segmented-SDO-Protokoll verwendet werden.

Manufacturer Hardware Version

Index	Subindex	Name	Typ	Attr.	Standardwert	Bedeutung
0x1009	0x00	Manufacturer Hardware Version	Vis-String	const	–	Hardwareversion

Zum Auslesen der Manufacturer Hardware Version wird das Segmented-SDO-Protokoll verwendet werden.

Manufacturer Software Version

Index	Subindex	Name	Typ	Attr.	Standardwert	Bedeutung
0x100A	0x00	Manufacturer Software Version	Vis-String	const	–	Softwareversion

Zum Auslesen der Manufacturer-Software-Version muss das Segmented-SDO-Protokoll verwendet werden.

Guard Time

Index	Subindex	Name	Typ	Attr.	Standardwert	Bedeutung
0x100C	0x00	Guard Time	U16	rw	0	Überwachungszeit für Node-Guarding

Angabe der Guard Time in Millisekunden. Der Wert 0 schaltet das Node-Guarding aus (Kap. 3.8.2.1, S. 32).

Life Time Factor

Index	Subindex	Name	Typ	Attr.	Standardwert	Bedeutung
0x100D	0x00	Life Time Factor	U8	rw	0	Zeitfaktor für Node-Guarding

Der Life Time Factor multipliziert mit der Guard Time ergibt die Life Time für das Node Guarding (Kap. 3.8, S. 29). Der Wert 0 schaltet das Node Guarding aus.

Store Parameters

Tab. 17: Parameter speichern

Index	Subindex	Name	Typ	Attr.	Standardwert	Bedeutung
0x1010	0x00	Number of Entries	U8	ro	9	Anzahl Objekteinträge
	0x01	Save All Parameters	U32	rw	1	Speichert alle Parameter
	0x02	Save Comm Parameters	U32	rw	1	Kommunikationsparameter speichern (Objektverzeichnis-Einträge 0x0000 bis 0x1FFF)
	0x03	Save App Parameters	U32	rw	1	Anwendungsparameter speichern (Objektverzeichnis-Einträge 0x2000 bis 0x6FFF)
	0x04	Save App Parameters 1	U32	rw	1	Anwendungsparameter für den direkten Wechsel (Satz 1) speichern
	0x05	Save App Parameters 2	U32	rw	1	Anwendungsparameter für den direkten Wechsel (Satz 2) speichern

Das Objekt Store Parameters speichert Konfigurationsparameter in den Flash-Speicher. Ein Lesezugriff liefert Informationen über die Speichermöglichkeiten. Das Schreiben der Signatur “save” auf den entsprechenden Subindex leitet den Speichervorgang ein.

Tab. 18: Signatur “save”

Signature	ISO 8 859 (“ASCII”)	hex
MSB	e	65h
	v	76h
	a	61h
LSB	s	73h

HINWEIS!

Der Flash-Speicher ist für 10 000 Schreibzyklen ausgelegt. Wird dieser Befehl mehr als 10 000 mal ausgeführt, ist die Funktion des Flash-Speichers nicht mehr gewährleistet.

- Häufiges Speichern vermeiden.
- Nach 10 000 Speicherzyklen Gerät wechseln.

Restore Default Parameters

Tab. 19: Wiederherstellen von Parametern

Index	Subindex	Name	Typ	Attr.	Standardwert	Bedeutung
0x1011	0x00	Number of Entries	U8	ro	6	Anzahl Objekteinträge
	0x01	Restore all Default Parameters	U32	rw	1	Alle Werkseinstellungen wiederherstellen
	0x02	Restore Comm Default Parameters	U32	rw	1	Werkseinstellungen für Kommunikations-Parameter (0x0000 bis 0x1FFF) wiederherstellen
	0x03	Restore App Default Parameters	U32	rw	1	Werkseinstellungen für Anwendungs-Parameter (ab 0x2000) wiederherstellen
	0x04	Reload User Parameters	U32	rw	1	Vom Benutzer zuletzt gespeicherte Anwendungsparameter (ab 0x2000) wiederherstellen
	0x05	Reload Application Parameters 1	U32	rw	1	Anwendungsparametersatz 1 für den direkten Wechsel
	0x06	Reload Application Parameters 2	U32	rw	1	Anwendungsparametersatz 2 für den direkten Wechsel

Das Objekt Restore Default Parameters lädt Standardkonfigurationsparameter. Die Standardkonfigurationsparameter sind entweder der Auslieferungszustand oder der letzte gespeicherte Zustand. Ein Lesezugriff liefert Informationen über die Restoremöglichkeit. Das Schreiben der Signatur “load” auf den entsprechenden Subindex leitet den Restorevorgang ein:

Tab. 20: Signatur “load”

Signature	ISO 8859 (“ASCII”)	hex
MSB	d	64h
	a	61h
	o	6Fh
LSB	l	6Ch

Der Auslieferungszustand darf nur bei abgeschalteter Endstufe geladen werden.

