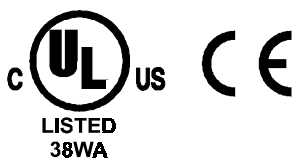


POWER CONVERSION EQUIPMENT



D	5.01007.01
----------	-------------------

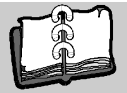
**OPTIONSKARTE
CANOPEN
FÜR V-REGLER**

BETRIEBSANLEITUNG



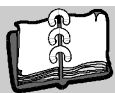
BAUMÜLLER

Titel	Betriebsanleitung
Produkt	Optionskarte CANopen
Stand	5. Dezember 2001
Copyright	<p>Diese Produktdokumentation darf vom Eigentümer ausschließlich für den internen Gebrauch in beliebiger Anzahl kopiert werden. Für andere Zwecke darf diese Produktdokumentation auch auszugsweise weder kopiert noch vervielfältigt werden.</p> <p>Verwertung und Mitteilung von Inhalten dieser Produktdokumentation sind nicht gestattet.</p> <p>Bezeichnungen bzw. Unternehmenskennzeichen in dieser Dokumentation können Marken sein, deren Benutzung durch Dritte für deren Zwecke die Rechte der Inhaber verletzen kann.</p>
Verbindlichkeit	<p>Diese Produktdokumentation ist Teil des Gerätes/der Maschine. Diese Produktdokumentation muß jederzeit für den Bediener zugänglich und in einem leserlichen Zustand sein. Bei Verkauf/Verlagerung des Gerätes/der Maschine muß diese Produktdokumentation vom Besitzer zusammen mit dem Gerät/der Maschine weitergegeben werden. 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 der vorliegenden Produktdokumentation werden entsprechende Produktdokumentationen mit einem früheren Stand außer Kraft gesetzt.</p> <p>Die Firma Baumüller behält sich vor, im Rahmen der eigenen Weiterentwicklung der Produkte die technischen Daten und ihre Handhabung von Baumüller-Produkten zu ändern.</p> <p>Es kann jedoch keine Gewährleistung bezüglich der Fehlerfreiheit dieser Produktdokumentation, soweit nicht in den Allgemeinen Verkaufs- und Lieferbedingungen anders beschrieben, übernommen werden.</p>
Hersteller	<p>Baumüller Nürnberg GmbH Ostendstr. 80 - 90 90482 Nürnberg Deutschland Tel. + 49 9 11 54 32-0 Fax -1 30</p>



Inhaltsverzeichnis

1	Einleitung	5
1.1	Verwendete Begriffe	5
1.2	Lieferumfang	5
1.3	Funktionsumfang	5
1.4	Grundlagen	7
1.5	EDS-Datei	10
2	Sicherheitshinweise	11
2.1	Gefahrenhinweise	11
2.2	Infozeichen	12
2.3	Qualifiziertes Personal	12
2.4	Bestimmungsgemäße Verwendung	12
3	Installation	13
3.1	Frontansicht V-Regler mit Optionskarte CANopen	13
3.2	Steckerbelegung	14
3.3	DIP-Schalter	15
3.3.1	Adresseinstellung	15
3.3.2	Baudrateneinstellung	16
4	Inbetriebnahme	17
4.1	Reglerinbetriebnahme	17
4.2	Schnittstelleninbetriebnahme	17
5	Kommunikation	19
5.1	Kommunikationsablauf auf der Optionskarte	19
5.2	Kommunikationszeiten	20
5.2.1	Aktualisierungszeiten zwischen Optionskarte und Regler	20
5.2.2	Verarbeitungszeiten auf der Optionskarte	21
6	Datenaustausch und Parametrierung	23
6.1	Verzeichnis der Objekte zur Kommunikationssteuerung	23
6.2	Netzwerkmanagement (NMT)	31
6.2.1	Zustandsmaschine der Kommunikation	31
6.2.2	Telegramme	32
6.2.3	Node Guarding	34
6.3	Bedarfsdaten (SDO)	35
6.3.1	Telegrammaufbau	36
6.3.2	Arten des SDO-Transfers	36
6.3.3	Objekt schreiben	37
6.3.4	Objekt lesen	40
6.3.5	Fehlerreaktionen	45
6.4	Synchronisation (SYNC)	47
6.5	Prozessdaten (PDO)	48
6.5.1	PDO-Mapping	48
6.5.2	Kommunikationsbeziehung über PDO	50
6.5.3	Beispiel für das PDO-Mapping	51
6.5.4	Eintrag in die BAPS	57



Inhaltsverzeichnis

7 Fehlerbehandlung	59
7.1 Problembehebung	59
7.2 Fehlertelegramm (EMCY)	60
7.2.1 Telegrammaufbau	60
7.2.2 Umsetzung der V-Regler-Fehlermeldungen auf CANopen-Fehlermeldungen	61
8 Kurzreferenz zur Parameterumsetzung	63
8.1 V-Regler-Parameter auf CANopen-Objekt	63
8.2 CANopen-Objekte auf V-Regler-Parameter	75
Anhang A - Abkürzungen	77
Anhang B - Umsetzungstabellen	79

1

EINLEITUNG

Die in der Automatisierungstechnik voranschreitende Dezentralisierung erfordert den Einsatz robuster serieller Feldbussysteme. Sie dienen sowohl der Kommunikation der einzelnen Regler und der Steuerung, wie auch dem Erfassen und Bedienen der unterschiedlichen Sensoren und Aktoren. Somit lassen sich Projektierungs- und Verkabelungsaufwand drastisch verringern.

Für den Antriebsregler wurde ein modulares Konzept vorgesehen, d.h., der Antriebsregler kann mit steckbaren Erweiterungskarten an unterschiedliche Anforderungen angepaßt werden. Die CANopen-Anschaltung ist als Optionskarte realisiert. Diese Optionskarte verbindet den V-Regler über den CAN-Bus mit anderen CANopen-Knoten (z.B. PC, SPS, weitere V-Regler, I/O-Module).

1.1 Verwendete Begriffe

Diese Betriebsanleitung beschreibt den Einsatz der Optionskarte CANopen im V-Regler. Der V-Regler wird auch als BUS 6 VC bezeichnet.

Eine Zusammenfassung der verwendeten Abkürzungen finden Sie in [▶Anhang A - Abkürzungen](#) ab Seite 77.

1.2 Lieferumfang

- ▶ Optionskarte CANopen 3.9819
- ▶ EDS-Datei (Electronic Data Sheet)
- ▶ Technische Beschreibung

1.3 Funktionsumfang

Für Datenübertragung stehen drei Kanäle zur Verfügung:

- ▶ zwei Prozessdatenkanäle (2 PDOs je Kommunikationsrichtung)
- ▶ ein Bedarfsdatenkanal (Server-SDO)

Mit dem SDO-Transfer kann über das Objektverzeichnis auf alle V-Regler-Parameter zugegriffen werden.

Mit PDOs können im zyklischen Datenaustausch Objekte übertragen werden. Für den PDO-Transfer sind nicht alle Objekte verfügbar. Informationen dazu sind in der EDS-Datei verfügbar.

Technische Merkmale der Optionskarte CANopen:

CPU	SAB 80C167CR
CAN-Controller	im Prozessor integriert
Speicher	4 kByte DP-RAM, 64 kByte RAM, 256 kByte Flash-EPROM
Baudrate	max. 1MBit/s
Physical Layer	ISO 11898
Temperaturbereich	0 ... 55°C
Betriebsspannung	+5 V (von Regler-Versorgung)
Potentialtrennung	Optokoppler, DC/DC-Wandler
Steckverbinder	Sub-D-Stecker und Buchse, 9polig

Abbildung 1: Technische Merkmale

Die Sub-D-Anschlüsse sind intern 1:1 verbunden und zum Regler galvanisch getrennt.

Prinzipieller Aufbau der Optionskarte CANopen:

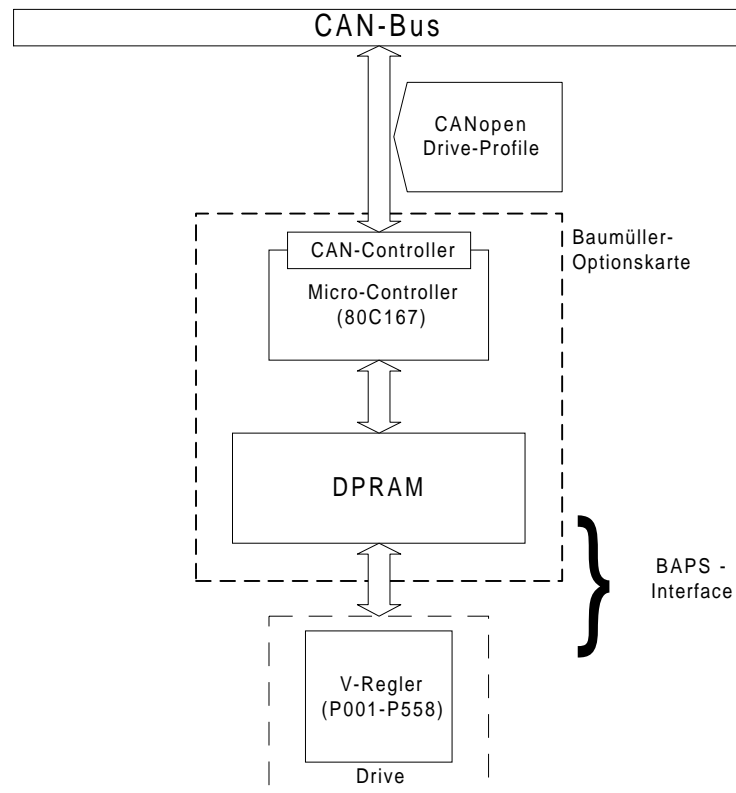


Abbildung 2: Konzept der CANopen-Anschaltung

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

Objektbuffer für CANopen-Telegramme

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

Abbildung 3: Objektbuffer für CANopen-Telegramme

1.4 Grundlagen

Grundlagen CAN

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.

CAN ist ein Multi-Master-Netzwerk. Jeder Teilnehmer kann gleichberechtigt und aktiv auf den Bus zugreifen. CAN verwendet die objektorientierte Adressierung, d.h., nicht die Teilnehmer erhalten eine Adresse, sondern die übermittelte Nachricht wird mit einem netzwerkweit festgelegten Identifier gekennzeichnet. Er stellt den codierten Namen der Nachricht dar.

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 Worst Case 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.

Es stehen im CAN 2.0A 2032 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.

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 Hammingdistanz (Maß der Fehlererkennung) von HD=6 erreicht, d.h., maximal fünf gleichzeitig auftretende Bitfehler innerhalb eines Telegramms werden sicher als Fehler erkannt.

Grundlagen CANopen

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

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

In den Geräteprofilen werden die anwendungs- und gerätespezifischen Festlegungen, die inhaltliche Bedeutung der Daten und die Gerätefunktionalität beschrieben. Geräteprofile existieren u.a. für Antriebe, I/O-Module, Geber oder programmierbare Geräte.

Die Optionskarte für den V-Regler ist nach dem Geräteprofil DS402 (Drives and Motion Control) implementiert.

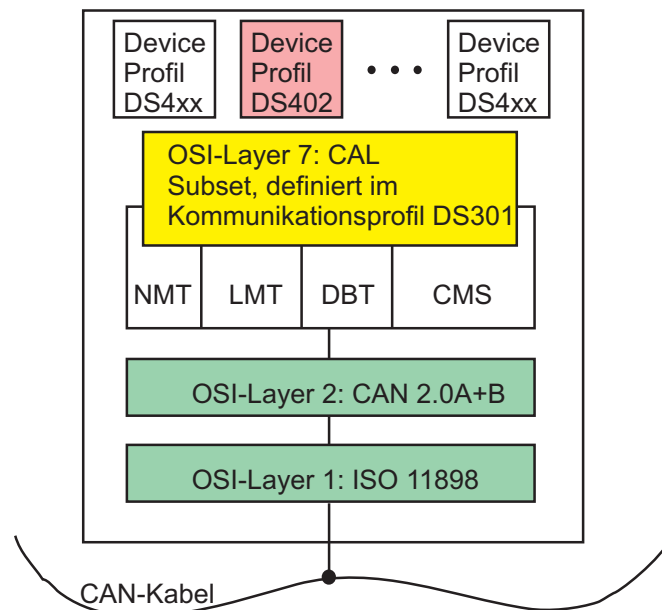


Abbildung 4: Profilstruktur von CANopen

Das zentrale Element eines jeden CANopen-Gerätes bildet sein Objektverzeichnis.

Index (hex)	Objekt
0000	nicht benutzt
0001 - 001F	Statische Datentypen
0020 - 003F	Komplexe Datentypen
0040 - 005F	Herstellerspezifische Datentypen
0060 - 007F	Geräteprofilsspezifische Statische Datentypen
0080 - 009F	Geräteprofilsspezifische Dynamische Datentypen
00A0 - 0FFF	Reserviert
1000 - 1FFF	Bereich für das Kommunikationsprofil
2000 - 5FFF	Bereich für herstellerspezifische Objekte
6000 - 9FFF	Bereich für das Geräteprofil
A000 - FFFF	Reserviert

Abbildung 5: Struktur des CANopen-Objektverzeichnisses

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)

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

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). Datengrößen von maximal vier Byte werden mit einer SDO übertragen (SDO Expedited Transfer). Typischerweise werden SDOs zur Gerätekonfigurierung benutzt. SDOs werden asynchron übertragen und vom Empfänger bestätigt. Mittels SDO kann auf alle vorhandenen Einträge im Objektverzeichnis zugegriffen werden.

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.

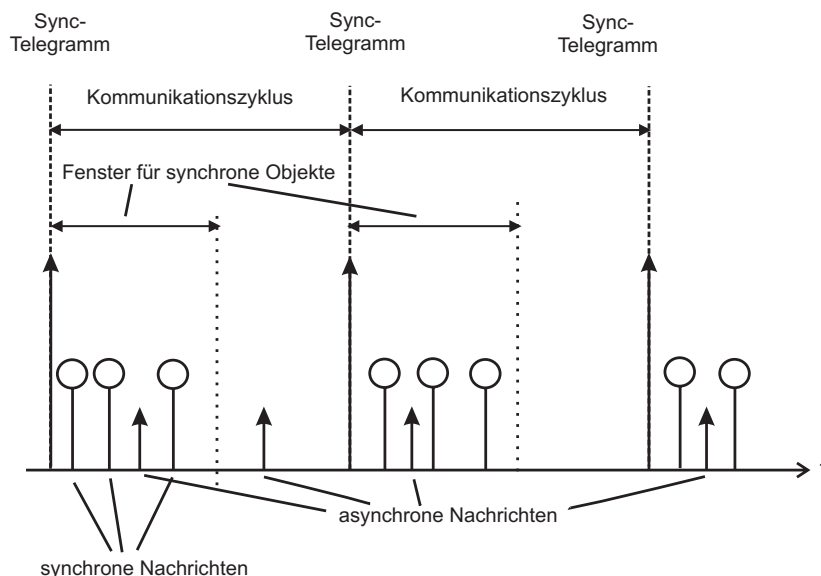


Abbildung 6: PDO Übertragungsarten

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 (siehe auch [►Prozessdaten \(PDO\)◄](#) ab Seite 48).

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						

Abbildung 7: Aufbau des Identifiers

Aus den sieben Bit für die Module-ID ergibt sich je CANopen-Netz 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} , 1A00 _{hex}
PDO1 (RX)	0100	513 - 639	1400 _{hex} , 1600 _{hex}
PDO2 (TX)	0101	641 - 767	1801 _{hex} , 1A01 _{hex}
PDO2 (RX)	0110	769 - 895	1401 _{hex} , 1601 _{hex}
SDO (TX)	1011	1409 - 1535	1200 _{hex}
SDO (RX)	1100	1537 - 1663	1200 _{hex}
Nodeguard	1110	1793 - 1919	100C _{hex} , 100D _{hex}

Abbildung 8: Peer-to-Peer Objekte nach dem Predefined Connection Set

CANopen definiert ein Boot Up des Netzwerkes mit vier Kommunikationszuständen:

- INITIALISIERUNG
- PRE-OPERATIONAL
- STOPPED
- OPERATIONAL

Die einzelnen Zustandsübergänge werden durch NMT-Kommandos ausgelöst.

1.5 EDS-Datei

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

Der Namensuffix dieser ASCII-Datei lautet *.eds.

SICHERHEITSHINWEISE

Jedes Baumüller-Gerät haben wir nach strengen Sicherheitsvorgaben konstruiert und gefertigt. Trotzdem kann die Arbeit mit dem Gerät für Sie gefährlich sein.

In diesem Kapitel beschreiben wir Gefahren, die bei der Arbeit mit dem Baumüller-Gerät auftreten können. Gefahren verdeutlichen wir mit Symbolen (Icons). Alle in dieser Dokumentation verwendeten Symbole werden wir auflisten und erklären.

Wie Sie sich vor den einzelnen Gefahren im konkreten Fall schützen können, können wir in diesem Kapitel nicht erklären. In diesem Kapitel geben wir ausschließlich allgemeine Schutzmaßnahmen. Die konkreten Schutzmaßnahmen werden wir in den nachfolgenden Kapiteln immer direkt nach dem Hinweis auf die Gefahr geben.

2.1 Gefahrenhinweise

Die Hinweise dienen einerseits der persönlichen Sicherheit des Anwenders und andererseits der Sicherheit vor Beschädigung der beschriebenen Produkte oder angeschlossenen Geräte.

Eine Gefahr teilen wir immer in eine der drei Gefahrenklassen ein. Jede Gefahrenklasse wird durch eines der folgenden Signalwörter gekennzeichnet:

GEFAHR (DANGER)

▶ erheblicher Sachschaden ▶ schwere Körperverletzung ▶ Tod - **wird** eintreffen

WARNUNG (WARNING)

▶ erheblicher Sachschaden ▶ schwere Körperverletzung ▶ Tod - **kann** eintreffen

VORSICHT (CAUTION)

▶ Sachschaden ▶ leichte Körperverletzung - **kann** eintreffen

Zur optischen Unterscheidung verwenden wir für jeden Klasse von Gefahrenhinweisen eine eigenen Umrandung für die Piktogramme und entsprechende Abgrenzungslinien.

Verwendete Gefahrenhinweise



GEFAHR (DANGER)

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

▶ erheblicher Sachschaden ▶ schwere Körperverletzung ▶ Tod

2.2 Infozeichen



WARNUNG (WARNING)

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

- ▶ erheblicher Sachschaden
- ▶ schwere Körperverletzung
- ▶ Tod



VORSICHT (CAUTION)

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

- ▶ Sachschaden
- ▶ leichte bis mittlere Verletzung

2.2 Infozeichen



HINWEIS

Dieser Hinweis ist eine besonders wichtige Information.

2.3 Qualifiziertes Personal

Qualifiziertes Personal im Sinne der sicherheitsbezogenen Hinweise in dieser Betriebsanleitung oder auf den Produkten selbst sind Personen, die mit Montage, Inbetriebsetzung und Betrieb des Produktes vertraut sind und über die ihrer Tätigkeit entsprechenden Qualifikation verfügen:

- ▶ Ausbildung oder Unterweisung bzw. Berechtigung Stromkreise und Geräte gemäß den Standards der Sicherheitstechnik in Betrieb zu nehmen, zu erden und zu kennzeichnen.
- ▶ Ausbildung oder Unterweisung gemäß den Standards der Sicherheitstechnik in Pflege und Gebrauch angemessener Sicherheitsausrüstung.

2.4 Bestimmungsgemäße Verwendung



WARNUNG (WARNING)

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

- ▶ erheblicher Sachschaden
- ▶ schwere Körperverletzung
- ▶ Tod

Die Einheit / das System darf nur für die in der Betriebsanleitung vorgesehenen Einsatzfälle und nur in Verbindung mit von der BAUMÜLLER NÜRNBERG GmbH empfohlenen bzw. zugelassenen Fremdgeräten und -komponenten verwendet werden.

Eigenmächtige Umbauten und Veränderungen an der Einheit sind aus Sicherheitsgründen nicht gestattet. Der Bediener ist verpflichtet, eintretende Veränderungen, die die Sicherheit der Einheit / des Systems beeinträchtigen könnten, sofort zu melden.

3

INSTALLATION

Dieses Kapitel beschreibt die Verkabelung und entsprechende Einstellung der Karte.

3.1 Frontansicht V-Regler mit Optionskarte CANopen

Die Optionskarte CANopen steht für unterschiedliche Bauformen des V-Reglers zur Verfügung.

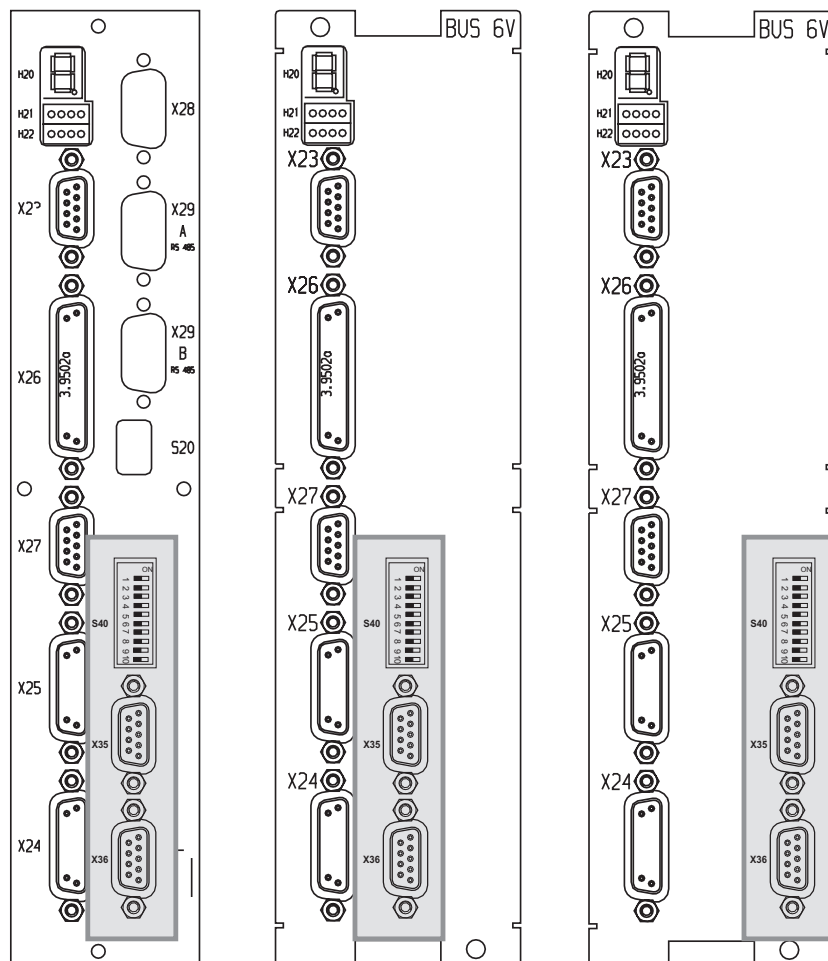
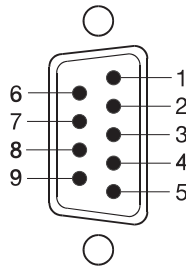


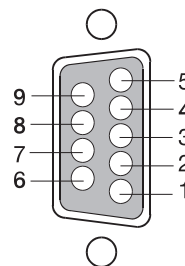
Abbildung 9: Frontansicht V-Regler mit Optionskarte CANopen

3.2 Steckerbelegung

Stecker X35



Buchse X36



Pin Nr.	Belegung
1	reserviert
2	CAN-LOW
3	GND
4	reserviert
5	reserviert
6	reserviert
7	CAN-HIGH
8	reserviert
9	reserviert

Abbildung 10: Stecker X 35 und Buchse X36

Die CAN-Leitung ist beidseitig mit einem Abschlusswiderstand von 120 ... 124 Ω zu terminieren. Die reservierten Pins sind nicht zu belegen.

