

# Applikationshandbuch

Sprache **Deutsch**  
Original  
Dokument-Nr. 5.14006.06  
Artikel-Nr. 00450922  
Stand 26.07.2019

be in motion

be in motion



  
**BAUMÜLLER**

**b maXX**

**CoE und  
POWERLINK für  
b maXX 2500 / 3300 /  
5000 und  
CANopen für  
b maXX 3300 / 5000**

<b>D</b>	5.14006.06
----------	------------

**Vor Beginn aller Arbeiten Betriebsanleitung lesen!**

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

© **Baumüller Nürnberg GmbH**

Ostendstr. 80 - 90  
90482 Nürnberg  
Deutschland

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

E-Mail: [mail@baumueller.de](mailto:mail@baumueller.de)  
Internet: [www.baumueller.de](http://www.baumueller.de)



# Inhaltsverzeichnis

<b>1 Allgemeines</b>	<b>7</b>
1.1 Informationen zum Applikationshandbuch CANopen / CoE / POWERLINK für b maXX 2500 / 3300 / 5000	7
1.2 Symbolerklärung	8
1.3 Haftungsbeschränkung	9
1.4 Urheberschutz	9
1.5 Mitgeltende Unterlagen	10
1.6 Gewährleistungsbestimmungen	10
1.7 Kundendienst	10
1.8 Verwendete Begriffe	10
<b>2 Grundlegende Sicherheitshinweise</b>	<b>11</b>
2.1 Gefahrenhinweise und Gebote	11
2.2 Infozeichen	11
<b>3 Objektverzeichnis</b>	<b>13</b>
3.1 Kommunikationsprofil	14
3.2 CANopen Geräteprofil CiA® 402	16
3.2.1 Kurze Übersicht	16
3.2.2 Betriebsarten und Feldbusobjekte	17
3.3 Herstellerspezifische Objekte	21
3.3.1 Arrayparameter	21
3.3.2 File Transfer over EtherCAT	23
<b>4 Kommunikation zum b maXX Regler</b>	<b>25</b>
4.1 Kommunikationsablauf	25
4.2 Parametrierung Feldbus Kommunikationszeiten	26
<b>5 Konfigurationsmöglichkeiten des Feldbus Slave</b>	<b>29</b>
5.1 Netzwerkeinstellungen für EoE (Ethernet over EtherCAT)	29
5.2 Sprachauswahl Online CoE-Objektverzeichnis	29
5.3 Factor Group nach CiA®402	30
5.4 CANopen Offset	32
5.5 Referenzfahrt für Positionierung erforderlich	32
5.6 Varianten der Positionierung, abhängig vom Zielmodus (Parameter 118.10)	33
<b>6 Grundlagen CAN / CANopen</b>	<b>35</b>
6.1 Literatur zum Thema CAN	35
6.2 Grundlagen CAN	36
<b>7 CANopen am b maXX 2500 / 3300 / 5000</b>	<b>39</b>
7.1 Allgemeines	39
7.2 Adresseinstellung	39
7.3 EDS-Datei	39
7.4 Diagnose	40
7.5 Datenaustausch und Parametrierung	40
7.6 Verzeichnis der Objekte zur Kommunikationssteuerung	41
7.7 Netzwerkmanagement (NMT)	46
7.7.1 Zustandsmaschine der Kommunikation	46
7.7.2 Telegramme	48



# Inhaltsverzeichnis

7.7.2.1	Zustandssteuerung	48
7.7.2.2	Boot Up	49
7.7.3	Node Guarding	49
7.8	Bedarfsdaten (SDO)	51
7.8.1	Telegrammaufbau	51
7.8.2	Arten des SDO-Transfers	52
7.8.3	Objekt schreiben	52
7.8.4	Objekt lesen	54
7.8.4.1	Expedited Transfer	54
7.8.4.2	Segmented Transfer	57
7.8.5	Fehlerreaktionen	59
7.9	Synchronisation (SYNC)	60
7.10	Prozessdaten (PDO)	61
7.10.1	PDO-Mapping	61
7.10.2	Kommunikationsbeziehung über PDO	65
7.10.3	Beispiel für das PDO-Mapping	68
7.10.4	Eintrag in die Feldbus Prozessdaten des Reglers	79
<b>8</b>	<b>Grundlagen EtherCAT</b>	<b>81</b>
8.1	Literatur zum Thema EtherCAT	81
8.2	Grundlagen EtherCAT	81
8.2.1	Topologie Daten	82
8.2.2	Rahmenaufbau	83
8.2.3	Geräteprofile	84
8.2.4	EtherCAT Frame-Struktur	84
8.2.5	EtherCAT Kommunikationszustände	87
8.2.6	Ethernet over EtherCAT (EoE) - TCP/IP- Tunneling über EtherCAT	90
<b>9</b>	<b>CoE am b maXX 2500 / 3300 / 5000</b>	<b>91</b>
9.1	Allgemeines	91
9.2	Adresseinstellung	91
9.3	XML-Datei	91
9.4	Diagnose	91
9.5	Datenaustausch und Parametrierung	92
9.6	Verzeichnis der Objekte zur Kommunikationssteuerung	92
9.7	Bedarfsdaten (SDO)	96
9.7.1	Telegrammaufbau nach CANopen	98
9.7.2	Arten des SDO-Transfers	99
9.7.3	Fehlerreaktionen	99
9.8	Prozessdaten (PDO)	100
9.8.1	PDO-Mapping	101
9.8.2	Synchronisation (SYNC)	103
9.8.3	Beispiel für das PDO-Mapping	104
9.8.4	Eintrag in die Feldbus-Prozessdaten des Reglers	107
<b>10</b>	<b>Grundlagen POWERLINK</b>	<b>109</b>
10.1	Literatur zum Thema POWERLINK	109
10.2	POWERLINK Allgemeines	109
<b>11</b>	<b>POWERLINK am b maXX 2500 / 3300 / 5000</b>	<b>113</b>
11.1	Allgemeines	113
11.2	Adresseinstellung	113
11.3	XDD-Datei	113
11.4	Diagnose	114
11.5	Datenaustausch und Parametrierung	114



11.6	Verzeichnis der Objekte zur Kommunikationssteuerung . . . . .	114
11.7	Netzwerkmanagement (NMT) . . . . .	118
11.8	Bedarfsdaten (SDO) . . . . .	119
11.8.1	Telegrammaufbau SoA . . . . .	120
11.8.2	Telegrammaufbau ASnd . . . . .	121
11.8.3	Fehlerreaktionen . . . . .	121
11.9	Synchronisation (SYNC) . . . . .	123
11.9.1	Telegrammaufbau SoC . . . . .	123
11.10	Prozessdaten (PDO) . . . . .	124
11.10.1	Telegrammaufbau PReq und PRes . . . . .	124
11.10.2	PDO-Mapping . . . . .	125
11.11	Konfigurationsbeispiel mit einer B&R X20 SPS . . . . .	128
11.12	PollResponse Chaining (ab Version 1.10) . . . . .	130
11.13	Ethernet Kommunikation (ab Version 1.10) . . . . .	131
11.13.1	Direkte Ethernet Kommunikation . . . . .	131
11.13.1.1	Konfiguration der IP Einstellungen mit ProDrive . . . . .	131
11.13.2	Kommunikation mit ProDrive . . . . .	132
11.13.3	Indirekte Ethernet Kommunikation . . . . .	133
11.14	Automatische Knotennummernvergabe (Dynamic Node Allocation) (ab Version 1.10) . . . . .	134
<b>Anhang A - Abkürzungen . . . . .</b>		<b>135</b>
<b>Anhang B - Kurzreferenz . . . . .</b>		<b>137</b>
B.1	2000er / 4000er Objektnummern ( <b>herstellerspezifische Objekte</b> ) . . . . .	137
B.2	6000er Objektnummern (Device Profile CiA <sup>®</sup> 402) . . . . .	139
<b>Anhang C - Umsetzungstabellen . . . . .</b>		<b>143</b>
<b>Anhang D - Technische Daten: POWERLINK Controlled Node . . . . .</b>		<b>159</b>
D.1	Technische Merkmale . . . . .	159
D.2	Datenkanäle zum <b>b maXX 5000</b> -Regler . . . . .	159
<b>Abbildungsverzeichnis . . . . .</b>		<b>161</b>
<b>Stichwortverzeichnis . . . . .</b>		<b>163</b>
<b>Revisionsübersicht . . . . .</b>		<b>165</b>



# 1

## ALLGEMEINES

### 1.1 Informationen zum Applikationshandbuch CANopen / CoE / POWERLINK für b maXX 2500 / 3300 / 5000

---

Dieses Applikationshandbuch gibt wichtige Hinweise zum Umgang mit dem Gerät. Voraussetzung für sicheres Arbeiten ist die Einhaltung aller angegebenen Sicherheitshinweise und Handlungsanweisungen.

Darüber hinaus sind die für den Einsatzbereich des Gerätes geltenden örtlichen Unfallverhütungsvorschriften und allgemeinen Sicherheitsbestimmungen einzuhalten.

Vor Beginn sämtlicher Arbeiten an dem Gerät ist die Betriebsanleitung, insbesondere das Kapitel Sicherheitshinweise, vollständig zu lesen. Die Betriebsanleitung ist Produktbestandteil und muss in unmittelbarer Nähe des Gerätes für das Personal jederzeit zugänglich aufbewahrt werden.

Für die Inbetriebnahme des Gerätes ist außerdem das Parameterhandbuch zu verwenden. Das Parameterhandbuch enthält Informationen zu den Parametern des Gerätes.

Das Applikationshandbuch CANopen / CoE / POWERLINK gibt Informationen zur Konfiguration und Inbetriebnahme in einem CANopen, EtherCAT bzw. POWERLINK Netzwerk von Geräten b maXX 2500 / 3300 / 5000, für Regler-Firmware ab Version 01.08.

### 1.2 Symbolerklärung

#### Warnhinweise

Warnhinweise sind in dieser Betriebsanleitung durch Symbole gekennzeichnet. Die Hinweise werden durch Signalworte eingeleitet, die das Ausmaß der Gefährdung zum Ausdruck bringen.

Die Hinweise unbedingt einhalten und umsichtig handeln, um Unfälle, Personen- und Sachschäden zu vermeiden.



#### **GEFAHR!**

...weist auf eine unmittelbar gefährliche Situation hin, die zum Tod oder zu schweren Verletzungen führt, wenn sie nicht gemieden wird.



#### **WARNUNG!**

...weist auf eine möglicherweise gefährliche Situation hin, die zum Tod oder zu schweren Verletzungen führen kann, wenn sie nicht gemieden wird.



#### **VORSICHT!**

...weist auf eine möglicherweise gefährliche Situation hin, die zu geringfügigen oder leichten Verletzungen führen kann, wenn sie nicht gemieden wird.



#### **ACHTUNG!**

...weist auf eine möglicherweise gefährliche Situation hin, die zu Sachschäden führen kann, wenn sie nicht gemieden wird.

#### Empfehlungen



#### **HINWEIS!**

...hebt nützliche Tipps und Empfehlungen sowie Informationen für einen effizienten und störungsfreien Betrieb hervor.



### 1.3 Haftungsbeschränkung

Alle Angaben und Hinweise in dieser Betriebsanleitung wurden unter Berücksichtigung der geltenden Normen und Vorschriften, dem Stand der Technik sowie unserer langjährigen Erkenntnisse und Erfahrungen zusammengestellt.

Der Hersteller übernimmt keine Haftung für Schäden aufgrund:

- Nichtbeachtung der Betriebsanleitung
- nichtbestimmungsgemäßer Verwendung
- Einsatz von nicht ausgebildeten Personal

Der tatsächliche Lieferumfang kann bei Sonderausführungen, Inanspruchnahme zusätzlicher Bestelloptionen oder aufgrund neuester technischer Änderungen von den hier beschriebenen Erläuterungen und Darstellungen abweichen.

Der Benutzer trägt die Verantwortung für die Durchführung von Service und Inbetriebnahme gemäß den Sicherheitsvorschriften der geltenden Normen und allen anderen relevanten staatlichen oder örtlichen Vorschriften betreffend Leiterdimensionierung und Schutz, Erdung, Trennschalter, Überstromschutz usw.

Für Schäden, die bei der Montage oder beim Anschluss entstehen, haftet derjenige, der die Montage oder Installation ausgeführt hat.

### 1.4 Urheberschutz

Die Betriebsanleitung vertraulich behandeln. Sie ist ausschließlich für die mit dem Gerät beschäftigten Personen bestimmt. Die Überlassung der Betriebsanleitung an Dritte ohne schriftliche Genehmigung des Herstellers ist unzulässig.



#### HINWEIS!

Die inhaltlichen Angaben, Texte, Zeichnungen, Bilder und sonstige Darstellungen sind urheberrechtlich geschützt und unterliegen den gewerblichen Schutzrechten. Jede missbräuchliche Verwertung ist strafbar.

**CiA® und  
CANopen®**

ist eine eingetragene Marke des CAN in Automation e.V.  
90429 Nürnberg, Deutschland

**EtherCAT®**

ist eine eingetragene Marke der Beckhoff Automation GmbH,  
33415 Verl, Deutschland

**POWERLINK®**

ist ein eingetragenes Markenzeichen der Ethernet POWERLINK  
Standardization Group, 15370 Fredersdorf, Deutschland

**b maXX®**

ist ein eingetragenes Markenzeichen von Baumüller Nürnberg  
GmbH, 90482 Nürnberg, Deutschland

## 1.5 Mitgeltende Unterlagen

---



### HINWEIS!

Bitte beachten Sie, dass Baumüller nicht verantwortlich ist zu überprüfen, ob durch den anwendungsspezifischen Einsatz der Baumüller Produkte/Komponenten oder der Ausführungen etwaige (Schutz-) Rechte Dritter verletzt werden.

## 1.5 Mitgeltende Unterlagen

---

Betriebsanleitung b maXX 3300 (5.11018) bzw. Betriebsanleitung b maXX 5000 (5.09021) und

Parameterhandbuch b maXX 3300 (5.12001) bzw. Parameterhandbuch b maXX 5000 (5.09022) in der jeweils aktuell geltenden Version.

## 1.6 Gewährleistungsbestimmungen

---

Die Gewährleistungsbestimmungen befinden sich als Dokument in den Verkaufsunterlagen.

Zulässig ist der Betrieb der hier beschriebenen Geräte gemäß den genannten Methoden/Verfahren / Maßgaben. Alles andere, z. B. auch der Betrieb in Einbaulagen, die hier nicht dargestellt werden, ist nicht zulässig und muss im Einzelfall geklärt werden. Werden die Geräte anders als hier beschrieben betrieben, so erlischt jegliche Gewährleistung.

## 1.7 Kundendienst

---

Für technische Auskünfte steht unser Kundendienst zur Verfügung. Hinweise über den zuständigen Ansprechpartner sind jederzeit per Telefon, Fax, E-Mail oder über das Internet abrufbar.

## 1.8 Verwendete Begriffe

---

Für dieses Baumüller-Produkt wird in dieser Dokumentation auch der Begriff „Gerät“ oder die Gerätebezeichnung BM3300 oder BM5000 verwendet. Eine Liste der verwendeten Abkürzungen siehe [▶Anhang A - Abkürzungen◀](#) ab Seite 135.

# GRUNDLEGENDE SICHERHEITSHINWEISE

In diesem Kapitel werden die Gefahren, die beim Parametrieren des Baumüller **b maXX 5000**-Reglerteils auftreten können und die Bedeutung des Infozeichens erklärt.

## 2.1 Gefahrenhinweise und Gebote

---



### WARNUNG!

#### Gefahr durch Änderung von Parametereinstellungen!

- Die Gefahr ist: **mechanische und elektrische Einwirkung**.  
Die Änderung von Parametern beeinflusst das Verhalten des Baumüller-Geräts und somit das Verhalten der Anlage und ihrer Komponenten. Wenn die Einstellungen der Parameter verändert werden, kann ein gefährliches Verhalten der Anlage und/oder ihrer Komponenten auftreten.
- Nach jeder Änderung der Parametereinstellungen ist eine Inbetriebnahme unter Beachtung aller Sicherheitshinweise und Sicherheitsvorschriften durchzuführen.

## 2.2 Infozeichen

---



### HINWEIS!

Der hier angegebene Hinweis ist eine besonders wichtige Information.



## OBJEKTVERZEICHNIS

Das zentrale Element eines Geräteprotokolls ist das Objektverzeichnis. Als Grundlage dafür dient das CANopen Objektverzeichnis.

Index	Objekt
0x0000	Nicht benutzt
0x0001 – 0x001F	Statische Datentypen
0x0020 – 0x003F	Komplexe Datentypen
0x0040 – 0x005F	Herstellerspezifische Datentypen
0x0060 – 0x007F	Geräteprofilsspezifische Statische Datentypen
0x0080 – 0x009F	Geräteprofilsspezifische Dynamische Datentypen
0x00A0 – 0x03FF	Reserviert
0x0400 – 0x041F	POWERLINK Spezifische Statische Datentypen
0x0420 – 0x04FF	POWERLINK Spezifische Dynamische Datentypen
0x0500 – 0x0FFF	Reserviert
0x1000 – 0x1FFF	Bereich für das Kommunikationsprofil
0x2000 – 0x3FFF	Bereich für herstellerspezifische Objekte Achse 1
0x4000 – 0x5FFF	Bereich für herstellerspezifische Objekte Achse 2
0x6000 – 0x9FFF	Bereich für das Geräteprofil
0xA000 – 0xBFFF	Bereich für das Interface-Profil
0xC000 – 0xFFFF	Reserviert

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

### 3.1 Kommunikationsprofil

Für die unterschiedlichen Bussysteme werden teilweise auch unterschiedliche Kommunikationsobjekte verwendet. Für Geräte mit Doppelachse gibt es zusätzliche Objekte.

Index	Bezeichnung	CANopen	CoE	Powerlink
0x1000	Device type	X	X	X
0x1001	Error register	X	X	X
0x1002	Vendor status register	X		
0x1005	COB-ID SYNC	X		
0x1006	Communication cycle period	X		X
0x1007	Synchronous window length	X		
0x1008	Device name	X	X	X
0x1009	Hardware version	X	X	X
0x100A	Software version	X	X	X
0x100C	Guard time	X		
0x100D	Life time factor	X		
0x1017	Producer heartbeat time	X		
0x1018	Identity	X	X	X
0x1020	Verify configuration			X
0x1030	Interface group			X
0x1050	Relative Latency Difference			X
0x1300	Sequence layer timeout			X
0x1400	Rx communication parameter 0 axis 1	X		X
0x1401	Rx communication parameter 1 axis 1	X		
0x1402	Rx communication parameter 2 axis 1	X		
0x1403	Rx communication parameter 3 axis 1	X		
0x1440	Rx communication parameter 0 axis 2	X		
0x1441	Rx communication parameter 1 axis 2	X		
0x1442	Rx communication parameter 2 axis 2	X		
0x1443	Rx communication parameter 3 axis 2	X		
0x1600	Rx mapping parameter 0 axis 1	X	X	X
0x1601	Rx mapping parameter 1 axis 1	X		
0x1602	Rx mapping parameter 2 axis 1	X		
0x1603	Rx mapping parameter 3 axis 1	X		
0x1640	Rx mapping parameter 0 axis 2	X		
0x1641	Rx mapping parameter 1 axis 2	X		
0x1642	Rx mapping parameter 2 axis 2	X		

Index	Bezeichnung	CANopen	CoE	Powerlink
0x1643	Rx mapping parameter 3 axis 2	X		
0x1700	Rx mapping parameter 1 axis 2		X	
0x1800	Tx communication parameter 0 axis 1	X		X
0x1801	Tx communication parameter 1 axis 1	X		
0x1802	Tx communication parameter 2 axis 1	X		
0x1803	Tx communication parameter 3 axis 1	X		
0x1840	Tx communication parameter 0 axis 2	X		
0x1841	Tx communication parameter 1 axis 2	X		
0x1842	Tx communication parameter 2 axis 2	X		
0x1843	Tx communication parameter 3 axis 2	X		
0x1A00	Rx mapping parameter 0 axis 1	X	X	X
0x1A01	Rx mapping parameter 1 axis 1	X		
0x1A02	Rx mapping parameter 2 axis 1	X		
0x1A03	Rx mapping parameter 3 axis 1	X		
0x1A40	Rx mapping parameter 0 axis 2	X		
0x1A41	Rx mapping parameter 1 axis 2	X		
0x1A42	Rx mapping parameter 2 axis 2	X		
0x1A43	Rx mapping parameter 3 axis 2	X		
0x1B00	Rx mapping parameter 0 axis 2		X	
0x1C00	Sync manager type		X	
0x1C02	Cycle diagnosis		X	
0x1C0B	CN loss of SoC			X
0x1C0D	CN loss of PReq			X
0x1C0F	CN CRC error			X
0x1C12	Number of assigned RxPDOs	X	X	
0x1C13	Number of assigned TxPDOs	X	X	
0x1C14	CN loss of SoC tolerance			X
0x1C32	Sync manager output parameter		X	
0x1C33	Sync manager input parameter		X	
0x1E40	IP Address Table			X
0x1E4A	IP Group			X
0x1F81	Node Assignment			X
0x1F82	Feature flags			X
0x1F83	EPL version			X
0x1F8C	Current NMT state			X

## 3.2 CANopen Geräteprofil CiA<sup>®</sup> 402

Index	Bezeichnung	CANopen	CoE	Powerlink
0x1F8D	PRes Payload Limit List			X
0x1F93	EPL Node ID			X
0x1F98	Cycle timing			X
0x1F99	CN basic Ethernet timeout			X
0x1F9A	Host Name			X
0x1F9E	Reset Command			X

### 3.2 CANopen Geräteprofil CiA<sup>®</sup> 402

CANopen, CoE (CAN application protocol over EtherCAT) sowie POWERLINK unterstützen das CANopen Geräteprofil CiA<sup>®</sup> 402 (Device Profile for Drives and Motion Control).

Die folgenden Betriebsarten des CANopen Geräteprofils werden unterstützt.

