

Programmierhandbuch

Sprache **Deutsch**
Original
Dokument-Nr. 5.02065.04
Artikel-Nr. 368694
Stand 02.05.2012

be in motion

be in motion



CANopen Slave

Programmierhandbuch

Version 03.12

D 5.02065.04

Vor Beginn aller Arbeiten Betriebsanleitung lesen!

Copyright	<p>Dieses Programmierhandbuch darf vom Eigentümer ausschließlich für den internen Gebrauch in beliebiger Anzahl kopiert werden. Für andere Zwecke darf dieses Programmierhandbuch auch auszugsweise weder kopiert noch vervielfältigt werden.</p> <p>Verwertung und Mitteilung von Inhalten dieses Programmierhandbuchs sind nicht gestattet. Bezeichnungen bzw. Unternehmenskennzeichen in diesem Programmierhandbuch können Marken sein, deren Benutzung durch Dritte für deren Zwecke die Rechte der Inhaber verletzen kann.</p>
Vorabinformation	<p>Achtung: Sofern das Ihnen vorliegende Dokument als Vorabinformation gekennzeichnet ist, gilt Folgendes:</p> <p>Bei dieser Version handelt es sich um technische Vorabinformationen, die die Anwender der beschriebenen Geräte und Funktionen frühzeitig erhalten sollen, um sich auf mögliche Änderungen bzw. funktionale Erweiterungen einstellen zu können.</p> <p>Diese Informationen sind als vorläufig zu verstehen, da diese noch nicht dem endgültigen Baumüller internen Review-Prozess unterzogen wurden. Insbesondere unterliegen diese Informationen noch Änderungen, so dass keine rechtliche Verbindlichkeit auf Grund von diesen Vorabinformationen hergeleitet werden kann. Baumüller übernimmt keine Haftung für Schäden, die sich aus dieser unter Umständen fehlerhaften oder unvollständigen Version ergeben können.</p> <p>Sollten Sie inhaltliche und / oder gravierende formale Fehler in dieser Vorabinformation erkennen oder vermuten, so bitten wir Sie, sich an den für Sie zuständigen Betreuer der Firma Baumüller zu wenden und uns über diese Mitarbeiter Ihre Erkenntnisse und Anmerkungen zukommen zu lassen, so dass Ihre Erkenntnisse und Anmerkungen beim Übergang von den Vorabinformationen zu den endgültigen (durch Baumüller gereviewten) Informationen berücksichtigt und ggf. eingepflegt werden können.</p> <p>Die im nachfolgenden Abschnitt unter „Verbindlichkeit“ genannten Bedingungen sind im Falle von Vorabinformationen ungültig.</p>
Verbindlichkeit	<p>Dieses Programmierhandbuch ist Teil des Gerätes/der Maschine. Dieses Programmierhandbuch muss jederzeit für den Bediener zugänglich und in einem leserlichen Zustand sein. Bei Verkauf/Verlagerung des Gerätes/der Maschine muss dieses Programmierhandbuch vom Besitzer zusammen mit dem Gerät/der Maschine weitergegeben werden.</p> <p>Nach Verkauf des Gerätes/der Maschine sind dieses Original und sämtliche Kopien an den Käufer zu übergeben. Nach Entsorgung oder anderem Nutzungsende sind dieses Original und sämtliche Kopien zu vernichten.</p> <p>Mit der Übergabe des vorliegenden Programmierhandbuchs werden entsprechende Parameterhandbücher mit einem früheren Stand außer Kraft gesetzt.</p> <p>Bitte beachten Sie, dass Angaben/Zahlen/Informationen aktuelle Werte zum Druckdatum sind. Zur Ausmessung, Berechnung und Kalkulation sind diese Angaben nicht rechtlich verbindlich.</p> <p>Die Firma Baumüller Nürnberg GmbH behält sich vor, im Rahmen der eigenen Weiterentwicklung der Produkte die technischen Daten und die Handhabung von Baumüller-Produkten zu ändern.</p> <p>Es kann jedoch keine Gewährleistung bezüglich der Fehlerfreiheit dieses Programmierhandbuchs, soweit nicht in den Allgemeinen Verkaufs- und Lieferbedingungen anders beschrieben, übernommen werden.</p>

© **Baumüller Nürnberg GmbH**

Ostendstr. 80 - 90
90482 Nürnberg
Deutschland

Tel. +49 9 11 54 32 - 0
Fax: +49 9 11 54 32 - 1 30

E-Mail: mail@baumueller.de
Internet: www.baumueller.de



Inhaltsverzeichnis

1	Einleitung	5
1.1	Allgemeines	5
1.2	Montage und Installation	5
1.3	Adresseinstellung	5
1.4	EDS-Datei	6
2	Grundlegende Sicherheitshinweise	7
2.1	Gefahrenhinweise und Gebote	7
2.2	Infozeichen	7
3	Grundlagen CAN/CANopen	9
3.1	Literatur zum Thema CAN	9
3.2	Grundlagen CAN	10
3.3	Grundlagen CANopen	11
3.4	Unterstützte Betriebsarten des Geräte Profils DSP 402	14
3.4.1	Kurze Übersicht	14
3.4.2	Betriebsarten und Feldbus-Objekte	15
4	Kommunikation zum b maXX[®] Regler	19
4.1	Kommunikationsablauf	19
4.2	Parametrierung der BACI Kommunikationszeiten	20
4.2.1	Parametrierung der BACI Kommunikationszeiten bei Lageregelung, Gleichlauf und IP-Mode	21
4.3	Konfigurationsmöglichkeiten der CANopen-Optionskarte in ProDrive/WinBASS II	22
4.3.1	Unterscheidung des Boot-Up Telegramms	22
4.3.2	Initialisierung CAN nach einem Busoff	23
4.3.3	Einstellungen ab der Reglerversion V03.00 (LC2)	23
4.3.4	Default Mapping nach CANopen Standard DSP 402 veränderbar	23
4.3.5	Node Guarding Reaktion	24
4.3.6	CANopen Offset	24
4.3.7	Reset nach einem SYNC-Telegrammausfall	24
4.3.8	Deaktivieren des Synchronisieren-Mechanismus zum Regler	24
4.3.9	EMCY-ErrorCode	24
4.4	Allgemeine Bemerkung zur CANopen Optionskarte	25
4.4.1	Applikationsparameter	26
4.4.2	Geschwindigkeitsprofil bei der Positionierung FBO 6086 _{hex}	26
4.4.3	Einstellbares Verhalten, wenn neues Ziel außerhalb der Software-Endschalter	26
4.4.4	Fehlerauslösung bei Fahrt in Hardware-Endschalter	26
4.4.5	Benutzer Einheiten BE	27
4.4.6	Getriebefaktor	28
4.4.7	CANopen Offset	29
4.4.8	Referenzfahrt für Positionierung erforderlich	29
4.4.9	Varianten der Positionierung, abhängig vom Positioniermodus (P601)	29
5	Datenaustausch und Parametrierung	31
5.1	Verzeichnis der Objekte zur Kommunikationssteuerung	31
5.2	Netzwerkmanagement (NMT)	43
5.2.1	Zustandsmaschine der Kommunikation	43
5.2.2	Telegramme	44
5.2.2.1	Zustandssteuerung	44



Inhaltsverzeichnis

5.2.2.2	Boot Up	46
5.2.3	Node Guarding	46
5.2.4	Heartbeat Protocol	48
5.3	Bedarfsdaten (SDO)	49
5.3.1	Telegrammaufbau	49
5.3.2	Arten des SDO-Transfers	50
5.3.3	Objekt schreiben	50
5.3.4	Objekt lesen	52
5.3.4.1	Expedited Transfer	52
5.3.4.2	Segmented Transfer	55
5.3.5	Fehlerreaktionen	57
5.4	Synchronisation (SYNC)	59
5.5	Prozessdaten (PDO)	60
5.5.1	PDO-Mapping	61
5.5.2	Kommunikationsbeziehung über PDO	65
5.5.3	Beispiel für das PDO-Mapping	67
5.5.4	Eintrag in die BACI	78
5.6	Fehlertelegramm (EMCY) nach DSP 402	79
5.6.1	Telegrammaufbau	79
5.6.2	Umsetzung der Fehlermeldungen nach DSP 402 V1.1	80
Anhang A - Abkürzungen		93
Anhang B - Kurzreferenz		95
B.1	4000'er Objektnummern (Herstellerspezifische Objekte)	95
B.2	6000'er Objektnummern (Device Profile DSP 402)	96
Anhang C - Umsetzungstabellen		101
Anhang D - Technische Daten		125
D.1	CANopen Optionskarte: Technische Merkmale	125
D.2	CANopen Optionskarte: Datenkanäle zum b maXX®-Regler	125
D.3	CANopen Optionskarte: CAN-Buffer	126
Abbildungsverzeichnis		127
Stichwortverzeichnis		129
Revisionsübersicht		131

1

EINLEITUNG

Das Programmierhandbuch ist ein wichtiger Bestandteil Ihres b maXX[®] 4400 Gerätes. Lesen Sie daher nicht zuletzt im Interesse Ihrer eigenen Sicherheit diese Dokumentation vollständig durch. In dieser Dokumentation erfahren Sie, wie die Fa. Baumüller Nürnberg GmbH die CANopen-Anschaltung auf dem Optionsmodul CANopen-Slave für die Gerätereihe b maXX[®] 4400 realisiert hat.

Die Einleitung beinhaltet allgemeine Informationen zum Optionsmodul CANopen-Slave.

1.1 Allgemeines

Das CANopen-Optionsmodul verbindet das b maXX[®] 4400 über den CAN-Bus mit anderen CANopen-Knoten (z. B. PC, SPS, weitere b maXX[®] 4400, I/O-Module).

Informationen zu Options- und Funktionsmodulen für die Gerätereihe b maXX[®] 4400 finden Sie in der Dokumentation 5.01040.

Informationen zur Programmierung des b maXX[®] 4400 Reglers finden Sie im Parameterhandbuch 5.02017.

1.2 Montage und Installation

Die Montage des Optionsmoduls CANopen-Slave beschreiben wir in der Dokumentation 5.02014.

1.3 Adresseinstellung

Die Adresseinstellung und die Einstellung der Baudrate des Optionsmoduls CANopen-Slave beschreiben wir in der Betriebsanleitung 5.02014.

1.4 EDS-Datei

Die EDS-Datei ist eine ASCII-Datei und dient der Beschreibung des Funktionsumfangs eines CANopen-Gerätes. Es ist ein elektronisches Datenblatt des CANopen-Gerätes. Die EDS-Datei wird von CANopen-Mastern bzw. Buskonfiguratoren genutzt. Die EDS-Datei beinhaltet Informationen über alle vom Slave unterstützten Objekte, Baudraten und weitere Merkmale.

Der Namensuffix der EDS-Datei ist *.eds.

GRUNDLEGENDE SICHERHEITSHINWEISE

In diesem Kapitel beschreiben wir Gefahren, die beim Parametrieren des Baumüller b maXX[®] 4400-Reglerteils auftreten können und wir erklären die Bedeutung des Infozeichens.

2.1 Gefahrenhinweise und Gebote



WARNUNG (WARNING)

Folgendes **kann eintreffen**, wenn Sie diesen Warnhinweis nicht beachten:

- schwere Körperverletzung
- Tod



Die Gefahr ist: **mechanische und elektrische Einwirkung**. *Die Änderung von Parametern beeinflusst das Verhalten des Baumüller-Geräts und somit das Verhalten der Anlage und ihrer Komponenten. Wenn Sie die Einstellungen der Parameter verändern, können Sie ein gefährliches Verhalten der Anlage und/oder ihrer Komponenten bewirken.*



Führen Sie nach jeder Änderung der Parametereinstellungen eine Inbetriebnahme durch unter Beachtung aller Sicherheitshinweise und Sicherheitsvorschriften.

2.2 Infozeichen



HINWEIS

Der hier angegebene Hinweis ist eine besonders wichtige Information.

GRUNDLAGEN CAN/CANOPEN

3.1 Literatur zum Thema CAN

Für grundlegende Informationen zum Thema CAN empfehlen wir folgende Literatur:

- „Etschberger“
CAN Controller-Area-Network
Konrad Etschberger
Carl Hauser Verlag München Wien
- „Lawrenz“
CAN Controller Area Network. Grundlagen und Praxis
Wolfhard Lawrenz
Hüthig Verlag
- „Zeltwanger“
CANopen
Holger Zeltwanger
VDE-Verlag
- www.can-cia.de
CAN in Automation e.V.
Am Weichselgarten 26
D-91058 Erlangen

3.2 Grundlagen CAN

	<p>Der CAN-Feldbus wird in Linienstruktur ausgeführt. Als physikalische Basis der Datenübertragung dient eine Dreidrahtleitung mit den Anschlüssen CAN_High, CAN_Low und CAN_Ground. CAN verwendet eine erdsymmetrische Übertragung, um Gleichtaktstörungen zu unterdrücken. Daher werden Differenzsignale ausgewertet.</p>
Netzwerk	<p>CAN ist ein Multi-Master-Netzwerk. Jeder Teilnehmer kann gleichberechtigt und aktiv auf den Bus zugreifen. CAN verwendet die objektorientierte Adressierung, d. h. die übermittelte Nachricht wird mit einem netzwerkweit festgelegten Identifier gekennzeichnet. Der Identifier stellt den codierten Namen der Nachricht dar.</p>
Buszugriff	<p>Der Buszugriff erfolgt über das CSMA/CA-Verfahren (Carrier Sense Multiple Access / Collision Avoidance). Da jeder Teilnehmer das Recht hat, nach Erkennung der notwendigen Busruhe, mit dem Senden seiner Nachricht zu beginnen, können Kollisionen entstehen. Dieses wird durch die bitweise Arbitrierung der zu sendenden Nachrichten vermieden. Dabei werden zwei Buspegel unterschieden, ein dominanter Pegel, logischer Bitwert 0, und ein rezessiver Pegel, logischer Bitwert 1. Im schlimmsten Fall beginnen alle sendewilligen Teilnehmer gleichzeitig mit dem Versenden ihrer Nachricht auf dem Bus. Wird ein rezessives Bit eines Teilnehmers von einem dominanten Bit eines anderen überschrieben, so zieht sich der „rezessive“ Knoten vom Bus zurück und versucht nach Erkennung der Busruhe erneut seine Nachricht abzusetzen. Somit ist gewährleistet, dass die wichtigste, höchstpriorie Nachricht (mit dem niedrigsten Identifier) kollisionsfrei und ohne Verzögerung übertragen wird. Aus diesem Grund ist es natürlich notwendig, dass jeder Identifier nur einmal am CAN-Bus vergeben sein darf.</p>
Identifier	<p>Es stehen in der CAN Spezifikation CAN 2.0A 2032 (CiA) unterschiedliche Identifier zur Verfügung. Jeder Teilnehmer kann unaufgefordert senden (Multi-Master-Fähigkeit). Ein Sender übermittelt seine Nachricht an alle CAN-Knoten (Broadcast), die anhand des Identifiers selbst entscheiden, ob sie die Nachricht weiterverarbeiten oder nicht.</p>
Fehler	<p>In einem CAN-Datentelegramm können bis zu acht Byte Nutzdaten übertragen werden. Zur Fehler- oder Überlastsignalisierung kann ein CAN-Knoten Error- oder Overload-Telegramme senden. Dieses geschieht auf Schicht 2 des OSI/ISO-Referenzmodells, dem Data Link Layer, also unabhängig von der Applikation. Aufgrund einer hochwertigen Fehlererkennung und -behandlung auf Schicht 2 wird eine Hamming-Distanz (Maß der Fehlererkennung) von $HD = 6$ erreicht, d. h. maximal fünf gleichzeitig auftretende Bitfehler innerhalb eines Telegramms werden sicher als Fehler erkannt.</p>

3.3 Grundlagen CANopen

CANopen ist ein offenes und damit herstellernerutrales Feldbussystem, welches auf den Layern 1 und 2 - Definitionen des CAN-Standards aufsetzt.

CAL-Spezifikation Das CANopen-Protokoll basiert auf der CAL-Spezifikation (Schicht 7-Protokoll). Bei CANopen werden Profile unterschieden. Das Kommunikationsprofil (DS 301) definiert die Art und Weise des Datenaustausches und allgemeine, für alle Geräte geltende Festlegungen.

Geräteprofil In den Geräteprofilen werden die anwendungs- und gerätespezifische Festlegungen, die inhaltliche Bedeutung der Daten und die Gerätefunktionalität beschrieben. Geräteprofile existieren u. a. für Antriebe, I/O-Module, Geber oder programmierbare Geräte. Das Optionsmodul CANopen-Slave für den b maXX[®] 4400 Regler ist nach dem Geräteprofil DSP402 (Drives and Motion Control) implementiert.

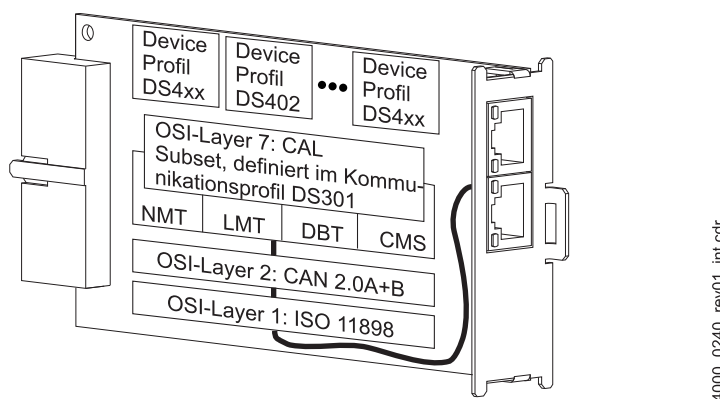


Abbildung 1: Profilstruktur von CANopen

Objektverzeichnis Das zentrale Element eines jeden CANopen-Gerätes ist das Objektverzeichnis des CANopen-Geräts.

Index (hex)	Objekt
0000	Nicht benutzt
0001 _{hex} - 001F _{hex}	Statische Datentypen
0020 _{hex} - 003F _{hex}	Komplexe Datentypen
0040 _{hex} - 005F _{hex}	Herstellerspezifische Datentypen
0060 _{hex} - 007F _{hex}	Geräteprofilsspezifische Statische Datentypen
0080 _{hex} - 009F _{hex}	Geräteprofilsspezifische Dynamische Datentypen
00A0 _{hex} - 0FFF _{hex}	Reserviert
1000 _{hex} - 1FFF _{hex}	Bereich für das Kommunikationsprofil
2000 _{hex} - 5FFF _{hex}	Bereich für herstellerspezifische Objekte
6000 _{hex} - 9FFF _{hex}	Bereich für das Geräteprofil
A000 _{hex} - AFFF _{hex}	Steuerungs-Objekte nach IEC 61131-3 programmierbare Geräte (DSP 405)

Die Objekte werden immer über einen Index (16 Bit) und zusätzlich über einen Subindex (8 Bit) adressiert.

CANopen unterscheidet vier Arten von Nachrichten:

- Administrative Nachrichten (z.B. Netzwerk-Management NMT, Layer-Management LMT)
- Servicedaten (SDO)
- Prozessdaten (PDO)
- Vordefinierte Nachrichten (z.B. Synchronisation, Time Stamp, Emergency)

NMT Mittels NMT-Diensten (Netzwerk-Management) werden die Kommunikationszustände des Gerätes gesteuert und überwacht.

SDO SDOs dienen der Übertragung größerer Datenmengen mit niedriger Priorität (Bedarfsdaten). Ein Datenblock mit mehr als vier Byte Nutzdaten wird dazu durch das CANopen-Protokoll segmentiert und auf mehrere SDOs verteilt (SDO Segmented Transfer). Datenrößen von maximal vier Byte werden mit einer SDO übertragen (SDO Expedited Transfer). Typischerweise werden SDOs zur Gerätekonfiguration benutzt. SDOs werden asynchron übertragen und vom Empfänger bestätigt. Mittels SDO kann auf alle Einträge im Objektverzeichnis zugegriffen werden.

PDO PDOs dienen dem Austausch von Prozessdaten (Daten mit hoher Priorität). PDOs können sowohl synchron als auch asynchron übertragen werden. Sie haben Broadcast-Charakter und werden vom Empfänger nicht bestätigt.

Synchron bedeutet, dass die Übertragung vom Synchronisations-Objekt abhängt. Der Inhalt der PDOs muss vom Anwender über SDOs festgelegt werden (variables PDO-Mapping). Dieses Mapping muss vor Beginn der Prozessdatenkommunikation abgeschlossen sein. In den Geräteprofilen sind Default-Mappings angegeben.

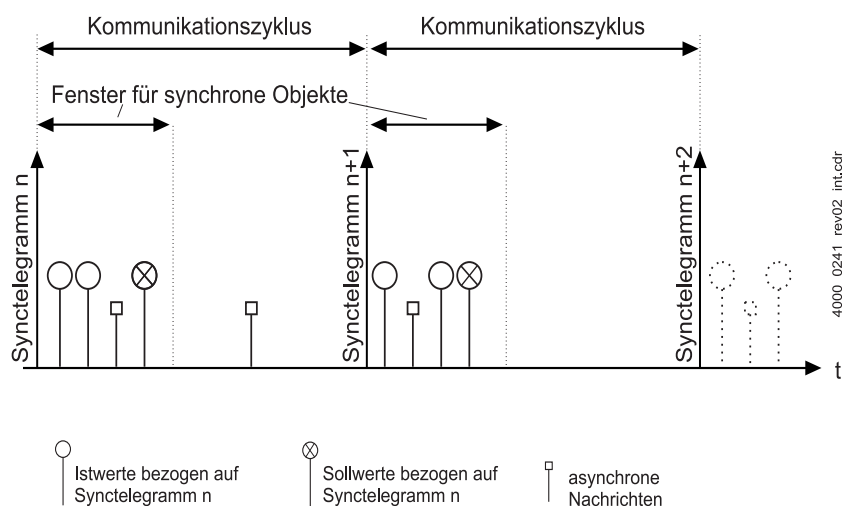


Abbildung 2: PDO Übertragungsarten

Die PDO-Kommunikation wird entweder durch das Auftreten bestimmter Ereignisse (z. B. Empfang eines SYNC-Telegramms oder Wertänderung) oder zeitgesteuert ausgelöst.

Um direkt nach einem Boot Up eine Peer-to-Peer Kommunikation zwischen Master und Slave aufbauen zu können, existiert eine vordefinierte Identifizierung. Diese Identifizierung kann vom Anwender umkonfiguriert werden.

Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Function Code				Module-ID						

Aus den sieben Bit für die Module-ID ergibt sich je CANopen-Netzwerk eine maximale Anzahl von 127 Knoten.

Objekt	Function Code binär	resultierende COB-ID	Objekt
NMT	0000	0	
SYNC	0001	128	1005 _{hex} , 1006 _{hex}
EMERGENCY	0001	129 - 255	1014 _{hex} , 1015 _{hex}
PDO1 (TX)	0011	385 - 511	1800 _{hex}
PDO1 (RX)	0100	513 - 639	1400 _{hex}
PDO2 (TX)	0101	641 - 767	1801 _{hex}
PDO2 (RX)	0110	769 - 895	1401 _{hex}
PDO3 (TX)	0111	896 - 1023	1802 _{hex}
PDO3 (RX)	1000	1025 - 1151	1402 _{hex}
PDO4 (TX)	1001	1153 - 1279	1803 _{hex}
PDO4 (RX)	1010	1281 - 1407	1403 _{hex}
SDO (TX)	1011	1409 - 1535	1200 _{hex}
SDO (RX)	1100	1537 - 1663	1200 _{hex}
Nodeguard	1110	1793 - 1919	100C _{hex} , 100D _{hex}

CANopen definiert ein Boot Up des Netzwerkes. Das einfache Boot Up enthält vier Kommunikationszustände:

- INITIALISIERUNG
- PRE-OPERATIONAL
- STOPPED
- OPERATIONAL

Die einzelnen Zustandsübergänge werden durch NMT-Kommandos ausgelöst. Das Optionsmodul CANopen-Slave wechselt nach dem Initialisieren automatisch in den Zustand PRE-OPERATIONAL. Weitere Informationen finden Sie in [▶Zustandsmaschine der Kommunikation◀](#) auf Seite 43.

3.4 Unterstützte Betriebsarten des Geräte Profils DSP 402

3.4.1 Kurze Übersicht

Folgende Betriebsarten werden unterstützt, d.h. alle mandatory Objekte sind auf dem Optionsmodul CANopen-Slave vorhanden	
Device control	optionale Objekte vollständig vorhanden
Objekte des Homing	optionale Objekte vollständig vorhanden
Objekte des Profile Position Mode	optionale Objekte nicht vollständig vorhanden
Position Control Funktion	optionale Objekte nicht vollständig vorhanden
Objekte des Velocity Mode	optionale Objekte nicht vollständig vorhanden
Objekte des Profile Velocity Mode	optionale Objekte nicht vollständig vorhanden
Common entries in the object dictionary (keine mandatory-Objekte vorhanden)	optionale Objekte nicht vollständig vorhanden
Interpolated Position Mode	optionale Objekte vollständig vorhanden

Folgende Betriebsarten werden nicht unterstützt, d.h. mindestens ein mandatory Objekt ist nicht vorhanden, es können auch optionale Objekte vorhanden sein.	
Profile Torque Mode	zwei Objekte

3.4.2 Betriebsarten und Feldbus-Objekte

Betriebsart	
Feld-Bus-Objekte	Feld-Bus-Name

Objekte des Homing Mode alle mandatory Objekte und alle optionalen Objekte werden unterstützt (Referenzfahrt)		
6098 _{hex}	mandatory	homing_method
6099 _{hex}	mandatory SIX 0 = 2	homing_speed
607C _{hex}	optional	home_offset
609A _{hex}	optional	homing_acceleration

Device control alle mandatory Objekte und alle optionalen Objekte werden unterstützt (Gerätekontrolle)		
6040 _{hex}	mandatory	controlword
6041 _{hex}	mandatory	statusword
6060 _{hex}	mandatory	modes_of_operation
6061 _{hex}	mandatory	modes_of_operation_display
605A _{hex}	optional	quick_stop_option_code
605B _{hex}	optional	shutdown_option_code
605C _{hex}	optional	disable_operation_option_code
605D _{hex}	optional	halt_reaction_option_code
605E _{hex}	optional	fault_reaction_option_code

Objekte des Profile Torque Mode ein optionales Objekt wird unterstützt (Momentenregelung)		
6072 _{hex}	optional	max_torque
6077 _{hex}	optional	torque_actual_value

Objekte des Profile Position Mode alle mandatory Objekte, zum Teil optionale Objekte werden unterstützt (Positionierung)			
607A _{hex}	mandatory		target_position
607D _{hex}	optional	SIX 0 = 2	software_position_limit
607F _{hex}	optional		max_profile_velocity
6080 _{hex}	optional		max_motor_speed
6081 _{hex}	mandatory		profile_velocity
6083 _{hex}	mandatory		profile_acceleration
6084 _{hex}	mandatory		profile_deceleration
6085 _{hex}	optional		quick_stop_deceleration
6086 _{hex}	mandatory		motion_profile_type

Objekte des Profile Velocity Mode alle mandatory Objekte, zum Teil optionale Objekte werden unterstützt (Drehzahlregelung)			
606A _{hex}	mandatory		sensor_selection_code
6069 _{hex}	mandatory		velocity_sensor_actual_value
606B _{hex}	mandatory		velocity_demand_value
606C _{hex}	mandatory		velocity_actual_value
606F _{hex}	optional		velocity_threshold
60FF _{hex}	mandatory		target_velocity
60F8 _{hex}	optional		max_slippage

Interpolated Position Mode alle Objekte, Objekte werden unterstützt (Positionierkontrolle)			
60C0 _{hex}	optional		Interpolation sub mode select
60C1 _{hex}	optional	SIX 0 = 1	Interpolation data record
60C2 _{hex}	mandatory	SIX 0 = 2	Interpolation time period
60C3 _{hex}	optional	SIX 0 = 2	Interpolation sync definition
60C4 _{hex}	optional	SIX 0 = 6	Interpolation data configuration

Position Control Funktion alle mandatory Objekte, zum Teil optionale Objekte werden unterstützt (Positionierkontrolle)			
6067 _{hex}	optional		position_window
6068 _{hex}	optional		position_window_time
6064 _{hex}	mandatory		position_actual_value
6063 _{hex}	optional		position_actual_value*
6062 _{hex}	optional		position_damand_value
6066 _{hex}	optional		following_error_time_out
60FB _{hex}	optional	SIX 0 = 28	position_control_parameter_set

Objekte des Velocity Mode alle mandatory Objekte, zum Teil optionale Objekte werden unterstützt (Drehzahlreglung)			
6042 _{hex}	mandatory		vl_target_velocity
6043 _{hex}	mandatory		vl_velocity_demand
6044 _{hex}	mandatory		vl_control_effort
6045 _{hex}	optional		vl_manipulated_velocity
6048 _{hex}	mandatory	SIX 0 = 2	vl_velocity_acceleration
6049 _{hex}	mandatory	SIX 0 = 2	vl_velocity_deceleration
6046 _{hex}	mandatory	SIX 0 = 2	vl_velocity_min_max_amount
604C _{hex}	optional	SIX 0 = 2	vl_manipulated_velocity
604D _{hex}	optional		vl_pole_number
604F _{hex}	optional		vl_ramp_function_time
6050 _{hex}	optional		vl_slow_down_time
6051 _{hex}	optional		vl_quick_stop_time

Common Entries im Object Dictionary Keine mandatory Objekte sind vorhanden, zum Teil werden optionale Objekte unterstützt (allgemeine Eingänge im Objekt Verzeichnis)			
60FD _{hex}	optional		digits_inputs
6007 _{hex}	optional		abort_connection_option_code
6510 _{hex}	optional	SIX 0 = 08	drive_date

3.4 Unterstützte Betriebsarten des Geräte Profils DSP 402

Factor Group Keine mandatory Objekte sind vorhanden, zum Teil werden optionale Objekte unterstützt (Gruppe der Benutzer Einheiten)			
6092 _{hex}	optional	SIX 0 = 2	feed_constant

KOMMUNIKATION ZUM b maXX[®] REGLER

In diesem Kapitel beschreiben wir die Datenkommunikation zwischen dem Optionsmodul CANopen-Slave und dem b maXX[®] 4400-Gerät.

4.1 Kommunikationsablauf

Das CANopen-Optionsmodul tauscht über ein Dual-Port-RAM Daten mit dem b maXX[®] 4400 Regler aus. Dieser Datenaustausch erfolgt in einem bestimmten Zeitraster über die BACI-Schnittstelle (Baumüller-Bus).

Das Optionsmodul CANopen-Slave stößt die Kommunikation mit dem b maXX[®] 4400 Regler an. Bei der Kommunikation werden zwei unterschiedliche Typen von Daten übertragen:

- Prozessdaten
- Bedarfsdaten

Die Prozessdaten werden immer zyklisch übertragen. In der Restzeit werden die Bedarfsdaten übertragen. Die Prozessdaten Übertragung findet in einem einstellbaren Zeitraster, dem SYNC-Intervall statt. Dabei werden die Sollwert und Istwerte mit jeweils verschiedenen Offset im SYNC-Intervall übertragen.

Bei den Betriebsarten Lageregelung und Gleichlauf muss die Zykluszeit des SYNC-Telegramms mit der BACI-Zykluszeit übereinstimmen. Bei den anderen Betriebsarten kann die BACI-Zykluszeit von der Zykluszeit des SYNC-Telegramms abweichen. Das SYNC-Telegramm ist hier als Triggerbedingung zu verstehen.

HINWEIS



Die zyklische Kommunikation ist nur im CANopen-Kommunikationszustand OPERATIONAL aktiv.

4.2 Parametrierung der BACI Kommunikationszeiten

Zwischen dem Optionsmodul CANopen-Slave und dem b maXX[®] Regler können 8 Sollwerte (Ausnahme: FBO 607A_{hex} Zielposition zählt als 2 FBOs, da es auf 2 Reglerparameter abgebildet wird) und 8 Istwerte als Prozessdaten in einem Kommunikationszyklus ausgetauscht werden. Welche Sollwerte und Istwerte Sie austauschen, legen Sie in den Mapping-Objekten auf dem Optionsmodul CANopen-Slave fest (Einstellung über SDO durch den CANopen-Master oder Default-Einstellung, siehe Kapitel [► Datenaustausch und Parametrierung ◄](#) ab Seite 31. Wie Sie die Kommunikation parametrieren wird in diesem Kapitel aufgezeigt.

Die Einstellung der Kommunikationszeiten zwischen Optionsmodul CANopen-Slave und b maXX[®]-Regler wird vorzugsweise über ProDrive/WinBASS II parametriert. Es ist auch möglich über den Feldbus zu parametrieren.

Auf der ProDrive/WinBASS II Seite „BACI“ (Optionsmodul1) wird die Kommunikationszykluszeit (Rate Sollwerte, Istwerte), der Zyklus-Offset der Sollwerte und der Zyklus-Offset der Istwerte eingestellt.

Der b maXX[®] Regler ruft alle 125 µs eine Kommunikationszeitscheibe auf, in der Prozessdatensollwerte oder Prozessdatenistwerte übertragen werden.

Die Kommunikationszykluszeit ist ein Vielfaches des Aufrufs der Kommunikationszeitscheibe (alle 125 µs). In der EditBox „Rate Sollwerte, Istwerte“ wird nur der Faktor angegeben, d. h. der Wert in der EditBox „Rate Sollwerte, Istwerte“ berechnet sich wie folgt:

$$\text{Zykluszeit Sollwerte, Istwerte (P800)} = \frac{\text{Kommunikationszykluszeit (in } \mu\text{s)}}{125 \mu\text{s}}$$

Beispiele:

Kommunikationszykluszeit = 2000 µs ⇒ Sollwerte, Istwerte = 16 (P800)

Kommunikationszykluszeit = 4000 µs ⇒ Sollwerte, Istwerte = 32 (P800)

Es wird empfohlen die Zykluszeit der BACI (P800) auf 1000 µs entspricht 8 einzustellen. Ausgenommen sind hiervon die Betriebsarten Lageregelung und Gleichlauf. Die Einstellungen hierzu siehe [► Parametrierung der BACI Kommunikationszeiten bei Lageregelung, Gleichlauf und IP-Mode ◄](#) ab Seite 21.

- Zyklus Offset Sollwerte (P818) = 7
- Zyklus Offset Istwerte (P819) = 0

HINWEIS



Beachten Sie bei der Festlegung von BM_u_Baci1M1Period: auf die BACI kann nur alle 125 µs zugegriffen werden.

Die Prozessdatensollwerte und die Prozessdatenistwerte werden in verschiedenen Kommunikationszeitscheiben übertragen. Deshalb gibt man für die Sollwerte einen anderen Zyklus-Offset an als für die Istwerte. Der Zyklus-Offset ist nichts anderes als die Nummer der Kommunikationszeitscheibe, in der die Daten übertragen werden.

HINWEIS

Andere Einstellungen für den Zyklus-Offset der Sollwerte und Istwerte sind möglich, jedoch ändert sich dadurch das Zeitverhalten.

Auf der ProDrive/WinBASS II Seite „BACI“ (Optionsmodul1) finden sie die Parameternummern der Sollwerte und der Istwerte (P801 bis P816). Diese dienen nur der Anzeige, da die Einstellung der Parameternummern für den Prozessdatenaustausch in den Mapping-Objekten auf dem Optionsmodul CANopen-Slave festgelegt wird. Die vorher gewählten Einstellungen für die Offsets müssen im Datensatz abgespeichert werden und der Regler muss neu gebootet werden.

HINWEIS

Wird die zyklische Kommunikation unterbrochen, z.B. beim Übergang von OPERATIONAL nach PRE-OPERATIONAL kann der Fehler/Warnung Alive Counter oder der Fehler zyklische Kommunikation auftreten.

HINWEIS

CANopen State Machine

Beim Übergang von PRE-OPERATIONAL nach OPERATIONAL werden die Parameternummern dem Mapping zugeordnet. Diese Zuordnung ist zeitintensiv und kann mehrere Millisekunden (bis zu 11 ms, je nach Zeiteinstellungen der BACI auch länger) andauern, solange wird keine PDO gesendet und auch keine RX-PDO bearbeitet.

4.2.1 Parametrierung der BACI Kommunikationszeiten bei Lageregelung, Gleichlauf und IP-Mode

Bei den Betriebsarten Lageregelung, Gleichlauf und IP-Mode müssen die Zykluszeiten der BACI und die Zykluszeiten des Feldbusses (SyncTelegramm) übereinstimmen, weil die Synchronisation des Reglers auf das Sync-Telegramm erfolgt.

Der kleinste Wert für die "BACI Zykluszeit" ist 2 ms, d. h. die kleinste Kommunikationszykluszeit ist 2 ms für diese Betriebsart. Der größte Wert für die "BACI Zykluszeit" ist 8 ms.