COB-ID Emergency-Nachricht

Index	Subindex	Name	Typ	Attr.	Standardwert	Bedeutung
0x1014	0x00	COB-ID EMCY	U32	rw	0x80 + Node-ID	CAN-Objekt-Identifier des Emergency Objects

Consumer Heartbeat Time

Index	Subindex	Name	Typ	Attr.	Standardwert	Bedeutung
0x1016	0x00	Number of Entries	U8	ro	1	Anzahl Objekteinträge
	0x01	Consumer Heartbeat Time	U32	rw	0	Heartbeat-Überwachungszeit

- Bit 0 bis 15 enthalten die Consumer Heartbeat Time in Millisekunden. Bei einem Wert von 0 ist die Consumer-Heartbeat-Funktion deaktiviert (Heartbeat)
- Bit 16 bis 23 enthalten die Knotennummer, an der die Heartbeat-Nachricht geschickt werden soll (Master Node ID).
- Bit 24 bis 31 sind unbenutzt (reserviert).

Producer Heartbeat Time

Index	Subindex	Name	Typ	Attr.	Standardwert	Bedeutung
0x1017	0x00	Producer Heartbeat Time	U16	rw	0	Heartbeat Sende-Zeitintervall

Das Objekt Producer Heartbeat Time enthält das Producer-Heartbeat-Zeitintervall in Millisekunden. Bei einem Wert von 0 ist die Producer-Heartbeat-Funktion deaktiviert (siehe Einstellung der Überwachungsfunktionen).

Identity Object

Index	Subindex	Name	Typ	Attr.	Standardwert	Bedeutung
0x1018	0x00	Number of Entries	U8	ro	4	Anzahl Objekteinträge
	0x01	Vendor ID	U32	ro	327	Herstellerkennnummer (FAULHABER: 327)
	0x02	Product Code	U32	ro	48	Produktkennnummer
	0x03	Revision Number	U32	ro	–	Versionsnummer
	0x04	Serial Number	U32	ro	–	Seriennummer

Error Behaviour

Index	Subindex	Name	Typ	Attr.	Standardwert	Bedeutung
0x1029	0x00	Number of Entries	U8	ro	1	Anzahl Objekteinträge
	0x01	Communication Error	U8	rw	0	Verhalten bei Kommunikationsfehlern 0 = Zustand Pre-Operational 1 = Keine Zustandsänderung 2 = Zustand Stopped

Der Motion Controller wechselt bei schweren Kommunikationsfehlern in den NMT-Zustand *Pre-Operational*. Über den Subindex 1 kann das Verhalten bei schweren Kommunikationsfehlern geändert werden.

Server SDO Parameter

Index	Subindex	Name	Typ	Attr.	Standardwert	Bedeutung
0x1200	0x00	Number of Entries	U8	ro	2	Anzahl Objekteinträge
	0x01	COB ID Client to Server (rx)	U32	ro	0x600 + Node-ID	CAN-Objekt-Identifier der Server RxSDO
	0x02	COB ID Server to Client (tx)	U32	ro	0x580 + Node-ID	CAN-Objekt-Identifier der Server TxSDO

Receive PDO1 Parameter

Index	Subindex	Name	Typ	Attr.	Standardwert	Bedeutung
0x1400	0x00	Number of Entries	U8	ro	2	Anzahl Objekteinträge
	0x01	COB ID Used by RxPDO1	U32	rw	0x200 + Node-ID	CAN-Objekt-Identifier der Server RxPDO1
	0x02	Transmission Type	U8	rw	255 (asynchr.)	PDO-Übertragungsart

Receive PDO2 Parameter

Index	Subindex	Name	Typ	Attr.	Standardwert	Bedeutung
0x1401	0x00	Number of Entries	U8	ro	2	Anzahl Objekteinträge
	0x01	COB ID Used by RxPDO2	U32	rw	0x300 + Node-ID	CAN-Objekt-Identifier der Server RxPDO2
	0x02	Transmission Type	U8	rw	255 (asynchr.)	PDO-Übertragungsart