#### 3.2.1 Kurze Übersicht

<b>Folgende Betriebsarten werden unterstützt, d. h. alle vorgeschriebenen Objekte sind vorhanden</b>	
Device Control	optionale Objekte nicht vollständig vorhanden
Homing Mode	optionale Objekte vollständig vorhanden
Profile Position Mode	optionale Objekte nicht vollständig vorhanden
Position Control Funktion	optionale Objekte nicht vollständig vorhanden
Profile Velocity Mode	optionale Objekte nicht vollständig vorhanden
Common Entries in the Object Dictionary (keine vorgeschriebenen Objekte vorhanden)	optionale Objekte nicht vollständig vorhanden
Velocity Mode	optionale Objekte nicht vollständig vorhanden
Cyclic Synchronous Position Mode	empfohlene Objekte nicht vollständig vorhanden
Cyclic Synchronous Velocity Mode	empfohlene Objekte nicht vollständig vorhanden
Cyclic Synchronous Torque Mode	empfohlene Objekte nicht vollständig vorhanden
Factor Group	optionale Objekte vollständig vorhanden

<b>Folgende Betriebsarten werden nicht unterstützt, d. h. mindestens ein vorgeschriebenes Objekt ist nicht vorhanden, es können auch optionale Objekte vorhanden sein</b>	
Profile Torque Mode	drei Objekte
Interpolated Position Mode	keine Objekte



## 3.2.2 Betriebsarten und Feldbusobjekte

<b>Betriebsart</b>		
Feldbus-Objekt-Nummer	vorgeschrieben / optional	Feldbus-Objekt-Name

<b>Device Control (Gerätekontrolle)</b> alle vorgeschriebenen und zum Teil optionale Objekte werden unterstützt		
0x6040	vorgeschrieben	Controlword
0x6041	vorgeschrieben	Statusword
0x605A	optional	Quick stop option code
0x605B	optional	Shutdown option code
0x605C	optional	Disable operation option code
0x6060	vorgeschrieben	Modes of operation
0x6061	vorgeschrieben	Modes of operation display

<b>Homing Mode (Referenzfahrt)</b> alle vorgeschriebenen Objekte vorhanden und alle optionalen Objekte werden unterstützt		
0x607C	optional	Home offset
0x6098	vorgeschrieben	Homing method
0x6099 SIX 0 = 2	vorgeschrieben	Homing speed
0x609A	optional	Homing acceleration
0x60B8	optional	Touch probe function
0x60B9	optional	Touch probe status
0x60BA	optional	Touch probe pos 1 pos value
0x60BB	optional	Touch probe pos 1 neg value
0x60BC	optional	Touch probe pos 2 pos value
0x60BD	optional	Touch probe pos 2 neg value

<b>Profile Position Mode (Positionierung)</b> alle vorgeschriebenen und zum Teil optionale Objekte werden unterstützt		
0x607A	vorgeschrieben	Target position
0x607D SIX 0 = 2	optional	Software position limit
0x6080	optional	Max motor speed
0x6081	vorgeschrieben	Profile velocity
0x6083	vorgeschrieben	Profile acceleration
0x6084	vorgeschrieben	Profile deceleration
0x6085	optional	Quick stop deceleration
0x6086	vorgeschrieben	Motion profile type

<b>Position Control Function (Positionierkontrolle)</b> alle vorgeschriebenen und zum Teil optionale Objekte werden unterstützt		
0x6063	optional	Position actual internal value
0x6064	vorgeschrieben	Position actual value
0x6067	optional	Position window
0x6068	optional	Position window time
0x60FB SIX 0=25	optional	Position control parameter set

<b>Profile Velocity Mode (Drehzahlregelung)</b> alle vorgeschriebenen und zum Teil optionale Objekte werden unterstützt		
0x6069	vorgeschrieben	Velocity sensor actual value
0x606A	vorgeschrieben	Sensor selection code
0x606B	vorgeschrieben	Velocity demand value
0x606C	vorgeschrieben	Velocity actual value
0x60FF	vorgeschrieben	Target velocity

<b>Common Entries in the Object Dictionary</b> (allgemeine Eingänge im Objektverzeichnis) keine vorgeschriebenen Objekte vorhanden, zum Teil werden optionale Objekte unterstützt		
0x6007	optional	Abort connection option code
0x603F	optional	Error code
0x6502	optional	Supported drive modes
0x6510 SIX 0= 13	optional	Drive date

<b>Velocity Mode (Drehzahlregelung)</b> alle vorgeschriebenen und zum Teil optionale Objekte werden unterstützt		
0x6042	vorgeschrieben	vl target velocity
0x6043	vorgeschrieben	vl velocity demand
0x6044	vorgeschrieben	vl control effort
0x6046 SIX 0 = 2	vorgeschrieben	vl velocity min max amount
0x604F	optional	vl ramp function time
0x6050	optional	vl slow down time

<b>Profile Torque Mode (Momentenregelung)</b> ein vorgeschriebenes und zwei optionale Objekte werden unterstützt		
0x6071	vorgeschrieben	Target torque
0x6072	optional	Max torque
0x6077	optional	Torque actual value

<b>Cyclic Synchronous Position Mode (Synchrone Positionierung)</b> alle vorgeschriebenen und zum Teil empfohlene Objekte werden unterstützt		
0x6064	vorgeschrieben	Position actual value
0x6077	empfohlen	Torque actual value
0x607A	vorgeschrieben	Target position
0x607D SIX 0 = 2	empfohlen	Software position limit
0x60B1	empfohlen	Velocity offset
0x60F4	empfohlen	Following error actual value

<b>Cyclic Synchronous Velocity Mode (Synchrone Geschwindigkeit)</b> alle vorgeschriebenen und zum Teil empfohlene Objekte werden unterstützt		
0x6064	vorgeschrieben	Position actual value
0x606C	empfohlen	Velocity actual value
0x6077	empfohlen	Torque actual value
0x60FF	vorgeschrieben	Target velocity

<b>Cyclic Synchronous Torque Mode (Synchrones Drehmoment)</b> alle vorgeschriebenen und zum Teil empfohlene Objekte werden unterstützt		
0x6064	empfohlen	Position actual value
0x6071	vorgeschrieben	Target torque
0x6077	vorgeschrieben	Torque actual value
0x60B2	empfohlen	Torque offset

<b>Factor Group (Gruppe der Benutzer Einheiten)</b> keine vorgeschriebenen Objekte vorhanden, alle optionalen Objekte werden unterstützt		
0x607E	optional	Polarity
0x608F SIX 0 = 2	optional	Position encoder resolution
0x6090 SIX 0 = 2	optional	Velocity encoder resolution
0x6091 SIX 0 = 2	optional	Gear ratio
0x6092 SIX 0 = 2	optional	Feed constant

### 3.3 Herstellerspezifische Objekte

Der Zugriff auf die herstellerspezifischen Objekte des b maXX 2500 / 3300 / 5000 erfolgt über den Bereich 0x2000 - 0x3FFF für die Achse 1 sowie den Bereich 0x4000 - 0x5FFF für die Achse 2 (nur b maXX 5000).

Die Zuordnung zwischen CANopen-Objektnummer und Reglerparameternummer ist in Anhang B beschrieben.

#### 3.3.1 Arrayparameter

Die Adressierung mittels CANopen Index und Subindex ist für die Array-Parameter des b maXX Reglers unzureichend. Daher sind zusätzliche herstellerspezifische Objekte definiert, um den Zugriff auf diese Parameter zu ermöglichen.

##### 0x2120 Fehlernummer

Subindex 0x01 ... 0x14

Anzeige der Fehlernummern aus Reglerparameter 100.3

Index	Subindex	Datentyp	Zugriff	Beschreibung
0x2120	0x00	UINT8	RO	Anzahl Subindizes
	0x01	UINT32	RO	Fehlernummer 0
	0x02	UINT32	RO	Fehlernummer 1
	0x03	UINT32	RO	Fehlernummer 2
	0x04	UINT32	RO	Fehlernummer 3
	0x05	UINT32	RO	Fehlernummer 4
	0x06	UINT32	RO	Fehlernummer 5
	0x07	UINT32	RO	Fehlernummer 6
	0x08	UINT32	RO	Fehlernummer 7
	0x09	UINT32	RO	Fehlernummer 8
	0x0A	UINT32	RO	Fehlernummer 9
	0x0B	UINT32	RO	Fehlernummer 10
	0x0C	UINT32	RO	Fehlernummer 11
	0x0D	UINT32	RO	Fehlernummer 12
	0x0E	UINT32	RO	Fehlernummer 13
0x0F	UINT32	RO	Fehlernummer 14	
0x10	UINT32	RO	Fehlernummer 15	
0x11	UINT32	RO	Fehlernummer 16	
0x12	UINT32	RO	Fehlernummer 17	
0x13	UINT32	RO	Fehlernummer 18	
0x14	UINT32	RO	Fehlernummer 19	

Die Subindizes von Objekt 0x2120 sind nur lesbar.

### 0x2121 Fehlerreaktionen

Fehlerreaktion zu Reglerfehlernummern aus Reglerparameter 100.4

Index	Subindex	Datentyp	Zugriff	Beschreibung
0x2121	0x00	UINT8	RO	Anzahl Subindizes
	0x01	UINT32	RW	Fehlercode
	0x02	UINT32	RO	Bestätigung Fehlercode
	0x03	INT32	RW	Fehlerreaktion

In Subindex 0x01 wird die Fehlernummer eingetragen, deren Fehlerreaktion geändert werden soll und in Subindex 0x03 die gewünschte Fehlerreaktion. Nach dem Schreiben von Objekt 0x2121.03 wird die Fehlerreaktion im Regler sofort aktualisiert.

Objekt 0x2121.02 dient lediglich als Bestätigung des korrektes Fehlereintrags in Objekt 0x2121.01 und ist nur lesbar. Falls die Fehlernummer im Regler unbekannt ist, steht in Objekt 0x2121.02 der Wert 0.

### 0x2125 Positioniersätze

Positioniersätze zu Reglerparameter 118.19

Index	Subindex	Datentyp	Zugriff	Beschreibung
0x2125	0x00	UINT8	RO	Anzahl Subindizes
	0x01	UINT32	RW	Positioniersatz
	0x02	UINT32	RW	Zielposition
	0x03	UINT16	RW	Zielmodus
	0x04	UINT32	RW	Geschwindigkeit
	0x05	UINT32	RW	Beschleunigung
	0x06	UINT32	RW	Verzögerung
	0x07	UINT32	RW	Ruck
	0x08	UINT16	RW	Verschleißzeit
	0x09	INT32	RW	Relative Zielposition

In Subindex 0x01 kann der Positioniersatz 1 bis 16 ausgewählt werden. Der Positioniersatz 0 ist der aktive Positioniersatz und kann nicht ausgewählt werden.

Durch das Schreiben von Subindex 0x02 bis 0x09 lassen sich die Daten der ausgewählten Positioniersätze ändern.

### 3.3.2 File Transfer over EtherCAT

Der b maXX 2500 / 3300 / 5000 CoE Slave unterstützt File Transfer über EtherCAT (FoE, die Übertragung von Dateien über den Feldbus EtherCAT). Dies ermöglicht z. B. Firmwareupdates oder das Aufspielen von Datensätzen. Die Voraussetzung dafür ist, dass der verwendete EtherCAT Master ebenfalls diese Funktion (FoE) unterstützt. Die Aktivierung des Firmwareupdates erfolgt durch Objekt 0x2300.

#### 0x2300 File transfer

Subindex	Name	Datentyp	Zugriff	Beschreibung
1	FoE file name	STRING	RO	FoE Dateiname
2	FoE file length	UINT32	RO	FoE Dateilänge
3	Command	UINT16	RW	0 – Abbrechen des Transfer oder Initialisieren der Internen Zustandsmaschine 1 – Speichern einer Datei in Baumüller FW Format (inkl. Datensätze) ins Flash 2 – Speichern einer Datei in Baumüller FW Format (incl. Datensätze) ins RAM
4	Status	UINT32	RO	RC_NO_ERROR = 0 (kein Fehler) RC_BUSY = 1 (Wartezustand) RC_ERROR = 2 (Allgemeiner Fehler) RC_DONE = 3 (Vorgang / Kommando beendet)
5	RC error	UINT32	RO	Baumüller RC Fehlermeldung
6	Extended info	UINT32	RO	Für Kommandos 1 bis 2 – Sendestatus in %
7	File Options	UINT32		Reserviert
8	File Type	UINT32		Reserviert
9	Axis Unit	UINT32		Nur für BM5800. 1-8: Achsmodul (AM); 0xFF: SE
10	File Source/ Destination	UINT32		Reserviert





# 4

## KOMMUNIKATION ZUM b maXX REGLER

In diesem Kapitel beschreiben wir die Datenkommunikation zwischen dem Felbus Slave und dem b maXX 2500 / 3300 / 5000 Regler.

### 4.1 Kommunikationsablauf

---

Der Feldbus Slave tauscht über ein Dual-Port-RAM Daten mit dem b maXX 2500 / 3300 / 5000 Regler aus. Dieser Datenaustausch erfolgt in einem bestimmten Zeitraster.

Der Feldbus Slave stößt die Kommunikation mit dem b maXX 2500 / 3300 / 5000 Regler an. Bei der Kommunikation werden zwei unterschiedliche Typen von Daten übertragen:

- Prozessdaten
- Bedarfsdaten

Die Prozessdaten werden immer zyklisch übertragen. In der Restzeit werden die Bedarfsdaten übertragen. Die Prozessdatenübertragung findet in einem einstellbaren Zeitraster, der Feldbus Zykluszeit statt. Dabei werden die Soll- und Istwerte mit einem Offset zum Intervall des Feldbus Zyklus übertragen. Das Synchronisierungstelegramm des Feldbus muss mit der Feldbus Zykluszeit (Parameter 131.18) übereinstimmen.

### 4.2 Parametrierung Feldbus Kommunikationszeiten

Zwischen dem Feldbus Slave und dem b maXX Regler können 16 Sollwerte und 16 Istwerte pro Achse als Prozessdaten in einem Kommunikationszyklus ausgetauscht werden. Welche Sollwerte und Istwerte ausgetauscht werden sollen, wird in den Mapping-Objekten über SDO durch den Feldbus Master eingestellt, siehe Kapitel [▷ Datenaustausch und Parametrierung ◁](#) ab Seite 114.

Die Einstellung der Kommunikationszeiten zwischen dem Feldbus Slave und b maXX Regler werden automatisch parametrieren. Anhand der vom Feldbus Master eingestellten Zykluszeit in Objekt 0x1006 (CANopen, POWERLINK) bzw. Objekt 0x1C32 SIX 2 (CoE) wird die Feldbus-Zykluszeit des Reglers im Parameter 131.18 gesetzt.

Wenn beim EtherCAT Distributed Clock aktiviert sind, dann wird Zykluszeit vom Distributed Clock Zeit übernommen.

Der b maXX Regler ruft in jedem Feldbuszyklus eine Kommunikationszeitscheibe auf, in der Prozessdatensollwerte und Prozessdatenistwerte übertragen werden.

Die Prozessdatensollwerte und Prozessdatenistwerte für jede Achse werden in der gleichen Kommunikationszeitscheibe übertragen. Die Übertragung erfolgt mit einem zeitlichen Versatz zur Synchronisierung. Dieser Sync Offset wird für die Soll- und Istwerte gemeinsam in Parameter 156.4 angegeben. Der Sync Offset wird automatisch auf die halbe Zykluszeit gesetzt:

Zykluszeit (0x1006)	Zykluszeit (0x1C32 SIX 02)	Feldbus Zykluszeit (131.18)	Sync Offset (156.4)
500 µs	500000 ns	500000 ns	250 µs
1000 µs	1000000 ns	1000000 ns	500 µs
2000 µs	2000000 ns	2000000 ns	1000 µs
4000 µs	4000000 ns	4000000 ns	2000 µs
8000 µs	8000000 ns	8000000 ns	4000 µs

Für die Synchronisation des Reglers mit einem externen Signal, z. B. dem Synchronisationssignal des Feldbus Masters, muss zusätzlich die Synchronisation im Regler aktiviert werden. Hierfür muss Bit 0 im Parameter 156.1 Synchronisation Modus gesetzt werden.

Status	
Status	Init
Einstellungen	Kommunikation
IP Adresse <input checked="" type="radio"/> Basis IP-Adresse + DIP Schalter Wert <input type="radio"/> Manuelle Eingabe MAC Adresse: 00-02-FB-FF-E7- Basis IP-Adresse: 192.168.0.0 Gateway: 0.0.0.0 DIP Schalterstellung: 0.0.0.1 Aktuelle IP-Adresse: 0.0.0.0 Slave Fehler-Code: 0 - kein Fehler Feldbus Zykluszeit: 1000 $\mu$ s Sprachauswahl des CoE-Objektverzeichnis: <input checked="" type="radio"/> Deutsch <input type="radio"/> English	Synchronisierung: Ein Status: Synchron <input type="checkbox"/> Manuelle Offset-Einstellung verwenden Sync Offset: 500,00 $\mu$ s Sync Toleranz: 20,00 $\mu$ s Feldbus Jitter: -0,18 $\mu$ s

Abbildung 1: ProDrive Feldbus-Slave

Fällt die Istwertübertragung zwischen Feldbus und Regler für mehr als zwei Feldbuszyklen aus, wird dies durch die Fehlernummer 1937 signalisiert. Fallen zwei Sollwerte aus, wird dies durch die Fehlernummer 1938 signalisiert.

Der Sync Offset kann auch manuell festgelegt werden. Hierbei muss im Parameter 156.1 Bit 1 gesetzt werden.

Die Einstellungen müssen im Datensatz des b maXX Reglers abgespeichert werden und der Regler muss neu gebootet werden.

**HINWEIS!**

Für die Synchronisation auf ein Feldbus-Signal muss die Synchronisation in Parameter 156.1 eingeschaltet werden.

**HINWEIS!**

Wird die zyklische Kommunikation unterbrochen, z. B. beim Neustart des Bus kann der Fehler 1937 oder 1938 auftreten.



# KONFIGURATIONSMÖGLICHKEITEN DES FELDBUS SLAVE

Durch Änderung der Slave Einstellungen in Parameter 131.9 kann das Verhalten des Feldbus Slave geändert werden.



## HINWEIS!

Geänderte Einstellungen führen zu einem geänderten Verhalten!

## 5.1 Netzwerkeinstellungen für EoE (Ethernet over EtherCAT)

---

Für EtherCAT Antriebe kann die Einstellung der IP Adresse durch die DIP Schalter am Gerät oder durch b maXX Parameter erfolgen.

- Bit 0**
- 0** IP Adresse = Basis IP-Adresse + DIP Schalter (131.12 + 131.13),  
Subnetzmaske = 255.255.0.0
  - 1** Auslesen der Netzwerkeinstellungen für EoE von Parametern 131.14, 131.16, 131.17

## 5.2 Sprachauswahl Online CoE-Objektverzeichnis

---

Bei CoE kann das Objektverzeichnis Online vom Master abgerufen werden. Die Sprache kann in den Slave Einstellungen zwischen Deutsch und Englisch geändert werden.

- Bit 1**
- 0** Deutsch
  - 1** Englisch

### 5.3 Factor Group nach CiA<sup>®</sup>402

(ab Firmware-Version 01.09)

Die Berechnung anhand der Factor Group erfolgt im b maXX Regler (siehe auch BM3000 Parameterhandbuch 5.12001 bzw. BM5000 Parameterhandbuch 5.09022). Die Berechnung muss in den Slave Einstellungen Parameter 131.9 aktiviert werden.

**Bit 14**

**0** Keine Berechnung nach CiA402 Factor Group

**1** Berechnung nach CiA402 Factor Group

Auf folgende Objekte wirkt sich die Factor Group aus:

Objekt Nummer	Normierung anhand Factor Group Objekt
0x6064	0x608F, 0x6091, 0x6092
0x6067	0x608F, 0x6091, 0x6092
0x606C	0x607E, 0x6090, 0x6091, 0x6092
0x607A	0x607E, 0x608F, 0x6091, 0x6092
0x607C	0x608F, 0x6091, 0x6092
0x607D SIX 1	0x607E, 0x608F, 0x6091, 0x6092
0x607D SIX 2	0x607E, 0x608F, 0x6091, 0x6092
0x6081	0x6090, 0x6091, 0x6092
0x6083	0x6090, 0x6091, 0x6092
0x6084	0x6090, 0x6091, 0x6092
0x6085	0x6090, 0x6091, 0x6092
0x6099 SIX 1	0x6090, 0x6091, 0x6092
0x6099 SIX 2	0x6090, 0x6091, 0x6092
0x609A	0x6090, 0x6091, 0x6092
0x60B1	0x607E, 0x6090, 0x6091, 0x6092
0x60BA	0x608F, 0x6091, 0x6092
0x60BB	0x608F, 0x6091, 0x6092
0x60BC	0x608F, 0x6091, 0x6092
0x60BD	0x608F, 0x6091, 0x6092
0x60F4	0x608F, 0x6091, 0x6092
0x60FF	0x607E, 0x6090, 0x6091, 0x6092

$$\text{Position encoder resolution (0x608F)} = \frac{\text{Encoder increments (0x608F.01)}}{\text{Motor revolutions (0x608F.02)}}$$

$$\text{Velocity encoder resolution (0x6090)} = \frac{\text{Encoder increments per second (0x6090.01)}}{\text{Motor revolutions per second (0x6090.02)}}$$

$$\text{Gear ratio (0x6091)} = \frac{\text{Motor revolutions (0x6091.01)}}{\text{Shaft revolutions (0x6091.02)}}$$

$$\text{Feed constant (0x6092)} = \frac{\text{Feed (0x6092.01)}}{\text{Shaft revolutions (0x6092.02)}}$$

**Polarity (0x607E):** Der Positionssollwert soll mit -1 multipliziert werden, wenn das Polarity Bit gesetzt ist. Das Polarity Bit hat keinen Einfluss auf den Homing Mode.

Bit 7	Bit 6	Bit 0 - 5
Position polarity	Velocity polarity	reserviert

Das Position Polarity Bit soll nur für Profile Position Mode (Lagezielvorgabe) und Cyclic Sync Position Mode (Lageregelung) verwendet werden. Das Velocity Polarity Bit soll nur für Profile Velocity Mode (Drehzahlregelung) und Cyclic Sync Velocity Mode (Drehzahlregelung) verwendet werden.

Alle Werte der Factor Group Objekte werden dimensionslos angegeben.

Die Normierung erfolgt folgendermaßen:

Schreiben von Werten von der Steuerung zum Regler:

$$\text{Positionswerte intern} = \frac{\text{Positionswerte} \cdot \text{Polarity (0x607E)} \cdot \text{Position encoder resolution (0x608F)} \cdot \text{Gear ratio (0x6091)}}{\text{Feed constant (0x6092)}}$$

$$\text{Drehzahlwerte intern} = \frac{\text{Drehzahlwerte} \cdot \text{Polarity (0x607E)} \cdot \text{Velocity encoder resolution (0x6090)} \cdot \text{Gear ratio (0x6091)}}{\text{Feed constant (0x6092)}}$$

Lesen von Werten vom Regler zur Steuerung:

$$\text{Positionswerte} = \frac{\text{Positionswerte intern} \cdot \text{Feed constant (0x6092)}}{\text{Position encoder resolution (0x608F)} \cdot \text{Gear ratio (0x6091)} \cdot \text{Polarity (0x607E)}}$$

$$\text{Drehzahlwerte} = \frac{\text{Drehzahlwerte intern} \cdot \text{Feed constant (0x6092)}}{\text{Velocity encoder resolution (0x6090)} \cdot \text{Gear ratio (0x6091)} \cdot \text{Polarity (0x607E)}}$$

### 5.4 CANopen Offset

Abbildung der Zahlenskala UINT32 auf INT32 (CANopen-Mode). Beim Schreiben/Lesen auf einige FB Objekte wird intern auf dem Feldbus Slave je nach Richtung ein Offset von  $2^{31}$  dazu addiert bzw. subtrahiert.

- Bit 16**
- 0** Zahlenskalawandlung von UINT32 auf INT32, bei der Positionierung wird ein Offset von  $2^{31}$  je nach Richtung auf die zugehörigen Feldbusobjekte addiert/subtrahiert
  - 1** es wird kein Offset addiert/subtrahiert  
Folgende Objekte sind davon betroffen:  
0x6062, 0x6064, 0x607A, 0x607C, 0x607D SIX 1, 0x607D SIX 2

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

### 5.5 Referenzfahrt für Positionierung erforderlich

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

Deaktiviert:

Zum Betrieb in der Betriebsart Positionieren ist keine Referenzfahrt erforderlich.

Aktiviert:

Wenn der Antrieb in Betriebsart Positionieren freigegeben wird, ohne dass vorher eine Referenzfahrt erfolgt ist, wird eine Fehlermeldung (Regler Fehler Nr. 900) ausgegeben und der Antrieb bleibt lagegeregelt auf der aktuellen Position stehen. Positionieraufträge werden nicht ausgeführt. Erst nachdem eine Referenzfahrt durchgeführt wurde (einmalig nach dem Einschalten), werden Positionieraufträge ausgeführt. Die Fehlermeldung kann nur quittiert werden, wenn eine Referenzfahrt durchgeführt wurde. Nach der Referenzierung kann eine Positionierung gestartet werden.