Um ein Gefühl dafür zu bekommen wie viele Telegramme bei welcher Baudrate in CANopen machbar sind, sind folgenden Überlegungen aufgezeigt.

Z. B. bei 1000 KBit Baudrate dauert die Übertragung eines 8 Byte langen Telegramms zwischen 130-140 µs, bei 500 KBit verdoppelt sich die Übertragungszeit.

Betrachtung bei einer Baudrate von 1 MBit. (Dies sind nur Richtwerte, denn wenn auch noch Servicedaten übertragen werden sollen, wird der CAN-Bus bis an seine Grenzen gebracht, bzw. können SynchronisationsSchwierigkeiten auftreten):

- Bei 2 ms können bei max. 4 Slaves im Ring 1 PDO je Richtung verarbeitet werden.
- Bei 4 ms können bei max. 7 Slaves im Ring 1 PDO je Richtung verarbeitet werden
- Bei 8 ms können bei max. 14 Slaves im Ring 1 PDO je Richtung verarbeitet werden.

Die Prozessdaten-Sollwerte und die Prozessdaten-Istwerte werden in verschiedenen Kommunikationszeitscheiben zum Regler übertragen. Deshalb gibt man für die Sollwerte einen anderen BACI Zyklus-Offset an, als für die Istwerte. Der BACI Zyklus-Offset ist nichts anderes als die Nummer der Kommunikationszeitscheibe, in der die Daten übertragen werden.

Bei einer Zykluszeit von 2000 μ s Sollwerte = 15 Istwert = 0

Bei einer Zykluszeit von 4000 μ s Sollwerte = 31 Istwert = 0

Bei einer Zykluszeit von 8000 μ s Sollwerte = 63 Istwert = 0

Die Einstellungen müssen im Datensatz abgespeichert werden und der Regler muss neu gebootet werden.

Andere Einstellungen für den Zyklus-Offset der Soll- und Istwerte sind möglich, jedoch kann die Übernahme der Daten in den nächsten Zyklus rutschen bzw. das Zeitverhalten der Datenübernahme ändert sich dadurch.

HINWEIS



Die Einstellungen gelten nur bei den Betriebsarten Lageregelung und Gleichlauf.

Auf der ProDrive/WinBASS II Seite "BACI" (Optionsmodul 1) finden sie die Parameternummern der Sollwerte und der Istwerte (P801 bis P816). Diese dienen nur der Anzeige, da die Einstellung der Parameternummern für den Prozessdatenaustausch in den Mapping-Objekten auf dem Optionsmodul CANopen-Slave festgelegt wird.

4.3 Konfigurationsmöglichkeiten der CANopen-Optionskarte in ProDrive/ WinBASS II

ProDrive/WinBASS II „Optionsmodul G/H - Konfiguration 1“ (P830 / P840).

Abhängig in welchem Steckplatz die CANopen-Optionskarte gesteckt ist.

HINWEIS



Einstellungen führen zu einem geänderten Verhalten!

4.3.1 Unterscheidung des Boot-Up Telegramms

Standard ist das Boot-Up Verhalten nach DS 301 V4.

Zur Unterscheidung des Boot-Up Telegramms nach DS 301 V3 oder DS 301 V4 kann der Parameter „Optionsmodul G - Konfiguration 1“ oder „Optionsmodul H - Konfiguration 1“ im b maXX[®] Regler eingestellt werden. Abhängig davon, in welchem Steckplatz die CANopen-Optionskarte gesteckt wird.

Bit 0 ⇒ 0: nach DS 301 V4

Bit 0 ⇒ 1: nach DS 301 V3

- 1 Boot-Up nach DS 301 V4, Boot-Up Telegramm mit ID = 700_{hex} + Node-ID, DLC = 1 Byte 0 mit den Daten = 0 gefüllt.

COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
701 _{hex}	01 _{hex}	00 _{hex}							

DSP 301 V4

- 2 Boot-Up nach DS 301 V3, ID = 80_{hex} + Node-ID, DLC = 0 und anschließenden ID = 80_{hex} + Node-ID, DLC = 8 Byte 0-7 mit den Daten = 0 aufgefüllt (Reset des Error Registers).

COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
81 _{hex}	00 _{hex}								
81 _{hex}	08 _{hex}	00 _{hex}	00 _{hex}	00 _{hex}	00 _{hex}	00 _{hex}	00 _{hex}	00 _{hex}	00 _{hex}

DSP 301 V3

4.3.2 Initialisierung CAN nach einem Busoff

Bit 1 ⇒ 0: Busoff bleibt bestehen

Bit 1 ⇒ 1: Initialisierung des CAN nach einer Wartezeit von ca. 200 ms, solange bis der Busoff beseitigt ist

4.3.3 Einstellungen ab der Reglerversion V03.00 (LC2)

(Änderungen der Einheiten einiger FBOs, z.B. Einheiten 1/10 U/min)

Bit 2 ⇒ 0: Auflösung für die Drehzahl 1 U/min

Bit 2 ⇒ 1: neue Funktionen, Neuerungen werden im weiteren Verlauf aufgeführt
Auflösung für die Drehzahl 1/10 U/min

Folgende Objekte sind davon betroffen:

6042_{hex}, 6043_{hex}, 6045_{hex}, 6048_{hex} SIX 1, 6049_{hex} SIX 1, 606B_{hex}, 606C_{hex}, 60FF_{hex}

4.3.4 Default Mapping nach CANopen Standard DSP 402 veränderbar:

Bit 3 ⇒ 0: Default Mapping nach DSP 402.

Sowie FBO 6007_{hex} Node Guarding-Reaktion = 0 (CANopen Standard).

Bit 3 ⇒ 1: Default Mapping bei dem alle FBO Objekte auf Null gesetzt werden, Triggertypen bleiben nach DSP 402 auf Default.

FBO 6007_{hex} Node Guarding-Reaktion wird dann auf 3 gesetzt.

4.3 Konfigurationsmöglichkeiten der CANopen-Optionskarte in ProDrive/ WinBASS II

4.3.5 Node Guarding Reaktion

Bit 4 \Rightarrow 0: CANopen State wechselt nicht nach PRE-OPERATIONAL sondern bleibt im aktuellen Zustand (Standard).

Bit 4 \Rightarrow 1: wie bisher, CANopen State wechselt immer nach PRE-OPERATIONAL nachdem ein Node Gard Event eingetreten ist.

4.3.6 CANopen Offset

Bit 5 \Rightarrow 0: Zahlenskala Wandlung von U32 auf INT32, bei der Positionierung wird ein Offset von 2^{31} je nach Richtung auf die zugehörigen FBOs addiert/subtrahiert.

Bit 5 \Rightarrow 1: es wird kein Offset benutzt.

Näheres siehe dazu [▶CANopen Offset◀](#) auf Seite 29.

4.3.7 Reset nach einem SYNC-Telegrammausfall

Bit 6 \Rightarrow 0: kein Reset der Synchronisation nach einem SYNC- Telegrammausfall

Bit 6 \Rightarrow 1: Reset der Synchronisation nach einem SYNC- Telegrammausfall

4.3.8 Deaktivieren des Synchronisieren-Mechanismus zum Regler

Der Mechanismus dient zum Ausgleich von größerem Jitter des Sync-Telegramms. Er ist im FPGA auf der Optionskarte implementiert.

Bit 8 \Rightarrow Mechanismus aktiviert

Bit 8 \Rightarrow Mechanismus deaktiviert

4.3.9 EMCY-ErrorCode

Im DSP 402 nicht definierte Reglerfehler werden zu dem herstellereigenen Errorcode $FF00_{hex}$ dazu addiert, z.B. Reglerfehler Nummer 167 (Bremse öffnet nicht), wird dann mit $FF00_{hex} + 00A7_{hex}$ (Nr. 167) = $FFA7_{hex}$ ausgegeben.

Bit 9 \Rightarrow 0: neues Verhalten wie oben beschrieben Standard z.B.: $FFA7_{hex}$

Bit 9 \Rightarrow 1: altes Verhalten z.B.: $FF00_{hex}$



HINWEIS

CANopen State Machine

Beim Übergang von PRE-OPERATIONAL nach OPERATIONAL werden die Parameternummer dem Mapping zugeordnet. Diese Zuordnung ist zeitintensiv und kann mehrere Millisekunden (bis zu 11 ms je nach Zeiteinstellungen der BACI auch länger) andauern, solange keine PDO gesendet und auch keine RX-PDO bearbeitet.

4.4 Allgemeine Bemerkung zur CANopen Optionskarte

Wichtig: Änderungen die über ProDrive/WinBASS II erfolgen, werden nicht automatisch auf der CANopen Optionskarte aktualisiert oder wahrgenommen. Der Zugriff auf den Regler sollte daher nach Möglichkeit mit FBO über CANopen statt finden.

Es können z. B. Änderungen über ProDrive/WinBASS II bei der Umschaltung zwischen relativen und absoluten Positioniermodi auf der CANopen Optionskarte während des Positionierbetriebs nicht wahrgenommen werden. Dazu zählen z. B. auch Änderungen der Betriebsart über ProDrive/WinBASS II. Die Umschaltung / Änderung sollte nur über CANopen stattfinden.

Folgende Reglerparameter sind betroffen:

P0830/ P0840	kein FBO nach DSP 402, Zugriff nur über herstellerspezifisches Objekt möglich ($433E_{\text{hex}}$ / 4348_{hex}),
P0304	($FBO\ 6060_{\text{hex}}$),
P1031	($FBO\ 6080_{\text{hex}}$),
P3050	($FBO\ 6092_{\text{hex}}\ SIX1$),
P3051	($FBO\ 6092_{\text{hex}}\ SIX2$),
P0601	(interne Umschaltung auf der CANopen Optionskarte durch das Steuerwort Bit 6, Betriebsart Positionierung, relative und absolute Modi),
P0531	($FBO\ 1006_{\text{hex}}$)
P0532	($FBO\ 1006_{\text{hex}}$),
P3314	($FBO\ 604C_{\text{hex}}\ SIX1$),
P3315	($FBO\ 604C_{\text{hex}}\ SIX2$)
P1190	($FBO\ 6086_{\text{hex}}$).

Besteht der Wunsch, die Änderungen doch über ProDrive/WinBASS II einzugeben, ist es möglich, die Parameter beim Übergang der CANopen Zustandsmaschine OPERATIONAL nach PRE-OPERATIONAL oder nach STOP zu aktualisieren. Weiterhin findet nach einem Abspeichern im Datensatz und einem Power-Down und Power-Up des Reglers eine Aktualisierung statt.

Die anschließenden FBOs können über Feldbus mit SIX 1/2 eingegeben werden:

$FBO\ 6048_{\text{hex}}\ SIX1, SIX2$, für die Ermittlung der Beschleunigung (P1172),

$FBO\ 6049_{\text{hex}}\ SIX1, SIX2$, für die Ermittlung der Verzögerung (P1173).

Die Normierungen der FBO können von denen von ProDrive/WinBASS II abweichen.

z. B.: Eingabe der Positioniergeschwindigkeit über den FB in [m/s] und Eingabe über ProDrive/WinBASS II in [INC/ms] entspricht einem Unterschied von Faktor 1000.

4.4 Allgemeine Bemerkung zur CANopen Optionskarte

4.4.1 Applikationsparameter

Die Applikationsparameter P3314, P3315, P3329, P3330 werden auf der CANopen Optionskarte verwendet und dürfen nicht anderweitig benutzt werden.

4.4.2 Geschwindigkeitsprofil bei der Positionierung FBO 6086_{hex}

Die Änderungen gelten für das Handling der SW/HW-Endschalter sowie der Referenzfahrt (z. B. Limits / Warnungen aktivieren). Abspeichern im Datensatz und ein Neustart sind notwendig, wenn die Einstellungen über ProDrive/WinBASS II erfolgen. Das Geschwindigkeitsprofil kann über das FBO 6086_{hex} eingestellt werden, auch während der Positionierung. Dabei wird der aktuelle Fahrbefehl zu Ende gefahren und dann der neue Fahrbefehl mit dem neuen Profil gestartet.

4.4.3 Einstellbares Verhalten, wenn neues Ziel außerhalb der Software-Endschalter

Dies ist in ProDrive/WinBASS II unter „Antriebsmanager 2 Warnung aktivieren“ einstellbar und im Datensatz abspeicherbar.

Wenn neues Ziel außerhalb \Rightarrow keine Bewegung;

Es wird eine CAN Emergency Meldung Code 8600_{hex} Positioning Controller (Regler Fehler Nr. 196 SW Endschalter 1, Regler Fehler Nr. 197 SW Endschalter 2) abgesetzt. Verhalten des Antriebs über 605A_{hex} einstellbar. Der Fehler muss quittiert werden und ein neuer Fahrsatz kann anschließend ausgeführt werden.

Wenn die aktuelle Position bereits außerhalb und das neue Ziel auch außerhalb ist \Rightarrow keine Bewegung;

Es wird eine CAN Emergency Meldung Code 8600_{hex} Positioning Controller (Regler Fehler Nr. 196 SW Endschalter 1, Regler Fehler Nr. 197 SW Endschalter 2) abgesetzt. Verhalten des Antriebs über 605A_{hex} einstellbar. Der Fehler muss quittiert werden und ein neuer Fahrsatz kann anschließend ausgeführt werden.

4.4.4 Fehlerauslösung bei Fahrt in Hardware-Endschalter

Die HW-Endschalterüberwachung ist in ProDrive/WinBASS II unter Antriebsmanager 2 Warnung aktivieren einstellbar.

Es wird eine CAN Emergency Meldung Code 8600_{hex} Positioning Controller (Regler Fehler Nr. 198 negativer HW Endschalter, Regler Fehler Nr. 199 positiver HW Endschalter) abgesetzt. Der erzeugte Fehler führt nicht zur Impulssperre. Er muss quittiert werden, bevor der Start eines neuen Fahrauftrages ausgeführt wird.

4.4.5 Benutzer Einheiten BE

Die Benutzereinheiten können über ProDrive/WinBASS II \Rightarrow unter „Umnormierung“ eingegeben werden. Im Datensatz abspeichern, um die BE auch nach einem Ausschalten zu sichern.

Sind die gewünschten BE eingestellt sollten diese auch bei nachfolgenden Updates des Reglers erhalten bleiben. Vorsichtshalber nochmals in ProDrive/WinBASS II überprüfen.

Wichtig: Im Default Datensatz ist für die Benutzereinheiten 1 BE = 1 INC eingestellt.

Über CANopen können die BE über das FBO 6092_{hex} eingegeben werden.

6092_{hex}: feed constant = feed / driving shaft revolutions

„driving shaft revolutions“ wird intern auf der CANopen-Optionskarte mit 65536 multipliziert. Maximale Eingabe für „feed“ (BE) ist $0 \dots 2^{24} - 1$.

SIX1 = feed [in Benutzer Einheiten z. B. 360.00 Grad, 1/100 Grad Auflösung]

Wird im b maXX[®] Regler auf P3050 abgebildet und ist im Datensatz abspeicherbar.

SIX2 = driving shaft revolutions [1 Umdrehung wird intern auf der CANopen-Optionskarte mit 65536 [INC] multipliziert].

Wird im b maXX[®] Regler auf P3051 abgebildet und ist im Datensatz abspeicherbar.

Die Anzahl der Umdrehungen ist auf 255 begrenzt.

Die Eingabe über den Feldbus 360,00 Grad wird auf der Optionskarte auf 65536 Inkremente für eine Umdrehung umgerechnet.

Beispiel: Positionssollwert in BE = 36000 entspricht 360,00 Grad (entspricht 65536 Inc).

Die Umrechnung auf der CANopen-Optionskarte für den Positionssollwert FBO 607A_{hex} sieht folgendermaßen aus:

Positions Sollwert [INC] im b maXX[®]

= FBO [BE] * driving shaft revolutions * 65536 [INC] / feed [BE]

= 36000 * 1 * 65536 / 36000 [BE * INC / BE]

= 65536 [INC]

HINWEIS



Die Berechnung der Benutzereinheiten ist sehr Zeit intensiv und sollte nicht bei zyklischen Betrieb (Lage und Gleichlauf) benutzt werden. Werden die Benutzereinheiten auf 1:1 gesetzt entfällt die Berechnung der BE.

Auf folgende FBOs wirken sich die BE aus:

6062_{hex}, 6063_{hex}, 6064_{hex}, 6067_{hex}, 607A_{hex}, 607C_{hex}, 607D_{hex} Sub1/2, 6081_{hex}, 6083_{hex}, 6084_{hex}, 6085_{hex}, 6099_{hex} Sub1/2, 609A_{hex}

4.4.6 Getriebefaktor

Zusätzlich zu den Benutzereinheiten kann ein Getriebefaktor eingestellt werden, der mit dem neu eingeführten Feldbus-Objekt 604C_{hex} eingestellt wird. Mit dem Getriebefaktor ist es nun möglich z. B. die Getriebeübersetzung oder andere Skalierungen zu berücksichtigen, woraus dann die notwendige Geschwindigkeit des Antriebes errechnet wird.

FBO 604C_{hex}:

vl_dimension_factor =

vl_dimension_factor_numerator / vl_dimension_factor_denominator

SIX1 = vl_dimension_factor_numerator

Auf P3314 im b maXX[®] abgebildet INT32 (-33000 ... 33000)

SIX2 = vl_dimension_factor_denominator

Auf P3315 im b maXX[®] abgebildet INT32 (-33000 ... 33000)

Die Umrechnung sieht z. B. folgendermaßen aus:

Sollwert Drehzahl Motor im b maXX[®]:

Für vl_dimension_factor_numerator = 10

und vl_dimension_factor_denominator = 5

Sollwert Drehzahl Motor = FBO[U/min] * vl_dimension_factor
= 100 * 10 / 5 [U/min]
= 200 [U/min]

HINWEIS



Die Berechnung des Getriebefaktors ist sehr zeitintensiv und sollte nicht bei zyklischen Betrieb (Lage- und Gleichlaufbetrieb) benutzt werden. Wird der Getriebefaktor auf 1:1 gesetzt entfällt die Berechnung.

Auf folgende FBOs wirkt sich der Getriebefaktor aus:

6042_{hex}, 6043_{hex}, 6045_{hex}, 6048_{hex} Sub01/02, 6049_{hex} Sub01/02, 606B_{hex}, 606C_{hex},
60FF_{hex}

4.4.7 CANopen Offset

Abbildung der Zahlenskala USIGN32 auf INT32 (CANopen-Mode). Beim Schreiben/Lesen auf einige FB Objekte wird intern auf der CANopen Optionskarte je nach Richtung ein Offset von 2^{31} dazu addiert bzw. subtrahiert.

Sollen die Positionswerte und die Zielposition ebenfalls in ProDrive/WinBASS II in der INT32 Zahlenskala dargestellt werden, kann auf der Seite „Umnormierung“ eine Checkbox für den Offset aktiviert werden.

Auf folgende FBOs wirkt sich der CANopen Offset aus:

$$(6062_{\text{hex}}, 6064_{\text{hex}}, 607A_{\text{hex}}, 607C_{\text{hex}}, 607D_{\text{hex}} \text{ Sub } \frac{1}{2}) - 2^{31}$$

$$(607A_{\text{hex}}, 607C_{\text{hex}}, 607D_{\text{hex}} \text{ Sub } \frac{1}{2}) + 2^{31}$$

4.4.8 Referenzfahrt für Positionierung erforderlich

In ProDrive/WinBASS II auf der Seite „Referenzfahrt“ mit der dafür vorgesehenen Checkbox kann die Aktivierung dafür erfolgen, ob der Antrieb eine Positionierung zulässt, wenn keine erstmalige Referenzfahrt erfolgt ist.

Deaktiviert: Zum Betrieb in der Betriebsart Positionieren ist keine Referenzfahrt erforderlich.

Aktiviert: Wenn der Antrieb in Betriebsart Positionieren freigegeben wird, ohne dass vorher eine Referenzfahrt erfolgt ist, wird eine Fehlermeldung (EMYC- Telegramm 8600_{hex} \Rightarrow Regler Fehler Nr. 200) ausgegeben und der Antrieb bleibt lagegeregelt auf der aktuellen Position stehen. Positionieraufträge werden nicht ausgeführt. Erst nachdem eine Referenzfahrt durchgeführt wurde (einmalig nach dem Einschalten), werden Positionieraufträge ausgeführt. Die Fehlermeldung kann nur quittiert werden, wenn eine Referenzfahrt durchgeführt wurde. Nach der Referenzierung kann eine Positionierung gestartet werden.

HINWEIS



Ist der CANopen-Mode als Standard definiert, ist eine Referenzfahrt erforderlich!

4.4.9 Varianten der Positionierung, abhängig vom Positioniermodus (P601)

HINWEIS



Es ist darauf zu achten, dass in ProDrive/WinBASS II unter Positionierung 0 auch der Positionier-Datensatz 0 eingestellt ist, ansonsten wird die Positionierung über den CANopen nicht korrekt durchgeführt. Das Umschalten zwischen den Positioniermodi „relativ“, „negativ/positiv“ und „absolut“ findet nur über das Steuerwort statt. Der Positionierung sollte im CANopen-Mode (Standard) immer eine Referenzierung vorausgehen.

Folgende Positioniermodi existieren:

Positioniermodus P601	Beschreibung
„Absolut/Relativ“ CANopen (Standardwert 9)	<ul style="list-style-type: none"> • Ziel steht in P0607 (INT32) • Umschaltung „absolut/relativ“ findet nur über das Steuerwort statt
„Relativ, positive und negativ“ (Wert 4)	<ul style="list-style-type: none"> • Ziel steht in P0607 (INT32) • keine Umschaltung „absolut/relativ“ über das Steuerwort
„Absolut relativ“ (Wert 10)	<ul style="list-style-type: none"> • Ziel steht in P0600 (USIGN32) • Umschaltung „absolut/relativ“ findet nur über das Steuerwort statt
Alle anderen Modis	<ul style="list-style-type: none"> • Ziel steht in P0600 (USIGN32) • keine Umschaltung „absolut/relativ“ über das Steuerwort • keine Umrechnung Datentyp = UINT32

Umschaltung „absolut/relativ“. Über das Steuerwort Bit 6

Bit 6 = 0 -> Absolut

Bit 6 = 1 -> Relativ

Die Umrechnung vom Datentyp INT32 <-> UINT32 soll heißen, ein Offset von 2^{31} wird je nach Richtung dazu addiert oder abgezogen. Dies ist notwendig, um eine einheitliche Darstellung der Feldbus-Objekte im Datentyp INT zu bekommen, weil einige Regler Parameter für die Positionierung ([▷CANopen Offset◁](#) auf Seite 29) als Datentyp USIGN realisiert sind. Für den Anwender sind also in der Positionierung die vorhandenen FBOs als Datentyp INT zu sehen.

Das Berücksichtigen des Offsets 2^{31} kann bei Bedarf deaktiviert werden. Hier in ProDrive/WinBASS II unter „Optionsmodul G, H - Konfiguration 1“.

Bit 5 -> 0:

Zahlenskala Wandlung von U32 auf den Datentyp INT32, bei der Positionierung wird ein Offset von 2^{31} je nach Richtung auf die zugehörigen FBO hinzu gerechnet.

Bit 5 -> 1:

Es wird kein Offset benutzt.

HINWEIS



Die Umrechnung im Positionier Mode auf P607 „Absolut / Relativ CANopen“ wird nicht deaktiviert

In **P1190** mit Bit 9 ist es möglich das automatische Einstellen des Modus „Absolut / Relativ CANopen“ während der Init Phase der CANopen Optionskarte zu deaktivieren:

P1190 Bit 9 = 0 -> aktiviert

P1190 Bit 9 = 1 -> deaktiviert.

DATENAUSTAUSCH UND PARAMETRIERUNG

Der Zugriff auf Daten oder Parameter erfolgt bei CANopen **immer** über Objekte.

Entsprechend der Profilstruktur werden Objekte zur Kommunikationssteuerung (Indizes $1XXX_{\text{hex}}$) und anwendungs- oder gerätespezifische Objekte unterschieden. Die letzteren gliedern sich in Objekte nach Profil DSP 402 (Indizes $6XXX_{\text{hex}}$) und herstellerspezifische Objekte (Indizes $4XXX_{\text{hex}}$). Eine Auflistung der $6XXX$ -er und $4XXX$ -er Objekte finden Sie in [►Anhang B - Kurzreferenz◀](#) ab Seite 95.

Wichtig:

Bei den herstellerspezifischen Objekten ($4XXX_{\text{hex}}$) ergibt sich der Objektindex aus

$4000_{\text{hex}} + b \text{ maXX}^{\text{®}}$ 4400-Parameternummer in Hexadezimal,

so ist z. B. das Objekt $412C_{\text{hex}}$ auf den $b \text{ maXX}^{\text{®}}$ 4400-Parameter 300, dem Controlword umgesetzt. Diese Objekte besitzen nur den Subindex 00_{hex} .

Objekt = $4000_{\text{hex}} + \text{P-Nummer}_{(\text{hex})}$

5.1 Verzeichnis der Objekte zur Kommunikationssteuerung

In diesem Abschnitt finden Sie alle vom Baumüller CANopen-Optionsmodul unterstützten Objekte des kommunikationsspezifischen Bereiches des Objektverzeichnis nach DS301.

Name	Index	Subindex	Datentyp	Defaultwert
Device Type	1000_{hex}	00_{hex}	U32	$XX020192_{\text{hex}}$

Dieses Objekt kann nur gelesen werden und enthält die Information, um welches Gerät es sich handelt (Drive nach DSP 402), nur lesbar.

5.1 Verzeichnis der Objekte zur Kommunikationssteuerung

Bit 31 .. 24 Herstellerspezifische Objekte:

Bit 25	Bit 24	Optionskarte für:
0	0	BKF/BKD 7000
0	1	V-Regler
1	0	b maXX® 4400

Name	Index	Subindex	Datentyp	Defaultwert
Error Register	1001 _{hex}	00 _{hex}	U8	0

Dieses Objekt kann nur gelesen werden. Das Objekt 1001_{hex} enthält eine Fehlerbitleiste, die folgende Bedeutung hat:

Bit	Bedeutung
0	Fehler aufgetreten, Allgemeiner-Fehler
1	Strom-Fehler
2	Spannungs-Fehler
3	Temperatur-Fehler
4	CAN - Kommunikations-Fehler
5	Geräteprofilspezifischer-Fehler
6	Nicht verwendet
7	Herstellerspezifische-Fehler

COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
80 _{hex} + Adresse	08 _{hex}	Emergency Error Code		Error Register	Herstellerspezifisches Fehlerfeld				

EMCY-Telegramm für Error Reset / No Error

Name	Index	Subindex	Datentyp	Defaultwert
Manufacturer Status Register	1002 _{hex}	00 _{hex}	U32	-

Dieses Objekt kann nur gelesen werden. Das Low Byte enthält das Low Byte des Regler-Statuswortes aus Parameter **P0301**.

Name	Index	Subindex	Datentyp	Defaultwert
Predefined Error Field	1003 _{hex}	00 _{hex}	U8	-
Neuester Fehler		01 _{hex}	U32	-
:		:	:	:
Ältester Fehler		0F _{hex}	U32	-

Dieses Objekt kann nur auf Subindex 00_{hex} beschrieben werden. Es beinhaltet eine Fehlerhistory von max. 15 aufgetretenen Fehlern, wobei in Subindex 01_{hex} der zeitlich zuletzt aufgetretene Fehler steht.

In Subindex 00_{hex} dieses Objektes steht die Anzahl der eingetragenen Fehler. Durch Schreiben von „00_{hex}“ auf Subindex 00_{hex} wird die Liste gelöscht.

XXXX_{hex} : XXXX_{hex}
 Emergency Code : herstellerspezifisch

Die Bedeutung der Fehlernummern finden Sie in [►Umsetzung der Fehlermeldungen nach DSP 402 V1.1◀](#) ab Seite 80 erläutert.

Name	Index	Subindex	Datentyp	Defaultwert
Number of PDOs supported	1004 _{hex}	00 _{hex}	U32	00040004 _{hex}
Synchronous PDOs		01 _{hex}	U32	00040004 _{hex}
Asynchronous PDOs		02 _{hex}	U32	00040004 _{hex}

Dieses Objekt ist nur lesbar. Subindex 00_{hex} enthält die Gesamtzahl der unterstützten PDOs, wobei im High-Word die Anzahl der Receive-PDOs und im Low-Word die Zahl der Transmit-PDO steht. Subindex 01_{hex} enthält die mögliche Anzahl an synchronen PDOs, Subindex 02_{hex} die mögliche Anzahl an asynchronen PDOs.

Die eingetragenen Werte bedeuten, dass vier Receive-PDOs und vier Transmit-PDOs zur Verfügung stehen, wobei jede der PDOs als synchron oder asynchron definiert werden kann.

Name	Index	Subindex	Datentyp	Defaultwert
COB-ID SYNC-Message	1005 _{hex}	00 _{hex}	U32	80 _{hex}

Dieses Objekt enthält Informationen über das Sync-Verhalten des Slaves. Der Slave ist kein SYNC-Master, d. h. es können nur SYNC-Telegramme empfangen werden. Die unteren 11 Bit im Low-Word geben den Identifier des SYNC-Telegramms an (80_{hex}), nur lesbar.

Name	Index	Subindex	Datentyp	Defaultwert
Communication Cycle Period	1006 _{hex}	00 _{hex}	U32	0

Falls das Sync-Telegramm aktiviert ist, muss das Sync-Interval auf die Zeit des Sync-Telegramms (1000, 2000, 4000 oder 8000 µs) eingestellt werden. Die eingestellte Zeit wirkt sich auf den Parameter **P0532** (Sync-Intervall) des b maXX[®] Reglers aus.

Name	Index	Subindex	Datentyp	Defaultwert
Synchronous Window Length	1007 _{hex}	00 _{hex}	U32	0

Dieses Objekt wird nicht ausgewertet.

Name	Index	Subindex	Datentyp	Defaultwert
Manufacturer Device Name	1008 _{hex}	00 _{hex}	VString	-

Dieses Objekt ist nur lesbar. Es enthält die Zeichenfolge: „b maXX 4400“.

Name	Index	Subindex	Datentyp	Defaultwert
Manufacturer Hardware Version	1009 _{hex}	00 _{hex}	VString	-

Dieses Objekt ist nur lesbar. Es enthält die aktuelle Hardware-Version des Optionsmoduls, z. B. die Zeichenfolge: „HV01.00“.

Name	Index	Subindex	Datentyp	Defaultwert
Manufacturer Software Version	100A _{hex}	00 _{hex}	VString	-

Dieses Objekt ist nur lesbar. Es enthält die aktuelle Software-Version des Optionsmoduls, z. B. die Zeichenfolge: „SV01.00“.

Name	Index	Subindex	Datentyp	Defaultwert
Node-ID	100B _{hex}	00 _{hex}	U32	Adresse

Dieses Objekt ist nur lesbar. Es enthält die aktuell eingestellte CANopen Knotenadresse (Node-ID). Es ist nur das Low Byte gültig.

Name	Index	Subindex	Datentyp	Defaultwert
Guard Time	100C _{hex}	00 _{hex}	U16	0

In diesem Objekt wird die Zeitbasis für das Node Guarding in Millisekunden eingestellt. Durch beschreiben mit dem Wert „0“ wird das Node Guarding deaktiviert.

Name	Index	Subindex	Datentyp	Defaultwert
Life Time Factor	100D _{hex}	00 _{hex}	U8	0

Der Wert dieses Objektes wird mit Objekt 100C_{hex} multipliziert und ergibt damit die Zeit für das Node Guarding. Durch Beschreiben mit den Wert „0“ wird das Node Guarding deaktiviert.

Name	Index	Subindex	Datentyp	Defaultwert
COB-ID Guarding Protocol	100E _{hex}	00 _{hex}	U32	700 _{hex} + Adresse

Dieses Objekt beinhaltet den Identifier des Node Guarding Telegramms. Der Identifier kann geändert werden; nicht abspeicherbar.

Name	Index	Subindex	Datentyp	Defaultwert
Number of SDOs supported	100F _{hex}	00 _{hex}	U32	00000001 _{hex}

Dieses Objekt ist nur lesbar. Es enthält die Anzahl der unterstützten SDOs. Im High Word steht die Zahl der Client SDOs, die das Gerät unterstützt. Das Low Word enthält die Anzahl der Server SDOs des Gerätes.

Name	Index	Subindex	Datentyp	Defaultwert
Store Parameters	1010 _{hex}	00 _{hex}	U8	02 _{hex}
Save all Parameters (not supported)		01 _{hex}	U32	00 _{hex}
Save communication Parameters		02 _{hex}	U32	00 _{hex}

Dieses Objekt unterstützt das Ablegen von Parametern in einem nicht-flüchtigen Speicher. Subindex 00_{hex} ist nur lesbar und enthält den größten zu unterstützenden Subindex (hier 2). Subindex 01_{hex} wird zur Zeit nicht unterstützt. Die 01_{hex} im Subindex 02_{hex} zeigt an, dass ein Abspeichern, hier speziell von Mapping- und Kommunikations-Parameter unterstützt wird. Das Speichern - nur im Zustand PRE-OPERATIONAL möglich - wird mit dem Wert 65766173_{hex} als U32 oder mit Wert „save“ als String eingeleitet und dauert einige 100 Millisekunden an. Es werden die zuletzt eingestellten Werte für die Mapping- und Kommunikations-Parameter abgespeichert.

Wird SIX 2 gelesen, kommt der Wert 1 zurück. Dies bedeutet, dass das Speichern nicht automatisch erfolgt, sondern angestoßen werden muss.

Name	Index	Subindex	Datentyp	Defaultwert
Restore Parameters	1011 _{hex}	00 _{hex}	U8	02 _{hex}
Restore all Parameters (not supported)		01 _{hex}	U32	00 _{hex}
Restore default communication Parameters		02 _{hex}	U32	00 _{hex}

Bei diesem Objekt werden die Kommunikations-Parameter nach DSP402 zurück auf Default-Werte gesetzt. Subindex 00_{hex} ist nur lesbar und enthält den größten zu unterstützenden Subindex. Subindex 01_{hex} wird nicht unterstützt. Mit dem Wert 1_{hex} im Subindex 02_{hex} werden die Default-Werte der Mapping und Kommunikations-Parameter unterstützt. Dies geschieht nur im Zustand PRE-OPERATIONAL mit dem Eingeben des Wertes 64616F6C_{hex} (U32) oder mit der Eingabe „load“ als String - Parameter.

Die Default-Werte werden erst nach Power-Down/Power-Up übernommen.

WARNUNG (WARNING)



Folgendes **kann eintreffen**, wenn Sie diesen Warnhinweis nicht beachten:

- schwere Körperverletzung
- Tod



Die Gefahr ist: **geändertes Mapping**. *Ein geändertes Mapping bewirkt, daß andere Parameter als geplant verwendet werden und somit der Antrieb unerwartet reagieren kann.*

Verhindern Sie in Ihrer Applikation, dass das Mapping unkontrolliert verändert werden kann.



Name	Index	Subindex	Datentyp	Defaultwert
COB-ID Emergency	1014 _{hex}	00 _{hex}	U32	80 _{hex} + Adresse

Dieses Objekt enthält den Identifier des EMCY-Telegramms nicht abspeicherbar.

Name	Index	Subindex	Datentyp	Defaultwert
Producer Heartbeat Time	1017 _{hex}	00 _{hex}	U8	03 _{hex}

Mit diesem Objekt wird die zyklische Zeit des Heartbeat Telegramms eingestellt. Ist die Zeit auf Null gesetzt, wird kein Heartbeat Telegramm gesendet. Die Auflösung ist 1 ms.

Name	Index	Subindex	Datentyp	Defaultwert
Identity Object	1018 _{hex}	00 _{hex}	U8	03 _{hex}
Vendor ID		01 _{hex}	U32	15H _{hex}
Product Code		02 _{hex}	U32	350442
Revision Number		03 _{hex}	U32	siehe unten

In diesem Objekt sind einige Informationen über das Gerät enthalten.

Die Revision Number enthält den aktuellen Stand der Firmware z. B. 00030002 für FW 6.1294.03.02.

Name	Index	Subindex	Datentyp	Defaultwert
Server SDO Parameter	1200 _{hex}	00 _{hex}	U8	02 _{hex}
COB-ID Client ⇒ Server		01 _{hex}	U32	NODEID + 600 _{hex}
COB-ID Server ⇒ Client		02 _{hex}	U32	NODEID + 580 _{hex}

Dieses Objekt ist Read Only.

Defaultwert bedeutet hier auch gleich Default Mapping nach DSP402

Name	Index	Subindex	Datentyp	Defaultwert
1. Receive PDO Parameter	1400 _{hex}	00 _{hex}	U8	02 _{hex}
COD-ID used by PDO		01 _{hex}	U32	200 _{hex} + Adresse
Transmission Type		02 _{hex}	U8	FF _{hex}

Dieses Objekt enthält Informationen zur Receive-PDO1. Subindex 00_{hex} ist nur lesbar. In Subindex 01_{hex} wird der Identifier der Receive-PDO1 eingetragen. Subindex 02_{hex} enthält den Triggertyp dieser PDO.