- Verbindung vom 1. Busteilnehmer zum 2. Busteilnehmer (usw.)

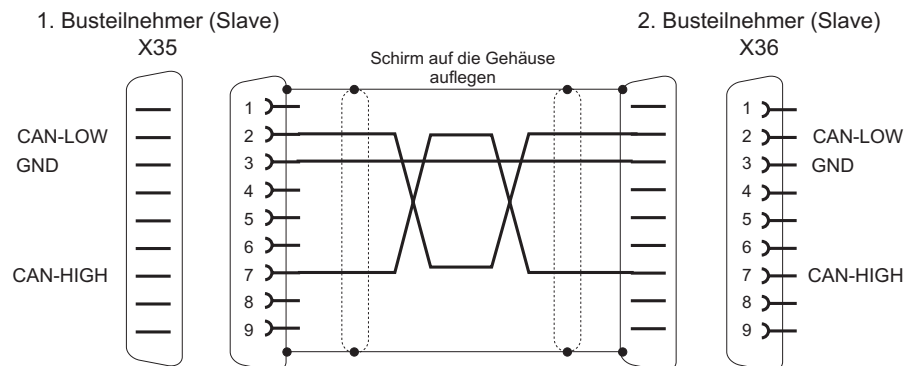


Abbildung 11: Verbindung der Busteilnehmer

- Busabschluss-Stecker für letzten Busteilnehmer

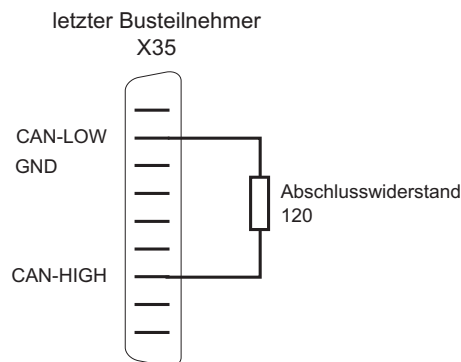
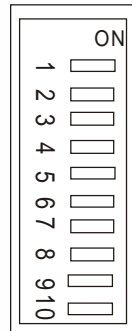


Abbildung 12: Busabschluss-Stecker

3.3 DIP-Schalter

Mittels DIP-Schalter werden die Datenübertragungsrate auf dem CAN-Bus und die Slave-Adresse selektiert.

DIP-Schalter S40



Pin Nr.	Belegung
1	Baudrate-Einstellung Bit 0 links = 0, rechts (ON) = 1
2	Baudrate-Einstellung Bit 1 links = 0, rechts (ON) = 1
3	Baudrate-Einstellung Bit 2 links = 0, rechts (ON) = 1
4	Bus-Adresse Bit 0 links = 0, rechts (ON) = 1
5	Bus-Adresse Bit 1 links = 0, rechts (ON) = 1
6	Bus-Adresse Bit 2 links = 0, rechts (ON) = 1
7	Bus-Adresse Bit 3 links = 0, rechts (ON) = 1
8	Bus-Adresse Bit 4 links = 0, rechts (ON) = 1
9	Bus-Adresse Bit 5 links = 0, rechts (ON) = 1
10	Bus-Adresse Bit 6 links = 0, rechts (ON) = 1

Abbildung 13: DIP-Schalter S40

3.3.1 Adresseinstellung

Die DIP-Schalter 4 bis 10 dienen der Einstellung der Geräteadresse (Module-ID) im CANopen-Netzwerk. Es lassen sich physikalisch Adressen von 0 bis 127 einstellen. Die Adressen können im CANopen-Netzwerk auch beliebig, d.h., nicht fortlaufend, vergeben werden. Es besteht kein Zusammenhang zwischen der örtlichen Lage des Gerätes im Netz und der Geräteadresse.

Die über DIP-Schalter eingestellte Adresse wird automatisch um den Wert 1 erhöht.

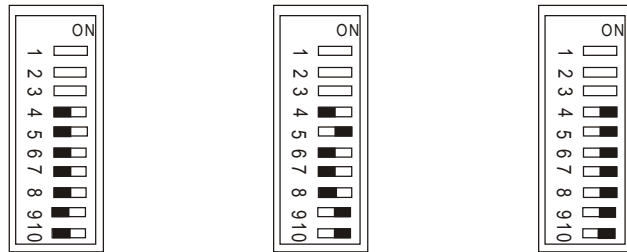
Adressbildung: tatsächliche Adresse = DIP-Schalteradresse + 1

Die DIP-Schalteradressen 0 und 127 führen zur tatsächlichen Adresse 1.

Innerhalb eines CANopen-Systems dürfen Geräteadressen nicht doppelt vergeben werden.

Die Einstellung der Adressen muß bei ausgeschaltetem Regler erfolgen, da jede Änderung der Adresse bei Betrieb wirkungslos bleibt.

3.3 DIP-Schalter



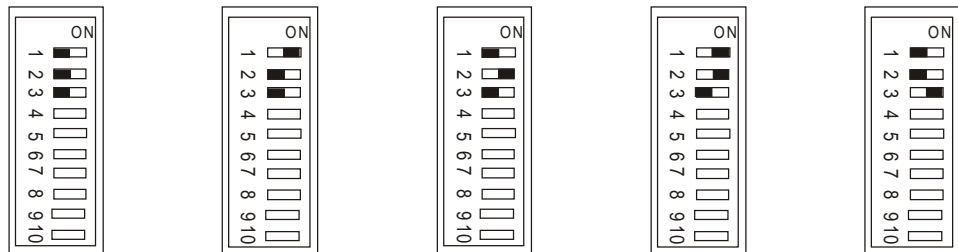
DIP-Adresse	0	98	127
Resultierende Modul-ID:	1	99	1

Abbildung 14: Beispiele für die Adresseinstellung

3.3.2 Baudrateneinstellung

Mittels der DIP-Schalter 1, 2 und 3 können Datenübertragungsraten von 20 kBit/s, 125 kBit/s, 250 kBit/s, 500 kBit/s, 1 Mbit/s eingestellt werden.

Die Einstellung muß bei ausgeschaltetem Regler erfolgen. Eine Veränderung der DIP-Schalterstellung während des Betriebes hat keinen Einfluß. Alle Geräte am selben CAN-Bus müssen die selbe Übertragungsrate verwenden.



DIP-Wert	0	1	2	3	4
Baudrate	20 kBit/s	125 kBit/s	250 kBit/s	500 kBit/s	1 Mbit/s

Abbildung 15: Einstellung der Baudrate

4

INBETRIEBNAHME

Die Inbetriebnahme des V-Reglers mit Optionskarte CANopen unterteilt sich in die Inbetriebnahme des V-Reglers und der Einstellungen der Schnittstellenparameter auf der Regler- und Master-Seite.

4.1 Reglerinbetriebnahme

Als erster Schritt der Inbetriebnahme sollte der V-Regler in seinen gewünschten Betriebsarten eingestellt und optimiert werden. Nach Abschluß der Inbetriebnahme sollte aus Sicherheitsgründen eine Speicherung der Daten auf dem PC erfolgen, um bei einer fehlerhaften Ansteuerung über das CANopen eine Sicherung der eingestellten Daten zu haben.

4.2 Schnittstelleninbetriebnahme

Für die Funktion der CANopen-Optionskarte sind im V-Regler-Parameter *P126 M Komm. Quelle* sowohl die zyklische als auch die Bedarfsdatenkommunikation freigegeben werden (z.B. P126 = 0x0C).

Folgende Schritte sind für den zyklischen Datenaustausch zwingend erforderlich:

- ▶ Adresse und Baudrate an DIP-Schalter S40 einstellen
- ▶ Gerät einschalten
- ▶ Kommunikationsquelle (*P126*) aktivieren
- ▶ PDO-Mapping erstellen, falls nicht das Default-Mapping verwendet wird (siehe auch [▶PDO-Mapping◀](#) ab Seite 48)
- ▶ PDO-Identifizier und Triggertyp konfigurieren, falls nicht die Defaultwerte benutzt werden (siehe auch [▶Kommunikationsbeziehung über PDO◀](#) ab Seite 50)
- ▶ In den Zustand OPERATIONAL wechseln (muß vom Master über NMT-Dienst vorgegeben werden)

KOMMUNIKATION

5.1 Kommunikationsablauf auf der Optionskarte

Die CANopen-Optionskarte tauscht über ein Dual-Port-RAM Daten mit dem V-Regler aus. Dieser Datenaustausch geschieht in einem bestimmten Zeitraster.

Für den Prozessdatenaustausch gelten die unten dargestellten Abläufe. Die Optionskarte stößt die Kommunikation mit dem V-Regler an.

Die Zykluszeit des Prozessdatenaustausches beträgt 500 μ s.

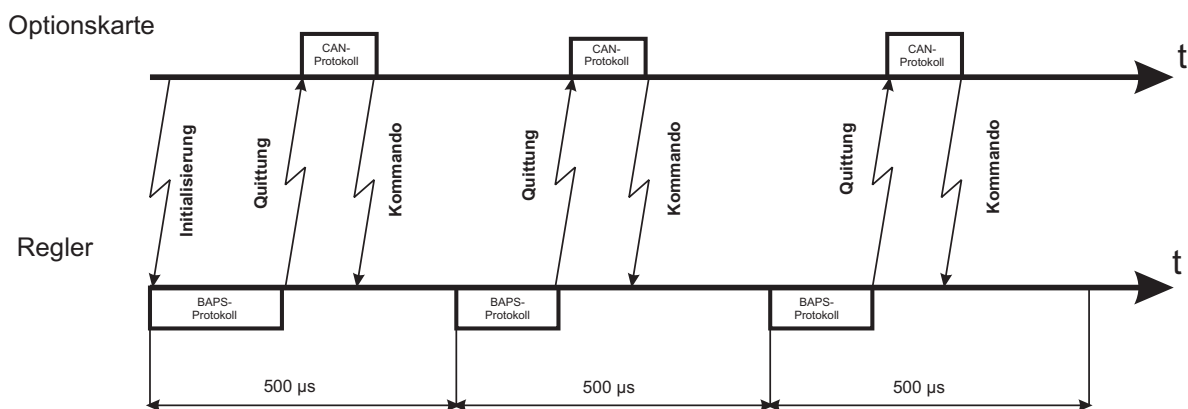


Abbildung 16: Kommunikationsablauf

Mit "Kommando" werden Steuerwort und zyklische Sollwerte zum Regler geschrieben, mit "Quittung" werden Statuswort und zyklische Istwerte vom Regler gelesen.

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

Die Bedarfsdatenkommunikation wird in der Restzeit ausgeführt.

5.2 Kommunikationszeiten

Die Gesamtzeit vom Absenden einer Nachricht im Master bis zum Wirksamwerden der Daten in Regler hängt von der Anzahl der zyklischen Daten, der Übertragungsrate und der Buslast ab.

5.2.1 Aktualisierungszeiten zwischen Optionskarte und Regler

Zeitscheibenverfahren

Bei dieser Übertragungsart werden die zyklischen Soll- und Istwerte in einem Zeitscheiben-Verfahren übertragen. Es ergeben sich folgende Aktualisierungszeiten zwischen der Zusatzkarte und dem V-Regler für die parametrisierten Soll- / Istwerte:

Soll- / Istwert-Nr.	1	2	3	4	5	6	7	8
Aktualisierungszeit	1 ms	2 ms	4 ms	8 ms	16 ms	32 ms	64 ms	128 ms

Beispiel für Wertübertragung zwischen V-Regler und Optionskarte bezogen auf einen Zeitraum von 20 ms:

Zeit [ms]	0	0,5	1		2		3		4		5		6	
Soll- / Istwert-Nr.	X	X	1	2	1	3	1	2	1	4	1	2	1	3
Aktualisierung ms	X	X	1	2	1	4	1	2	1	8	1	2	1	4
Zeit [ms]	7		8		9		10		11		12		13	
Soll- / Istwert-Nr.	1	2	1	5	1	2	1	3	1	2	1	4	1	2
Aktualisierung ms	1	2	1	16	1	2	1	4	1	2	1	8	1	2
Zeit [ms]	14		15		16		17		18		19		20	
Soll- / Istwert-Nr.	1	3	1	2	1	6	1	2	1	3	1	2	1	4
Aktualisierung ms	1	4	1	2	1	32	1	2	1	4	1	2	1	8

Abbildung 17: Zeitscheibenverfahren (siehe auch Profibus-Dokumentation)

2 Soll- und bis zu 4 Istwerte je Zyklus

Bei dieser Übertragungsart zwischen V-Regler und CANopen-Karte können nur max. 2 Soll- und max. 4 Istwerte verwendet werden, die dann in jedem Kommunikationszyklus zwischen V-Regler und CANopen-Karte übertragen werden. Das Zeitraster der Kommunikation beträgt auch hier 500µs. Dies bedeutet, dass alle 0,5 ms das Status- und Steuerwort sowie die Soll- und Istwerte aktualisiert werden.

Werden in den PDOs **insgesamt** bis zu 2 Sollwerte **und** bis zu 4 Istwerte gemappt, wobei Statuswort -, Steuerwort und Dummies nicht mitgerechnet werden, so wird immer der Modus "2 Soll- und max. 4 Istwerte" verwendet. In allen anderen Fällen, d.h. die Gesamtzahl der gemappten Objekte übersteigt 2 Sollwerte **oder** 4 Istwerte (Status-, Steuerwort und Dummies werden nicht mitgezählt), kommt das Zeitscheibenverfahren zur Anwendung.

Die Modus-Wahl wird von der CANopen-Optionskarte selbständig durchgeführt. Der Anwender hat darauf keinen Einfluß.

5.2.2 Verarbeitungszeiten auf der Optionskarte

Die Verarbeitungszeiten auf der CANopen-Karte hängen von der PDO-Konfiguration ab. Exemplarisch gelten folgende Werte für eventgetriggerte PDOs:

PDO-Konfiguration	Empfangs-Nachricht	Sende-Nachricht
1 x 16 Bit normiert (Objekt 2006 _{hex})	180 µs	170 µs
1 x 32 Bit unnormiert (Objekt 2257 _{hex})	170 µs	170 µs
4 x 16 Bit gemischt (Objekte 6042 _{hex} , 2006 _{hex} , 2241 _{hex} , 2004 _{hex})	680 µs	600 µs
2 x 32 Bit gemischt (Objekte 60FB _{hex} 09 _{hex} , 60FB _{hex} 0A _{hex})	330 µs	270 µs

Abbildung 18: Verarbeitungszeiten für Empfangs- und Sende PDOs

Die Zeiten gelten für die Telegrammbearbeitung, Normierung, Objektkonvertierung zwischen CAN-Controller und internen DP-RAM-Schnittstelle zum Regler

6

DATENAUSTAUSCH UND PARAMETRIERUNG

Der Zugriff auf Daten oder Parameter ist bei CANopen nur über Objekte möglich.

Entsprechend der Profilstruktur werden Objekte zur Kommunikationssteuerung (vgl. Verzeichnis der Objekte zur Kommunikationssteuerung, Indizes $1XXX_{hex}$) und anwendungs- oder gerätespezifische Objekte unterschieden. Die letzteren gliedern sich in Objekte nach Profil DS 402 (Indizes $6XXX_{hex}$) und herstellerspezifische Objekte (Indizes $2XXX_{hex}$, $3XXX_{hex}$). Diese Objekte sind im Anhang aufgelistet.

Wichtig: Bei den herstellerspezifischen Objekten ($2XXX_{hex}$) ergibt sich der Objektindex aus 2000_{hex} + V-Regler-Parameternummer, z.B. ist das Objekt 2126_{hex} auf den V-Regler-Parameter *126 M Kommunikationsquelle* umgesetzt, diese Objekte besitzen nur den Subindex 00_{hex} .

6.1 Verzeichnis der Objekte zur Kommunikationssteuerung

Hier sind sämtliche von der Baumüller CANopen-Optionskarte unterstützten Objekte des kommunikationsspezifischen Bereiches des Objektverzeichnis nach DS301 aufgeführt.

► **Objekt 1000_{hex}**

Name	Index	Subindex	Datentyp	Defaultwert
Device Type	1000_{hex}	00_{hex}	U32	00020192_{hex}

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

► Objekt 1001_{hex}

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

Dieses Objekt kann nur gelesen werden.

Es enthält eine Fehlerbitleiste, die folgende Bedeutung hat:

Bit	Bedeutung
0	Fehler aufgetreten
1	Nicht verwendet
2	Nicht verwendet
3	Nicht verwendet
4	Kommunikationsfehler CAN
5	Nicht verwendet
6	Nicht verwendet
7	Herstellerspezifischer Fehler

► Objekt 1002_{hex}

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-Statusworts aus Parameter *P121*.

► Objekt 1003_{hex}

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.

Im Low-Word steht der CANopen Error-Code (siehe auch [►Umsetzung der V-Regler-Fehlermeldungen auf CANopen-Fehlermeldungen](#) ab Seite 61)

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.

► **Objekt 1004_{hex}**

Name	Index	Subindex	Datentyp	Defaultwert
Number of PDOs supported	1004 _{hex}	00 _{hex}	U32	00020002 _{hex}
Synchronous PDOs		01 _{hex}	U32	00020002 _{hex}
Asynchronous PDOs		02 _{hex}	U32	00020002 _{hex}

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

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

► **Objekt 1005_{hex}**

Name	Index	Subindex	Datentyp	Defaultwert
COB-ID SYNC-Message	1005 _{hex}	00 _{hex}	U32	80000080h

Dieses Objekt enthält Informationen über das Sync-Verhalten des Slaves. Das MSB im High Word besagt, dass der Slave das SYNC-Telegramm nicht erzeugt, sondern nur empfängt. Die unteren 11 Bit im Low Word geben den Identifier des SYNC-Telegramms an (80_{hex}).

► **Objekt 1006_{hex}**

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

Dieses Objekt wird intern nicht ausgewertet.

► **Objekt 1007_{hex}**

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

Dieses Objekt wird intern nicht ausgewertet.

▶ **Objekt 1008_{hex}**

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

Dieses Objekt ist nur lesbar. Es enthält die Zeichenfolge: "BM_DRIVE".

▶ **Objekt 1009_{hex}**

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

Dieses Objekt ist nur lesbar. Es enthält die aktuelle Hardware-Version der Optionskarte, z.B. die Zeichenfolge: "HV03.00".

▶ **Objekt 100A_{hex}**

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

Dieses Objekt ist nur lesbar. Es enthält die aktuelle Software-Version der Optionskarte, z.B. die Zeichenfolge: "SV02.00".

▶ **Objekt 100B_{hex}**

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

Dieses Objekt ist nur lesbar. Es enthält die aktuell eingestellte Knotenadresse. Es ist nur das Low Byte gültig.

▶ **Objekt 100C_{hex}**

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.

► **Objekt 100D_{hex}**

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.

► **Objekt 100E_{hex}**

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 Protokolls.

► **Objekt 100F_{hex}**

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.

► **Objekt 1014_{hex}**

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

Dieses Objekt enthält den Identifier des EMCY-Telegramms.

► **Objekt 1400_{hex}**

Name	Index	Subindex	Datentyp	Defaultwert
1. Receive PDO Parameter	1400 _{hex}	00 _{hex}	U8	04 _{hex}
COD-ID used by PDO		01 _{hex}	U32	200 _{hex} + Adresse
Transmission Type		02 _{hex}	U8	FF _{hex}
Inhibit Time		03 _{hex}	U16	00 _{hex}
CMS-Priority Group		04 _{hex}	U8	02 _{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. Die Subindizes 03_{hex} und 04_{hex} werden intern nicht verwendet.

► **Objekt 1401_{hex}**

Name	Index	Subindex	Datentyp	Defaultwert
2. Receive PDO Parameter	1401 _{hex}	00 _{hex}	U8	04 _{hex}
COD-ID used by PDO		01 _{hex}	U32	300 _{hex} + Adresse
Transmission Type		02 _{hex}	U8	FF _{hex}
Inhibit Time		03 _{hex}	U16	00 _{hex}
CMS-Priority Group		04 _{hex}	U8	04 _{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. Die Subindizes 03_{hex} und 04_{hex} werden intern nicht verwendet.

► **Objekt 1600_{hex}**

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

Diese 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](#) auf Seite 48)

► **Objekt 1601_{hex}**

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 (vgl. Kap. 5.5.1).

► **Objekt 1800_{hex}**

Name	Index	Subindex	Datentyp	Defaultwert
1. Transmit PDO Parameter	1800 _{hex}	00 _{hex}	U8	04 _{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	01 _{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. Die Subindizes 03_{hex} und 04_{hex} werden intern nicht verwendet.

► **Objekt 1801_{hex}**

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}

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. Die Subindizes 03_{hex} und 04_{hex} werden intern nicht verwendet.

► **Objekt 1A00_{hex}**

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 48).

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 möglich. Außerdem ist auch das Node Guarding möglich.

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

- (1) automatischer Übergang von INITIALISIERUNG nach PRE-OPERATIONAL
- (2) Start_Remote_Node
- (3) Stop_Remote_Node
- (4) Enter_Pre-Operational_State
- (5) Reset_Node
- (6) Reset_Communication

6.2.2 Telegramme

NMT-Telegramme zur Kommunikationssteuerung besitzen nach dem Predefined Connection Set den Default-Identifier "0" (siehe auch [►Grundlagen CANopen◄](#) ab Seite 8).