#### **HINWEIS!**

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



## 5.6 Varianten der Positionierung, abhängig vom Zielmodus (Parameter 118.10)

**HINWEIS!**

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

Folgende Positioniermodi existieren:

Zielmodus P118.10	Beschreibung
"Absolut/Relativ" CANopen (Standardwert 9)	<ul style="list-style-type: none"> <li>• Ziel steht in P118.16 (INT32)</li> <li>• Umschaltung "absolut/relativ" findet nur über das Steuerwort statt</li> </ul>
"Relativ, positive und negativ" (Wert 4)	<ul style="list-style-type: none"> <li>• Ziel steht in P118.16 (INT32)</li> <li>• keine Umschaltung "absolut/relativ" über das Steuerwort</li> </ul>
"Absolut relativ" (Wert 10)	<ul style="list-style-type: none"> <li>• Ziel steht in P118.9 (UINT32)</li> <li>• Umschaltung "absolut/relativ" findet nur über das Steuerwort statt</li> </ul>
"Absolut mit kürzestem Weg / Relativ" CANopen (Wert 12) (ab V1.10)	<ul style="list-style-type: none"> <li>• Ziel steht in P118.16 (INT32)</li> <li>• Umschaltung "absolut/relativ" findet nur über das Steuerwort statt</li> <li>• Bei absolutem Ziel erfolgt die Positionierung in Richtung des kürzeren Weges</li> </ul>
"Absolut ohne Bereichsverschiebung / Relativ" CANopen (Wert 16) (ab V1.10)	<ul style="list-style-type: none"> <li>• Ziel steht in P118.16 (INT32)</li> <li>• Umschaltung "absolut/relativ" findet nur über das Steuerwort statt</li> <li>• keine interne Bereichsverschiebung um <math>2^{31}</math> Inkremente bei absoluter Zielangabe</li> </ul>
"Absolut mit kürzestem Weg ohne Bereichsverschiebung / Relativ" CANopen (Wert 17) (ab V1.10)	<ul style="list-style-type: none"> <li>• Ziel steht in P118.16 (INT32)</li> <li>• Umschaltung "absolut/relativ" findet nur über das Steuerwort statt</li> <li>• keine interne Bereichsverschiebung um <math>2^{31}</math> Inkremente bei absoluter Zielangabe</li> <li>• Bei absolutem Ziel erfolgt die Positionierung in Richtung des kürzeren Weges</li> </ul>
Alle anderen Modis	<ul style="list-style-type: none"> <li>• Ziel steht in P118.9 (UINT32)</li> <li>• keine Umschaltung "absolut/relativ" über das Steuerwort</li> <li>• keine Umrechnung (Datentyp = UINT32)</li> </ul>

## 5.6 Varianten der Positionierung, abhängig vom Zielmodus (Parameter 118.10)

Umschaltung "absolut/relativ", über das Steuerwort Bit 6

**Bit 6**

**0** Absolut

**1** Relativ

Die Umrechnung vom Datentyp INT32  $\Leftrightarrow$  UINT32 soll heißen, ein Offset von  $2^{31}$  wird je nach Richtung dazu addiert oder abgezogen. Dies ist notwendig, um eine einheitliche Darstellung der Feldbusobjekte im Datentyp INT zu bekommen, weil einige Regler Parameter für die Positionierung (siehe [▶CANopen Offset◀](#) auf Seite 32) als Datentyp UINT realisiert sind. Für den Anwender sind also in der Positionierung die vorhandenen Feldbusobjekte als Datentyp INT zu sehen.



### **HINWEIS!**

Die Umrechnung im Positioniermodus auf P118.16 "Absolut/Relativ CANopen" wird nicht deaktiviert.

# GRUNDLAGEN CAN / CANOPEN

## 6.1 Literatur zum Thema CAN

---

Für grundlegende Informationen zum Thema CAN empfehlen wir folgende Literatur:

- CAN Controller-Area-Network  
Konrad Etschberger  
Carl Hauser Verlag München Wien
- CAN Controller Area Network. Grundlagen und Praxis  
Wolfhard Lawrenz  
Hüthig Verlag
- CANopen  
Holger Zeltwanger  
VDE-Verlag
- CANopen Device Profile for Drives and Motion Control  
CiA Draft Standard CiA<sup>®</sup> 402  
CAN in Automation (CiA)
- [www.can-cia.de](http://www.can-cia.de)  
CAN in Automation (CiA)  
Kontumazgarten 3  
D-90429 Nürnberg

### 6.2 Grundlagen CAN

	<p>Der CAN-Feldbus wird in Linienstruktur ausgeführt. Als physikalische Basis der Datenübertragung dient eine Dreidrahtleitung mit den Anschlüssen CAN_High, CAN_Low und CAN_Ground. CAN verwendet eine erdsymmetrische Übertragung, um Gleichtaktstörungen zu unterdrücken. Daher werden Differenzsignale ausgewertet.</p>
Netzwerk	<p>CAN ist ein Multi-Master-Netzwerk. Jeder Teilnehmer kann gleichberechtigt und aktiv auf den Bus zugreifen. CAN verwendet die objektorientierte Adressierung, d. h. die übermittelte Nachricht wird mit einem netzwerkweit festgelegten Identifier gekennzeichnet. Der Identifier stellt den codierten Namen der Nachricht dar.</p>
Buszugriff	<p>Der Buszugriff erfolgt über das CSMA/CA-Verfahren (Carrier Sense Multiple Access / Collision Avoidance). Da jeder Teilnehmer das Recht hat, nach Erkennung der notwendigen Busruhe, mit dem Senden seiner Nachricht zu beginnen, können Kollisionen entstehen. Dieses wird durch die bitweise Arbitrierung der zu sendenden Nachrichten vermieden. Dabei werden zwei Buspegel unterschieden, ein dominanter Pegel, logischer Bitwert 0, und ein rezessiver Pegel, logischer Bitwert 1. Im schlimmsten Fall beginnen alle sendewilligen Teilnehmer gleichzeitig mit dem Versenden ihrer Nachricht auf dem Bus. Wird ein rezessives Bit eines Teilnehmers von einem dominanten Bit eines anderen überschrieben, so zieht sich der „rezessive“ Knoten vom Bus zurück und versucht nach Erkennung der Busruhe erneut seine Nachricht abzusetzen. Somit ist gewährleistet, dass die wichtigste, höchstpriorie Nachricht (mit dem niedrigsten Identifier) kollisionsfrei und ohne Verzögerung übertragen wird. Aus diesem Grund ist es natürlich notwendig, dass jeder Identifier nur einmal am CAN-Bus vergeben sein darf.</p>
Identifier	<p>Es stehen in der CAN Spezifikation CAN 2.0A 2032 (CiA) unterschiedliche Identifier zur Verfügung. Jeder Teilnehmer kann unaufgefordert senden (Multi-Master-Fähigkeit). Ein Sender übermittelt seine Nachricht an alle CAN-Knoten (Broadcast), die anhand des Identifiers selbst entscheiden, ob sie die Nachricht weiterverarbeiten oder nicht.</p>
Fehler	<p>In einem CAN-Datentelegramm können bis zu acht Byte Nutzdaten übertragen werden. Zur Fehler- oder Überlastsignalisierung kann ein CAN-Knoten Error- oder Overload-Telegramme senden. Dieses geschieht auf Schicht 2 des OSI/ISO-Referenzmodells, dem Data Link Layer, also unabhängig von der Applikation. Aufgrund einer hochwertigen Fehlererkennung und -behandlung auf Schicht 2 wird eine Hamming-Distanz (Maß der Fehlererkennung) von HD = 6 erreicht, d. h. maximal fünf gleichzeitig auftretende Bitfehler innerhalb eines Telegramms werden sicher als Fehler erkannt.</p>
	<p>CANopen ist ein offenes und damit herstellernerutrales Feldbussystem, welches auf den Layern 1 und 2 - Definitionen des CAN-Standards aufsetzt.</p>
CAL-Spezifikation	<p>Das CANopen-Protokoll basiert auf der CAL-Spezifikation (Schicht 7-Protokoll). Bei CANopen werden Profile unterschieden. Das Kommunikationsprofil (CiA 301) definiert die Art und Weise des Datenaustausches und allgemeine, für alle Geräte geltende Festlegungen.</p>
Geräteprofil	<p>In den Geräteprofilen werden die anwendungs- und gerätespezifische Festlegungen, die inhaltliche Bedeutung der Daten und die Gerätefunktionalität beschrieben. Geräteprofile existieren u. a. für Antriebe, I/O-Module, Geber oder programmierbare Geräte. Der CANopen-Slave für den b maXX 2500 / 3300 / 5000 Regler ist nach dem Geräteprofil CiA 402 (Drives and Motion Control) implementiert.</p> <p>CANopen unterscheidet vier Arten von Nachrichten:</p>

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

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

**SDO** SDOs dienen der Übertragung größerer Datenmengen mit niedriger Priorität (Bedarfsdaten). Ein Datenblock mit mehr als vier Byte Nutzdaten wird dazu durch das CANopen-Protokoll segmentiert und auf mehrere SDOs verteilt (SDO Segmented Transfer). Datenblöcke 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 Einträge im Objektverzeichnis zugegriffen werden.

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

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

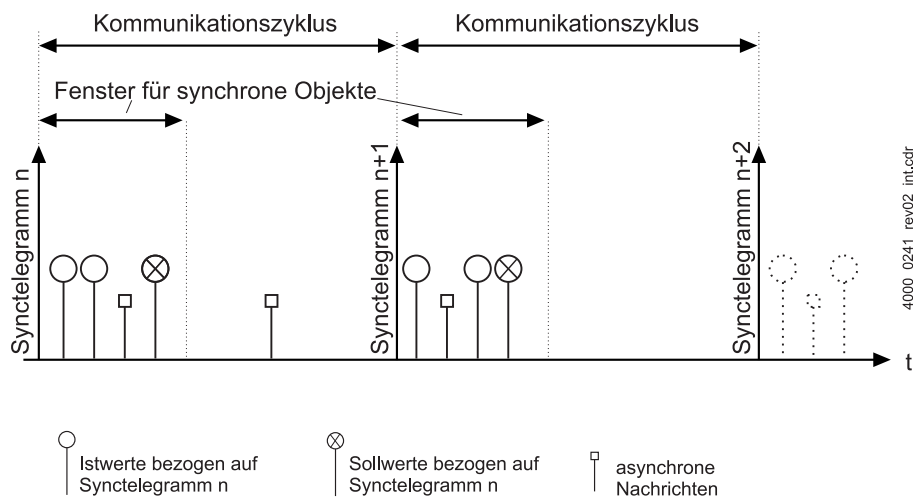


Abbildung 2: PDO Übertragungsarten

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

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

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

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

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

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

- INITIALISIERUNG
- PRE-OPERATIONAL
- STOPPED
- OPERATIONAL

Die einzelnen Zustandsübergänge werden durch NMT-Kommandos ausgelöst. Der CANopen-Slave wechselt nach dem Initialisieren automatisch in den Zustand PRE-OPERATIONAL. Weitere Informationen finden Sie in [▶ Netzwerkmanagement \(NMT\) ◀](#) ab Seite 46.

# CANOPEN AM B MAXX 2500 / 3300 / 5000

## 7.1 Allgemeines

---

Der b maXX 2500 / 3300 / 5000 CANOpen Slave verbindet den b maXX 2500 / 3300 / 5000 über den CAN-Bus mit anderen CAN-Knoten (z. B. PC, SPS, weitere b maXX Geräte, I/O-Module).

Informationen zur Installation und zum Umgang mit der Gerätereihe b maXX 3300 / 5000 finden Sie in der Dokumentation 5.11018 / 5.09021.

Informationen zur Programmierung des b maXX 3300 / 5000 Reglers finden Sie im Parameterhandbuch 5.12001 / 5.09022.

## 7.2 Adresseinstellung

---

Die Einstellung der Knoten-Adresse des b maXX 3300 / 5000 CANOpen Slave ist in der Betriebsanleitung b maXX 3300 / 5500 (Dokumentation 5.11018 / 5.09021) beschrieben.

## 7.3 EDS-Datei

---

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

Der Namensweiterung der EDS-Datei ist \*.eds.

Die Datei ist im Downloadbereich der Baumüller Homepage erhältlich.

### 7.4 Diagnose

---

CANopen ist ein auf CAN basierendes Kommunikationsprotokoll und folgt dem Standard EN 50325-4 (CiA 301). Somit können für die Systemdiagnose Standardtools und Geräte zur Analyse von CAN-Netzwerken verwendet werden.

### 7.5 Datenaustausch und Parametrierung

---

Der Zugriff auf Daten oder Parameter erfolgt immer über CANopen Objekte. Entsprechend der Profilstruktur werden Objekte zur Kommunikationssteuerung (Indizes 0x1XXX) und anwendungs- oder gerätespezifische Objekte unterschieden. Die letzteren gliedern sich in Objekte nach Profil CiA 402 (Indizes 0x6XXX) und herstellerspezifische Objekte (Indizes 0x2XXX bzw. 0x4XXX).

Eine Auflistung der 6XXXer und 0x2XXXer bzw. 0x4XXXer Objekte finden Sie in [>Kurzreferenz<](#) ab Seite 137.



#### **HINWEIS!**

#### **Wichtig:**

Die Adressierung der herstellerspezifischen Parameter ist in Anhang [>B.1 2000er / 4000er Objektnummern \(herstellerspezifische Objekte\)<](#) ab Seite 137 beschrieben.



## 7.6 Verzeichnis der Objekte zur Kommunikationssteuerung

In diesem Abschnitt finden Sie alle vom Baumüller CANopen Slave unterstützten Objekte des kommunikationsspezifischen Bereiches des Objektverzeichnis nach CiA<sup>®</sup> 301.

Name	Index	Subindex	Datentyp	Defaultwert
Device Type	0x1000	0x00	UINT32	0x00020192

Dieses Objekt kann nur gelesen werden und enthält die Information, um welches Gerät es sich handelt (Drive nach CiA<sup>®</sup> 402), nur lesbar.

Name	Index	Subindex	Datentyp	Defaultwert
Error Register	0x1001	0x00	UINT8	0x0

Dieses Objekt kann nur gelesen werden. Das Objekt 1001<sub>hex</sub> enthält eine Fehlerbitleiste, die folgende Bedeutung hat:

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

COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
80 <sub>hex</sub> + Adresse	08 <sub>hex</sub>	Emergency Error Code		Error Register	Herstellerspezifisches Fehlerfeld				

EMCY-Telegramm für Error Reset / No Error

Name	Index	Subindex	Datentyp	Defaultwert
Manufacturer Status Register	0x1002	0x00	UINT32	-

Dieses Objekt kann nur gelesen werden. Das Low Byte enthält das Low Byte des Regler-Statuswortes aus Parameter **108.3**.

Name	Index	Subindex	Datentyp	Defaultwert
COB-ID SYNC-Message	0x1005	0x00	UINT32	0x80

Dieses Objekt enthält Informationen über das Sync-Verhalten des Slaves. Der Slave ist kein SYNC-Master, d. h. es können nur SYNC-Telegramme empfangen werden. Die unteren 11 Bit im Low-Word geben den Identifier des SYNC-Telegramms an (0x80), nur lesbar.

Name	Index	Subindex	Datentyp	Defaultwert
Communication Cycle Period	0x1006	0x00	UINT32	0

Falls das Sync-Telegramm aktiviert ist, muss das Sync-Interval auf die Zeit des Sync-Telegramms (1000  $\mu$ s, 2000  $\mu$ s, 4000  $\mu$ s oder 8000  $\mu$ s) eingestellt werden. Die eingestellte Zeit wirkt sich auf die Parameter **131.18** (Feldbus Zykluszeit) und **156.4** (Sync-Offset) des b maXX<sup>®</sup> Reglers aus.

Name	Index	Subindex	Datentyp	Defaultwert
Synchronous Window Length	0x1007	0x00	UINT32	0

Dieses Objekt wird nicht ausgewertet.

Name	Index	Subindex	Datentyp	Defaultwert
Manufacturer Device Name	0x1008	0x00	VString	-

Dieses Objekt ist nur lesbar. Es enthält die Gerätebezeichnung.

Name	Index	Subindex	Datentyp	Defaultwert
Manufacturer Hardware Version	0x1009	0x00	VString	-

Dieses Objekt ist nur lesbar. Es enthält die aktuelle Hardware-Version des Reglers aus Parameter **102.25**.

Name	Index	Subindex	Datentyp	Defaultwert
Manufacturer Software Version	0x100A	0x00	VString	-

Dieses Objekt ist nur lesbar. Es enthält die aktuelle Software-Version des Reglers, z. B. die Zeichenfolge: „01.08.00 S (Build 109)“.

Name	Index	Subindex	Datentyp	Defaultwert
Guard Time	0x100C	0x00	UINT16	0

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

Name	Index	Subindex	Datentyp	Defaultwert
Life Time Factor	0x100D	0x00	UINT8	0

Der Wert dieses Objektes wird mit Objekt 0x100C multipliziert und ergibt damit die Zeit für das Node Guarding. Durch Beschreiben mit den Wert „0“ wird das Node Guarding deaktiviert.

Name	Index	Subindex	Datentyp	Defaultwert
Producer Heartbeat Time	0x1017	0x00	UINT8	0x03

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

Name	Index	Subindex	Datentyp	Defaultwert
Identity Object	0x1018	0x00	UINT8	0x04
Vendor ID		0x01	UINT32	0x0000015H
Product Code		0x02	UINT32	0x26483052
Revision Number		0x03	UINT32	0x00000000
Serial Number		0x04	UINT32	0x00000000

In diesem Objekt sind einige Informationen über das Gerät enthalten.

Name	Index	Subindex	Datentyp	Defaultwert
RPDO0...3 Communication Parameter Axis 1	0x1400 ... 0x1403	0x00	UINT8	0x05
COB-ID used by RPDO		0x01	UINT32	(0x200 ... 0x500) + Adresse
Transmission Type		0x02	UINT8	0
Inhibit Time		0x03	UINT16	0
Compatibility Entry		0x04	UINT8	0
Event Timer		0x05	UINT16	0

Diese Objekte enthalten Informationen zur Receive-PDO 0 bis 3 für Achse 1. In Subindex 0x01 wird der Identifier der Receive-PDO 0 bis 3 für Achse 1 eingetragen. Subindex 0x02 enthält den Triggertyp dieser PDO. In Subindex 0x03 wird die Inhibit Time eingestellt, die die minimale Verzögerungszeit für ein Transmissionsintervall darstellt. Der Eingabewert ist definiert als ein Multiplikator von 100 µs. Subindex 0x04 wird nicht verwendet. Subin-

dex 0x05 dient der Zeiteinstellung für timergetriggerte Sende-PDOs. Die Auflösung ist 1 Millisekunde.

Name	Index	Subindex	Datentyp	Defaultwert
RPDO0...3 Communcation Parameter Axis 2	0x1440 ... 0x1443	0x00	UINT8	0x05
COB-ID used by RPDO		0x01	UINT32	0
Transmission Type		0x02	UINT8	0
Inhibit Time		0x03	UINT16	0
Compatibility Entry		0x04	UINT8	0
Event Timer		0x05	UINT16	0

Diese Objekte enthalten Informationen zur Receive-PDO 0 bis 3 für Achse 2. In Subindex 0x01 wird der Identifier der Receive-PDO 0 bis 3 für Achse 2 eingetragen. Subindex 0x02 enthält den Triggertyp dieser PDO. In Subindex 0x03 wird die Inhibit Time eingestellt, die die minimale Verzögerungszeit für ein Transmissionsintervall darstellt. Der Eingabewert ist definiert als ein Multiplikator von 100 µs. Subindex 0x04 wird nicht verwendet. Subindex 0x05 dient der Zeiteinstellung für timergetriggerte Sende-PDOs. Die Auflösung ist 1 Millisekunde.

Name	Index	Subindex	Datentyp	Defaultwert
RPDO0...3 Mapping Axis 1	0x1600 ... 0x1603	0x00	UINT8	0x01
		0x01	UINT32	0x60400010
		:	:	
		n	UINT32	

Dieses Objekt enthält den Inhalt der Receive-PDO 0 bis 3 für Achse 1. In Subindex 0x00 steht die Gesamtzahl der nachfolgenden Einträge. Die Gesamtzahl der gemappten Objekte darf den Sollwertrahmen von max. 16 Objekten nicht überschreiten (siehe auch [>PDO-Mapping<](#) ab Seite 61).

Name	Index	Subindex	Datentyp	Defaultwert
RPDO0...3 Mapping Axis 2	0x1640 ... 0x1643	0x00	UINT8	0x01
		0x01	UINT32	0x60400010
		:	:	
		n	UINT32	

Dieses Objekt enthält den Inhalt der Receive-PDO 0 bis 3 für Achse 2. In Subindex 0x00 steht die Gesamtzahl der nachfolgenden Einträge. Die Gesamtzahl der gemappten Objekte darf den Sollwertrahmen von max. 16 Objekten nicht überschreiten (siehe auch [>PDO-Mapping<](#) ab Seite 61).

Name	Index	Subindex	Datentyp	Defaultwert
TPDO0...3 Communication Parameter Axis 1	0x1800 ... 0x1803	0x00	UINT8	0x06
COB-ID used by TPDO		0x01	UINT32	(0x180 ... 0x480) + Adresse
Transmission Type		0x02	UINT8	0
Inhibit Time		0x03	UINT16	0
Compatibility Entry		0x04	UINT8	0
Event Timer		0x05	UINT16	0
SYNC Start Value		0x06	UINT8	0

Diese Objekte enthalten Informationen zur Transmit-PDO 0 bis 3 für Achse 1. In Subindex 0x01 wird der Identifier der Transmit-PDO 0 bis 3 für Achse 1 eingetragen. Subindex 0x02 enthält den Triggertyp dieser PDO. In Subindex 0x03 wird die Inhibit Time eingestellt, die die minimale Verzögerungszeit für ein Transmissionsintervall darstellt. Der Eingabewert ist definiert als ein Multiplikator von 100 µs. Subindex 0x04 wird nicht verwendet. Subindex 0x05 dient der Zeiteinstellung für timergetriggerte Sende-PDOs. Die Auflösung ist 1 Millisekunde. Subindex 0x06 enthält die SYNC Start Time. Diese ist mit Null definiert, da der Zähler der SYNC Message durch die PDO nicht geändert wird.

Name	Index	Subindex	Datentyp	Defaultwert
TPDO0...3 Communication Parameter Axis 2	0x1840 ... 0x1843	0x00	UINT8	0x06
COB-ID used by TPDO		0x01	UINT32	0
Transmission Type		0x02	UINT8	0
Inhibit Time		0x03	UINT16	0
Compatibility Entry		0x04	UINT8	0
Event Timer		0x05	UINT16	0
SYNC Start Value		0x06	UINT8	0

Diese Objekte enthalten Informationen zur Transmit-PDO 0 bis 3 für Achse 2. In Subindex 0x01 wird der Identifier der Transmit-PDO 0 bis 3 für Achse 2 eingetragen. Subindex 0x02 enthält den Triggertyp dieser PDO. In Subindex 0x03 wird die Inhibit Time eingestellt, die die minimale Verzögerungszeit für ein Transmissionsintervall darstellt. Der Eingabewert ist definiert als ein Multiplikator von 100 µs. Subindex 0x04 wird nicht verwendet. Subindex 0x05 dient der Zeiteinstellung für timergetriggerte Sende-PDOs. Die Auflösung ist 1 Millisekunde. Subindex 0x06 enthält die SYNC Start Time. Diese ist mit Null definiert, da der Zähler der SYNC Message durch die PDO nicht geändert wird.

Name	Index	Subindex	Datentyp	Defaultwert
TPDO0...3 Mapping Axis 1	0x1A00 ... 0x1A03	0x00	UINT8	0x01
		0x01	UINT32	0x60410010
		:	:	
		n	UINT32	

Dieses Objekt enthält den Inhalt der Transmit-PDO 0 bis 3 für Achse 1. In Subindex 0x00 steht die Gesamtzahl der nachfolgenden Einträge. Die Gesamtzahl der gemappten Objekte darf den Istwertrahmen von max. 16 Objekten nicht überschreiten. (siehe auch [>PDO-Mapping<](#) ab Seite 61).