Name	Index	Subindex	Datentyp	Defaultwert
2. Receive PDO Parameter	1401 _{hex}	00 _{hex}	U8	02 _{hex}
COD-ID used by PDO		01 _{hex}	U32	300 _{hex} + Adresse
Transmission Type		02 _{hex}	U8	FF _{hex}

Dieses Objekt enthält Informationen zur Receive-PDO2. Subindex 00_{hex} ist nur lesbar. In Subindex 01_{hex} wird der Identifier der Receive-PDO2 eingetragen. Subindex 02_{hex} enthält den Triggertyp dieser PDO.

Name	Index	Subindex	Datentyp	Defaultwert
3. Receive PDO Parameter	1402 _{hex}	00 _{hex}	U8	02 _{hex}
COD-ID used by PDO		01 _{hex}	U32	400 _{hex} + Adresse
Transmission Type		02 _{hex}	U8	FF _{hex}

Dieses Objekt enthält Informationen zur Receive-PDO3. Subindex 00_{hex} ist nur lesbar. In Subindex 01_{hex} wird der Identifier der Receive-PDO3 eingetragen. Subindex 02_{hex} enthält den Triggertyp dieser PDO.

Name	Index	Subindex	Datentyp	Defaultwert
4. Receive PDO Parameter	1403 _{hex}	00 _{hex}	U8	02 _{hex}
COD-ID used by PDO		01 _{hex}	U32	500 _{hex} + Adresse
Transmission Type		02 _{hex}	U8	FF _{hex}

Dieses Objekt enthält Informationen zur Receive-PDO4. Subindex 00_{hex} ist nur lesbar. In Subindex 01_{hex} wird der Identifier der Receive-PDO4 eingetragen. Subindex 02_{hex} enthält den Triggertyp dieser PDO.

Name	Index	Subindex	Datentyp	Defaultwert
1. Receive PDO Mapping	1600 _{hex}	00 _{hex}	U8	1
		01 _{hex}	U32	60400010 _{hex}
		:	:	
		n _{hex}	U32	

Dieses Objekt enthält den Inhalt der Receive-PDO1. In Subindex 00_{hex} steht die Gesamtzahl der nachfolgenden Einträge. In Subindex 01_{hex} ist defaultmäßig das Steuerwort (Objekt 6040_{hex} Subindex 00_{hex} Länge 10_{hex} Bit) eingetragen. Die Gesamtzahl der gemappten Bytes darf den CAN-Telegrammrahmen von max. acht Byte nicht überschreiten (siehe auch [PDO-Mapping](#) ab Seite 61).

Name	Index	Subindex	Datentyp	Defaultwert
2. Receive PDO Mapping	1601 _{hex}	00 _{hex}	U8	02 _{hex}
		01 _{hex}	U32	60400010 _{hex}
		02 _{hex}	U32	60600008 _{hex}
		:	:	
		n _{hex}	U32	

Dieses Objekt enthält den Inhalt der Receive-PDO2. In Subindex 00_{hex} steht die Gesamtzahl der nachfolgenden Einträge. In Subindex 01_{hex} ist defaultmäßig das Steuerwort (Objekt 6040_{hex} Subindex 00_{hex} Länge 10_{hex} Bit) eingetragen. In Subindex 02_{hex} steht defaultmäßig das Objekt der Sollbetriebsart (Index 6060_{hex} Subindex 00 Länge 08_{hex} Bit) Die Gesamtzahl der gemappten Bytes darf den CAN-Telegrammrahmen von max. acht Byte nicht überschreiten (siehe auch [▷PDO-Mapping◁](#) ab Seite 61).

Name	Index	Subindex	Datentyp	Defaultwert
3. Receive PDO Mapping	1602 _{hex}	00 _{hex}	U8	02 _{hex}
		01 _{hex}	U32	60400010 _{hex}
		02 _{hex}	U32	607A0020 _{hex}
		:	:	
		n _{hex}	U32	

Dieses Objekt enthält den Inhalt der Receive-PDO3. In Subindex 00_{hex} steht die Gesamtzahl der nachfolgenden Einträge. In Subindex 01_{hex} ist defaultmäßig das Steuerwort (Objekt 6040_{hex} Subindex 00_{hex} Länge 10_{hex} Bit) eingetragen. In Subindex 02_{hex} steht defaultmäßig das Objekt der Zielposition 1 (Index 607A_{hex} Subindex 00 Länge 20_{hex} Bit) Die Gesamtzahl der gemappten Bytes darf den CAN-Telegrammrahmen von max. acht Byte nicht überschreiten (siehe auch [▷PDO-Mapping◁](#) ab Seite 61).

Name	Index	Subindex	Datentyp	Defaultwert
4. Receive PDO Mapping	1603 _{hex}	00 _{hex}	U8	02 _{hex}
		01 _{hex}	U32	60400010 _{hex}
		02 _{hex}	U32	60FF0020 _{hex}
		:	:	
		n _{hex}	U32	

Dieses Objekt enthält den Inhalt der Receive-PDO4. In Subindex 00_{hex} steht die Gesamtzahl der nachfolgenden Einträge. In Subindex 01_{hex} ist defaultmäßig das Steuerwort (Objekt 6040_{hex} Subindex 00_{hex} Länge 10_{hex} Bit) eingetragen. In Subindex 02_{hex} steht defaultmäßig das Objekt der Drehzahl-Stellgröße (Index 60FF_{hex} Subindex 00 Länge 20_{hex} Bit) Die Gesamtzahl der gemappten Bytes darf den CAN-Telegrammrahmen von max. acht Byte nicht überschreiten (siehe auch [▷PDO-Mapping◁](#) ab Seite 61).

Name	Index	Subindex	Datentyp	Defaultwert
1. Transmit PDO Parameter	1800 _{hex}	00 _{hex}	U8	05 _{hex}
COD-ID used by PDO		01 _{hex}	U32	180 _{hex} + Adresse
Transmission Type		02 _{hex}	U8	FF _{hex}
Inhibit Time		03 _{hex}	U16	00 _{hex}
CMS-Priority Group		04 _{hex}	U8	03 _{hex}
Event Timer		05 _{hex}	U16	00 _{hex}

Dieses Objekt enthält Informationen zur Transmit-PDO1. Subindex 00_{hex} ist nur lesbar. In Subindex 01_{hex} wird der Identifier der Transmit-PDO1 eingetragen. Subindex 02_{hex} enthält den Triggertyp dieser PDO. In Subindex 03_{hex} wird die inhibit time eingestellt, die die minimale Verzögerungszeit für ein Transmissionsintervall darstellt. Der Eingabewert ist definiert als ein Multiplikator von 100µs. Subindex 04_{hex} wird nicht verwendet. Subindex 05_{hex} dient der Zeiteinstellung für timergetriggerte Sende-PDOs. Die Auflösung ist 1 Millisekunde.

Name	Index	Subindex	Datentyp	Defaultwert
2. Transmit PDO Parameter	1801 _{hex}	00 _{hex}	U8	04 _{hex}
COD-ID used by PDO		01 _{hex}	U32	280 _{hex} + Adresse
Transmission Type		02 _{hex}	U8	FF _{hex}
Inhibit Time		03 _{hex}	U16	00 _{hex}
CMS-Priority Group		04 _{hex}	U8	03 _{hex}
Event Timer		05 _{hex}	U16	00 _{hex}

Dieses Objekt enthält Informationen zur Transmit-PDO2. Subindex 00_{hex} ist nur lesbar. In Subindex 01_{hex} wird der Identifier der Transmit-PDO2 eingetragen. Subindex 02_{hex} enthält den Triggertyp dieser PDO. In Subindex 03_{hex} wird die inhibit time eingestellt, die die minimale Verzögerungszeit für ein Transmissionsintervall darstellt. Der Eingabewert ist definiert als ein Multiplikator von 100µs. Subindex 04_{hex} wird nicht verwendet. Subindex 05_{hex} dient der Zeiteinstellung für timergetriggerte Sende-PDOs. Die Auflösung ist 1 Millisekunde.

Name	Index	Subindex	Datentyp	Defaultwert
3. Transmit PDO Parameter	1802 _{hex}	00 _{hex}	U8	04 _{hex}
COD-ID used by PDO		01 _{hex}	U32	380 _{hex} + Adresse
Transmission Type		02 _{hex}	U8	FE _{hex}
Inhibit Time		03 _{hex}	U16	00 _{hex}
CMS-Priority Group		04 _{hex}	U8	03 _{hex}
Event Timer		05 _{hex}	U16	00 _{hex}

Dieses Objekt enthält Informationen zur Transmit-PDO3. Subindex 00_{hex} ist nur lesbar. In Subindex 01_{hex} wird der Identifier der Transmit-PDO3 eingetragen. Subindex 02_{hex} enthält den Triggertyp dieser PDO. In Subindex 03_{hex} wird die inhibit time eingestellt, die die minimale Verzögerungszeit für ein Transmissionsintervall darstellt. Der Eingabewert ist definiert als ein Multiplikator von 100µs. Subindex 04_{hex} wird nicht verwendet. Subindex 05_{hex} dient der Zeiteinstellung für timergetriggerte Sende-PDOs. Die Auflösung ist 1 Millisekunde.

Name	Index	Subindex	Datentyp	Defaultwert
4. Transmit PDO Parameter	1803 _{hex}	00 _{hex}	U8	04 _{hex}
COD-ID used by PDO		01 _{hex}	U32	480 _{hex} + Adresse
Transmission Type		02 _{hex}	U8	FE _{hex}
Inhibit Time		03 _{hex}	U16	00 _{hex}
CMS-Priority Group		04 _{hex}	U8	03 _{hex}
Event Timer		05 _{hex}	U16	00 _{hex}

Dieses Objekt enthält Informationen zur Transmit-PDO4. Subindex 00_{hex} ist nur lesbar. In Subindex 01_{hex} wird der Identifier der Transmit-PDO4 eingetragen. Subindex 02_{hex} enthält den Triggertyp dieser PDO. In Subindex 03_{hex} wird die inhibit time eingestellt, die die minimale Verzögerungszeit für ein Transmissionsintervall darstellt. Der Eingabewert ist definiert als ein Multiplikator von 100 µs. Subindex 04_{hex} wird nicht verwendet. Subindex 05_{hex} dient der Zeiteinstellung für timergetriggerte Sende-PDOs. Die Auflösung ist 1 Millisekunde.

Name	Index	Subindex	Datentyp	Defaultwert
1. Transmit PDO Mapping	1A00 _{hex}	00 _{hex}	U8	01 _{hex}
		01 _{hex}	U32	60410010 _{hex}
		:	:	
		n _{hex}		

Dieses Objekt enthält den Inhalt der Transmit-PDO1. In Subindex 00_{hex} steht die Gesamtzahl der nachfolgenden Einträge. In Subindex 01_{hex} ist defaultmäßig das Statuswort (Objekt 6041_{hex} Subindex 00_{hex} Länge 10_{hex} Bit) eingetragen. Die Gesamtzahl der gemappten Bytes darf den CAN-Telegrammrahmen von max. acht Byte nicht überschreiten (siehe auch [▷PDO-Mapping◁](#) ab Seite 61).

Name	Index	Subindex	Datentyp	Defaultwert
2. Transmit PDO Mapping	1A01 _{hex}	00 _{hex}	U8	02 _{hex}
		01 _{hex}	U32	60410010 _{hex}
		02 _{hex}	U32	60610008 _{hex}
		:	:	
		n _{hex}	U32	

Diese Objekt enthält den Inhalt der Transmit-PDO2. In Subindex 00_{hex} steht die Gesamtzahl der nachfolgenden Einträge. In Subindex 01_{hex} ist defaultmäßig das Statuswort (Objekt 6041_{hex} Subindex 00_{hex} Länge 10_{hex} Bit) eingetragen. In Subindex 02_{hex} steht defaultmäßig das Objekt der Ist-Betriebsart (Index 6061_{hex} Subindex 00 Länge 08_{hex} Bit). Die Gesamtzahl der gemappten Bytes darf den CAN-Telegrammrahmen von max. acht Byte nicht überschreiten (siehe auch [▷PDO-Mapping◁](#) ab Seite 61).

Name	Index	Subindex	Datentyp	Defaultwert
3. Transmit PDO Mapping	1A02 _{hex}	00 _{hex}	U8	02 _{hex}
		01 _{hex}	U32	60410010 _{hex}
		02 _{hex}	U32	60640020 _{hex}
		:	:	
		n _{hex}	U32	

Dieses Objekt enthält den Inhalt der Transmit-PDO3. In Subindex 00_{hex} steht die Gesamtzahl der nachfolgenden Einträge. In Subindex 01_{hex} ist defaultmäßig das Statuswort (Objekt 6041_{hex} Subindex 00_{hex} Länge 10_{hex} Bit) eingetragen. In Subindex 02_{hex} steht defaultmäßig das Objekt der Position-Istwert (Index 6064_{hex} Subindex 00 Länge 20_{hex} Bit). Die Gesamtzahl der gemappten Bytes darf den CAN-Telegrammrahmen von max. acht Byte nicht überschreiten (siehe auch [▷PDO-Mapping◁](#) ab Seite 61).

Name	Index	Subindex	Datentyp	Defaultwert
4. Transmit PDO Mapping	1A03 _{hex}	00 _{hex}	U8	02 _{hex}
		01 _{hex}	U32	60410010 _{hex}
		02 _{hex}	U32	606C0020 _{hex}
		n _{hex}	U32	

Diese Objekt enthält den Inhalt der Transmit-PDO4. In Subindex 00_{hex} steht die Gesamtzahl der nachfolgenden Einträge. In Subindex 01_{hex} ist defaultmäßig das Statuswort (Objekt 6041_{hex} Subindex 00_{hex} Länge 10_{hex} Bit) eingetragen. In Subindex 02_{hex} steht defaultmäßig das Objekt der Drehzahl-Istwert (Index 606C_{hex} Subindex 00 Länge 20_{hex} Bit). Die Gesamtzahl der gemappten Bytes darf den CAN-Telegrammrahmen von max. acht Byte nicht überschreiten (siehe auch [PDO-Mapping](#) ab Seite 61).

5.2 Netzwerkmanagement (NMT)

Kommandos des Netzwerkmanagements dienen vorwiegend der Steuerung der Kommunikationszustände im CANopen-Netz.

5.2.1 Zustandsmaschine der Kommunikation

Hier ist das Zustandsdiagramm der Kommunikation des CANopen Slaves dargestellt.

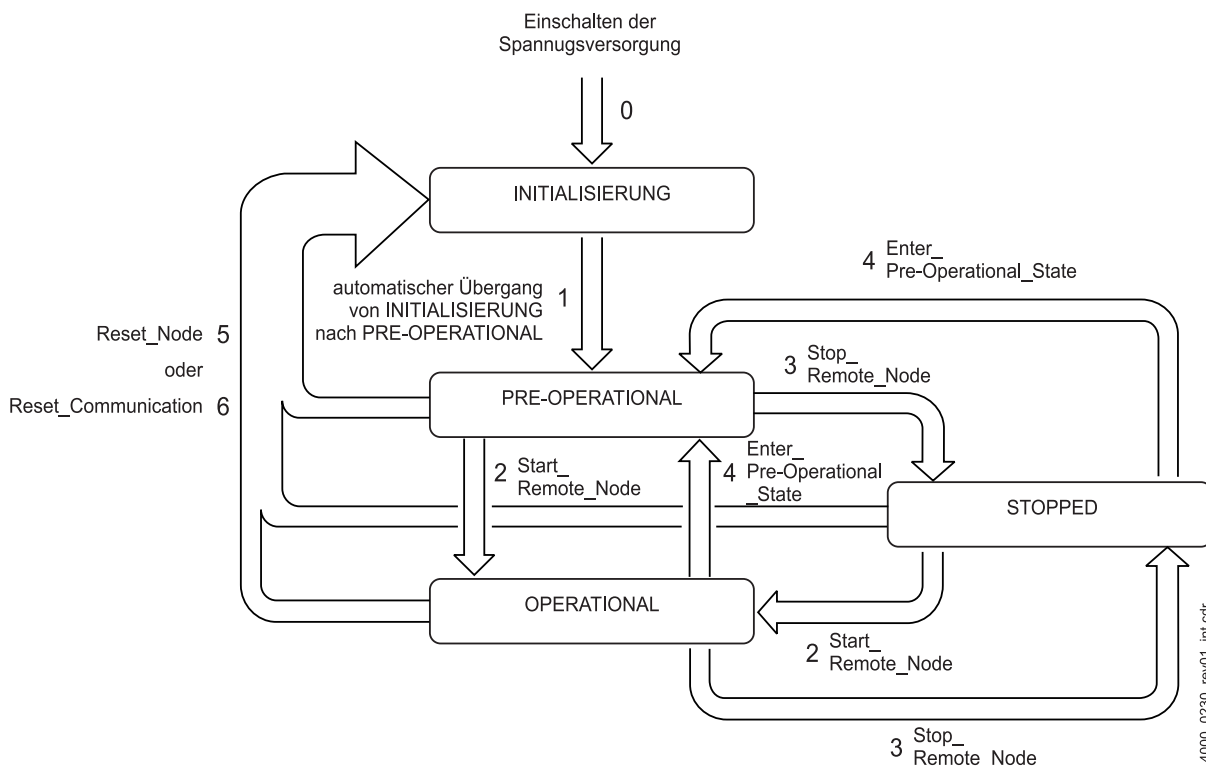


Abbildung 3: Zustandsmaschine der Kommunikation

5.2 Netzwerkmanagement (NMT)

Nach der INITIALISIERUNG (ausgelöst durch das Einschalten des Gerätes) wird automatisch der Zustand PRE-OPERATIONAL erreicht. Befindet sich ein Slave in diesem Status, kann er über SDOs konfiguriert werden. Der Datenaustausch über PDOs ist nicht möglich.

Im Zustand STOPPED ist nur das Node Guarding aktiviert. Weder SDOs noch PDOs können gesendet oder empfangen werden.

Im Zustand OPERATIONAL (normaler Betriebszustand) sind PDO- und SDO-Datenaustausch, sowie das Node Guarding möglich.

Die einzelnen Zustandsübergänge werden von einem NMT-Master initiiert. Das Baumüller CANopen-Optionsmodul kann folgende NMT-Kommandos verarbeiten:

1 automatischer Übergang von INITIALISIERUNG nach PRE-OPERATIONAL

Hinweis:

Beim Übergang von Pre-Operational nach Operational werden die Parameternummern dem Mapping zugeordnet. Diese Zuordnung ist zeitaufwändig und kann mehrere Millisekunden dauern, solange wird keine PDO gesendet und auch keine RX-PDO bearbeitet.

Es können nicht 3 NMT Befehle innerhalb von 15 ms bearbeitet werden.

2 Start_Remote_Node

3 Stop_Remote_Node

4 Enter_Pre-Operational_State

5 Reset_Node

6 Reset_Communication

Es können nicht 3 NMT Befehle innerhalb von 8 ms bearbeitet werden.

5.2.2 Telegramme

NMT-Telegramme zur Kommunikationssteuerung besitzen nach dem Predefined Connection Set den Default-Identifizier „0“ (siehe auch [►Grundlagen CAN◄](#) ab Seite 10).

5.2.2.1 Zustandssteuerung

Je NMT-Telegramm werden zwei Datenbyte übertragen. Das Datenbyte 0 enthält den Command Specifier CS, das Datenbyte 1 enthält die Geräteadresse. Ist die Adresse 0 eingetragen, so werden mit dem entsprechenden Kommando alle Knoten angesprochen (Broadcast).

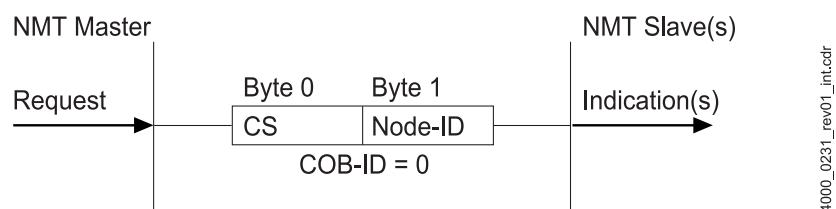


Abbildung 4: NMT-Telegramm zur Steuerung der Kommunikationszustände

CS	Bezeichnung	Wirkung
1	Start_Remote_Node	Starten des Normalbetriebes
2	Stop_Remote_Node	Deaktivieren der PDO- und SDO-Kommunikation
128	Enter_Pre-Operational_State	Übergang in Konfigurationsmodus
129	Reset_Node	Kontrolliertes Rücksetzen des gesamten Objektverzeichnis' auf Default-Werte
130	Reset_Communication	Rücksetzen des Kommunikationsteils im Objektverzeichnis auf Default-Werte

Ein Telegramm, welches den Knoten 16 in den Konfigurationsmodus bringt, sieht wie folgt aus:

COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
00 _{hex}	02 _{hex}	80 _{hex}	10 _{hex}						

Diese Telegramme sind unbestätigt, d. h. kein NMT-Slave quittiert dem NMT-Master die korrekt empfangene Nachricht.

HINWEIS



Der b maXX[®] 4400 sieht z. Zt. keinen Geräte-Reset per Software vor.

Nach dem Einschalten der Versorgungsspannung und einem Reset meldet sich der CANopen-Slave mit dem Boot Up Telegramm (siehe auch [▶Boot Up◀](#) ab Seite 46). Die gesamte Resetsequenz dauert vom Empfang des Kommandos Reset_Node bis zur Rückmeldung mittels Boot Up Telegramm einige wenige Sekunden.

WARNUNG (WARNING)



Folgendes **kann eintreffen**, wenn Sie diesen Warnhinweis nicht beachten:

- schwere Körperverletzung
- Tod



Die Gefahr ist: **mechanische und elektrische Einwirkung**. Wird im laufenden zyklischen Betrieb ein Reset ausgelöst, kann dies zu unerwünschten Zuständen in der Applikation führen, da im Regler der Bootdatensatz geladen und auf der CANopen-Optionskarte das zuletzt im Flash abgespeicherte (mit dem Objekt 1010_{hex} nach DS301) Default-Mapping gesetzt wird.



Prüfen Sie das Mapping nach jedem Reset.

5.2.2.2 Boot Up

Standard wird das Boot-Up Verhalten nach DS 301 V4

Zur Unterscheidung des Boot-Up Telegramms nach DS 301 V3 oder V4 kann der Parameter „Optionsmodul G - Konfiguration 1“ oder „Optionsmodul H Konfiguration 1“ b maXX[®] Regler eingestellt werden. Abhängig in welchem Steckplatz die CANopen Optionskarte gesteckt wird.

Bit 0 ⇒ 0: nach DS 301 V4 (Default)

Bit 0 ⇒ 1: nach DS 301 V3

Boot-Up nach DS 301 V4,

Boot-Up Telegramm mit ID = 700_{hex} + Node-ID, DLC = 1 Byte 0 mit den Daten = 0 gefüllt.

COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
701 _{hex}	01 _{hex}	00 _{hex}							

DSP 301 V4

Boot-Up nach DS 301 V3, ID = 80_{hex} + Node-ID, DLC = 0 und anschliessenden

ID = 80_{hex} + Node-ID, DLC = 8 Byte 0 - 7 mit den Daten = 0 aufgefüllt (Reset des Error Register).

COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
81 _{hex}	00 _{hex}								
81 _{hex}	08 _{hex}	00 _{hex}	00 _{hex}	00 _{hex}	00 _{hex}	00 _{hex}	00 _{hex}	00 _{hex}	00 _{hex}

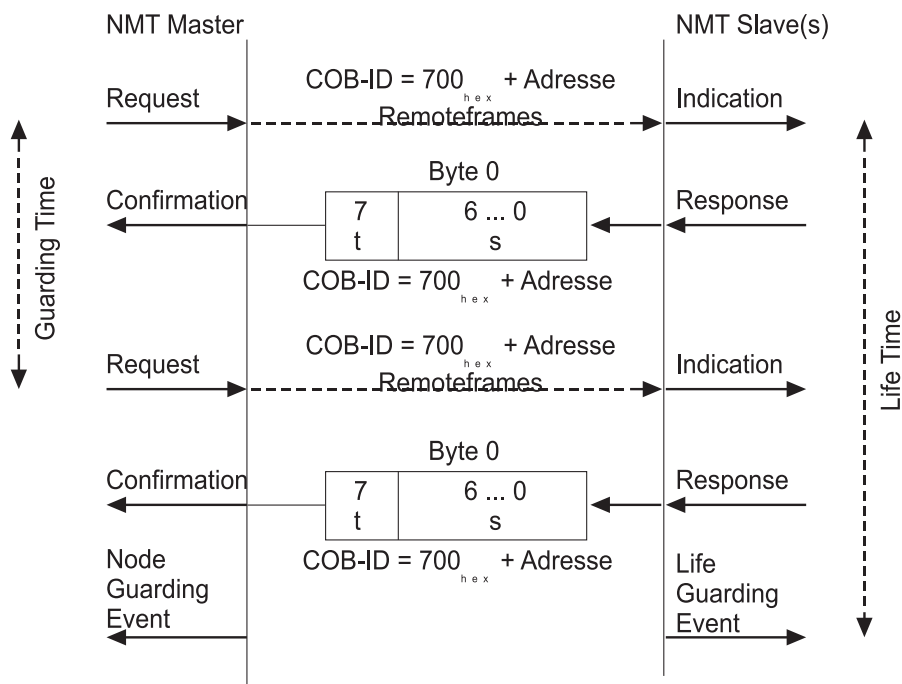
DS 301 V3

5.2.3 Node Guarding

Das Node Guarding dient der Überwachung der Slave durch den Master. Gleichzeitig kann der Slave den Master überwachen (Life Guarding).

Der Master fragt in bestimmten Intervallen durch Remoteframes die Slaves ab. Remoteframes sind spezielle Telegramme, mit denen es möglich ist, Datentelegramme anzufordern. Remoteframes besitzen den gleichen COB-ID wie das zugehörige Datentelegramm, weisen jedoch eine Datenlänge von 1 Byte auf. Zur Unterscheidung zwischen Remote- und Datentelegramm (Telegrammunterscheidung erfolgt normalerweise durch die COB-ID), dient im Steuerfeld des Remotetelegramms das sogenannte RTR-Bit. Im Remoteframe steht das RTR-Bit auf „1“, im Datentelegramm auf „0“.

Die COB-ID ergibt sich aus 700_{hex} + Adresse, laut Predefined Connection Set. Diese COB-ID kann auch geändert werden. Das dafür erforderliche Objekt ist 100E_{hex}.



4000_0232_rev02_int.cdr

Abbildung 5: Node Guarding Protokoll

In den Objekten 100C_{hex} und 100D_{hex} wird die Guarding-Zeit eingestellt. Innerhalb dieser Zeit muss der Slave vom Master eine Guarding-Anforderung (Remotetelegramm) empfangen haben. Ist dies nicht der Fall, tritt das Life Guarding Event im Slave ein. Dadurch wechselt der Slave in den Zustand PRE-OPERATIONAL und im Regler wird die in Objekt 6007_{hex} vorgegebene Reaktion ausgelöst.

Bleibt die Antwort des Slaves innerhalb einer bestimmten Zeit aus, wird im Master das Node Guarding Event ausgelöst. Sind keine Zeiten eingestellt, antwortet der Slave auf jedes RTR, jedoch ohne Überwachung der Lifetime

Aus der Antwort des Slaves auf eine Node-Guarding-Anforderung des Masters ist der aktuelle Kommunikationszustand des Slaves erkennbar. Das Antworttelegramm besteht aus einem Datenbyte (siehe auch [Abbildung 5](#) auf Seite 47). Je nach Kommunikationszustand unterscheidet sich das Feld „s“. Außerdem wird bei zwei aufeinanderfolgenden Telegrammen das Togglebit „t“ verändert.

Kommunikationsphase	Kennung s	Resultierende Daten mit	
		t = 0	t = 1
PRE-OPERATIONAL	7F _{hex} (127)	7F _{hex} (127)	FF _{hex} (255)
OPERATIONAL	05 _{hex} (5)	05 _{hex}	85 _{hex} (133)
STOPPED	04 _{hex} (4)	04 _{hex}	84 _{hex} (132)

Das Node Guarding ist in allen Kommunikationsphasen verfügbar. Das Togglebit wird nur in der Phase INITIALISIERUNG auf seinen Defaultwert zurückgesetzt. Dies bedeutet, dass auch bei Zustandswechseln der Togglemechanismus weitergeführt wird.

Gestartet wird das Node Guarding im Slave nach Empfang des ersten Guarding-Anforderungstelegramms. Ab diesem Zeitpunkt läuft im Slave die in den Objekten 100C_{hex} und 100D_{hex} parametrisierte Überwachungszeit.

HINWEIS



Die Node Guarding Zeit sollte mindestens 1,5 mal größer eingestellt werden als die vom Master gesendeten Remotetelegramme.

5.2.4 Heartbeat Protocol

Das Heartbeat Protocol dient der Überwachung des/der Slaves durch den Master. Anders als beim Node Guarding gibt es keine RTR-Frames, sondern der Slave sendet zyklisch Heartbeat Messages. Einer oder mehrere Heartbeat Consumer empfangen bzw. überwachen die Heartbeat Messages. Wird die Heartbeat Message nicht innerhalb der eingestellten Heartbeat Zeit gesendet, löst der Master (Heartbeat Consumer) ein Heartbeat Ereignis aus. Beim Slave wird die Heartbeat Time im FB-Objekt 1017_{hex} eingestellt. Die Auflösung beträgt 1 Millisekunde.

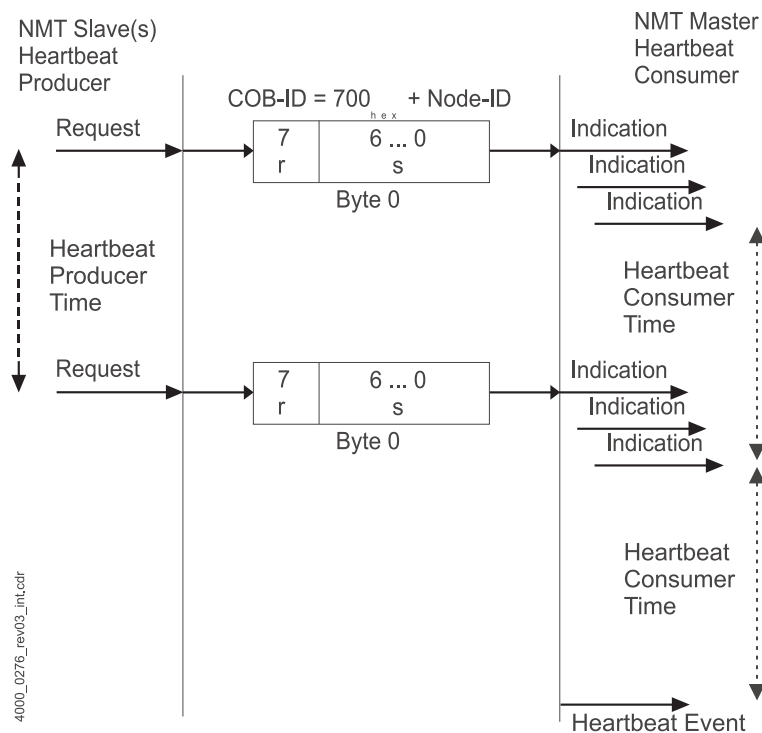


Abbildung 6: Heartbeat Protokoll

Aus der Heartbeat Message des Slaves ist der aktuelle Kommunikationszustand des Slaves erkennbar. Das Heartbeat Telegramm besteht aus einem Datenbyte

r: reserved (= 0)

s: je nach Kommunikationszustand unterscheidet sich das Feld „s“.

Kommunikationsphase	Kennungs
BOOT UP	00 _{hex}
PRE-OPERATIONAL	7F _{hex} (127)
OPERATIONAL	05 _{hex} (5)
STOPPED	04 _{hex} (4)

Es wird jeweils entweder das Node Guarding oder das Heartbeat unterstützt. Ist die Heartbeat Zeit ungleich NULL, wird das Heartbeat-Protocol aktiviert.

5.3 Bedarfsdaten (SDO)

Service Daten Objekte (SDO) dienen dem Austausch von Nachrichten ohne Echtzeitanforderungen. Deshalb sind im Predefined Connection Set (siehe auch [►Grundlagen CAN](#) ab Seite 10) niederpriorie COB-IDs dafür vorgesehen. Die SDOs werden zur Parametrierung der Slaves und zur Einstellung der Kommunikationsbeziehungen für die PDOs verwendet. Der Datenzugriff erfolgt ausschließlich über die Objektliste. SDOs sind immer bestätigte Daten, d. h. der Sender erhält eine Quittung vom Empfänger. Der Datenaustausch über SDOs kann nur asynchron ablaufen (siehe auch [►Synchronisation \(SYNC\)](#) ab Seite 59).

SDOs folgen dem Client-Server-Modell. Der Client initiiert die Kommunikation und der Server antwortet darauf. Ein Server kann eine SDO-Kommunikation nicht beginnen. Das Baumüller CANopen-Optionsmodul unterstützt eine Server SDO und keine Client SDO.

5.3.1 Telegrammaufbau

Die COB-ID der Request SDO ergibt sich aus 600_{hex} + Adresse, bei Response-SDOs aus 580_{hex} + Adresse. Das Datenfeld des CAN-Datentelegramms (8 Byte) für eine SDO gliedert sich in drei Teile, einen Command Specifier CS (1 Byte), einem Multiplexor M (3 Byte) und dem eigentlichen Nutzdatenbereich D0 - D3 (4 Byte).

COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
600 _{hex} + Adresse	08 _{hex}	CS	M	M	M	D0	D1	D2	D3
580 _{hex} + Adresse	08 _{hex}	CS	M	M	M	D0	D1	D2	D3

Der Multiplexor M besteht aus dem 16 Bit Index eines Objektes und dem dazugehörigen acht Bit breiten Subindex. Bei segmentierten Telegrammen wird der Nutzdatenbereich um die drei Byte des Multiplexors erweitert, wodurch je Telegramm sieben Byte Nutzdaten übertragen werden können. Der Command Specifier CS klassifiziert die verschiedenen SDO-Typen.

5.3 Bedarfsdaten (SDO)

5.3.2 Arten des SDO-Transfers

Die Baumüller CANopen-Anschaltung unterstützt den Expedited Transfer und den Segmented Transfer, wobei letzterer nur für die Objekte 1008_{hex}, 1009_{hex} und 100A_{hex} Manufacturer Device Name genutzt wird.

Expedited Transfer

Es können Objekte geschrieben oder gelesen werden, deren Daten maximal 4 Byte umfassen. Es sind nur zwei Telegramme erforderlich, eine Anforderung und eine Antwort. Alle Objekte mit den Indizes 1XXX_{hex}, 4XXX_{hex}, 6XXX_{hex} sind über Expedited SDOs ansprechbar mit Ausnahme der Objekte 1008_{hex}, 1009_{hex} und 100A_{hex}.

Segmented Transfer

Für Objekte mit Daten größer als 4 Byte ist der Segmented Transfer erforderlich. Dabei werden die Nutzdaten auf mehrere Telegramme verteilt. Dies ist nur beim Lesen der Objekte 1008_{hex}, 1009_{hex} und 100A_{hex} nötig.

5.3.3 Objekt schreiben

Zum Schreiben von Objekten wird bei der Baumüller CANopen-Anbindung der Expedited Transfer genutzt. Ein SDO-Client (Master) sendet einen Schreib-Request an den Slave (Baumüller CANopen Anbindung). Dieser Slave führt die Anforderung aus und quittiert dies mit Response.

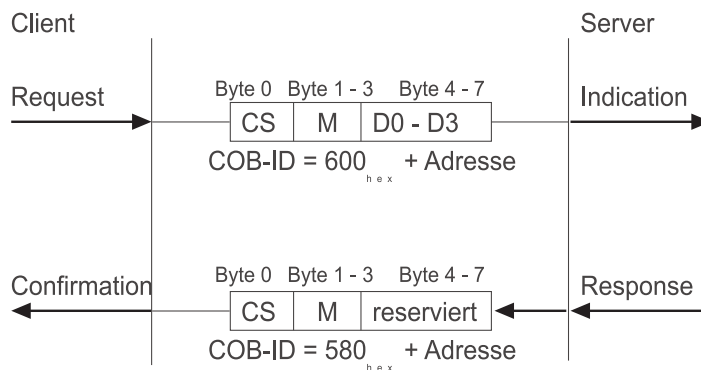


Abbildung 7: Initiate SDO Download Protocol

Der Command Specifier CS für den Request hängt von der Nutzdatenlänge ab. D0 ist das LSB, D3 das MSB des zu übertragenden Datums.

Datenlänge in D0 - D3	Command Specifier CS
1 Byte	2F _{hex}
2 Byte	2B _{hex}
4 Byte	23 _{hex}

Der Command Specifier CS für die Response beträgt 60_{hex}, der Multiplexor ist identisch zu dem des Requests, das Datenfeld ist ohne Bedeutung (reserviert).

