

# Kommunikations- handbuch

MC 5010

MCS

MC 5005

MC 3001

MC 5004

MC 3603

MC 5004 P STO

IMC

**RS232 / USB**

## Impressum

---

Version:  
7. Auflage, 2.02.2024

Copyright  
by Dr. Fritz Faulhaber GmbH & Co. KG  
Faulhaberstraße 1 · 71101 Schönaich

Alle Rechte, auch die der Übersetzung, vorbehalten.  
Ohne vorherige ausdrückliche schriftliche Genehmigung  
der Dr. Fritz Faulhaber GmbH & Co. KG darf kein Teil  
dieser Beschreibung vervielfältigt, reproduziert, in einem  
Informationssystem gespeichert oder verarbeitet oder in  
anderer Form weiter übertragen werden.

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](http://www.faulhaber.com)

# Inhalt

<b>1</b>	<b>Zu diesem Dokument .....</b>	<b>4</b>
1.1	Gültigkeit dieses Dokuments .....	4
1.2	Mitgeltende Dokumente .....	4
1.3	Umgang mit diesem Dokument .....	4
1.4	Abkürzungsverzeichnis .....	5
1.5	Symbole und Kennzeichnungen .....	6
<b>2</b>	<b>Überblick .....</b>	<b>7</b>
2.1	Grundaufbau des Motion Controllers .....	7
2.2	Voraussetzung für die Kommunikation .....	8
2.2.1	Betrieb über die RS232-Schnittstelle.....	8
2.2.1.1	Betrieb eines einzelnen Motion Controllers .....	8
2.2.1.2	RS232-Netzwerkbetrieb.....	9
2.2.2	Betrieb über die USB-Schnittstelle .....	9
2.3	FAULHABER Motion Manager .....	10
2.4	Parameter speichern und wiederherstellen .....	11
2.4.1	Parameter speichern .....	11
2.4.2	Einstellungen wiederherstellen .....	12
2.4.3	Parametersatz wechseln .....	12
<b>3</b>	<b>Protokollbeschreibung .....</b>	<b>15</b>
3.1	Einführung .....	15
3.2	Kommunikationsdienste .....	17
3.3	SDO (Servicedatenobjekt) .....	18
3.3.1	Expedited Transfer .....	18
3.3.1.1	Objektverzeichnis auslesen .....	18
3.3.1.2	Objektverzeichnis beschreiben .....	19
3.3.2	Segmented Transfer.....	20
3.3.2.1	SDO Block Upload.....	20
3.3.2.2	SDO Block Download.....	23
3.3.3	SDO-Fehlerbehandlung .....	26
3.4	Emergency-Objekt (Fehlermeldung) .....	28
3.5	Gerätesteuerung .....	30
3.5.1	BootUp-Nachricht .....	30
3.5.2	Reset Node .....	30
3.5.3	Device Control.....	31
3.6	Einträge im Objektverzeichnis .....	31
3.7	Fehlerbehandlung .....	32
3.7.1	Gerätefehler .....	32
<b>4</b>	<b>Kommunikationseinstellungen .....</b>	<b>34</b>
<b>5</b>	<b>Parameterbeschreibung .....</b>	<b>35</b>
5.1	Kommunikationsobjekte nach CiA 301 .....	35
5.2	Herstellerspezifische Objekte .....	38

## Zu diesem Dokument

# 1 Zu diesem Dokument

## 1.1 Gültigkeit dieses Dokuments

Dieses Dokument beschreibt:

- Kommunikation mit dem Antrieb über RS232
- Basisdienste der Kommunikationsstruktur
- Methoden für den Parameterzugriff
- Antrieb aus Kommunikationssicht

Dieses Dokument richtet sich an Softwareentwickler und Projekt Ingenieure mit Schnittstellen-Erfahrung.

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

Diese Handbücher können im PDF-Format von der Internetseite [www.faulhaber.com/manuals](http://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.

## Zu diesem Dokument

### 1.4 Abkürzungsverzeichnis

Abkürzung	Bedeutung
Attr.	Attribut
CAN	Controller Area Network
CiA	CAN in Automation e.V.
COB ID	Communication Object Identifier
CRC	Cyclic Redundancy Check
CS	Command Specifier
EEPROM	Electrically Erasable Programmable Read-Only Memory
EMCY	Emergency
FIFO	First In – First Out
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
MOSFET	Metal-Oxide Semiconductor Field-Effect Transistor
MSB	Most Significant Byte
OD	Objektverzeichnis
PP	Profile Position
PV	Profile Velocity
ro	read only
RTR	Remote Request
rw	read-write
SDO	Servicedatenobjekt
PLC	Speicherprogrammierbare Steuerung
Sxx	Datentyp Signed (negative und positive Zahlen) mit Bitgröße xx
Uxx	Datentyp Unsigned (positive Zahlen) mit Bitgröße xx

## Zu diesem Dokument

---

### 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 des Motion Controllers

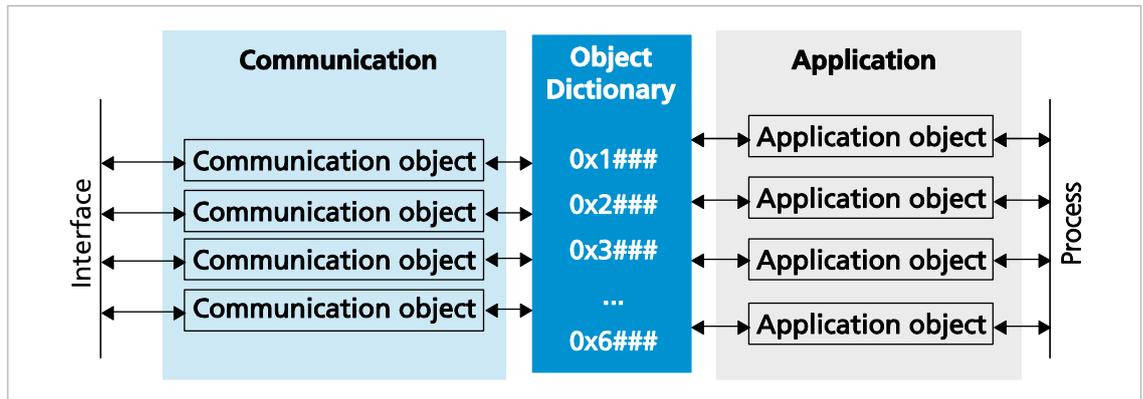


Abb. 1: Grundaufbau des Motion Controllers

#### Kommunikationsdienste

Der Master kommuniziert über die Schnittstelle und unter Verwendung der Kommunikationsdienste mit dem Objektverzeichnis (siehe Kap. 3.2, S. 17). Die Kommunikationsdienste sind angelehnt an die Systematik von CANopen-Geräten.

#### 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.

## Überblick

### 2.2 Voraussetzung für die Kommunikation

Die FAULHABER-Antriebe werden mit der Knotennummer 0xFF (unkonfiguriert) und einer RS232-Übertragungsrate von 115 200 Bit/s ausgeliefert. Für den Betrieb an einer USB- oder RS232-Schnittstelle müssen bei der Erstinbetriebnahme eine eindeutige Knotennummer und für RS232 zusätzlich eine passende Baudrate eingestellt werden.