Name	Index	Subindex	Datentyp	Defaultwert
TPDO0...3 Mapping Axis 2	0x1A40 ... 0x1A43	0x00	UINT8	0x01
		0x01	UINT32	0x60410010
		:	:	
		n	UINT32	

Dieses Objekt enthält den Inhalt der Transmit-PDO 0 bis 3 für Achse 2. In Subindex 0x00 steht die Gesamtzahl der nachfolgenden Einträge. Die Gesamtzahl der gemappten Objekte darf den Istwertrahmen von max. 16 Objekten nicht überschreiten. (siehe auch [>PDO-Mapping<](#) ab Seite 61).

## 7.7 Netzwerkmanagement (NMT)

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

### 7.7.1 Zustandsmaschine der Kommunikation

Hier ist das Zustandsdiagramm der Kommunikation des CANopen Slaves dargestellt.

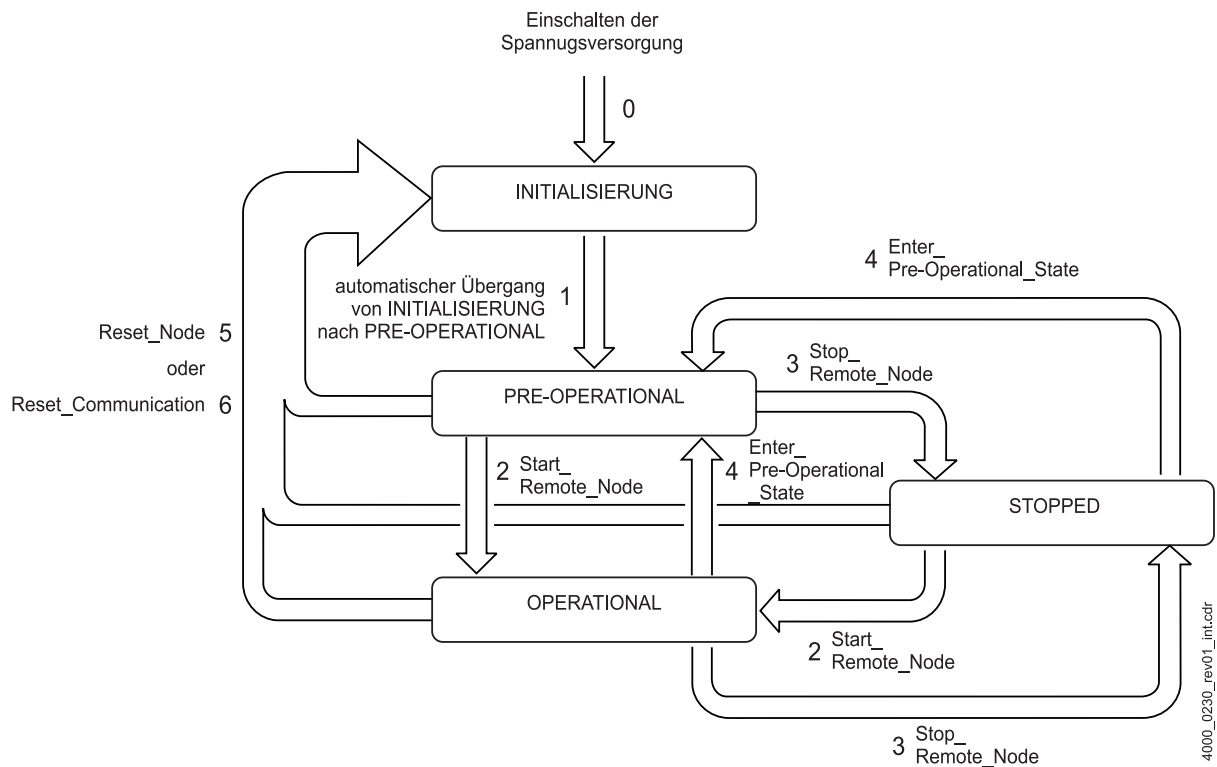


Abbildung 3: Zustandsmaschine der Kommunikation

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

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

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

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

**1** automatischer Übergang von INITIALISIERUNG nach PRE-OPERATIONAL

Hinweis:

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

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

**2** Start\_Remote\_Node

**3** Stop\_Remote\_Node

**4** Enter\_Pre-Operational\_State

**5** Reset\_Node

**6** Reset\_Communication

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

## 7.7.2 Telegramme

NMT-Telegramme zur Kommunikationssteuerung besitzen nach dem Predefined Connection Set den Default-Identifier „0“ (siehe auch [►Grundlagen CAN / CANopen◄](#) ab Seite 35).

### 7.7.2.1 Zustandssteuerung

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

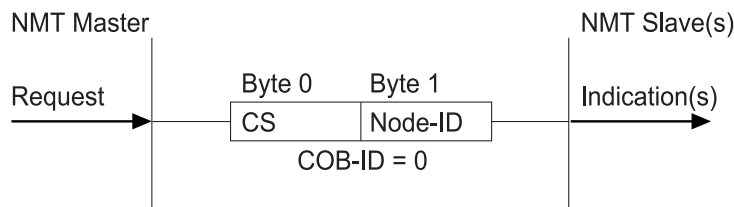


Abbildung 4: NMT-Telegramm zur Steuerung der Kommunikationszustände

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

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

COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
00 <sub>hex</sub>	02 <sub>hex</sub>	80 <sub>hex</sub>	10 <sub>hex</sub>						

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



**HINWEIS!**

Der b maXX<sup>®</sup> 5000 sieht z. Zt. keinen Geräte-Reset per Software vor.

Nach dem Einschalten der Versorgungsspannung und einem Reset meldet sich der CANopen-Slave mit dem Boot Up Telegramm (siehe auch [▶Boot Up◀](#) auf Seite 49). Die gesamte Resetsequenz dauert vom Empfang des Kommandos Reset\_Node bis zur Rückmeldung mittels Boot Up Telegramm einige wenige Sekunden.

**WARNUNG!****Gefahr durch mechanische und elektrische Einwirkung!**

Wird im laufenden zyklischen Betrieb ein Reset ausgelöst, kann dies zu unerwünschten Zuständen in der Applikation führen, da im Regler der Bootdatensatz geladen wird.

Deshalb:

- Prüfen Sie das Mapping nach jedem Reset.

### 7.7.2.2 Boot Up

Das Boot-Up Verhalten nach CiA 301 V4 wird mit der Gerätereihe b maXX 3300 / 5000 unterstützt.

Boot-Up nach CiA 301 V4,

Boot-Up Telegramm mit ID = 0x700 + Node-ID, DLC = 1 Byte 0 mit den Daten = 0 gefüllt.

COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
701 <sub>hex</sub>	01 <sub>hex</sub>	00 <sub>hex</sub>							

CiA 301 V4

### 7.7.3 Node Guarding

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

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

Die COB-ID ergibt sich aus  $0x700 + \text{Adresse}$ , laut Predefined Connection Set. Diese COB-ID kann auch geändert werden. Das dafür erforderliche Objekt ist  $0x100E$ .

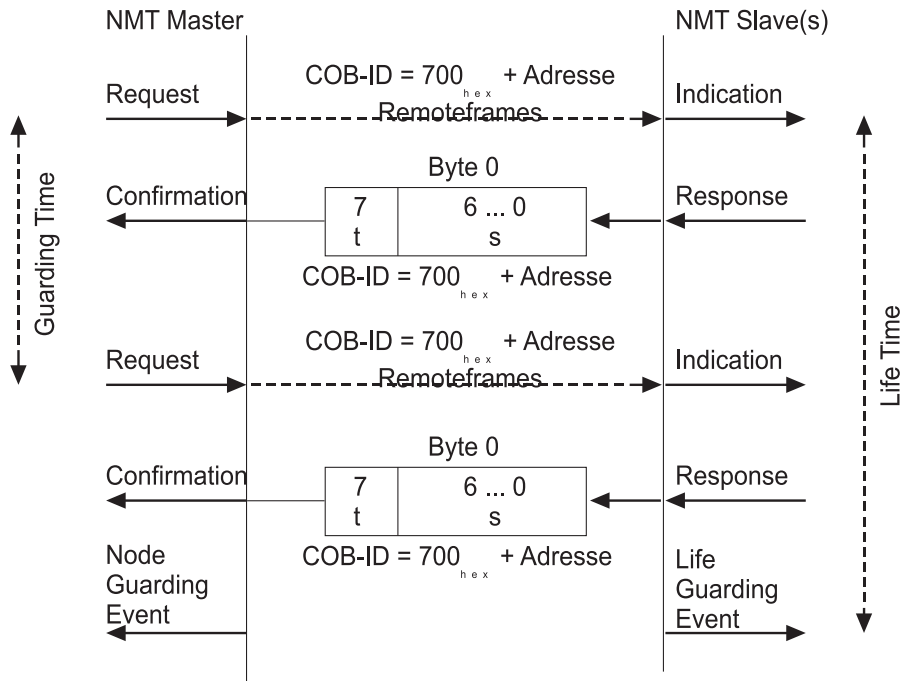


Abbildung 5: Node Guarding Protokoll

In den Objekten  $0x100C$  und  $0x100D$  wird die Guarding-Zeit eingestellt. Innerhalb dieser Zeit muss der Slave vom Master eine Guarding-Anforderung (Remotetelegramm) empfangen haben. Ist dies nicht der Fall, tritt das Life Guarding Event im Slave ein. Dadurch wechselt der Slave in den Zustand PRE-OPERATIONAL und im Regler wird die in Objekt  $0x6007$  vorgegebene Reaktion ausgelöst.

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

Aus der Antwort des Slaves auf eine Node-Guarding-Anforderung des Masters ist der aktuelle Kommunikationszustand des Slaves erkennbar. Das Antworttelegramm besteht aus einem Datenbyte (siehe auch [Abbildung 54](#) auf Seite 50). 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	0x7F (127)	0x7F (127)	0xFF (255)
OPERATIONAL	0x05 (5)	0x05	0x85 (133)
STOPPED	0x04 (4)	0x04	0x84 (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 0x100C und 0x100D parametrisierte Überwachungszeit.



#### HINWEIS!

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

## 7.8 Bedarfsdaten (SDO)

Service Daten Objekte (SDO) dienen dem Austausch von Nachrichten ohne Echtzeitanforderungen. Deshalb sind im Predefined Connection Set (siehe auch [►Grundlagen CAN / CANopen](#) ab Seite 35) 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 60).

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

### 7.8.1 Telegrammaufbau

Die COB-ID der Request SDO ergibt sich aus 0x600 + Adresse, bei Response-SDOs aus 0x580 + Adresse. Das Datenfeld des CAN-Datentelegramms (8 Byte) für eine SDO gliedert sich in drei Teile, einen Command Specifier CS (1 Byte), einem Multiplexor M (3 Byte) und dem eigentlichen Nutzdatenbereich D0 - D3 (4 Byte).

COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x600+ Adresse	0x08	CS	M	M	M	D0	D1	D2	D3
0x580 + Adresse	0x08	CS	M	M	M	D0	D1	D2	D3

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

## 7.8.2 Arten des SDO-Transfers

Die Baumüller CANopen-Anschaltung unterstützt den Expedited Transfer und den Segmented Transfer, wobei letzterer nur für die Objekte 0x1008, 0x1009 und 0x100A Manufacturer Device Name genutzt wird.

### Expedited Transfer

Es können Objekte geschrieben oder gelesen werden, deren Daten maximal 4 Byte umfassen. Es sind nur zwei Telegramme erforderlich, eine Anforderung und eine Antwort. Alle Objekte mit den Indizes 0x1XXX, 0x4XXX, 0x6XXX sind über Expedited SDOs ansprechbar mit Ausnahme der Objekte 0x1008, 0x1009 und 0x100A.

### 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 0x1008, 0x1009 und 0x100A nötig.

## 7.8.3 Objekt schreiben

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

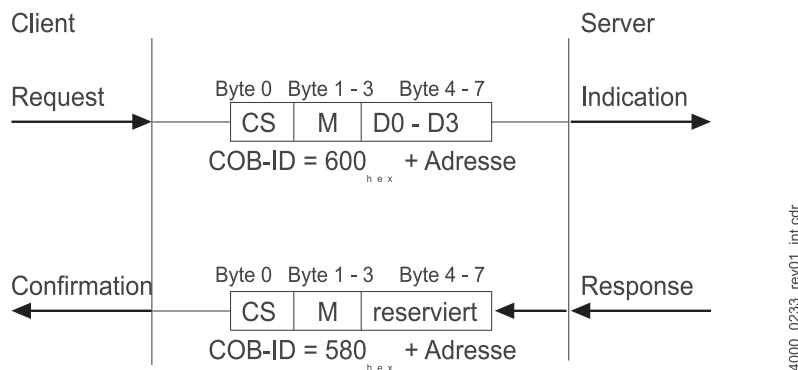


Abbildung 6: Initiate SDO Download Protocol

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

Datenlänge in D0 - D3	Command Specifier CS
1 Byte	0x2F
2 Byte	0x2B
4 Byte	0x23

Der Command Specifier CS für die Response beträgt 0x60, der Multiplexor ist identisch zu dem des Requests, das Datenfeld ist ohne Bedeutung (reserviert).

**Beispiel**

Auf das Objekt 0x6060, Subindex 0x00, des Slaves mit der Adresse 4 soll der Wert „-3“ (0xFD) geschrieben werden. Die Datenbreite dieses Objektes beträgt acht Bit.

Request

		CS	Multiplexor			D0	D1	D2	D3
COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x604	0x08	0x2F	0x60	0x60	0x00	0xFD	0x00	0x00	0x00

Basis-Adresse 0x600  
+ Slave-Adresse 0x4

Objekt 0x6060

Subindex 0x00

Wert -3

Response

		CS	Multiplexor			D0	D1	D2	D3
COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x584	0x08	0x60	0x60	0x60	0x00	0x00	0x00	0x00	0x00

Auf das Objekt 0x43E9, Subindex 0x00, des Slaves mit der Adresse 4 soll der Wert „12“ (0x0C) geschrieben werden. Die Datenbreite dieses Objektes beträgt 16 Bit.

Request

		CS	Multiplexor			D0	D1	D2	D3
COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x604	0x08	0x2B	0xE9	0x43	0x00	0x0C	0x00	0x00	0x00

Basis-Adresse 0x600  
+ Slave-Adresse 0x4

Objekt 0x43E9

Subindex 0x00

Wert 12

Response

		CS	Multiplexor			D0	D1	D2	D3
COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x584	0x08	0x60	0xE9	0x43	0x00	0x00	0x00	0x00	0x00

## 7.8 Bedarfsdaten (SDO)

Auf das Objekt 0x1800, Subindex 0x02, des Slaves mit der Adresse 4 soll der Wert „0x60610008“ geschrieben werden. Die Datenbreite dieses Objektes beträgt 32 Bit.

Request

		CS	Multiplexor			D0	D1	D2	D3
COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x604	0x08	0x23	0x00	0x18	0x02	0x08	0x00	0x61	0x60

Basis-Adresse 0x600  
+ Slave-Adresse 0x4

Objekt 0x1800

Subindex 0x02

Wert 0x60 61 00 08

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
0x584	0x08	0x60	0x00	0x18	0x02	0x00	0x00	0x00	0x00

### 7.8.4 Objekt lesen

Zum Lesen von Objekten wird bei der Baumüller CANopen-Anbindung der Expedited Transfer, bei den Objekten 0x1008, 0x1009 und 0x100A der Segmented Transfer benutzt.

#### 7.8.4.1 Expedited Transfer

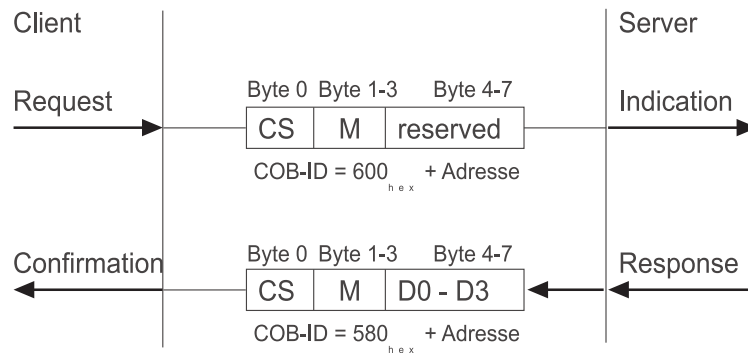


Abbildung 7: Initiate SDO Upload Expedited

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 0x40. 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	0x4F
2 Byte	0x4B
4 Byte	0x43

Der Multiplexor von Request und Response stimmt überein.

### Beispiel

Das Objekt 0x6061, Subindex 0x00, des Slaves mit der Adresse 4 soll gelesen werden. Die Datenbreite dieses Objektes beträgt 1 Byte.

### Request

		CS	Multiplexor				D0	D1	D2	D3
COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	
0x604	0x08	0x40	0x61	0x60	0x00	0x00	0x00	0x00	0x00	

### 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	
0x584	0x08	0x4F	0x61	0x60	0x00	0xD0	0x00	0x00	0x00	

Basis-Adresse 0x580  
+ Slave-Adresse 0x4

Objekt 0x6061

Subindex 0x00

Wert Daten 208

Das Objekt 0x6041, Subindex 0x00, des Slaves mit der Adresse 4 soll gelesen werden. Die Datenbreite dieses Objektes beträgt 2 Byte.

### Request

		CS	Multiplexor				D0	D1	D2	D3
COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	
0x604	0x08	0x40	0x41	0x60	0x00	0x00	0x00	0x00	0x00	

### 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	
0x584	0x08	0x4B	0x41	0x60	0x00	D0	D1	0x00	0x00	

Basis-Adresse 0x580  
+ Slave-Adresse 0x4

Objekt 0x6041

Subindex 0x00

Wert: DB high DB low

## 7.8 Bedarfsdaten (SDO)

Das Objekt 0x1400, Subindex 0x00, des Slaves mit der Adresse 4 soll gelesen werden. Die Datenbreite dieses Objektes beträgt 4 Byte.

Request

		CS	Multiplexor			D0	D1	D2	D3
COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x604	0x08	0x40	0x00	0x14	0x01	0x00	0x00	0x00	0x00

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
0x584	0x08	0x43	0x00	0x14	0x01	D0	D1	D2	D3

Basis-Adresse 0x580  
+ Slave-Adresse 0x4

Objekt 0x1400

Subindex 0x00

Wert: DB<sub>3</sub> DB<sub>2</sub> DB<sub>1</sub> DB<sub>0</sub>



## 7.8.4.2 Segmented Transfer

Zunächst wird mit dem Initiate SDO Upload Protocol eine Leseanforderung an den Slave geschickt. Der Slave antwortet mit dem Command Specifier CS 0x41. 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).

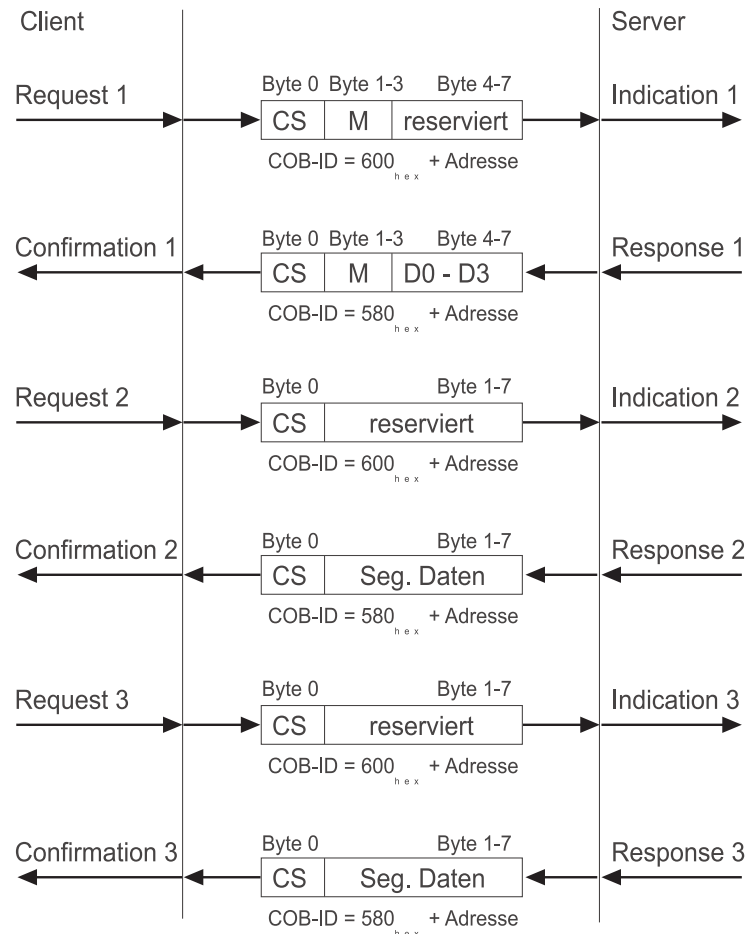


Abbildung 8: Upload SDO Segmented Protocol

## 7.8 Bedarfsdaten (SDO)

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

z. B. das Lesen des Objektes 0x1008 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	
0x604	0x08	0x40	0x08	0x10	0x00	0x00	0x00	0x00	0x00	

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	
0x584	0x08	0x41	0x08	0x10	0x00	0x06	0x00	0x00	0x00	

Byte 0 in Response 1 (Command Specifier 0x41) besagt, dass das Nutzdatenfeld die Anzahl der zu übertragenden Nutzdatenbytes enthält (6).

Request 2

		CS								
COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	
0x604	0x08	0x60	0x00	0x00	0x00	0x00	0x00	0x00	0x00	

Response 2

		CS	D0	D1	D2	D3	D4	D5	D6
COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x584	0x08	0x14	0x62	0x20	0x6D	0x61	0x58	0x58	0x00

Byte 0 in Request 2 (Command Specifier 0x60) bedeutet, dass das erste Segment (die sechs Byte) übertragen werden sollen. Byte 0 in Response 2 (Command Specifier, 0x14) besagt, dass das Nutzdatenfeld (sechs Byte) gültige Daten enthalten und dass dieses Segment zugleich das letzte ist.

Das Ergebnis der Übertragung lautet: „b maXX“.

### 7.8.5 Fehlerreaktionen

Fehlerhafte SDO-Zugriffe werden mit Abort Codes abgewiesen. Der Aufbau dieser Abort-Telegramme ist identisch zu dem in [▶Abbildung 7.8.1◀](#) auf Seite 51 dargestellten SDO-Telegramm. Das Datenfeld enthält einen Abort Code mit 4 Bytes.

Bei fehlerhaften Zugriffen auf kommunikationsspezifische Objekte (0x1XXX) werden folgende Meldungen unterschieden:

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

Fehlerhafte Zugriffe auf alle anderen Objekte (0x4XXX und 0x6XXX) werden global mit folgenden Codes abgewiesen:

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

## 7.9 Synchronisation (SYNC)

### Beispiel

Objekt 0x1008 Subindex 0x01 von Slave 4 soll gelesen werden. Das Objekt *0x1008 Manufacturer Device Name* besitzt aber nur Subindex 0x00.

### 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
0x604	0x08	0x40	0x08	0x10	0x01	0x00	0x00	0x00	0x00

### 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
0x584	0x08	0x80	0x08	0x10	0x01	0x11	0x00	0x09	0x06

Basis-Adresse 0x580  
+ Slave-Adresse 0x4

Objekt 0x1008

Subindex 0x01

Code 0x06 09 00 11

Der Command Specifier CS (Byte 0, 0x80) im Response Telegramm gibt an, dass es sich um ein Abort Telegramm handelt. Die Multiplexor von Request und Response stimmen überein.