Receive PDO3 Parameter

Index	Subindex	Name	Typ	Attr.	Standardwert	Bedeutung
0x1402	0x00	Number of Entries	U8	ro	2	Anzahl Objekteinträge
	0x01	COB ID Used by RxPDO3	U32	rw	0x400 + Node-ID	CAN-Objekt-Identifier der Server RxPDO3
	0x02	Transmission Type	U8	rw	255 (asynchr.)	PDO-Übertragungsart

Receive PDO4 Parameter

Index	Subindex	Name	Typ	Attr.	Standardwert	Bedeutung
0x1403	0x00	Number of Entries	U8	ro	2	Anzahl Objekteinträge
	0x01	COB ID Used by RxPDO4	U32	rw	0x500 + Node-ID	CAN-Objekt-Identifier der Server RxPDO4
	0x02	Transmission Type	U8	rw	255 (asynchr.)	PDO-Übertragungsart

Receive PDO1 Mapping

Index	Subindex	Name	Typ	Attr.	Standardwert	Bedeutung
0x1600	0x00	Number of Mapped Objects	U8	ro	1	Anzahl gemappter Objekte
	0x01	RxPDO1 Mapping Entry 1	U32	rw	0x60400010	Verweis auf 16-Bit Controlword (0x6040)
	0x02	RxPDO1 Mapping Entry 2	U32	rw	0	
	0x03	RxPDO1 Mapping Entry 3	U32	rw	0	
	0x04	RxPDO1 Mapping Entry 4	U32	rw	0	

Receive PDO2 Mapping

Index	Subindex	Name	Typ	Attr.	Standardwert	Bedeutung
0x1601	0x00	Number of Mapped Objects	U8	ro	2	Anzahl gemappter Objekte
	0x01	RxPDO2 Mapping Entry 1	U32	rw	0x60400010	Verweis auf 16-Bit Controlword (0x6040)
	0x02	RxPDO2 Mapping Entry 2	U32	rw	0x607A0020	Verweis auf 32-Bit Target Position (0x607A)
	0x03	RxPDO2 Mapping Entry 3	U32	rw	0	
	0x04	RxPDO2 Mapping Entry 4	U32	rw	0	

Receive PDO3 Mapping

Index	Subindex	Name	Typ	Attr.	Standardwert	Bedeutung
0x1602	0x00	Number of Mapped Objects	U8	ro	2	Anzahl gemappter Objekte
	0x01	RxPDO3 Mapping Entry 1	U32	rw	0x60400010	Verweis auf 16-Bit Controlword (0x6040)
	0x02	RxPDO3 Mapping Entry 2	U32	rw	0x60FF0020	Verweis auf 32-Bit Target Velocity (0x60FF)
	0x03	RxPDO3 Mapping Entry 3	U32	rw	0	
	0x04	RxPDO3 Mapping Entry 4	U32	rw	0	

Receive PDO4 Mapping

Index	Subindex	Name	Typ	Attr.	Standardwert	Bedeutung
0x1603	0x00	Number of Mapped Objects	U8	ro	2	Anzahl gemappter Objekte
	0x01	RxPDO4 Mapping Entry 1	U32	rw	0x60400010	Verweis auf 16-Bit Controlword (0x6040)
	0x02	RxPDO4 Mapping Entry 2	U32	rw	0x60710010	Verweis auf 16-Bit Target Torque (0x6071)
	0x03	RxPDO4 Mapping Entry 3	U32	rw	0	
	0x04	RxPDO4 Mapping Entry 4	U32	rw	0	

Transmit PDO1 Parameter

Index	Subindex	Name	Typ	Attr.	Standardwert	Bedeutung
0x1800	0x00	Number of Entries	U8	ro	2	Anzahl Objekteinträge
	0x01	COB ID Used by TxPDO1	U32	rw	0x180 + Node-ID	CAN-Objekt-Identifier der TxPDO1
	0x02	Transmission Type	U8	rw	255 (asynchr.)	PDO-Übertragungsart

Transmit PDO2 Parameter

Index	Subindex	Name	Typ	Attr.	Standardwert	Bedeutung
0x1801	0x00	Number of Entries	U8	ro	2	Anzahl Objekteinträge
	0x01	COB ID used by TxPDO2	U32	rw	0x280 + Node-ID	CAN-Objekt-Identifier der TxPDO2
	0x02	Transmission Type	U8	rw	255 (asynchr.)	PDO-Übertragungsart