Der Motion Controller verwendet über USB und RS232 dasselbe Kommunikationsprotokoll.

Bei einer Änderung der Knotennummer bzw. der Baudrate muss die Antwort auf die alte Knotennummer bzw. Baudrate stattfinden.

#### 2.2.1 Betrieb über die RS232-Schnittstelle

##### 2.2.1.1 Betrieb eines einzelnen Motion Controllers

1. Verbindung mit einer Host-Schnittstelle (typischerweise PC oder SPS) herstellen.

- Tx-Datenleitung der Host-Seite mit Rx-Pin des Antriebs verbinden
- Rx-Datenleitung der Host-Seite mit Tx-Pin des Antriebs verbinden (Nullmodemleitung)



Alternativ kann auf der PC-Seite auch ein USB/RS232-Umsetzer verwendet werden.

2. Host-Schnittstelle entsprechend den Antriebseinstellungen konfigurieren (siehe Kap. 4, S. 34):

- Gleiche Baudrate
- 8 Datenbits, keine Parität, 1 Stopbit, keine Flusststeuerung

3. Motion Controller einschalten.

☞ Die Kommunikation wird aufgebaut. Der Antrieb meldet sich auf der zuletzt eingestellten Baudrate mit einer Bootup-Message.

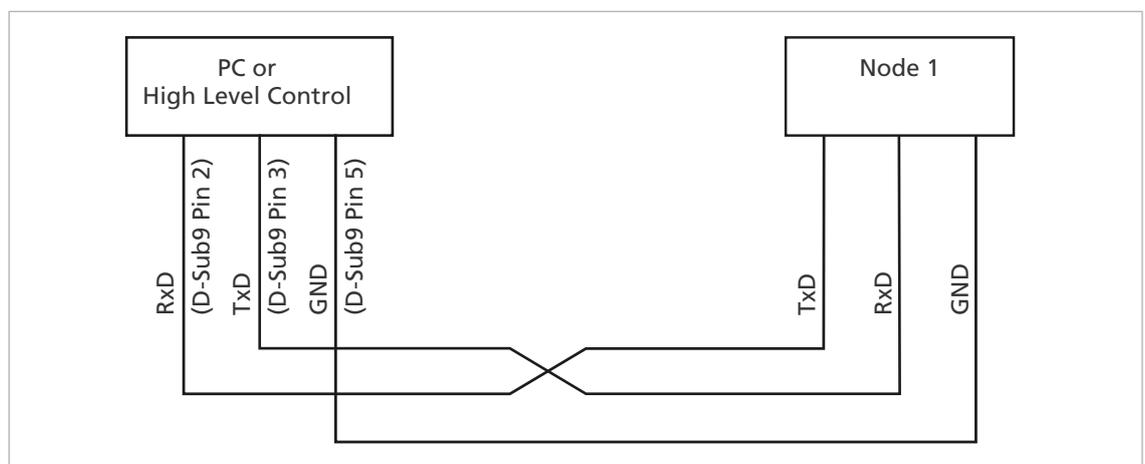


Abb. 2: Verdrahtung zwischen PC/Steuerung und einem Antrieb

## Überblick

### 2.2.1.2 RS232-Netzwerkbetrieb

**i** Der RS232-Netzwerkbetrieb ist für die Produkte MC 3001 B/P nicht möglich.

Mehrere Motion Controller können an einer einzigen RS232-Host-Schnittstelle betrieben werden.

- ▶ Tx-Leitungen und Rx-Leitungen der Controller parallel schalten.

**i** Im Netzwerkbetrieb dürfen die Antriebe keine asynchronen Nachrichten senden, da diese die Kommunikation mit einem anderen Antrieb stören können. Im Objekt 0x2400.04 können asynchrone Antworten deaktiviert werden.

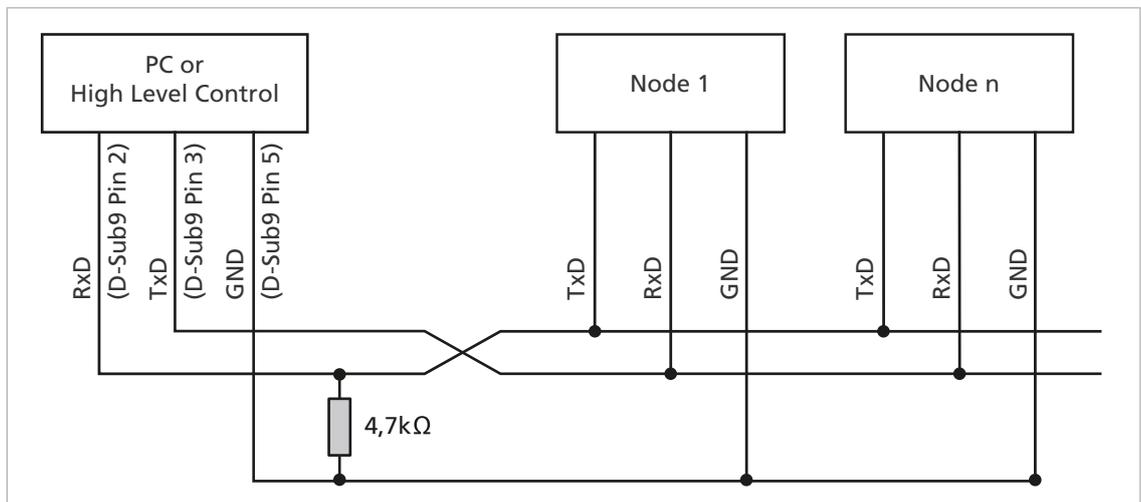


Abb. 3: Verdrahtung mit mehreren Motion Control Systemen im RS232-Netzwerkbetrieb

### 2.2.2 Betrieb über die USB-Schnittstelle

**i** Die USB-Schnittstelle kann zur Verbindung mit dem Motion Manager verwendet werden. Der passende Treiber wird mit dem Motion Manager automatisch installiert.

- ✓ USB-Kabel mit Mini-USB-Stecker ist vorhanden
- 1. Verbindung mit einer Host-Schnittstelle (typischerweise PC) herstellen.
  - ↗ USB-Kabel mit Mini USB-Stecker auf der Geräteseite verwenden.
- 2. Motion Controller einschalten.
  - ↗ Die Kommunikation wird aufgebaut.  
Der Antrieb meldet sich mit einer Bootup-Message.

**i** Für die Verwendung der USB-Schnittstelle in einer eigenen Anwendung wird eine passende Treiberanbindung benötigt. Informationen hierüber erhalten Sie auf Anfrage.

### 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.

## Überblick

### 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. 35). Beim Einschalten des Motors werden die Parameter automatisch aus dem nicht-flüchtigen Speicher in den flüchtigen Speicher (RAM) geladen.