## 7.9 Synchronisation (SYNC)

Zur Synchronisation der Slaves wird das SYNC-Telegramm verwendet. Dieses Telegramm ist unbestätigt (Broadcast). Es enthält keine Daten. Die COB-ID steht im Objekt *0x1005 COB-ID SYNC*. Defaultmäßig ist 0x80 festgelegt. Das Optionsmodul CANopen-Slave kann SYNC-Telegramme empfangen. Es ist kein SYNC-Master!

Der Empfang eines SYNC-Telegramms mit dem im Objekt 0x1005 dargestellten Identifier erzeugt auf dem Optionsmodul CANopen einen Interrupt, der an den b maXX<sup>®</sup> Regler weitergeleitet wird. Damit kann dieses Signal zur Synchronisation des b maXX<sup>®</sup> Reglers verwendet werden. Innerhalb eines SYNC-Intervalls (Kommunikationszyklus) 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. Die Einstellung der Zykluszeit für das SYNC-Telegramm wird in Objekt 0x1006 vorgenommen. Siehe hierzu [►Verzeichnis der Objekte zur Kommunikationssteuerung◄](#) ab Seite 41.

Die Kommunikationszykluszeit im Regler wird automatisch von Objekt 0x1006 übernommen.

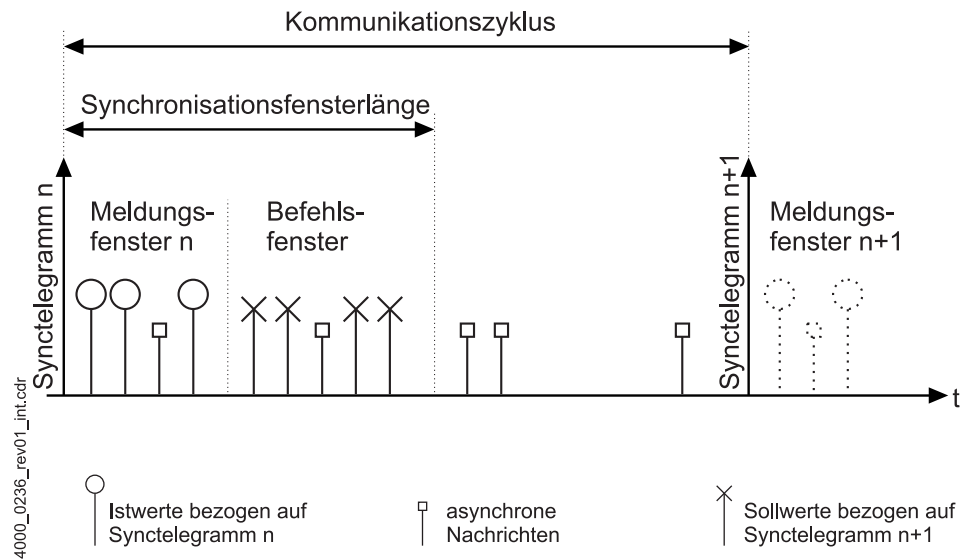


Abbildung 9: Kommunikationszyklus

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

## 7.10 Prozessdaten (PDO)

Prozess-Daten-Objekte sind unbestätigte Telegramme mit hochpriorigen COB-IDs. Sie sind auf den Austausch von Daten mit Echtzeitanforderungen optimiert. In den PDOs kann der gesamte CAN-Datenrahmen (acht Byte) für die Nutzdatenübertragung verwendet werden. Deshalb muss das Format des Datenaustauschs über PDOs vor Beginn der Kommunikation zwischen Sender und Empfänger definiert sein (Mapping). Das Senden und Empfangen von PDOs kann auf unterschiedliche Weise ausgelöst werden (siehe auch [►Kommunikationsbeziehung über PDO◄](#) ab Seite 65).

### 7.10.1 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. Für diese Parametrierung werden SDOs genutzt. Das Mapping wird über im Objektverzeichnis adressierbare Objekte eingestellt. Für jede PDO existieren pro Achse vier solcher Objekte (siehe auch [►Verzeichnis der Objekte zur Kommunikationssteuerung◄](#) ab Seite 41). Eines der Objekte bestimmt den Inhalt der PDO, das zweite die Kommunikationsbeziehung bzw. Triggung.

Prozess-Daten-Objekt	Objekt für Inhalt	Objekt für die Kommunikationsbeziehung
TX-PDO1 Axis1	0x1A00	0x1800
TX-PDO2 Axis1	0x1A01	0x1801
TX-PDO3 Axis1	0x1A02	0x1802
TX-PDO4 Axis1	0x1A03	0x1803
TX-PDO1 Axis2	0x1A40	0x1840
TX-PDO2 Axis2	0x1A41	0x1841
TX-PDO3 Axis2	0x1A42	0x1842
TX-PDO4 Axis2	0x1A43	0x1843
RX-PDO1 Axis1	0x1600	0x1400
RX-PDO2 Axis1	0x1601	0x1401
RX-PDO3 Axis1	0x1602	0x1402
RX-PDO4 Axis1	0x1603	0x1403
RX-PDO1 Axis2	0x1640	0x1440
RX-PDO2 Axis2	0x1641	0x1441
RX-PDO3 Axis2	0x1642	0x1442
RX-PDO4 Axis2	0x1643	0x1443

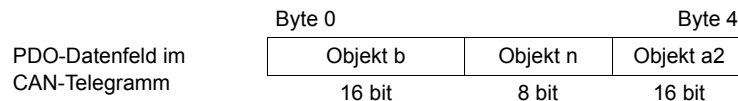
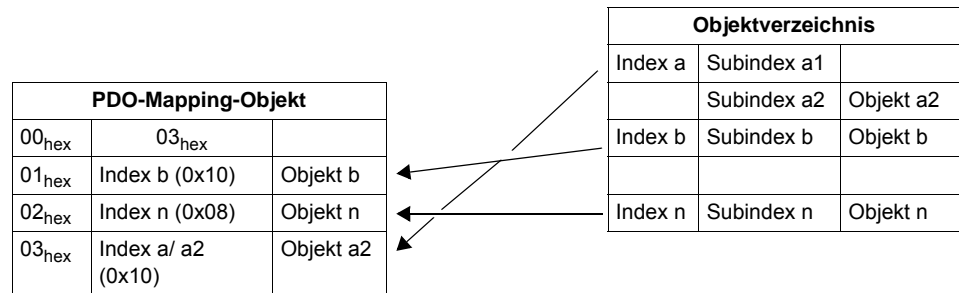


### HINWEIS!

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

Zur Nutzdatenübertragung stellt ein CAN-Datentelegramm maximal acht Byte zur Verfügung. Durch das Mapping wird der logische Inhalt dieser maximal acht Byte festgelegt. Für diese Festlegung werden bestimmte Angaben über das zu mappende Objekt benötigt: Objektindex, Subindex und Länge des Datums. Aus dem Objektverzeichnis werden die entsprechenden Objekte in das Mapping-Objekt eingetragen. Die Reihenfolge dieses Eintrages, festgelegt durch den Subindex des Mapping-Objektes, bestimmt die Reihenfolge der Daten im CAN-Telegramm. In den Mapping-Objekten (0x1600 ... 0x1603, 0x1640 ... 0x1643, 0x1A00 ... 0x1A03, 0x1A40 ... 0x1A43) werden die zu mappenden Objekte an die entsprechenden Subindizes (mit 0x01 beginnend) geschrieben, z. B. wird auf Objekt 0x1600 Subindex 0x01 der Wert 0x60400010 eingetragen. Dies bedeutet, die ersten beiden Bytes der in RX-PDO1 empfangenen Daten werden auf das Steuerwort (Objekt 0x6040 Subindex 0x00) geschrieben. Das Objekt 0x6040 ist auf den Parameter 108.1 Steuerwort umgesetzt (siehe auch [Anhang C - Umsetzungstabellen](#) ab Seite 143). Damit wird das erste Wort des in RX-PDO1 emp-

fangenen Telegramms auf das Steuerwort des b maXX Reglers geschrieben. In Subindex 0x00 muss die Anzahl der zu mappenden Objekte (Anzahl der mit gültigen Objekten belegten Subindizes) eingetragen werden. Ein ausführliches Beispiel für das Mapping ist in [►Beispiel für das PDO-Mapping◄](#) ab Seite 68 erläutert.



Das Default-Mapping ist in [►Verzeichnis der Objekte zur Kommunikationssteuerung◄](#) ab Seite 41 beschrieben.

Um ein bestehendes Mapping zu deaktivieren, können die Werte in den Subindizes überschrieben werden, oder es ist der Wert „0“ auf den Subindex 0x00 des entsprechenden Mapping-Objektes (0x1600 ... 0x1603, 0x1640 ... 0x1643, 0x1A00 ... 0x1A03, 0x1A40 ... 0x1A43) zu schreiben. Damit wird das gesamte Mapping der jeweiligen PDO deaktiviert, der Eintrag aber bleibt erhalten.

Zudem kann das Deaktivieren der Mapping Objekte über die zugehörige Kommunikationsobjekte 0x1400 bis 0x1403, 0x1440 bis 0x1443, 0x1800 bis 0x1803, 0x1840 bis 0x1843 im Subindex 1 mit Bit 31 auf 1 setzen erfolgen.

Beachte: Dabei muss die COBID mitgeschrieben werden.



#### HINWEIS!

Bei der Einstellung des Mappings in den (0x1600 ... 0x1603, 0x1640 ... 0x1643, 0x1A00 ... 0x1A03, 0x1A40 ... 0x1A43) ist jeweils der Subindex 0x00 mit der richtige Anzahl der gemappten Objekte zuletzt zu beschreiben.

#### Sollwerte:

Die zulässigen zyklische Sollwerte sind in einer Tabelle mit der Spalte „PDO-Mapping“ als „RX“ gekennzeichnet. Die Tabelle ist im Anhang B.2 (für die 6000'er Objektnummern) zu finden. Bei den herstellereigenen Parameter (2000er / 4000er Objekte) muss im Parameterhandbuch b maXX<sup>®</sup> 3300 (5.12001) bzw. im Parameterhandbuch b maXX<sup>®</sup> 5000 (5.09022) nachgeschaut werden.

#### Istwerte:

Die zulässigen zyklische Istwerte sind in einer Tabelle mit der Spalte „PDO-Mapping“ als „TX“ gekennzeichnet. Die Tabelle ist im Anhang B.2 (für die 6000'er Objektnummern) zu finden. Bei den herstellereigenen Parametern (2000er / 4000er Objekte) muss im

Parameterhandbuch b maXX<sup>®</sup> 3300 (5.12001) bzw. im Parameterhandbuch b maXX<sup>®</sup> 5000 (5.09022) nachgeschlagen werden. Dort finden Sie eine detaillierte Beschreibung der b maXX<sup>®</sup>-Parameter.

Fehlerhafte Mapping-Konfigurationen (unzulässige Objekte in 0x1600 ... 0x1603, 0x1640 ... 0x1643, 0x1A00 ... 0x1A03, 0x1A40 ... 0x1A43) werden durch Abort Codes über SDO gemeldet.

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

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



### HINWEIS!

Der Feldbus Slave hat kein Default Mapping

Wenn ein falscher Parameter (z. B. ein Istwert Parameter) bei den Sollwerten gemappt wird, wechselt die Achse nicht in den Zustand „Operational“.

### Dummy-Mapping

CANopen-Slave stellt 2 Dummy-Objekte bereit: ein 1 Byte Dummy-Objekt und ein 2 Byte Dummy-Objekt, welches ebenfalls in eine PDO gemappt werden kann. Diese Objekte haben die Indices 0x0005 (1 Byte Dummy) und 0x0006 (2 Byte Dummy). Das Dummy-Objekt dient als Platzhalter, um nur bestimmte Objekte innerhalb eines CAN-Telegramms zu verwenden (siehe auch [►Beispiel für das PDO-Mapping◄](#) ab Seite 68).



### HINWEIS!

Das aktuell eingestellte Mapping geht nach dem Ausschalten oder einem Reset Befehl der Zustandsmaschine verloren.

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

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

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



**HINWEIS!**

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

### 7.10.2 Kommunikationsbeziehung über PDO

In jedem Mapping-Objekt existiert ein Objekt für die Einstellung der Kommunikation. Der Objektindex hat einen Offset von -0x200 zum entsprechenden Mapping-Objekt.

Prozess Daten Objekt Mapping	Objekt für Inhalt	Objekt für die Kommunikationsbeziehung
TX-PDO1 Axis 1	0x1A00	0x1800
TX-PDO2 Axis 1	0x1A01	0x1801
TX-PDO3 Axis 1	0x1A02	0x1802
TX-PDO4 Axis 1	0x1A03	0x1803
TX-PDO1 Axis 2	0x1A40	0x1840
TX-PDO2 Axis 2	0x1A41	0x1841
TX-PDO3 Axis 2	0x1A42	0x1842
TX-PDO4 Axis 2	0x1A43	0x1843
RX-PDO1 Axis 1	0x1600	0x1400
RX-PDO2 Axis 1	0x1601	0x1401
RX-PDO3 Axis 1	0x1602	0x1402
RX-PDO4 Axis 1	0x1603	0x1403
RX-PDO1 Axis 2	0x1640	0x1440
RX-PDO2 Axis 2	0x1641	0x1441
RX-PDO3 Axis 2	0x1642	0x1442
RX-PDO4 Axis 2	0x1643	0x1443

Der Aufbau dieser Objekte ist in [▶Verzeichnis der Objekte zur Kommunikationssteuerung◀](#) ab Seite 41 beschrieben.

Das Kriterium für die Übernahme einer auf dem CAN-Bus übertragenen Nachricht in den Feldbus Slave ist die passende COB-ID. Die COB-ID wird in den Steuerobjekten 0x1400 - 0x1403, 0x1440 - 0x1443, 0x1800 - 0x1803, 0x1800 - 0x1803 unter Subindex 0x01 eingestellt. Stimmt der hier parametrisierte Identifier mit dem über den CAN-Bus gesendeten Nachrichten-Identifier überein, wird das Telegramm in den eigenen Telegrammpuffer übernommen.

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

z.B. 0x1400 Subindex 1

		CS	Multiplexor			D0	D1	D2	D3
COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x601	0x08	0x23	0x00	0x14	0x01	0x01	0x00	0x00	0x01

Objekt 0x1400
Wichtig
Bit 31

Weiterhin werden bei CANopen für das Senden und Empfangen Triggerbedingungen definiert, die PDOs in synchron und asynchron gliedern. Die Triggerbedingungen werden in den Objekten 0x1400 - 0x1403, 0x1440 - 0x1443, 0x1800 - 0x1803, 0x1840 - 0x1843 jeweils im Subindex 0x02 eingestellt.



### HINWEIS!

Nicht benötigte PDOs sollten deaktiviert werden, um eine Beeinflussung auszuschließen. Das Deaktivieren erfolgt durch das Beschreiben der Subindizes 0 der Objekte 0x1600 bis 0x1603, 0x1640 bis 0x1643, 0x1A00 bis 0x1A03 und 0x1A40 bis 0x1A43 mit 0 oder mit Bit 31 in SIX 2 der Objekte 0x1401 bis 0x1403, 0x1441 bis 0x1443, 0x1801 bis 0x1803 und 0x1841 bis 0x1843.

**Synchrone PDO** Das Senden und Empfangen ist an das SYNC-Telegramm gebunden (siehe auch [>Synchronisation \(SYNC\)<](#) ab Seite 60).

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

Wert in Subindex 0x02	Typ	Wirkung	
		TX-PDO 0x1800, 0x1801, 0x1802, 0x1803	RX-PDO 0x1400, 0x1401, 0x1402, 0x1403
0x00 (0)	asynchron	Das Senden erfolgt nach jedem empfangenen SYNC-Telegramm <b>und</b> es ist ein Ereignis eingetreten.	Vor dem letzten SYNC-Telegramm empfangene PDO mit passender COB-ID wird übernommen
0x01 (1)	synchron	Das Senden erfolgt nach jedem empfangenen SYNC-Telegramm	Vor dem letzten SYNC-Telegramm empfangene PDO mit passender COB-ID wird übernommen
0x02 - 0xF0 (2 - 240)	synchron	Das Senden erfolgt nach Empfang der eingestellten Anzahl von SYNC-Telegrammen	Vor dem letzten SYNC-Telegramm empfangene PDO mit passender COB-ID wird übernommen

Wert in Subindex 0x02	Typ	Wirkung	
		TX-PDO 0x1800, 0x1801, 0x1802, 0x1803	RX-PDO 0x1400, 0x1401, 0x1402, 0x1403
0xFC (252)	RTR synchron	Senden erfolgt nach Empfang des RTR -Telegramms mit passender COB-ID. Aktualisierung der PDO erfolgt nach Empfang des SYNC-Telegramms.	keine
0xFD (253)	RTR asynchron	Senden erfolgt nach Empfang des RTR -Telegramms mit passender COB-ID.	keine
0xFE (254)	asynchron	Das Senden erfolgt zeitgesteuert	keine
0xFF (255)	asynchron	Das Senden erfolgt ereignisgesteuert	Jede PDO mit passender COB-ID wird übernommen

Zeitgesteuertes Senden bedeutet, dass die Sendebedingung an einen Timer gebunden ist. Dieser Timer wird für die TX-PDO1 mittels Subindex 0x05 im Objekt 0x1800 (16 Bit) eingestellt. Analog lässt sich der Timer für TX-PDO2, TX-PDO3 und TX-PDO4 im Subindex 0x05 von Objekt 0x1801, 0x1802, 0x1803 setzen. Die Auflösung beträgt jeweils 1 Millisekunde. Der Timer oder die Timer werden beim Zustandswechsel nach OPERATIONAL gestartet. Das Senden der entsprechenden TX-PDO erfolgt dann zyklisch mit der in den Timern eingestellten Zeit. Die Timer werden gelöscht, indem auf die Subindizes 0x05 der Objekte 0x1800 - 0x1803 der Wert „0“ geschrieben wird.

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

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

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

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

### 7.10.3 Beispiel für das PDO-Mapping

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

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

Weiterhin erhält Knoten 2 in seiner RX-PDO1 das Steuerwort vom Master. Knoten 7 empfängt in RX-PDO 2 ebenfalls ein Steuerwort vom Master. Die Konfiguration ist in [Abbildung 10](#) auf Seite 68 dargestellt. Das Objekt 0x6086 wird im Zusammenhang mit dem Dummy-Mapping verwendet.

Das b maXX<sup>®</sup> 5000 mit der Adresse 2 sendet seinen Drehzahl-Istwert und das Statuswort alle 10 ms. Der Knoten 7 sendet sein Statuswort nur nach 3-maligem Empfang eines Sync-Telegramms (vom Master).

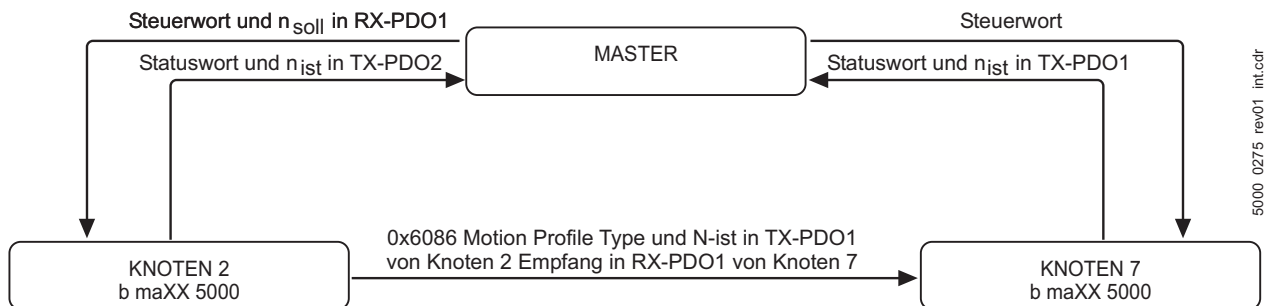


Abbildung 10: Beispiel-Mapping mit zwei b maXX<sup>®</sup> 5000

**1. Schritt: Bestimmen der notwendigen Objekte**

Ermitteln Sie aus der Objektliste (siehe [►Anhang C - Umsetzungstabellen◄](#) ab Seite 143 und [►Verzeichnis der Objekte zur Kommunikationssteuerung◄](#) ab Seite 41) die entsprechende Objekte des Objektverzeichnisses.

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

<b>108.3</b> Statuswort	↔	0x6041 Statuswort
<b>108.1</b> Steuerwort	↔	0x6040 Steuerwort
<b>110.5</b> Sollwertauswahl HLG Eingang	↔	0x6042 Drehzahlsollwert am HLG
<b>18.22</b> Drehzahl-Istwert	↔	0x6044 Control Effort
<b>118.2</b> Positionierung Modus	↔	0x6086 Motion Profile Type

Zum Einstellen des Mappings sind folgende Objekte nötig:

- Knoten 2      0x1A00 (1. Transmit PDO Mapping), 0x1800 (1. Transmit PDO Parameter)  
                   0x1A01 (2. Transmit PDO Mapping), 0x1801 (2. Transmit PDO Parameter)
- Knoten 7      0x1600 (1. Receive PDO Mapping), 0x1400 (1. Receive PDO Parameter)  
                   0x1601 (2. Receive PDO Mapping), 0x1401 (2. Receive PDO Parameter)

**2. Schritt: Mapping konfigurieren**

Zum Einstellen des Mappings werden die SDOs des Expedited Transfers (siehe auch [►Bedarfsdaten \(SDO\)◄](#) ab Seite 51) verwendet. Diese können über einen Master, einen Buskonfigurator o. ä. initiiert werden.

Mapping für Slave 2

Schreiben des ersten zu mappenden Objektes mit Index (0x6086), Subindex (0x00) und Länge (0x10) auf 0x1A00 Subindex 0x01 (TX-PDO 1). Das Objekt soll vom Slave 7 nicht ausgewertet.

Request

		CS	Multiplexor			D0	D1	D2	D3
COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x602	0x08	0x23	0x00	0x1A	0x01	0x10	0x00	0x86	0x60

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
0x582	0x08	0x60	0x00	0x1A	0x01	0x00	0x00	0x00	0x00

Schreiben des zweiten zu mappenden Objektes mit Index (0x6044), Subindex (0x00) und Länge (0x10) auf 0x1A00 Subindex 0x02 (TX-PDO 1).

Request

		CS	Multiplexor			D0	D1	D2	D3
COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x602	0x08	0x23	0x00	0x1A	0x02	0x10	0x00	0x44	0x60

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
0x582	0x08	0x60	0x00	0x1A	0x02	0x00	0x00	0x00	0x00

Schreiben der Anzahl der gemappten Objekte (0x02) auf 0x1A00 Subindex 0x00 (TX-PDO 1).

Request

		CS	Multiplexor			D0	D1	D2	D3
COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x602	0x08	0x2F	0x00	0x1A	0x00	0x02	0x00	0x00	0x00

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
0x582	0x08	0x60	0x00	0x1A	0x00	0x00	0x00	0x00	0x00

Der Inhalt von Objekt 0x1A00 sieht wie folgt aus:

<b>0x1A00</b>	<b>0x00</b>	0x02
	<b>0x01</b>	0x60860010
	<b>0x02</b>	0x60440010

Schreiben des ersten zu mappenden Objektes mit Index (0x6041), Subindex (0x00) und Länge (0x10) auf 0x1A01 Subindex 0x01 (TX-PDO 2).