Zustandssteuerung

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

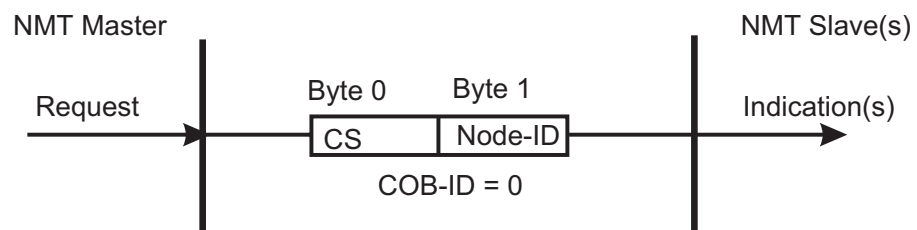


Abbildung 20: NMT-Telegramm zur Steuerung der Kommunikationszustände

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

Abbildung 21: Command Specifier und deren Wirkung im Slave

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}						

Abbildung 22: Telegrammaufbau Enter_Pre-Operational_State für Knoten 16

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

Wichtig: Der V-Regler sieht keinen Software-Reset vor. Um das NMT-Kommando `Reset_Node` auszuführen, müssen im V-Regler verschiedene Abläufe gestartet werden (vgl. V-Regler Dokumentation). Die CANopen-Optionskarte gibt diese Kommandos über den Bedarfsdatenkanal des V-Reglers vor. Diese Schritte werden in der Restzeit abgearbeitet. Die korrekte Abarbeitung der einzelnen Kommandos wird intern quittiert.

Schritt	Aktion	Betroffene V-Regler Parameter
1	Schnellhalt ausführen	P120 M Steuerwort
2	Vorbereiten zum Urlöschen des Datensatzes	P190 DSV Kommando P191 DSV Status
3	Urlöschen	P190 DSV Kommando
4	Laden und Speichern des Boot Datensatzes	P190 DSV Kommando P191 DSV Status
5	Freigabe der zyklischen und der Bedarfsdaten-kommunikation	P126 M Kommunikationsquelle
6	Speichern des Datensatzes	P190 DSV Kommando P191 DSV Status
7	Reset der kommunikationsspezif. Objekte	----
8	V-Regler Bootvorgang auslösen	P160 BS Auswahl P169 BS Wert

Nach einem Reset meldet sich der V-Regler mit dem Boot Up Telegramm (siehe [► Verarbeitungszeiten auf der Optionskarte ◄](#) auf Seite 21). Die gesamte Resetsequenz dauert vom Empfang des Kommandos `Reset_Node` bis zur Rückmeldung mittels Boot Up Telegramm ca. 10 Sekunden. Wurde ein Reset ausgeführt, steht die Kommunikationsquelle des Reglers

6.2 Netzwerkmanagement (NMT)

P126 auf 0x0C. Dies bedeutet, daß sowohl der Bedarfsdaten- als auch der Prozeßdatenkanal zwischen Optionskarte und Regler freigegeben ist.

Achtung! Wird im laufenden zyklischen Betrieb ein Reset ausgelöst, kann dies zu unerwünschten Zuständen in der Applikation führen, da der Regler neu gebootet und auf der CANopen-Optionskarte das Default-Mapping gesetzt wird.

Boot Up

Nach dem Einschalten des Slaves (Versorgungsspannung 24V liegt am V-Regler an) und erfolgreicher INITIALISIERUNG, sendet der V-Regler eine Boot Up Nachricht. Dieses Telegramm besitzt die COB-ID 80_{hex} + Slaveadresse und enthält keine Daten.

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

Abbildung 23: Boot Up Telegramm von Knoten 1

6.2.3 Node Guarding

Das Node Guarding dient der Überwachung der Slaves 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 0 Byte auf. Zur Unterscheidung zwischen Remote- und Datentelegramm (Telegrammunterscheidung erfolgt normalerweise durch die COB-ID), dient im DLC 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 lt. Predefined Connection Set. Diese COB-ID kann auch geändert werden. Das dafür erforderliche Objekt ist $100E_{\text{hex}}$.

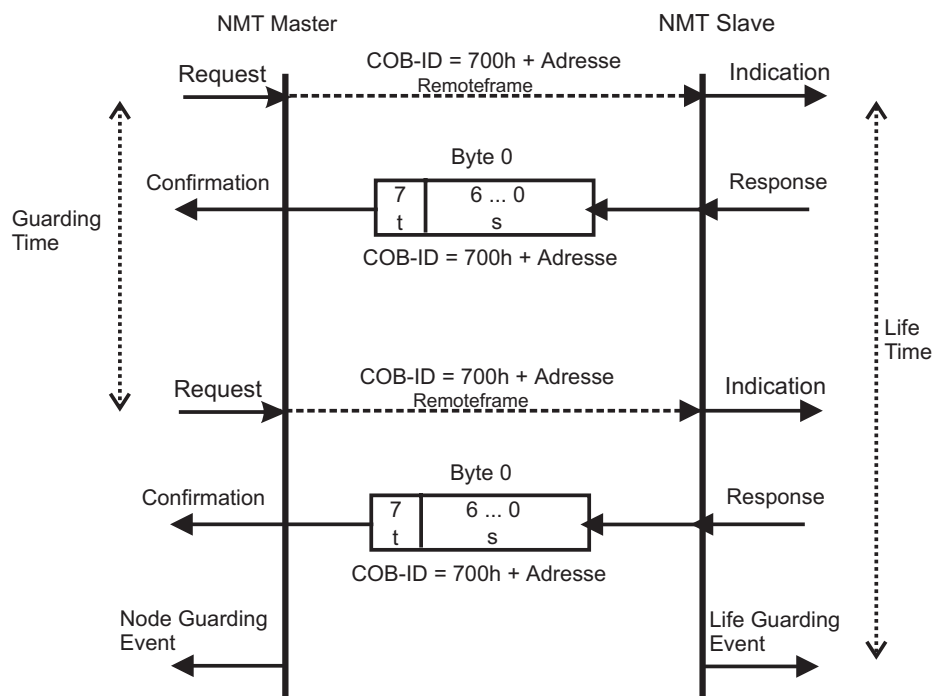


Abbildung 24: Node Guarding Protokoll

In den Objekten 100C_{hex} und 1000D_{hex} wird die Guarding-Zeit eingestellt. Innerhalb dieser Zeit muß der Slave vom Master eine Guarding-Anforderungen (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 auf den Parameter P120 M Steuerwort des V-Reglers wird ein im Objekt 3004_{hex} hinterlegtes Bitmuster geschrieben. Defaultmäßig lautet dieses Bitmuster: 8002h. Dies bedeutet "Schnellhalt" (vgl. V-Regler Dokumentation). Es kann auch jede beliebige andere Vorgabe für P120 M Steuerwort als Reaktion auf ein Life Guarding Event eingestellt werden.

Bleibt die Antwort des Slaves innerhalb einer bestimmten Zeit aus, wird im Master das Node Guarding Event ausgelöst.

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 [▶Abbildung 24◀](#) auf Seite 34). 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.

6.3 Bedarfsdaten (SDO)

Service Daten Objekte (SDO) dienen dem Austausch von Nachrichten ohne Echtzeitanforderungen. Deshalb sind im Predefined Connection Set (siehe auch [▶Grundlagen CANopen◀](#) ab Seite 8) 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 47).

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

6.3.1 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 – D4 (4 Byte).

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

Abbildung 25: SDO-Telegramm

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 7 Byte Nutzdaten übertragen werden können. Der Command Specifier CS klassifiziert die verschiedenen SDO-Typen.

6.3.2 Arten des SDO-Transfers

Die Baumüller CANopen-Anschaltung unterstützt den Expedited Transfer und den Segmented Transfer, wobei letzterer nur für die Objekte 1008_{hex} *Manufacturer Device Name*, 1009_{hex} *Manufacturer Hardware Version* und $100A_{\text{hex}}$ *Manufacturer Software Version* 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 $1XXX_{\text{hex}}$, $2XXX_{\text{hex}}$, $6XXX_{\text{hex}}$, $3XXX_{\text{hex}}$ sind über Expedited SDOs ansprechbar mit der Ausnahme von Objekt 1008_{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 $100A_{\text{hex}}$ nötig.

6.3.3 Objekt schreiben

Zum Schreiben von Objekten wird bei der Baumüller CANopen-Anbindung ausschließlich 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 der Response.

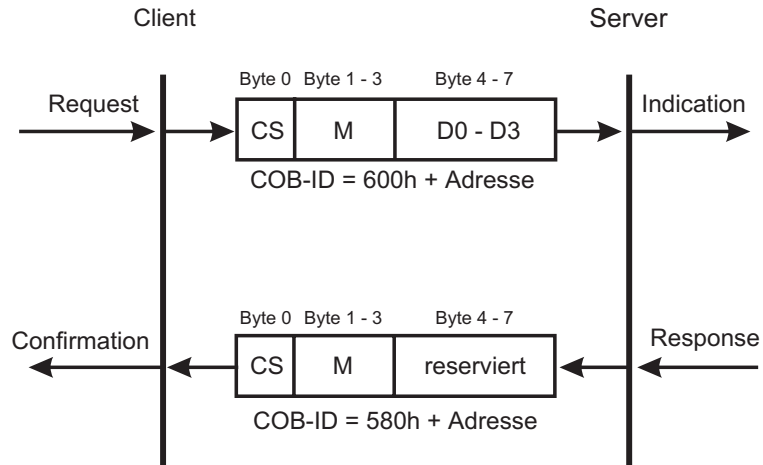


Abbildung 26: Initiate SDO Download Protocol

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

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

Abbildung 27: Aus verschiedenen Datenbreiten resultierende Command Specifier

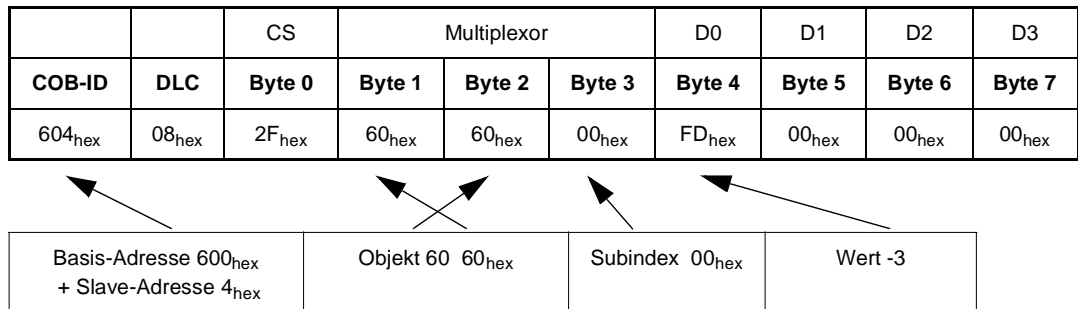
Der Command Specifier CS für die Response beträgt 60_{hex}, der Multiplexor ist identisch zu dem des Requests, das Datenfeld ohne Bedeutung (reserviert).

6.3 Bedarfsdaten (SDO)

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. Die Telegramme sind in der folgenden Abbildung dargestellt.

Request:



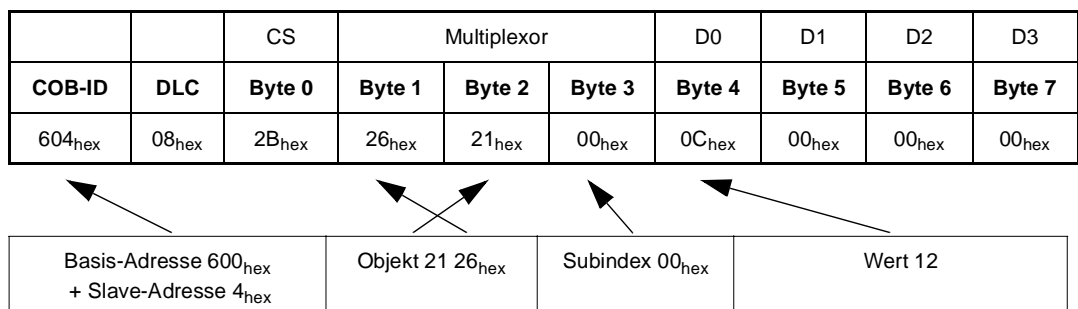
Response:

		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}	

Abbildung 28: Initiate SDO Download Protocol für 8 Bit Nutzdaten

Auf das Objekt 2126_{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. Die Telegramme sind in der folgenden Abbildung dargestellt.

Request:



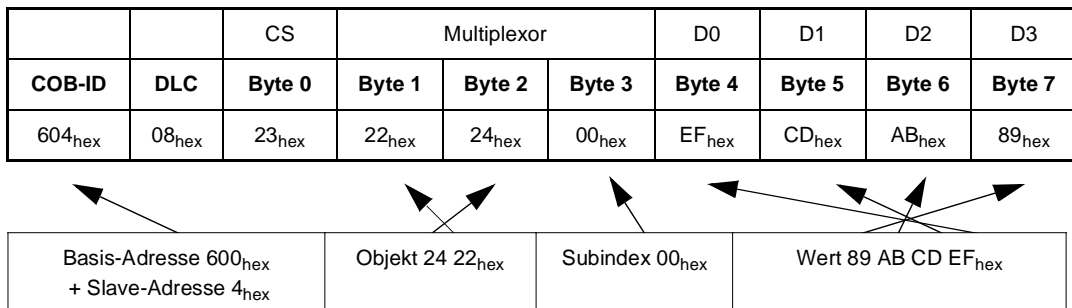
Response:

		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}	26 _{hex}	21 _{hex}	00 _{hex}	00 _{hex}	00 _{hex}	00 _{hex}	00 _{hex}	

Abbildung 29: Initiate SDO Download Protocol für 16 Bit Nutzdaten

Auf das Objekt 2422_{hex}, Subindex 00_{hex}, des Slaves mit der Adresse 4 soll der Wert "89ABCDEF_{hex}" geschrieben werden. Die Datenbreite dieses Objektes beträgt 32 Bit. Die Telegramme sind in der folgenden Abbildung dargestellt.

Request:



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}	22 _{hex}	24 _{hex}	00 _{hex}	00 _{hex}	00 _{hex}	00 _{hex}	00 _{hex}	

Abbildung 30: Initiate SDO Download Protocol für 32 Bit Nutzdaten

6.3.4 Objekt lesen

Zum Lesen von Objekten wird bei der Baumüller CANopen-Anbindung der Expedited, bei Objekt 1008_{hex} der Segmented Transfer benutzt.

Expedited Transfer

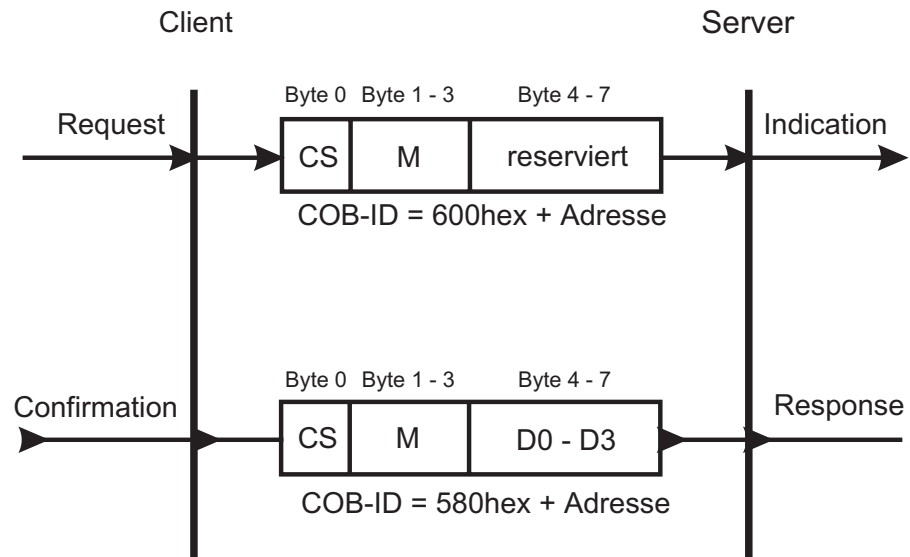


Abbildung 31: Initiate SDO Upload Expedited

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}

Abbildung 32: Aus verschiedenen Datenbreiten resultierende Command Specifier

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 acht Bit. Die Telegramme sind in der folgenden Abbildung dargestellt.

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:

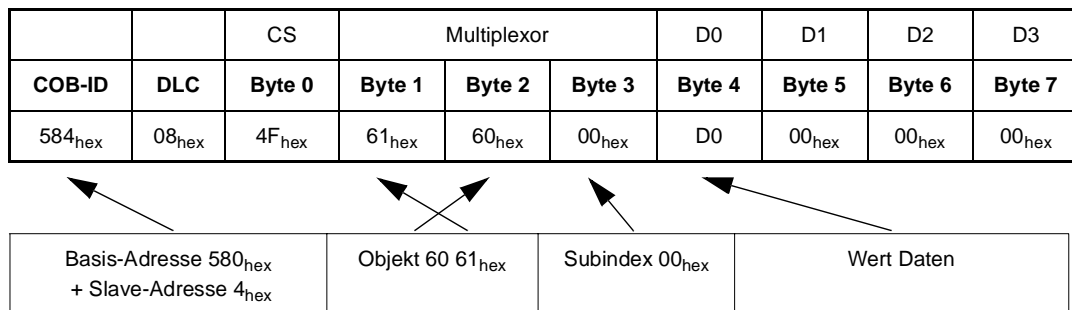


Abbildung 33: Initiate SDO Upload Protocol für 8 Bit Nutzdaten

Das Objekt 2126_{hex} , Subindex 00_{hex} , des Slaves mit der Adresse 4 soll gelesen werden. Die Datenbreite dieses Objektes beträgt 16 Bit. Die Telegramme sind in der folgenden Abbildung dargestellt.

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}	26_{hex}	21_{hex}	00_{hex}	00_{hex}	00_{hex}	00_{hex}	00_{hex}	

6.3 Bedarfsdaten (SDO)

Response:

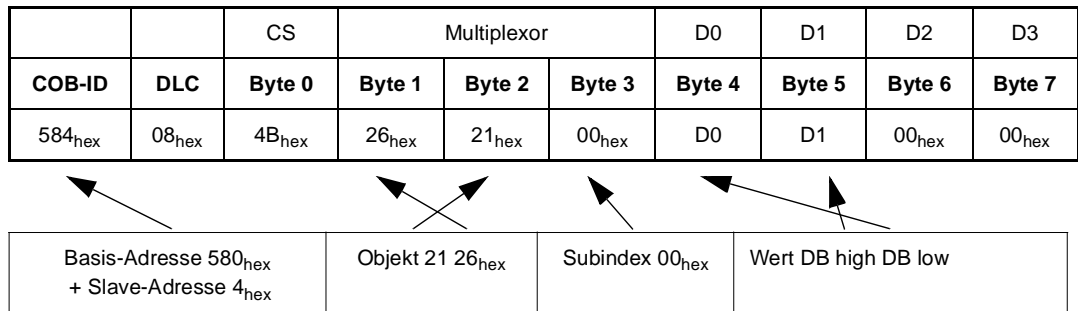
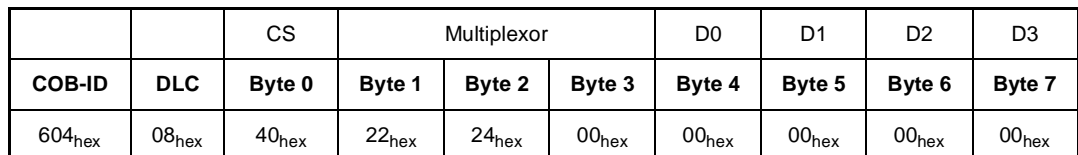


Abbildung 34: Initiate SDO Upload Protocol für 16 Bit Nutzdaten

Das Objekt 2422_{hex}, Subindex 00_{hex}, des Slaves mit der Adresse 4 soll gelesen werden. Die Datenbreite dieses Objektes beträgt 32 Bit. Die Telegramme sind in der folgenden Abbildung dargestellt.

Request:



Response:

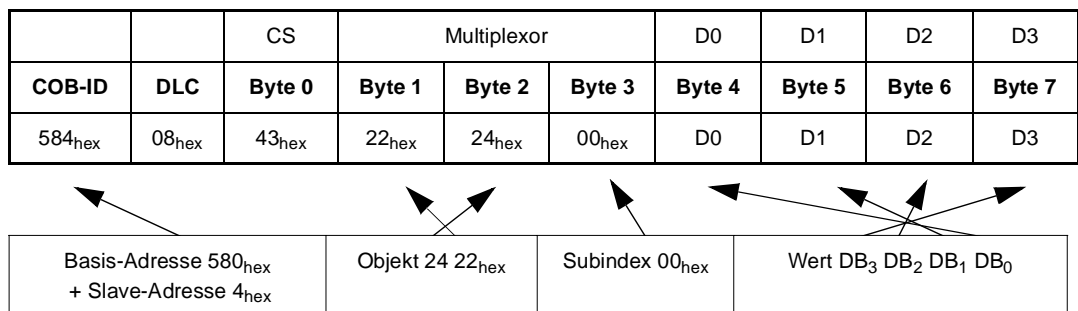


Abbildung 35: Initiate SDO Upload Protocol für 32 Bit Nutzdaten

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).