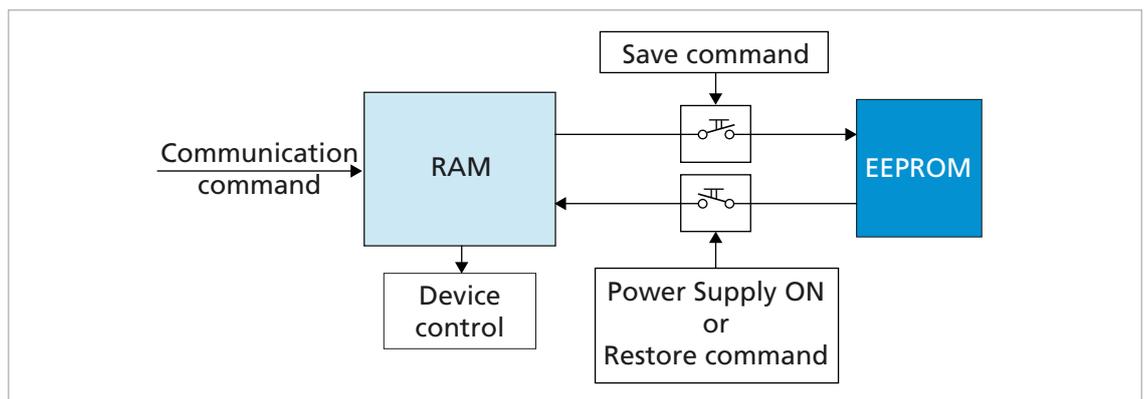


Abb. 4: Parameter speichern und wiederherstellen

Folgende Parameter können mit dem Restore-Befehl geladen werden (siehe Kap. 5.1, S. 35):

- 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. 38).

- ▶ Auf Subindex 01 bis 05 des Objekts 0x1010 die Signatur "save" schreiben (siehe Tab. 39).

# Überblick

## 2.4.2 Einstellungen wiederherstellen

**i** 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. 40).

1. Auf Subindex 01 bis 06 des Objekts 0x1011 die Signatur "load" schreiben (siehe Tab. 41).
  - ↪ 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.

**i** 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 [ $As \cdot 1e^{-6}$ ]
	0x02	Integral Time TN	U16	rw	Reglernachstellzeit [100 $\mu s$ ]
0x2346	0x01	Set Point Velocity Filter Time $T_F$	U16	rw	Filterzeit $T_F$ [100 $\mu s$ ]
	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

## Überblick

### 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 <sup>2</sup> ]
0x6084	0x00	Profile Deceleration	U32	rw	Profile Deceleration [1/s <sup>2</sup> ]
0x6086	0x00	Motion Profile Type	S16	rw	Bewegungsprofiltyp: 0: Lineares Profil 1: Sin <sup>2</sup> Geschwindigkeit

## Überblick

Index	Subindex	Name	Typ	Attr.	Bedeutung
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.

# Protokollbeschreibung

## 3 Protokollbeschreibung

### 3.1 Einführung

Über Protokolldienste können Einträge im Objektverzeichnis geschrieben und gelesen werden.

Die für die RS232- und USB-Schnittstelle definierten Dienste lehnen sich an die CANopen-Dienste an, sind jedoch auf die Eigenschaften der RS232-Schnittstelle angepasst.

Die CiA (CAN in Automation) definiert in der CiA 301 folgende Aspekte:

- Kommunikationsstruktur
- Steuer- und Überwachungsfunktionen

Für eine Reihe von Geräteklassen sind in der CiA 402 CANopen-Geräteprofile für Antriebe definiert.

Der gleichzeitige Zugriff auf den Antrieb sowohl von RS232 Seite auch über die USB Schnittstelle wird unterstützt. Die über beide Schnittstellen eintreffenden Nachrichten werden intern in einer gemeinsamen Warteschlange abgelegt und sequentiell bearbeitet (FIFO). Die Quittierung durch den Antrieb erfolgt über die gleiche Schnittstelle.

**i** Jede Anfrage vom Host wird mit einer Quittierung durch den Antrieb abgeschlossen. Die Maximalzahl der im Antrieb gepufferten Anfragen ist begrenzt. Wenn keine weiteren Anfragen in die Warteschlange eingereicht werden können, wird eine entsprechende Meldung versandt und die Anfrage verworfen.

#### Telegrammaufbau

Für die Kommunikation über die USB- und RS232-Schnittstelle wird ein binäres Protokoll mit Botschaften variabler Länge verwendet.

Tab. 1: Schematischer Aufbau eines USB/RS232-Telegramms

Byte	Name	Bedeutung
1. Byte	SOF	Zeichen (S) als Start of Frame
2. Byte	Nutzdatenlänge	Telegrammlänge ohne SOF/EOF (Paketlänge)
3. Byte	Knotennummer	Knotennummer des Slaves (0 = Broadcast)
4. Byte	Befehlscode	Siehe Tab. 2
5. – N. Byte	Daten	Datenbereich (Länge = Paketlänge – 4)
(N+1). Byte	CRC	CRC8 mit Polynom 0xD5 über Byte 2–N
(N+2). Byte	EOF	Zeichen (E) als End of Frame

Telegrammfehler (z. B. falsche CRC, falsche Länge, ungültige Befehlscodes) werden nicht rückgemeldet. Die Knotennummer der empfangenen Nachricht, insbesondere bei CRC-Fehlern, kann nicht eindeutig bestimmt werden. Das Telegramm bleibt unbeantwortet und der Master muss nach Time-Out eine erneute Anfrage senden.

In der nachfolgenden Beschreibung sind die Zeichen SOF und EOF nicht mehr aufgeführt. Hier wird nur auf die zwischenliegenden Bytes eingegangen und Byte 1 entspricht dem 2. Byte im Gesamt-Telegramm-Frame.

#### Framelänge

Die Länge des gesamten Frames inkl. SOF und EOF ist die Nutzdatenlänge + 2 Byte. Die Nutzdatenlänge ist auf 62 Byte begrenzt.

# Protokollbeschreibung

## Befehlscodes

Tab. 2: Funktionen der Befehlscodes

Befehlscode	Name	Funktion
0x00	Bootup	Bootup-Nachricht / Reset Node (Receive / Request)
0x01	SDORead	Lesen von Objektverzeichnis-Eintrag (Request / Response)
0x02	SDOWrite	Schreiben eines Objektverzeichnis-Eintrags (Request / Response)
0x03	SDOError	SDO-Fehler (Abort Request / Error Response)
0x04	Controlword	Schreiben des Controlword (Request / Response)
0x05	Statusword	Empfang des Statusword (Receive)
0x06	TraceLog	Trace-Request für Trace-Logger (Request / Response)
0x07	Emcy	Empfang einer Emergency-Meldung (Receive)
0x08	SDOBlockReadInit	Initialisierung des Segmented SDO Block Upload (Request / Response)
0x09	SDOBlockReadUpload	Durchführung des Segmented SDO Block Upload (Request / Response / Acknowledge)
0x0A	SDOBlockReadEnd	Ende des Segmented SDO Block Upload (Request / Acknowledge)
0x0B	SDOBlockWriteInit	Initialisierung des Segmented SDO Block Download (Request / Response)
0x0C	SDOBlockWriteDownload	Durchführung des Segmented SDO Block Download (Request / Response)
0x0D	SDOBlockWriteEnd	Ende des Segmented SDO Block Download (Request / Response)

## Daten

Das Datenformat ist an das CANopen-Datenformat angelehnt. Die Datenübertragung beginnt immer mit dem niedrigwertigsten Byte.