Beispiel

Auf das Objekt 6060_{hex}, Subindex 00_{hex}, des Slaves mit der Adresse 4 soll der Wert „-3“ (FD_{hex}) geschrieben werden. Die Datenbreite dieses Objektes beträgt acht Bit.

Request

		CS	Multiplexor				D0	D1	D2	D3
COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	
604 _{hex}	08 _{hex}	2F _{hex}	60 _{hex}	60 _{hex}	00 _{hex}	FD _{hex}	00 _{hex}	00 _{hex}	00 _{hex}	

↙ Basis-Adresse 600_{hex}
+ Slave-Adresse 4_{hex}

↖ Objekt 60 60_{hex}

↗ Subindex 00_{hex}

↘ Wert -3

Response

		CS	Multiplexor				D0	D1	D2	D3
COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	
584 _{hex}	08 _{hex}	60 _{hex}	60 _{hex}	60 _{hex}	00 _{hex}	00 _{hex}	00 _{hex}	00 _{hex}	00 _{hex}	

Auf das Objekt 43E9_{hex}, Subindex 00_{hex}, des Slaves mit der Adresse 4 soll der Wert „12“ (0C_{hex}) geschrieben werden. Die Datenbreite dieses Objektes beträgt 16 Bit.

Request

		CS	Multiplexor				D0	D1	D2	D3
COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	
604 _{hex}	08 _{hex}	2B _{hex}	E9 _{hex}	43 _{hex}	00 _{hex}	0C _{hex}	00 _{hex}	00 _{hex}	00 _{hex}	

↙ Basis-Adresse 600_{hex}
+ Slave-Adresse 4_{hex}

↖ Objekt 43 E9_{hex}

↗ Subindex 00_{hex}

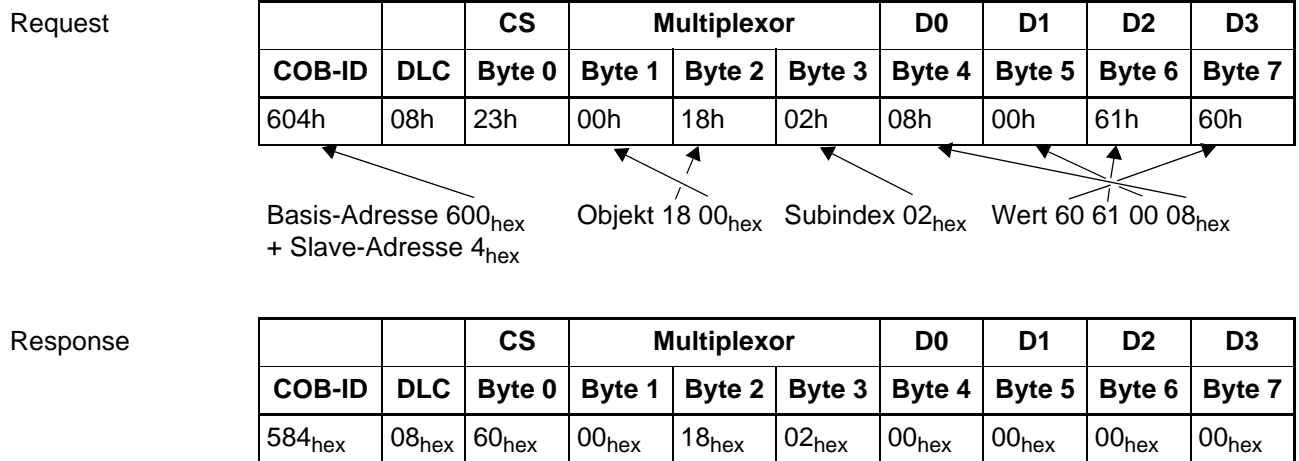
↘ Wert 12

Response

		CS	Multiplexor				D0	D1	D2	D3
COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	
584 _{hex}	08 _{hex}	60 _{hex}	E9 _{hex}	43 _{hex}	00 _{hex}	00 _{hex}	00 _{hex}	00 _{hex}	00 _{hex}	

5.3 Bedarfsdaten (SDO)

Auf das Objekt 1800_{hex} , Subindex 02_{hex} , des Slaves mit der Adresse 4 soll der Wert „ 60610008_{hex} “ geschrieben werden. Die Datenbreite dieses Objektes beträgt 32 Bit.



5.3.4 Objekt lesen

Zum Lesen von Objekten wird bei der Baumüller CANopen-Anbindung der Expedited Transfer, bei den Objekten 1008_{hex} , 1009_{hex} und $100A_{\text{hex}}$ der Segmented Transfer benutzt.

5.3.4.1 Expedited Transfer

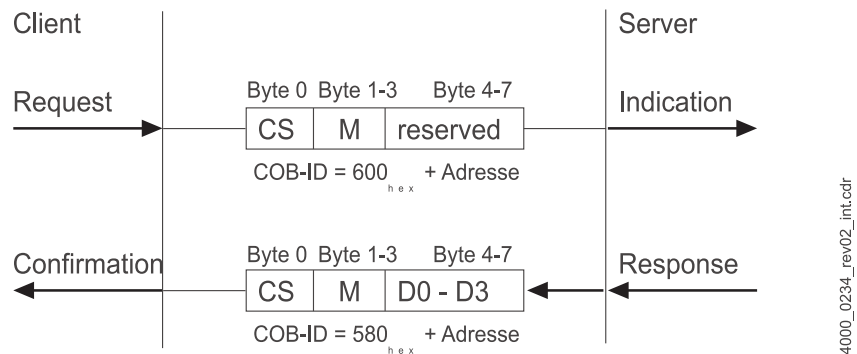


Abbildung 8: Initiate SDO Upload Expedited

Ein SDO Client (Master) sendet einen Lese-Request an den Slave (Baumüller CANopen-Anbindung). Dieser Slave führt die Anforderung aus und schickt die geforderten Daten im Antworttelegramm (Response).

Der Command Specifier CS für den Request beträgt immer 40_{hex}. Der Command Specifier CS für die Response hängt von der Nutzdatenlänge ab. D0 ist das LSB, D3 das MSB.

Datenlänge in D0 - D3	Command Specifier CS
1 Byte	4F _{hex}
2 Byte	4B _{hex}
4 Byte	43 _{hex}

Der Multiplexor von Request und Response stimmt überein.

Beispiel

Das Objekt 6061_{hex}, Subindex 00_{hex}, des Slaves mit der Adresse 4 soll gelesen werden. Die Datenbreite dieses Objektes beträgt 1 Byte.

Request

		CS	Multiplexor				D0	D1	D2	D3
COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	
604 _{hex}	08 _{hex}	40 _{hex}	61 _{hex}	60 _{hex}	00 _{hex}	00 _{hex}	00 _{hex}	00 _{hex}	00 _{hex}	

Response

		CS	Multiplexor				D0	D1	D2	D3
COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	
584 _{hex}	08 _{hex}	4F _{hex}	61 _{hex}	60 _{hex}	00 _{hex}	D0 _{hex}	00 _{hex}	00 _{hex}	00 _{hex}	

↙ Basis-Adresse 580_{hex}
+ Slave-Adresse 4_{hex}

↖ Objekt 60 61_{hex}

↗ Subindex 00_{hex}

↘ Wert Daten

Das Objekt 6041_{hex}, Subindex 00_{hex}, des Slaves mit der Adresse 4 soll gelesen werden. Die Datenbreite dieses Objektes beträgt 2 Byte.

Request

		CS	Multiplexor				D0	D1	D2	D3
COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	
604 _{hex}	08 _{hex}	40 _{hex}	41 _{hex}	60 _{hex}	00 _{hex}	00 _{hex}	00 _{hex}	00 _{hex}	00 _{hex}	

Response

		CS	Multiplexor				D0	D1	D2	D3
COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	
584 _{hex}	08 _{hex}	4B _{hex}	41 _{hex}	60 _{hex}	00 _{hex}	D0	D1	00 _{hex}	00 _{hex}	

↙ Basis-Adresse 580_{hex}
+ Slave-Adresse 4_{hex}

↖ Objekt 60 41_{hex}

↗ Subindex 00_{hex}

↘ Wert: DB high DB low

5.3 Bedarfsdaten (SDO)

Das Objekt 1400_{hex} , Subindex 01_{hex} , des Slaves mit der Adresse 4 soll gelesen werden. Die Datenbreite dieses Objektes beträgt 4 Byte.

Request

		CS	Multiplexor			D0	D1	D2	D3
COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
604_{hex}	08_{hex}	40_{hex}	00_{hex}	14_{hex}	01_{hex}	00_{hex}	00_{hex}	00_{hex}	00_{hex}

Response

		CS	Multiplexor			D0	D1	D2	D3
COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
584_{hex}	08_{hex}	43_{hex}	00_{hex}	14_{hex}	01_{hex}	D0	D1	D2	D3

Basis-Adresse 580_{hex}
+ Slave-Adresse 4_{hex}

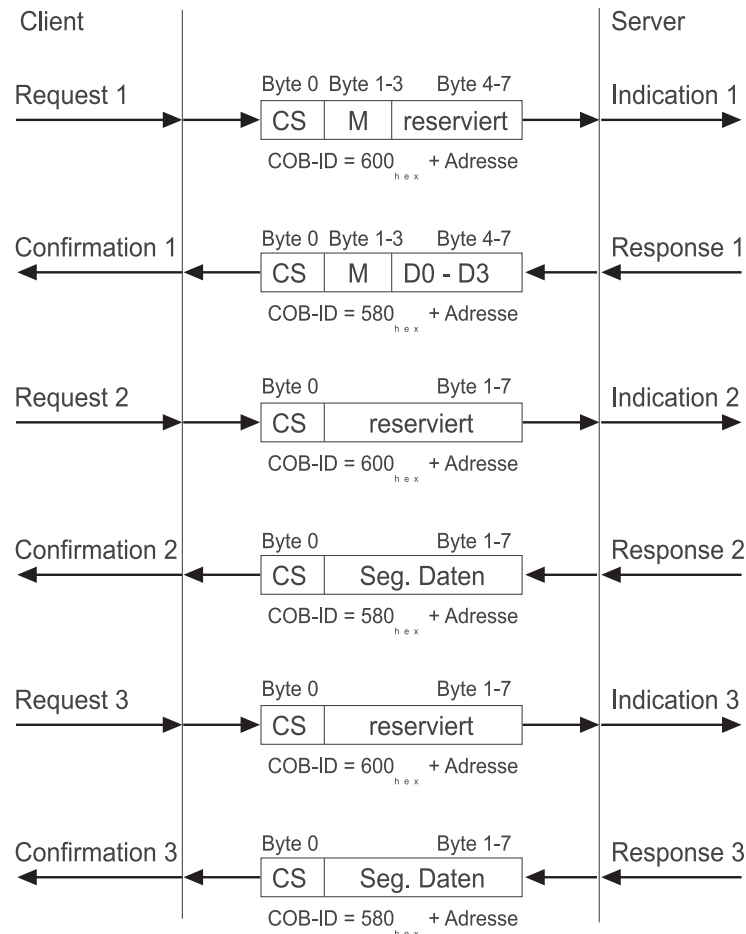
Objekt $14\ 00_{\text{hex}}$

Subindex 00_{hex}

Wert: $DB_3\ DB_2\ DB_1\ DB_0$

5.3.4.2 Segmented Transfer

Zunächst wird mit dem Initiate SDO Upload Protocol eine Leseanforderung an den Slave geschickt. Der Slave antwortet mit dem Command Specifier CS 41_{hex}. Im Datenfeld wird die Gesamtzahl der zu übertragenden Nutzdatenbytes zurückgeliefert (Request 1, Response 1). In den folgenden Zyklen werden diese Nutzdaten übertragen (Request 2, Response 2, Request 3 und Response 3).



4000_0235_rev02_int.cdr

Abbildung 9: Upload SDO Segmented Protocol

5.3 Bedarfsdaten (SDO)

Die Command Specifier enthalten ein Togglebit, dessen Wert bei jedem Transfer wechselt.

z. B. das Lesen des Objektes 1008_{hex} Manufacturer Device Name von Slave 4:

Request 1

		CS	Multiplexor				D0	D1	D2	D3
COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	
604 _{hex}	08 _{hex}	40 _{hex}	08 _{hex}	10 _{hex}	00 _{hex}	00 _{hex}	00 _{hex}	00 _{hex}	00 _{hex}	

Response 1

		CS	Multiplexor				D0	D1	D2	D3
COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	
584 _{hex}	08 _{hex}	41 _{hex}	08 _{hex}	10 _{hex}	00 _{hex}	6 _{hex}	00 _{hex}	00 _{hex}	00 _{hex}	

Byte 0 in Response 1 (Command Specifier 41_{hex}) besagt, dass das Nutzdatenfeld die Anzahl der zu übertragenden Nutzdatenbytes enthält (6).

Request 2

		CS								
COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	
604 _{hex}	08 _{hex}	60 _{hex}	00 _{hex}	00 _{hex}	00 _{hex}	00 _{hex}	00 _{hex}	00 _{hex}	00 _{hex}	

Response 2

		CS	D0	D1	D2	D3	D4	D5	D6
COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
584 _{hex}	08 _{hex}	14 _{hex}	62 _{hex}	20 _{hex}	6D _{hex}	61 _{hex}	58 _{hex}	58 _{hex}	00 _{hex}

Byte 0 in Request 2 (Command Specifier 60_{hex}) bedeutet, dass das erste Segment (die sechs Byte) übertragen werden sollen. Byte 0 in Response 2 (Command Specifier, 14_{hex}) besagt, dass das Nutzdatenfeld (sechs Byte) gültige Daten enthalten und dass dieses Segment zugleich das letzte ist.

Das Ergebnis der Übertragung lautet: „b maXX“.

5.3.5 Fehlerreaktionen

Fehlerhafte SDO-Zugriffe werden mit Abort Codes abgewiesen. Der Aufbau dieser Abort-Telegramme ist identisch zu dem in [▶Abbildung 5.3.1◀](#) auf Seite 49 dargestellten SDO-Telegramm. Das Datenfeld enthält einen Abort Code mit 4 Bytes.

Bei fehlerhaften Zugriffen auf kommunikationsspezifische Objekte (1XXX_{hex}) werden folgende Meldungen unterschieden:

Abort Code	Bedeutung
05 _{hex} 03 _{hex} 00 _{hex} 00 _{hex}	Parameter inkonsistent (Toggle Bit hat nicht gewechselt)
05 _{hex} 04 _{hex} 00 _{hex} 01 _{hex}	Client/Server Kommando Specific CS nicht gültig oder unbekannt.
06 _{hex} 01 _{hex} 00 _{hex} 02 _{hex}	Schreiben auf schreibgeschütztes Objekt
06 _{hex} 02 _{hex} 00 _{hex} 00 _{hex}	Objekt existiert nicht
06 _{hex} 04 _{hex} 00 _{hex} 41 _{hex}	Daten können nicht gemappt werden (z. B. falsche Längenangabe)
06 _{hex} 06 _{hex} 00 _{hex} 00 _{hex}	Zugriffsfehler auf die Hardware (speichern/laden vom Flash)
06 _{hex} 07 _{hex} 00 _{hex} 10 _{hex}	Inkorrekte Datenwert Länge
06 _{hex} 09 _{hex} 00 _{hex} 11 _{hex}	Subindex existiert nicht
06 _{hex} 09 _{hex} 00 _{hex} 30 _{hex}	Wertebereich überschritten (bei Schreibzugriffen)
06 _{hex} 09 _{hex} 00 _{hex} 31 _{hex}	Wert zu groß (bei Schreibzugriffen)
08 _{hex} 00 _{hex} 00 _{hex} 20 _{hex}	Daten können nicht zur Applikation übertragen oder abgespeichert werden
08 _{hex} 00 _{hex} 00 _{hex} 22 _{hex}	Daten können aufgrund des derzeitigen Kommunikationszustandes nicht gemappt werden (z. B. Mapping ändern im Zustand OPERATIONAL)

Fehlerhafte Zugriffe auf alle anderen Objekte (4XXX_{hex} und 6XXX_{hex}) werden global mit folgenden Codes abgewiesen:

Abort Code	Bedeutung
06 _{hex} 01 _{hex} 00 _{hex} 00 _{hex}	Fehler im Datenformat
06 _{hex} 01 _{hex} 00 _{hex} 02 _{hex}	Element nicht änderbar
06 _{hex} 02 _{hex} 00 _{hex} 00 _{hex}	Element nicht vorhanden
06 _{hex} 09 _{hex} 00 _{hex} 31 _{hex}	Wert zu groß (bei Schreibzugriffen)
06 _{hex} 09 _{hex} 00 _{hex} 32 _{hex}	Wert zu klein (bei Schreibzugriffen)
08 _{hex} 00 _{hex} 00 _{hex} 00 _{hex}	Allgemeiner Fehler aufgetreten
08 _{hex} 00 _{hex} 00 _{hex} 21 _{hex}	Daten im Moment nicht verfügbar

5.3 Bedarfsdaten (SDO)

Beispiel Objekt 1008_{hex} Subindex 01_{hex} von Slave 4 soll gelesen werden. Das Objekt 1008_{hex} *Manufacturer Device Name* besitzt aber nur Subindex 00_{hex} .

Request

		CS	Multiplexor			D0	D1	D2	D3
COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
604_{hex}	08_{hex}	40_{hex}	08_{hex}	10_{hex}	01_{hex}	00_{hex}	00_{hex}	00_{hex}	00_{hex}

Response

		CS	Multiplexor			D0	D1	D2	D3
COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
584_{hex}	08_{hex}	80_{hex}	08_{hex}	10_{hex}	01_{hex}	11_{hex}	00_{hex}	09_{hex}	06_{hex}

Basis-Adresse 580_{hex}
+ Slave-Adresse 4_{hex}

Objekt 1008_{hex}

Subindex 01_{hex}

Code 06090011_{hex}

Der Command Specifier CS (Byte 0, 80_{hex}) im Response Telegramm gibt an, dass es sich um ein Abort Telegramm handelt. Die Multiplexor von Request und Response stimmen überein.

5.4 Synchronisation (SYNC)

Zur Synchronisation der Slaves wird das SYNC-Telegramm verwendet. Dieses Telegramm ist unbestätigt (Broadcast). Es enthält keine Daten. Die COB-ID steht im Objekt 1005_{hex} COB-ID SYNC. Defaultmäßig ist 80_{hex} festgelegt. Das Optionsmodul CANopen-Slave kann SYNC-Telegramme empfangen. Es ist kein SYNC-Master!

Der Empfang eines SYNC-Telegramms mit dem im Objekt 1005_{hex} dargestellten Identifier erzeugt auf dem Optionsmodul CANopen einen Interrupt, der an den $b\ maXX^{\circledR}$ Regler weitergeleitet wird. Damit kann dieses Signal zur Synchronisation des $b\ maXX^{\circledR}$ Reglers verwendet werden. Innerhalb eines SYNC-Intervalls (Kommunikationszyklus) müssen alle entsprechenden Telegramme an alle projektierten Slaves gesendet werden. Übertragungsrate, Leitungslänge, Anzahl der Knoten, Größe der Telegramme sowie Verarbeitungszeiten auf der CANopen-Optionskarte sind dabei zu berücksichtigen. Die Einstellung der Zykluszeit für das SYNC-Telegramm wird in Objekt 1006_{hex} vorgenommen. Siehe hierzu [►Verzeichnis der Objekte zur Kommunikationssteuerung◄](#) ab Seite 31. Weiterhin muss die Kommunikationszykluszeit im ProDrive/WinBASS II auf der Seite BACI Optionsmodul1 angepasst werden.

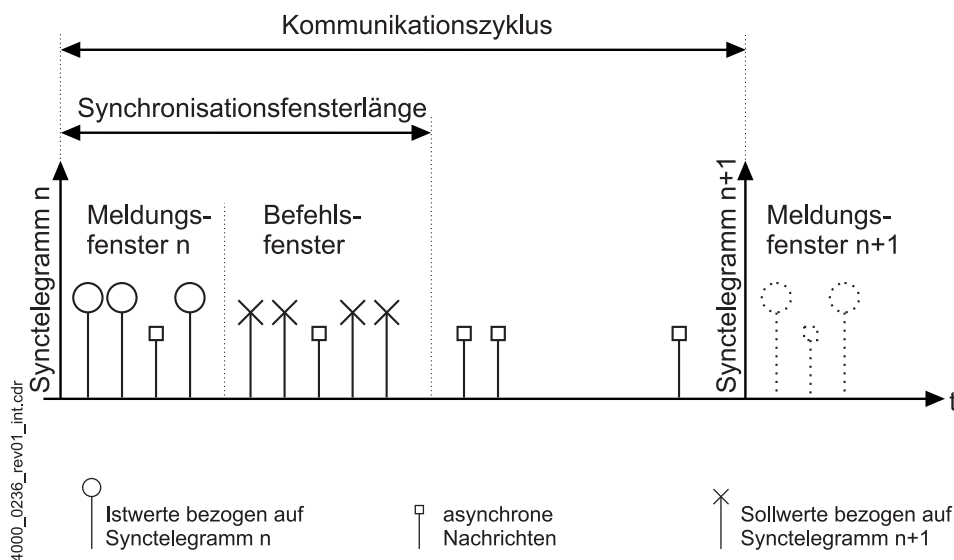


Abbildung 10: Kommunikationszyklus

Nach Empfang des SYNC-Telegramms senden die Slaves zunächst ihre Istwerte mittels synchroner PDOs im Meldungs-fenster, bevor im Befehls-fenster die Sollwerte vom Master an die Slaves ebenfalls mittels synchroner PDOs übertragen werden. Die Sollwerte werden von den Slaves mit dem nächsten SYNC-Telegramm übernommen (siehe auch [►Kommunikationsbeziehung über PDO◄](#) ab Seite 65). Asynchrone Nachrichten (SDOs, PDOs, NMT) können jederzeit auftreten.

Wird der Regler nicht über CANopen synchronisiert, darf im Regler keine Überwachung der Synchronisierung erfolgen. Das SYNC-Telegramm kann weiterhin als Triggerbedingung verwendet werden.

Einstellungen zur Synchronisierung des Reglers in ProDrive oder WINBASS II

Zusätzlich zu den BACI Einstellungen muss bei den Betriebsarten Lageregelung und Gleichlauf in ProDrive/WinBASS II auf der Seite Synchronisierung folgende Einstellung durchgeführt werden.

Quelle für Sync-Signal: Sync1 oder Sync2 von der BACI verwenden

Das Sync-Intervall muss auf die Zeit eingestellt werden, mit der das SYNC-Telegramm gesendet wird (kann auch über FBO 0x1006h eingestellt werden)

Fehlerreaktion bei Sync-Fehlern: nach Bedarf

Sync Offset: 100

Sync Toleranz: max. 40 μ s, sonst je nach Jitter des Masters.

Die Sync-Toleranz ist die Zeit, bei der der Regler keine Fehlermeldung ausgibt, wenn das Sync-Telegramm innerhalb dieser Zeit empfangen wird. Damit bei größerem Jitter die Synchronisation doch noch stattfinden kann, ist auf der Optionskarte ein Mechanismus (PLL) implementiert, der den Jitter ausgleichen soll. Die "PLL" wird nach dem ersten empfangenen Sync-Telegramm aktiviert. Das Sync-Telegramm muss ohne Unterbrechung gesendet werden. Fällt es während des Betriebes weg und wird wieder zugeschaltet, kann es im "worst case" eine halbe Minute dauern, bis die "PLL" wieder synchron zum Sync-Telegramm läuft, auch wenn ProDrive/WinBASS II nach kurzer Zeit anzeigt, dass der Regler schon synchron läuft.

Ab FPGA Version 0107 kann die "PLL" auf der Optionskarte deaktiviert werden. Dies ist aber nur zu empfehlen, wenn der Master ein Jitter kleiner als 1,6 μ s (2 ms Zykluszeit) für das Sync-Telegramm liefern kann.

Die Einstellungen müssen im Datensatz abgespeichert werden und der Regler muss neu gebootet werden.

HINWEIS



Wird die zyklische Kommunikation unterbrochen, z. B. beim Übergang von OPERATIONAL nach PRE-OPERATIONAL kann der Fehler/Warnung Alive Counter oder der Fehler zyklische Kommunikation auftreten. Der Fehler zyklische Kommunikation jedoch nur, wenn diese Überwachung aktiviert ist .

Werden die Betriebsarten Lageregelung und Gleichlauf nicht verwendet, sind keine extra Einstellungen für die BACI Zeiten und für die Synchronisierung notwendig.

5.5 Prozessdaten (PDO)

Prozess-Daten-Objekte sind unbestätigte Telegramme mit hochprioren COB-IDs. Sie sind auf den Austausch von Daten mit Echtzeitanforderungen optimiert. In den PDOs kann der gesamte CAN-Datenrahmen (acht Byte) für die Nutzdatenübertragung verwendet werden. Deshalb muss das Format des Datenaustauschs über PDOs vor Beginn der Kommunikation zwischen Sender und Empfänger definiert sein (Mapping). Das Senden

und Empfangen von PDOs kann auf unterschiedliche Weise ausgelöst werden (siehe auch [►Kommunikationsbeziehung über PDO◄](#) ab Seite 65).

HINWEIS



Alle in den PDOs konfigurierten Objekte werden zwischen dem Optionsmodul CANopen-Slave und dem b maXX[®] Regler als zyklische Daten übertragen (siehe auch [►Kommunikationsablauf◄](#) ab Seite 19). Da die zyklische Datenübertragung nur im Zustand OPERATIONAL stattfindet, darf auch nur in diesem Zustand die Kommunikationsüberwachung in ProDrive/WinBASS II BACI aktiviert werden (Timeout für zyklische Kommunikation **P0836** (BACI)).

5.5.1 PDO-Mapping

Mapping ist ein Verfahren zur Zuordnung von Variablen / Objekten an PDOs. Mit den PDOs werden diese Variablen / Objekte über den CAN-Bus transportiert. Durch das Mapping wird der zyklische Datenaustausch projiziert. Für diese Parametrierung werden SDOs genutzt. Das Mapping wird über im Objektverzeichnis adressierbare Objekte eingestellt. Für jede PDO existieren vier solcher Objekte (siehe auch [►Verzeichnis der Objekte zur Kommunikationssteuerung◄](#) ab Seite 31). Eines der Objekte bestimmt den Inhalt der PDO, das zweite die Kommunikationsbeziehung bzw. Triggerung.

Prozess-Daten-Objekt	Objekt für Inhalt	Objekt für die Kommunikationsbeziehung
TX-PDO1	1A00 _{hex}	1800 _{hex}
TX-PDO2	1A01 _{hex}	1801 _{hex}
TX-PDO3	1A02 _{hex}	1802 _{hex}
TX-PDO4	1A03 _{hex}	1803 _{hex}
RX-PDO1	1600 _{hex}	1400 _{hex}
RX-PDO2	1601 _{hex}	1401 _{hex}
RX-PDO3	1602 _{hex}	1402 _{hex}
RX-PDO4	1603 _{hex}	1403 _{hex}

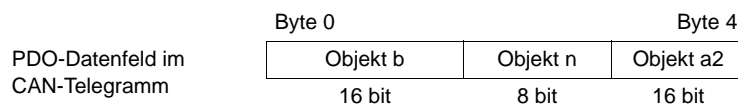
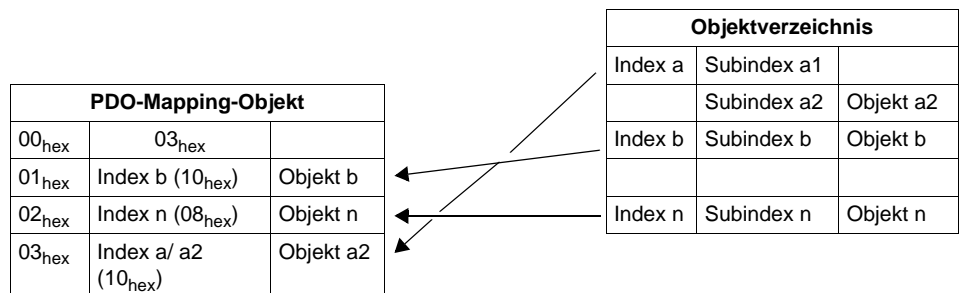
HINWEIS



Das Mapping kann nicht im Zustand OPERATIONAL geändert werden. Ein neues Mapping wird erst beim Übergang nach OPERATIONAL aktiviert.

Zur Nutzdatenübertragung stellt ein CAN-Datentelegramm maximal acht Byte zur Verfügung. Durch das Mapping wird der logische Inhalt dieser maximal acht Byte festgelegt. Für diese Festlegung werden bestimmte Angaben über das zu mappende Objekt benötigt: Objektindex, Subindex und Länge des Datums. Aus dem Objektverzeichnis werden die entsprechenden Objekte in das Mapping-Objekt eingetragen. Die Reihenfolge dieses Eintrages, festgelegt durch den Subindex des Mapping-Objektes, bestimmt die Reihenfolge der Daten im CAN-Telegramm. In den Mapping-Objekten (1600_{hex}, 1601_{hex},

1602_{hex}, 1603_{hex}, 1A00_{hex}, 1A01_{hex}, 1A02_{hex}, 1A03_{hex}) werden die zu mappenden Objekte an die entsprechenden Subindizes (mit 01_{hex} beginnend) geschrieben, z. B. wird auf Objekt 1600_{hex} Subindex 01_{hex} der Wert 60400010_{hex} eingetragen. Dies bedeutet, die ersten beiden Bytes der in RX-PDO1 empfangenen Daten werden auf das Steuerwort (Objekt 6040_{hex} Subindex 00_{hex}) geschrieben. Das Objekt 6040_{hex} ist auf den b maXX[®] 4400-Parameter **P0300** Steuerwort umgesetzt (siehe auch [Anhang C - Umsetzungstabellen](#) ab Seite 101). Damit wird das erste Wort des in RX-PDO1 empfangenen Telegramms auf das Steuerwort des b maXX[®] 4400 geschrieben. In Subindex 00_{hex} muss die Anzahl der zu mappenden Objekte (Anzahl der mit gültigen Objekten belegten Subindizes) eingetragen werden. Ein ausführliches Beispiel für das Mapping ist in [Beispiel für das PDO-Mapping](#) ab Seite 67 erläutert.



Das Default-Mapping ist in [Verzeichnis der Objekte zur Kommunikationssteuerung](#) ab Seite 31 beschrieben.

Um ein bestehendes Mapping zu deaktivieren, können die Werte in den Subindizes überschrieben werden, oder es ist der Wert „0“ auf den Subindex 00_{hex} des entsprechenden Mapping-Objektes (1600_{hex}, 1601_{hex}, 1602_{hex}, 1603_{hex}, 1A00_{hex}, 1A01_{hex}, 1A02_{hex}, 1A03_{hex}) zu schreiben. Damit wird das gesamte Mapping der jeweiligen PDO deaktiviert, der Eintrag aber bleibt erhalten.

Zudem kann das Deaktivieren der Mapping Objekte über die zugehörige Kommunikationsobjekte 1400_{hex} bis 1403_{hex}, 1800_{hex} bis 1803_{hex} im Subindex 1 mit Bit 31 auf 1 setzen erfolgen.

Beachte: Dabei muss die COBID mitgeschrieben werden.

HINWEIS



Bei der Einstellung des Mappings in den (1600_{hex}, 1601_{hex}, 1602_{hex}, 1603_{hex}, 1A00_{hex}, 1A01_{hex}, 1A02_{hex}, 1A03_{hex}) ist jeweils der Subindex 00_{hex} mit der richtige Anzahl der gemappten Objekte zuletzt zu beschreiben.

Sollwerte: Die zulässigen zyklische Sollwerte sind in einer Tabelle mit der Spalte „PDO-Mapping“ als „RX“ gekennzeichnet. Die Tabelle ist im Anhang B.2 (für die 6000'er Objektnummern) zu finden. Bei den herstellerspezifischen Parameter (4000'er Objekte) muss im Parameterhandbuch b maXX[®] 4400 Grundgerät (5.02017), Kapitel 6.1.4 Attribute, für das b maXX[®] 4400 nachgeschaut werden.

Istwerte: Die zulässigen zyklische Istwerte sind in einer Tabelle mit der Spalte „PDO-Mapping“ als „TX“ gekennzeichnet. Die Tabelle ist im Anhang B.2 (für die 6000'er Objektnummern) zu finden. Bei den herstellerspezifischen Parametern (4000'er Objekte) muss im Parameterhandbuch b maXX[®] 4400 Grundgerät (5.02017), Kapitel 6.1.4 Attribute, für das b maXX[®] 4400 nachgeschlagen werden. Eine detaillierte Beschreibung der b maXX[®]-Parameter finden Sie im Parameterhandbuch b maXX[®].

Fehlerhafte Mapping-Konfigurationen (unzulässige Objekte in 1600_{hex}, 1601_{hex}, 1602_{hex}, 1603_{hex}, 1A00_{hex}, 1A01_{hex}, 1A02_{hex}, 1A03_{hex}) werden durch Abort Codes über SDO gemeldet.

Die zyklischen Soll-/Istwerte werden lückenlos in der BACI-Konfiguration initialisiert, d. h. der erste Sollwert von PDO1 steht an erster Stelle in der BACI, der zweite Sollwert von PDO1 an zweiter Stelle usw. Anschließend folgen die Sollwerte der PDO2. Analog gilt für die Istwert-Initialisierung der erste Istwert von PDO1 steht an erster Stelle in der BACI, der zweite Istwert von PDO1 an zweiter Stelle usw.

Wird nicht der volle Umfang von PDO1 (max. 4 Sollwerte) genutzt, rücken die Werte von PDO2 nach. Im zyklischen Datenbereiche der BACI treten keine Lücken auf.

HINWEIS



Wird das Regler Status Word (FBO 6041_{hex}) für die zyklische Kommunikation erwünscht, so muss das Status Word in die erste PDO an erster Stelle eingetragen werden!

Ist das Status Word nicht gewünscht, wird es trotzdem in der BACI eingetragen, aber es wird in diesem Fall bei der Feldbusübertragung nicht berücksichtigt. Das Status Word ist an dieser Stelle für interne Abfragen notwendig. Es können dann nur 7 weitere Istwerte benutzt werden.

Wenn ein falscher Parameter (z. B. ein Istwert Parameter) bei den Sollwerten gemappt wird, sendet der Slave ein EMYC-Telegramm. Gesendet wird in der Regel nur die erste Fehlernummer wenn mehrere Fehlermeldungen anstehen. Also nicht unbedingt die Fehlernummer, die gerade den Fehler ausgelöst hat. (Daher bei mehreren Fehlermeldungen in ProDrive/WINBASS II alle anstehenden Fehler auslesen).

In diesem Fall meldet der Regler die Fehlernummer 54.4100 "Optionsmodul G/H Falsche Parameter-Nr. bei SollwertParameter Nr. n".

Die Optionskarte signalisiert zwar über die LED, dass sie in OP ist, aber es werden keine zyklischen Sollwerte zum Regler weitergereicht. Der Grund dafür ist, dass der Regler die Konfiguration der BACI abgebrochen hat.

HINWEIS



In diesem Zustand ist ein Neustart der Optionskarte notwendig.

Dummy-Mapping Das Optionsmodul CANopen-Slave stellt 2 Dummy-Objekte bereit: ein 1 Byte Dummy-Objekt und ein 2 Byte Dummy-Objekt, welches ebenfalls in eine PDO gemappt werden kann. Diese Objekte haben die Indices 0005_{hex} (1 Byte Dummy) und 0006_{hex} (2 Byte Dummy). Das Dummy-Objekt dient als Platzhalter, um nur bestimmte Objekte innerhalb eines CAN-Telegramms zu verwenden (siehe auch [►Beispiel für das PDO-Mapping◄](#) ab Seite 67).

HINWEIS



Das aktuell eingestellte Mapping geht nach dem Ausschalten oder einem Reset Befehl der Zustandsmaschine verloren. Danach stellt sich das Mapping ein, das als letztes über das Objekt 1010_{hex} abgespeichert wurde. Wurde keines abgespeichert, wird das Default Mapping verwendet.

Beschreiben von gleichen Feldbusobjekten (FBO) über Servicedaten SD und Prozessdaten PD.

In der Regel überschreiben PD Schreibzugriffe zyklisch SD Schreibzugriffe auf das gleiche FBO. Dies ist auch dann der Fall, wenn die PD mit dem gleichen FBO gar nicht gesendet wird, aber ein anderes FBO in einer anderen PD. Der Grund dafür ist, dass alle eingetragenen BACI-Parameter bei einer Änderung von einem darin enthaltenen Parameter übertragen werden.

In einzelnen Fällen kann es vorkommen, dass ein Schreibzugriff über SD erfolgreich gewesen ist, aber zuverlässig ist das nicht.