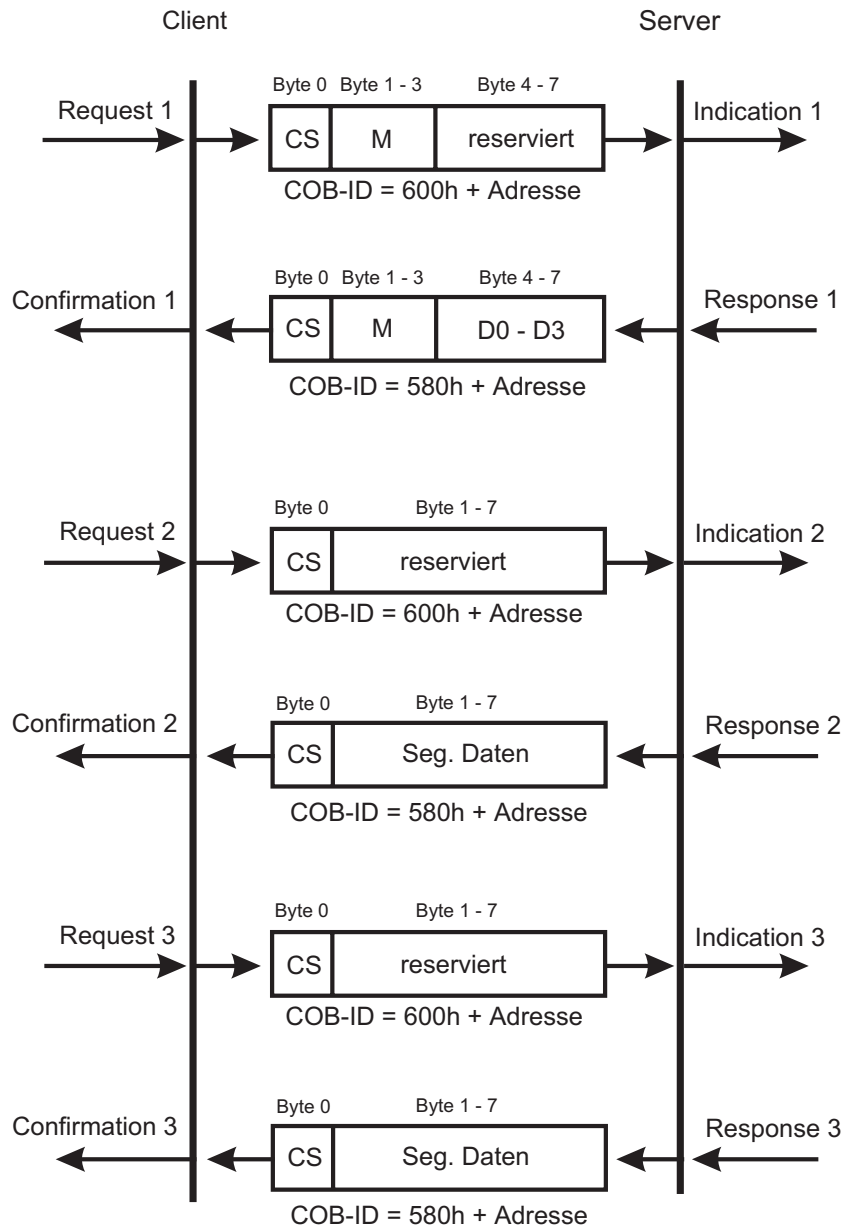


Abbildung 36: Upload SDO Segmented Protocol

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

6.3 Bedarfsdaten (SDO)

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}	08_{hex}	00_{hex}	00_{hex}	00_{hex}

Abbildung 37: Initiate SDO Upload für Objekt 1008_{hex}

Byte 0 in Response 1 (Command Specifier 41_{hex}) besagt, dass das Nutzdatenfeld die Anzahl der zu übertragenden Nutzdatenbytes enthält (08_{hex}).

Request 2:

		CS	Daten						
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	Daten						
COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
584_{hex}	08_{hex}	00_{hex}	42_{hex}	$4D_{\text{hex}}$	$5F_{\text{hex}}$	44_{hex}	52_{hex}	49_{hex}	56_{hex}

Abbildung 38: Erstes Segment beim Lesen von Objekt 1008_{hex}

Byte 0 in Request 2 (Command Specifier 60_{hex}) bedeutet, dass das erste Segment (die ersten sieben Byte) übertragen werden sollen. Byte 0 in Response 2 (Command Specifier, 00_{hex}) besagt, dass das gesamte Nutzdatenfeld (sieben Byte) gültige Daten enthält.

Request 3:

		CS	Daten						
COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
604 _{hex}	08 _{hex}	70 _{hex}	00 _{hex}	00 _{hex}	00 _{hex}	00 _{hex}	00 _{hex}	00 _{hex}	00 _{hex}

Response 3:

		CS	Daten						
COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
584 _{hex}	08 _{hex}	1D _{hex}	45 _{hex}	00 _{hex}	00 _{hex}	44 _{hex}	52 _{hex}	49 _{hex}	56 _{hex}

Abbildung 39: Letztes Segment beim Lesen von Objekt 1008h

Byte 0 in Request 3 (Command Specifier 70_{hex}) bedeutet, dass das nächste Segment (das achte Byte) übertragen werden soll. Byte 0 in Response 2 (Command Specifier 1D_{hex}) heißt, dass im Nutzdatenfeld (sieben Byte) nur das erste Byte gültige Daten enthält, und außerdem dieses Segment zugleich das letzte zu übertragende ist.

Das Ergebnis der Übertragung lautet: "BM_DRIVE".

6.3.5 Fehlerreaktionen

Fehlerhafte SDO-Zugriffe werden mit Abort Codes abgewiesen. Der Aufbau dieser Abort-Telegramme ist identisch zu dem in [▶Abbildung 25◀](#) auf Seite 36 dargestellten SDO-Telegramm. Das Datenfeld enthält einen vier Byte großen Abort Code.

Bei fehlerhaften Zugriffen auf kommunikationsspezifische Objekte (1XXXh) werden folgende Meldungen unterschieden:

Abort Code	Bedeutung
05 _{hex} 03 _{hex} 00 _{hex} 00 _{hex}	Parameter inkonsistent (Toggle Bit hat nicht gewechselt)
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} 07 _{hex} 00 _{hex} 00 _{hex}	Zugriff wegen Hardwarefehler fehlgeschlagen
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} 22 _{hex}	Daten können aufgrund des derzeitigen Kommunikationszustandes nicht gemappt werden (z.B. Mapping ändern im Zustand OPERATIONAL)

Abbildung 40: Abbruch Meldungen bei SDO-Zugriffen

6.3 Bedarfsdaten (SDO)

Fehlerhafte Zugriffe auf alle anderen Objekte ($3XXX_{\text{hex}}$, $2XXX_{\text{hex}}$ und $6XXX_{\text{hex}}$) werden global mit dem Code $06_{\text{hex}} 02_{\text{hex}} 00_{\text{hex}} 00_{\text{hex}}$ Objekt existiert nicht abgelehnt.

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}	

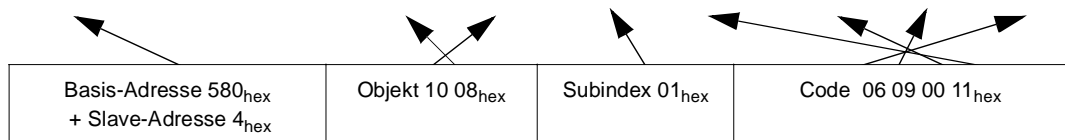


Abbildung 41: Abort Telegramm

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.

6.4 Synchronisation (SYNC)

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

Der Empfang eines SYNC-Telegramms mit dem im Objekt 1005_{hex} eingestellten Identifier erzeugt auf der CANopen-Anschaltung des V-Reglers einen Interrupt, der an den Regler weitergeleitet wird. Auf dieses Signal kann sich das Betriebssystem des V-Reglers synchronisieren. Die Voraussetzung hierfür ist das Einhalten des maximalen Jitters (vgl. V-Regler Dokumentation). Die Zykluszeit der SYNC-Telegramme kann nur die im V-Regler Parameter *P167 BS Sync.-Slot* einstellbaren Werte annehmen. Die SYNC-Zykluszeit muß mit der in *P167 BS Sync.-Slot* eingestellten Zeit übereinstimmen (vgl. V-Regler Dokumentation). Innerhalb eines SYNC-Intervalls müssen alle entsprechenden Telegramme an alle projizierten Slaves gesendet werden. Übertragungsrate, Leitungslänge, Anzahl der Knoten, Größe der Telegramme sowie Verarbeitungszeiten auf der CANopen-Optionskarte sind dabei zu berücksichtigen.

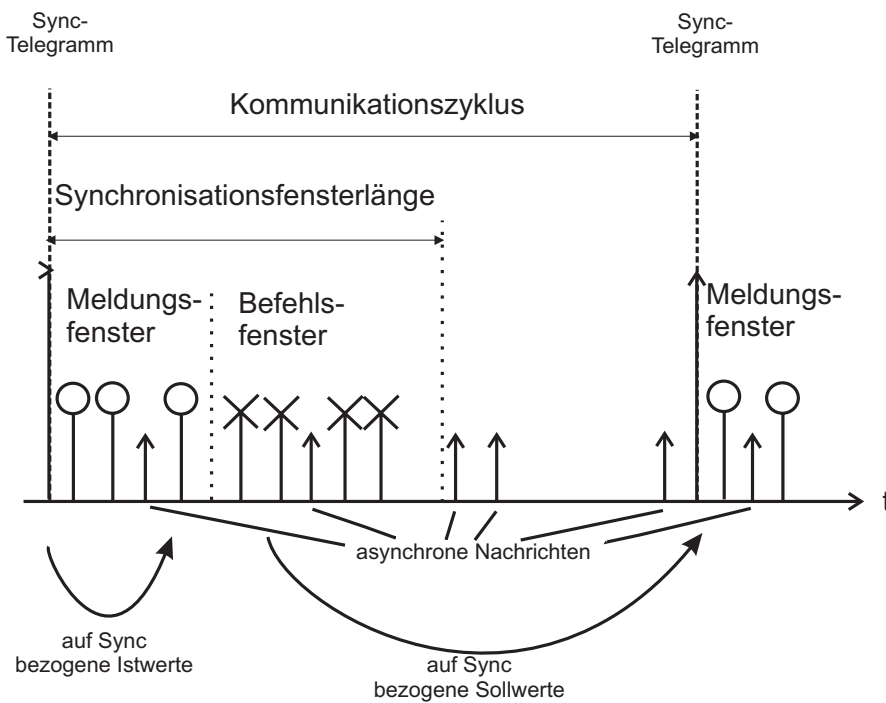


Abbildung 42: Kommunikationszyklus

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 50). Asynchrone Nachrichten (SDOs, PDOs, NMT) dürfen jederzeit auftreten, synchrone Telegramme nur innerhalb der Synchronisationsfensterlänge.

6.5 Prozessdaten (PDO)

Prozessdaten-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 50).

Wichtig: Alle in den PDOs konfigurierten Objekte werden im BAPS-Protokoll zwischen Baumüller CANopen-Optionskarte und V-Regler als zyklische Daten übertragen (siehe auch [►Kommunikationsablauf auf der Optionskarte◄](#) auf Seite 19). Da die zyklische Datenübertragung nur im Zustand OPERATIONAL stattfindet, darf auch nur in diesem Zustand die Kommunikationsüberwachung *P127 M Komm.-Überwachung* und *P128 M Überwachungszeit* aktiviert sein (vgl. V-Regler Dokumentation).

6.5.1 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 projektiert. Das Mapping wird über im Objektverzeichnis adressierbare Objekte eingestellt. Dafür werden SDOs verwendet. Für jede PDO existieren zwei solcher Objekte (siehe auch [►Verzeichnis der Objekte zur Kommunikationssteuerung◄](#) ab Seite 23). Ein Objekt bestimmt den Inhalt der PDO, das zweite die Kommunikationsbeziehung bzw. Triggerung.

Prozess Daten Objekt	Objekt für Inhalt	Objekt für Kommunikationsbeziehung
TX-PDO1	1A00 _{hex}	1800 _{hex}
TX-PDO2	1A01 _{hex}	1801 _{hex}
RX-PDO1	1600 _{hex}	1400 _{hex}
RX-PDO2	1601 _{hex}	1401 _{hex}

Wichtig: 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. Das Prinzip ist in [►Abbildung 43◄](#) auf Seite 49 dargestellt. 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}, 1A00_{hex}, 1A01_{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 erste 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 V-Regler-Parameter *P120 M Steuerwort* umgesetzt (vgl. Anhang). Damit wird das erste Wort des in RX-PDO1 empfangenen Telegramms auf das Steuerwort des V-Reglers geschrieben. Ein ausführliches Beispiel für das Mapping ist in [►Beispiel für das PDO-Mapping◄](#) ab Seite 51 erläutert.

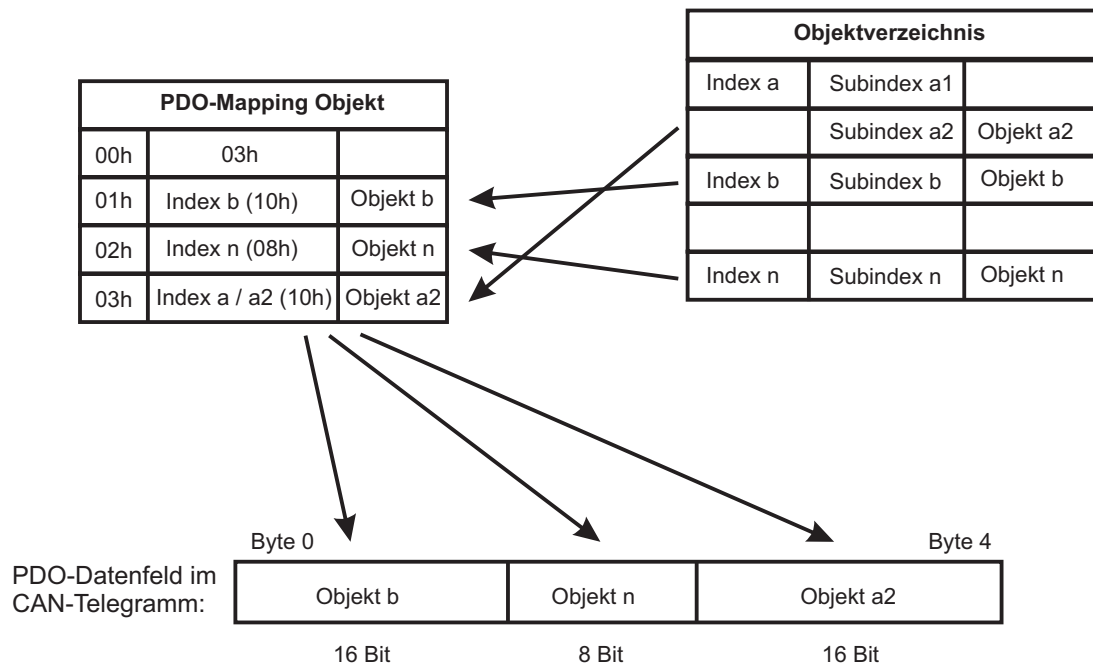


Abbildung 43: PDO-Mapping

Das Default-Mapping ist in [Verzeichnis der Objekte zur Kommunikationssteuerung](#) auf Seite 23 beschrieben.

Um ein Mapping zu ändern, können die Werte überschrieben werden. Um ein Mapping zu löschen es ist der Wert "0" auf den Subindex 00_{hex} des entsprechenden Mapping-Objektes (1600_{hex}, 1601_{hex}, 1A00_{hex}, 1A01_{hex}) zu schreiben.

Wichtig: Bei der Einstellung des Mappings in den Objekten (1600_{hex}, 1601_{hex}, 1A00_{hex}, 1A01_{hex}) ist jeweils der Subindex 00_{hex} mit der richtige Anzahl der gemappten Objekte als letztes zu beschreiben.

Dummy-Mapping

Die Baumüller CANopen-Anschaltung stellt ein 16 Bit Dummy-Objekt bereit, welches ebenfalls in eine PDO gemappt werden kann. Dieses Objekt hat den Index 0006_{hex}. Das Dummy-Objekt dient als Platzhalter, um nur bestimmte Objekte innerhalb eines CAN-Telegramms zu verwenden (siehe auch [Beispiel für das PDO-Mapping](#) auf Seite 51).

Wichtig: Das aktuell eingestellte Mapping geht nach dem Ausschalten verloren. Ausschalten ist gleichbedeutend mit einem Reset. Danach stellt sich das Default-Mapping ein.

6.5.2 Kommunikationsbeziehung über PDO

Für den Aufbau von Kommunikationsbeziehungen und zur Kommunikationssteuerung über PDOs existiert zu jeder PDO ein Objekt. Der Objektindex hat einen Offset von -200_{hex} zum entsprechenden Mapping-Objekt.

Prozess Daten Objekt	Mapping-Objekt	Steuer-Objekt
RX-PDO1	1600_{hex}	1400_{hex}
RX-PDO2	1601_{hex}	1401_{hex}
TX-PDO1	$1A00_{\text{hex}}$	1800_{hex}
TX-PDO2	$1A01_{\text{hex}}$	1801_{hex}

Der Aufbau dieser Objekte ist in [► Verzeichnis der Objekte zur Kommunikationssteuerung ◀](#) auf Seite 23 beschrieben.

Das Kriterium für die Übernahme einer auf dem CAN-Bus übertragenen Nachricht ist die passende COB-ID. Die COB-ID wird in den Steuer-Objekten (1400_{hex} , 1401_{hex} , 1800_{hex} , 1801_{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 Telegramm-Puffer übernommen.

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} , 1401_{hex} , 1800_{hex} , 1801_{hex} jeweils im Subindex 02_{hex} eingestellt.

Synchrone PDO:

Das Senden und Empfangen ist an das SYNC-Telegramm gebunden (siehe [► Synchronisation \(SYNC\) ◀](#) ab Seite 47).

Asynchrone PDO:

Das Senden und Empfangen ist an bestimmte Ereignisse gebunden.

Wert in Subindex 02_{hex}	Typ	Wirkung	
		TX-PDO 1800_{hex} , 1801_{hex}	RX-PDO 1400_{hex} , 1401_{hex}
00_{hex} , 01_{hex} (0, 1)	Synchron	Senden erfolgt nach jedem empfangenen SYNC-Telegramm	Vor dem letzten SYNC-Telegramm empfangene PDO mit passender COB-ID wird übernommen
$02_{\text{hex}} - F0_{\text{hex}}$ (2 – 240)	Synchron	Senden erfolgt nach Empfang der eingestellten Anzahl von SYNC-Telegramm	Vor dem letzten SYNC-Telegramm empfangene PDO mit passender COB-ID wird übernommen
FE_{hex} (254)	Asynchron	Senden erfolgt zeitgesteuert	Jede PDO mit passender COB-ID wird übernommen
FF_{hex} (255)	Asynchron	Senden erfolgt ereignisgesteuert	Jede PDO mit passender COB-ID wird übernommen

Zeitgesteuertes Senden bedeutet, die Sendebedingung ist an einen Timer gebunden. Dieser Timer wird mittels Objekt 3003_{hex} (16 Bit) eingestellt. Die Auflösung beträgt Millisekunden. Der Timer wird beim Zustandswechsel nach OPERATIONAL gestartet. Das Senden der entsprechenden TX-PDO erfolgt dann zyklisch mit der in Objekt 3003_{hex} eingestellten Zeit. Der Timer wird gelöscht, indem auf Objekt 3003_{hex} Subindex 00_{hex} der Wert "0" geschrieben wird. Das Beschreiben von Objekt 3003_{hex} kann in jedem Kommunikationszustand erfolgen. Der Wert wird sofort wirksam.

Wichtig: Der Timer gilt für beide PDOs gleichzeitig. Sind beide TX-PDOs als zeitgesteuert parametrierbar, werden beide mit der selben Zykluszeit laut Objekt 3003_{hex} gesendet.

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

Ereignisgesteuertes Senden bedeutet, die Sendebedingung ist an die Änderung eines Wertes eines der gemappten Objekte gebunden. Sind beispielweise 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, alle PDOs mit passender COB-ID werden übernommen.

6.5.3 Beispiel für das PDO-Mapping

Die V-Regler mit den CANopen-Adresse 2 und 7 sollen immer identische Drehzahl-Sollwerte aufweisen. Der Parameter *P006 HLG Eingang 3* des V-Reglers 7 soll dazu mit dem Wert des Parameters *P004 HLG Eingang 2 des V-Reglers 2* übereinstimmen. Weiterhin soll der CANopen-Master über das Statuswort *P121 M Statuswort* und die Ist-Betriebsart *P123 M Ist-Betriebsart* von V-Regler 2 informiert werden. Das Senden soll ereignisgesteuert mit TX-PDO2 erfolgen. Der Slave 7 empfängt das Telegramm in RX-PDO1.

1. Schritt: Bestimmen der notwendigen Objekte

Aus der Objektliste (vgl. Anhang und Kap. 5.1) sind die entsprechenden Objekte des Objektverzeichnisses aus den Parameternummern zu ermitteln.

Der V-Regler mit der CANopen-Adresse 2 sendet folgende Parameter, die mit den angegebenen Objekten korrespondieren:

P121 M Statuswort ↔ 6041_{hex} Statuswort

P004 HLG Eingang 2 ↔ 2004_{hex} Drehzahl-Sollwert

P123 M Ist-Betriebsart ↔ 6061_{hex} Ist-Betriebsart

Der V-Regler mit der CANopen-Adresse 7 empfängt das PDO-Telegramm und schreibt den entsprechenden Wert auf folgenden Parameter, der mit dem angegebenen Objekt korrespondiert:

P006 HLG Eingang 2 ↔ 2006_{hex} Drehzahl-Sollwert

Zum Einstellen des Mappings sind folgende Objekte nötig:

Knoten 2: 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)

2. Schritt: Mapping konfigurieren

Zum Einstellen des Mappings werden die SDOs des Expedited Transfers (siehe auch [Bedarfsdaten \(SDO\)](#) auf Seite 35) verwendet. Diese können über einen Master, einen Buskonfigurator o.ä. initiiert werden.

Mapping für Slave 2:

Schreiben des ersten zu mappenden Objektes mit Index (6041_{hex}), Subindex (00_{hex}) und Länge (10_{hex}) auf $1A01_{\text{hex}}$ Subindex 01_{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}	$1A_{\text{hex}}$	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}	$1A_{\text{hex}}$	01_{hex}	00_{hex}	00_{hex}	00_{hex}	00_{hex}	

Abbildung 44: Erstes zu mappendes Objekt

Schreiben des zweiten zu mappenden Objektes mit Index (2004_{hex}), Subindex (00_{hex}) und Länge (10_{hex}) auf $1A01_{\text{hex}}$ 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}	$1A_{\text{hex}}$	02_{hex}	10_{hex}	00_{hex}	04_{hex}	20_{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}	$1A_{\text{hex}}$	02_{hex}	00_{hex}	00_{hex}	00_{hex}	00_{hex}	

Abbildung 45: Zweites zu mappendes Objekt

Schreiben des dritten zu mappenden Objektes mit Index (6061_{hex}), Subindex (00_{hex}) und Länge (08_{hex}) auf 1A01_{hex} Subindex 03_{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}	03 _{hex}	08 _{hex}	00 _{hex}	61 _{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}	03 _{hex}	00 _{hex}	00 _{hex}	00 _{hex}	00 _{hex}	

Abbildung 46: Drittes zu mappendes Objekt

Schreiben der Anzahl der gemappten Objekte (03_{hex}) auf 1A01_{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	
602 _{hex}	08 _{hex}	2F _{hex}	01 _{hex}	1A _{hex}	00 _{hex}	03 _{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}	

Abbildung 47: Anzahl der zu mappenden Objekte

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

1A01_{hex}	00_{hex}	03 _{hex}
	01_{hex}	60410010 _{hex}
	02_{hex}	20040008 _{hex}
	03_{hex}	60610010 _{hex}

Mapping für Slave 7:

Slave 7 soll nur den Drehzahl-Sollwert auswerten. Das Steuerwort ist an erste Stelle gemappt, der Drehzahl-Sollwert an zweite Stelle. Daher muß für die erste Stelle das Dummy-Objekt verwendet werden.