### CRC (Cyclic Redundancy Check)

Für die Checksummenberechnung wird folgender Algorithmus über alle Bytes (ohne SOF/EOF) des zu übertragenden Telegramms verwendet:

```
#define Polynom 0xD5

uint8_t CalcCRCByte(uint8_t u8Byte, uint8_t u8CRC)
{
    uint8_t i;
    u8CRC = u8CRC ^ u8Byte;
    for (i = 0; i < 8; i++) {
        if (u8CRC & 0x01) {
            u8CRC = (u8CRC >> 1) ^ Polynom;
        }
        else {
            u8CRC >>= 1;
        }
    }
    return u8CRC;
}
```

Als Startwert für die CRC8 wird 0xFF verwendet.

# Protokollbeschreibung

## 3.2 Kommunikationsdienste

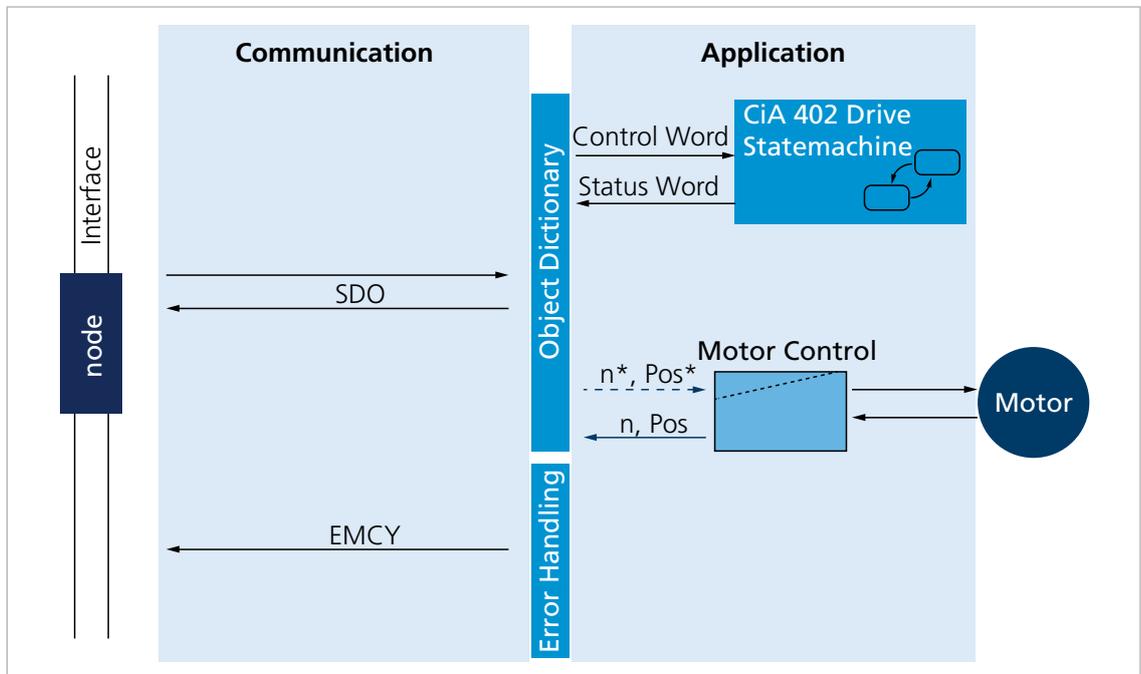


Abb. 5: Kommunikationsdienste des Motion Controllers

Folgende Kommunikationsdienste sind vorhanden:

- Bootup-Nachricht
- Schreib- bzw. Lesedienst auf je einen einzelnen Parameter (SDO-Nachricht)
- Direkter Schreibzugriff auf das Controlword des Antriebs
- Direkter Lesezugriff auf das Statusword des Antriebs
- Kommunikationsdienst zur Signalisierung von Fehlerzuständen durch eine im Fehlerfall vom Antrieb übertragene Nachricht (EMCY)
- Kommunikationsdienst zum Zugriff auf Werte des Datenloggers (Trace)

## Protokollbeschreibung

### 3.3 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. 3: Einteilung der SDO-Übertragungsarten

Übertragungsart	Byteanzahl	Verwendungszweck
Expedited Transfer	maximal 4 Byte	Lesen und Beschreiben einzelner numerischer Parameter
Segmented Transfer	mehr als 58 Byte	Lesen von Text-Parametern (z. B. Gerätename, Firmware-Version) und Übertragung von Datenblöcken (z. B. Trace-Puffer)

#### 3.3.1 Expedited Transfer

##### 3.3.1.1 Objektverzeichnis auslesen

Über den SDO-Read-Dienst können Einträge im Objektverzeichnis ausgelesen werden. Die Telegramme werden grundsätzlich quittiert.

Tab. 4: Request

Byte	Inhalt	Beschreibung
1	7	Nutzdatenlänge 7 Byte
2	Knotennummer	Knotennummer
3	0x01	Befehl SDORead
4	Index LB	Index des Objekteintrags LB
5	Index HB	Index des Objekteintrags HB
6	Subindex	Subindex des Objekteintrags
7	CRC	Prüfsumme

Tab. 5: Response

Byte	Inhalt	Beschreibung
1	Länge	Nutzdatenlänge > 7 Byte
2	Knotennummer	Knotennummer
3	0x01	Befehl SDORead
4	Index LB	Index des Objekteintrags LB
5	Index HB	Index des Objekteintrags HB
6	Subindex	Subindex des Objekteintrags
7–N	Wert	Aktueller Wert des angegebenen Objekteintrags
(N+1)	CRC	Prüfsumme

Kann das angegebene Objekt nicht ausgelesen werden, wird als Response ein SDO-Fehler gemäß CiA301 gemeldet (siehe Kap. 3.3.3, S. 26).

## Protokollbeschreibung

### 3.3.1.2 Objektverzeichnis beschreiben

Über den SDO-Write-Dienst können Einträge im Objektverzeichnis beschrieben werden. Die Telegramme werden grundsätzlich quittiert.