HINWEIS



In diesem Zusammenhang sollte man vermeiden auf das gleiche Feldbusobjekt über SD und PD zuzugreifen.

5.5.2 Kommunikationsbeziehung über PDO

In jedem Mapping-Objekt existiert ein Objekt für die Einstellung der Kommunikation. Der Objektindex hat einen Offset von -200_{hex} zum entsprechenden Mapping-Objekt.

Prozess Daten Objekt Mapping	Objekt für Inhalt	Objekt für die Kommunikationsbeziehung
TX-PDO1	$1A00_{hex}$	1800_{hex}
TX-PDO2	$1A01_{hex}$	1801_{hex}
TX-PDO3	$1A02_{hex}$	1802_{hex}
TX-PDO4	$1A03_{hex}$	1803_{hex}
RX-PDO1	1600_{hex}	1400_{hex}
RX-PDO2	1601_{hex}	1401_{hex}
RX-PDO3	1602_{hex}	1402_{hex}
RX-PDO4	1603_{hex}	1403_{hex}

Der Aufbau dieser Objekte ist in [Verzeichnis der Objekte zur Kommunikationssteuerung](#) ab Seite 31 beschrieben.

Das Kriterium für die Übernahme einer auf dem CAN-Bus übertragenen Nachricht in das Optionsmodul CANopen-Slave ist die passende COB-ID. Die COB-ID wird in den Steuerobjekten $1400_{hex} - 1403_{hex}$, $1800_{hex} - 1803_{hex}$ unter Subindex 01_{hex} eingestellt. Stimmt der hier parametrisierte Identifier mit dem über den CAN-Bus gesendeten Nachrichten-Identifier überein, wird das Telegramm in den eigenen Telegrammpuffer übernommen.

Die PDOs können an dieser Stelle auch deaktiviert werden, dabei wird das Bit 31 mit 1 beschrieben.

z.B. 1400_{hex} Subindex 1

		CS	Multiplexor				D0	D1	D2	D3
COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	
601_{hex}	08_{hex}	23_{hex}	00_{hex}	14_{hex}	01_{hex}	01_{hex}	00_{hex}	00_{hex}	01_{hex}	

Objekt 14 00_{hex}
Wichtig
Bit 31

Weiterhin werden bei CANopen für das Senden und Empfangen Triggerbedingungen definiert, die PDOs in synchron und asynchron gliedern. Die Triggerbedingungen werden in den Objekten $1400_{hex} - 1403_{hex}$, $1800_{hex} - 1803_{hex}$ jeweils im Subindex 02_{hex} eingestellt.

HINWEIS



Nicht benötigte PDOs sollten deaktiviert werden, um eine Beeinflussung auszuschließen. Das Deaktivieren erfolgt durch das Beschreiben der Subindizes 0 der Objekte 1600_{hex} bis 1603_{hex} und $1A00_{hex}$ bis $1A03_{hex}$ mit 0 oder mit Bit 31 in SIX 2 der Objekte 1401_{hex} bis 1403_{hex} und 1801_{hex} bis 1803_{hex} .

Synchrone PDO Das Senden und Empfangen ist an das SYNC-Telegramm gebunden (siehe auch [>Synchronisation \(SYNC\)<](#) ab Seite 59).

Asynchrone PDO Das Senden und Empfangen ist an bestimmte Ereignisse gebunden.

Wert in Subindex 02 _{hex}	Typ	Wirkung	
		TX-PDO 1800 _{hex} , 1801 _{hex} , 1802 _{hex} , 1803 _{hex}	RX-PDO 1400 _{hex} , 1401 _{hex} , 1402 _{hex} , 1403 _{hex}
00 _{hex} (0)	asynchron	Das Senden erfolgt nach jedem empfangenen SYNC-Telegramm und es ist ein Ereignis eingetreten.	Vor dem letzten SYNC-Telegramm empfangene PDO mit passender COB-ID wird übernommen
01 _{hex} (1)	synchron	Das Senden erfolgt nach jedem empfangenen SYNC-Telegramm	Vor dem letzten SYNC-Telegramm empfangene PDO mit passender COB-ID wird übernommen
02 _{hex} - F0 _{hex} (2 - 240)	synchron	Das Senden erfolgt nach Empfang der eingestellten Anzahl von SYNC-Telegrammen	Vor dem letzten SYNC-Telegramm empfangene PDO mit passender COB-ID wird übernommen
FC _{hex} (252)	RTR synchron	Senden erfolgt nach Empfang des RTR -Telegramms mit passender COB-ID. Aktualisierung der PDO erfolgt nach Empfang des SYNC-Telegramms.	keine
FD _{hex} (253)	RTR asynchron	Senden erfolgt nach Empfang des RTR -Telegramms mit passender COB-ID.	keine
FE _{hex} (254)	asynchron	Das Senden erfolgt zeitgesteuert	keine
FF _{hex} (255)	asynchron	Das Senden erfolgt ereignisgesteuert	Jede PDO mit passender COB-ID wird übernommen

Zeitgesteuertes Senden bedeutet, dass die Sendebedingung an einen Timer gebunden ist. Dieser Timer wird für die TX-PDO1 mittels Subindex 05_{hex} im Objekt 1800_{hex} (16 Bit) eingestellt. Analog lässt sich der Timer für TX-PDO2, TX-PDO3 und TX-PDO4 im Subindex 05_{hex} von Objekt 1801_{hex}, 1802_{hex}, 1803_{hex} setzen. Die Auflösung beträgt jeweils 1 Millisekunde. Der Timer oder die Timer werden beim Zustandswechsel nach OPERATIONAL gestartet. Das Senden der entsprechenden TX-PDO erfolgt dann zyklisch mit der in den Timern eingestellten Zeit. Die Timer werden gelöscht, indem auf die Subindizes 05_{hex} der Objekte 1800_{hex} - 1803_{hex} der Wert „0“ geschrieben wird.

Zeitgesteuertes Empfangen existiert nicht! Die Wirkung entspricht dem ereignisgesteuerten Empfang.

Ereignisgesteuertes Senden bedeutet, dass die Sendebedingung an die Änderung eines Wertes der gemappten Objekte gebunden ist. Sind beispielsweise 3 Objekte gemappt (Statuswort, Drehzahl-Istwert, Ist-Betriebsart), wird die PDO gesendet sobald sich mindestens einer der drei Werte ändert. Bleiben die Werte konstant, wird keine PDO gesendet. Dadurch lässt sich die Buslast verringern (Telegramme werden nur übertragen, wenn sie neue Informationen enthalten).

Ereignisgesteuertes Empfangen heißt, dass alle PDOs mit passender COB-ID übernommen werden.

Bei dem Transmission Types RTR, synchron / RTR asynchron (Typen 252 und 253) wird die PDO mit passender COB-ID erst nach Empfang des RTR Telegramms gesendet. Beim Typ 252 wird die TX-PDO nach jedem empfangenen SYNC-Telegramm aktualisiert, aber noch nicht gesendet. Beim Typ 253 erfolgt die Aktualisierung der PDO nach Empfang des RTR-Telegramms (abhängig von der BACI-Zykluszeit). RTR-Telegramme sind nur für TX-PDOs möglich.

5.5.3 Beispiel für das PDO-Mapping

Das Optionsmodul CANopen-Slave mit dem Knoten 2 empfängt vom Master einen Drehzahl Sollwert in RX-PDO1. Dieser Drehzahl Sollwert soll auf den Hochlaufgeber-Eingang geschrieben werden. Das Optionsmodul CANopen-Slave mit dem Knoten 7 soll immer den identischen Drehzahl-Istwert aufweisen wie Knoten 2. Dieser Wert wird auf den Hochlaufgeber-Eingang von Knoten 7 geschrieben. Die Realisierung dieser Konfiguration sieht folgendermaßen aus:

Der Master schickt den Drehzahl Sollwert an Knoten 2. Sobald Knoten 2 eine Änderung dieses Wertes erkennt, sendet er seinen Istwert an Knoten 7.

Weiterhin erhält Knoten 2 in seiner RX-PDO1 das Steuerwort vom Master. Knoten 7 empfängt in RX-PDO 2 ebenfalls ein Steuerwort vom Master. Die Konfiguration ist in [▶ Abbildung 11 ◀](#) auf Seite 67 dargestellt. Das Objekt 6086_{hex} wird im Zusammenhang mit dem Dummy-Mapping verwendet.

Das b maXX[®] 4400 mit der Adresse 2 sendet seinen Drehzahl-Istwert und das Statuswort alle 10 ms. Der Knoten 7 sendet sein Statuswort nur nach 3-maligem Empfang eines Sync-Telegramms (vom Master).

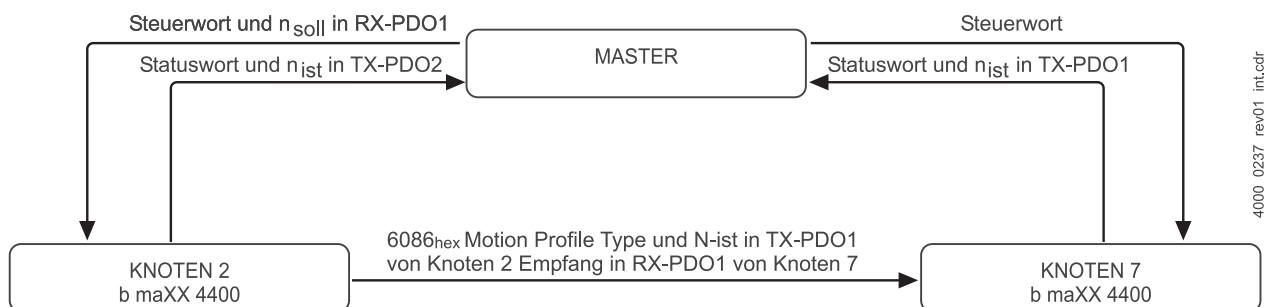


Abbildung 11: Beispiel-Mapping mit zwei b maXX[™] 4400

1. Schritt: Bestimmen der notwendigen Objekte

Ermitteln Sie aus der Objektliste (siehe [►Anhang C - Umsetzungstabellen◄](#) ab Seite 101 und [►Verzeichnis der Objekte zur Kommunikationssteuerung◄](#) ab Seite 31) die entsprechende Objekte des Objektverzeichnisses.

Für die Geräte sind folgende Parameter, die mit den angegebenen Objekten korrespondieren, relevant:

P0301 Statuswort	↔	6041 _{hex} Statuswort
P0300 Steuerwort	↔	6040 _{hex} Steuerwort
P1171 Sollwertauswahl HLG Eingang	↔	6042 _{hex} Drehzahlsollwert am HLG
P0353 Drehzahl-Istwert	↔	6044 _{hex} Control Effort
P1190 Positionierung Modus	↔	6086 _{hex} Motion Profile Type

Zum Einstellen des Mappings sind folgende Objekte nötig:

Knoten 2	1A00 _{hex} (1. Transmit PDO Mapping), 1800 _{hex} (1. Transmit PDO Parameter) 1A01 _{hex} (2. Transmit PDO Mapping), 1801 _{hex} (2. Transmit PDO Parameter)
Knoten 7	1600 _{hex} (1. Receive PDO Mapping), 1400 _{hex} (1. Receive PDO Parameter) 1601 _{hex} (2. Receive PDO Mapping), 1401 _{hex} (2. Receive PDO Parameter)

2. Schritt: Mapping konfigurieren

Zum Einstellen des Mappings werden die SDOs des Expedited Transfers (siehe auch [►Bedarfsdaten \(SDO\)◄](#) ab Seite 49) verwendet. Diese können über einen Master, einen Buskonfigurator o. ä. initiiert werden.

Mapping für Slave 2

Schreiben des ersten zu mappenden Objektes mit Index (6086_{hex}), Subindex (00_{hex}) und Länge (10_{hex}) auf 1A00_{hex} Subindex 01_{hex} (TX-PDO 1). Das Objekt soll vom Slave 7 nicht ausgewertet.

Request

		CS	Multiplexor			D0	D1	D2	D3
COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
602 _{hex}	08 _{hex}	23 _{hex}	00 _{hex}	1A _{hex}	01 _{hex}	10 _{hex}	00 _{hex}	86 _{hex}	60 _{hex}

Response

		CS	Multiplexor			D0	D1	D2	D3
COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
582 _{hex}	08 _{hex}	60 _{hex}	00 _{hex}	1A _{hex}	01 _{hex}	00 _{hex}	00 _{hex}	00 _{hex}	00 _{hex}

Schreiben des zweiten zu mappenden Objektes mit Index (6044_{hex}), Subindex (00_{hex}) und Länge (10_{hex}) auf 1A00_{hex} Subindex 02_{hex} (TX-PDO 1).

Request			CS	Multiplexor			D0	D1	D2	D3
	COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
	602 _{hex}	08 _{hex}	23 _{hex}	00 _{hex}	1A _{hex}	02 _{hex}	10 _{hex}	00 _{hex}	44 _{hex}	60 _{hex}

Response			CS	Multiplexor			D0	D1	D2	D3
	COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
	582 _{hex}	08 _{hex}	60 _{hex}	00 _{hex}	1A _{hex}	02 _{hex}	00 _{hex}	00 _{hex}	00 _{hex}	00 _{hex}

Schreiben der Anzahl der gemappten Objekte (02_{hex}) auf 1A00_{hex} Subindex 00_{hex} (TXP-DO 1).

Request			CS	Multiplexor			D0	D1	D2	D3
	COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
	602 _{hex}	08 _{hex}	2F _{hex}	00 _{hex}	1A _{hex}	00 _{hex}	02 _{hex}	00 _{hex}	00 _{hex}	00 _{hex}

Response			CS	Multiplexor			D0	D1	D2	D3
	COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
	582 _{hex}	08 _{hex}	60 _{hex}	00 _{hex}	1A _{hex}	00 _{hex}	00 _{hex}	00 _{hex}	00 _{hex}	00 _{hex}

Der Inhalt von Objekt 1A00_{hex} sieht wie folgt aus:

1A00_{hex}	00_{hex}	02 _{hex}
	01_{hex}	60860010 _{hex}
	02_{hex}	60440010 _{hex}

Das nachfolgende Mapping des Objekts 6041_{hex} wird vollständigshalber gezeigt, muss aber nicht durchgeführt werden, da es im Default-Mapping eingestellt ist.

5.5 Prozessdaten (PDO)

Schreiben des ersten zu mappenden Objektes mit Index (6041_{hex}), Subindex (00_{hex}) und Länge (10_{hex}) auf 1A01_{hex} Subindex 01_{hex} (TX-PDO 2).

Request

		CS	Multiplexor			D0	D1	D2	D3
COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
602 _{hex}	08 _{hex}	23 _{hex}	01 _{hex}	1A _{hex}	01 _{hex}	10 _{hex}	00 _{hex}	41 _{hex}	60 _{hex}

Response

		CS	Multiplexor			D0	D1	D2	D3
COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
582 _{hex}	08 _{hex}	60 _{hex}	01 _{hex}	1A _{hex}	01 _{hex}	00 _{hex}	00 _{hex}	00 _{hex}	00 _{hex}

Schreiben des zweiten zu mappenden Objektes mit Index (6044_{hex}), Subindex (00_{hex}) und Länge (10_{hex}) auf 1A01_{hex} Subindex 02_{hex}.

Request

		CS	Multiplexor			D0	D1	D2	D3
COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
602 _{hex}	08 _{hex}	23 _{hex}	01 _{hex}	1A _{hex}	02 _{hex}	10 _{hex}	00 _{hex}	44 _{hex}	60 _{hex}

Response

		CS	Multiplexor			D0	D1	D2	D3
COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
582 _{hex}	08 _{hex}	60 _{hex}	10 _{hex}	1A _{hex}	02 _{hex}	00 _{hex}	00 _{hex}	00 _{hex}	00 _{hex}

Schreiben der Anzahl der gemappten Objekte (02_{hex}) auf 1A01_{hex} Subindex 00_{hex} (TX-PDO 2).

Request

		CS	Multiplexor			D0	D1	D2	D3
COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
602 _{hex}	08 _{hex}	2F _{hex}	01 _{hex}	1A _{hex}	00 _{hex}	02 _{hex}	00 _{hex}	00 _{hex}	00 _{hex}

Response

		CS	Multiplexor			D0	D1	D2	D3
COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
582 _{hex}	08 _{hex}	60 _{hex}	01 _{hex}	1A _{hex}	00 _{hex}	00 _{hex}	00 _{hex}	00 _{hex}	00 _{hex}

Der Inhalt von Objekt 1A01_{hex} sieht wie folgt aus:

1A01 _{hex}	00 _{hex}	02 _{hex}
	01 _{hex}	60410010 _{hex}
	02 _{hex}	60440010 _{hex}

Schreiben des ersten zu mappenden Objektes mit Index (6040_{hex}), Subindex (00_{hex}) und Länge (10_{hex}) auf 1600_{hex} Subindex 01_{hex} (RX-PDO 1).

Request

		CS	Multiplexor			D0	D1	D2	D3
COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
602 _{hex}	08 _{hex}	23 _{hex}	00 _{hex}	16 _{hex}	01 _{hex}	10 _{hex}	00 _{hex}	40 _{hex}	60 _{hex}

Response

		CS	Multiplexor			D0	D1	D2	D3
COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
582 _{hex}	08 _{hex}	60 _{hex}	00 _{hex}	16 _{hex}	01 _{hex}	00 _{hex}	00 _{hex}	00 _{hex}	00 _{hex}

Schreiben des zweiten zu mappenden Objektes mit Index (6042_{hex}), Subindex (00_{hex}) und Länge (10_{hex}) auf 1600_{hex} Subindex 02_{hex} (RX-PDO 1).

Request

		CS	Multiplexor			D0	D1	D2	D3
COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
602 _{hex}	08 _{hex}	23 _{hex}	00 _{hex}	16 _{hex}	02 _{hex}	10 _{hex}	00 _{hex}	42 _{hex}	60 _{hex}

Response

		CS	Multiplexor			D0	D1	D2	D3
COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
582 _{hex}	08 _{hex}	60 _{hex}	00 _{hex}	16 _{hex}	02 _{hex}	00 _{hex}	00 _{hex}	00 _{hex}	00 _{hex}

Schreiben der Anzahl der gemappten Objekte (02_{hex}) auf 1600_{hex} Subindex 00_{hex} (RX-PDO 1).

Request			CS	Multiplexor			D0	D1	D2	D3
	COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
	602 _{hex}	08 _{hex}	2F _{hex}	00 _{hex}	16 _{hex}	00 _{hex}	02 _{hex}	00 _{hex}	00 _{hex}	00 _{hex}

Response			CS	Multiplexor			D0	D1	D2	D3
	COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
	582 _{hex}	08 _{hex}	60 _{hex}	00 _{hex}	16 _{hex}	00 _{hex}	00 _{hex}	00 _{hex}	00 _{hex}	00 _{hex}

Der Inhalt von Objekt 1600_{hex} sieht wie folgt aus:

1600_{hex}	00_{hex}	02 _{hex}
	01_{hex}	60400010 _{hex}
	02_{hex}	60420010 _{hex}

Mapping für Slave 7 Slave 7 soll in RX-PDO 1 nur den Drehzahlsollwert von Slave 2 (hier Drehzahl-Istwert) auswerten. Der Drehzahlsollwert ist an zweite Stelle der TX-PDO 1 von Slave 2 gemappt. Daher muss für die erste Stelle das Dummy-Objekt verwendet werden.

Schreiben des ersten zu mappenden Objektes mit Index (6041_{hex}), Subindex (00_{hex}) und Länge (10_{hex}) auf 1A00_{hex} Subindex 01_{hex} (TX-PDO 1).

Dies ist aber auch als Default-Mapping eingetragen und muss nicht unbedingt noch einmal eingetragen werden.

Request			CS	Multiplexor			D0	D1	D2	D3
	COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
	607 _{hex}	08 _{hex}	23 _{hex}	00 _{hex}	1A _{hex}	01 _{hex}	10 _{hex}	00 _{hex}	41 _{hex}	60 _{hex}

Response			CS	Multiplexor			D0	D1	D2	D3
	COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
	587 _{hex}	08 _{hex}	60 _{hex}	00 _{hex}	1A _{hex}	01 _{hex}	00 _{hex}	00 _{hex}	00 _{hex}	00 _{hex}

Schreiben des zweiten zu mappenden Objektes mit Index (6044_{hex}), Subindex (00_{hex}) und Länge (10_{hex}) auf 1A00_{hex} Subindex 02_{hex} (TX-PDO 1).

Request

		CS	Multiplexor			D0	D1	D2	D3
COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
607 _{hex}	08 _{hex}	23 _{hex}	00 _{hex}	1A _{hex}	02 _{hex}	10 _{hex}	00 _{hex}	44 _{hex}	60 _{hex}

Response

		CS	Multiplexor			D0	D1	D2	D3
COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
587 _{hex}	08 _{hex}	60 _{hex}	00 _{hex}	1A _{hex}	02 _{hex}	00 _{hex}	00 _{hex}	00 _{hex}	00 _{hex}

Schreiben der Anzahl der gemappten Objekte (02_{hex}) auf 1A00_{hex} Subindex 00_{hex} (TX-PDO 1).

Request

		CS	Multiplexor			D0	D1	D2	D3
COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
607 _{hex}	08 _{hex}	2F _{hex}	00 _{hex}	1A _{hex}	00 _{hex}	02 _{hex}	00 _{hex}	00 _{hex}	00 _{hex}

Response

		CS	Multiplexor			D0	D1	D2	D3
COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
587 _{hex}	08 _{hex}	60 _{hex}	00 _{hex}	1A _{hex}	00 _{hex}	00 _{hex}	00 _{hex}	00 _{hex}	00 _{hex}

Der Inhalt von Objekt 1A00_{hex} sieht wie folgt aus:

1A00 _{hex}	00 _{hex}	02 _{hex}
	01 _{hex}	60410010 _{hex}
	02 _{hex}	60440010 _{hex}

5.5 Prozessdaten (PDO)

Schreiben des ersten zu mappenden Objektes (Dummy Objekt 16Bit) mit Index (0006_{hex}), Subindex (00_{hex}) und Länge (10_{hex}) auf 1600_{hex} Subindex 01_{hex} (RX-PDO 1).

Request

		CS	Multiplexor			D0	D1	D2	D3
COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
607 _{hex}	08 _{hex}	23 _{hex}	00 _{hex}	16 _{hex}	01 _{hex}	10 _{hex}	00 _{hex}	06 _{hex}	00 _{hex}

Response

		CS	Multiplexor			D0	D1	D2	D3
COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
587 _{hex}	08 _{hex}	60 _{hex}	00 _{hex}	16 _{hex}	01 _{hex}	00 _{hex}	00 _{hex}	00 _{hex}	00 _{hex}

Schreiben des zweiten zu mappenden Objektes mit Index (6042_{hex}), Subindex (00_{hex}) und Länge (10_{hex}) auf 1600_{hex} Subindex 02_{hex}.

Request

		CS	Multiplexor			D0	D1	D2	D3
COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
607 _{hex}	08 _{hex}	23 _{hex}	00 _{hex}	16 _{hex}	02 _{hex}	10 _{hex}	00 _{hex}	42 _{hex}	60 _{hex}

Response

		CS	Multiplexor			D0	D1	D2	D3
COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
587 _{hex}	08 _{hex}	60 _{hex}	00 _{hex}	16 _{hex}	02 _{hex}	00 _{hex}	00 _{hex}	00 _{hex}	00 _{hex}

Schreiben der Anzahl der gemappten Objekte (02_{hex}) auf 1600_{hex} Subindex 00_{hex}.

Request

		CS	Multiplexor			D0	D1	D2	D3
COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
607 _{hex}	08 _{hex}	2F _{hex}	00 _{hex}	16 _{hex}	00 _{hex}	02 _{hex}	00 _{hex}	00 _{hex}	00 _{hex}

Response

		CS	Multiplexor			D0	D1	D2	D3
COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
587 _{hex}	08 _{hex}	60 _{hex}	00 _{hex}	16 _{hex}	00 _{hex}	00 _{hex}	00 _{hex}	00 _{hex}	00 _{hex}

Der Inhalt von Objekt 1600_{hex} sieht wie folgt aus:

1600 _{hex}	00 _{hex}	02 _{hex}
	01 _{hex}	00060010 _{hex}
	02 _{hex}	60420010 _{hex}

Schreiben des ersten zu mappenden Objektes mit Index (6040_{hex}), Subindex (00_{hex}) und Länge (10_{hex}) auf 1601_{hex} Subindex 01_{hex} (RX-PDO 2).

Request

		CS	Multiplexor			D0	D1	D2	D3
COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
607 _{hex}	08 _{hex}	23 _{hex}	01 _{hex}	16 _{hex}	01 _{hex}	10 _{hex}	00 _{hex}	40 _{hex}	60 _{hex}

Response

		CS	Multiplexor			D0	D1	D2	D3
COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
587 _{hex}	08 _{hex}	60 _{hex}	01 _{hex}	16 _{hex}	01 _{hex}	00 _{hex}	00 _{hex}	00 _{hex}	00 _{hex}

Schreiben der Anzahl der gemappten Objekte (01_{hex}) auf 1601_{hex} Subindex 00_{hex}.

Request

		CS	Multiplexor			D0	D1	D2	D3
COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
607 _{hex}	08 _{hex}	2F _{hex}	01 _{hex}	16 _{hex}	00 _{hex}	01 _{hex}	00 _{hex}	00 _{hex}	00 _{hex}

Response

		CS	Multiplexor			D0	D1	D2	D3
COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
587 _{hex}	08 _{hex}	60 _{hex}	01 _{hex}	16 _{hex}	00 _{hex}	00 _{hex}	00 _{hex}	00 _{hex}	00 _{hex}

Der Inhalt von Objekt 1601_{hex} sieht wie folgt aus:

1601 _{hex}	00 _{hex}	01 _{hex}
	01 _{hex}	60400010 _{hex}

Der Datenaustausch zwischen den b maXX[®] 4400 über die PDOs ist in [▶Abbildung 12](#) dargestellt. Beispiel für eine Querkommunikation. Der Drehzahlwert von Slave 2 wird Drehzahlsollwert von Slave 7.

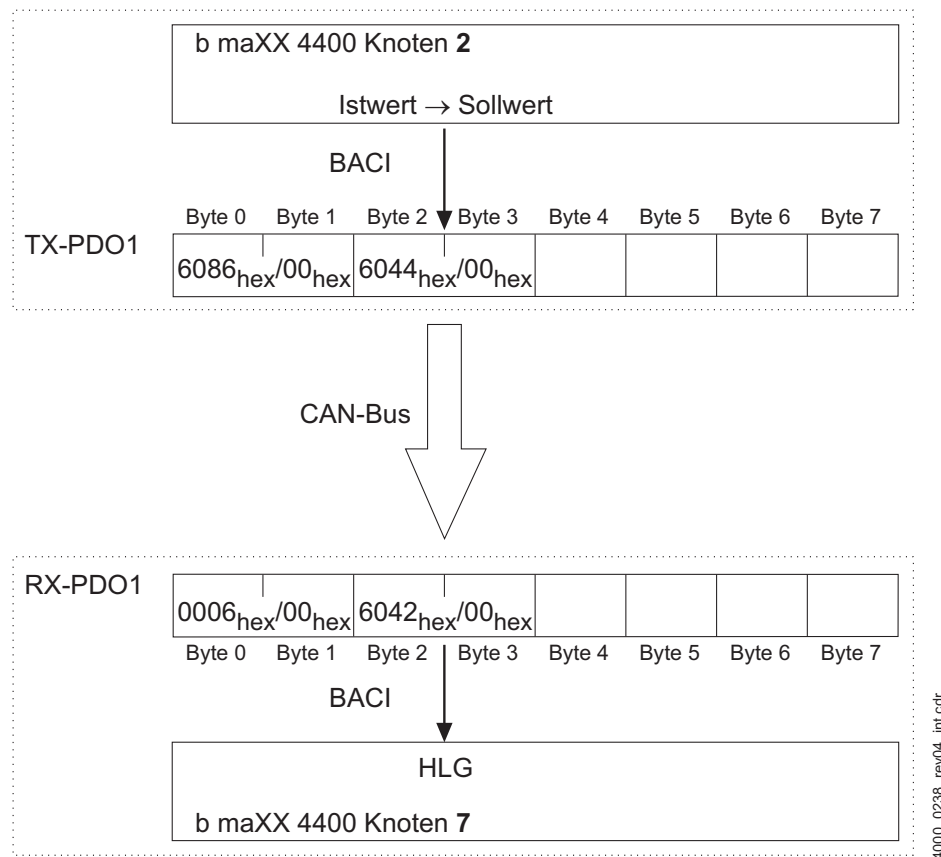


Abbildung 12: Telegrammaufbau für Beispiel-Mapping

Von der TX-PDO1 des b maXX[®] 4400 Knoten 2 werden im b maXX[®] 4400 Knoten 7 nur die Bytes 2 und 3 ausgewertet, weil in der RX-PDO1 nur die Byte 2 und 3 gültig mit Parameternummern verknüpft sind.

Über SDOs werden die Kommunikationsparameter beider Slaves eingestellt. Damit eine Kommunikationsbeziehung aufgebaut werden kann, müssen die COB-IDs von Sender und Empfänger übereinstimmen. Die COB-ID der TX-PDO1 von Slave 2 ist defaultmäßig auf 182_{hex} eingestellt. Die COB-ID der RX-PDO1 von Slave 7 lautet defaultmäßig 207_{hex}. Welche COB-ID verwendet werden soll, liegt in der Verantwortung des Anwenders. Im Beispiel wird die COB-ID 207_{hex} verwendet.

Kommunikationsparameter für Slave 2:

Auf Objekt 1800_{hex} Subindex 01_{hex} wird der Wert 207_{hex} eingetragen. Auf Subindex 02_{hex} wird der Wert „FF_{hex}“ (ereignisgesteuert) geschrieben.

1800_{hex}		
	01_{hex}	207 _{hex}
	02_{hex}	FF _{hex}

Auf Objekt 1801_{hex} Subindex 02_{hex} wird der Wert FE_{hex} (Timer-getriggert) eingetragen. Auf Subindex 05_{hex} wird der Wert „0A_{hex}“ geschrieben (Timerwert = 10 ms).

1801_{hex}		
	02_{hex}	FE _{hex}
	05_{hex}	0A _{hex}

Die übrigen Subindizes behalten ihre Defaultwerte.

Der Triggertyp von RX-PDO1 wird auf ereignisgesteuert (1400_{hex} Subindex 02_{hex} = FF_{hex}) gestellt.

Kommunikationsparameter für Slave 7:

Auf Objekt 1400_{hex} Subindex 02_{hex} wird der Wert „FF_{hex}“ (ereignisgesteuert) geschrieben. In Subindex 01_{hex} steht defaultmäßig 207_{hex}.

1400_{hex}		
	01_{hex}	207 _{hex}
	02_{hex}	FF _{hex}

Die beiden TX-PDOs von Knoten 7 behalten ihre Default-COB-IDs. Der Triggertyp von TX-PDO1 wird mit dem Wert 03_{hex} auf SYNC getriggert eingestellt. TX-PDO2 wird als ereignisgesteuert parametriert.

1800_{hex}		
	01_{hex}	187 _{hex}
	02_{hex}	03 _{hex}

1801_{hex}		
	01_{hex}	287 _{hex}
	02_{hex}	FF _{hex}

5.5.4 Eintrag in die BACI

Es können maximal acht zyklische Soll- und acht zyklische Istwerte zwischen dem Optionsmodul CANopen-Slave und dem b maXX[®]-Regler gleichzeitig ausgetauscht werden. Alle Werte werden in einem Zyklus aktualisiert. Die Soll-/Istwerte können bei CANopen auf je vier PDOs verteilt sein.

Zum Beispiel sollen je zwei Sollwerte für TX-PDO1 und TX-PDO2 gemappt werden, dabei ergibt sich folgende BACI- Konfiguration.:

BACI- Position	PDO	PDO- Position
1	TX-PDO1	1. Objekt
2	TX-PDO1	2. Objekt
3	TX-PDO2	1. Objekt
4	TX-PDO2	2. Objekt

Für RX-PDOs gilt das selbe Verfahren.

Die Einträge in der BACI erfolgen lückenfrei, mit dem 1. Objekt von PDO1 beginnend.

HINWEIS



Das Dummy-Objekt wird in der BACI- Initialisierung nicht berücksichtigt.

Beginnend beim ersten Objekt von PDO 1 werden die Inhalte der PDOs abwechselnd auf ihre Gültigkeit für die BACI- Konfiguration (kein Dummy) abgefragt. Ist das Objekt gültig, dann wird dieses an die nächste freie Stelle der BACI- Konfiguration eingetragen. Ist das PDO-Mapping fehlerhaft (falsche Parameternummern o.ä.), wird keine zyklische Kommunikation zwischen Optionskarte und b maXX[®] 4400 gestartet.

HINWEIS



Werden in den vorhandenen PDOs gleicher Richtung mehrmals die selbe Objektnummer gemappt, so erscheint das Objekt nur einmal in der BACI- Konfiguration.

Hierbei muss beachtet werden, dass sich die Daten unter Umständen gegenseitig beeinflussen.

5.6 Fehlertelegramm (EMCY) nach DSP 402

Emergency-Telegramme dienen der Anzeige von b maXX[®] 4400-Fehlern. Dieses Telegramm wird gesendet, sobald der b maXX[®] 4400 einen internen Fehler erkannt hat. Bei jedem neu hinzukommenden Fehler wird einmalig ein Emergency-Telegramm gesendet. Eine Telegrammwiederholung erfolgt nicht.

5.6.1 Telegrammaufbau

Der Nutzdatenbereich des Emergency-Telegramms gliedert sich in drei Teile:

COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
80 _{hex} + Adresse	08 _{hex}	Emergency Error Code		Error Register	Herstellerspezifisches Fehlerfeld				

Nach dem Predefined Connection Set ergibt sich die COB-ID aus 80_{hex} + Knotenadresse.

Der Emergency Error Code (Byte 0, 1) ist in CANopen DSP 402 definiert. Die Umsetzung auf b maXX[®] 4400-Fehlernummern wird in [►Umsetzung der Fehlermeldungen nach DSP 402 V1.1◄](#) ab Seite 80 dargestellt.

Das Error Register entspricht dem Inhalt von Objekt 1001h (siehe auch [►Verzeichnis der Objekte zur Kommunikationssteuerung◄](#) ab Seite 31).

Die ersten beiden Bytes des herstellerspezifischen Fehlerfeldes enthalten die b maXX[®] 4400-Fehlernummer.

Beispiel

Slave 5 hat einen Geberfehler an Geber 1 erkannt (Leitungsbruch Geber 1). Das EMCY-Telegramm hat dann folgende Form:

COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
85 _{hex}	08 _{hex}	00 _{hex}	73 _{hex}	81 _{hex}	0 _{hex}	73 _{hex}	00 _{hex}	00 _{hex}	00 _{hex}

Stehen mehrere Fehler an und wird ein Fehler gelöscht, sendet das Optionsmodul CANopen-Slave das EMCY-Telegramm mit der nächsten Fehlernummer. Sind alle Fehler quittiert, wird vom Slave das Telegramm „Error Reset / No Error“ gesendet.

COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
80 _{hex} + Adresse	08 _{hex}	00 _{hex}	00 _{hex}	00 _{hex}	00 _{hex}	00 _{hex}	00 _{hex}	00 _{hex}	00 _{hex}

5.6 Fehlertelegramm (EMCY) nach DSP 402

5.6.2 Umsetzung der Fehlermeldungen nach DSP 402 V1.1

Die Beschreibung der Regler-Fehlermeldungen und Hinweise zur Beseitigung der Störungen können Sie der Betriebsanleitung b maXX[®] 5.01040 entnehmen. Die folgende Tabelle zeigt die Umsetzung der Regler-Fehlermeldungen auf CANopen-Fehlermeldungen.