Request			<b>CS</b>	<b>Multiplexor</b>			<b>D0</b>	<b>D1</b>	<b>D2</b>	<b>D3</b>
	<b>COB-ID</b>	<b>DLC</b>	<b>Byte 0</b>	<b>Byte 1</b>	<b>Byte 2</b>	<b>Byte 3</b>	<b>Byte 4</b>	<b>Byte 5</b>	<b>Byte 6</b>	<b>Byte 7</b>
	0x602	0x08	0x23	0x01	0x1A	0x01	0x10	0x00	0x41	0x60

Response			<b>CS</b>	<b>Multiplexor</b>			<b>D0</b>	<b>D1</b>	<b>D2</b>	<b>D3</b>
	<b>COB-ID</b>	<b>DLC</b>	<b>Byte 0</b>	<b>Byte 1</b>	<b>Byte 2</b>	<b>Byte 3</b>	<b>Byte 4</b>	<b>Byte 5</b>	<b>Byte 6</b>	<b>Byte 7</b>
	0x582	0x08	0x60	0x01	0x1A	0x01	0x00	0x00	0x00	0x00

Schreiben des zweiten zu mappenden Objektes mit Index (0x6044), Subindex (0x00) und Länge (0x10) auf 0x1A01 Subindex 0x02.

Request			<b>CS</b>	<b>Multiplexor</b>			<b>D0</b>	<b>D1</b>	<b>D2</b>	<b>D3</b>
	<b>COB-ID</b>	<b>DLC</b>	<b>Byte 0</b>	<b>Byte 1</b>	<b>Byte 2</b>	<b>Byte 3</b>	<b>Byte 4</b>	<b>Byte 5</b>	<b>Byte 6</b>	<b>Byte 7</b>
	0x602	0x08	0x23	0x01	0x1A	0x02	0x10	0x00	0x44	0x60

Response			<b>CS</b>	<b>Multiplexor</b>			<b>D0</b>	<b>D1</b>	<b>D2</b>	<b>D3</b>
	<b>COB-ID</b>	<b>DLC</b>	<b>Byte 0</b>	<b>Byte 1</b>	<b>Byte 2</b>	<b>Byte 3</b>	<b>Byte 4</b>	<b>Byte 5</b>	<b>Byte 6</b>	<b>Byte 7</b>
	0x582	0x08	0x60	0x10	0x1A	0x02	0x00	0x00	0x00	0x00

Schreiben der Anzahl der gemappten Objekte (0x02) auf 0x1A01 Subindex 0x00 (TX-PDO 2).

Request			<b>CS</b>	<b>Multiplexor</b>			<b>D0</b>	<b>D1</b>	<b>D2</b>	<b>D3</b>
	<b>COB-ID</b>	<b>DLC</b>	<b>Byte 0</b>	<b>Byte 1</b>	<b>Byte 2</b>	<b>Byte 3</b>	<b>Byte 4</b>	<b>Byte 5</b>	<b>Byte 6</b>	<b>Byte 7</b>
	0x602	0x08	0x2F	0x01	0x1A	0x00	0x02	0x00	0x00	0x00

Response			<b>CS</b>	<b>Multiplexor</b>			<b>D0</b>	<b>D1</b>	<b>D2</b>	<b>D3</b>
	<b>COB-ID</b>	<b>DLC</b>	<b>Byte 0</b>	<b>Byte 1</b>	<b>Byte 2</b>	<b>Byte 3</b>	<b>Byte 4</b>	<b>Byte 5</b>	<b>Byte 6</b>	<b>Byte 7</b>
	0x582	0x08	0x60	0x01	0x1A	0x00	0x00	0x00	0x00	0x00

Der Inhalt von Objekt 0x1A01 sieht wie folgt aus:

<b>0x1A01</b>	<b>0x00</b>	0x02
	<b>0x01</b>	0x60410010
	<b>0x02</b>	0x60440010

Schreiben des ersten zu mappenden Objektes mit Index (0x6040), Subindex (0x00) und Länge (0x10) auf 0x1600 Subindex 0x01 (RX-PDO 1).

Request

		CS	Multiplexor			D0	D1	D2	D3
COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x602	0x08	0x23	0x00	0x16	0x01	0x10	0x00	0x40	0x60

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
0x582	0x08	0x60	0x00	0x16	0x01	0x00	0x00	0x00	0x00

Schreiben des zweiten zu mappenden Objektes mit Index (0x6042), Subindex (0x00) und Länge (0x10) auf 0x1600 Subindex 0x02 (RX-PDO 1).

Request

		CS	Multiplexor			D0	D1	D2	D3
COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x602	0x08	0x23	0x00	0x16	0x02	0x10	0x00	0x42	0x60

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
0x582	0x08	0x60	0x00	0x16	0x02	0x00	0x00	0x00	0x00



Schreiben der Anzahl der gemappten Objekte (0x02) auf 0x1600 Subindex 0x00 (RX-PDO 1).

Request			<b>CS</b>	<b>Multiplexor</b>			<b>D0</b>	<b>D1</b>	<b>D2</b>	<b>D3</b>
	<b>COB-ID</b>	<b>DLC</b>	<b>Byte 0</b>	<b>Byte 1</b>	<b>Byte 2</b>	<b>Byte 3</b>	<b>Byte 4</b>	<b>Byte 5</b>	<b>Byte 6</b>	<b>Byte 7</b>
	0x602	0x08	0x2F	0x00	0x16	0x00	0x02	0x00	0x00	0x00

Response			<b>CS</b>	<b>Multiplexor</b>			<b>D0</b>	<b>D1</b>	<b>D2</b>	<b>D3</b>
	<b>COB-ID</b>	<b>DLC</b>	<b>Byte 0</b>	<b>Byte 1</b>	<b>Byte 2</b>	<b>Byte 3</b>	<b>Byte 4</b>	<b>Byte 5</b>	<b>Byte 6</b>	<b>Byte 7</b>
	0x582	0x08	0x60	0x00	0x16	0x00	0x00	0x00	0x00	0x00

Der Inhalt von Objekt 0x1600 sieht wie folgt aus:

<b>0x1600</b>	<b>0x00</b>	0x02
	<b>0x01</b>	0x60400010
	<b>0x02</b>	0x60420010

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

Schreiben des ersten zu mappenden Objektes mit Index (0x6041), Subindex (0x00) und Länge (0x10) auf 0x1A00 Subindex 0x01 (TX-PDO 1).

Request			<b>CS</b>	<b>Multiplexor</b>			<b>D0</b>	<b>D1</b>	<b>D2</b>	<b>D3</b>
	<b>COB-ID</b>	<b>DLC</b>	<b>Byte 0</b>	<b>Byte 1</b>	<b>Byte 2</b>	<b>Byte 3</b>	<b>Byte 4</b>	<b>Byte 5</b>	<b>Byte 6</b>	<b>Byte 7</b>
	0x607	0x08	0x23	0x00	0x1A	0x01	0x10	0x00	0x41	0x60

Response			<b>CS</b>	<b>Multiplexor</b>			<b>D0</b>	<b>D1</b>	<b>D2</b>	<b>D3</b>
	<b>COB-ID</b>	<b>DLC</b>	<b>Byte 0</b>	<b>Byte 1</b>	<b>Byte 2</b>	<b>Byte 3</b>	<b>Byte 4</b>	<b>Byte 5</b>	<b>Byte 6</b>	<b>Byte 7</b>
	0x587	0x08	0x60	0x00	0x1A	0x01	0x00	0x00	0x00	0x00

## 7.10 Prozessdaten (PDO)

Schreiben des zweiten zu mappenden Objektes mit Index (0x6044), Subindex (0x00) und Länge (0x10) auf 0x1A00 Subindex 0x02 (TX-PDO 1).

Request

		CS	Multiplexor			D0	D1	D2	D3
COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x607	0x08	0x23	0x00	0x1A	0x02	0x10	0x00	0x44	0x60

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
0x587	0x08	0x60	0x00	0x1A	0x02	0x00	0x00	0x00	0x00

Schreiben der Anzahl der gemappten Objekte (0x02) auf 0x1A00 Subindex 0x00 (TX-PDO 1).

Request

		CS	Multiplexor			D0	D1	D2	D3
COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x607	0x08	0x2F	0x00	0x1A	0x00	0x02	0x00	0x00	0x00

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
0x587	0x08	0x60	0x00	0x1A	0x00	0x00	0x00	0x00	0x00

Der Inhalt von Objekt 0x1A00 sieht wie folgt aus:

<b>0x1A00</b>	<b>0x00</b>	0x02
	<b>0x01</b>	0x60410010
	<b>0x02</b>	0x60440010

Schreiben des ersten zu mappenden Objektes (Dummy Objekt 16Bit) mit Index (0x0006), Subindex (0x00) und Länge (0x10) auf 0x1600 Subindex 0x01 (RX-PDO 1).

Request			<b>CS</b>	<b>Multiplexor</b>			<b>D0</b>	<b>D1</b>	<b>D2</b>	<b>D3</b>
	<b>COB-ID</b>	<b>DLC</b>	<b>Byte 0</b>	<b>Byte 1</b>	<b>Byte 2</b>	<b>Byte 3</b>	<b>Byte 4</b>	<b>Byte 5</b>	<b>Byte 6</b>	<b>Byte 7</b>
	0x607	0x08	0x23	0x00	0x16	0x01	0x10	0x00	0x06	0x00

Response			<b>CS</b>	<b>Multiplexor</b>			<b>D0</b>	<b>D1</b>	<b>D2</b>	<b>D3</b>
	<b>COB-ID</b>	<b>DLC</b>	<b>Byte 0</b>	<b>Byte 1</b>	<b>Byte 2</b>	<b>Byte 3</b>	<b>Byte 4</b>	<b>Byte 5</b>	<b>Byte 6</b>	<b>Byte 7</b>
	0x587	0x08	0x60	0x00	0x16	0x01	0x00	0x00	0x00	0x00

Schreiben des zweiten zu mappenden Objektes mit Index (0x6042), Subindex (0x00) und Länge (0x10) auf 0x1600 Subindex 0x02.

Request			<b>CS</b>	<b>Multiplexor</b>			<b>D0</b>	<b>D1</b>	<b>D2</b>	<b>D3</b>
	<b>COB-ID</b>	<b>DLC</b>	<b>Byte 0</b>	<b>Byte 1</b>	<b>Byte 2</b>	<b>Byte 3</b>	<b>Byte 4</b>	<b>Byte 5</b>	<b>Byte 6</b>	<b>Byte 7</b>
	0x607	0x08	0x23	0x00	0x16	0x02	0x10	0x00	0x42	0x60

Response			<b>CS</b>	<b>Multiplexor</b>			<b>D0</b>	<b>D1</b>	<b>D2</b>	<b>D3</b>
	<b>COB-ID</b>	<b>DLC</b>	<b>Byte 0</b>	<b>Byte 1</b>	<b>Byte 2</b>	<b>Byte 3</b>	<b>Byte 4</b>	<b>Byte 5</b>	<b>Byte 6</b>	<b>Byte 7</b>
	0x587	0x08	0x60	0x00	0x16	0x02	0x00	0x00	0x00	0x00

Schreiben der Anzahl der gemappten Objekte (0x02) auf 0x1600 Subindex 0x00.

Request			<b>CS</b>	<b>Multiplexor</b>			<b>D0</b>	<b>D1</b>	<b>D2</b>	<b>D3</b>
	<b>COB-ID</b>	<b>DLC</b>	<b>Byte 0</b>	<b>Byte 1</b>	<b>Byte 2</b>	<b>Byte 3</b>	<b>Byte 4</b>	<b>Byte 5</b>	<b>Byte 6</b>	<b>Byte 7</b>
	0x607	0x08	0x2F	0x00	0x16	0x00	0x02	0x00	0x00	0x00

Response			<b>CS</b>	<b>Multiplexor</b>			<b>D0</b>	<b>D1</b>	<b>D2</b>	<b>D3</b>
	<b>COB-ID</b>	<b>DLC</b>	<b>Byte 0</b>	<b>Byte 1</b>	<b>Byte 2</b>	<b>Byte 3</b>	<b>Byte 4</b>	<b>Byte 5</b>	<b>Byte 6</b>	<b>Byte 7</b>
	0x587	0x08	0x60	0x00	0x16	0x00	0x00	0x00	0x00	0x00

Der Inhalt von Objekt 0x1600 sieht wie folgt aus:

<b>0x1600</b>	<b>0x00</b>	0x02
	<b>0x01</b>	0x00060010
	<b>0x02</b>	0x60420010

Schreiben des ersten zu mappenden Objektes mit Index (0x6040), Subindex (0x00) und Länge (0x10) auf 0x1601 Subindex 0x01 (RX-PDO 2).

Request

		CS	Multiplexor			D0	D1	D2	D3
COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x607	0x08	0x23	0x01	0x16	0x01	0x10	0x00	0x40	0x60

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
0x587	0x08	0x60	0x01	0x16	0x01	0x00	0x00	0x00	0x00

Schreiben der Anzahl der gemappten Objekte (0x01) auf 0x1601 Subindex 0x00.

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
0x607	0x08	0x2F	0x01	0x16	0x00	0x01	0x00	0x00	0x00

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
0x587	0x08	0x60	0x01	0x16	0x00	0x00	0x00	0x00	0x00

Der Inhalt von Objekt 0x1601 sieht wie folgt aus:

<b>0x1601</b>	<b>0x00</b>	0x01
	<b>0x01</b>	0x60400010

Der Datenaustausch zwischen den b maXX<sup>®</sup> 5000 über die PDOs ist in ►Abbildung 11◄ dargestellt. Beispiel für eine Querkommunikation. Der Drehzahlwert von Slave 2 wird Drehzahlsollwert von Slave 7.

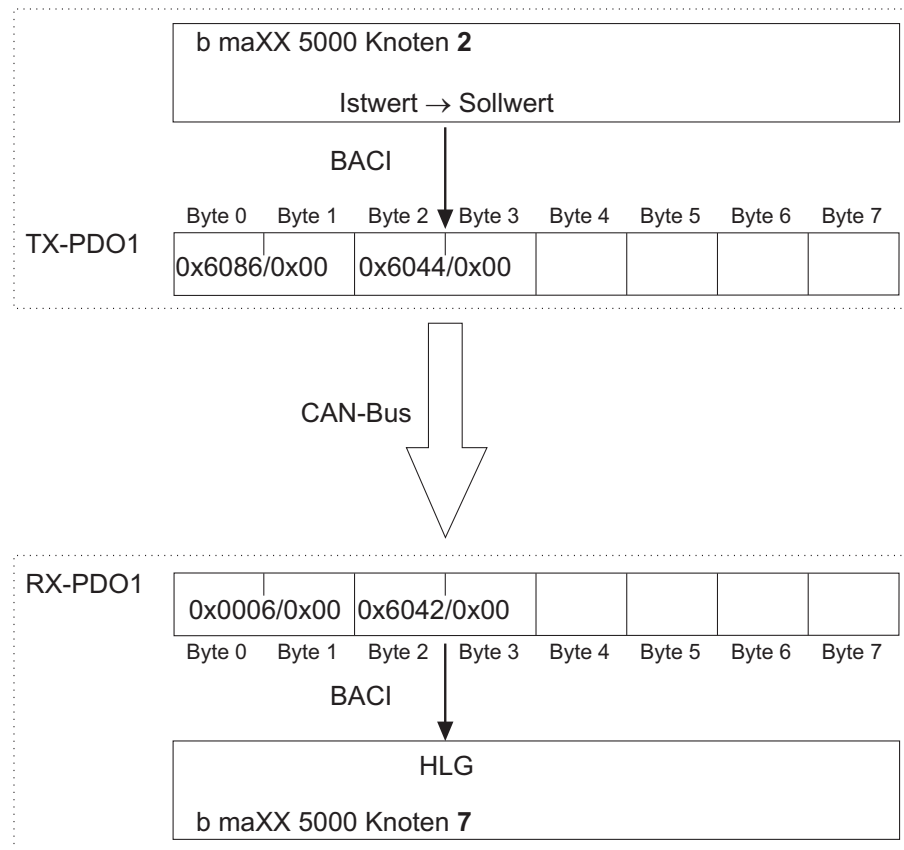


Abbildung 11: Telegrammaufbau für Beispiel-Mapping

Von der TX-PDO1 des b maXX<sup>®</sup> 5000 Knoten 2 werden im b maXX<sup>®</sup> 5000 Knoten 7 nur die Bytes 2 und 3 ausgewertet, weil in der RX-PDO1 nur die Byte 2 und 3 gültig mit Parameternummern verknüpft sind.

Über SDOs werden die Kommunikationsparameter beider Slaves eingestellt. Damit eine Kommunikationsbeziehung aufgebaut werden kann, müssen die COB-IDs von Sender und Empfänger übereinstimmen. Die COB-ID der TX-PDO1 von Slave 2 ist defaultmäßig auf 0x182 eingestellt. Die COB-ID der RX-PDO1 von Slave 7 lautet defaultmäßig 0x207. Welche COB-ID verwendet werden soll, liegt in der Verantwortung des Anwenders. Im Beispiel wird die COB-ID 0x207 verwendet.

### Kommunikationsparameter für Slave 2:

Auf Objekt 0x1800 Subindex 0x01 wird der Wert 0x207 eingetragen. Auf Subindex 0x02 wird der Wert „0xFF“ (ereignisgesteuert) geschrieben.

<b>0x1800</b>		
	<b>0x01</b>	0x207
	<b>0x02</b>	0xFF

Auf Objekt 0x1801 Subindex 0x02 wird der Wert 0xFE (Timer-getriggert) eingetragen. Auf Subindex 0x05 wird der Wert „0x0A“ geschrieben (Timerwert = 10 ms).

<b>0x1801</b>		
	<b>0x02</b>	0xFE
	<b>0x05</b>	0x0A

Die übrigen Subindizes behalten ihre Defaultwerte.

Der Triggertyp von RX-PDO1 wird auf ereignisgesteuert (0x1400 Subindex 0x02 = 0xFF) gestellt.

### Kommunikationsparameter für Slave 7:

Auf Objekt 0x1400 Subindex 0x02 wird der Wert „0xFF“ (ereignisgesteuert) geschrieben. In Subindex 0x01 steht defaultmäßig 0x207.

<b>0x1400</b>		
	<b>0x01</b>	0x207
	<b>0x02</b>	0xFF

Die beiden TX-PDOs von Knoten 7 behalten ihre Default-COB-IDs. Der Triggertyp von TX-PDO1 wird mit dem Wert 0x03 auf SYNC getriggert eingestellt. TX-PDO2 wird als ereignisgesteuert parametrisiert.

<b>0x1800</b>		
	<b>0x01</b>	0x187
	<b>0x02</b>	0x03

<b>0x1801</b>		
	<b>0x01</b>	0x287
	<b>0x02</b>	0xFF

#### 7.10.4 Eintrag in die Feldbus Prozessdaten des Reglers

Es können maximal 16 zyklische Soll- und 16 zyklische Istwerte zwischen dem CANopen-Slave und dem b maXX<sup>®</sup>-Regler gleichzeitig ausgetauscht werden. Alle Werte werden in einem Zyklus aktualisiert. Die Soll-/Istwerte können bei CANopen auf je vier PDOs verteilt sein.

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

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

Für RX-PDOs gilt das selbe Verfahren.

Die Einträge in die Feldbus Prozessdatenliste des Reglers erfolgen lückenfrei, mit dem 1. Objekt von PDO1 beginnend.



#### HINWEIS!

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

Beginnend beim ersten Objekt von PDO 1 werden die Inhalte der PDOs abwechselnd auf ihre Gültigkeit für die Prozessdaten-Konfiguration (kein Dummy) abgefragt. Ist das Objekt gültig, dann wird dieses an die nächste freie Stelle der Prozessdaten-Konfiguration eingetragen. Ist das PDO-Mapping fehlerhaft (falsche Parameternummern o.ä.), wird keine zyklische Kommunikation zwischen Slave und b maXX<sup>®</sup> 5000 gestartet.



#### HINWEIS!

Werden in den vorhandenen PDOs gleicher Richtung mehrmals die selbe Objekt-nummer gemappt, so erscheint das Objekt mehrmals in der Prozessdaten-Konfiguration.

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





# GRUNDLAGEN ETHERCAT

## 8.1 Literatur zum Thema EtherCAT

---

Für grundlegende Informationen zum Thema EtherCAT empfehlen wir folgende Literatur:

- EtherCAT Communication Specification  
EtherCAT Technology Group (ETG)
- EtherCAT Slave Controller IP Core for Altera FPGAs Hardware Data Sheet  
EtherCAT Technology Group (ETG)
- [www.ethercat.org](http://www.ethercat.org)  
EtherCAT Technology Group (ETG)  
Ostendstraße 196  
D-90482 Nürnberg

## 8.2 Grundlagen EtherCAT

---

Die Real Time Ethernet Control Automation Technology (EtherCAT) wurde von der Fa. Beckhoff als neuer Feldbus Standard entwickelt. Um EtherCAT als offenen Standard vertreiben zu können, wurde zu diesem Zweck die Ethernet Technology Group ETG gegründet. Die ETG gilt als eine Vereinigung von Interessenten, Herstellern und Anwendern. Diese Vereinigung hatte im Dezember 2006: 421 Mitglieder aus 31 Ländern. Diese Mitglieder tragen in technischen Arbeitskreisen zur Weiterentwicklung der Technologie bei.

### 8.2.1 Topologie Daten

Es können mehrere Bustopologien genutzt werden, z.B. Linien-, Baum- oder Sterntopologie (► [Abbildung 12](#)◄ auf Seite 82).

Bis zu 65535 Teilnehmer können erreicht werden, dadurch ist die Netzwerkausdehnung nahezu unbegrenzt (>500 km).

Für die Übertragung reicht ein Standard Ethernet Patch Kabel (CAT5). Es werden die Vollduplex-Eigenschaften von 100 BASE-TX vollständig ausgenutzt, so dass effektive Datenraten von >100 MBit/s (>90 % von 2 x 100 MBit/s) erreichbar sind. Die Leitungslänge zwischen zwei Teilnehmern wird mit bis zu 100 m angegeben.

Alternativ können auch Lichtwellenleiter-Varianten von 50 m bis 2000 m eingesetzt werden

Vorteilhaft ist auch, dass während des Betriebes Geräte hinzugeführt bzw. entfernt werden können „Hot Connect / Disconnect von Bussegmenten“.

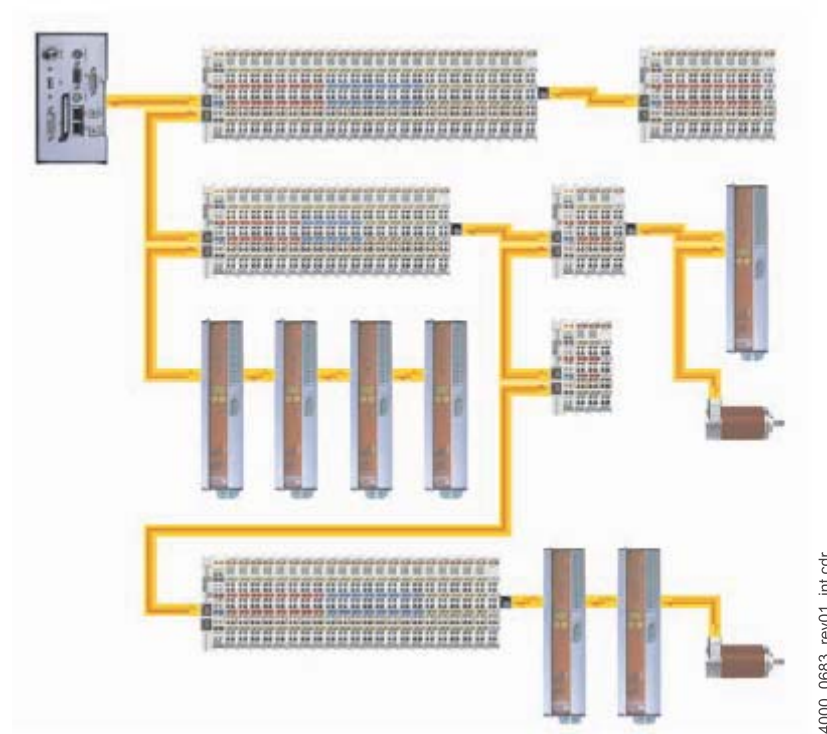


Abbildung 12: Flexible Topologie: Linie, Baum oder Stern [1]

8.2.2 Rahmenaufbau

Das EtherCAT-Protokoll ist speziell für die Prozessdaten optimiert worden. Dies gelingt Dank eines speziellen Ethertypes (88A4h), der direkt in einem Ethernet-Frame transportiert wird. Es kann aus mehreren Subtelegramms bestehen, die jeweils einen Speicherbereich des bis zu 4 Gigabyte grossen logischen Prozessabbildes bedienen. Das Adressieren der Daten kann wahlfrei erfolgen, dabei ist die Reihenfolge der physikalischen Reihenfolge unabhängig von der datentechnischen Reihenfolge der Teilnehmer im Netz.