Tab. 6: Request

Byte	Inhalt	Beschreibung
1	Länge	Nutzdatenlänge > 7 Byte
2	Knotennummer	Knotennummer
3	0x02	Befehl SDOWrite
4	Index LB	Index des Objekteintrags LB
5	Index HB	Index des Objekteintrags HB
6	Subindex	Subindex des Objekteintrags
7–N	Wert	Neuer Wert für den angegebenen Objekteintrag
(N+1)	CRC	Prüfsumme

Tab. 7: Response

Byte	Inhalt	Beschreibung
1	7	Nutzdatenlänge 7 Byte
2	Knotennummer	Knotennummer
3	0x02	Befehl SDOWrite
4	Index LB	Index des Objekteintrags LB
5	Index HB	Index des Objekteintrags HB
6	Subindex	Subindex des Objekteintrags
7	CRC	Prüfsumme

Kann das angegebene Objekt nicht beschrieben werden, wird als Antwort ein SDO-Fehler gemäß CiA301 gemeldet (siehe Kap. 3.3.3, S. 26).

# Protokollbeschreibung

## 3.3.2 Segmented Transfer

### 3.3.2.1 SDO Block Upload

Das Segmented SDO Block Upload Protokoll ist an die CiA301 angelehnt.

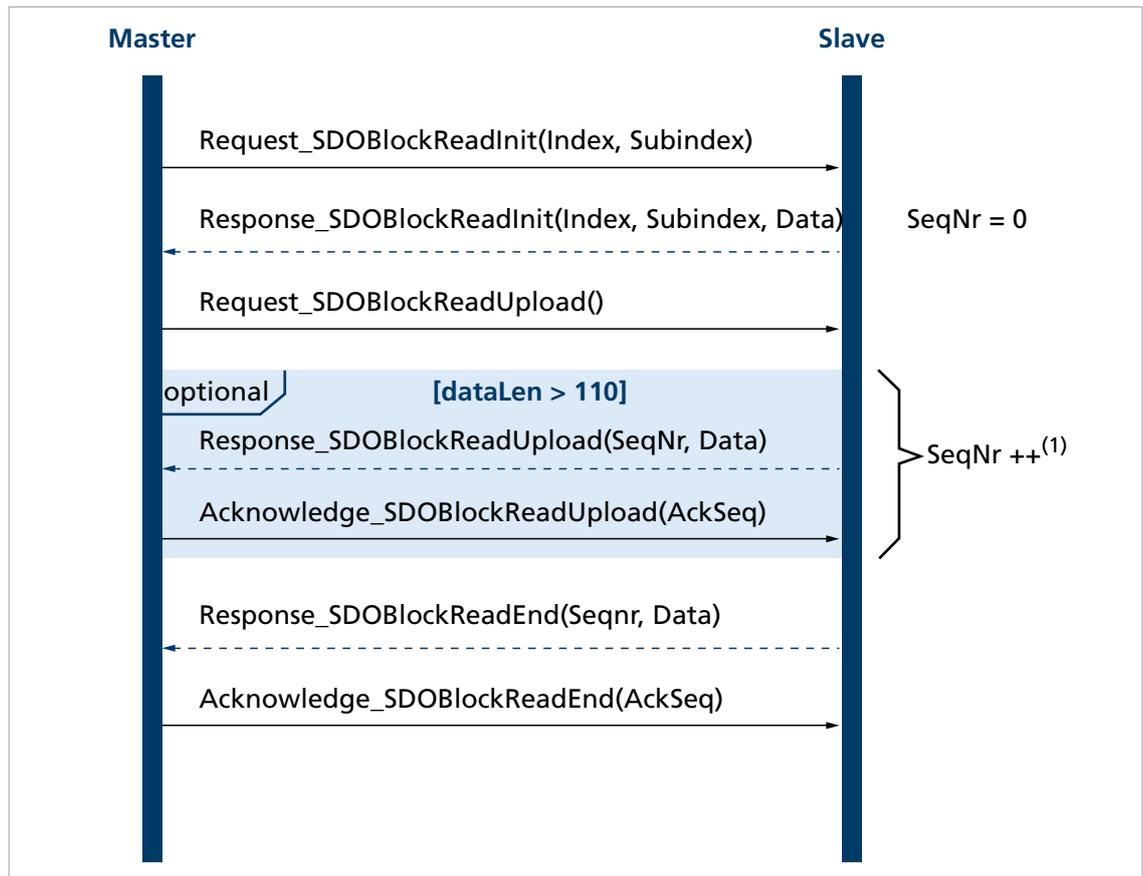


Abb. 6: Sequenzdiagramm SDO Block Upload

1) So oft wiederholen, bis nur noch weniger als 58 Byte zu übertragen sind

Tab. 8: Request SDO Initiate Block Upload (Master to Slave)

Byte	Inhalt	Beschreibung
1	7	Nutzdatenlänge 7 Byte
2	Knotennummer	Knotennummer
3	0x08	Befehl SDOBlockReadInit
4	Index LB	Index des Objekteintrags LB
5	Index HB	Index des Objekteintrags HB
6	Subindex	Subindex des Objekteintrags
7	CRC	Prüfsumme

## Protokollbeschreibung

Tab. 9: Response SDO Initiate Block Upload (Slave to Master)

Byte	Inhalt	Beschreibung
1	Länge	Nutzdatenlänge > 7 Byte
2	Knotennummer	Knotennummer
3	0x08	Befehl SDOBlockReadInit
4	Index LB	Index des Objekteintrags LB
5	Index HB	Index des Objekteintrags HB
6	Subindex	Subindex des Objekteintrags
7	Datenlänge LB	Gesamtlänge der zu übertragenden Daten in Byte LB
8	Datenlänge HB	Gesamtlänge der zu übertragenden Daten in Byte HB
9–N	Daten	Daten des ersten Segments (max. 53 Byte)
(N+1)	CRC	Prüfsumme

Kann das angegebene Objekt nicht ausgelesen werden, wird als Response ein SDO-Fehler gemäß CiA301 gemeldet (siehe Kap. 3.3.3, S. 26).

Tab. 10: Request SDO Block Upload (Master to Slave)

Byte	Inhalt	Beschreibung
1	4	Nutzdatenlänge 4 Byte
2	Knotennummer	Knotennummer
3	0x09	Befehl SDOBlockReadUpload
4	CRC	Prüfsumme

Tab. 11: Response SDO Block Upload (Slave to Master)

Byte	Inhalt	Beschreibung
1	Länge	Nutzdatenlänge > 5 Byte
2	Knotennummer	Knotennummer
3	0x09	Befehl SDOBlockReadUpload
4	SeqNr	Sequenznummer beginnend mit 1
5–N	Data	Daten des jeweiligen Segments (max. 57 Byte)
(N+1)	CRC	Prüfsumme

Tab. 12: Acknowledge SDO Block Upload (Master to Slave)

Byte	Inhalt	Beschreibung
1	5	Nutzdatenlänge 5 Byte
2	Knotennummer	Knotennummer
3	0x09	Befehl SDOBlockReadUpload
4	AckSeq	Empfangene Sequenznummer
5	CRC	Prüfsumme

Wenn die zu übertragende Datenlänge größer als 110 Byte ist (max. Datenlänge SDOBlockReadInit + max. Datenlänge SDOBlockReadEnd), wird das SDOBlockReadUpload-Telegramm so oft hintereinander mit fortlaufender Sequenznummer vom Slave gesendet und vom Master quittiert, bis der komplette Block übertragen wurde. Das letzte Blocksegment wird mit dem Befehl „SDOBlockReadEnd“ gekennzeichnet.