Schreiben des ersten zu mappenden Objektes mit Index (0006_{hex}), Subindex (00_{hex}) und Länge (10_{hex}) auf 1600_{hex} Subindex 01_{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}	1A _{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} x	60 _{hex}	00 _{hex}	1A _{hex}	01 _{hex}	00 _{hex}	00 _{hex}	00 _{hex}	00 _{hex}	

Abbildung 48: Erstes zu mappendes Objekt

Schreiben des zweiten zu mappenden Objektes mit Index (2006_{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}	06 _{hex}	20 _{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}	

Abbildung 49: Zweites zu mappendes Objekt

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}	

Abbildung 50: Anzahl der zu mappenden Objekte

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

1600 _{hex}	00 _{hex}	02 _{hex}
	01 _{hex}	00060010 _{hex}
	02 _{hex}	20060010 _{hex}

Der Datenaustausch zwischen den V-Reglern über die PDOs ist in der folgenden Abbildung dargestellt.

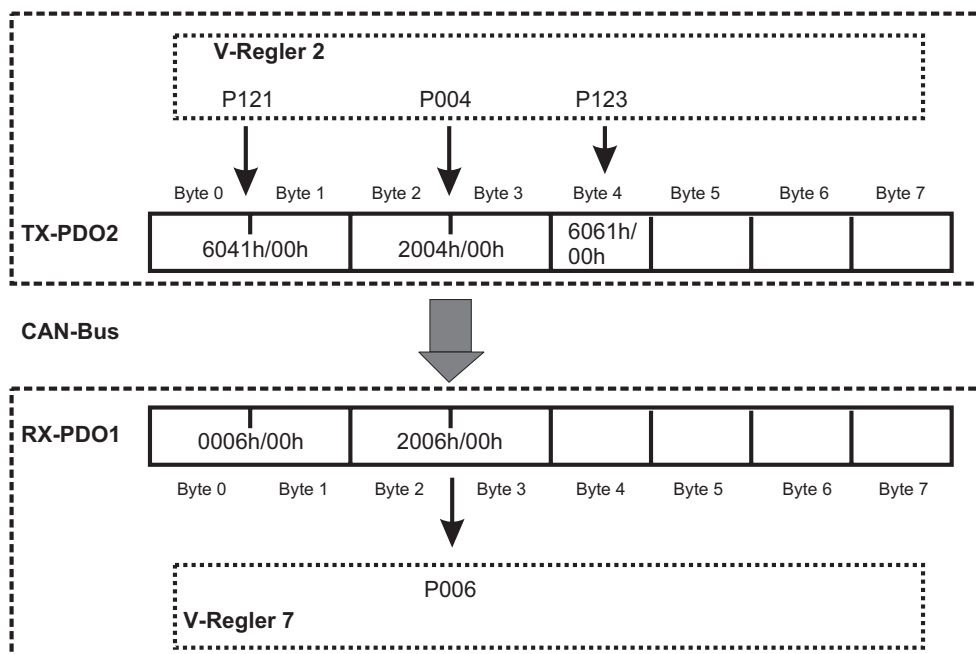


Abbildung 51: Telegrammaufbau für Mapping

Von der TX-PDO2 des V-Reglers 2 werden im V-Regler 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 übereinstimmen. Die COB-ID der TX-PDO2 von Slave 2 ist defaultmäßig auf 282_{hex} eingestellt. Die COB-ID der RX-PDO1 von Slave 7 lautet defaultmäßig 207_{h} . 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 1801_{hex} Subindex 01_{hex} wird der Wert 207_{hex} eingetragen. Auf Subindex 02_{hex} wird der Wert " FF_{hex} " (ereignisgetriggert) geschrieben.

1801_{hex}		
	01_{hex}	207_{hex}
	02_{hex}	FF_{hex}

Kommunikationsparameter für Slave 7:

Auf Objekt 1400_{hex} Subindex 01_{hex} wird der Wert 207_{hex} eingetragen. Auf Subindex 02_{hex} wird der Wert " FF_{hex} " (ereignisgetriggert) geschrieben.

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

6.5.4 Eintrag in die BAPS

Wie in [▶Zeitscheibenverfahren](#) auf Seite 20 beschrieben können maximal acht zyklische Soll- und acht zyklische Istwerte zwischen Optionskarte und V-Regler ausgetauscht werden. Die Aktualisierungszeiten der einzelnen Werte hängen von ihrer Position in der BAPS-Konfiguration ab.

Die Soll / Istwerte können bei CANopen auf je zwei PDOs verteilt sein. Bei vier Sollwerten je TX-PDO ergibt sich folgende BAPS-Konfiguration, vorausgesetzt, die PDOs enthalten kein Dummy-Objekt und kein Steuer- oder Statuswort:

BAPS-Position	PDO	PDO-Position	Aktualisierungszeit
1	TX-PDO1	1. Objekt	1 ms
2	TX-PDO2	1. Objekt	2 ms
3	TX-PDO1	2. Objekt	4 ms
4	TX-PDO2	2. Objekt	8 ms
5	TX-PDO1	3. Objekt	16 ms
6	TX-PDO2	3. Objekt	32 ms
7	TX-PDO1	4. Objekt	64 ms
8	TX-PDO2	4. Objekt	128 ms

Für RX-PDOs gilt das selbe Verfahren.

Die Einträge erfolgen abwechselnd und lückefrei, mit dem 1. Objekt von PDO1 beginnend.

Wichtig: Steuer- und Statuswort werden aus dem Mapping herausgelöst und auf der BAPS gesondert behandelt. Steuer- und Statuswort werden jede Millisekunde aktualisiert. Das Dummy-Objekt wird nicht berücksichtigt.

Wichtig: Beginnend beim ersten Objekt von PDO 1 werden die Inhalte der PDOs abwechselnd auf ihre Gültigkeit für die BAPS-Konfiguration (kein Dummy und kein Steuer- oder Statuswort) abgefragt. Ist das Objekt gültig, dann wird dieses an die nächste freie Stelle der BAPS-Konfiguration eingetragen. Ist es ungültig, wird das entsprechende Objekt der anderen PDO überprüft, usw. Ungültige Objekte sind im Falle der Konfiguration der zyklischen BAPS: Steuer- und Statuswort, Dummy-Objekte und nicht vorhandene Objekte/Parameter.

Beispiel:

TX-PDO1 enthält folgendes Mapping:

1A00_{hex}	00_{hex}	04 _{hex}	
	01_{hex}	60410010 _{hex}	Statuswort
	02_{hex}	20040010 _{hex}	HLG-Eingang 2
	03_{hex}	60610010 _{hex}	Ist-Betriebsart
	04_{hex}	20530010 _{hex}	M-Grenze

TX-PDO2 enthält folgendes Mapping:

1A01_{hex}	00_{hex}	03 _{hex}	
	01_{hex}	20060010	HLG-Eingang 3
	02_{hex}	00060010 _{hex}	Dummy
	03_{hex}	20250010 _{hex}	Geber 1 Status

Die daraus resultierende BAPS-Konfiguration:

BAPS-Position	PDO	PDO-Position	Aktualisierungszeit
1	TX-PDO 2	2006 _{hex} (P006)	1 ms
2	TX-PDO 1	2004 _{hex} (P004)	2 ms
3	TX-PDO 1	6061 _{hex} (P123)	4 ms
4	TX-PDO 2	2025 _{hex} (P025)	8 ms
5	TX-PDO 1	2053 _{hex} (P053)	16 ms
6	--	--	32 ms
7	--	--	64 ms
8	--	--	128 ms

Der erste Eintrag in der BAPS-Konfiguration enthält das erste Objekt von TX-PDO2, da das erste Objekt von TX-PDO1 das Statuswort ist und somit aus dem Mapping herausgelöst wird.

Wichtig: Ist das PDO-Mapping fehlerhaft (falsche Parameternummern o.ä.), wird keine zyklische Kommunikation zwischen Optionskarte und V-Regler gestartet.

Wichtig: Wird in einer oder beiden PDOs mehrmals die selbe Objektnummer gemappt, so erscheint das Objekt in der entsprechenden Anzahl in der BAPS-Konfiguration. Ausnahme: Status- und Steuerwort

7

FEHLERBEHANDLUNG

7.1 Problembehebung

Problem	Hinweise zur Beseitigung
CANopen-Slave meldet sich nicht am Netz	<ul style="list-style-type: none">• Verkabelung überprüfen• Adresse an DIP-Schalter kontrollieren• Kommunikationsquelle P126 für zyklische Daten und Bedarfsdaten freigeben
Slave sendet Errorframes	<ul style="list-style-type: none">• Baudrate überprüfen• Verkabelung kontrollieren• Abschlußstecker überprüfen
Keine zyklische Kommunikation	<ul style="list-style-type: none">• Kommunikationsquelle P126 freigeben• PDO-Mapping kontrollieren
V-Regler läßt sich nicht in Zustand 4 schalten	<ul style="list-style-type: none">• Timeout-Überwachung der Kommunikation P127 / P128 überprüfen

7.2 Fehlertelegramm (EMCY)

Emergency-Telegramme dienen der Anzeige von V-Regler-Fehlern. Dieses Telegramm wird gesendet, sobald der V-Regler einen internen Fehler erkannt hat. Bei jedem neu hinzukommenden Fehler wird einmalig ein Emergency-Telegramm gesendet. Eine Telegrammwiederholung erfolgt nicht.

7.2.1 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				

Abbildung 52: Aufbau des Emergency-Telegramms

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

Der Emergency Error Code (Byte 0, 1) ist in CANopen definiert. Die Umsetzung auf V-Regler-Fehlernummern wird in [Umsetzung der V-Regler-Fehlermeldungen auf CANopen-Fehlermeldungen](#) ab Seite 61 dargestellt.

Das Error Register entspricht dem Inhalt von Objekt 1001_{hex} (siehe auch [Verzeichnis der Objekte zur Kommunikationssteuerung](#) auf Seite 23).

Die ersten beiden Bytes des herstellerspezifischen Fehlerfeldes enthalten die V-Regler-Fehlernummer aus Parameter *P124 M Fehler-Code* (vgl. V-Regler Dokumentation).

Beispiel:

Slave 5 hat einen Geberfehler an Geber 2 erkannt (Absolutlage unbekannt). 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}	20 _{hex}	73 _{hex}	81 _{hex}	04 _{hex}	03 _{hex}	00 _{hex}	00 _{hex}	00 _{hex}

Abbildung 53: EMCY-Telegramm für Geberfehler 0304_{hex}

Stehen mehrere Fehler an und wird ein Fehler gelöscht, sendet der Baumüller 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}

Abbildung 54: EMCY-Telegramm für Error Reset / No Error

7.2.2 Umsetzung der V-Regler-Fehlermeldungen auf CANopen-Fehlermeldungen

Die Beschreibung der V-Regler Fehlermeldungen und Hinweise zur Beseitigung der Störungen sind der V-Regler Dokumentation zu entnehmen.

V-Regler Fehler-Code	Bezeichnung lt. V-Regler	CANopen Fehler-Code
0001 _{hex}	Timeout BASS-Protokoll	8100 _{hex}
0002 _{hex}	Timeout USS-Protokoll	8100 _{hex}
0003 _{hex}	Timeout DP-RAM zyklisch	8100 _{hex}
0004 _{hex}	Timeout DP-RAM Bedarfsdaten	8100 _{hex}
0005 _{hex}	System-Boot-Vorgang	5530 _{hex}
0006 _{hex}	Timeout Fehlerreaktion	8100 _{hex}
0010 _{hex}	Fehler Switch	6100 _{hex}
0110 _{hex}	Störung Einspeise-Einheit	5100 _{hex}
0201 _{hex}	Überspannung Uz _k	5120 _{hex}
0202 _{hex}	Überstrom	2310 _{hex}
0203 _{hex}	Fehlerstrom	2300 _{hex}
0204 _{hex}	Störung Hilfsspannung	3000 _{hex}
0205 _{hex}	Übertemperatur Leistungsteil	4210 _{hex}
0206 _{hex}	Störung Sicherheitsrelais	5441 _{hex}
0207 _{hex}	Transistorfehler	5410 _{hex}
0208 _{hex}	Phase U oben	5410 _{hex}
0209 _{hex}	Phase U unten	5410 _{hex}
020A _{hex}	Phase V oben	5410 _{hex}
020B _{hex}	Phase V unten	5410 _{hex}
020C _{hex}	Phase W oben	5410 _{hex}
020D _{hex}	Phase W unten	5410 _{hex}
020E _{hex}	Leistungsteilkennung unbekannt	5410 _{hex}
020F _{hex}	Falscher Leistungsteiltyp	5410 _{hex}
0210 _{hex}	Störung Leistungsteil	5400 _{hex}
0301 _{hex}	Überdrehzahl Geber 1	7310 _{hex}
0302 _{hex}	Überdrehzahl Geber 2	7310 _{hex}
0303 _{hex}	Absolutlage Geber 1 unbekannt	7320 _{hex}
0304 _{hex}	Absolutlage Geber 2 unbekannt	7320 _{hex}
0401 _{hex}	I ² t-Überwachung Motor	7120 _{hex}
0501 _{hex}	Übertemperatur Motor	7120 _{hex}
0502 _{hex}	Kurzschluß Temperatursensor	7120 _{hex}
0601 _{hex}	Schleppfehler dynamisch	7120 _{hex}
0602 _{hex}	Schleppfehler statisch	8611 _{hex}
0702 _{hex}	Blockierüberwachung	7121 _{hex}
0801 _{hex}	Modul Code ungültig	7300 _{hex}

7.2 Fehlertelegramm (EMCY)

V-Regler Fehler-Code	Bezeichnung lt. V-Regler	CANopen Fehler-Code
0802 _{hex}	Falsches Adaptermodul	7300 _{hex}
0803 _{hex}	Keine Kommunikation mit Geber	7300 _{hex}
0804 _{hex}	Leistungsbruch Geber 1	7300 _{hex}
0805 _{hex}	Falsche Adresse	7300 _{hex}
0806 _{hex}	Geber meldet Fehler	7300 _{hex}
0807 _{hex}	Falsche Kommando	7300 _{hex}
0808 _{hex}	Falsche Checksumme	7300 _{hex}
0809 _{hex}	Fehler Lagekorrektur	7300 _{hex}
080A _{hex}	Unbekannter Gebercode	7300 _{hex}
080B _{hex}	Timeout Fehler Kommunikation	7300 _{hex}
0901 _{hex}	Kopierfehler EEPROM	5530 _{hex}
0902 _{hex}	Boot-Datensatz fehlt	5530 _{hex}
0903 _{hex}	Checksummen-Fehler im Boot-Datensatz	5530 _{hex}
0A01 _{hex}	Modul-Code ungültig	7300 _{hex}
0A02 _{hex}	Falsches Adaptermodul	7300 _{hex}
0A03 _{hex}	Keine Kommunikation mit Geber	7300 _{hex}
0A04 _{hex}	Leistungsbruch Geber 2	7300 _{hex}
0A05 _{hex}	Falsche Adresse	7300 _{hex}
0A06 _{hex}	Geber meldet Fehler	7300 _{hex}
0A07 _{hex}	Falsche Kommando	7300 _{hex}
0A08 _{hex}	Falsche Checksumme	7300 _{hex}
0A09 _{hex}	Fehler Lagekorrektur	7300 _{hex}
0A0A _{hex}	Unbekannter Gebercode	7300 _{hex}
0A0B _{hex}	Timeout Fehler Kommunikation	7300 _{hex}
0A04 _{hex}	Leistungsbruch Geber 2	7300 _{hex}
0B01 _{hex}	Hauptprogramm-Rechenzeit Überschreitung	7400 _{hex}
0B02 _{hex}	Task-Rechenzeit Überschreitung	7400 _{hex}
0B03 _{hex}	Sync. IR Rechenzeit Überschreitung	7400 _{hex}
0B04 _{hex}	DSP-Rechenzeit Überschreitung	7400 _{hex}
0B05 _{hex}	Fehler beim Linken	7400 _{hex}
0B06 _{hex}	Fehler Zeitscheibensystem	7400 _{hex}
0C01 _{hex}	Illegaler externer Buszugriff	7400 _{hex}
0C02 _{hex}	Illegaler Befehlszugriff	7400 _{hex}
0C03 _{hex}	Illegaler Wort-Operandzugriff	7400 _{hex}
0C04 _{hex}	Schutzfehler	7400 _{hex}
0C05 _{hex}	Undefinierter Operationscode	7400 _{hex}
0C06 _{hex}	Stack-Bereichsunterschreitung	7400 _{hex}
0C07 _{hex}	Stack-Bereichsüberschreitung	7400 _{hex}
0C08 _{hex}	Externer nichtmaskierbarer Interrupt	7400 _{hex}
0C09 _{hex}	Watchdog-Timeout	7400 _{hex}

KURZREFERENZ ZUR PARAMETERUM- SETZUNG

8.1 V-Regler-Parameter auf CANopen-Objekt

V-Reglerparameter	CANopen-Objekt-Nummer		Betriebsart nach DS 402/ Herstellerspezifische Parameter
	Index	Subindex	
P001	6043 _{hex} /6044 _{hex}	00 _{hex}	Velocity Mode
P002	6042 _{hex}	00 _{hex}	Velocity Mode
P003	2003 _{hex}	00 _{hex}	Hochlaufgeber
P004	2004 _{hex}	00 _{hex}	Hochlaufgeber
P005	2005 _{hex}	00 _{hex}	Hochlaufgeber
P006	2006 _{hex}	00 _{hex}	Hochlaufgeber
P007	2007 _{hex}	00 _{hex}	Hochlaufgeber
P009	2009 _{hex}	00 _{hex}	Hochlaufgeber
P010	2010 _{hex}	00 _{hex}	Hochlaufgeber
P011	2011 _{hex}	00 _{hex}	Hochlaufgeber
P012	2012 _{hex}	00 _{hex}	Hochlaufgeber
P013	2013 _{hex}	00 _{hex}	Hochlaufgeber
P014	2014 _{hex}	00 _{hex}	Hochlaufgeber
P016	2016 _{hex}	00 _{hex}	Hochlaufgeber
P017	2017 _{hex}	00 _{hex}	Objektliste
P018	2018 _{hex}	00 _{hex}	Objektliste
P019	2019 _{hex}	00 _{hex}	Gebermanager
P020	2020 _{hex}	00 _{hex}	Auswertung Geber 1
P021	2021 _{hex}	00 _{hex}	Auswertung Geber 1
P022	2022 _{hex}	00 _{hex}	Auswertung Geber 1
P023	2023 _{hex}	00 _{hex}	Auswertung Geber 1
P024	2024 _{hex}	00 _{hex}	Auswertung Geber 1
P025	2025 _{hex}	00 _{hex}	Auswertung Geber 1
P026	2026 _{hex}	00 _{hex}	Auswertung Geber 1
P030	2030 _{hex}	00 _{hex}	Feldwinkelberechnung

V-Reglerparameter	CANopen-Objekt-Nummer		Betriebsart nach DS 402/ Herstellerspezifische Parameter
	Index	Subindex	
P031	2031 _{hex}	00 _{hex}	Feldwinkelberechnung
P032	2032 _{hex}	00 _{hex}	Feldwinkelberechnung
P033	604D _{hex}	00 _{hex}	Velocity Mode
P034	2034 _{hex}	00 _{hex}	Feldwinkelberechnung
P035	2035 _{hex}	00 _{hex}	Feldwinkelberechnung
P036	2036 _{hex}	00 _{hex}	Feldwinkelberechnung
P037	2037 _{hex}	00 _{hex}	Feldwinkelberechnung
P038	2038 _{hex}	00 _{hex}	Drehzahlregelung
P039	2039 _{hex}	00 _{hex}	Feldwinkelberechnung
P040	2040 _{hex}	00 _{hex}	Auswertung Geber 1
P041	2041 _{hex}	00 _{hex}	Auswertung Geber 1
P042	2042 _{hex}	00 _{hex}	Auswertung Geber 1
P043	2043 _{hex}	00 _{hex}	Auswertung Geber 1
P044	2044 _{hex}	00 _{hex}	Auswertung Geber 1
P045	2045 _{hex}	00 _{hex}	Auswertung Geber 1
P046	2046 _{hex}	00 _{hex}	Auswertung Geber 1
P047	2047 _{hex}	00 _{hex}	Drehzahlregelung
P048	2048 _{hex}	00 _{hex}	Stromregelung
P049	2049 _{hex}	00 _{hex}	Drehzahlregelung
P050	2050 _{hex}	00 _{hex}	Drehzahlregelung
P051	2051 _{hex}	00 _{hex}	Drehzahlregelung
P052	2052 _{hex}	00 _{hex}	Drehzahlregelung
P053	2053 _{hex}	00 _{hex}	Drehzahlregelung
P054	2054 _{hex}	00 _{hex}	Drehzahlregelung
P055	2055 _{hex}	00 _{hex}	Drehzahlregelung
P056	2056 _{hex}	00 _{hex}	Drehzahlregelung
P057	2057 _{hex}	00 _{hex}	Drehzahlregelung
P058	2058 _{hex}	00 _{hex}	Drehzahlregelung
P059	2059 _{hex}	00 _{hex}	Drehzahlregelung
P060	2060 _{hex}	00 _{hex}	Drehzahlregelung
P061	2061 _{hex}	00 _{hex}	Drehzahlregelung
P062	2062 _{hex}	00 _{hex}	Auswertung Geber 1
P065	2065 _{hex}	00 _{hex}	Drehmoment-Überwachung
P066	2066 _{hex}	00 _{hex}	Drehmoment-Überwachung
P067	2067 _{hex}	00 _{hex}	Stromregelung
P068	2068 _{hex}	00 _{hex}	Stromregelung
P069	2069 _{hex}	00 _{hex}	Stromregelung
P070	2070 _{hex}	00 _{hex}	Stromregelung
P071	2071 _{hex}	00 _{hex}	Stromregelung
P072	2072 _{hex}	00 _{hex}	Stromregelung
P073	2073 _{hex}	00 _{hex}	Stromregelung
P074	2074 _{hex}	00 _{hex}	Stromregelung
P075	2075 _{hex}	00 _{hex}	Stromregelung
P076	2076 _{hex}	00 _{hex}	Stromregelung