Das Versenden erfolgt mit einem minimalen Versatz von wenigen Bitzeiten.

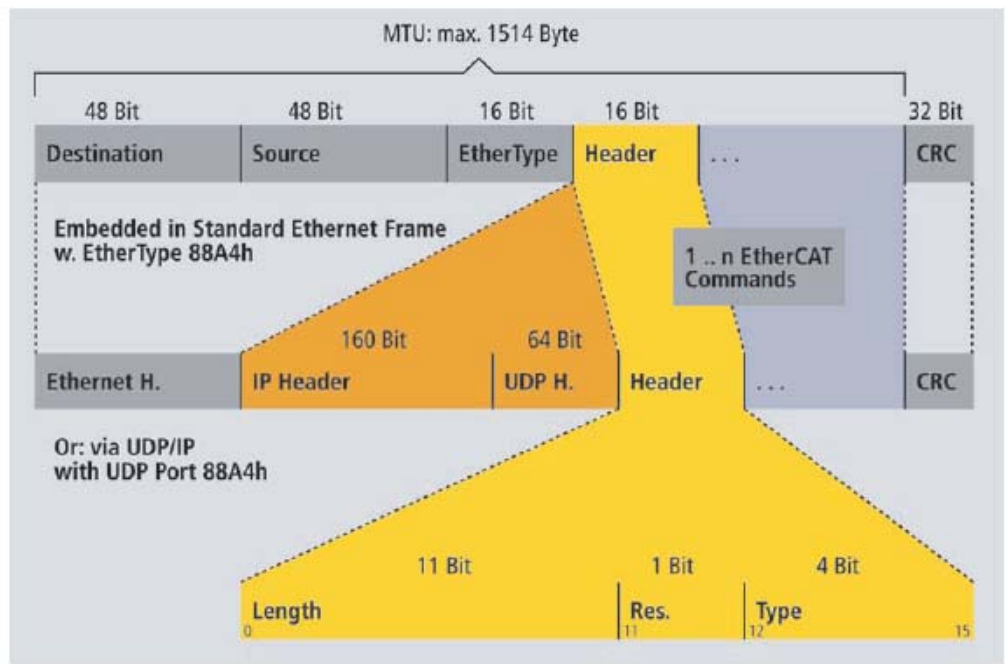


Abbildung 13: EtherCAT: Standard -IEEE 802.3-Frames [1]

### 8.2.3 Geräteprofile

Für verschiedene Aufgaben in der Automatisierung gibt es speziell dafür zugeschnitten Feldbussysteme z.B. CANopen. Die Feldbussysteme sind oft in Standards zusammengefasst. Beim EtherCAT wird darauf verzichtet eigene Profile für schon vorhandene Standards zu entwickeln, vielmehr wird auf schon vorhandenes aufgebaut.

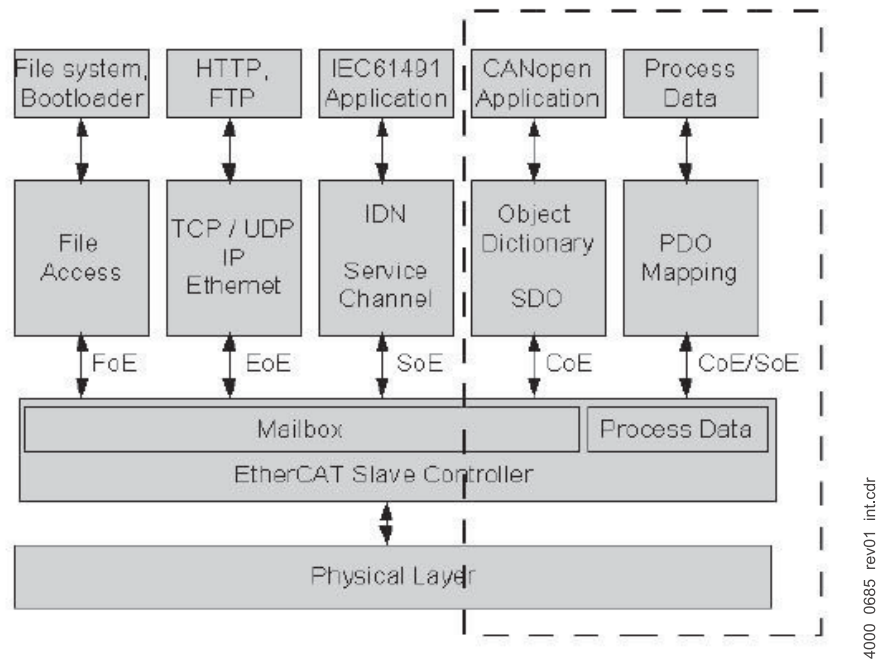


Abbildung 14: Device Profile bei EtherCAT[1]

### 8.2.4 EtherCAT Frame-Struktur

EtherCAT Telegramme werden, eingebettet in einem Ethernet Telegramm, mit diesen gesendet. Das Telegramm enthält einen Ethernet-Header (a), einen EtherCAT-Header (b) und folgend dann n EtherCAT-Telegramme.

Das EtherCAT-Telegramm (c) seinerseits ist aufgeteilt in einen EtherCAT-Header, Datenbereich und einen Counter Bereich.

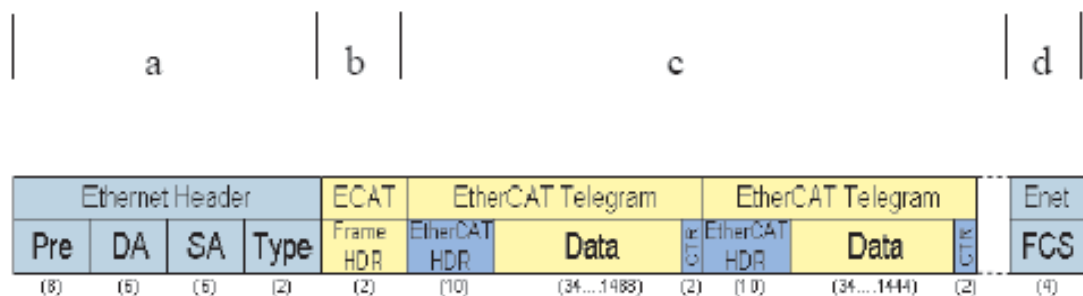


Abbildung 15: EtherCAT - Frame [1]

**a) Ethernet Header:**

- Pre** Die Präambel dient dem Empfänger zur Synchronisation und zur Lokalisierung, sie besteht aus einer Folge von '10101010' je Byte.  
Im Präambel ist das SFD Byte enthalten: SFD: „start of frame delimiter“ kennzeichnet den Frame-Anfang; Bitmuster 10101011.
- DA** Destination MAC address.
- SA** Source MAC address.  
Ziel-/Quelladresse: spezifizieren das empfangende (möglicherweise mehrere) und das zu sendende Ethernet Telegramm; innerhalb eines LAN nur einheitliche Längen erlaubt (16 oder 48 Bit)
- Type** Definiert den EtherType. Der EtherType zeigt an welches Protokoll der nächsthöheren Schicht\* innerhalb der Nutzdaten benutzt wird. 88A4<sub>hex</sub> definiert den EtherCAT Type.

\* ISO-OSI-Schichtmodell

**b) EtherCAT Frame Header:**

Das EtherCAT Frame Header hat eine Länge von 2 Byte. Hier sind die Information über die Datenlänge und den Datentype des folgenden Telegramms enthalten.

**c) EtherCAT-Telegramm:**

Das EtherCAT -Telegram unterteilt sich in ein Telegramm-Header, den zu übertragenden Daten und dem Working Counter. Der Working Counter wird von jedem bearbeitendem Slave inkrementiert.

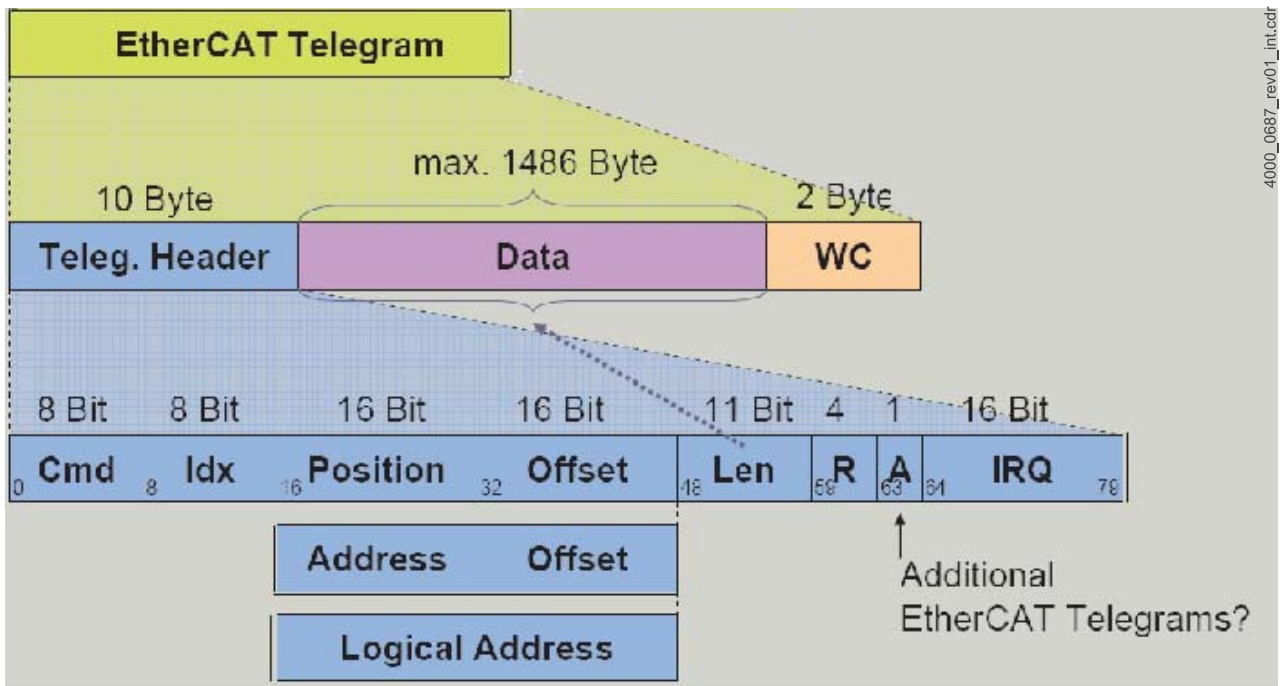


Abbildung 16: EtherCAT Telegramm [1]

Der „EtherCAT Telegram Header“ hat eine Länge von 10 Byte. Er enthält Informationen über die folgenden Daten.

- **CMD**, 1 Byte. Kodiert das vom Master abgesetzte EtherCAT Kommando, das entweder schreibend oder lesend gekennzeichnet sein kann.
- **IDX**, 1 Byte. Index des Frames. Wird vom Slave unverändert weitergegeben, damit kann der Master das Telegramm beim Empfang wieder leicht zuordnen.
- Die **Position**, zeigt die Adresse oder die physikalische Position des Slaves an. Zusätzlich wird ein Offset mit angegeben.  
Aufgeteilt in:

ADP (2 Byte) Adress-Page	abhängig vom verwendeten Kommando
ADO (2 Byte) Adress-Offset	abhängig vom verwendeten Kommando
INT Interrupt-Feld	
- **LEN**, 2 Bytes.  
In den Bits 0 bis 10 wird die Länge des nachfolgenden Datenblocks gespeichert. Die Bits 11 bis 15 dienen für verschiedene Zwecke als Flags.  
Bit 63(A) zeigt an ob ein zusätzliches EtherCAT Telegramm nachfolgend gesendet wird.

Der Datenbereich ist maximal 1486 Byte groß. Im Datenbereich eines Ethernet-Frames können mehrere EtherCAT-Frames und somit mehrere Kommandos an verschiedene Slaves enthalten sein. Die physikalische Reihenfolge der Slaves im Strang muss dabei in der Regel nicht beachtet werden. Durch die Eigenschaft, dass mehrere EtherCAT-Kommandos in einem Ethernet-Frame Platz finden und durch ein Memory Mapping in den Slaves, welches den Zugriff auf die Speicherbereiche mehrerer Slaves mit einem EtherCAT-Kommando erlaubt, wird die Nutzdatenrate erheblich erhöht. Dadurch wird das Problem des hohen Overheads von Ethernet bei geringem jedoch wiederholten Datenaufkommen gelöst.

Das EtherCAT-Telegramm endet mit einem 2 Byte großen Working Counter. Jeder Slave, der eine Nachricht erfolgreich empfangen hat inkrementiert den Counter. Der Master hat damit die Möglichkeit Fehler zu erkennen.

#### d) Frame Check Sequence (FCS):

Das FCS Feld stellt eine 32-Bit-CRC-Prüfsumme dar. Ist die Prüfsumme nach Verwendung der FCS ungleich Null, war die Übertragung fehlerhaft.

8.2.5 EtherCAT Kommunikationszustände

Das AL Management in EtherCAT beschreibt das Handling der EtherCAT State Machine (ESM). Es sind der Zustand und die Zustandsänderung des jeweiligen Slaves in einer Applikation beschrieben. Der aktuelle Zustand des ECT Slaves wird im Statusregister aufgezeigt und Zustandsänderungen werden im Control Register, vom Master initiiert, angezeigt.

EtherCAT definiert vier Kommunikationszustände. Die Kommunikationszustände (State) und deren Übergänge (Transitionen) siehe [▶Abbildung 17◀](#).

State Änderungen werden vom Master angefragt. Der Slave antwortet korrekt wenn er den Wechsel vollzogen hat oder setzt eine Fehlermeldung ab wenn der Wechsel nicht vollzogen werden konnte.

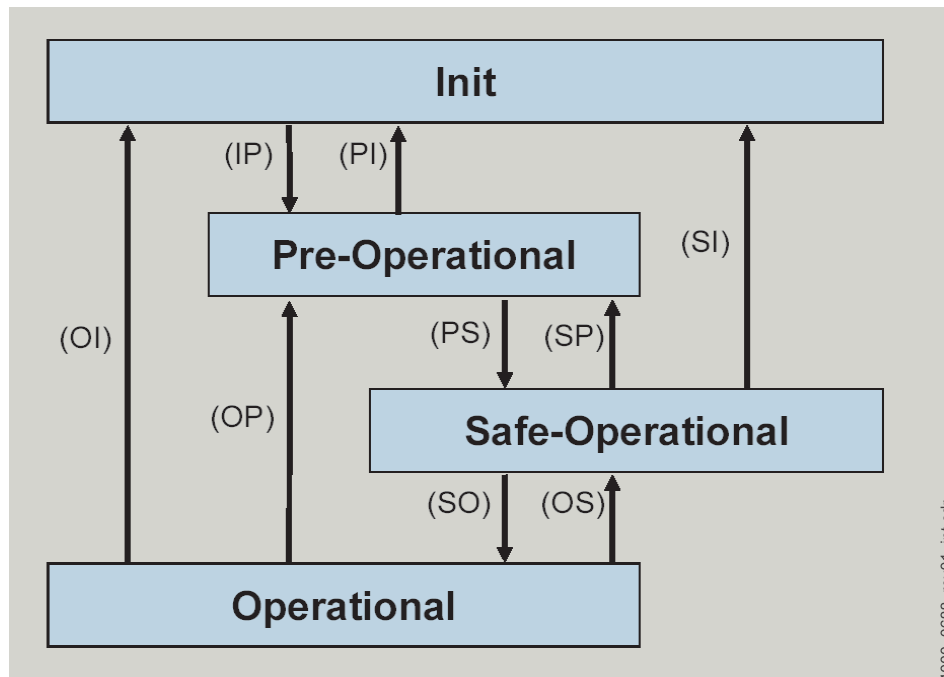


Abbildung 17: EtherCAT Kommunikationsübergänge [1]

Zustände:

- **Init:**  
Initialisierung des Slaves. In der Init Phase ist keine direkte Kommunikation auf der Applikationsebene möglich
- **Pre-Operational:**  
In diesem Zustand kann eine Mailbox für eine Service Daten Kommunikation konfiguriert sein (wenn der Slave diese unterstützt). Service Daten Kommunikation ist dann möglich, aber noch keine Prozessdatenkommunikation.
- **Safe-Operational:**  
In diesem Zustand ist weiterhin die Service Daten Kommunikation möglich. Außerdem werden nur vom Slave ausgehende Daten, TX-Data gesendet. RX-Daten vom Master werden noch ignoriert. Mailbox ist weiterhin möglich.

- **Operational:**  
Mailbox und Zyklische Kommunikation in beide Richtungen (TxPDO und RxPDO) sind nun möglich. Mailbox ist weiterhin möglich.

Die Transitionen werden in der folgenden Tabelle aufgezeigt.

State transition	Local management service
IP	Start Mailbox Communication
PI	Stop Mailbox Communication
PS	Start Input Update
SP	Stop Input Update
SO	Start Output Update
OS	Stop Output Update
OP	Stop Output Update, Stop Input Update
Si	Stop Input Update, Stop Mailbox Communication
OI	Stop Output Update, Stop Input Update, Stop Mailbox Communication

Übergänge:

Können die Anforderung des Masters zu einen Zustandswechsel durch den Slave nicht erfüllt werden, weil z.B. ein falsches Mapping vorliegt, so hat der Slave die Möglichkeit durch Senden einer Fehlermeldung dies dem Master mitzuteilen. Diese Nachricht ist ähnlich wie die der Gerätesteuerung unterteilt.

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
A0 <sub>hex</sub>	00 <sub>hex</sub>	08 <sub>hex</sub>	04 <sub>hex</sub>	00 <sub>hex</sub>	00 <sub>hex</sub>	00 <sub>hex</sub>	00 <sub>hex</sub>

**Byte 0** und **Byte 1** beinhalten den Emergency Error Code.

Vom CoE Standard sind zwei Einträge definiert.

A000<sub>hex</sub>: Übergang von PRE-OPERATIONAL nach SAFE-OPERATIONAL war nicht erfolgreich

A001<sub>hex</sub>: Übergang von SAFE-OPERATIONAL nach OPERATIONAL war nicht erfolgreich

**Byte 2:**

Anhand der folgenden Tabelle ist zu erkennen welche Meldung abgesetzt wird, wenn eine falsche Parametrierung der SyncManager erfolgt.

SyncManager2 (Prozessdaten out RxPDO)	08 <sub>hex</sub>	SyncManager falsch parametrier
	09 <sub>hex</sub>	PDO Länge stimmt nicht mit der Mappinglänge der Objekte überein
	0A <sub>hex</sub>	SyncManager-Einstellungen an unzulässiger Adresse
SyncManager3 (Prozessdaten in TxPDO)	0C <sub>hex</sub>	SyncManager falsch parametrier
	0D <sub>hex</sub>	PDO Länge stimmt nicht mit der Mappinglänge der Objekte überein
	0E <sub>hex</sub>	SyncManager-Einstellungen an unzulässiger Adresse



Für die SyncManager0 und SyncManager1 kann keine Meldung abgesetzt werden, weil damit die Mailboxen beschrieben werden. Sind die Mailboxen falsch konfiguriert, bleibt der Slave im Zustand Init. Der nicht erfolgte Wechsel nach PRE-OPERATIONAL wird in dem Fall nur über den AI-Status dem Master mitgeteilt.

Bei falschen Syncmanager-Einstellungen wird zuerst die EMCY für SyncManager2 abgesetzt, egal ob SyncManager3 auch falsch konfiguriert ist. Erst wenn der erste Fehler beseitigt ist, wird dann die nächste Emergency gesendet.

**Byte 3:**

Gibt die Anzahl der folgenden Bytes an, entweder 4 Byte (oder 2 Byte EMCY-Codes bei Fehler des Gerätes).

**Byte 4-7:**

	Byte 4	Byte 5	Byte 6	Byte 7
SM2 Adressen Fehler	0	0	0	0
SM2 falsche Länge	High-Byte	Low-Byte	High-Byte	Low-Byte
	Minimale Länge der Syncmanager		Maximale Länge der Syncmanager	
SM2 falsch parametrier	High-Byte	Low-Byte	High-Byte	Low-Byte
	Kleinste zulässige Adresse		Größte zulässige Adresse	
SM3 Adressen Fehler	0	0	0	0
SM3 falsche Länge	High-Byte	Low-Byte	High-Byte	Low-Byte
	Minimale Länge der Syncmanager		Maximale Länge der Syncmanager	
SM3 falsch parametrier	High-Byte	Low-Byte	High-Byte	Low-Byte
	Kleinste zulässige Adresse		Größte zulässige Adresse	

Herstellerspezifischer Fehlercode:

**Byte 0 und Byte 1:**

A0A0<sub>hex</sub>:

der Fehlercode erscheint, wenn der Antrieb synchron laufen soll, aber nach einer gewissen Zeit (in Abhängigkeit vom b maXX<sup>®</sup>-Gerät und vom Gerätezustand, von 100 ms bis 30 s) immer noch nicht synchron läuft.

Byte 2 beinhaltet FF<sub>hex</sub> und Byte 4-7 hat den Wert Null.

**Synchronisierung**

Die exakte Synchronisierung von Teilnehmern bei EtherCAT erfolgt nach dem Prinzip der verteilten Uhren, wie im neuen Standard IEEE 1588 beschrieben. Jeder Slave hat eine unabhängig laufende Uhr implementiert. Dabei wird die Uhrzeit der Haupt-Uhr via EtherCAT zu den Neben-Uhren übertragen. Um die Laufzeit des Synchronisations-Telegramms zu berücksichtigen wird eine Laufzeitmessung durchgeführt. Dafür sendet der Master ein Broadcast-Telegramm aus, in das alle Slaves den Empfangszeitpunkt dieses Broadcast-Telegramms bezüglich ihrer Uhr notieren. Dadurch sind die Laufzeiten bestimmt und können vom Master entsprechend berücksichtigt werden. Bei EtherCAT wird die Haupt-Uhr in einem Slave-Gerät konfiguriert, sodass auch hierfür keine spezielle Hardware im Master erforderlich ist. Die Synchronisationsgenauigkeit ist dabei deutlich

unter einer  $\mu\text{s}$ , bei 300 Teilnehmern und 120 m Leitungslänge wurden Abweichungen von  $\pm 20 \text{ ns}$  erzielt [1].

Die notwendigen Einstellungen des Slaves durch den Master bzw. Einstellungen im Datensatz werden in [►Synchronisation \(SYNC\)◄](#) ab Seite 103 beschrieben.

### 8.2.6 Ethernet over EtherCAT (EoE) - TCP/IP- Tunneling über EtherCAT

---

Für die Ethernet-Kommunikation zu EtherCAT-Slaves (z. B. zum b maXX<sup>®</sup>-Regler mit EtherCAT-Slave, hier speziell PROPROG-Kommunikation für das Bedientool ProDrive) werden die TCP-Pakete innerhalb der EtherCAT-Pakete übertragen (tunneling). In diesem Fall muss für jeden EtherCAT-Slave eine eigene IP-Adresse eingestellt werden. Über diese IP-Adresse wird der EtherCAT-Slave als Ethernet-Teilnehmer angesprochen.