Transmit PDO3 Parameter

Index	Subindex	Name	Typ	Attr.	Standardwert	Bedeutung
0x1802	0x00	Number of Entries	U8	ro	2	Anzahl Objekteinträge
	0x01	COB ID Used by TxPDO3	U32	rw	0x380 + Node-ID	CAN-Objekt-Identifier der TxPDO3
	0x02	Transmission Type	U8	rw	255 (asynchr.)	PDO-Übertragungsart

Transmit PDO4 Parameter

Index	Subindex	Name	Typ	Attr.	Standardwert	Bedeutung
0x1803	0x00	Number of Entries	U8	ro	2	Anzahl Objekteinträge
	0x01	COB ID Used by TxPDO4	U32	rw	0x480 + Node-ID	CAN-Objekt-Identifier der TxPDO4
	0x02	Transmission Type	U8	rw	255 (asynchr.)	PDO-Übertragungsart

Transmit PDO1 Mapping

Index	Subindex	Name	Typ	Attr.	Standardwert	Bedeutung
0x1A00	0x00	Number of Mapped Objects	U8	rw	1	Anzahl gemappter Objekte
	0x01	TxPDO1 Mapping Entry 1	U32	rw	0x60410010	Verweis auf 16-Bit Statusword (0x6041)
	0x02	TxPDO1 Mapping Entry 2	U32	rw	0	
	0x03	TxPDO1 Mapping Entry 3	U32	rw	0	
	0x04	TxPDO1 Mapping Entry 4	U32	rw	0	

Transmit PDO2 Mapping

Index	Subindex	Name	Typ	Attr.	Standardwert	Bedeutung
0x1A01	0x00	Number of Mapped Objects	U8	rw	2	Anzahl gemappter Objekte
	0x01	TxPDO2 Mapping Entry 1	U32	rw	0x60410010	Verweis auf 16-Bit Statusword (0x6041)
	0x02	TxPDO2 Mapping Entry 2	U32	rw	0x60640020	Verweis auf 32-Bit Position Actual Value (0x6064)
	0x03	TxPDO2 Mapping Entry 3	U32	rw	0	
	0x04	TxPDO2 Mapping Entry 4	U32	rw	0	

Transmit PDO3 Mapping

Index	Subindex	Name	Typ	Attr.	Standardwert	Bedeutung
0x1A02	0x00	Number of Mapped Objects	U8	rw	2	Anzahl gemappter Objekte
	0x01	TxPDO3 Mapping Entry 1	U32	rw	0x60410010	Verweis auf 16-Bit Statusword (0x6041)
	0x02	TxPDO3 Mapping Entry 2	U32	rw	0x606C0020	Verweis auf 32-Bit Velocity Actual Value (0x606C)
	0x03	TxPDO3 Mapping Entry 3	U32	rw	0	
	0x04	TxPDO3 Mapping Entry 4	U32	rw	0	

Transmit PDO4 Mapping

Index	Subindex	Name	Typ	Attr.	Standardwert	Bedeutung
0x1A03	0x00	Number of Mapped Objects	U8	rw	2	Anzahl gemappter Objekte
	0x01	TxPDO4 Mapping Entry 1	U32	rw	0x60410010	Verweis auf 32-Bit Position Actual Value (0x6064)
	0x02	TxPDO4 Mapping Entry 2	U32	rw	0x60770010	Verweis auf 16-Bit Torque Actual Value (0x6077)
	0x03	TxPDO4 Mapping Entry 3	U32	rw	0	
	0x04	TxPDO4 Mapping Entry 4	U32	rw	0	

5.2 Herstellerspezifische Objekte

FAULHABER Fehlerregister (0x2320)

Index	Subindex	Name	Typ	Attr.	Standardwert	Bedeutung
0x2320	0x00	Fault Register	U16	ro	–	FAULHABER Fehlerregister

Das FAULHABER Fehlerregister enthält bitcodiert die zuletzt aufgetretenen Fehler. Die Fehler können durch Selektion der gewünschten Fehlerarten über das Objekt Error Mask (0x2321) maskiert werden.