V-Reglerparameter	CANopen-Objekt-Nummer		Betriebsart nach DS 402/ Herstellerspezifische Parameter
	Index	Subindex	
P077	2077 _{hex}	00 _{hex}	Stromregelung
P078	2078 _{hex}	00 _{hex}	Stromregelung
P079	2079 _{hex}	00 _{hex}	Stromregelung
P080	2080 _{hex}	00 _{hex}	Stromregelung
P081	2081 _{hex}	00 _{hex}	Stromregelung
P082	2082 _{hex}	00 _{hex}	Stromregelung
P083	2083 _{hex}	00 _{hex}	Stromregelung
P084	2084 _{hex}	00 _{hex}	Stromregelung
P086	2086 _{hex}	00 _{hex}	Stromregelung
P087	2087 _{hex}	00 _{hex}	Einspeisung
P088	2088 _{hex}	00 _{hex}	Motor-Überlastüberwachung
P089	2089 _{hex}	00 _{hex}	Motor-Überlastüberwachung
P090	2090 _{hex}	00 _{hex}	Leistungsteil
P091	2091 _{hex}	00 _{hex}	Motor-Überlastüberwachung
P092	2092 _{hex}	00 _{hex}	Drehmom.-Überwachung
P093	2093 _{hex}	00 _{hex}	Motor-Überlastüberwachung
P094	2094 _{hex}	00 _{hex}	Drehmom.-Überwachung
P095	2095 _{hex}	00 _{hex}	Drehmom.-Überwachung
P096	2096 _{hex}	00 _{hex}	Drehmom.-Überwachung
P097	2097 _{hex}	00 _{hex}	Drehmom.-Überwachung
P098	2098 _{hex}	00 _{hex}	Drehmom.-Überwachung
P099	2099 _{hex}	00 _{hex}	Stromregelung
P100	2100 _{hex}	00 _{hex}	PWM
P101	2101 _{hex}	00 _{hex}	PWM
P102	2102 _{hex}	00 _{hex}	PWM
P103	2103 _{hex}	00 _{hex}	PWM
P104	2104 _{hex}	00 _{hex}	Geber 2
P105	2105 _{hex}	00 _{hex}	Geber 2
P106	2106 _{hex}	00 _{hex}	Geber 2
P107	2107 _{hex}	00 _{hex}	Geber 1
P108	2108 _{hex}	00 _{hex}	Geber 1
P109	2109 _{hex}	00 _{hex}	Geber 1
P110	2110 _{hex}	00 _{hex}	Einspeisung
P111	2111 _{hex}	00 _{hex}	Einspeisung
P112	2112 _{hex}	00 _{hex}	Einspeisung
P113	2113 _{hex}	00 _{hex}	Leistungsteil
P114	2114 _{hex}	00 _{hex}	Leistungsteil
P115	2115 _{hex}	00 _{hex}	Leistungsteil
P116	2116 _{hex}	00 _{hex}	Leistungsteil
P117	2117 _{hex}	00 _{hex}	Leistungsteil
P118	2118 _{hex}	00 _{hex}	Leistungsteil
P119	2119 _{hex}	00 _{hex}	Leistungsteil
P120	6040 _{hex}	00 _{hex}	Device Control
P121	6041 _{hex}	00 _{hex}	Device Control

V-Reglerparameter	CANopen-Objekt-Nummer		Betriebsart nach DS 402/ Herstellerspezifische Parameter
	Index	Subindex	
P122	6060 _{hex}	00 _{hex}	Device Control
P123	6061 _{hex}	00 _{hex}	Device Control
P124	2124 _{hex}	00 _{hex}	Antriebs-Manager
P125	2125 _{hex}	00 _{hex}	Antriebs-Manager
P126	2126 _{hex}	00 _{hex}	Antriebs-Manager
P127	2127 _{hex}	00 _{hex}	Antriebs-Manager
P128	2128 _{hex}	00 _{hex}	Antriebs-Manager
P129	2129 _{hex}	00 _{hex}	Antriebs-Manager
P130	605D _{hex}	00 _{hex}	Device Control
P131	605A _{hex}	00 _{hex}	Device Control
P132	605C _{hex}	00 _{hex}	Device Control
P133	605B _{hex}	00 _{hex}	Device Control
P134	2134 _{hex}	00 _{hex}	Antriebs-Manager
P135	2135 _{hex}	00 _{hex}	Antriebs-Manager
P136	2136 _{hex}	00 _{hex}	Antriebs-Manager
P137	2137 _{hex}	00 _{hex}	Antriebs-Manager
P138	2138 _{hex}	00 _{hex}	Antriebs-Manager
P139	2139 _{hex}	00 _{hex}	Leistungsteil
P140	2140 _{hex}	00 _{hex}	Sollwert-Generator
P141	2141 _{hex}	00 _{hex}	Sollwert-Generator
P142	2142 _{hex}	00 _{hex}	Sollwert-Generator
P143	2143 _{hex}	00 _{hex}	Sollwert-Generator
P144	2144 _{hex}	00 _{hex}	Sollwert-Generator
P145	2145 _{hex}	00 _{hex}	Sollwert-Generator
P146	2146 _{hex}	00 _{hex}	Sollwert-Generator
P147	2147 _{hex}	00 _{hex}	Sollwert-Generator
P148	2148 _{hex}	00 _{hex}	Sollwert-Generator
P149	2149 _{hex}	00 _{hex}	Sollwert-Generator
P150	2150 _{hex}	00 _{hex}	Sollwert-Generator
P151	2151 _{hex}	00 _{hex}	Motortemp.-Überwachung
P152	2152 _{hex}	00 _{hex}	Motortemp.-Überwachung
P153	2153 _{hex}	00 _{hex}	Motortemp.-Überwachung
P154	2154 _{hex}	00 _{hex}	Motortemp.-Überwachung
P155	2155 _{hex}	00 _{hex}	Motortemp.-Überwachung
P156	2156 _{hex}	00 _{hex}	Motortemp.-Überwachung
P157	2157 _{hex}	00 _{hex}	Motortemp.-Überwachung
P159	2159 _{hex}	00 _{hex}	Betriebssystem
P160	2160 _{hex}	00 _{hex}	Betriebssystem
P161	2161 _{hex}	00 _{hex}	Betriebssystem
P162	2162 _{hex}	00 _{hex}	Betriebssystem
P163	2163 _{hex}	00 _{hex}	Betriebssystem
P164	2164 _{hex}	00 _{hex}	Betriebssystem
P165	2165 _{hex}	00 _{hex}	Betriebssystem
P166	2166 _{hex}	00 _{hex}	Betriebssystem

V-Reglerparameter	CANopen-Objekt-Nummer		Betriebsart nach DS 402/ Herstellerspezifische Parameter
	Index	Subindex	
P167	2167 _{hex}	00 _{hex}	Betriebssystem
P168	2168 _{hex}	00 _{hex}	Betriebssystem
P169	2169 _{hex}	00 _{hex}	Betriebssystem
P170	2170 _{hex}	00 _{hex}	Service-Schnittstelle
P171	2171 _{hex}	00 _{hex}	Service-Schnittstelle
P172	2172 _{hex}	00 _{hex}	Service-Schnittstelle
P173	2173 _{hex}	00 _{hex}	Service-Schnittstelle
P174	2174 _{hex}	00 _{hex}	Betriebssystem
P175	2175 _{hex}	00 _{hex}	BAPS
P176	2176 _{hex}	00 _{hex}	BAPS
P177	2177 _{hex}	00 _{hex}	BAPS
P178	2178 _{hex}	00 _{hex}	BAPS
P180	2180 _{hex}	00 _{hex}	USS
P181	2181 _{hex}	00 _{hex}	USS
P182	2182 _{hex}	00 _{hex}	USS
P183	2183 _{hex}	00 _{hex}	USS
P184	2184 _{hex}	00 _{hex}	USS
P185	2185 _{hex}	00 _{hex}	USS
P186	2186 _{hex}	00 _{hex}	USS
P188	2188 _{hex}	00 _{hex}	Antriebs-Manager
P189	605E _{hex}	00 _{hex}	Device Control
P190	2190 _{hex}	00 _{hex}	Datensatzverwaltung
P191	2191 _{hex}	00 _{hex}	Datensatzverwaltung
P192	2192 _{hex}	00 _{hex}	Datensatzverwaltung
P193	2193 _{hex}	00 _{hex}	Datensatzverwaltung
P194	2194 _{hex}	00 _{hex}	Datensatzverwaltung
P195	2195 _{hex}	00 _{hex}	Datensatzverwaltung
P196	2196 _{hex}	00 _{hex}	Datensatzverwaltung
P197	2197 _{hex}	00 _{hex}	Datensatzverwaltung
P198	2198 _{hex}	00 _{hex}	Datensatzverwaltung
P200	60FB _{hex}	01 _{hex}	Position Control Function
P201	60FB _{hex}	02 _{hex}	Position Control Function
P202	60FB _{hex}	03 _{hex}	Position Control Function
P203	2203 _{hex}	00 _{hex}	Lageregelung
P204	2204 _{hex}	00 _{hex}	Lageregelung
P205	60FB _{hex}	04 _{hex}	Position Control Function
P206	60FB _{hex}	05 _{hex}	Position Control Function
P207	60FB _{hex}	06 _{hex}	Position Control Function
P208	2208 _{hex}	00 _{hex}	Lageregelung
P209	2209 _{hex}	00 _{hex}	Lageregelung
P210	60FB _{hex}	07 _{hex}	Position Control Function
P211	2211 _{hex}	00 _{hex}	Lageregelung
P212	2212 _{hex}	00 _{hex}	Lageregelung
P213	60FB _{hex}	08 _{hex}	Position Control Function

V-Reglerparameter	CANopen-Objekt-Nummer		Betriebsart nach DS 402/ Herstellerspezifische Parameter
	Index	Subindex	
P214	6066 _{hex}	00 _{hex}	Position Control Function
P218	60FB _{hex}	09 _{hex}	Position Control Function
P219	60FB _{hex}	0A _{hex}	Position Control Function
P220	60FB _{hex}	0B _{hex}	Position Control Function
P224	2224 _{hex}	00 _{hex}	Geber-Manager
P225	2225 _{hex}	00 _{hex}	Geber-Manager
P226	2226 _{hex}	00 _{hex}	Geber-Manager
P227	2227 _{hex}	00 _{hex}	Geber-Manager
P228	2228 _{hex}	00 _{hex}	Geber-Manager
P229	2229 _{hex}	00 _{hex}	Geber-Manager
P230	2230 _{hex}	00 _{hex}	Geber 2
P231	2231 _{hex}	00 _{hex}	Geber 2
P232	2232 _{hex}	00 _{hex}	Geber 2
P233	2233 _{hex}	00 _{hex}	Geber-Manager
P234	2234 _{hex}	00 _{hex}	Geber-Manager
P235	2235 _{hex}	00 _{hex}	Geber-Manager
P236	2236 _{hex}	00 _{hex}	Geber 2
P237	2237 _{hex}	00 _{hex}	Geber 2
P238	2238 _{hex}	00 _{hex}	Geber 2
P239	2239 _{hex}	00 _{hex}	Geber 2
P240	2240 _{hex}	00 _{hex}	Geber 2
P241	2241 _{hex}	00 _{hex}	Geber 2
P242	2242 _{hex}	00 _{hex}	Geber 2
P243	2243 _{hex}	00 _{hex}	Geber 2
P244	2244 _{hex}	00 _{hex}	Geber 2
P247	2247 _{hex}	00 _{hex}	Geber 2
P248	2248 _{hex}	00 _{hex}	Geber 2
P249	2249 _{hex}	00 _{hex}	Geber 2
P250	2250 _{hex}	00 _{hex}	Gleichlauf
P251	2251 _{hex}	00 _{hex}	Gleichlauf
P252	2252 _{hex}	00 _{hex}	Gleichlauf
P253	2253 _{hex}	00 _{hex}	Gleichlauf
P254	2254 _{hex}	00 _{hex}	Gleichlauf
P255	2255 _{hex}	00 _{hex}	Gleichlauf
P256	2256 _{hex}	00 _{hex}	Gleichlauf
P257	2257 _{hex}	00 _{hex}	Gleichlauf
P258	2258 _{hex}	00 _{hex}	Gleichlauf
P259	2259 _{hex}	00 _{hex}	Gleichlauf
P260	2260 _{hex}	00 _{hex}	Motor-Modell
P261	2261 _{hex}	00 _{hex}	Motor-Modell
P262	2262 _{hex}	00 _{hex}	Motor-Modell
P263	2263 _{hex}	00 _{hex}	Motor-Modell
P264	2264 _{hex}	00 _{hex}	Motor-Modell
P265	2265 _{hex}	00 _{hex}	Motor-Modell

V-Reglerparameter	CANopen-Objekt-Nummer		Betriebsart nach DS 402/ Herstellerspezifische Parameter
	Index	Subindex	
P266	2266 _{hex}	00 _{hex}	Motor-Modell
P267	2267 _{hex}	00 _{hex}	Motor-Modell
P268	2268 _{hex}	00 _{hex}	Motor-Modell
P269	2269 _{hex}	00 _{hex}	Motor-Modell
P270	2270 _{hex}	00 _{hex}	Motor-Poti
P271	2271 _{hex}	00 _{hex}	Motor-Poti
P272	2272 _{hex}	00 _{hex}	Motor-Poti
P273	2273 _{hex}	00 _{hex}	Motor-Poti
P274	2274 _{hex}	00 _{hex}	Motor-Poti
P275	2275 _{hex}	00 _{hex}	Motor-Poti
P276	2276 _{hex}	00 _{hex}	Motor-Poti
P277	2277 _{hex}	00 _{hex}	Analoge Eingänge
P278	2278 _{hex}	00 _{hex}	Analoge Eingänge
P279	2279 _{hex}	00 _{hex}	Analoge Eingänge
P280	2280 _{hex}	00 _{hex}	Analoge Eingänge
P281	2281 _{hex}	00 _{hex}	Analoge Eingänge
P282	2282 _{hex}	00 _{hex}	Analoge Eingänge
P283	2283 _{hex}	00 _{hex}	Analoge Eingänge
P284	2284 _{hex}	00 _{hex}	Analoge Eingänge
P285	2285 _{hex}	00 _{hex}	Analoge Eingänge
P286	2286 _{hex}	00 _{hex}	Analoge Eingänge
P287	2287 _{hex}	00 _{hex}	Analoge Eingänge
P288	2288 _{hex}	00 _{hex}	Analoge Eingänge
P289	2289 _{hex}	00 _{hex}	Analoge Eingänge
P290	2290 _{hex}	00 _{hex}	Analoge Eingänge
P291	2291 _{hex}	00 _{hex}	Motor-Modell
P292	2292 _{hex}	00 _{hex}	Motor-Modell
P293	2293 _{hex}	00 _{hex}	Motor-Modell
P294	2294 _{hex}	00 _{hex}	Motor-Modell
P295	2295 _{hex}	00 _{hex}	Motor-Modell
P296	2296 _{hex}	00 _{hex}	Motor-Modell
P297	2297 _{hex}	00 _{hex}	Motor-Modell
P300	2300 _{hex}	00 _{hex}	Gleichlauf
P301	2301 _{hex}	00 _{hex}	Gleichlauf
P302	2302 _{hex}	00 _{hex}	Analoge Eingänge
P311	2311 _{hex}	00 _{hex}	Gleichlauf
P312	2312 _{hex}	00 _{hex}	Gleichlauf
P321	2321 _{hex}	00 _{hex}	Gleichlauf
P322	2322 _{hex}	00 _{hex}	Gleichlauf
P323	2323 _{hex}	00 _{hex}	Gleichlauf
P324	2324 _{hex}	00 _{hex}	Gleichlauf
P325	2325 _{hex}	00 _{hex}	Gleichlauf
P326	2326 _{hex}	00 _{hex}	Gleichlauf
P327	2327 _{hex}	00 _{hex}	Gleichlauf

V-Reglerparameter	CANopen-Objekt-Nummer		Betriebsart nach DS 402/ Herstellerspezifische Parameter
	Index	Subindex	
P328	2328 _{hex}	00 _{hex}	Gleichlauf
P329	2329 _{hex}	00 _{hex}	Gleichlauf
P330	2330 _{hex}	00 _{hex}	Analoge Ausgänge
P331	2331 _{hex}	00 _{hex}	Analoge Ausgänge
P332	2332 _{hex}	00 _{hex}	Analoge Ausgänge
P334	2334 _{hex}	00 _{hex}	Analoge Ausgänge
P335	2335 _{hex}	00 _{hex}	Analoge Ausgänge
P336	2336 _{hex}	00 _{hex}	Analoge Ausgänge
P337	2337 _{hex}	00 _{hex}	Analoge Ausgänge
P338	2338 _{hex}	00 _{hex}	Analoge Ausgänge
P342	2342 _{hex}	00 _{hex}	Digitale Eingänge
P343	2343 _{hex}	00 _{hex}	Digitale Eingänge
P344	2344 _{hex}	00 _{hex}	Digitale Eingänge
P345	2345 _{hex}	00 _{hex}	Digitale Eingänge
P350	2350 _{hex}	00 _{hex}	LED
P351	2351 _{hex}	00 _{hex}	LED
P352	2352 _{hex}	00 _{hex}	LED
P353	2353 _{hex}	00 _{hex}	LED
P354	2354 _{hex}	00 _{hex}	LED
P355	2355 _{hex}	00 _{hex}	LED
P356	2356 _{hex}	00 _{hex}	LED
P357	2357 _{hex}	00 _{hex}	LED
P358	2358 _{hex}	00 _{hex}	LED
P359	2359 _{hex}	00 _{hex}	LED
P360	2360 _{hex}	00 _{hex}	LED
P361	2361 _{hex}	00 _{hex}	LED
P362	2362 _{hex}	00 _{hex}	LED
P370	2370 _{hex}	00 _{hex}	Digitale Eingänge
P371	2371 _{hex}	00 _{hex}	Digitale Eingänge
P372	2372 _{hex}	00 _{hex}	Digitale Eingänge
P373	2373 _{hex}	00 _{hex}	Digitale Eingänge
P374	2374 _{hex}	00 _{hex}	Digitale Eingänge
P375	2375 _{hex}	00 _{hex}	Digitale Eingänge
P376	2376 _{hex}	00 _{hex}	Digitale Eingänge
P377	2377 _{hex}	00 _{hex}	Digitale Eingänge
P378	2378 _{hex}	00 _{hex}	Digitale Eingänge
P379	2379 _{hex}	00 _{hex}	Digitale Eingänge
P380	2380 _{hex}	00 _{hex}	Digitale Eingänge
P381	2381 _{hex}	00 _{hex}	Digitale Eingänge
P382	2382 _{hex}	00 _{hex}	Digitale Eingänge
P383	2383 _{hex}	00 _{hex}	Digitale Ausgänge
P384	2384 _{hex}	00 _{hex}	Digitale Ausgänge
P385	2385 _{hex}	00 _{hex}	Digitale Ausgänge
P386	2386 _{hex}	00 _{hex}	Digitale Ausgänge

V-Reglerparameter	CANopen-Objekt-Nummer		Betriebsart nach DS 402/ Herstellerspezifische Parameter
	Index	Subindex	
P387	2387 _{hex}	00 _{hex}	Digitale Ausgänge
P388	2388 _{hex}	00 _{hex}	Digitale Ausgänge
P389	2389 _{hex}	00 _{hex}	Digitale Ausgänge
P390	2390 _{hex}	00 _{hex}	Digitale Ausgänge
P391	2391 _{hex}	00 _{hex}	Digitale Ausgänge
P392	2392 _{hex}	00 _{hex}	Digitale Ausgänge
P400	60FB _{hex}	0C _{hex}	Position Control Function
P401	60FB _{hex}	0D _{hex}	Position Control Function
P402	6092 _{hex}	01 _{hex}	Factor Group
P403	6092 _{hex}	02 _{hex}	Factor Group
P406	60FB _{hex}	0E _{hex}	Position Control Function
P408	2408 _{hex}	00 _{hex}	Positionierung
P409	2409 _{hex}	00 _{hex}	Positionierung
P410	2410 _{hex}	00 _{hex}	Positionierung
P411	2411 _{hex}	00 _{hex}	Positionierung
P412	6099 _{hex}	01 _{hex}	Homing Mode
P413	609A _{hex}	00 _{hex}	Homing Mode
P414	6098 _{hex}	00 _{hex}	Profile Position Mode
P415	607A _{hex}	00 _{hex}	Profile Position Mode
P416	60FB _{hex}	15 _{hex}	Position Control Function
P417	2417 _{hex}	00 _{hex}	Positionierung
P418	2418 _{hex}	00 _{hex}	Positionierung
P419	60C5 _{hex}	00 _{hex}	Profile Position Mode
P420	60C6 _{hex}	00 _{hex}	Profile Position Mode
P421	2421 _{hex}	00 _{hex}	Positionierung
P422	2422 _{hex}	00 _{hex}	Profile Position Mode
P423	60FB _{hex}	16 _{hex}	Position Control Function
P424	2424 _{hex}	00 _{hex}	Profile Position Mode
P425	2425 _{hex}	00 _{hex}	Profile Position Mode
P426	2426 _{hex}	00 _{hex}	Profile Position Mode
P427	2427 _{hex}	00 _{hex}	Profile Position Mode
P428	2428 _{hex}	00 _{hex}	Positionierung
P429	6067 _{hex}	00 _{hex}	Position Control Function
P430	6068 _{hex}	00 _{hex}	Position Control Function
P431	2431 _{hex}	00 _{hex}	Positionierung
P432	60FB _{hex}	0F _{hex}	Position Control Function
P433	60FD _{hex}	00 _{hex}	Common Entries
P434	60FB _{hex}	10 _{hex}	Position Control Function
P435	2435 _{hex}	00 _{hex}	Positionierung
P436	6062 _{hex}	00 _{hex}	Position Control Function
P437	6064 _{hex}	00 _{hex}	Position Control Function
P438	607B _{hex}	11 _{hex}	Position Control Function
P439	607D _{hex}	01 _{hex}	Profile Position Mode
P440	607D _{hex}	02 _{hex}	Profile Position Mode