## Protokollbeschreibung

Tab. 13: Response SDO Block Upload End (Slave to Master)

Byte	Inhalt	Beschreibung
1	Länge	Nutzdatenlänge > 5 Byte
2	Knotennummer	Knotennummer
3	0x0A	Befehl SDOBlockReadEnd
4	SeqNr	Sequenznummer des letzten Segments (>0)
5–N	Data	Daten des letzten Segments (max. 57 Byte)
(N+1)	CRC	Prüfsumme

Tab. 14: Acknowledge SDO Block Upload End (Master to Slave)

Byte	Inhalt	Beschreibung
1	5	Nutzdatenlänge 5 Byte
2	Knotennummer	Knotennummer
3	0x0A	Befehl SDOBlockReadEnd
4	AckSeq	Letzte empfangene Sequenznummer
5	CRC	Prüfsumme

- Wenn AckSeq = 0 ist, wurde das übertragene Segment nicht korrekt empfangen und das Segment muss erneut übertragen werden.
- Bei einem Verarbeitungsfehler auf Controllerseite wird eine SDO-Error-Response gemäß Kap. 3.3.3, S. 26 versendet (z. B. Timeout).
- Ein Abbruch der Block-Übertragung durch den Master wird über ein Abort-SDO-Transfer-Telegramm signalisiert (siehe Kap. 3.3.3, S. 26).

## Protokollbeschreibung

### 3.3.2.2 SDO Block Download

Das Segmented SDO Block Download Protokoll ist an die CiA301 angelehnt.

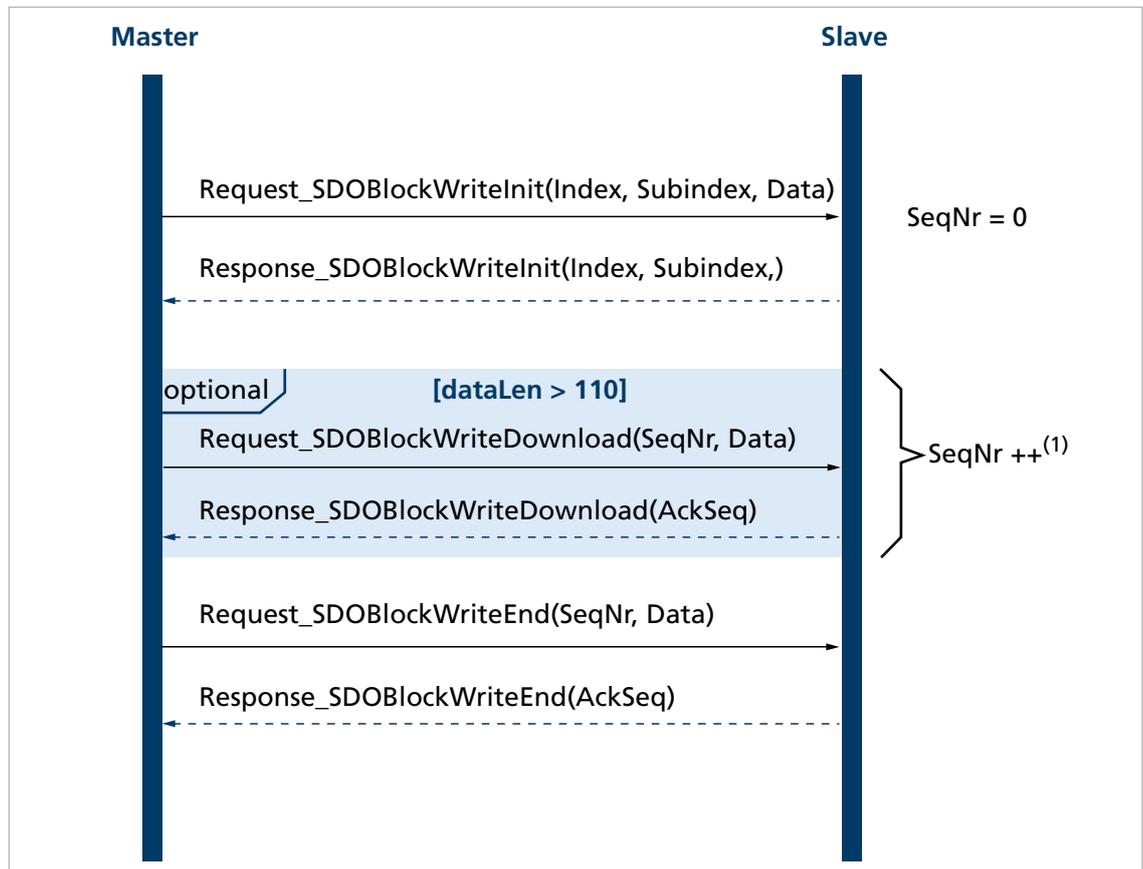


Abb. 7: Sequenzdiagramm SDO Block Download

1) So oft wiederholen, bis nur noch weniger als 58 Byte zu übertragen sind

Tab. 15: Request SDO Initiate Block Download (Master to Slave)

Byte	Inhalt	Beschreibung
1	Länge	Nutzdatenlänge >7 Byte
2	Knotennummer	Knotennummer
3	0x0B	Befehl SDOBlockWriteInit
4	Index LB	Index des Objekteintrags LB
5	Index HB	Index des Objekteintrags HB
6	Subindex	Subindex des Objekteintrags
7	Datenlänge LB	Gesamtlänge der zu übertragenden Daten in Byte LB
8	Datenlänge HB	Gesamtlänge der zu übertragenden Daten in Byte HB
9–N	Daten	Daten des ersten Segments (max. 53 Byte)
(N+1)	CRC	Prüfsumme

## Protokollbeschreibung

Tab. 16: Response SDO Initiate Block Download (Slave to Master)

Byte	Inhalt	Beschreibung
1	7	Nutzdatenlänge 7 Byte
2	Knotennummer	Knotennummer
3	0x0B	Befehl SDOBlockWriteInit
4	Index LB	Index des Objekteintrags LB
5	Index HB	Index des Objekteintrags HB
6	Subindex	Subindex des Objekteintrags
7	CRC	Prüfsumme

Tab. 17: Request SDO Block Download (Master to Slave)

Byte	Inhalt	Beschreibung
1	Länge	Nutzdatenlänge > 5 Byte
2	Knotennummer	Knotennummer
3	0x0C	Befehl SDOBlockWriteDownload
4	SeqNr	Sequenznummer beginnend mit 1
5–N	Data	Daten des jeweiligen Segments (max. 57 Byte)
(N+1)	CRC	Prüfsumme

Tab. 18: Response SDO Block Download (Slave to Master)

Byte	Inhalt	Beschreibung
1	Länge	Nutzdatenlänge > 5 Byte
2	Knotennummer	Knotennummer
3	0x0C	Befehl SDOBlockWriteDownload
4	AckSeq	Empfangene Sequenznummer
5	CRC	Prüfsumme

Das SDOBlockWriteDownload-Telegramm wird so oft hintereinander mit fortlaufender Sequenznummer vom Master gesendet und vom Slave quittiert, bis der komplette Block übertragen wurde. Das letzte Block-Segment wird mit dem Befehl „SDOBlockWriteEnd“ gekennzeichnet.