Error Mask (0x2321)

Index	Subindex	Name	Typ	Attr.	Standardwert	Bedeutung
0x2321	0x00	Number of Entries	U8	ro	6	Anzahl Objekteinträge
	0x01	Emergency Mask	U16	rw	0xFFFF	Fehler, für die eine Fehlermeldung verschickt werden
	0x02	Fault Mask	U16	rw	0x0000	Fehler, für die die Zustandsmaschine des Antriebs in den Zustand *Fault Reaction Active* geht
	0x03	Error Out Mask	U16	rw	0x0000	Fehler, für die der Fehler-Ausgangspin gesetzt wird
	0x04	Disable Voltage Mask	U16	ro	0x4024	Fehler, die den Antrieb abschalten (nicht konfigurierbar)
	0x05	Disable Voltage User Mask	U16	rw	0x0000	Fehler, die den Antrieb abschalten (konfigurierbar)
	0x06	Quick Stop Mask	U16	rw	0x0000	Fehler, für die die Zustandsmaschine des Antriebs in den Zustand *Quick Stop Active* geht

Die Zustände der Antriebs-Zustandsmaschine sind in der Dokumentation der Antriebsfunktionen beschrieben.

Trace Configuration

Index	Subindex	Name	Typ	Attr.	Standardwert	Bedeutung
0x2370	0x00	Number of Entries	U8	ro	10	Anzahl Objekteinträge
	0x01	Trigger Source	U32	wo	0	Triggerquelle
	0x02	Trigger Threshold	S32	rw	0	Triggerschwelle
	0x03	Trigger Delay Offset	S16	rw	0	Triggerverzögerung
	0x04	Trigger Mode	U16	rw	0	Triggermodus
	0x05	Buffer Length	U16	rw	100	Pufferlänge
	0x06	Sample Time	U8	rw	1	Abtastrate der Aufzeichnung 1: in jedem Abtastschritt
	0x07	Trace Source of Channel 1	U32	wo	0	Tracequelle am Kanal 1
	0x08	Trace Source of Channel 2	U32	wo	0	Tracequelle am Kanal 2

Index	Subindex	Name	Typ	Attr.	Standardwert	Bedeutung
	0x09	Trace Source of Channel 3	U32	wo	0	Tracequelle am Kanal 3
	0x0A	Trace Source of Channel 4	U32	wo	0	Tracequelle am Kanal 4

Trace Buffer

Index	Subindex	Name	Typ	Attr.	Standardwert	Bedeutung
0x2371	0x00	Number of Entries	U8	ro	5	Anzahl Objekteinträge
	0x01	Trace State	U16	ro	0	Triggerstatus
	0x02	Trace Value of Channel 1	Vis-String	ro	–	Signalpuffer Kanal 1
	0x03	Trace Value of Channel 2	Vis-String	ro	–	Signalpuffer Kanal 2
	0x04	Trace Value of Channel 3	Vis-String	ro	–	Signalpuffer Kanal 3
	0x05	Trace Value of Channel 4	Vis-String	ro	–	Signalpuffer Kanal 4

CAN Baudrate Index und Knotennummer

Index	Subindex	Name	Typ	Attr.	Standardwert	Bedeutung
0x2400	0x00	Number of Entries	U8	ro	8	Anzahl Objekteinträge
	0x01	CAN Rate	U8	rw	9	Index der CAN-Baudrate gemäß Tab. 15
	0x03	Node ID	U8	rw	1	Knotennummer
	0x04	Communication Settings	U32	rw	0	Bitmaske für Kommunikationseinstellungen gemäß Tab. 21
	0x06	ComState	U16	ro	0	Bitmaske für Kommunikationsstatus gemäß Tab. 22

Tab. 21: Bedeutung der Bits zu 0x2400.04 (Communication Settings)

Bit	Beschreibung
0	Can Mandatory
1	AsyncDriveStatus
2…31	Reserved

Tab. 22: Bedeutung der Bits zu 0x2400.06 (ComState)

Bit	Beschreibung
0…6	Reserved
7	Transmit Overflow Signaled
8	BufferOverflow
9	GuardingFailed
10	GuardAgain
11	BusOffEnd
12	BusOffStart
13…14	Reserved
15	PDOLength

DR. FRITZ FAULHABER GMBH & CO. KG
Antriebssysteme

Faulhaberstraße 1
71101 Schönaich • Germany
Tel. +49(0)7031/638-0
Fax +49(0)7031/638-100
info@faulhaber.de
www.faulhaber.com

7000.00050, 7. Auflage, 2.02.2024