V-Reglerparameter	CANopen-Objekt-Nummer		Betriebsart nach DS 402/ Herstellerspezifische Parameter
	Index	Subindex	
P441	6073 _{hex}	12 _{hex}	Position Control Function
P442	2442 _{hex}	00 _{hex}	Referenzfahrbetrieb
P443	6099 _{hex}	02 _{hex}	Homing Mode
P444	60FB _{hex}	13 _{hex}	Position Control Function
P445	60FB _{hex}	14 _{hex}	Position Control Function
P446	2446 _{hex}	00 _{hex}	Positionierung
P450	2450 _{hex}	00 _{hex}	Frei konfig. Statuswort
P451	2451 _{hex}	00 _{hex}	Frei konfig. Statuswort
P452	2452 _{hex}	00 _{hex}	Frei konfig. Statuswort
P453	2453 _{hex}	00 _{hex}	Frei konfig. Statuswort
P454	2454 _{hex}	00 _{hex}	Frei konfig. Statuswort
P455	2455 _{hex}	00 _{hex}	Frei konfig. Statuswort
P456	2456 _{hex}	00 _{hex}	Frei konfig. Statuswort
P457	2457 _{hex}	00 _{hex}	Frei konfig. Statuswort
P458	2458 _{hex}	00 _{hex}	Frei konfig. Statuswort
P459	2459 _{hex}	00 _{hex}	Frei konfig. Statuswort
P460	2460 _{hex}	00 _{hex}	Frei konfig. Statuswort
P461	2461 _{hex}	00 _{hex}	Frei konfig. Statuswort
P462	2462 _{hex}	00 _{hex}	Frei konfig. Statuswort
P463	2463 _{hex}	00 _{hex}	Frei konfig. Statuswort
P464	2464 _{hex}	00 _{hex}	Frei konfig. Statuswort
P465	2465 _{hex}	00 _{hex}	Frei konfig. Statuswort
P466	2466 _{hex}	00 _{hex}	Frei konfig. Statuswort
P467	2467 _{hex}	00 _{hex}	Frei konfig. Statuswort
P468	2468 _{hex}	00 _{hex}	ZK-Parameter
P469	2469 _{hex}	00 _{hex}	ZK-Parameter
P470	2470 _{hex}	00 _{hex}	ZK-Parameter
P471	2471 _{hex}	00 _{hex}	ZK-Parameter
P472	2472 _{hex}	00 _{hex}	ZK-Parameter
P473	2473 _{hex}	00 _{hex}	ZK-Parameter
P474	2474 _{hex}	00 _{hex}	ZK-Parameter
P475	2475 _{hex}	00 _{hex}	ZK-Parameter
P476	2476 _{hex}	00 _{hex}	ZK-Parameter
P477	2477 _{hex}	00 _{hex}	ZK-Parameter
P478	2478 _{hex}	00 _{hex}	ZK-Parameter
P479	2479 _{hex}	00 _{hex}	ZK-Parameter
P480	2480 _{hex}	00 _{hex}	ZK-Parameter
P481	2481 _{hex}	00 _{hex}	ZK-Parameter
P482	2482 _{hex}	00 _{hex}	ZK-Parameter
P483	2483 _{hex}	00 _{hex}	ZK-Parameter
P484	2484 _{hex}	00 _{hex}	ZK-Parameter
P485	2485 _{hex}	00 _{hex}	ZK-Parameter
P486	2486 _{hex}	00 _{hex}	ZK-Parameter
P487	2487 _{hex}	00 _{hex}	ZK-Parameter

V-Reglerparameter	CANopen-Objekt-Nummer		Betriebsart nach DS 402/ Herstellerspezifische Parameter
	Index	Subindex	
P488	2488 _{hex}	00 _{hex}	ZK-Parameter
P489	2489 _{hex}	00 _{hex}	ZK-Parameter
P490	2490 _{hex}	00 _{hex}	ZK-Parameter
P491	2491 _{hex}	00 _{hex}	ZK-Parameter
P492	2492 _{hex}	00 _{hex}	ZK-Parameter
P493	2493 _{hex}	00 _{hex}	ZK-Parameter
P494	2494 _{hex}	00 _{hex}	ZK-Parameter
P495	2495 _{hex}	00 _{hex}	ZK-Parameter
P496	2496 _{hex}	00 _{hex}	ZK-Parameter
P497	2497 _{hex}	00 _{hex}	ZK-Parameter
P498	2498 _{hex}	00 _{hex}	ZK-Parameter
P499	2499 _{hex}	00 _{hex}	ZK-Parameter
P500	2500 _{hex}	00 _{hex}	Externe Digit. Eingänge
P501	2501 _{hex}	00 _{hex}	Externe Digit. Eingänge
P502	2502 _{hex}	00 _{hex}	Externe Digit. Eingänge
P503	2503 _{hex}	00 _{hex}	Externe Digit. Eingänge
P504	2504 _{hex}	00 _{hex}	Externe Digit. Eingänge
P505	2505 _{hex}	00 _{hex}	Externe Digit. Eingänge
P506	2506 _{hex}	00 _{hex}	Externe Digit. Eingänge
P507	2507 _{hex}	00 _{hex}	Externe Digit. Eingänge
P508	2508 _{hex}	00 _{hex}	Externe Digit. Eingänge
P509	2509 _{hex}	00 _{hex}	Externe Digit. Eingänge
P510	2510 _{hex}	00 _{hex}	Externe Digit. Eingänge
P511	2511 _{hex}	00 _{hex}	Externe Digit. Eingänge
P512	2512 _{hex}	00 _{hex}	Externe Digit. Eingänge
P513	2513 _{hex}	00 _{hex}	Externe Digit. Eingänge
P514	2514 _{hex}	00 _{hex}	Externe Digit. Eingänge
P515	2515 _{hex}	00 _{hex}	Externe Digit. Eingänge
P516	2516 _{hex}	00 _{hex}	Externe Digit. Eingänge
P517	2517 _{hex}	00 _{hex}	Externe Digit. Eingänge
P518	2518 _{hex}	00 _{hex}	Externe Digit. Eingänge
P519	2519 _{hex}	00 _{hex}	Externe Digit. Eingänge
P520	2520 _{hex}	00 _{hex}	Externe Digit. Eingänge
P521	2521 _{hex}	00 _{hex}	Externe Digit. Eingänge
P522	2522 _{hex}	00 _{hex}	Externe Digit. Eingänge
P523	2523 _{hex}	00 _{hex}	Externe Digit. Eingänge
P524	2524 _{hex}	00 _{hex}	Externe Digit. Eingänge
P525	2525 _{hex}	00 _{hex}	Externe Digit. Eingänge
P526	2526 _{hex}	00 _{hex}	Externe Digit. Eingänge
P527	2527 _{hex}	00 _{hex}	Externe Digit. Eingänge
P528	2528 _{hex}	00 _{hex}	Externe Digit. Eingänge
P529	2529 _{hex}	00 _{hex}	Externe Digit. Eingänge
P530	2530 _{hex}	00 _{hex}	Externe Digit. Eingänge
P531	2531 _{hex}	00 _{hex}	Externe Digit. Eingänge

V-Reglerparameter	CANopen-Objekt-Nummer		Betriebsart nach DS 402/ Herstellerspezifische Parameter
	Index	Subindex	
P532	2532 _{hex}	00 _{hex}	Externe Digit. Eingänge
P533	2533 _{hex}	00 _{hex}	Externe Digit. Ausgänge
P535	2535 _{hex}	00 _{hex}	Externe Digit. Ausgänge
P536	2536 _{hex}	00 _{hex}	Externe Digit. Ausgänge
P537	2537 _{hex}	00 _{hex}	Externe Digit. Ausgänge
P538	2538 _{hex}	00 _{hex}	Externe Digit. Ausgänge
P539	2539 _{hex}	00 _{hex}	Externe Digit. Ausgänge
P540	2540 _{hex}	00 _{hex}	Externe Digit. Ausgänge
P541	2541 _{hex}	00 _{hex}	Externe Digit. Ausgänge
P542	2542 _{hex}	00 _{hex}	Externe Digit. Ausgänge
P543	2543 _{hex}	00 _{hex}	Externe Digit. Ausgänge
P544	2544 _{hex}	00 _{hex}	Externe Digit. Ausgänge
P545	2545 _{hex}	00 _{hex}	Externe Digit. Ausgänge
P546	2546 _{hex}	00 _{hex}	Externe Digit. Ausgänge
P547	2547 _{hex}	00 _{hex}	Externe Digit. Ausgänge
P548	2548 _{hex}	00 _{hex}	Externe Digit. Ausgänge
P549	2549 _{hex}	00 _{hex}	Externe Digit. Ausgänge
P550	2550 _{hex}	00 _{hex}	Externe Digit. Ausgänge
P551	2551 _{hex}	00 _{hex}	Externe Digit. Ausgänge
P552	2552 _{hex}	00 _{hex}	Externe Digit. Ausgänge
P553	2553 _{hex}	00 _{hex}	Externe Digit. Ausgänge
P554	2554 _{hex}	00 _{hex}	Externe Digit. Ausgänge
P555	2555 _{hex}	00 _{hex}	Externe Digit. Ausgänge
P556	2556 _{hex}	00 _{hex}	Externe Digit. Ausgänge
P557	2557 _{hex}	00 _{hex}	Externe Digit. Ausgänge
P558	2558 _{hex}	00 _{hex}	Externe Digit. Ausgänge

Spezielle Objekte:

3003 _{hex}	00 _{hex}	--	Timer für PDOs
3004 _{hex}	00 _{hex}	P120	Code für Guarding Event

8.2 CANopen-Objekte auf V-Regler-Parameter

Herstellerspezifische Objekte ergeben sich aus $2000_{\text{hex}} + \text{Parameternummer}$. Derartige Objekte verfügen ausschließlich über Subindex 00_{hex} .

Beispiel:

Parameter P053 \leftrightarrow Objektidex 2053_{hex} Subindex 00_{hex}

CANopen-Objekt-Nummer		V-Reglerparameter	Betriebsart nach DS 402/ Herstellerspezifische Parameter
Index	Subindex		
6040_{hex}	00_{hex}	P120	Device Control
6041_{hex}	00_{hex}	P121	Device Control
6042_{hex}	00_{hex}	P002	Velocity Mode
$6043_{\text{hex}}/6044_{\text{hex}}$	00_{hex}	P001	Velocity Mode
$604D_{\text{hex}}$	00_{hex}	P033	Velocity Mode
$605A_{\text{hex}}$	00_{hex}	P131	Device Control
$605B_{\text{hex}}$	00_{hex}	P133	Device Control
$605C_{\text{hex}}$	00_{hex}	P132	Device Control
$605D_{\text{hex}}$	00_{hex}	P130	Device Control
$605E_{\text{hex}}$	00_{hex}	P189	Device Control
6060_{hex}	00_{hex}	P122	Device Control
6061_{hex}	00_{hex}	P123	Device Control
6062_{hex}	00_{hex}	P436	Position Control Function
6064_{hex}	00_{hex}	P437	Position Control Function
6066_{hex}	00_{hex}	P214	Position Control Function
6067_{hex}	00_{hex}	P429	Position Control Function
6068_{hex}	00_{hex}	P430	Position Control Function
$607A_{\text{hex}}$	00_{hex}	P415	Profile Position Mode
$607D_{\text{hex}}$	01_{hex}	P439	Profile Position Mode
$607D_{\text{hex}}$	02_{hex}	P440	Profile Position Mode
6092_{hex}	01_{hex}	P402	Factor Group
6092_{hex}	02_{hex}	P403	Factor Group
6098_{hex}	00_{hex}	P414	Profile Position Mode
6099_{hex}	01_{hex}	P412	Homing Mode
6099_{hex}	02_{hex}	P443	Homing Mode
$609A_{\text{hex}}$	00_{hex}	P413	Homing Mode
$60C5_{\text{hex}}$	00_{hex}	P419	Profile Position Mode
$60C6_{\text{hex}}$	00_{hex}	P420	Profile Position Mode
$60FB_{\text{hex}}$	01_{hex}	P200	Position Control Function
$60FB_{\text{hex}}$	02_{hex}	P201	Position Control Function
$60FB_{\text{hex}}$	03_{hex}	P202	Position Control Function
$60FB_{\text{hex}}$	04_{hex}	P205	Position Control Function
$60FB_{\text{hex}}$	05_{hex}	P206	Position Control Function
$60FB_{\text{hex}}$	06_{hex}	P207	Position Control Function
$60FB_{\text{hex}}$	07_{hex}	P210	Position Control Function
$60FB_{\text{hex}}$	08_{hex}	P213	Position Control Function

CANopen-Objekt-Nummer		V-Reglerparameter	Betriebsart nach DS 402/ Herstellerspezifische Parameter
Index	Subindex		
60FB _{hex}	09 _{hex}	P218	Position Control Function
60FB _{hex}	0A _{hex}	P219	Position Control Function
60FB _{hex}	0B _{hex}	P220	Position Control Function
60FB _{hex}	0C _{hex}	P400	Position Control Function
60FB _{hex}	0D _{hex}	P401	Position Control Function
60FB _{hex}	0E _{hex}	P406	Position Control Function
60FB _{hex}	0F _{hex}	P432	Position Control Function
60FB _{hex}	10 _{hex}	P434	Position Control Function
60FB _{hex}	11 _{hex}	P438	Position Control Function
60FB _{hex}	12 _{hex}	P441	Position Control Function
60FB _{hex}	13 _{hex}	P444	Position Control Function
60FB _{hex}	14 _{hex}	P445	Position Control Function
60FB _{hex}	15 _{hex}	P416	Position Control Function
60FB _{hex}	16 _{hex}	P423	Position Control Function
60FD _{hex}	00 _{hex}	P433	Common Entries



ANHANG A - ABKÜRZUNGEN

ASCII	American Standard Code for Information Interchange
BAPS	Baumüller Antriebe Parallele Schnittstelle
CAN	Controller Area Network
COB-ID	CAN Objekt Identifier
CSMA/CA	Carrier Sense Multiple Access/Collision Avoidance
D0 - D3	Nutzdatendatenbytes innerhalb des CAN-Datentelegramms
EDS	Electronic Data Sheet
ISO	International Standardization Organization
LSB	Least Significant Byte
MSB	Most Significant Byte
NMT	Network Management
OSI	Open System for Interconnection
PDO	Prozess Daten Objekt
SDO	Service Daten Objekt
SYNC	Synchronisations- Telegramm
U16	Unsigned 16
U32	Unsigned 32
U8	Unsigned 8



ANHANG B - UMSETZUNGSTABELLEN

Dieses Kapitel beinhaltet die Tabellen, welche die Umsetzung der CANopen-Kommunikationsobjekte → V-Regler-Kommunikationsparameter und umgekehrt spezifizieren. Die Umsetzung erfolgt unter Angabe der Wertebereiche ($x=x_{\min}..x_{\max}$) 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 (HEX) → P-Nr.:** Abbildung der CANopen-Objektindizes auf V-Regler-Parameter
- ▶ **V-Regler-Parameter:** Bezeichnung des V-Regler-Parameters
- ▶ **P-Nr. → Index (HEX):** Umsetzung der V-Regler-Parameter auf CANopen-Objektindizes

CANopen-Objekt	Index (HEX)	→	P-Nr.	V-Regler-Parameter	P-Nr.	→	Index (HEX)	Kommentar
controlword	6040	→	P120	Steuer-Wort	P120	→	6040	
Umrechnungsformalismus	$x=-32768..32767$	→	$y=x$		$x=0x0..0xFFFF$	→	$y=x$	
Switch On Bit 0	unverändert			Einschalten	unverändert			
Disable Voltage Bit 1	unverändert			Spannung sperren	unverändert			
Quick Stop Bit 2	unverändert			Schnellhalt	unverändert			
Enable Op. Bit 3	unverändert			Betrieb freigeben	unverändert			
Op.-Mode-Spec. Bit 4	unverändert			betriebsartabhängig	unverändert			
Op.-Mode-Spec. Bit 5	unverändert			betriebsartabhängig	unverändert			
Op.-Mode-Spec. Bit 6	unverändert			betriebsartabhängig	unverändert			
Reset Fault Bit 7	unverändert			Reset Störung	unverändert			
Halt Bit 8	unverändert			Reserve (immer 0)	unverändert			
reserved Bit 9	unverändert			Reserve (immer 0)	unverändert			
reserved Bit 10	unverändert			Reserve (immer 0)	unverändert			
Man.specific Bit 11	unverändert			betriebsartabhängig	unverändert			
Man.specific Bit 12	unverändert			betriebsartabhängig	unverändert			
Man.specific Bit 13	unverändert			betriebsartabhängig	unverändert			
Man.specific Bit 14	unverändert			betriebsartabhängig	unverändert			
Man.specific Bit 15	unverändert			Schreibschutz	unverändert			
statusword	6041	→	P121	Status-Wort	P121	→	6041	
Umrechnungsformalismus	$x=0..65535$	→	$y=x$		$x=0x0..0xFFFF$	→	$y=x$	
Ready To Switch On Bit 0	unverändert			Einschaltbereit	unverändert			
Switched On Bit 1	unverändert			Eingeschaltet	unverändert			
Operation Enabled Bit 2	unverändert			Betrieb freigegeben	unverändert			

CANopen-Objekt	Index (HEX)	→	P-Nr.	V-Regler-Parameter	P-Nr.	→	Index (HEX)	Kommentar
Fault Bit 3	unverändert			Störung	unverändert			
Voltage Disabled Bit 4	unverändert			Spannung gesperrt	unverändert			
Quick Stop Bit 5	unverändert			Schnell-Halt	unverändert			
Switched On Enabled Bit 6	unverändert			Einschaltsperr	unverändert			
Warning Bit 7	unverändert			Reserve (immer 0)	unverändert			
Man.specific Bit 8	unverändert			Reserve (immer 0)	unverändert			
Remote Bit 9	unverändert			Remote	unverändert			
Targeted Reached Bit 10	unverändert			Sollwert erreicht	unverändert			
Int.Limit Active Bit 11	unverändert			Reserve (immer 0)	unverändert			
Op.Mode Spec. Bit 12	unverändert			betriebsartabhängig	unverändert			
Op.Mode Spec. Bit 13	unverändert			betriebsartabhängig	unverändert			
Man.specific Bit 14	unverändert			konf. Statusbits	unverändert			
Man.specific Bit 15	unverändert			konf. Statusbits	unverändert			
vl_target_velocity	6042	→	P002	Eingang 1	P002	→	6042	
	x=-32768..32767	→	y=x / 2		x=-16383..16383	→	y=x * 2	Die benutzerdefinierte Einheit (speed units) von vl_target_velocity wird im V-Regler als U/min interpretiert
vl_velocity_demand	6043 / read only			Ausgang	P001	→	6043	
	-32268..32267				x=-16383..16383	→	y=x * 2	
vl_control_effort	6044 / read only			Ausgang	P001	→	6044	
	-32268..32267				x=-16383..16383	→	y=x * 2	
vl_pole_number	604D	→	P033	Polpaarzahl	P033	→	604D	
	x=0..255	→	y=x / 2		x=1..120	→	y=x * 2	
quick_stop_option_code	605A	→	P131	Schnellhalt	P131	→	605A	
Umrechnungsformalismus	x=-32768..32767	→	y=x		x=0..3	→	y=x	
Manufacturer specific	x=-32768..-1	→	y=x	unbenutzt	-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 Rücklauf-rampe	x=1	→	y=x	
Slow Down On Quick Stop Ramp	x=2	→	y=x	Rücklauf auf Schnellhal-trampe	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	4			
Slow Down On Slow Down Ramp and stay in Quick-Stop	5			unbenutzt	5			
Slow Down On Quick Stop Ramp and stay in Quick-Stop	6			unbenutzt	6			
Slow Down On Current and stay in Quick-Stop	7			unbenutzt	7			
Slow Down On Voltage Limit and stay in Quick-Stop	8			unbenutzt	8			
reserved	9..32767			unbenutzt	9..32767			
shutdown_option_code	605B	→	P133	Stillsetzen	P133	→	605B	
Manufacturer specific	x=-32768..-3	→	y=x	unbenutzt	-32768..-3			
Manufacturer specific	x=-2	→	y=2	unbenutzt	-2			
Manufacturer specific	x=-1	→	y=3	unbenutzt	-1			
Disable Drive	x=0	→	y=x	Antrieb gesperrt	x=0	→	y=x	
Slow Down...	x=1	→	y=x	Rücklauf auf Rücklauf-rampe	x=1	→	y=x	

CANopen-Objekt	Index (HEX)	→	P-Nr.	V-Regler-Parameter	P-Nr.	→	Index (HEX)	Kommentar
reserved	2			Rücklauf auf Schnellhaltrampe	x=2	→	y=-1	
reserved	3			Rücklauf an Stromgrenze	x=3	→	y=-2	
reserved	4..32767			unbenutzt	4..32767			
disable_operation_option_code	605C	→	P132	Sperren	P132	→	605C	
Manufacturer specific	x=-32768..-3	→	y=x	unbenutzt	-32768..-3			
Manufacturer specific	x=-2	→	y=3	unbenutzt	-2			
Manufacturer specific	x=-1	→	y=2	unbenutzt	-1			
Disable Drive	x=0	→	y=x	Antrieb gesperrt	x=0	→	y=x	
Slow Down...	x=1	→	y=x	Rücklauf auf Rücklauf-rampe	x=1	→	y=x	
reserved	2			Rücklauf auf Schnellhaltrampe	x=2	→	y=-1	
reserved	3			Rücklauf an Stromgrenze	x=3	→	y=-2	
reserved	4..32767			unbenutzt	4..32767			
stop_option_code	605D	→	P130	Halt	P130	→	605D	
Umrechnungsformalismus	x=-32768..32767	→	y=x		x=0..3	→	y=x	
Manufacturer specific	x=-32768..-1	→	y=x	unbenutzt	-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 Rücklauf-rampe	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	unbenutzt	4			
reserved	5..32767			unbenutzt	5..32767			
fault_reaction_option_code	605E	→	P189	Fehlerreaktion	P189	→	605E	
Umrechnungsformalismus	x=-32768..32767	→	y=x		x=0..3	→	y=x	
Manufacturer specific	x=-32768..-1	→	y=x	unbenutzt	-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 Rücklauf-rampe	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	unbenutzt	4			
reserved	5..32767			unbenutzt	5..32767			
modes_of_operation	6060	→	P122	Soll-Betriebsart	P122	→	6060	
Umrechnungsformalismus	x=-128..127	→	y=x		x=-128..127	→	y=x	
Manufacturer specific	x=-128..-7	→	y=x	unbenutzt	-128..-7			
Manufacturer specific	x=-6	→	5	unbenutzt	-6			
Manufacturer specific	x=-5	→	y=x	Gleichlauf el. Getriebe	x=-5	→	y=x	
Manufacturer specific	x=-4	→	y=x	Lageregelung	x=-4	→	y=x	
Manufacturer specific	x=-3	→	y=x	Drehzahlregelung	x=-3	→	y=2	
Manufacturer specific	x=-2	→	y=x	Stromregelung	x=-2	→	y=x	
Manufacturer specific	x=-1	→	y=x	Rastlage	x=-1	→	y=x	
reserved	0			unbenutzt	0			
Profile Position Mode	x=1	→	y=x	Lagezielvorgabe	x=1	→	y=x	