Tab. 19: Request SDO Block Download End (Master to Slave)

Byte	Inhalt	Beschreibung
1	Länge	Nutzdatenlänge > 5 Byte
2	Knotennummer	Knotennummer
3	0x0D	Befehl SDOBlockWriteEnd
4	SeqNr	Sequenznummer des letzten Segments
5–N	Data	Daten des jeweiligen Segments (max. 57 Byte)
(N+1)	CRC	Prüfsumme

## Protokollbeschreibung

Tab. 20: Response SDO Block Download End (Slave to Master)

Byte	Inhalt	Beschreibung
1	5	Nutzdatenlänge 5 Byte
2	Knotennummer	Knotennummer
3	0x0D	Befehl SDOBlockWriteEnd
4	AckSeq	Empfangene Sequenznummer
5	CRC	Prüfsumme

- Wenn AckSeq = 0 ist, wurde das übertragene Segment nicht korrekt empfangen und das Segment muss erneut übertragen werden.
- Bei einem Verarbeitungsfehler auf Controllerseite wird eine SDO-Error-Response gemäß Kap. 3.3.1.2, S. 19 versendet (z. B. Timeout).
- Ein Abbruch der Block-Übertragung durch den Master wird über ein Abort-SDO-Transfer-Telegramm signalisiert (siehe Kap. 3.3.3, S. 26).

Tab. 21: Response

Byte	Inhalt	Beschreibung
1	7	Nutzdatenlänge 7 Byte
2	Knotennummer	Knotennummer
3	0x03	Befehl SDOError
4	Index LB	Index des Objekteintrags LB
5	Index HB	Index des Objekteintrags HB
6	Subindex	Subindex des Objekteintrags
7	CRC	Prüfsumme

## Protokollbeschreibung

### 3.3.3 SDO-Fehlerbehandlung

#### SDO-Fehlermeldung

Tab. 22: Error-Response

Byte	Inhalt	Beschreibung
1	11	Nutzdatenlänge 11 Byte
2	Knotennummer	Knotennummer
3	0x03	Befehl SDOError
4	Index LB	Index des Objekteintrags LB
5	Index HB	Index des Objekteintrags HB
6	Subindex	Subindex des Objekteintrags
7	Error0	Zusatzfehlercode LB (siehe Tab. 24)
8	Error1	Zusatzfehlercode HB (siehe Tab. 24)
9	Error2	Fehlercode (siehe Tab. 24)
10	Error3	Fehlerklasse (siehe Tab. 24)
11	CRC	Prüfsumme

#### Abbruch einer Block-Übertragung durch den Master (Abort-SDO-Telegramm)

Tab. 23: Request Abort SDO Transfer

Byte	Inhalt	Beschreibung
1	11	Nutzdatenlänge 11 Byte
2	Knotennummer	Knotennummer
3	0x03	Befehl SDOError
4	Index LB	Index des Objekteintrags LB
5	Index HB	Index des Objekteintrags HB
6	Subindex	Subindex des Objekteintrags
7	Error0	Zusatzfehlercode LB (siehe Tab. 24)
8	Error1	Zusatzfehlercode HB (siehe Tab. 24)
9	Error2	Fehlercode (siehe Tab. 24)
10	Error3	Fehlerklasse (siehe Tab. 24)
11	CRC	Prüfsumme

## Protokollbeschreibung

### Fehler-Bytecodierung

Bei Abbruch des SDO-Protokolls im Fehlerfall werden die Bytes 7 bis 10 (Error0 bis Error3) codiert.

Tab. 24: Fehler-Bytecodierung

Fehler-klasse	Fehlercode	Zusatz-code	Beschreibung
0x05	0x04	0x0001	SDO-Command 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	0x0043	Allgemeine Parameter-Inkompatibilität
0x06	0x04	0x0047	Allgemeiner interner Inkompatibilitätsfehler 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

## Protokollbeschreibung

### 3.4 Emergency-Objekt (Fehlermeldung)

Das Emergency-Objekt informiert asynchron den Master über Fehler und muss nicht abgefragt werden. Das Emergency-Objekt ist immer 12 Byte groß (ohne SOF und EOF). Im RS232-Netzwerkbetrieb darf die Emergency-Nachricht nicht übertragen werden.

Tab. 25: Nutzdatenbelegung des Emergency-Telegramms

Byte	Inhalt	Beschreibung
1	12	Nutzdatenlänge 12 Byte
2	Knotennummer	Knotennummer
3	0x07	Befehl EMCY
4	Error0	Fehlercode LB
5	Error1	Fehlercode HB
6	Error-Reg	Fehler-Register (Inhalt von Objekt 0x1001)
7	Manuf. Spec. Error0	FAULHABER Fehler-Register LB (Inhalt von Objekt 0x2320)
8	Manuf. Spec. Error1	FAULHABER Fehler-Register HB
9	Manuf. Spec. Error2	Reserviert (0)
10	Manuf. Spec. Error3	Reserviert (0)
11	Manuf. Spec. Error4	Reserviert (0)
12	CRC	Prüfsumme

Belegung der Nutzdaten:

- Error0(LB)/Error1(HB): 16-Bit-Error-Code
- Error-Reg: Error-Register (Inhalt von Objekt 0x1001, siehe Kap. 5.1, S. 35)
- FE0(LB)/FE1(HB): 16-Bit FAULHABER Fehlerregister (Inhalt von Objekt 0x2320, siehe Tab. 33)
- Bytes 9 bis 11: 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. 26 listet alle Fehler auf, die über Emergency-Nachrichten gemeldet werden, sofern der entsprechende Fehler in der Emergency-Mask für das FAULHABER Fehlerregister gesetzt ist (Kap. 3.7.1, S. 32).

# Protokollbeschreibung

Tab. 26: 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 <sup>a)</sup>
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
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	CAN Overrun	0x0800	11	ComError	4	Kommunikations-Fehler
0x8130	CAN Guarding Failed					
0x8140	CAN Recovered From Bus-Off					
0x8310	RS232 overrun					
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.

# Protokollbeschreibung

**Beispiel:**

Eine Emergency-Nachricht mit der Nutzdatenbelegung in Tab. 27 wird in folgendem Fall versendet:

- In der Error Mask 0x2321 ist unter Subindex 1 (Emergency Mask) Bit 1 (Schleppfehler) gesetzt (siehe Tab. 35).
- 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. 27: Beispielhafte Nutzdatenbelegung einer Emergency-Nachricht

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

## 3.5 Gerätesteuerung

### 3.5.1 BootUp-Nachricht

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.

 Im RS232-Netzwerkbetrieb darf die BootUp-Nachricht nicht versendet werden.