Regler Fehlercode	Bezeichnung (vom b maXX [®] -Regler)	CANopen Fehlercode
0000 _{hex}	Reserve	--
0001 _{hex}	Watchdog-Fehler	7400 _{hex}
0002 _{hex}	Falscher oder unerwarteter Interrupt ist aufgetreten	7400 _{hex}
0003 _{hex}	NMI-Interrupt ist aufgetreten - falscher Buszugriff	7400 _{hex}
0010 _{hex}	Fehler beim Booten des Systems	5530 _{hex}
0011 _{hex}	Softwarefehler (u.a. switch)	5530 _{hex}
0012 _{hex}	Fehler beim Konfigurieren des Zeitscheiben-Betriebssystems	7400 _{hex}
0013 _{hex}	Zeitscheiben-Zeitverletzung	7400 _{hex}
0014 _{hex}	kein Speicher mehr frei	7400 _{hex}
0015 _{hex}	Software-Fehler: ungültiger Fehlercode	7400 _{hex}
0016 _{hex}	Software-Fehler: ungültiger Warnungscode	7400 _{hex}
0017 _{hex}	FPGA-Version ist nicht kompatibel zur Firmware	7400 _{hex}
0018 _{hex}	Fehler beim Schreiben auf Zielparameter durch Zweipunktreglerausgang	
0019 _{hex}	Checksummenfehler im ext Flash (Systemdaten wie z.B. Systemzeit)	
001A _{hex}	LT-Typ (z.B. NWR) wird von der Firmware nicht unterstützt	
0020 _{hex}	Timeout ProProg-Protokoll	8100 _{hex}
0021 _{hex}	Protokollfehler	7400 _{hex}
0022 _{hex}	Falscher Modultyp	7400 _{hex}
0023 _{hex}	Zu viele Daten in Liste bzw. Telegramm	7400 _{hex}
0024 _{hex}	Zu wenig Daten in Liste bzw. Telegramm	7400 _{hex}
0025 _{hex}	Ungültiger Operand	7400 _{hex}
0026 _{hex}	Gerät unterstützt nur VARSTAT_MEMORY	7400 _{hex}
0027 _{hex}	Ungültige Operandenadresse (log. Adresse)	7400 _{hex}
0028 _{hex}	Wert kleiner als Minimalwert	7400 _{hex}
0029 _{hex}	Wert größer als Maximalwert	7400 _{hex}
002A _{hex}	Parameter ist read-only	7400 _{hex}

Regler Fehlercode	Bezeichnung (vom b maXX [®] -Regler)	CANopen Fehlercode
002B _{hex}	Parameter kann aufgrund Betriebszustand nicht geändert werden	7400 _{hex}
002C _{hex}	Parameterwert ist ungültig	7400 _{hex}
002D _{hex}	ProDrive/WinBASS ist nicht mehr angeschlossen oder reagiert nicht	7400 _{hex}
0030 _{hex}	Fehler im SmallModule_A (um die genaue Fehlernummer zu erfahren, muss der Parameter P0240 im b maXX [®] gelesen werden, die Fehlerbezeichnung des P0240 ist im Anschluss an diese Tabelle beschrieben)	FF00 _{hex}
0031 _{hex}	Fehler im SmallModule_B (um die genaue Fehlernummer zu erfahren, muss der Parameter P0240 im b maXX [®] gelesen werden, die Fehlerbezeichnung des P0241 ist im Anschluss an diese Tabelle beschrieben)	FF00 _{hex}
0032 _{hex}	Fehler im SmallModule_C (um die genaue Fehlernummer zu erfahren, muss der Parameter P0242 im b maXX [®] gelesen werden, die Fehlerbezeichnung des P0240 ist im Anschluss an diese Tabelle beschrieben)	FF00 _{hex}
0033 _{hex}	Fehler im SmallModule_D (um die genaue Fehlernummer zu erfahren, muss der Parameter P0243 im b maXX [®] gelesen werden, die Fehlerbezeichnung des P0240 ist im Anschluss an diese Tabelle beschrieben)	FF00 _{hex}
0034 _{hex}	Fehler im SmallModule_F (um die genaue Fehlernummer zu erfahren, muss der Parameter P0244 im b maXX [®] gelesen werden, die Fehlerbezeichnung des P0240 ist im Anschluss an diese Tabelle beschrieben)	FF00 _{hex}
0035 _{hex}	Fehler im BigModule_G (um die genaue Fehlernummer zu erfahren, muss der Parameter P0245 im b maXX [®] gelesen werden, die Fehlerbezeichnung des P0240 ist im Anschluss an diese Tabelle beschrieben)	FF00 _{hex}
0036 _{hex}	Fehler im BigModule_H (um die genaue Fehlernummer zu erfahren, muss der Parameter P0246 im b maXX [®] gelesen werden, die Fehlerbezeichnung des P0240 ist im Anschluss an diese Tabelle beschrieben)	FF00 _{hex}
0037 _{hex}	Fehler im BigModule_J (um die genaue Fehlernummer zu erfahren, muss der Parameter P0247 im b maXX [®] gelesen werden, die Fehlerbezeichnung des P0240 ist im Anschluss an diese Tabelle beschrieben)	FF00 _{hex}
0038 _{hex}	Fehler im BigModule_K (um die genaue Fehlernummer zu erfahren, muss der Parameter P0248 im b maXX [®] gelesen werden, die Fehlerbezeichnung des P0240 ist im Anschluss an diese Tabelle beschrieben)	FF00 _{hex}
0039 _{hex}	Fehler im BigModule_L (um die genaue Fehlernummer zu erfahren, muss der Parameter P0249 im b maXX [®] gelesen werden, die Fehlerbezeichnung des P0240 ist im Anschluss an diese Tabelle beschrieben)	FF00 _{hex}

5.6 Fehlertelegramm (EMCY) nach DSP 402

Regler Fehlercode	Bezeichnung (vom b maXX [®] -Regler)	CANopen Fehlercode
003A _{hex}	Fehler im BigModule_M (um die genaue Fehlernummer zu erfahren, muss der Parameter P0250 im b maXX [®] gelesen werden, die Fehlerbezeichnung des P0240 ist im Anschluss an diese Tabelle beschrieben)	FF00 _{hex}
003B _{hex}	Timeout beim Systemhochlauf	7400 _{hex}
003C _{hex}	CRC-Fehler in SPI-Übertragung Modul ⇒ Regler	7400 _{hex}
003D _{hex}	CRC-Fehler in SPI-Übertragung Regler ⇒ Modul	7400 _{hex}
0040 _{hex}	Netzausfall	3100 _{hex}
0041 _{hex}	Phasenausfall	3130 _{hex}
0042 _{hex}	Unterspannung Netz	3120 _{hex}
0043 _{hex}	Überspannung Netz	3110 _{hex}
0044 _{hex}	Unterspannung 24 V	3100 _{hex}
0050 _{hex}	Kommunikationsfehler nach Hiperface-Spezifikation (um die dazugehörige Fehlernummer zu erfahren muss der Parameter P0233 im b maXX [®] gelesen werden, die Fehler-Bezeichnung ist im Anschluss an dieser Tabelle beschrieben)	FF00 _{hex}
0051 _{hex}	Temperaturschwelle des Kühlkörpers ist überschritten	4210 _{hex}
0052 _{hex}	Überspannung Uzk	3210 _{hex}
0053 _{hex}	Überstrom Leistungsteil	2310 _{hex}
0054 _{hex}	Erdstrom	2240 _{hex}
0055 _{hex}	Temperaturschwelle der Innenluft ist überschritten	4110 _{hex}
0057 _{hex}	Sicherheitsrelais aus (bzw. Sicherheitsrelais defekt)	5441 _{hex}
0058 _{hex}	Sicherheitsrelais aus (Sicherheitsrelais O.K., aber Spannung nicht vorhanden)	5442 _{hex}
0059 _{hex}	Leistungsteil nicht betriebsbereit	5400 _{hex}
005A _{hex}	Phasenausfall	3130 _{hex}
005B _{hex}	Netzausfall	3100 _{hex}
005C _{hex}	Unterspannung Netz	3120 _{hex}
005D _{hex}	Überspannung Netz	3110 _{hex}
005E _{hex}	Unterspannung Uzk	3100 _{hex}
0060 _{hex}	Temperaturfühler des Motors kurzgeschlossen ($T_M \leq -30 \text{ °C}$)	4320 _{hex}
0061 _{hex}	Temperaturfühler des Motors nicht angeschlossen ($T_M > +300 \text{ °C}$)	4310 _{hex}
0062 _{hex}	Fehler Motortemperatur - Abschaltchwelle überschritten	4310 _{hex}

Regler Fehlercode	Bezeichnung (vom b maXX [®] -Regler)	CANopen Fehlercode
0063 _{hex}	Fehler $I^2t > 100\%$ im Motor	7120 _{hex}
0064 _{hex}	Fehler CurrentDriveMax > MotorPeakCurrent	
0070 _{hex}	Kommunikationsfehler nach Hiperface-Spezifikation (um die dazugehörige Fehlernummer zu erfahren muss der Parameter P0234/P0235 im b maXX [®] gelesen werden, die Fehler-Bezeichnung ist im Anschluss an dieser Tabelle beschrieben)	FF00 _{hex}
0071 _{hex}	Modulcode ungültig	7300 _{hex}
0072 _{hex}	Fehler beim Schreiben der Geberlage	7300 _{hex}
0073 _{hex}	Leistungsbruch Geber 1	7300 _{hex}
0074 _{hex}	Überdrehzahl Geber 1	7310 _{hex}
0075 _{hex}	Amplitudengrenze überschritten	7300 _{hex}
0076 _{hex}	Gebertyp unbekannt	7300 _{hex}
0077 _{hex}	Datenfeld für Motordaten ungültig	7120 _{hex}
0078 _{hex}	Motordaten ungültig	7120 _{hex}
0079 _{hex}	Fehler beim Speichern der Motordaten	7120 _{hex}
007A _{hex}	Motordaten schreibgeschützt (nicht BM-Motoren)	7120 _{hex}
007B _{hex}	Feldwinkel Fehler	7300 _{hex}
007C _{hex}	Geber ohne Temperaturerfassung	7300 _{hex}
007D _{hex}	EEPROM-Kapazität im Geber nicht ausreichend	
0080 _{hex}	Kommunikationsfehler nach Hiperface-Spezifikation (um die dazugehörige Fehlernummer zu erfahren muss der Parameter P0235 im b maXX [®] gelesen werden, die Fehler-Bezeichnung ist im Anschluss an dieser Tabelle beschrieben)	FF00 _{hex}
0081 _{hex}	Modulcode ungültig	7300 _{hex}
0082 _{hex}	Fehler beim Schreiben der Geberlage	7300 _{hex}
0083 _{hex}	Leistungsbruch Geber 2	7300 _{hex}
0084 _{hex}	Überdrehzahl Geber 2	7310 _{hex}
0085 _{hex}	Amplitudengrenze überschritten	7300 _{hex}
0086 _{hex}	Gebertyp unbekannt	7300 _{hex}
0087 _{hex}	Datenfeld für Motordaten ungültig	7120 _{hex}
0088 _{hex}	Motordaten ungültig	7120 _{hex}
0089 _{hex}	Fehler beim Speichern der Motordaten	7120 _{hex}
008A _{hex}	Motordaten schreibgeschützt (nicht BM-Motoren)	7120 _{hex}
008B _{hex}	Feldwinkel Fehler	7300 _{hex}
008C _{hex}	Geber ohne Temperaturerfassung	7300 _{hex}

5.6 Fehlertelegramm (EMCY) nach DSP 402

Regler Fehlercode	Bezeichnung (vom b maXX [®] -Regler)	CANopen Fehlercode
008D _{hex}	EEPROM-Kapazität im Geber nicht ausreichend	
0090 _{hex}	Absolutlage des Geber nicht bekannt	7320 _{hex}
0091 _{hex}	Absolutlage des Geber 2 nicht bekannt	7320 _{hex}
0092 _{hex}	Geber Modul 1 fehlt	7300 _{hex}
0093 _{hex}	Geber Modul 2 fehlt	7300 _{hex}
0094 _{hex}	Geber Modul fehlt für Messwertspeicherung	7300 _{hex}
0095 _{hex}	Bei Resolver keine Messwertspeicherung möglich	7300 _{hex}
0096 _{hex}	Triggerung auf Nullimpuls und Geber ist kein Inkrementalgeber	7300 _{hex}
0097 _{hex}	Digital I/O-Modul wird benötigt und fehlt	7300 _{hex}
0098 _{hex}	Inkrementalgeber-Nachbildungsmodul wird benötigt und fehlt	7300 _{hex}
0099 _{hex}	Gebermodul 1 wird für Inkrementalgeber-Nachbildung benötigt und fehlt	7300 _{hex}
009A _{hex}	Gebermodul 2 wird für Inkrementalgeber-Nachbildung benötigt und fehlt	7300 _{hex}
009B _{hex}	Fehler beim Initialisieren des Inkrementalgeber-Nachbildungsmoduls	7300 _{hex}
009C _{hex}	Inkrementalgeber-Nachbildungsmodul meldet Fehler	7300 _{hex}
009D _{hex}	Inkrementalgeber-Nachbildung: Option „Start nach erstem Nullpuls“ für Nicht-Inkrementalgeber gewählt	7300 _{hex}
009E _{hex}	SSI-Geber-Nachbildung Modul fehlt	7300 _{hex}
009F _{hex}	Fehler in Sollwertquelle Geber 1 bzw. Geber 2	7300 _{hex}
00A0 _{hex}	Zeitüberwachung Proprog Kommunikation	8100 _{hex}
00A1 _{hex}	Zeitüberwachung BACI Kommunikation	8100 _{hex}
00A2 _{hex}	Zeitüberwachung zyklische Kommunikation	8110 _{hex}
00A3 _{hex}	Zeitüberwachung Bedarfsdaten Übertragung	8100 _{hex}
00A4 _{hex}	Feldbus Fehler	8100 _{hex}
00A5 _{hex}	Regler nicht synchron zu externem Signal	8100 _{hex}
00A6 _{hex}	Fehler bei der Bremsansteuerung	8100 _{hex}
00A7 _{hex}	Bremse oeffnet nicht	
00A8 _{hex}	Bremse schließt nicht	
00A9 _{hex}	Zyklische Bremszustandsueberwachung meldet Fehler	
00AA _{hex}	Zyklische Bremsbelagsueberwachung meldet Fehler	

Regler Fehlercode	Bezeichnung (vom b maXX [®] -Regler)	CANopen Fehlercode
00AB _{hex}	DIO-Modul zur Ansteuerung/Rueckmeldung der Bremse fehlt	
00AC _{hex}	Haltemoment vor Öffnen der Bremse nicht erreicht	
00B0 _{hex}	reserviert	5530 _{hex}
00B1 _{hex}	reserviert	5530 _{hex}
00B2 _{hex}	Checksummenfehler im EEPROM	5530 _{hex}
00B3 _{hex}	Kein Bootdatensatz	5530 _{hex}
00B4 _{hex}	Inkompatibel SW	5530 _{hex}
00B5 _{hex}	Datensatzumschaltung: DS nicht vorhanden	5530 _{hex}
00B6 _{hex}	Checksummenfehler im PSI	5530 _{hex}
00B7 _{hex}	PSI ist gelöscht	5530 _{hex}
00B8 _{hex}	PSI-Daten sind ungültig	5530 _{hex}
00B9 _{hex}	Selbstopoptimierungstabellen ungültig - Selbstopoptimierung erneut durchführen	5530 _{hex}
00BA _{hex}	A/D-Korrekturtabelle ungültig	5530 _{hex}
00BB _{hex}	EEPROM ist gelöscht	
00C0 _{hex}	Schleppfehler dynamisch	8611 _{hex}
00C1 _{hex}	Schleppfehler statisch	8611 _{hex}
00C2 _{hex}	Geber 1 für Lageregelung verwendet aber nicht aktiv	7300 _{hex}
00C3 _{hex}	Geber 2 für Lageregelung verwendet aber nicht aktiv	7300 _{hex}
00C4 _{hex}	Software-Endschalterüberwachung 1 aktiv	8600 _{hex}
00C5 _{hex}	Software-Endschalterüberwachung 2 aktiv	8600 _{hex}
00C6 _{hex}	Hardware-Endschalterüberwachung 1 aktiv	8600 _{hex}
00C7 _{hex}	Hardware-Endschalterüberwachung 2 aktiv	8600 _{hex}
00C8 _{hex}	Positionierung ohne Referenzfahrt gestartet	8600 _{hex}
00C9 _{hex}	Sollwert im Modus Set-of-setpoints nicht rechtzeitig angekommen	8600 _{hex}
00CA _{hex}	Überwachung der Modulo-Position aktiv: Zielposition > Modulo-Position	8600 _{hex}
00CB _{hex}	Spindelpositionierung: Fehler bei Initialisierung des Triggers	8600 _{hex}
00CC _{hex}	Spindelpositionierung: Timeout beim Triggersignal (Nullimpuls/Schalteingangs)	8600 _{hex}
00CD _{hex}	Fehler bei Homing (Referenzfahrt) aufgetreten und Vorgang abgebrochen	
00D0 _{hex}	Antrieb blockiert	7121 _{hex}
00D1 _{hex}	Geber 1 für Motorführung verwendet aber nicht aktiv	7300 _{hex}

5.6 Fehlertelegramm (EMCY) nach DSP 402

Regler Fehlercode	Bezeichnung (vom b maXX [®] -Regler)	CANopen Fehlercode
00D2 _{hex}	Geber 2 für Motorführung verwendet aber nicht aktiv	7300 _{hex}
00D3 _{hex}	Überdrehzahl Open Loop	7300 _{hex}
00E0 _{hex}	Uebersetzungsfehler	
00E1 _{hex}	Laufzeitfehler	
00EA _{hex}	Fehler Schnellentladung (z.Zt. nur im BM41XX (NRW))	
00EB _{hex}	Slave hat sich wegen Fehler im Master abgeschaltet (Folgereaktion auf Fehler im Master, wenn so parametrisiert)	
00EC _{hex}	Keine BA Drehzahlregelung, obwohl Antrieb als Slave	
00ED _{hex}	Sammelfehler Rückzug Konfiguration ungültig (z.B. motorischer Betrieb bei Netzausfall nicht erlaubt)	
00EE _{hex}	Rückzugspositionierung konnte nicht beendet werden	
00EF _{hex}	Applikationsfehler (ausgelöst durch Steuerwort2)	
<p>Die herstellereigenen Error Codes 0030_{hex} bis 003B_{hex}, 0050_{hex}, 0070_{hex} und 0080_{hex} werden zusammenfassend über ein EMY- Telegramm mit dem CANopen Error Code FF00_{hex} ausgegeben. Die genaue Bezeichnung dafür kann in den folgenden Parameter P0233, P0234, P0235 und P0240 bis P0250 abgelesen werden und die Beschreibung zu den ausgelesenen Fehlernummern ist nachfolgend aufgezeichnet.</p>		

Regler Fehlercode	Bezeichnung (vom b maXX [®] -Regler)	CANopen Fehlercode
<p>Die nachfolgenden b maXX[®] Error Codes werden nicht über ein EMY- Telegramm ausgegeben.</p> <p>Error Code (0050_{hex}) ⇒ P0233 Kommunikationsfehler nach Hiperface-Spezifikation (AmpHiperfaceError)</p>		
06 _{hex}	Datenüberlauf	
07 _{hex}	Bitrahmen- Fehler	
08 _{hex}	Ungültiger Kommandozustand	
09 _{hex}	Parity- Fehler	
0A _{hex}	Check-Summe der übertragenen Daten ist falsch	
0B _{hex}	Unbekannte Befehlscode	
0C _{hex}	Anzahl der übertragenen Daten falsch	
0D _{hex}	Unzulässiges Argument	
0E _{hex}	Datenfeld ist nicht beschreibbar	
0F _{hex}	Falscher Zugriffcode	
10 _{hex}	Datenfeld in seiner Größe nicht veränderbar	
11 _{hex}	Angegebene Wortadresse außerhalb Datenfeld	
12 _{hex}	Zugriff auf nicht existierendes Datenfeld	
24 _{hex}	Falsche LT-Daten Checksumme	
25 _{hex}	keine Antwort vom LT	
42 _{hex}	Ungültige Antwort	
<p>Error Code (0070_{hex}) ⇒ P0234 Kommunikationsfehler nach Hiperface-Spezifikation (Enc1HiperfaceError)</p> <p>Error Code (0080_{hex}) ⇒ P0235 Kommunikationsfehler nach Hiperface-Spezifikation (Enc2HiperfaceError)</p>		
01 _{hex}	Analogsignale außerhalb Spezifikation	
02 _{hex}	Interner Winkeloffset fehlerhaft	
03 _{hex}	Tabelle über Datenfeldpartitionierung zerstört	
04 _{hex}	Analoge Grenzwerte nicht verfügbar	
05 _{hex}	Interner I2C-Bus nicht funktionsfähig	
06 _{hex}	Interner Checksummenfehler	
07 _{hex}	Interner Watchdog-Fehler - Geber-Reset	
09 _{hex}	Parity- Fehler	
0A _{hex}	Checksumme der übertragenen Daten ist falsch	
0B _{hex}	Unbekannte Befehlscode	
0C _{hex}	Anzahl der übertragenen Daten falsch	

5.6 Fehlertelegramm (EMCY) nach DSP 402

Regler Fehlercode	Bezeichnung (vom b maXX [®] -Regler)	CANopen Fehlercode
0D _{hex}	Unzulässiges Argument	
0E _{hex}	Datenfeld ist nicht beschreibbar	
0F _{hex}	Falscher Zugriffscode	
10 _{hex}	Datenfeld in seiner Größe nicht veränderbar	
11 _{hex}	Angegebene Wortadresse außerhalb Datenfeld	
12 _{hex}	Zugriff auf nicht existierendes Datenfeld	
1C _{hex}	Betragsüberwachung der Analogsignale	
1D _{hex}	Sendestrom kritisch	
1E _{hex}	Gebertemperatur kritisch	
1F _{hex}	Drehzahl zu hoch - keine Positionsbildung möglich	
20 _{hex}	Position Singleturn unzulässig	
21 _{hex}	Positionsfehler Multiturn	
22 _{hex}	Positionsfehler Multiturn	
23 _{hex}	Positionsfehler Multiturn	
24 _{hex}	Falsche MT-Daten Checksumme	
40 _{hex}	Keine Antwort vom Hiperface-Geber	
41 _{hex}	Keine Antwort vom EnDatGeber	
42 _{hex}	unbrauchbare Antwort auf Geberkommando	
50 _{hex}	CRC hat Fehler festgestellt	
51 _{hex}	Ungültiges Kommando	
52 _{hex}	Adresse bzw. MRS-Code im Antwort-Telegramm falsch	
53 _{hex}	Alarmbit des Gebers ist gesetzt	
54 _{hex}	Speicher im Geber ist belegt	
55 _{hex}	Checksummenfehler beim Lesen der Motordaten	
56 _{hex}	Motordaten-Datenlänge und/oder Datenversion von Geber und Reglerfirmware sind nicht identisch	
57 _{hex}	Einschalttest hat keine EnDat-Schnittstelle am Geber festgestellt	
58 _{hex}	Überschreitung des auswertbaren Übertragungsformates	
59 _{hex}	Überschreitung der auswertbaren Messschrittlänge	
5A _{hex}	Signalperiodenlänge < Messschrittlänge	
60 _{hex}	Fehler Beleuchtung	
61 _{hex}	Fehler Signalamplitude	
62 _{hex}	Fehler Positionswert	

Regler Fehlercode	Bezeichnung (vom b maXX [®] -Regler)	CANopen Fehlercode
63 _{hex}	Fehler Überspannung	
64 _{hex}	Fehler Unterspannung	
65 _{hex}	Fehler Überstrom	
66 _{hex}	Fehler Batterie	
Error Code (0030 _{hex} 0034 _{hex}) ⇒ P0240...P0244 Fehler im SmallModule 1 bis 5		
01 _{hex}	Modul nicht erkannt	
02 _{hex}	Erkannter Modul am unzulässigen Platz	
03 _{hex}	Digitaler Ausgang kurzgeschlossen	
04 _{hex}	Falscher Zielparameterwert durch digitalen Eingang	
05 _{hex}	Direkter PLC-IO-Zugriff für dieses Modul nicht erlaubt	
07 _{hex}	Modul im Regler nicht erlaubt	
Error Code (0035 _{hex} 0040 _{hex}) ⇒ P0245...P0250 Fehler im BigModule 1 bis 6		
1000 _{hex}	Falsche Parameter-Nr. bei Sollwert Parameter 1	
1001 _{hex}	Falsche Parameter-Nr. bei Sollwert Parameter 2	
1002 _{hex}	Falsche Parameter-Nr. bei Sollwert Parameter 3	
1003 _{hex}	Falsche Parameter-Nr. bei Sollwert Parameter 4	
1004 _{hex}	Falsche Parameter-Nr. bei Sollwert Parameter 5	
1005 _{hex}	Falsche Parameter-Nr. bei Sollwert Parameter 6	
1006 _{hex}	Falsche Parameter-Nr. bei Sollwert Parameter 7	
1007 _{hex}	Falsche Parameter-Nr. bei Sollwert Parameter 8	
1008 _{hex}	Falsche Parameter-Nr. bei Sollwert Parameter 9	
1009 _{hex}	Falsche Parameter-Nr. bei Sollwert Parameter 10	
100A _{hex}	Falsche Parameter-Nr. bei Sollwert Parameter 11	
100B _{hex}	Falsche Parameter-Nr. bei Sollwert Parameter 12	
100C _{hex}	Falsche Parameter-Nr. bei Sollwert Parameter 13	
100D _{hex}	Falsche Parameter-Nr. bei Sollwert Parameter 14	
100E _{hex}	Falsche Parameter-Nr. bei Sollwert Parameter 15	
100F _{hex}	Falsche Parameter-Nr. bei Sollwert Parameter 16	
1010 _{hex}	Falsche Parameter-Nr. bei Istwert Parameter 1	
1011 _{hex}	Falsche Parameter-Nr. bei Istwert Parameter 2	
1012 _{hex}	Falsche Parameter-Nr. bei Istwert Parameter 3	
1013 _{hex}	Falsche Parameter-Nr. bei Istwert Parameter 4	
1014 _{hex}	Falsche Parameter-Nr. bei Istwert Parameter 5	

5.6 Fehlertelegramm (EMCY) nach DSP 402

Regler Fehlercode	Bezeichnung (vom b maXX [®] -Regler)	CANopen Fehlercode
1015 _{hex}	Falsche Parameter-Nr. bei Istwert Parameter 6	
1016 _{hex}	Falsche Parameter-Nr. bei Istwert Parameter 7	
1017 _{hex}	Falsche Parameter-Nr. bei Istwert Parameter 8	
1018 _{hex}	Falsche Parameter-Nr. bei Istwert Parameter 9	
1019 _{hex}	Falsche Parameter-Nr. bei Istwert Parameter 10	
101A _{hex}	Falsche Parameter-Nr. bei Istwert Parameter 11	
101B _{hex}	Falsche Parameter-Nr. bei Istwert Parameter 12	
101C _{hex}	Falsche Parameter-Nr. bei Istwert Parameter 13	
101D _{hex}	Falsche Parameter-Nr. bei Istwert Parameter 14	
101E _{hex}	Falsche Parameter-Nr. bei Istwert Parameter 15	
101F _{hex}	Falsche Parameter-Nr. bei Istwert Parameter 16	
1020 _{hex}	Ungültiger Wert bei Sollwert-Parameter-Nr. 1	
1021 _{hex}	Ungültiger Wert bei Sollwert-Parameter-Nr. 2	
1022 _{hex}	Ungültiger Wert bei Sollwert-Parameter-Nr. 3	
1023 _{hex}	Ungültiger Wert bei Sollwert-Parameter-Nr. 4	
1024 _{hex}	Ungültiger Wert bei Sollwert-Parameter-Nr. 5	
1025 _{hex}	Ungültiger Wert bei Sollwert-Parameter-Nr. 6	
1026 _{hex}	Ungültiger Wert bei Sollwert-Parameter-Nr. 7	
1027 _{hex}	Ungültiger Wert bei Sollwert-Parameter-Nr. 8	
1028 _{hex}	Ungültiger Wert bei Sollwert-Parameter-Nr. 9	
1029 _{hex}	Ungültiger Wert bei Sollwert-Parameter-Nr. 10	
102A _{hex}	Ungültiger Wert bei Sollwert-Parameter-Nr. 11	
102B _{hex}	Ungültiger Wert bei Sollwert-Parameter-Nr. 12	
102C _{hex}	Ungültiger Wert bei Sollwert-Parameter-Nr. 13	
102D _{hex}	Ungültiger Wert bei Sollwert-Parameter-Nr. 14	
102E _{hex}	Ungültiger Wert bei Sollwert-Parameter-Nr. 15	
102F _{hex}	Ungültiger Wert bei Sollwert-Parameter-Nr. 16	
1030 _{hex}	Ungültiger Wert für Sollwert-Periode	
1031 _{hex}	Ungültiger Wert für Istwert-Periode	
1032 _{hex}	Falscher Wert für Cycle-Offset Sollwerte	
1033 _{hex}	Falscher Wert für Cycle-Offset Istwerte	
1034 _{hex}	BACI-Timeout bei zyklischen Daten	
1035 _{hex}	BACI-Timeout bei Bedarfsdaten	
1036 _{hex}	Überprüfung ergab fehlerhafte Checksumme	

Regler Fehlercode	Bezeichnung (vom b maXX [®] -Regler)	CANopen Fehlercode
1037 _{hex}	Hochlauf: Timeout beim Warten auf Slave-Type bzw. beim Warten auf Rücksetzen von Config-Pending-Flag	
1038 _{hex}	Falscher Datentransfer-Struktur-Typ	
1039 _{hex}	Interner Fehler: Falscher BACI-Zustand	
103A _{hex}	Zugriffskonflikt mit Slave bei zyklischer Kommunikation	
103B _{hex}	Fehler zyklische Kommunikation: Parameterwert falsch	
103C _{hex}	Fehler zyklische Kommunikation: Alive-Counter Konflikt	
103D _{hex}	Cmd-Interface: Kanalnummer falsch (0 oder > 6)	
103E _{hex}	Cmd-Interface: Angegebener Kanal existiert nicht	
103F _{hex}	Cmd-Interface: Interner Fehler - Falscher Pointer	
1040 _{hex}	Cmd-Interface: Interner Fehler - Falscher Zustand	
1041 _{hex}	Cmd-Interface: Falsche Paketnummer	
1042 _{hex}	Cmd-Interface: Falsche Kommandonummer	
1043 _{hex}	Cmd-Interface: Falscher Zustand bei Pakethandling	
1044 _{hex}	Cmd-Interface: Timeout bei Kommandobearbeitung	
1045 _{hex}	Cmd-Interface: Falsche Paketlänge	
1046 _{hex}	Cmd-Interface: Kein Deskriptor mehr verfügbar (zu wenig Speicher)	
1047 _{hex}	Cmd-Interface: Falscher Paktetyp	
1048 _{hex}	Cmd-Interface: Checksummenfehler	
1049 _{hex}	Modulkennung: PCI-Fehler beim Lesen	
104A _{hex}	Modulkennung: PCI-Fehler beim Schreiben	
104B _{hex}	Modulkennung: allgemeiner Fehler beim Lesen	
104C _{hex}	Modulkennung: allgemeiner Fehler beim Schreiben	
104D _{hex}	Interner Fehler	
104E _{hex}	Konfiguration zyklische Dienste: Parameter nicht bzw. nicht zykl. beschreibbar	
104F _{hex}	Konfiguration zyklische Dienste: Ungültige Parameternummer	
1050 _{hex}	Falscher Optionsmodule Fehlercode (einstellbar mit P1007)	
2000 _{hex}	Fehler CANopen-Timeout auf CAN-Bus (Node Guarding)	



ANHANG A - ABKÜRZUNGEN

BACI	Baumüller Antriebe serielle Schnittstelle
CA	Collision Avoidance
CSMA	Carrier Sense Multiple Access
DS	Draft Standard
DSP	Draft Standard Proposal
EMCY	Fehlertelegramm
HD	Hamming Distanz
ID	Ident-Nummer
LMT	Layer-Management
M	Multiplexer
NMT	Netzwerk-Management
PC	Personal Computer
PDO	Prozessdaten
SDO	Servicedaten
SIX	Index
SPS	speicherprogrammierte Steuerung
SYNC	Synchronisation



ANHANG B - KURZREFERENZ

Die folgende Kurzreferenz zeigt den Zusammenhang zwischen CANopen-Objektnummer und dem b maXX[®] Regler-Parameternummern (siehe Betriebsanleitung b maXX[®] 5.02017).

B.1 4000'er Objektnummern (Herstellerspezifische Objekte)

Herstellerspezifische Objekte ergeben sich aus
 $4000_{\text{hex}} + \text{Parameternummer}_{\text{hex}}$
Deshalb ist der Subindex für alle 4000'er-Parameter immer 00_{hex} .

Beispiel Parameter **P0053** \Rightarrow Objektindex 4035_{hex} Subindex 00_{hex}

B.2 6000'er Objektnummern (Device Profile DSP 402)

Auf zahlreiche Parameter ist es möglich sowohl über 4000'er **als** auch über ein oder mehrere 6000'er Objekte zuzugreifen.

Es existieren nur wenige Parameter, auf die ausschließlich mit einem 6000'er-Parameter zugegriffen werden kann (606A_{hex}, 6048_{hex} SIX 1, 6049_{hex} SIX 1, 604C_{hex} SIX 1 und SIX 2).

HINWEIS



Zwischen den 6000'er und 4000'er Objekten können unterschiedliche Normierungen auftreten!