CANopen-Objekt	Index (HEX)	→	P-Nr.	V-Regler-Parameter	P-Nr.	→	Index (HEX)	Kommentar
Velocity Mode	x=2	→	-3	Geschwindigkeitsvorgabe 1	x=2	→	y=3	
Profile Velocity Mode	x=3	→	2	unbenutzt	3			
Torque Profile Mode	x=4	→	y=x	unbenutzt	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=x	unbenutzt	7			
reserved	x=8..127			unbenutzt	8..127			
modes_of_operation_display	6061	→	P123	Ist-Betriebsart	P123	→	6061	
Umrechnungsformalismus	x=-128..127	→	y=x		x=-128..127	→	y=x	
Manufacturer specific	x=-128..-7	→	y=x	unbenutzt	-128..-7			
Manufacturer specific	x=-6	→	y=5	unbenutzt	-6			
Manufacturer specific	x=-5	→	y=x	Gleichlauf el. Getriebe	x=-5	→	y=x	
Manufacturer specific	x=-4	→	y=x	Lageregelung	x=-4	→	y=x	
Manufacturer specific	x=-3	→	y=x	Drehzahlregelung	x=-3	→	y=2	
Manufacturer specific	x=-2	→	y=x	Stromregelung	x=-2	→	y=x	
Manufacturer specific	x=-1	→	y=x	Rastlage	x=-1	→	y=x	
reserved	0			unbenutzt	0			
Profile Position Mode	x=1	→	y=x	Lagezielvorgabe	x=1	→	y=x	
Velocity Mode	x=2	→	y=-3	Geschwindigkeitsvorgabe 1	x=2	→	y=3	
Profile Velocity Mode	x=3	→	y=2	unbenutzt	3			
Torque Profile Mode	x=4	→	y=x	unbenutzt	4			
reserved	5			Handfahrbetrieb	x=5	→	y=-6	
Homing Mode	x=6	→	y=x	Referenzbetrieb	x=6	→	y=x	
Interpolated Position Mode	x=7	→	y=x	unbenutzt	7			
reserved	8..127			unbenutzt	8..127			
position_demand_value	6062 / read only			Lage-Sollwert	P436	→	6062	
	x=-2^31..2^31-1				x=0x0..0xFFFFFFFF	→	y=x	
position_actual_value	6064 / read only			Lage-Istwert	P437	→	6064	
	x=-2^31..2^31-1				x=0x0..0xFFFFFFFF	→	y=x	
following_error_time_out	6066	→	P214	SF-Zeit	P214	→	6066	
	x=0..65535	→	y=x	Der interne Wertebereich von x=0.000..65.000 wird bereits auf einen INTEGER-Wert konvertiert.	x=0..65000	→	y=x	
position_window	6067	→	P429	Pos.-Fenster	P429	→	6067	
	x=0..2^32-1	→	y=x		x=0x0..0xFFFFFFFF	→	y=x	
position_window_time	6068	→	P430	Pos.-Fensterzeit	P430	→	6068	
	x=0..65535	→	y=x		x=0x0..0xFFFFFFFF	→	y=x	
target_position		→	P415	Zielposition 1	P415	→	607A	
	x=-2^31..2^31-1	→	y=x		x=0x0..0xFFFFFFFF	→	y=x	
software_position_limit	607D	→	P439 P440	SW-Endschalter	P439 P440	→	607D	
	Sub. 01	→	P439		P439	→	Sub. 01	
	x=-2^31..2^31-1	→	y=x		x=0x0..0xFFFFFFFF	→	y=x	
	Sub. 02	→	P440		P440	→	Sub. 02	

CANopen-Objekt	Index (HEX)	→	P-Nr.	V-Regler-Parameter	P-Nr.	→	Index (HEX)	Kommentar
	x=-2 ³¹ ..2 ³¹ -1	→	y=x		x=0x0..0xFFFFFFFF	→	y=x	
Feed_constant	6092	→	P402 P403	Lage-Norm	P402 P403	→	6092	
	Sub. 01	→	P403	Lage-Norm N	P403	→	Sub. 01	
	x=0..2 ³² -1	→	y=x		x=1..32768	→	y=x	
	Sub. 02	→	P402	Lage-Norm Z	P402	→	Sub. 02	
	x=0..2 ³² -1	→	y=x		x=1..65535	→	y=x	
homing_method	6098	→	P414	Ref. Fahrmodus	P414	→	6098	
Manufacturer specific	x=-128..-7	→	y=x	unbenutzt	-128..-7			
Manufacturer specific	x=-6	→	y=-ab06	Anfahren des nächsten Gebernulwinkels	x=-1006 x=-2006	→	y=-6	
Manufacturer specific	x=-5	→	y=-5	Anfahren des pos. Endschalters	x=-5	→	y=-5	
Manufacturer specific	x=-4	→	y=-4	Anfahren des neg. Endschalters	x=-4	→	y=-4	
Manufacturer specific	x=-3	→	y=-3	Referenzpunkt setzen	x=-3	→	y=-3	
Manufacturer specific	x=-2	→	y=-ab02	Angahren des Gebernulwinkels (li)	x=-2002 x=-1002 x=-1102 x=-2102	→	y=-2	
Manufacturer specific	x=-1	→	y=-ab01	Angahren des Gebernulwinkels (re)	x=-1001 x=-2001 x=-1101 x=-2101	→	y=-1	
No homing operation	0			unbenutzt	0			
Homing on the neg.Limit Switch	x=1	→	y=ab01	Anfahren des neg. Endschalters	x=2001 x=1001 x=1101 x=2101	→	y=1	
Homing on the pos.Limit Switch	x=2	→	y=ab02	Anfahren des pos. Endschalters	x=1002 x=2002 x=1102 x=2102	→	y=2	
Homing on the pos.Home Switch & Index Pulse	x=3	→	y=ab03	Anfahren des pos. Nullpunktumschalters	x=1003 x=2003 x=1103 x=2103	→	y=3	
Homing on the pos.Home Switch & Index Pulse	x=4	→	y=ab04	Anfahren des pos. Nullpunktumschalters	x=1004 x=2004 x=1104 x=2104	→	y=4	
Homing on the neg.Home Switch & Index Pulse	x=5	→	y=ab05	Anfahren des neg. Nullpunktumschalters	x=1005 x=2005 x=1105 x=2105	→	y=5	
Homing on the neg.Home Switch & Index Pulse	x=6	→	y=ab06	Anfahren des neg. Nullpunktumschalters	x=1006 x=2006 x=1106 x=2106	→	y=6	
CANopen-Spec.	7..34			unbenutzt	7..34			
reserviert	35..127			unbenutzt	35..127			
				Gebereingang	P226	→	1.Stelle a plus Wert aus 6098	
Umrechnungsformalismus für x & 0x0001 !=0x0001				Gebereingang Bit0=0	x=0x0..0x007F	→	a=1	
Umrechnungsformalismus für x & 0x0001 ==0x0001				Gebereingang Bit0=1	x=0x0..0x007F	→	a=2	
				GM-Mode Bit1..15	unrelevant			
				Gebertyp	P228	→	2.Stelle b plus Wert aus 6098	
Umrechnungsformalismus für x & 0x000A==0x000A oder x & 0x000C==0x000C oder x & 0x000D==0x000D				Geber 1 (a=1) Bit4..7=A, C, D	x=0x0..0x00FF	→	b=0	
Umrechnungsformalismus für x & 0x000B==0x000B oder x & 0x000E==0x000E				Bit4..7=B, E	x=0x0..0x00FF	→	b=1	
Umrechnungsformalismus für x & 0x000A==0x000A oder x & 0x000C==0x000C oder x & 0x000D==0x000D				Geber 2 (a=2) Bit0..3=A, C, D	x=0x0..0x00FF	→	b=0	

CANopen-Objekt	Index (HEX)	→	P-Nr.	V-Regler-Parameter	P-Nr.	→	Index (HEX)	Kommentar
Umrechnungsformalismus für x & 0x000B==0x000B oder x & 0x000E==0x000E				Bit0..3=B, E	x=0x0..0x00FF	→	b=1	
				Bit8..15	unrelevant			
homing_speeds	6099	→	P412 P443	Ref. Geschw.	P412 P443	→	6099	
	Sub. 01	→	P412		P412	→	Sub. 01	
speed_during_search_for_switch	x=0..2 ³² -1	→	y=x	Referenzgeschw.	x=1..13200	→	y=x	
	Sub. 02	→	P443		P443	→	Sub. 02	
homing_acceleration	609A	→	P413	Referenzbeschleunigung	P413	→	609A	
	x=0..2 ³² -1	→	y=x	Der interne Wertebereich von x=0.25..650.00 wird bereits auf einen INTEGER-Wert konvertiert.	x=25..65000	→	y=x	
max_acceleration	60C5	→	P419	Pos.-Beschl.	P419	→	60C5	
	x=0..2 ³² -1	→	y=x	Der interne Wertebereich von x=0.25..650.00 wird bereits auf einen INTEGER-Wert konvertiert.	x=25..65000	→	y=x	
max_acceleration	60C5	→	P426	Pos.-Beschl. 2	P426	→	60C5	
	x=0..2 ³² -1	→	y=x	Der interne Wertebereich von x=0.25..650.00 wird bereits auf einen INTEGER-Wert konvertiert.	x=25..65000	→	y=x	
max_deceleration	60C6	→	P420	Pos.-Verzög.	P420	→	60C6	
	x=0..2 ³² -1	→	y=x	Der interne Wertebereich von x=0.25..650.00 wird bereits auf einen INTEGER-Wert konvertiert.	x=25..65000	→	y=x	
position_control_parameter_set	60FB							
	Sub. 01 / read only			Status	P200	→	Sub. 01	
	x=0..2 ¹⁶ -1				x=0x0..0xFFFF	→	y=x	
	Sub. 02	→	P201	Mode	P201	→	Sub. 02	
	x=0..2 ¹⁶ -1	→	y=x		x=0x0..0x0007	→	y=x	
	Sub. 03	→	P202	Kv-Faktor	P202	→	Sub. 03	
	x=0..2 ¹⁶ -1	→	y=x		x=0..32000	→	y=x	
	Sub. 04	→	P205	Rev-Sollwert	P205	→	Sub. 05	
	x=0..2 ³² -1	→	y=x		x=0x0..0xFFFFFFFF	→	y=x	
	Sub. 05	→	P206	Phi-Sollwert	P206	→	Sub. 06	
	x=0..2 ³² -1	→	y=x		x=0x0..0xFFFFFFFF	→	y=x	
	Sub. 06	→	P207	N-Vorsteuerung	P207	→	Sub. 07	
	x=0..2 ¹⁶ -1	→	y=x	Der interne Wertebereich von x=0.00..125.00 wird bereits auf einen INTEGER-Wert konvertiert.	x=0..12500	→	y=x	
	Sub. 07 / read only			Schleppfehler	P210	→	Sub. 08	
	x=0..2 ³² -1				x=0x8000 0001..0xFFFF FFFF	→	y=x	
	Sub. 08	→	P213	Istwert	P213	→	Sub. 09	
	x=0..2 ¹⁶ -1	→	y=x	Der interne Wertebereich von x=0.00..50.00 wird bereits auf einen INTEGER-Wert konvertiert.	x=0..5000	→	y=x	
	Sub. 09	→	P218	Rev-Istwert	P218	→	Sub. 10	
	x=0..2 ³² -1	→	y=x		x=0x0..0xFFFFFFFF	→	y=x	

CANopen-Objekt	Index (HEX)	→	P-Nr.	V-Regler-Parameter	P-Nr.	→	Index (HEX)	Kommentar
	Sub. 0A	→	P219	Phi-Istwert	P219	→	Sub. 11	
	x=0..2 ³² -1	→	y=x		x=0x0..0xFFFFFFFF	→	y=x	
	Sub. 0B	→	P220	N-Vorst.-Ausgang	P220	→	Sub. 12	
	x=-32768..32767	→	y=x	Der interne Wertebereich von x=-100.00..100.00 wird bereits auf einen INTE-GER-Wert konvertiert.	x=-10000..10000	→	y=x	
	Sub. 0C / read only			Modul-State	P400	→	Sub. 21	
	x=0..2 ¹⁶ -1				x=0x0..0xFFFF	→	y=x	
	Sub. 0D	→	P401	Akt. Satz Nummer	P401	→	Sub. 22	
	x=0..2 ⁸ -1	→	y=x		x=1..2	→	y=x	
	Sub. 0E	→	P406	Modus (/POS98/ S.14)	P406	→	Sub. 23	
	x=0..2 ¹⁶ -1	→	y=x		x=0x0000..FFFF	→	y=x	
	Sub. 0F	→	P432	Referenzpunkt	P432	→	Sub. 24	
	x=0..2 ³² -1	→	y=x		x=0x0000..FFFF	→	y=x	
	Sub. 10	→	P434	Modus Schalter	P434	→	Sub. 25	
	x=0..2 ¹⁶ -1	→	y=x		x=0x0000..FFFF	→	y=x	
	Sub. 11 / read only			Soll-Geschw.	P438	→	Sub. 26	
	x=-32768..32767				x=-13200..13200	→	y=x	
	Sub. 12	→	P441	Verschleiß	P441	→	Sub. 27	
	x=0..2 ¹⁶ -1	→	y=x		x=0..8191	→	y=x	
	Sub. 13	→	P444	Clip-Umgebung 1	P444	→	Sub. 28	
	x=0..2 ³² -1	→	y=x		x=0x1..0xFFFFFFFF	→	y=x	
	Sub. 14	→	P445	Clip-Umgebung 2	P445	→	Sub. 29	
	x=0..2 ³² -1	→	y=x		x=0x1..0xFFFFFFFF	→	y=x	
	Sub. 15	→	P416	Zielangabe 1	P416	→	Sub. 41	
	x=-128..127	→	y=x		x=-2..2	→	y=x	
	Sub. 16	→	P423	Zielangabe 2	P423	→	Sub. 42	
	x=-128..127	→	y=x		x=-2..2	→	y=x	
digital_inputs	60FD / read only			Zustand Schalter	P433	→	60FD	
Umrechnungsformalismus für x & 0x00000003==0x00000002	x=0..2 ³² -1				x=0x0..0xFFFF	→	y=x-1	
Umrechnungsformalismus für x & 0x00000003==0x00000001	x=0..2 ³² -1				x=0x0..0xFFFF	→	y=x+1	
Umrechnungsformalismus für sonstigen Fall	x=0..2 ³² -1				x=0x0..0xFFFF	→	y=x	
negative limit switch	Bit 0			Zustand Endschalter pos.	Bit 0	→	Bit 1	
positive limit switch	Bit 1			Zustand Endschalter neg.	Bit 1	→	Bit 0	
home switch	Bit 2			Zustand Nullpunktumschalter	Bit 2	→	Bit 2	
interlock	Bit 3			reserviert	Bit 3..15			
reserved	Bit 4..15							
Man. specific	Bit 16..31							

Objekte 2XXX_{hex}

Die Umsetzungen der übrigen Objekte erfolgt nach DS301 im herstellerspezifischen Teil des Objektverzeichnisses. Dabei werden zur Basis 2000_{hex} die jeweiligen Parameternummern addiert (2000_{hex} + V-Parameter-Nummer ⇒ Objektindex).

Beispiel:

Zugriff auf *P126 Komm.-quelle* erfolgt über das Objekt 2126_{hex} (2000_{hex} + 126 ⇒ 2126_{hex}).

Die Wertebereiche sind bei den Objekten 2XXX_{hex} identisch zu denen der korrespondierenden V-Regler-Parameter. Die Objekte 2XXX_{hex} besitzen ausschließlich den Subindex 00_{hex}.

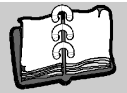
Die Wertumsetzung erfolgt so, wie in der V-Regler-Dokumentation beschrieben. Für Schreibzugriffe muß der zu übertragende Wert mit $10^{\text{Anzahl der Dezimalstellen}}$ multipliziert werden. Für Leszugriffe muß der zurückgelieferte durch $10^{\text{Anzahl der Dezimalstellen}}$ dividiert werden.

Beispiel:

Ist der Wertebereich des V-Regler-Parameters –100.00...+100.00, dann ergibt sich ein Wertebereich für Zugriffe über

CANopen von –10000...+10000.

V-Regler-Parameter, die Bitmuster oder Hexadezimalwerte verarbeiten (z.B. Modus- oder Status-Parameter), werden 1:1 umgesetzt.



Abbildungsverzeichnis

Technische Merkmale	6
Konzept der CANopen-Anschaltung	6
Objektbuffer für CANopen-Telegramme	7
Profilstruktur von CANopen.....	8
Struktur des CANopen-Objektverzeichnisses	8
PDO Übertragungsarten	9
Aufbau des Identifiers	10
Peer-to-Peer Objekte nach dem Predefined Connection Set	10
Frontansicht V-Regler mit Optionskarte CANopen	13
Stecker X 35 und Buchse X36	14
Verbindung der Busteilnehmer.....	14
Busabschluss-Stecker.....	14
DIP-Schalter S40	15
Beispiele für die Adresseinstellung	16
Einstellung der Baudrate.....	16
Kommunikationsablauf.....	19
Zeitscheibenverfahren (siehe auch Profibus-Dokumentation)	20
Verarbeitungszeiten für Empfangs- und Sendende PDOs.....	21
Zustandsmaschine der Kommunikation	31
NMT-Telegramm zur Steuerung der Kommunikationszustände	32
Command Specifier und deren Wirkung im Slave	33
Telegrammaufbau Enter_Pre-Operational_State für Knoten 16.....	33
Boot Up Telegramm von Knoten 1	34
Node Guarding Protokoll.....	34
SDO-Telegramm	36
Initiate SDO Download Protocol.....	37
Aus verschiedenen Datenbreiten resultierende Command Specifier.....	37
Initiate SDO Download Protocol für 8 Bit Nutzdaten.....	38
Initiate SDO Download Protocol für 16 Bit Nutzdaten.....	39
Initiate SDO Download Protocol für 32 Bit Nutzdaten.....	39
Initiate SDO Upload Expedited	40
Aus verschiedenen Datenbreiten resultierende Command Specifier.....	40
Initiate SDO Upload Protocol für 8 Bit Nutzdaten	41
Initiate SDO Upload Protocol für 16 Bit Nutzdaten	42
Initiate SDO Upload Protocol für 32 Bit Nutzdaten	42
Upload SDO Segmented Protocol	43
Initiate SDO Upload für Objekt 1008hex	44
Erstes Segment beim Lesen von Objekt 1008hex	44
Letztes Segment beim Lesen von Objekt 1008h	45
Abbruch Meldungen bei SDO-Zugriffen	45
Abort Telegramm	46
Kommunikationszyklus.....	47
PDO-Mapping	49
Erstes zu mappendes Objekt.....	52
Zweites zu mappendes Objekt.....	52
Drittes zu mappendes Objekt.....	53
Anzahl der zu mappenden Objekte.....	53
Erstes zu mappendes Objekt.....	54
Zweites zu mappendes Objekt.....	54
Anzahl der zu mappenden Objekte.....	55
Telegrammaufbau für Mapping.....	55
Aufbau des Emergency-Telegramms.....	60
EMCY-Telegramm für Geberfehler 0304hex	60
EMCY-Telegramm für Error Reset / No Error	60



Stichwortverzeichnis

A		INITIALISIERUNG	10, 32
Abort Codes	45	K	
Abschlusswiderstand	14	Kommunikationsbeziehungen	50
Adresseinstellung	15	Kommunikationsprofil	8
Aktualisierungszeiten	57	L	
Arbitrierung	7	Lieferumfang	5
Asynchrone PDO	50	M	
Aufbau der Optionskarte	6	Mapping	9, 48
B		Module-ID	10
BAPS	57	Multiplexor M	36
Baudrateneinstellung	16	N	
Baumüller	11	Netzwerk-Management	9
Bedarfsdatenkanal	5	Netzwerkmanagement	31
Bedarfsdatenkommunikation	19	NMT	9
Begriffe		Node Guarding	26, 27, 34
Definition	5	O	
Broadcast	7	Objektbuffer	7
C		Objekte	23
COB-ID	50	Objektindex	23
Command Specifier	32	Objektverzeichnis	8, 48
CSMA/CA-Verfahren	7	OPERATIONAL	10, 19, 32
D		P	
Datenübertragung	5	PDO	5, 9, 48
Default-Mapping	17	asynchrone	9, 50
Dezentralisierung	5	synchrone	9, 50
DIP-Schalter	15	Pegel	
Dummy-Mapping	49	dominanter	7
Dummy-Objekt	49, 54, 57	rezessiver	7
E		PRE-OPERATIONAL	10, 32
EDS-Datei	10	Prozess Daten Objekte	48
EMCY-Telegramm	27	Prozessdaten	9
Ereignisgesteuert	50, 51	Prozessdatenkanäle	5
Expedited Transfer	9	R	
F		Remoteframes	34
Full-CAN-Konzept	6	S	
G		SDO	5, 9, 35
Gefahr	11	SDO-Server	37
Geräteprofile	8	Segmented Transfer	9
H		Service Daten Objekte	35
Hammingdistanz	8	Slave-Adresse	15
Hardware-Version	26	Software-Reset	33
HINWEIS	12	Software-Version	26
I		Statuswort	57
Identifiers	10	Steuer-Objekt	50
Index	9	Steuerwort	57
Infozeichen	12	STOPPED	10, 32



Stichwortverzeichnis

Subindex	9
SYNC	47
Synchrone PDO	50
Synchronisation	47
SYNC-Telegramm	25

T

Technische Merkmale	6
Timer	51
Togglebit	35, 43
Triggerbedingung	50
Triggertyp	28

V

Verarbeitungszeiten	21
Vorsicht	12
V-Regler-Fehler	60

W

Warnung	12
---------	----

Z

Zeitgesteuert	51
Zeitscheibenverfahren	20
Zykluszeit	19