Tab. 28: Aufbau einer Boot-Up-Nachricht

Byte	Inhalt	Beschreibung
1	Länge	Nutzdatenlänge > 4 Byte
2	Knotennummer	Knotennummer
3	0x00	Befehl BootUp
4-N	Device Name	Gerätenamen als BootUp-Message
(N+1)	CRC	Prüfsumme

### 3.5.2 Reset Node

Ein Soft-Reset kann vom Master über folgendes Telegramm ausgeführt werden:

Tab. 29: Aufbau einer Reset-Node-Nachricht

Byte	Inhalt	Beschreibung
1	4	Nutzdatenlänge 4 Byte
2	Knotennummer	Knotennummer
3	0x00	Befehl BootUp
4	CRC	Prüfsumme

## Protokollbeschreibung

### 3.5.3 Device Control

Über Device Control können Zustandsänderungen eingeleitet und aktuelle Zustände ausgelesen werden.

Tab. 30: Request Write Controlword (Objekt 0x6040.00 im Objektverzeichnis)

Byte	Inhalt	Beschreibung
1	6	Nutzdatenlänge 6 Byte
2	Knotennummer	Knotennummer
3	0x04	Befehl Controlword
4	Controlword LB	Neuer Controlword-Wert gemäß CiA402
5	Controlword HB	Neuer Controlword-Wert gemäß CiA402
6	CRC	Prüfsumme

Tab. 31: Response

Byte	Inhalt	Beschreibung
1	5	Nutzdatenlänge 5 Byte
2	Knotennummer	Knotennummer
3	0x04	Befehl Controlword
4	Error	Fehlercode: 0 = Ok
5	CRC	Prüfsumme

Das Statusword wird bei einer Zustandsänderung asynchron vom Antrieb versendet und kann nicht direkt abgefragt werden (hierfür kann der Befehl SDORead verwendet werden).

Tab. 32: Receive Statusword (Objekt 0x6041.00 im Objektverzeichnis)

Byte	Inhalt	Beschreibung
1	6	Nutzdatenlänge 6 Byte
2	Knotennummer	Knotennummer
3	0x05	Befehl Statusword
4	Statusword LB	Aktueller Wert des Statuswords gemäß CiA402
5	Statusword HB	Aktueller Wert des Statuswords gemäß CiA402
6	CRC	Prüfsumme

## 3.6 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. 35)
- Herstellerspezifischer Bereich (Index 0x2000 bis 0x5FFF, enthält herstellerspezifische Objekte, siehe Kap. 5.2, S. 38)
- Standardisierte Geräteprofile (0x6000 bis 0x9FFF, enthält die vom Motion Controller unterstützten Objekte, siehe Dokumentation der Antriebsfunktionen)

# Protokollbeschreibung

## 3.7 Fehlerbehandlung

### 3.7.1 Gerätefehler

Tab. 33: 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. 34: 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. 3.4, S. 28).

Die Error Mask beschreibt die Behandlung interner Fehler entsprechend der Fehlercodierung (siehe Tab. 34).

## Protokollbeschreibung

Tab. 35: 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 <i>Fault Reaction Active</i> 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 <i>Quick Stop Active</i> 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.

# Kommunikationseinstellungen

## 4 Kommunikationseinstellungen

- Die Knotennummern 1 bis 127 können eingestellt werden.
- Eine RS232-Übertragungsrate gemäß Tab. 36 kann durch Angabe des Index 0 bis 3 eingestellt werden.
- USB benötigt keine Angabe der Übertragungsrate

Tab. 36: RS232-Bit-Timing-Parameter

Baudrate	Index
9 600 Bit/s	0
19 200 Bit/s	1
57 600 Bit/s	2
115 200 Bit/s	3

Die Kommunikationsparameter werden durch Beschreiben folgender Objekte im Objektverzeichnis eingestellt.

Tab. 37: Baudrate Index und Knotennummer

Index	Subindex	Name	Typ	Attr.	Standardwert	Bedeutung
0x2400	0x00	Number of Entries	U8	ro	8	Anzahl Objekteinträge
	0x02	RS232 Rate	U8	rw	3	Index der Baudrate gemäß Tab. 36
	0x03	Node ID	U8	rw	1	Knotennummer

Eine Änderung der Kommunikationsparameter wird mit der letzten Einstellung für die Baudrate bzw. die Knotennummer quittiert. Nach der Befehlsquittierung sind die neuen Einstellungen gültig. Erst nach einem Save-Befehl für Anwendungsparameter werden die geänderten Einstellungen dauerhaft übernommen und sind nach dem nächsten Einschalten verfügbar.

## Parameterbeschreibung

### 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. 3.4, S. 28 beschrieben.

Durch Schreiben einer 0 auf Subindex 0 wird der Fehlerspeicher gelöscht (siehe Tab. 34).

##### 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.

## Parameterbeschreibung

### 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.

### Store Parameters

Tab. 38: 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. 39: 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.

## Parameterbeschreibung

### Restore Default Parameters

Tab. 40: 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 Werkzeugeinstellungen wiederherstellen
	0x02	Restore Comm Default Parameters	U32	rw	1	Werkzeugeinstellungen für Kommunikations-Parameter (0x0000 bis 0x1FFF) wiederherstellen
	0x03	Restore App Default Parameters	U32	rw	1	Werkzeugeinstellungen 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. 41: 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.

### 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	Herstellernummer (FAULHABER: 327)
	0x02	Product Code	U32	ro	48	Produktkennnummer
	0x03	Revision Number	U32	ro	-	Versionsnummer
	0x04	Serial Number	U32	ro	-	Seriennummer

## Parameterbeschreibung

### 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 <i>Fault Reaction Active</i> 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 <i>Quick Stop Active</i> 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

## Parameterbeschreibung

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

### RS232 Baudrate Index und Knotennummer

Index	Subindex	Name	Typ	Attr.	Standardwert	Bedeutung
0x2400	0x00	Number of Entries	U8	ro	8	Anzahl Objekteinträge
	0x02	RS232 Rate	U8	rw	3	Index der Baudrate gemäß Tab. 36
	0x03	Node ID	U8	rw	1	Knotennummer
	0x04	Communication Settings	U32	rw	0	Bitmaske für Kommunikationseinstellungen gemäß Tab. 42
	0x05	RS232 NetMode	U8	rw	0	Einstellung des RS232-Netzwerkbetriebs: <ul style="list-style-type: none"> <li>▪ 0: Netzwerkbetrieb nicht aktiv</li> <li>▪ 1: Netzwerkbetrieb aktiv</li> </ul>
	0x06	ComState	U16	ro	0	Bitmaske für Kommunikationsstatus gemäß Tab. 43

Tab. 42: Bedeutung der Bits zu 0x2400.04 (Communication Settings)

Bit	Beschreibung
0	Transmit EMCY
1	AsyncDriveStatus
2...31	Reserved

Tab. 43: Bedeutung der Bits zu 0x2400.06 (ComState)

Bit	Beschreibung
0...6	Reserved
7	Transmit Overflow Signaled
8	BufferOverflow
9...14	Reserved
15	PDOLength