TX: Transmit; RX: Receive; r: read; w: write; ro: read only; wo: write only

CANopen-Objekt-Nummer		Parameter-Nummer	PDO-Mapping	Access - Typ	Betriebsart nach DSP 402
Index	Subindex				
6007 _{hex}	00 _{hex}	P0300	TX / RX	rw	Common Entries
6040 _{hex}	00 _{hex}	P0300	TX / RX	rw	Device Control
6041 _{hex}	00 _{hex}	P0301	TX	ro	Device Control
6042 _{hex}	00 _{hex}	P1171	TX / RX	rw	Velocity Mode
6043 _{hex}	00 _{hex}	P0351	TX	ro	Velocity Mode
6044 _{hex}	00 _{hex}	P0353	TX	ro	Velocity Mode
6046 _{hex}	01 _{hex}	P1041	TX	ro	Velocity Mode
6046 _{hex}	02 _{hex}	P1041, P1042	TX / RX	rw	Velocity Mode
6048 _{hex}	01 _{hex}	P3329	TX / RX	rw	Velocity Mode
6048 _{hex}	02 _{hex}	P1172	TX / RX	rw	Velocity Mode
6049 _{hex}	01 _{hex}	P3330	TX / RX	rw	Velocity Mode
6049 _{hex}	02 _{hex}	P1173	TX / RX	rw	Velocity Mode
604C _{hex}	01 _{hex}	P3314	TX / RX	rw	Velocity Mode
604C _{hex}	02 _{hex}	P3315	TX / RX	rw	Velocity Mode
604D _{hex}	00 _{hex}	P0065	TX	rw	Velocity Mode
605E _{hex}	00 _{hex}	P1007	TX		Device Control
604F _{hex}	00 _{hex}	P1172	TX / RX	rw	Velocity Mode
6050 _{hex}	00 _{hex}	P1173	TX / RX	rw	Velocity Mode
6051 _{hex}	00 _{hex}	P1174	TX / RX	rw	Velocity Mode

CANopen-Objekt-Nummer		Parameternummer	PDO-Mapping	Access - Typ	Betriebsart nach DSP 402
Index	Subindex				
605A _{hex}	00 _{hex}	P1004	TX	rw	Device Control
605B _{hex}	00 _{hex}	P1005	TX	rw	Device Control
605C _{hex}	00 _{hex}	P1006	TX	rw	Device Control
605D _{hex}	00 _{hex}	P1003	TX	rw	Device Control
6060 _{hex}	00 _{hex}	P1000	- / RX	rw	Device Control
6061 _{hex}	00 _{hex}	P0304	TX	ro	Device Control
6062 _{hex}	00 _{hex}	P0463	TX	ro	Position Control Function
6063 _{hex}	00 _{hex}	P0362	TX	ro	Position Control Function
6064 _{hex}	00 _{hex}	P0462	TX	ro	Position Control Function
6066 _{hex}	00 _{hex}	P1056	TX	rw	Position Control Function
6067 _{hex}	00 _{hex}	P1194	TX / RX	rw	Position Control Function
6068 _{hex}	00 _{hex}	P1195	TX	rw	Position Control Function
6069 _{hex}	00 _{hex}	P0362	TX / RX	rw	Profile Velocity Mode
606A _{hex}	00 _{hex}	-	-	ro	Profile Velocity Mode
606B _{hex}	00 _{hex}	P0352	TX	ro	Profile Velocity Mode
606C _{hex}	00 _{hex}	P0353	TX	ro	Profile Velocity Mode
606F _{hex}	00 _{hex}	P1073	TX / RX	rw	Profile Velocity Mode
6072 _{hex}	00 _{hex}	P0357	TX / RX	rw	Profile Torque Mode
6077 _{hex}	00 _{hex}	P0344	TX	ro	Profile Torque Mode
607A _{hex}	00 _{hex}	P0600/ P0607	TX / RX	rw	Profile Position Mode
607C _{hex}	00 _{hex}	P1200	TX / RX	rw	Homing Mode
607D _{hex}	01 _{hex}	P1196	TX	rw	Profile Position Mode
607D _{hex}	02 _{hex}	P1197	TX	rw	Profile Position Mode
607F _{hex}	00 _{hex}	P0057	TX	rw	Profile Position Mode
6080 _{hex}	00 _{hex}	P1031	TX	rw	Profile Position Mode
6081 _{hex}	00 _{hex}	P0602	TX	rw	Profile Position Mode
6083 _{hex}	00 _{hex}	P0603	TX	rw	Profile Position Mode
6084 _{hex}	00 _{hex}	P0604	TX	rw	Profile Position Mode
6085 _{hex}	00 _{hex}	P1213	TX	rw	Profile Position Mode
6086 _{hex}	00 _{hex}	P1190	TX	rw	Profile Position Mode

CANopen-Objekt-Nummer		Parameternummer	PDO-Mapping	Access - Typ	Betriebsart nach DSP 402
Index	Subindex				
6092 _{hex}	01 _{hex}	P1193	TX	rw	Factor Group
6092 _{hex}	02 _{hex}	P3050	TX	rw	Factor Group
6098 _{hex}	00 _{hex}	P3051	TX	rw	Homing Mode
6099 _{hex}	01 _{hex}	P1201	TX / RX	rw	Homing Mode
6099 _{hex}	02 _{hex}	P1202	TX / RX	rw	Homing Mode
609A _{hex}	00 _{hex}	P1203	TX / RX	rw	Homing Mode
60C0 _{hex}	00 _{hex}			ro	Interpolated Mode
60C1 _{hex}	01 _{hex}	P0369	TX / RX	rw	Interpolated Mode
60C2 _{hex}	01 _{hex}			ro	Interpolated Mode
60C2 _{hex}	02 _{hex}	P0532	TX / RX	rw	Interpolated Mode
60C3 _{hex}	01 _{hex}	P3331	TX / RX	rw	Interpolated Mode
60C3 _{hex}	02 _{hex}	P3331	TX / RX	rw	Interpolated Mode
60C4 _{hex}	01 _{hex}	P3331	TX / RX	rw	Interpolated Mode
60C4 _{hex}	02 _{hex}	P3331	TX / RX	rw	Interpolated Mode
60C4 _{hex}	03 _{hex}	P3331	TX / RX	rw	Interpolated Mode
60C4 _{hex}	04 _{hex}	P3331	TX / RX	rw	Interpolated Mode
60C4 _{hex}	05 _{hex}	P3331	RX	wo	Interpolated Mode
60C4 _{hex}	06 _{hex}	P3331	RX	wo	Interpolated Mode
60F8 _{hex}	00 _{hex}	P1054	TX / RX	rw	Profile Velocity Mode
60FB _{hex}	01 _{hex}	P0360	TX	ro	Position Control Function
60FB _{hex}	02 _{hex}	P1050	TX	rw	Position Control Function
60FB _{hex}	03 _{hex}	P1051	TX	rw	Position Control Function
60FB _{hex}	04 _{hex}	P0364	TX / RX	rw	Position Control Function
60FB _{hex}	05 _{hex}	P0363	TX / RX	rw	Position Control Function
60FB _{hex}	06 _{hex}	P1053	TX	rw	Position Control Function
60FB _{hex}	07 _{hex}	P0367	TX	ro	Position Control Function
60FB _{hex}	08 _{hex}	P0362	TX / RX	rw	Position Control Function
60FB _{hex}	09 _{hex}	P0392	TX	ro	Position Control Function
60FB _{hex}	0A _{hex}	P0391	TX	ro	Position Control Function
60FB _{hex}	0B _{hex}	P0365	TX	ro	Position Control Function
60FB _{hex}	0C _{hex}	P0460	TX	ro	Position Control Function

CANopen-Objekt-Nummer		Parameternummer	PDO-Mapping	Access - Typ	Betriebsart nach DSP 402
Index	Subindex				
60FB _{hex}	0D _{hex}	P1191	TX / RX	rw	Position Control Function
60FB _{hex}	0E _{hex}	P1190	TX	rw	Position Control Function
60FB _{hex}	0F _{hex}	P1200	TX	rw	Position Control Function
60FB _{hex}	10 _{hex}	P1208	TX	rw	Position Control Function
60FB _{hex}	11 _{hex}	P0464	TX	ro	Position Control Function
60FB _{hex}	12 _{hex}	P0605	TX / RX	rw	Position Control Function
60FB _{hex}	13 _{hex}	P1198	TX / RX	rw	Position Control Function
60FB _{hex}	14 _{hex}	P1199	TX / RX	rw	Position Control Function
60FB _{hex}	15 _{hex}	P0601	TX / RX	rw	Position Control Function
60FB _{hex}	16 _{hex}	P0608	TX / RX	rw	Position Control Function
60FB _{hex}	17 _{hex}	P0370	TX / RX	rw	Position Control Function
60FB _{hex}	18 _{hex}	P1209	TX / RX	rw	Position Control Function
60FB _{hex}	19 _{hex}	P1204	TX / RX	rw	Position Control Function
60FB _{hex}	1A _{hex}	P0353	TX	ro	Position Control Function
60FB _{hex}	1B _{hex}	P0262 P0263	TX	ro	Position Control Function
60FD _{hex}	00 _{hex}	P0461	TX	ro	Common entries
60FF _{hex}	00 _{hex}	P1171	TX / RX	rw	Profile Velocity Mode
6510 _{hex}	01 _{hex}	P0001	TX	ro	Info
6510 _{hex}	02 _{hex}	P0002	TX	ro	Info
6510 _{hex}	03 _{hex}	P0003	TX	ro	Info
6510 _{hex}	04 _{hex}	P0004	TX	ro	Info
6510 _{hex}	05 _{hex}	P0005	TX	ro	Info
6510 _{hex}	06 _{hex}	P0009	TX	ro	Info
6510 _{hex}	07 _{hex}	P0555	TX	ro	Info
6510 _{hex}	08 _{hex}	P0556	TX	ro	Info



ANHANG C - UMSETZUNGS- TABELLEN

Dieses Kapitel beinhaltet die Tabellen, welche die Umsetzung der CANopen-Kommunikationsobjekte in b maXX[®]-Regler-Kommunikationsparameter und umgekehrt spezifizieren. Die Umsetzung erfolgt unter Angabe der Wertebereiche ($x = x_{\min} \dots x_{\max}$) und der Abbildungsfunktion $x = f(x)$ (im einfachsten Fall wird der Wert nur durchgereicht: $y = x$).

Die Tabellen enthalten folgende Einträge:

CANopen-Objekt:	Bezeichnung des CANopen-Objektes aus DS402
Index ▶ P-Nr.:	Abbildung der CANopen-Objektindizes auf b maXX [®] -Regler-Parameter
Regler-Parameter:	Bezeichnung des Regler-Parameters
P-Nr. ▶ Index:	Umsetzung der b maXX [®] -Regler-Parameter auf CANopen-Objektindizes

CANopen-Objekt	Index Wertebereich	P-Nr. Normierung	Regler-Parameter	P-Nr. Wertebereich	Index) Rück-Normierung	Kommentar
abort_connection_option_code	6007 _{hex}	▶ P0300		P0300	▶ 6007 _{hex}	Der Parameter 6007 _{hex} wirkt im b maXX® auf das Steuerwort, wobei lokale Variablen benutzt werden Bemerkung: Quick Stop Reaktionen über FBO 605A _{hex} einstellbar. Für den Modus Malfunction wird über schreiben auf P3145 ein Fehler ausgelöst, dessen Reaktion über FBO 605E _{hex} eingestellt werden kann. Nur bei Änderungen in „Optionsmodul G/H-Konfiguration 1“ Bit 3 = 1, Defaultwert wird dann 3 statt 0
	x = -32768 .. 32767	▶ y = x		x = 0 .. FFFF _{hex}	▶ y = x	
No action	x = 0		unbenutzt	x = 0	▶ y = x	
Malfunction	x = 1		Reaktion einstellbar	x = 1	▶ y = x	
Device control command "disable_voltage"	x = 2	▶ y = x	Spannung sperren	x = 2	▶ y = x	
Device control command "quick_stop"	x = 3	▶ y = 4	Schnellhalt	x = 3	▶ y = 4	
reserved	x = 4 .. 32767	▶ y = x	unbenutzt	x = 4 .. 32767	▶ y = x	
Manufacturer specific	x = -32768 .. -1					
Controlword	6040 _{hex}	▶ P0300	Steuerwort	P0300	▶ 6040 _{hex}	Bit 6 im Steuerwort: Bit 6 = 0 : Positionier- Modus „absolut“ Bit 6 = 1 : Positionier- Modus „relativ, negativ positiv“ Es wird über CANopen und dem Steuerwort keine anderer Positionsmodus unterstützt. In der BA = 7 (IP-Mode) wird Bit 4 immer auf 1 gesetzt.
	x = 0 .. FFFF _{hex}	▶ y = x		x = 0 .. FFFF _{hex}	▶ y = x	
Switch On	Bit 0	▶ unverändert	Einschalten	Bit 0	▶ unverändert	
Disable Voltage	Bit 1	▶ unverändert	Spannung sperren	Bit 1	▶ unverändert	
Quick Stop	Bit 2	▶ unverändert	Schnellhalt	Bit 2	▶ unverändert	
Enable Op.	Bit 3	▶ unverändert	Betrieb freigeben	Bit 3	▶ unverändert	
Operation mode specific	Bit 4	▶ unverändert	betriebsartabhängig	Bit 4	▶ unverändert	
Operation mode specific	Bit 5	▶ unverändert	betriebsartabhängig	Bit 5	▶ unverändert	
Operation mode specific	Bit 6	▶ unverändert	betriebsartabhängig	Bit 6	▶ unverändert	
Reset Fault	Bit 7	▶ unverändert	Reset Störung	Bit 7	▶ unverändert	
Operation mode specific	Bit 8	▶ unverändert	betriebsartabhängig	Bit 8	▶ unverändert	
reserved	Bit 9	▶ unverändert	Reserve (immer 0)	Bit 9	▶ unverändert	
reserved	Bit 10	▶ unverändert	Reserve (immer 0)	Bit 10	▶ unverändert	
Manufacturer specific	Bit 11	▶ unverändert	betriebsartabhängig	Bit 11	▶ unverändert	
Manufacturer specific	Bit 12	▶ unverändert	betriebsartabhängig	Bit 12	▶ unverändert	
Manufacturer specific	Bit 13	▶ unverändert	betriebsartabhängig	Bit 13	▶ unverändert	
Manufacturer specific	Bit 14	▶ unverändert	betriebsartabhängig	Bit 14	▶ unverändert	
Manufacturer specific	Bit 15	▶ unverändert	Schreibschutz	Bit 15	▶ unverändert	

CANopen-Objekt	Index Wertebereich	P-Nr. Normierung	Regler-Parameter	P-Nr. Wertebereich	Index) Rück-Normierung	Kommentar	
Statusword	6041 _{hex} /ro		Statuswort	P0301	6041 _{hex}	In der BA = 7: IP-Mode Wenn das Steuerwort Bit 4 gesetzt wird, ist Bit 12 gesetzt.	
	x = 0 .. FFFF _{hex}			x = 0 .. FFFF _{hex}	y = x		
Ready To Switch On			Einschaltbereit	Bit 0	▶ unverändert		
Switched On			Eingeschaltet	Bit 1	▶ unverändert		
Operation Enabled			Betrieb freigegeben	Bit 2	▶ unverändert		
Fault			Störung	Bit 3	▶ unverändert		
Voltage Disabled			Spannung gesperrt	Bit 4	▶ unverändert		
Quick Stop			Schnell-Halt	Bit 5	▶ unverändert		
Switched On Enabled			Einschaltsperr	Bit 6	▶ unverändert		
Warning			Warning	Bit 7	▶ unverändert		In ProDrive/WinBASS über Antriebsmanager einstellbar
Man. specific			betriebsartabhängig	Bit 8	▶ unverändert		
Remote			Remote	Bit 9	▶ unverändert		
Targed Reached			Sollwert erreicht	Bit 10	▶ unverändert		
Internal Limit Active			betriebsartabhängig	Bit 11	▶ unverändert		In ProDrive/WinBASS über Antriebsmanager einstellbar
Operation mode specific			betriebsartabhängig	Bit 12	▶ unverändert		
Operation mode specific			betriebsartabhängig	Bit 13	▶ unverändert		
Manufacturer .specific			konf. Statusbits	Bit 14	▶ unverändert		
Manufacturer specific			konf. Statusbits	Bit 15	▶ unverändert		
vl_target_velocity	6042 _{hex}	▶ P1171	RFG1Input	P1171	6042 _{hex}	Die benutzerdefinierte Einheit (speed units) wird im b maXX [®] -Regler als U/min interpretiert. Die Skalierung der Getriebeübersetzung ist im FBO 604C _{hex} abgelegt. Nur bei Änderungen in „Optionsmodul G/H-Konfiguration 1“ Bit 2 = 1: Angabe der gewünschten Drehzahl in 1/10 U/min. z.B.: 200,0 Umdrehungen/min ⇒ Eingabe 2000.	
	x = -32768 .. 32767	▶ y = x * 4000 _{hex} / MotorMax-Speed		x = -32768 .. 32767	y = x * MotorMaxSpeed / 4000 _{hex}		



CANopen-Objekt	Index Wertebereich	P-Nr. Normierung	Regler-Parameter	P-Nr. Wertebereich	Index Rück-Normierung	Kommentar
vl_velocity_demand	6043 _{hex} /ro		RFG Output	P0351	6043 _{hex}	Die benutzerdefinierte Einheit (speed units) wird im b maXX [®] -Regler als U/min interpretiert. Die Skalierung der Getriebeübersetzung ist im FBO 604C _{hex} abgelegt. Nur bei Änderungen in „Optionsmodul G/H-Konfiguration 1“ Bit 2 = 1: Interner Sollwert für die Drehzahl in 1/10 U/min, gleiche Einheit wie Objekt 6042 _{hex} . z.B.: 200,0 Umdrehungen /min ⇒ Eingabe 2000.
				x = 8000 _{hex} .. 7FFF _{hex}	y = x*MotorMaxSpeed / 4000 _{hex}	
vl_control_effort	6044 _{hex} /ro		SpeedActValue	P0353	6044 _{hex}	Die benutzerdefinierte Einheit (speed units) wird im b maXX [®] -Regler als U/min interpretiert. Die Skalierung der Getriebeübersetzung ist im FBO 604C _{hex} abgelegt. Nur bei Änderungen in Optionsmodul G/H-Konfiguration 1 Bit 2 = 1: Interner Sollwert für die Drehzahl in 1/10 U/min, gleiche Einheit wie Objekt 6042 _{hex} . z.B.: 200,0 Umdrehungen /min ⇒ Eingabe 2000.
				x = 8000 _{hex} .. 7FFF _{hex}	y = x*MotorMaxSpeed / 4000 _{hex}	
vl_control_effort	6045 _{hex} /ro		SpeedActValue	P0352	6045 _{hex}	Die benutzerdefinierte Einheit (speed units) wird im b maXX [®] -Regler als U/min interpretiert. Die Skalierung der Getriebeübersetzung ist im FBO 604C _{hex} abgelegt. Nur bei Änderungen in Optionsmodul G/H-Konfiguration 1 Bit2 = 1: Interner Sollwert für die Drehzahl in 1/10 U/min, gleiche Einheit wie Objekt 6042 _{hex} . z.B.: 200,0 Umdrehungen /min ⇒ Eingabe 2000.
				x = 8000 _{hex} .. 7FFF _{hex}	y = x*MotorMaxSpeed / 4000 _{hex}	
vl_velocity_min_max_amount	6046 _{hex} /ro					
vl_velocity_min_amount	Sub. 01 _{hex}	„keine“	SpeedSet_Ulim	„kein“	Sub. 01 _{hex}	Sub. 1 ist immer Null, die min. Grenze ist auf Null festgelegt
				x = 0	y = x	
vl_velocity_max_amount	Sub. 02 _{hex}	▶ P1042 / P1041	SpeedSet_Llim	P1042 / P1041	Sub. 02 _{hex}	Die max. Grenze wirkt sich im b maXX [®] auf beide Dreh-Richtungen symmetrisch aus. Die benutzerdefinierte Einheit (speed units) wird im b maXX [®] -Regler als U/min interpretiert
	x = 0 .. FFFFFFFF _{hex}	▶ y = x* 40000000 _{hex} / MotorMaxSpeed	P1041: x = 0 .. 40000000 _{hex} P1042: x = C0000000 _{hex} .. 0		y = x*MotorMaxSpeed / 40000000 _{hex}	



CANopen-Objekt	Index Wertebereich	P-Nr. Normierung	Regler-Parameter	P-Nr. Wertebereich	Index) Rück-Normierung	Kommentar
vl_velocity_acceleration	6048 _{hex}	<p>a = dv/dt; Da im Regler für die Beschleunigung kein Parameter vorhanden ist, wird durch variieren der Hochlaufzeit P1172 die gewünschte Beschleunigung erreicht. Sie wird auf die maximale Drehzahl des Reglers normiert. Die Berechnung für die gewünschte Beschleunigung erfolgt erst nach Eingabe von dv in SIX1 und dann dt in SIX2. Erst dann wird die korrekt berechnete Zeit auf die Hochlaufzeit geschrieben. Die Rekonstruktion der eingestellten Beschleunigung ist nach einem Neubooten nicht mehr möglich!</p>				
vl_delta_speed	Sub. 01 _{hex}		Applikationsparameter 16		Sub. 01 _{hex}	Die Skalierung der Getriebeübersetzung ist im FBO 604C _{hex} abgelegt. Nur bei Änderungen in Optionsmodul G/H-Konfiguration 1 Bit 2 = 1: Interner Sollwert für die Drehzahl in 1/10 U/min, gleiche Einheit wie Objekt 6042 _{hex} . z.B.: 200,0 Umdrehungen /min ⇒ Eingabe 2000.
	x = 0 .. FFFFFFFF _{hex}	y = Δv		x = 8000 .. 7FFF _{hex}	y = Δv	
vl_delta_time	Sub. 02 _{hex}	P1172	RFG1RampUpTime	P1172	Sub. 02 _{hex}	delta_time wird in Sekunden angegeben; entspricht der Hochlaufgeber-Hochlaufzeit
	x = 0 .. FFFFFFFF _{hex}	y = Δt*Motor-MaxSpeed / Δv*100		x = 0 .. 65000	y = Δt	
vl_velocity_deceleration	6049 _{hex}	<p>a = dv/dt; Da im Regler für die Verzögerung kein Parameter vorhanden ist, wird durch variieren der Rücklaufzeit P1173 die gewünschte Verzögerung erreicht. Sie wird auf die maximale Drehzahl des Reglers normiert. Die Berechnung für die gewünschte Verzögerung erfolgt erst nach Eingabe von dv in SIX1 und dann dt in SIX2. Erst dann wird die korrekt berechnete Zeit auf die Rücklaufzeit geschrieben. Die Rekonstruktion der eingestellten Beschleunigung ist nach einem Neubooten nicht mehr möglich!</p>				
vl_delta_speed	Sub. 01 _{hex}		Applikationsparameter 17		Sub. 01 _{hex}	Die Skalierung der Getriebeübersetzung ist im FBO 604C _{hex} abgelegt. Nur bei Änderungen in Optionsmodul G/H-Konfiguration 1 Bit 2 = 1: Interner Sollwert für die Drehzahl in 1/10 U/min, gleiche Einheit wie Objekt 6042 _{hex} . z.B.: 200,0 Umdrehungen /min ⇒ Eingabe 2000.
	x = 0 .. FFFFFFFF _{hex}	y = Δv		x = 8000 .. 7FFF _{hex}	y = Δv	
vl_delta_time	Sub. 02 _{hex}	P1173	RFG1RampDownTime	P1173	Sub. 02 _{hex}	delta_time wird in Sekunden angegeben; entspricht der Hochlaufgeber-Rücklaufzeit
	x = 0 .. FFFFFFFF _{hex}	y = Δt*Motor-MaxSpeed / - Δv*100		x = 0 .. 65000	y = Δt	



CANopen-Objekt	Index Wertebereich	P-Nr. Normierung	Regler-Parameter	P-Nr. Wertebereich	Index) Rück-Normierung	Kommentar
vl_dimension_factor	604C _{hex}				604C _{hex}	Die Umrechnung im Regler sieht z. B. folgendermaßen aus: Sollwert Drehzahl Motor im b maXX®: Für vl_dimension_factor_numerator = 10 und vl_dimension_factor_denominator = 5 Sollwert Drehzahl Motor = FBO [U/min]*vl_dimension_factor = 100*10 / 5 [U/min] = 200 [U/min]
vl_dimension_factor_numerator	Sub. 01 _{hex} X=-2 ³¹ .. 2 ³¹ -1	y=x		x=-33000 .. 33000	Sub. 01 _{hex} y=x	
vl_dimension_factor_denominator	Sub. 02 _{hex} X=-2 ³¹ .. 2 ³¹ -1	y=x		x=-33000 .. 33000	Sub. 02 _{hex} y=x	
vl_pole_number	604D _{hex} x = 0 .. 255	P0065 y = x / 2	PolPaarzahl	P0065 x = 1..120	604D _{hex} y = x*2	
vl_ramp_function_time	604F _{hex} x = 0 .. FFFFFFFF _{hex}	P1172 y = x	RFG1RampUpTime	P1172 x = 0 .. 65000	604F _{hex} y = x	Hochlaufgeber Hochlaufzeit (1 = 1/1000 s -> 1s = 1000). Die Auflösung beträgt 10 ms
vl_slow_down_time	6050 _{hex} x = 0 .. FFFFFFFF _{hex}	P1173 y = x	RFG1RampDownTime	P1173 x = 0..65000	6050 _{hex} y = x	Hochlaufgeber Hochlaufzeit (1 = 1/1000 s, 1s = 1000). Die Auflösung beträgt 10 ms
vl_quick_stop_time	6051 _{hex} x = 0 .. FFFFFFFF _{hex}	P1174 y = x	RFG1StopTime	P1174 x = 0..65000	6051 _{hex} y = x	Hochlaufgeber Hochlaufzeit (1 = 1/1000 s, 1s = 1000). Die Auflösung beträgt 10 ms

CANopen-Objekt	Index Wertebereich	P-Nr. Normierung	Regler-Parameter	P-Nr. Wertebereich	Index) Rück-Normierung	Kommentar
quick_stop_option_code	605A _{hex}	▶ P1004	QuickstopCode (Schnellhalt)	P1004	▶ 605A _{hex}	
Umrechnungsformalismus	x = -32768 .. 32767	▶ y = x		x = 0 .. 3	▶ y = x	
Manufacturer specific	x = -32768 .. -1	▶ y = x	unbenutzt	x = -32768 .. -1		
Disable Drive	x = 0	▶ y = x	Antrieb gesperrt	x = 0	▶ y = x	
Slow Down On Slow Down Ramp	x = 1	▶ y = x	Rücklauf auf Rücklauframpe	x = 1	▶ y = x	
Slow Down On Quick Stop Ramp	x = 2	▶ y = x	Rücklauf auf Schnellhaltrampe	x = 2	▶ y = x	
Slow Down On Current Ramp	x = 3	▶ y = x	Rücklauf an Stromgrenze	x = 3	▶ y = x	
Slow Down On Voltage Limit	x = 4	▶ y = x	Rücklauf an Spannungsgrenze		y = 4	
Slow Down On Slow Down Ramp and stay in Quick-Stop	x = 5	▶ y = x	Rücklauf auf Rücklauframpe und im Quickstop bleiben		y = 5	
Slow Down On Quick Stop Ramp and stay in Quick-Stop	x = 6	▶ y = x	Rücklauf auf Schnellhaltrampe und im Quickstop bleiben		y = 6	
Slow Down On Current and stay in Quick-Stop	x = 7	▶ y = x	Rücklauf an Stromgrenze und im Quickstop bleiben		y = 7	
Slow Down On Voltage Limit and stay in Quick-Stop	x = 8	▶ y = x	Rücklauf an Spannungsgrenze und im Quickstop bleiben		y = 8	
reserved	x = 9 .. 32767		unbenutzt		y = 9 .. 32767	
shutdown_option_code	605B _{hex}	▶ P1005	ShutDownCode (Stillsetzen)	P1005	▶ 605B _{hex}	
Manufacturer specific	x = -32768 .. -3	▶ y = x	unbenutzt	x = -32768..-3		
Manufacturer specific	x = -2	▶ y = 3	Rücklauf an Stromgrenze	x = 3	▶ y = -2	
Manufacturer specific	x = -1	▶ y = 2	Rücklauf an Schnellhaltrampe	x = 2	▶ y = -1	
Disable Drive	x = 0	▶ y = x	Antrieb gesperrt	x = 0	▶ y = x	
Slow Down On Slow Down Ramp	x = 1	▶ y = x	Rücklauf auf Rücklauframpe	x = 1	▶ y = x	Des ausgewählten HLG einstellbar über P1174 HLG Haltezeit oder in 6051 _{hex}
Reserved	x = 2 .. 32767		unbenutzt		y = .. 32767	



CANopen-Objekt	Index Wertebereich	P-Nr. Normierung	Regler-Parameter	P-Nr. Wertebereich	Index) Rück-Normierung	Kommentar
disable_operation_option_code	605C _{hex}	▶ P1006	DisableOpCode (Sperrern)	P1006	▶ 605C _{hex}	
Manufacturer specific	x = -32768 .. -3	▶ y = x	unbenutzt	x = -32768..-3		
Manufacturer specific	x = -2	▶ y = 3	unbenutzt		y = -2	
Manufacturer specific	x = -1	▶ y = 2	unbenutzt		y = -1	
Disable Drive	x = 0	▶ y = x	Antrieb gesperrt		▶ y = 0	
Slow Down ...	x = 1	▶ y = x	Rücklauf auf Rücklauframpe	x = 1	▶ y = x	
reserved	x = 2		Rücklauf auf Schnellhalt-rampe	x = 2	▶ y = -1	
reserved	x = 3		Rücklauf an Stromgrenze	x = 3	▶ y = -2	
reserved	x = 4 .. 32767		unbenutzt		y = 4 .. 32767	
stop_option_code	605D _{hex}	▶ P1003	StopOptionCode (Halt)	P1003	▶ 605D _{hex}	
Umrechnungsformalismus	x = -32768 .. 32767	▶ y = x		x = 0 .. 3	▶ y = x	
Manufacturer specific	x = -32768 .. -1	▶ y = x	unbenutzt		y = -32768 .. -1	
Disable Drive	x = 0	▶ y = x	Antrieb gesperrt	x = 0	▶ y = x	
Slow Down On Slow Down Ramp	x = 1	▶ y = x	Rücklauf auf Rücklauframpe	x = 1	▶ y = x	Des ausgewählten HLG einstellbar über P1174 HLG Haltezeit oder in 6051 _{hex}
Slow Down On Quick Stop Ramp	x = 2	▶ y = x	Rücklauf auf Schnellhalt-rampe	x = 2	▶ y = x	
Slow Down On Current Ramp	x = 3	▶ y = x	Rücklauf an Stromgrenze	x = 3	▶ y = x	
Slow Down On Voltage Limit	x = 4	▶ y = x	unbenutzt		y = 4	
reserved	x = 5 .. 32767		unbenutzt		y = 5 .. 32767	

CANopen-Objekt	Index Wertebereich	P-Nr. Normierung	Regler-Parameter	P-Nr. Wertebereich	Index) Rück-Normierung	Kommentar
fault_reaction_option_code	605E _{hex}	▶ P1007	ErrorReactionCode	P1007	▶ 605E _{hex}	z.Z. nur für statische und dynamische Schleppefehler und für die Reaktionen für das FBO 6007 _{hex} „Modus 1 Malfunction“ einstellbar
Umrechnungsformalismus	x = -32768 .. 32767	▶ y = x		x = 0 .. 3	▶ y = x	
Manufacturer specific	x = -32768 .. -1	▶ y = x	unbenutzt	x = -32768 .. -1		
Disable Drive, motor is free to rotate	x = 0	▶ y = x	Antrieb gesperrt	x = 0	▶ y = x	
Slow down on slow down ramp	x = 1	▶ y = x	Rücklauf auf Rücklauframpe	x = 1	▶ y = x	
Slow down on quick stop ramp	x = 2	▶ y = x	Rücklauf auf Schnellhalt-rampe	x = 2	▶ y = x	
Slow down on current ramp	x = 3	▶ y = x	Rücklauf an Stromgrenze	x = 3	▶ y = x	
Slow down on voltage limit	x = 4	▶ y = x	Rücklauf an Spannungsgrenze		y = 4	
reserved	x = 5 .. 32767		unbenutzt		y = 5 .. 32767	



CANopen-Objekt	Index Wertebereich	P-Nr. Normierung	Regler-Parameter	P-Nr. Wertebereich	Index) Rück-Normierung	Kommentar
modes_of_operation	6060 _{hex} /rw	▶ P1000	OperationModeSet	P1000	6060	<p>y', x' ≙ nur intern im Regler</p> <p>In Abhängigkeit der eingestellten BA: BA = -4 Lageregelung BA = 7 IP-Mode (intern wird im Regler auf BA = -4 geschaltet)</p>
Umrechnungsformalismus	x = -128 .. 127	▶ y = x		x = -128 .. 127	y = x	
Manufacturer specific	x = -106	▶ y' = -6				
Manufacturer specific	x = -7	▶ y = x	Selbstoptimierung	x = -7	y = x	
Manufacturer specific	x = -6	▶ y = 5	Spindelpositionierung	x = -6	y = -106	
Manufacturer specific	x = -5	▶ y = x	Gleichlauf el. Getriebe	x = -5	y = x	
Manufacturer specific	x = -4	▶ y = x	Lageregelung oder Interpolated Position Mode	x = -4	y = -4 oder y = 7	
Manufacturer specific	x = -3	▶ y = x	Drehzahlregelung	x = -3	y = 3	
Manufacturer specific	x = -2	▶ y = x	Stromregelung	x = -2	y = x	
Manufacturer specific	x = -1	▶ y = x	Rastlage	x = -1	y = x	
reserved	x = 0		unbenutzt	x = 0		
Profile Position Mode	x = 1	▶ y = x	Lagezielvorgabe	x = 1	y = x	
Velocity Mode	x = 2	▶ y = x	Geschwindigkeitsvorgabe	x = 2	y = x	
Profile Velocity Mode	x = 3	▶ y = -3	unbenutzt	x = 3	y = -3	
Torque Profile Mode	x = 4	▶ y = x	unbenutzt	x = 4		
reserved	x = 5		Handfahrbetrieb	x = 5	y = -6	
Homing Mode	x = 6	▶ y = x	Referenzbetrieb	x = 6	y = x	
Interpolated Position Mode	x = 7	▶ y = -4	unbenutzt	x = 7		
reserved	x = 8 .. 127		unbenutzt	x = 8 .. 127		

CANopen-Objekt	Index Wertebereich	P-Nr. Normierung	Regler-Parameter	P-Nr. Wertebereich	Index) Rück-Normierung	Kommentar
modes_of_operation_display	6061 _{hex} /ro		OperationModeAct (Ist-Betriebsart)	P0304	6061 _{hex}	Die CANopen Standardbezeichnung, siehe 6060 _{hex} - Handfahrbetrieb ist im Regler der Wert 5, in CANopen wird er auf den Wert -6 gelegt - Spindelpositionierung ist im Regler der Wert -6, in CANopen wird er auf den Wert -106 gelegt
Umrechnungsformalismus				x = -128 .. 127	y = x	
			Selbstoptimierung	x = -7	y = x	
			Spindelpositionierung	x = -6	y = -106	
			Gleichlauf el. Getriebe	x = -5	y = x	
			Lageregelung	x = -4	y = x	
			Drehzahlregelung	x = -3	y = 3	
			Stromregelung	x = -2	y = x	
			Rastlage	x = -1	y = x	
			unbenutzt	x = 0		
			Lagezielvorgabe	x = 1	y = x	
			Geschwindigkeitsvorgabe 1	x = 2	y = x	
			unbenutzt	x = 3		
			unbenutzt	x = 4		
			Handfahrbetrieb	x = 5	y = -6	
			Referenzbetrieb	x = 6	y = x	
			unbenutzt	x = 7		
			unbenutzt	x = 8 .. 127		
position_demand_value	6062 _{hex} /ro		PPosSetValue (Lage-Istwert)	P0463	6062 _{hex}	USIGN32 wird auf der CANopen Optionskarte mit einem Offset von 2 ³¹ versehen. USIGN32 -> INT32. (Offset von 2 ³¹ wird subtrahiert. BE - Übersetzung hinzugefügt. Nur bei Änderungen in Optionsmodul G/H-Konfiguration 1 Bit 2 = 1, Angabe der aktuellen Position des Antriebes Benutzereinheiten z.B. 1/100° für Drehbewegungen ausgehend vom Referenzpunkt.
				x = 80000000 .. 7FFFFFFF _{hex}	y = x-2 ³¹	
position_actual_value*	6063 _{hex} /ro		PPosActValue (Lage-Istwert)	P0362	6063 _{hex}	BE - Übersetzung hinzugefügt. Nur bei Änderungen in Optionsmodul G/H-Konfiguration 1 Bit 2 = 1: Angabe der aktuellen Position des Antriebes Benutzereinheiten z.B. 1/100° für Drehbewegungen ausgehend vom Referenzpunkt.
				x = 80000000 .. 7FFFFFFF _{hex}	y = x	



CANopen-Objekt	Index Wertebereich	P-Nr. Normierung	Regler-Parameter	P-Nr. Wertebereich	Index) Rück-Normierung	Kommentar
position_actual_value	6064 _{hex} /ro		PPosActValue (Lage-Istwert)	P0462	6064 _{hex}	USIGN32 wird auf der CANopen Optionskarte mit einem Offset von 2 ³¹ versehen. USIGN32 ⇒ INT32. (Offset von 2 ³¹ wird subtrahiert. BE - Übersetzung hinzugefügt. Nur bei Änderungen in Optionsmodul G/H-Konfiguration 1 Bit 2 = 1: Angabe der Aktuellen Position des Antriebes Benutzereinheiten z.B. 1/100° für Drehbewegungen ausgehend vom Referenzpunkt.
				x = 80000000 .. 7FFFFFFF _{hex}	y = x-2 ³¹	
following_error_time_out	6066 _{hex} /ro		PosDevTime	P1056	6066 _{hex}	Die Einheit beträgt im CANopen-Objekt und im b maXX [®] -Regler-Parameter in ms.
				x = 0 .. 65000	y = x	
position_window	6067 _{hex}	▶ P1194	PPosWindow (Pos.-Fenster)	P1194	6067 _{hex}	
	x = 0 .. FFFFFFFF _{hex}	▶ y = x		x = 0 .. FFFFFFFF _{hex}	y = x	
position_window_time	6068 _{hex}	▶ P1195	PPosWindow Time (Pos.-Fensterzeit)	P1195	6068 _{hex}	
	x = 0 .. 65535	▶ y = x		x = 1 .. FFFF _{hex}	y = x	
velocity_sensor_actual_value	6069 _{hex}	▶ P0362	Enc1ActAngle	P0391	6069 _{hex}	
	x = -2 ¹⁵ .. 2 ¹⁵ -1	▶ y = x		x = 0 .. FFFFFFFF _{hex}	y = x	
sensor_selection_code	606A _{hex} /ro		„kein“			Der b maXX [®] -Regler unterstützt nur Positions-Encoder, daher nur Anzeige.
velocity_actual_value_from_position_encoder				x = 0	y = x	
velocity_actual_value_from_velocity_encoder			nicht unterstützt			
velocity_demand_value	606B _{hex} /ro		SetValueTotal	P0352	606B _{hex}	Die benutzerdefinierte Einheit (speed units) wird im Regler als U/min interpretiert. Nur bei Änderungen in Optionsmodul G/H-Konfiguration 1 Bit 2 = 1: Angabe der akt. Drehzahl in 1/10 U/min. z.B.: 200,0 Umdrehungen => Eingabe 2000.
				x = 8000 _{hex} .. 7FFF _{hex}	y = x*Motor-MaxSpeed / 4000 _{hex}	

CANopen-Objekt	Index Wertebereich	P-Nr. Normierung	Regler-Parameter	P-Nr. Wertebereich	Index) Rück-Normierung	Kommentar
velocity_actual_value	606C _{hex} /ro		SpeedActValue	P0353 x = -2 ³¹ .. 2 ³¹ -1	606C _{hex} y = x*Motor-MaxSpeed / 40000000 _{hex}	Die Skalierung der Getriebeübersetzung ist im FBO 604C _{hex} abgelegt Nur bei Änderungen in Optionsmodul G/H-Konfiguration 1 Bit 2 = 1: Angabe der aktuellen Drehzahl in 1/10 U/min. z.B.: 200,0 Umdrehungen ⇒ Eingabe 2000.
velocity_threshold	606F _{hex} x = -2 ³¹ .. 2 ³¹ -1	P1073 y = x*4000 _{hex} / 10000	Enc1Mon_Llim	P1073 x = -0 .. 1000 _{hex}	606F _{hex} y = x*10000 / 4000 _{hex}	Die Schwelle kann im b maXX [®] -Regler bis auf 25% der maximalen Drehzahl des Reglers heraufgesetzt werden. Die Eingabe erfolgt dann in U/min z.B. Max: 1000 U/min Eingabe für 25 % = 250 U/min
max_torque	6072 _{hex} x = 0 ... FFFF _{hex}	P0357 y = x*4000 _{hex} / 1000	TrqSynDirect	P0357 x = 0000 .. FFFF _{hex}	6072 _{hex} y = x*1000 / 4000 _{hex}	
Torque actual value	6077 _{hex} /ro x = -2 ¹⁶ ... 2 ¹⁶ - 1	P0344 y = x * 10000 / 1638				1000 entspricht 100,0 % bezogen auf das Nennmoment P1036
target_position	607A _{hex} x = 80000000 _{hex} .. 7FFFFFFF _{hex}	P0607 (P0600) y = x	PPosTarget1	P0607 (P0600) x = 800000000 .. 7FFFFFFF _{hex}	607A _{hex} y = x	BE - Übersetzung hinzugefügt Nur bei Änderungen in „Optionsmodul G/H-Konfiguration 1“ Bit 2 = 1: Benutzereinheiten z.B. 1/100° für Drehbewegungen ausgehend vom Referenzpunkt.
home_offset	607C _{hex} x = -2 ³¹ .. 2 ³¹ -1	P1200 y = x+2 ³¹	PPosEncoderOffset	P1200 x = 0 .. 2 ³²	607C _{hex} y = x-2 ³¹	Abweichung der Homeposition vom Referenz- bzw. Endschalter BE - Übersetzung und ein Offset von 2 ³¹ hinzugefügt (Zahlenskalawandlung). Nur bei Änderungen in „Optionsmodul G/H-Konfiguration 1“ Bit 2 = 1: angegeben in BE z.B. 1/100 ° für Drehbewegungen. Keine Grenzwertüberwachung.
software_position_limit	607D _{hex} Sub. 01 _{hex} x = -2 ³¹ .. 2 ³¹ -1	P1196 y = x	SW-Endschalter PPosSWLimitSwitch1	P1196 x = 0 .. FFFFFFFF _{hex}	607D _{hex} Sub. 01 _{hex} y = x	USIGN32 wird auf der CANopen Optionskarte mit einem Offset von 2 ³¹ versehen. USIGN32 ⇒ INT32. (Offset von 2 ³¹ wird Subtrahiert. BE - Übersetzung hinzugefügt.
	Sub. 02 _{hex} x = -2 ³¹ .. 2 ³¹ -1	P1197 y = x	PPosSWLimitSwitch2	P1197 x = 0 .. FFFFFFFF _{hex}	Sub. 02 _{hex} y = x	



CANopen-Objekt	Index Wertebereich	P-Nr. Normierung	Regler-Parameter	P-Nr. Wertebereich	Index) Rück-Normierung	Kommentar
max_profile_velocity	607F _{hex}	▶ P0057	MotorNomSpeed	P0057	▶ 607F _{hex}	Die benutzerdefinierte Einheit (speed units) wird im Regler als U/min interpretiert
	x = 0 .. 2 ³² -1	▶ y = x		x = 1 ... 24000	▶ y = x	
max_motor_speed	6080 _{hex}	▶ P1031	SpeedMax	P1031	▶ 6080	Die benutzerdefinierte Einheit (speed units) wird im Regler als U/min interpretiert
	x = 0 .. FFFF _{hex}	▶ y = x		x = 20 .. 24000	▶ y = x	
profile velocity	6081 _{hex}	▶ P0602	PPosSetSpeed1	P0602	▶ 6081 _{hex}	Nur bei Änderungen in Optionsmodul G/H-Konfiguration 1 Bit2 = 1: Angabe der gewünschten Fahrgeschwindigkeit des Fahrauftrages in BE z.B. 1/100 °/s für Drehbewegungen. s ⇒ ms [1/1000]. BE hinzugefügt. Bei Grenzwertüberschreitung des Parameters im b maXX [®] werden die Max. bzw. Min.-werte gesetzt ohne eine Fehlermeldung abzusetzen.
	x = 0 .. 2 ³²	▶ y = x		x = 1 .. 13200	▶ y = x	
profile acceleration	6083 _{hex}	▶ P0603	PPosAcceleraton1	P0603	▶ 6083 _{hex}	Nur bei Änderungen in Optionsmodul G/H-Konfiguration 1 Bit2 = 1: Anfahrt-/Bremsbeschleunigung des Fahrauftrages angegeben in BE z.B. 10°/s ² für Drehbewegungen. s ² ⇒ ms ² [1/10000]
	x = 0 .. 2 ³²	▶ y = x		x = 25 .. 45000	▶ y = x	
profile deceleration	6084 _{hex}	▶ P0604	PPosDeceleraton1	P0604	▶ 6084 _{hex}	Bei Grenzwertüberschreitung des Parameters im b maXX [®] werden die Max. bzw. Min.-werte gesetzt ohne eine Fehlermeldung abzusetzen.
	x = 0 .. 2 ³²	▶ y = x		x = 25 .. 45000	▶ y = x	
quick_stop_deceleration	6085 _{hex}	▶ P1213	PPosStopDeceleraton	P1213	▶ 6085 _{hex}	Nur bei Änderungen in Optionsmodul G/H-Konfiguration 1 Bit2 = 1: Anfahrt-/Bremsbeschleunigung des Fahrauftrages angegeben in BE z.B. 10°/s ² für Drehbewegungen. s ² ⇒ ms ² [1/10000]
	x = 0 .. 2 ³²	▶ y = x		x = 25 .. 45000	▶ y = x	
motion profile type	6086 _{hex}	▶ P1190	PPosMode	P1190	▶ 6086 _{hex}	Im Regler
	x = -2 ¹⁶ ... 2 ¹⁶ -1	▶		x = 0.. FFFF _{hex}	▶	
Manufacturer specific	x = -32768..-1		Unbenutzt		▶ y = x	

CANopen-Objekt	Index Wertebereich	P-Nr. Normierung	Regler-Parameter	P-Nr. Wertebereich	Index) Rück-Normierung	Kommentar
Linear ramp (trapezoidal profile)	x = 0	▶ Bit3 und Bit4	Trapez	Bit3 und Bit4	▶ 0	Geschwindigkeitsprofil: Bit 4 Bit 3: 0 0: Trapez 0 1: Sin ² 1 0: S-Kurve 1 1: Reserviert
Sin ² ramp	x = 1	▶ Bit3 und Bit4	Sin ²	Bit3 und Bit4	▶ 1	
Jerk-free ramp	x = 2	▶ Bit3 und Bit4	S-Kurve	Bit3 und Bit4	▶ 2	
Jerk-limited ramp	x = 3		Unbenutzt			
Reserved for future profile types	x = 4..32767	y = x	Unbenutzt			
feed_constant	6092 _{hex} /ro				▶ 6092 _{hex}	
feed	Sub. 01 _{hex}	▶ P3050	PosScalingUserUnit	P3050	▶ Sub. 01 _{hex}	
	x = 0 .. FFFFFFFF _{hex}	▶ y = x		x = 2 ²⁴ .. 1	▶ y = x	
shaft_revolutions	Sub. 02 _{hex}	▶ P3051	PosScalingRevolution	P3051	▶ Sub. 02 _{hex}	
	x = 0 .. FFFFFFFF _{hex}	▶ y = x		x = 1 .. 2 ²⁴ -1	▶ y = x	



CANopen-Objekt	Index Wertebereich	P-Nr. Normierung	Regler-Parameter	P-Nr. Wertebereich	Index) Rück-Normierung	Kommentar
homing_method	6098 _{hex}	▶ P1205	PPosHomingMode (Ref. Fahrmodus)	P1205	▶ 6098 _{hex}	
Manufacturer specific	x = -128 .. -12		unbenutzt		▶ y = -128 .. -7	
Manufacturer specific	x = -10	▶ y = -10	Anfahren mechanischer Anschlag mit Nullimpuls, Linksdrehung	x = -10	▶ y = -10	
Manufacturer specific	x = -9	▶ y = -9	Anfahren mechanischer Anschlag mit Nullimpuls, Rechtsdrehung	x = -9	▶ y = -9	
Manufacturer specific	x = -8	▶ y = -8	Anfahren mechanischer Anschlag, Linksdrehung	x = -8	▶ y = -8	
Manufacturer specific	x = -7	▶ y = -7	Anfahren mechanischer Anschlag, Rechtsdrehung	x = -7	▶ y = -7	
Manufacturer specific	x = -6	▶ y = -6	Anfahren des nächsten Gebernullwinkels	x = -6	▶ y = -6	
Manufacturer specific	x = -5	▶ y = -5	Anfahren des pos. Endschal- ters	x = -5	▶ y = -5	
Manufacturer specific	x = -4	▶ y = -4	Anfahren des neg. Endschal- ters	x = -4	▶ y = -4	
Manufacturer specific	x = -3	▶ y = -3	Referenzpunkt setzen	x = -3	▶ y = -3	
Manufacturer specific	x = -2	▶ y = -2	Anfahren des Gebernullwin- kels bzw. des Nullimpulses mit Linksdrehung	x = -2	▶ y = -2	
Manufacturer specific	x = -1	▶ y = -1	Anfahren des Gebernullwin- kels bzw. des Nullimpulses mit Rechtsdrehung	x = -1	▶ y = -1	
No homing operation	x = 0		unbenutzt		▶ y = 0	
Homing on the neg. Limit Switch	x = 1	▶ y = 1	Anfahren des neg. Endschal- ters mit Gebernullwinkel- bzw. Nullimpulsreferenzierung	x = 1	▶ y = 1	
Homing on the pos.Limit Switch	x = 2	▶ y = 2	Anfahren des pos. Endschal- ters mit Gebernullwinkel- bzw. Nullimpulsreferenzierung	x = 2	▶ y = 2	
Homing on the positive Home Switch & Index Pulse	x = 3	▶ y = 3	Anfahren des pos. Nullpunk- tumschalers mit Gebernull- winkel- bzw. Nullimpulsreferenzierung	x = 3	▶ y = 3	
Homing on the positive Home Switch & Index Pulse	x = 4	▶ y = 4	Anfahren des pos. Nullpunk- tumschalers mit Gebernull- winkel- bzw. Nullimpulsreferenzierung	x = 4	▶ y = 4	

CANopen-Objekt	Index Wertebereich	P-Nr. Normierung	Regler-Parameter	P-Nr. Wertebereich	Index) Rück-Normierung	Kommentar
Homing on the negative Home Switch & Index Pulse	x = 6	y = 6	Anfahren des neg. Nullpunktumschalters mit Gebernulldrehung- bzw. Nullimpulsreferenzierung	x = 6	y = 6	
Zero reference cam switch, left to pos. edge with Zero pulse; CW move	x = 7	y = 7	Nullpunktschalter, links von pos. Flanke mit Nullimpuls; Rechtsdrehung	x = 7	y = 7	
Zero reference cam switch, right fo pos. edge with Zero pulse; CW move	x = 8	y = 8	Nullpunktschalter, rechts von pos. Flanke mit Nullimpuls; Rechtsdrehung	x = 8	y = 8	
Zero reference cam switch, left to neg. edge with Zero pulse; CW move	x = 9	y = 9	Nullpunktschalter, links von neg. Flanke mit Nullimpuls; Rechtsdrehung	x = 9	y = 9	
Zero reference cam switch, right to neg. edge with Zero pulse; CW move	x = 10	y = 10	Nullpunktschalter, rechts von neg. Flanke mit Nullimpuls; Rechtsdrehung	x = 10	y = 10	
Zero reference cam switch, right to neg. edge with Zero pulse; CCW move	x = 11	y = 11	Nullpunktschalter, rechts von neg. Flanke mit Nullimpuls; Linksdrehung	x = 11	y = 11	
Zero reference cam switch, right fo pos. edge with Zero pulse; CCW move	x = 12	y = 12	Nullpunktschalter, rechts von pos. Flanke mit Nullimpuls; Linksdrehung	x = 12	y = 12	
Zero reference cam switch, left to neg. edge with Zero pulse; CCW move	x = 13	y = 13	Nullpunktschalter, links von neg. Flanke mit Nullimpuls; Linksdrehung	x = 13	y = 13	
Zero reference cam switch, right to neg. edge with Zero pulse; CCW move	x = 14	y = 14	Nullpunktschalter, rechts von neg. Flanke mit Nullimpuls; Linksdrehung	x = 14	y = 14	
CANopen Spec.	x = 15, 16		unbenutzt			
Negative Limit Switch	x = 17	y = 17	negativer Endschalters	x = 17	y = 17	
Positive Limit Switch	x = 18	y = 18	positiver Endschalters	x = 18	y = 18	
Positive Zero reference switch, CCW move	x = 19	y = 19	positiver Nullpunktumschalter; Linksdrehung	x = 19	y = 19	
Positive Zero reference switch, CW move	x = 20	y = 20	positiver Nullpunktumschalter; Rechtsdrehung	x = 20	y = 20	
Negative Zero reference switch, CW move	x = 21	y = 21	negativer Nullpunktumschalter; Rechtsdrehung	x = 21	y = 21	



CANopen-Objekt	Index Wertebereich	P-Nr. Normierung	Regler-Parameter	P-Nr. Wertebereich	Index) Rück-Normierung	Kommentar
Negative Zero reference switch, CCW move	x = 22	▶ y = 22	negativer Nullpunktschalter; Linksdrehung	x = 22	▶ y = 22	
Zero reference cam switch, left to pos. edge; CW move	x = 23	▶ y = 23	Nullpunktschalter, links von pos.; Rechtsdrehung	x = 23	▶ y = 23	
Zero reference cam switch, right to pos. edge; CW move	x = 24	▶ y = 24	Nullpunktschalter, rechts von pos. Flanke; Rechtsdrehung	x = 24	▶ y = 24	
Zero reference cam switch, left to neg. edge; CW move	x = 25	▶ y = 25	Nullpunktschalter, links neg. Flanke; Rechtsdrehung	x = 25	▶ y = 25	
Zero reference cam switch, right to neg. edge; CW move	x = 26	▶ y = 26	Nullpunktschalter, rechts von neg. Flanke; Rechtsdrehung	x = 26	▶ y = 26	
Zero reference cam switch, right to neg. edge; CCW move	x = 27	▶ y = 27	Nullpunktschalter, rechts von neg. Flanke; Linksdrehung	x = 27	▶ y = 27	
Zero reference cam switch, left to neg. edge; CCW move	x = 28	▶ y = 28	Nullpunktschalter, links von neg. Flanke; Linksdrehung	x = 28	▶ y = 28	
Zero reference cam switch, right to pos. edge; CCW move	x = 29	▶ y = 29	Nullpunktschalter, rechts von pos. Flanke; Linksdrehung	x = 29	▶ y = 29	
Zero reference cam switch, left to pos. edge; CCW move	x = 30	▶ y = 30	Nullpunktschalter, links von pos. Flanke; Linksdrehung	x = 30	▶ y = 30	
CANopen-Spec.	31..32		unbenutzt	31..32		
Nearest Zero pulse; CCW move	x = 33	▶ y = 33	nächster Nullimpuls; Linksdrehung	x = 33	▶ y = 33	
Nearest Zero pulse; CW move	x = 34	▶ y = 34	nächster Nullimpuls mit Rechtsdrehung	x = 34	▶ y = 34	
Homing on the Current Position	x = 35	▶ y = 35	Referenzpunkt setzen	x = 35	▶ y = 35	
reserved	x = 36 .. 127		unbenutzt			

CANopen-Objekt	Index Wertebereich	P-Nr. Normierung	Regler-Parameter	P-Nr. Wertebereich	Index) Rück-Normierung	Kommentar	
homing_speeds	6099 _{hex}		(Ref. Geschw.)		6099 _{hex}	Nur bei Änderungen in Optionsmodul G/H-Konfiguration 1 Bit 2 = 1: Angabe der gewünschten Fahrgeschwindigkeiten des Fahrauftrages in BE z.B. 1/100 °/s für Drehbewegungen. s ⇒ ms [1/1000]. Bei Grenzwertüberschreitung des Parameters im b maXX [®] werden die Max. bzw. Min.-werte gesetzt ohne eine Fehlermeldung abzusetzen.	
speed_during_search_for_switch	Sub. 01 _{hex} x = 0 .. 2 ³²	▶ P1201 y = x	PPosHomingSpeed	P1201 x = 1..13200	▶ Sub. 01 _{hex} y = x		
speed_during_search_for_zero	Sub. 02 _{hex} x = 0 .. 2 ³²	▶ P1202 y = x	PPosHomingFinalSpeed	P1202 x = 1..50	▶ Sub. 02 _{hex} y = x		
homing_acceleration	609A _{hex} x = 0 .. 2 ³²	▶ P1203 y = x	PPosHomingAcceler (Referenzbeschleunigung)	P1203 x = 25 .. 45000	▶ 609A _{hex} y = x	Nur bei Änderungen in Optionsmodul G/H-Konfiguration 1 Bit 2 = 1: Die Benutzerdefinierte Einheit (acceleration units) von homing_acceleration in BE z.B. 10 °/s für Drehbewegungen. S ² ⇒ ms ² [1/1000]. Bei Grenzwertüberschreitung des Parameters im b maXX [®] werden die Max. bzw. Min.-werte gesetzt ohne eine Fehlermeldung abzusetzen.	
Interpolation_submode_select	60C0 _{hex} /ro x = -2 ¹⁶ ... 2 ¹⁶ - 1 x = 0 x = 1. . 32767	Sub. 0 = 1 -32768 ... -1, ▶ linear Interpolation ▶ reserved	Es wird auf keinen Parameter im Regler geschrieben.			Objekt 60C0 _{hex} enthält die Information zum IP-Mode. Es wird nur die lineare Interpolation unterstützt.	
Interpolation_data_record	60C1 _{hex} /ro						In Sub.0 ist die Größe des Buffers lesbar. Es wird nur Sub.1 unterstützt. Die 16 oberen Bit entsprechen der Anzahl der Umdrehungen und die 16 unteren Bit entsprechen dem Winkel in Inc. Synchronisieren erfolgt über das Sync-Telegramm vom Master
	Sub. 0 x = 0 ... 255	▶ 1					
	Sub. 1 x = 0 ... 2 ³² - 1	▶ P369 y = x		Lage-Sollwert	P369 x = 0 ... 2 ³² - 1		
Interpolation_time_period	60C2 _{hex}	Sub. 0 = 2				Zulässig sind nur Einträge, die 2000, 4000 und 8000 µs entsprechen. Ist keine Zykluszeit auf P532 eingetragen gibt es eine Fehlermeldung.	



CANopen-Objekt	Index Wertebereich	P-Nr. Normierung	Regler-Parameter	P-Nr. Wertebereich	Index) Rück-Normierung	Kommentar
lp_time_units	Sub. 1	▶ kein	Sync-Slot	kein	▶ Sub. 1	10⁴ip_time_index * seconds Beispiel: Wenn 4000 µs eingestellt werden sollen, so muss auf Sub1 der Wert 4 und auf Sub2 der Wert -3 eingestellt werden. Erst beim Schreiben von SIX2 wird die Zykluszeit im Regler übernommen.
	x = 1 ... 255	x = -2 ¹⁶ ... 2 ¹⁶ - 1		x = -2 ¹⁶ ... 2 ¹⁶ - 1	x = 1 ... 255	
lp_time_index	Sub. 2	▶ P532	Sync-Slot	P532	▶ Sub. 2	
	x = -128 ... 63	x = -2 ¹⁶ ... 2 ¹⁶ - 1		x = -2 ¹⁶ ... 2 ¹⁶ - 1	x = 1 ... 255	
Interpolation_sync_definition	60C3 _{hex}	▶ Sub. 0 = 2				Sub.1: der Synchronisation Modus wird festgelegt. Fest -> Wert = 0: Synchronisation bezieht sich auf das SYNC-Telegramm. Sub.2 ist nur lesbar und zeigt die Anzahl an, ab welchem SYNC die Phi-Werte im Buffer incrementiert und übernommen werden sollen. Es wird nur die Übernahme bei jedem empfangenen SYNC-Telegramm unterstützt. D. h. Standard = "0". Es wird ein Dummy Parameter (Applikationsparameter 3331) verwendet.
synchronize on group	Sub. 1	▶ P3331 (Applikationsparameter)	General Sync is used (Standard)	P3331 (Applikationsparameter)	▶ 60C3 _{hex} Sub 1	
lp_sync every n event	Sub. 2 ro x = 0 ... 255	▶ P3331 (Applikationsparameter)				
Interpolation_data_configuration	60C4 _{hex}	▶ Sub. 0 = 6				
max_buffer_size	Sub. 1 ro	▶ 1				
actual_size	Sub. 2	▶ 1	Es wird nur ein Buffer unterstützt.			
buffer_organisation	Sub. 3	▶ 0..1	0 = FIFO-Buffer; 1= Ring-Buffer			
buffer_position	Sub. 4 ro	▶ 1	Da nur ein Buffer vorhanden immer 1			
size_ofdata_record	Sub. 5 ro	▶ 1	Da nur ein Buffer vorhanden immer 1			
buffer_clear	Sub. 6	▶ 0..1	(kein Einfluss)			
position_control_parameter_set	60FB _{hex}					Herstellerspezifisches CANopen-Objekt
	Sub. 01 _{hex} /ro		PosCtrlStatus	P0360 x = 0 .. 2 ¹⁶ -1	▶ Sub. 01 _{hex} ▶ y = x	Default = 0
	Sub. 02 _{hex} x = 0 .. 2 ¹⁶ -1	▶ P1050 ▶ y = x	PosCtrlMode	P1050 x = 0 .. 2 ¹⁶ -1	▶ Sub. 02 _{hex} ▶ y = x	Default = 0



CANopen-Objekt	Index Wertebereich	P-Nr. Normierung	Regler-Parameter	P-Nr. Wertebereich	Index) Rück-Normierung	Kommentar
	Sub. 03 _{hex} x = 0 .. 32767	▶ P1051 y = 0 .. 32767	PosCtrl_Kv-Faktor	P1051 x = 0 .. 32767	▶ Sub. 03 _{hex} y = x	Default = 10,0
	Sub. 04 _{hex} x = 0 .. 2 ¹⁶ -1	▶ P0364 y = x	PosSetRev	P0364 x = 0 .. 2 ¹⁶ -1	▶ Sub. 04 _{hex} y = x	Default = 0
	Sub. 05 _{hex} x = 0 .. 2 ¹⁶ -1	▶ P0363 y = x	PosSetAngle	P0363 x = 0 .. 2 ¹⁶ -1	▶ Sub. 05 _{hex} y = x	Default = 0
	Sub. 06 _{hex} x = 0 .. 2 ¹⁶ -1	▶ P1053 y = 0 .. 5000 _{hex}	SpeedFeedForFactor	P1053 x = 0 .. 5000 _{hex}	▶ Sub. 06 _{hex} y = 0 .. 2 ¹⁶ -1	Default = 4000 _{hex}
	Sub. 07 _{hex} /ro		PosCtrlDev	P0367 x = -2 ³¹ .. 2 ³¹ -1	▶ Sub. 07 _{hex} y = x	Default = 0
	Sub. 08 _{hex}		PosActValue	P0362 x = 0 .. 2 ³² -1	▶ Sub. 08 _{hex} y = x	Default = 0
	Sub. 09 _{hex} /ro		Enc1ActRev	P0392 x = 0 .. 2 ³² -1	▶ Sub. 09 _{hex} y = x	Default = 0
	Sub. 0A _{hex} /ro		Enc1ActAngle	P0391 x = 0 .. 2 ³²	▶ Sub. 0A _{hex} y = x	
	Sub. 0B _{hex} /ro		SpeedFeedFor	P0365 x = -2 ³¹ .. 2 ³¹ -1	▶ Sub. 0B _{hex} y = x	
	Sub. 0C _{hex} /ro		PPosStatus	P0460 x = 0 .. FFFF _{hex}	▶ Sub. 0C _{hex} y = x	
	Sub. 0D _{hex} x = 0 .. 2 ¹⁶ -1	▶ P1191 y = x	PPosActRecordNumber	P1191 x = 0 .. 2 ¹⁶ -1	▶ Sub. 0D _{hex} y = x	
	Sub. 0E _{hex} x = 0 .. 2 ¹⁶ -1	▶ P1190 y = x	PPosMode	P1190 x = 0 .. FFFF _{hex}	▶ Sub. 0E _{hex} y = x	
	Sub. 0F _{hex} x = 0 .. 2 ³² -1	▶ P1200 y = x	PPosHomePosition	P1200 x = 0 .. FFFFFFFF _{hex}	▶ Sub. 0F _{hex} y = x	
	Sub. 10 _{hex} x = 0 .. 2 ¹⁶ -1	▶ P1208 y = x	PPosSwitchMode	P1208 x = 0 .. FFFF _{hex}	▶ Sub. 10 _{hex} y = x	
	Sub. 11 _{hex} /ro		PPosSpeedSetValue	P0464 x = -32768 .. 32767	▶ Sub. 11 _{hex} y = x	



CANopen-Objekt	Index Wertebereich	P-Nr. Normierung	Regler-Parameter	P-Nr. Wertebereich	Index) Rück-Normierung	Kommentar
	Sub. 12 _{hex} x = 0 .. 2 ¹⁶ -1	▶ P0605 ▶ y = x	PPosBend0	P0605 x = 0 .. 8191	▶ Sub. 12 _{hex} ▶ y = x	
	Sub. 13 _{hex} x = 0 .. 2 ³² -1	▶ P1198 ▶ y = x	PPosClipEnvironment1	P1198 x = 0.. FFFFFFFF _{hex}	▶ Sub. 13 _{hex} ▶ y = x	
	Sub. 14 _{hex} x = 0 .. 2 ³² -1	▶ P1199 ▶ y = x	PPosClipEnvironment2	P1199 x = 0 .. FFFFFFFF _{hex}	▶ Sub. 14 _{hex} ▶ y = x	
	Sub. 15 _{hex} x = -2 ¹⁵ .. 2 ¹⁵ -1	▶ P0601 ▶ y = x	PPosTargetInput0	P0601 x = -2 ¹⁵ .. 2 ¹⁵ -1	▶ Sub. 15 _{hex} ▶ y = x	
	Sub. 16 _{hex} x = -2 ¹⁵ .. 2 ¹⁵ -1	▶ P0608 ▶ y = x	PposTargetInput1	P0608 -2 ¹⁵ .. 2 ¹⁵ -1	▶ Sub. 16 _{hex} ▶ y = x	
	Sub. 17 _{hex} x = 0 .. 2 ³²	▶ P0370 ▶ y = x	PosIpSetAngle	P0370 x = 0 .. 2 ³²	▶ Sub. 17 _{hex} ▶ y = x	
	Sub. 18 _{hex} x = 0 .. 2 ¹⁶	▶ P1209 ▶ y = x	PPosEncoderOffset	P1209 x = 0 .. 2 ¹⁶	▶ Sub. 18 _{hex} ▶ y = x	
	Sub. 19 _{hex} x = 0 .. 2 ¹⁶	▶ P1209 ▶ y = x	PPosHomingDeceler	P1204 x = 0 .. 2 ¹⁶	▶ Sub. 19 _{hex} ▶ y = x	
	Sub. 1A _{hex} ro		SpeedActValue	P0353 x = -2 ³² .. 2 ³² -1	▶ Sub. 1A _{hex} ▶ y = x	Der Drehzahlwert (P0353) wird von einem 32 Bit Wert auf 16384 umnormiert. 100 % der maximalen Drehzahl (in P1031) entsprechen 16384 Einheiten. Es wird der Betrag ausgegeben.
	Sub. 1B _{hex} ro		AmpWarning/MotorWarning	P0262, P0263 x = 0 .. 2 ¹⁶	▶ Sub. 1B _{hex} ▶	Bit 0 P0263 Bit 1 Motortemperatur hat Schwelle 2 überschritten Bit 1 P0263 Bit 1 Motortemperatur hat Schwelle 2 überschritten Bit 2 P0262 Bit 1 Leistungsteil Temperatur > 80°C Bit 3 nicht belegt Bit 4 P0263 Bit 0 Motortemperatur hat Schwelle 1 überschritten Bit 5 P0263 Bit 0 Motortemperatur hat Schwelle 1 überschritten

CANopen-Objekt	Index Wertebereich	P-Nr. Normierung	Regler-Parameter	P-Nr. Wertebereich	Index) Rück-Normierung	Kommentar
digital_inputs	60FD _{hex} /ro x = 0 .. 2 ¹⁶	▶ P0461	DigInOutStatus	P0461 x = 0 .. FFFF _{hex}	▶ 60FD _{hex} ▶ y = 0 .. 2 ³²	
Negative limit switch			Zustand Endschalter neg.	Bit 0	▶ Bit 0	
Positive limit switch			Zustand Endschalter pos.	Bit 1	▶ Bit 1	
Home switch			Zustand Nullpunktumschalter	Bit 2	▶ Bit 2	
Interlock			reserviert	Bit 4		
reserved			reserviert	Bit 3..15		
Man. specific			unbenutzt		Bit 16..31	
target_velocity	60FF _{hex} x = -2 ³¹ .. 2 ³¹ -1	▶ P1171 ▶ y = x * 40000000 _{hex} / MotorMax-Speed	RFG1Input	P1171 x = 8000 _{hex} .. 7FFF _{hex}	▶ 60FF _{hex} ▶ y = x * 40000000 _{hex} / MotorMax-Speed	Die benutzerdefinierte Einheit (velocity units) wird im b maXX [®] -Regler als U/min interpretiert. Nur bei Änderungen in Optionsmodul G/H-Konfiguration 1 Bit 2 = 1: Angabe der akt. Drehzahl in 1/10 U/min. z.B.: 200,0 Umdrehungen => Eingabe 2000.
drive_data	6510 _{hex}				6510 _{hex}	
Manufacturer specific	Sub. 01 _{hex} / ro			P0001	▶ Sub. 01 _{hex}	
			ControllerType		▶ y = x	
Manufacturer specific	Sub. 02 _{hex} / ro			P0002	▶ Sub. 02 _{hex}	
			SoftwareType		▶ y = x	
Manufacturer specific	Sub. 03 _{hex} / ro			P0003	▶ Sub. 03 _{hex}	
			SoftwareID		▶ y = x	
Manufacturer specific	Sub. 04 _{hex} / ro			P0004	▶ Sub. 04 _{hex}	
			SoftwareVersion		▶ y = x	
Manufacturer specific	Sub. 05 _{hex} / ro			P0005	▶ Sub. 05 _{hex}	
			ParamTableVersion		▶ y = x	
Manufacturer specific	Sub. 06 _{hex} / ro			P0009	▶ Sub. 06 _{hex}	
			AmpSW_Version		▶ y = x	
Manufacturer specific	Sub. 07 _{hex} / ro			P0555	▶ Sub. 07 _{hex}	
			FbgaVersion		▶ y = x	
Manufacturer specific	Sub. 08 _{hex} / ro			P0556	▶ Sub. 08	
			BootloaderVersion		▶ y = x	





ANHANG D - TECHNISCHE DATEN

In diesem Anhang finden Sie eine Übersicht der Technischen Daten der CANopen Optionskarte.

D.1 CANopen Optionskarte: Technische Merkmale

CPU	SAB 80C167CR
FPGA	XCS2S15-5TQ144C der SpartanII Reihe (Fa. XILINX)
CAN-Controller	im Prozessor integriert
Speicher	4 kByte DP-RAM, 256 kByte RAM, 1 MByte Flash-Eprom
Baudrate	max. 1MBit/s, 500kBits, 250kBits, 125kBits, 20kBits
Physical Layer	ISO 11898
Betriebsspannung	+5 V intern
Potentialtrennung	Optokoppler, DC/DC-Wandler
Steckverbinder	2 RJ45 Buchsen, 8-polig ¹⁾

¹⁾ Die RJ45-Anschlüsse sind intern 1:1 verbunden und zum Regler galvanisch getrennt

D.2 CANopen Optionskarte: Datenkanäle zum b maXX[®]-Regler

Für die Datenübertragung vom b maXX[®]-Regler zum Optionsmodul CANopen-Slave stehen drei Kanäle zur Verfügung:

- zwei Prozessdatenkanäle (4 PDOs je Kommunikationsrichtung)
- ein Bedarfsdatenkanal (Server-SDO)

Mit PDOs können im zyklischen Datenaustausch Objekte übertragen werden. Für den PDO-Transfer sind nicht alle Objekte verfügbar.

Mit dem SDO-Transfer kann über das Objektverzeichnis auf alle b maXX[®] 4400-Parameter zugegriffen werden.

D.3 CANopen Optionskarte: CAN-Buffer

Das Baumüller CANopen-Interface entspricht dem Full-CAN-Konzept, d. h. es steht für jeden Nachrichtentyp ein CAN-Buffer zur Verfügung.

CAN Buffer Bezeichnung	Übertragungsrichtung
NMT-Nachricht	Empfangen
TX-PDO1	Senden
TX-PDO2	Senden
TX-PDO3	Senden
TX-PDO4	Senden
RX-PDO1	Empfangen
RX-PDO2	Empfangen
RX-PDO3	Empfangen
RX-PDO4	Empfangen
SDO_TX	Senden
SDO_RX	Empfangen
Synchronisations-Objekt	Empfangen
Emergency-Objekt	Senden
Node Guard-Objekt	Empfangen / Senden



Abbildungsverzeichnis

Profilstruktur von CANopen.....	11
PDO Übertragungsarten	12
Zustandsmaschine der Kommunikation	43
NMT-Telegramm zur Steuerung der Kommunikationszustände	44
Node Guarding Protokoll.....	47
Heartbeat Protokoll	48
Initiate SDO Download Protocol.....	50
Initiate SDO Upload Expedited	52
Upload SDO Segmented Protocol	55
Kommunikationszyklus.....	59
Beispiel-Mapping mit zwei b maXX® 4400	67
Telegrammaufbau für Beispiel-Mapping	76





Stichwortverzeichnis

A		S	
Abort Codes	57	SDO	12
Arbitrierung	10	synchron	12
		Synchronisation	59
B		Z	
Bedarfsdaten	19	Zugriffe	
Buszugriff	10	fehlerhaft	57
		Zustandsmaschine	43
C			
CAL-Spezifikation	11		
CSMA/CA	10		
E			
Emergency-Telegramme	79		
F			
fehlerhafte Zugriffe	57		
G			
Geräteprofil	11		
Guarding-Zeit	47		
I			
Identifiervergabe	13		
K			
Knotenanzahl			
maximal	13		
Kommunikationszustände			
boot-up	13		
L			
Leitung	10		
Literatur zum Thema CAN	9		
N			
Netzwerkmanagement	43		
NMT	12		
Node Guarding	46		
O			
Objektverzeichnis	11		
P			
PDO	12		
PDO-Mapping	61		
fehlerhaft	78		
Prozessdaten	19		
R			
Regler-Fehlermeldungen	80		
Remoteframes	46		
RTR-Bit	46		

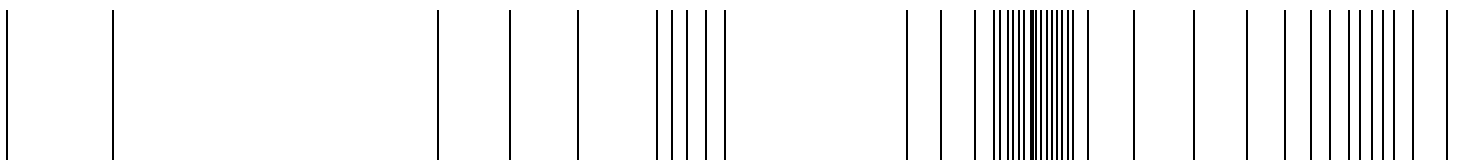




Revisionsübersicht

Version	Stand	Änderungen
5.02065.03	15.02.2006	Konfiguration CANopen-Optionskarte, Fehlertelegramme, Umsetzungstabellen
5.02065.04	06.02.2012	Erweiterung für Lageregelung und Gleichlauf Überarbeitung Fehlermeldungen (Kap. 5.6.2) Überarbeitung 6000'er Objektnummern (Anh. B.2) Überarbeitung Umsetzungstabellen (Anh. C)

be in motion



Baumüller Nürnberg GmbH Ostendstraße 80-90 90482 Nürnberg T: +49(0)911-5432-0 F: +49(0)911-5432-130 www.baumueller.de

Alle Angaben in diesem Programmierhandbuch sind unverbindliche Kundeninformationen, unterliegen einer ständigen Weiterentwicklung und werden fortlaufend durch unseren permanenten Änderungsdienst aktualisiert. Bitte beachten Sie, dass Angaben/Zahlen/Informationen aktuelle Werte zum Druckdatum sind.
Zur Ausmessung, Berechnung und Kalkulationen sind diese Angaben nicht rechtlich verbindlich. Bevor Sie in diesem Programmierhandbuch aufgeführte Informationen zur Grundlage eigener Berechnungen und/oder Verwendungen machen, informieren Sie sich bitte, ob Sie den aktuellsten Stand der Informationen besitzen.
Eine Haftung für die Richtigkeit der Informationen wird daher nicht übernommen.