Die Einstellung der IP-Adresse ist:

192.168.XXX.XXX

192.168 wird fest vergeben.

XXX bedeutet Einstellung der DIP Schalter auf der HW

Ein EtherCAT Master hat auch die Möglichkeit die IP-Adresse zu ändern (falls dieses vom Master unterstützt wird).

Dabei kann eine beliebige IP-Adresse gewählt werden.

Die Portnummer für die PROPROG-Kommunikation ist  $5043_{\text{hex}}$  (=  $20547_{\text{dez}}$ ).

Weil die EoE Kommunikation über die Mailboxen des EtherCAT erfolgt, sollte die Abfrage der Mailboxen häufiger erfolgen (zwischen 5 ms und 50 ms).

# CoE AM B MAXX 2500 / 3300 / 5000

## 9.1 Allgemeines

---

Der b maXX 2500 / 3300 / 5000 CoE Slave verbindet den b maXX 2500 / 3300 / 5000 über den EtherCAT-Bus mit anderen EtherCAT-Knoten (z.B. PC, SPS, weitere b maXX Geräte, I/O-Module).

Informationen zur Installation und zum Umgang mit der Gerätereihe b maXX 3300 / 5000 finden Sie in der Dokumentation 5.11018 / 5.09021.

Informationen zur Programmierung des b maXX 3300 / 5000 Reglers finden Sie im Parameterhandbuch 5.12001 / 5.09022.

## 9.2 Adresseinstellung

---

Die Einstellung der Knoten-Adresse des b maXX 3300 / 5000 CoE Slave ist in der Betriebsanleitung b maXX 3300 / 5500 (Dokumentation 5.11018 / 5.09021) beschrieben.

## 9.3 XML-Datei

---

In der XML-Datei sind Informationen enthalten, die ein Master braucht, um z. B. die FMMU (Fieldbus Memory Management Unit) und die SyncManager auf dem CoE-Slave zu konfigurieren.

Die XML-Datei ist im Downloadbereich der Baumüller Homepage erhältlich.

## 9.4 Diagnose

---

CANopen folgt dem Ethernet-Standard IEEE 802.3. Somit können für die Systemdiagnose Standardtools wie Wireshark (Freeware) oder OmniPeek und Standardgeräte wie Hubs bzw. Switches und PC-Netzwerkschnittstellen verwendet werden.

### 9.5 Datenaustausch und Parametrierung

Der Zugriff auf Daten oder Parameter findet bei CoE über CAN Objekte statt. Detaillierte Beschreibung siehe [►Datenaustausch und Parametrierung◄](#) ab Seite 40.

### 9.6 Verzeichnis der Objekte zur Kommunikationssteuerung

In diesem Abschnitt finden Sie alle vom Baumüller CoE Slave unterstützten Objekte des kommunikationsspezifischen Bereiches des Objektverzeichnisses nach CiA 301.

Name	Index	Subindex	Datentyp	Defaultwert
Device Type	0x1000	0x00	UINT32	0x00020192

Dieses Objekt kann nur gelesen werden und enthält die Information, um welches Gerät es sich handelt (Drive nach CiA<sup>®</sup> 402), nur lesbar.

Name	Index	Subindex	Datentyp	Defaultwert
Error Register	0x1001	0x00	UINT8	0x0

Dieses Objekt kann nur gelesen werden. Das Objekt 0x1001 enthält eine Fehlerbitleiste, die folgende Bedeutung hat:

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

COB-ID	DLC	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x80 + Adresse	0x08	Emergency Error Code		Error Register	Herstellerspezifisches Fehlerfeld				

EMCY-Telegramm für Error Reset / No Error

Name	Index	Subindex	Datentyp	Defaultwert
Manufacturer Device Name	0x1008	0x00	VString	-

Dieses Objekt ist nur lesbar. Es enthält die Gerätebezeichnung.

Name	Index	Subindex	Datentyp	Defaultwert
Manufacturer Hardware Version	0x1009	0x00	VString	-

Dieses Objekt ist nur lesbar. Es enthält die aktuelle Hardware-Version des Reglers aus Parameter **102.25**.

Name	Index	Subindex	Datentyp	Defaultwert
Manufacturer Software Version	0x100A	0x00	VString	-

Dieses Objekt ist nur lesbar. Es enthält die aktuelle Software-Version des Reglers, z. B. die Zeichenfolge: „01.08.00 S (Build 109)“.

Name	Index	Subindex	Datentyp	Defaultwert
Identity Object	0x1018	0x00	UINT8	0x04
Vendor ID		0x01	UINT32	0x0000015A
Product Code		0x02	UINT32	0x26483052
Revision Number		0x03	UINT32	0x00000000
Serial Number		0x04	UINT32	0x00000000

In diesem Objekt sind einige Informationen über das Gerät enthalten.

Name	Index	Subindex	Datentyp	Defaultwert
RPDO Mapping Axis 1	0x1600	0x00	UINT8	0x01
		0x01	UINT32	0x60400010
		:	:	
		n	UINT32	

Dieses Objekt enthält den Inhalt der Receive-PDO für Achse 1. In Subindex 0x00 steht die Gesamtzahl der nachfolgenden Einträge. Die Gesamtzahl der gemappten Objekte darf den Sollwertrahmen von max. 16 Objekten nicht überschreiten (siehe auch [►PDO-Mapping](#) ab Seite 101).

Name	Index	Subindex	Datentyp	Defaultwert
RPDO Mapping Axis 2	0x1700	0x00	UINT8	0x01
		0x01	UINT32	0x60400010
		:	:	
		n	UINT32	

Dieses Objekt enthält den Inhalt der Receive-PDO für Achse 2. In Subindex 0x00 steht die Gesamtzahl der nachfolgenden Einträge. Die Gesamtzahl der gemappten Objekte darf den Sollwertrahmen von max. 16 Objekten nicht überschreiten (siehe auch [▷PDO-Mapping◀](#) ab Seite 101).

Name	Index	Subindex	Datentyp	Defaultwert
TPDO Mapping Axis 1	0x1A00	0x00	UINT8	0x01
		0x01	UINT32	0x60410010
		:	:	
		n	UINT32	

Dieses Objekt enthält den Inhalt der Transmit-PDO für Achse 1. In Subindex 0x00 steht die Gesamtzahl der nachfolgenden Einträge. Die Gesamtzahl der gemappten Objekte darf den Istwertrahmen von max. 16 Objekten nicht überschreiten. (siehe auch [▷PDO-Mapping◀](#) ab Seite 101).

Name	Index	Subindex	Datentyp	Defaultwert
TPDO Mapping Axis 2	0x1B00	0x00	UINT8	0x01
		0x01	UINT32	0x60410010
		:	:	
		n	UINT32	

Dieses Objekt enthält den Inhalt der Transmit-PDO für Achse 2. In Subindex 0x00 steht die Gesamtzahl der nachfolgenden Einträge. Die Gesamtzahl der gemappten Objekte darf den Istwertrahmen von max. 16 Objekten nicht überschreiten. (siehe auch [▷PDO-Mapping◀](#) ab Seite 101).

Name	Index	Subindex	Datentyp	Defaultwert
Sync Manager Communication Type	0x1C00	0x00	UINT8	0x04
Communication Type Manager 0 Mailbox receive (Master to Master)		0x01	UINT8	1 (SM0)
Communication Type Manager 1 Mailbox transmit (Slave to Master)		0x02	UINT8	2 (SM1)
Communication Type Manager 2 RPDO (Master to Slave)		0x03	UINT8	3 (SM2)
Communication Type Manager 3 TPDO (Slave to Master)		0x04	UINT8	4 (SM3)

In diesem Objekt sind Informationen über die Sync Manager Einstellungen enthalten. Das Objekt ist nur lesbar.

Name	Index	Subindex	Datentyp	Defaultwert
Sync Manager Communication Type Channel 2 Number of assigned RPDOs	0x1C12	0x00	UINT8	0 ... 255
PDO Mapping Object Index des zugehörigen RPDO		0x01	UINT16	0x1600:RPDO

Das Objekt beinhaltet Informationen über den Kommunikationstyp des Sync Manager Kanal 2 (Process Data Output). Es wird angezeigt wie viele und welche RPDOs vom Slave unterstützt werden. Beim b maXX CoE-Slave ist das eine RPDO.

Name	Index	Subindex	Datentyp	Defaultwert
Sync Manager Communication Type Channel 3 Number of assigned TPDOs	0x1C13	0x00	UINT8	0 ... 255
PDO Mapping Object Index des zugehörigen TPDO		0x01	UINT16	0x1A00:TPDO

Das Objekt beinhaltet Informationen über den Kommunikationstyp des Sync Manager Kanal 3 (Process Data Input). Es wird angezeigt wie viele und welche TPDOs vom Slave unterstützt werden. Beim b maXX CoE-Slave ist das eine TPDO.

Name	Index	Subindex	Datentyp	Defaultwert
Sync Manager output parameter	0x1C32	0x00	UINT8	0x20
Sync mode		0x01	UINT16	siehe unten
Cycle time		0x02	UINT32	Zykluszeit in ns

Sync Mode:

0	keine Synchronisierung des Reglers
0x02	DC-Mode Sync0 Synchronisierung auf AL Event Sync0
0x22	SM2 Synchronisierung auf AL Event Sync Manager 2 (RPDO vom Master)

In diesem Objekt sind Informationen über die Synchronisierungsart des Sync Managers enthalten. Die Zykluszeit wird in ns angegeben, z. B. 1 ms = 1.000.000 ns

Name	Index	Subindex	Datentyp	Defaultwert
Sync Manager input parameter	0x1C33	0x00	UINT8	0x20
Sync mode		0x01	UINT16	siehe unten
Cycle time		0x02	UINT32	Zykluszeit in ns

Sync Mode:

0	keine Synchronisierung des Reglers
0x02	DC-Mode Sync0 Synchronisierung auf AL Event Sync0
0x22	SM2 Synchronisierung auf AL Event Sync Manager 2 (RPDO vom Master)

In diesem Objekt sind Informationen über die Synchronisierungsart des Sync Managers enthalten. Die Zykluszeit wird in ns angegeben, z. B. 1 ms = 1.000.000 ns.

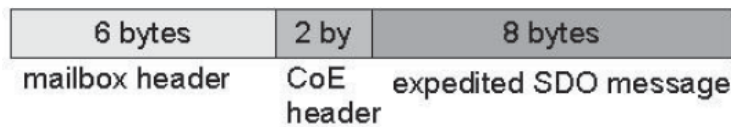
## 9.7 Bedarfsdaten (SDO)

Service Daten Objekte (SDO) dienen dem Austausch von Nachrichten ohne Echtzeitanforderungen. Die SDOs werden zur Parametrierung der Slaves und zur Einstellung der Kommunikationsbeziehungen für die PDOs verwendet. Der Datenzugriff erfolgt aus-



schließlich über die Objektliste. SDOs sind immer bestätigte Daten, d.h. der Sender erhält eine Quittung vom Empfänger. Für die Übertragung der SDO werden in ECT die Mailbox Dienste in Anspruch genommen.

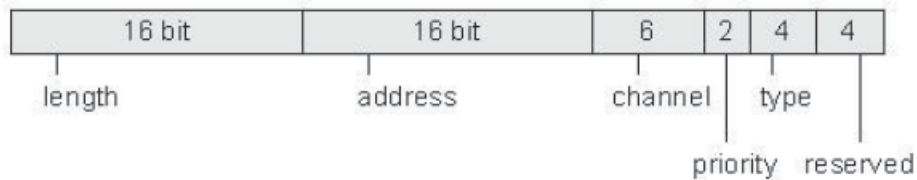
Die Mailbox ist aufgeteilt in einen Telegramm-Header und den Mailbox Datenbytes. In [▶Abbildung 18◀](#) auf Seite 97 ist der Mailbox -Aufbau schematisch dargestellt.



4000\_0689\_rev01\_int.cdr

Abbildung 18: Aufbau der Mailbox

Der Aufbau des Mailbox-Headers ist weiter unterteilt:



4000\_0690\_rev01\_int.cdr

Abbildung 19: Mailbox-Header

Length	Anzahl der Bytes der Mailbox, die anschließend dem Header folgen
Address	ECT-Adresse des jeweiligen Slaves
Type	Typ des benutzten Protokolls der Mailbox, z.B. 3. CoE (CANopen over EtherCAT)

Der CoE-Header unterteilt sich wie folgt:



4000\_0691\_rev01\_int.cdr

Abbildung 20: CoE-Header

PDO number	Mit der Mailbox ist es auch möglich PDOs zu übertragen. Hier wird angegeben, ob die Mailbox für PDO Übertragung konfiguriert wurde.
Type	0: Reserved 1: Emergency Message 2: SDO Request 3: SDO Response 4: TPDO 5: RPDO 6: Remote Übertragung der TPDO 7: Remote Übertragung der RPDO 8: SDO Information 9 - 15: Reserved

### 9.7.1 Telegrammaufbau nach CANopen

Der Telegrammaufbau bei ECT ist in den Datenbytes nach dem CANopen-Standard definiert. Allerdings kann die Grenze von 8 Byte aufgebrochen werden, je nachdem ob ein Slave dieses unterstützt oder nicht.

Das Datenfeld des Datentelegramms (8 Byte) für eine SDO gliedert sich in drei Teile, einem Command Specifier CS (1 Byte), einem Multiplexor M (3 Byte) und dem eigentlichen Nutzdatenbereich D0 - D3 (4 Byte).

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
CS	M	M	M	D0	D1	D2	D3

Der Multiplexor M besteht aus dem 16 Bit Index eines Objektes und dem dazugehörigen acht Bit breiten Subindex.

Der Command Specifier CS für eine Schreibanforderung im Expedited transfer lautet für die verschiedenen Längen:

Datenlängen in D0 - D3	Command Specifier CS
1 Byte	0x2F
2 Byte	0x2B
4 Byte	0x23

Der CS für die Antwort auf eine Schreibanforderung beträgt CS = 0x60 oder im Fehlerfall CS = 0x80.

Der Command Specifier CS für eine Leseanforderung im Expedited transfer lautet CS = 0x40.

Die Antwort für die verschiedenen Längen ist dann:

Datenlängen in D0 - D3	Command Specifier CS
1 Byte	0x4F
2 Byte	0x4B
4 Byte	0x43

### 9.7.2 Arten des SDO-Transfers

Die Baumüller Anschaltung unterstützt den Expedited Transfer und den Segmented Transfer, wobei letzterer nur für die Objekte 0x1008, 0x1009 und 0x100A Manufacturer Device Name genutzt wird.

#### Expedited Transfer

Es können Objekte geschrieben oder gelesen werden, deren Daten maximal 4 Byte umfassen. Es sind nur zwei Telegramme erforderlich, eine Anforderung und eine Antwort. Alle Objekte mit den Indizes 0x1XXX, 0x4XXX, 0x6XXX sind über Expedited SDOs ansprechbar mit Ausnahme der Objekte 0x1008, 0x1009 und 0x100A.

#### Segmented Transfer

Für Objekte mit Daten größer als 4 Byte ist der Segmented Transfer erforderlich. Dabei wird die 8 Byte Grenze für die Nutzdaten aufgebrochen. Dies ist nur beim Lesen der Objekte 0x1008, 0x1009 und 0x100A möglich.

### 9.7.3 Fehlerreaktionen

Fehlerhafte SDO-Zugriffe werden mit Abort Codes abgewiesen. Der Aufbau dieser Abort-Telegramme ist identisch zu dem auf [Seite 98](#) dargestellten SDO-Telegramm. Das Datenfeld enthält einen Abort Code mit 4 Bytes.

Bei fehlerhaften Zugriffen auf kommunikationsspezifische Objekte (0x1XXX) werden folgende Meldungen unterschieden:

Abort Code	Bedeutung
05 <sub>hex</sub> 03 <sub>hex</sub> 00 <sub>hex</sub> 00 <sub>hex</sub>	Parameter inkonsistent (Toggle Bit hat nicht gewechselt)
05 <sub>hex</sub> 04 <sub>hex</sub> 00 <sub>hex</sub> 00 <sub>hex</sub>	SDO Protokoll Zeitüberschreitung
05 <sub>hex</sub> 04 <sub>hex</sub> 00 <sub>hex</sub> 01 <sub>hex</sub>	Client/Server Kommando Specific CS nicht gültig oder unbekannt.
05 <sub>hex</sub> 04 <sub>hex</sub> 00 <sub>hex</sub> 05 <sub>hex</sub>	Speicherbereich überschritten
06 <sub>hex</sub> 01 <sub>hex</sub> 00 <sub>hex</sub> 00 <sub>hex</sub>	Fehler im Datenformat
06 <sub>hex</sub> 01 <sub>hex</sub> 00 <sub>hex</sub> 01 <sub>hex</sub>	Lesen auf ein nur schreibbares Objekt
06 <sub>hex</sub> 01 <sub>hex</sub> 00 <sub>hex</sub> 02 <sub>hex</sub>	Schreiben auf ein schreibgeschütztes Objekt
06 <sub>hex</sub> 02 <sub>hex</sub> 00 <sub>hex</sub> 00 <sub>hex</sub>	Objekt existiert nicht im Objektverzeichnis
06 <sub>hex</sub> 04 <sub>hex</sub> 00 <sub>hex</sub> 41 <sub>hex</sub>	Daten können nicht gemappt werden (z. B. falsche Längenangabe)
06 <sub>hex</sub> 04 <sub>hex</sub> 00 <sub>hex</sub> 42 <sub>hex</sub>	Die Objektnummer und die Länge des Objekts welches gemappt werden soll liegt außerhalb der PDO Länge

Abort Code	Bedeutung
06 <sub>hex</sub> 04 <sub>hex</sub> 00 <sub>hex</sub> 43 <sub>hex</sub>	Allgemeine Parameterinkompatibilität
06 <sub>hex</sub> 06 <sub>hex</sub> 00 <sub>hex</sub> 00 <sub>hex</sub>	Zugriffsfehler auf die Hardware (speichern/laden vom Flash)
06 <sub>hex</sub> 07 <sub>hex</sub> 00 <sub>hex</sub> 10 <sub>hex</sub>	Inkorrekte Datenwert Länge
06 <sub>hex</sub> 09 <sub>hex</sub> 00 <sub>hex</sub> 11 <sub>hex</sub>	Subindex existiert nicht
06 <sub>hex</sub> 09 <sub>hex</sub> 00 <sub>hex</sub> 30 <sub>hex</sub>	Wertebereich überschritten (bei Schreibzugriffen)
06 <sub>hex</sub> 09 <sub>hex</sub> 00 <sub>hex</sub> 31 <sub>hex</sub>	Wert zu groß (bei Schreibzugriffen)
06 <sub>hex</sub> 09 <sub>hex</sub> 00 <sub>hex</sub> 32 <sub>hex</sub>	Wert zu klein (bei Schreibzugriffen)
08 <sub>hex</sub> 00 <sub>hex</sub> 00 <sub>hex</sub> 00 <sub>hex</sub>	Allgemeiner Fehler
08 <sub>hex</sub> 00 <sub>hex</sub> 00 <sub>hex</sub> 20 <sub>hex</sub>	Daten können nicht zur Applikation übertragen oder abgespeichert werden
08 <sub>hex</sub> 00 <sub>hex</sub> 00 <sub>hex</sub> 22 <sub>hex</sub>	Daten können aufgrund des derzeitigen Kommunikationszustandes nicht gemappt werden (z. B. Mapping ändern im Zustand OPERATIONAL)

## 9.8 Prozessdaten (PDO)

Prozess-Daten-Objekte (PDO) sind auf den Austausch von Daten mit Echtzeitanforderungen optimiert. In den PDOs auf der CoE-Optionskarte können maximal 64 Byte je Kommunikationsrichtung für die Nutzdatenübertragung/zyklische Kommunikation verwendet werden. Für den Datenaustausch über die PDOs muss die exakte Lage der Objekte im EtherCAT-Frame vor Beginn der Kommunikation zwischen Sender und Empfänger definiert werden. Die „Fieldbus Memory Management Unit FMMU“ ordnet den logischen Adressraum des EtherCAT Busses den physikalischen Adressraum des Slaves zu. Die Konfiguration findet üblicherweise in der Init-Phase durch den Master statt.

Die Prozessdaten des EtherCAT Slaves werden durch die SyncManager Channels beschrieben. Jeder SyncManager beschreibt einen zusammenhängenden Speicherbereich der zyklischen Daten. Mit den SyncManager wird auch die Mailbox beschrieben. Der EtherCAT Slave unterstützt 4 SyncManager, 2 für die Mailbox einen je Richtung und 2 SyncManager als RPDO bzw. TPDO. Die Konfiguration der SyncManager erfolgt wie auch bei den FMMUs durch den Master. Bitweise Adressierung ist im CoE-Standard vorgesehen, bei der CoE-Optionskarte aber nicht möglich (nur byteweise Adressierung wird unterstützt).

Für die Übertragung der zyklischen Daten und die Synchronisierung des Reglers sind drei Synchronisierungsarten möglich. Synchronisierung deaktiviert (der Betrieb ist nur bis zum Zustand SAFE-OPERATIONAL möglich), Synchronisierung auf SyncManager2 (RPDO Achse 1) und Synchronisierung auf Distributed Clocks DC.

**HINWEIS!**

Alle in den PDOs konfigurierten Objekte werden zwischen der CoE Slave und dem b maXX<sup>®</sup>-Regler als zyklische Daten übertragen (siehe auch [►Kommunikationsablauf◄](#) ab Seite 25). Da die zyklische Datenübertragung (speziell die RPDOs) nur im Zustand OPERATIONAL stattfindet, sollte auch nur in diesem Zustand die Kommunikationsüberwachung in ProDrive aktiviert werden, weil in den anderen Zuständen (z. B. PRE-OPERATIONAL) sonst eine Fehlermeldung generiert wird. Diese muss dann nach Übergang zu OPERATIONAL quittiert werden.

### 9.8.1 PDO-Mapping

Mapping ist ein Verfahren zur Zuordnung von Variablen/Objekten an PDOs. Mit den PDOs werden diese Variablen/Objekte über den Bus transportiert. Durch das Mapping wird der zyklischen Datenaustausch projiziert. Für diese Parametrierung werden SDOs genutzt. Das Mapping wird über im Objektverzeichnis adressierbare Objekte eingestellt. Für jede PDO existiert ein solches Objekt.

Prozess-Daten-Objekt	Objekt für Inhalt
TPDO Achse 1	0x1A00
TPDO Achse 2	0x1B00
RPDO Achse 1	0x1600
RPDO Achse 2	0x1700

**HINWEIS!**

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

Durch das Mapping wird der logische Inhalt der PDOs festgelegt. Für diese Festlegung werden bestimmte Angaben über das zu mappende Objekt benötigt: Objektindex, Subindex und Länge des Datums. Aus dem Objektverzeichnis werden die entsprechenden Objekte in das Mapping-Objekt eingetragen. Die Reihenfolge dieses Eintrages, festgelegt durch den Subindex des Mapping-Objektes, bestimmt die Reihenfolge der Daten im EtherCAT-Telegramm. In den Mapping-Objekten (0x1600, 0x1A00) werden die zu mappenden Objekte an die entsprechenden Subindizes (mit 0x01 beginnend) geschrieben, z. B. wird auf Objekt 0x1600 Subindex 0x01 der Wert 0x60400010 eingetragen. Dies bedeutet, die ersten beiden Bytes der in RPDO Achse 1 empfangenen Daten werden auf das Steuerwort (Objekt 0x6040 Subindex 0x00) geschrieben. Das Objekt 0x6040 ist auf den b maXX<sup>®</sup>-Parameter **108.1** Steuerwort umgesetzt (siehe auch [►Anhang C - Umsetzungstabellen◄](#) ab Seite 143). Damit wird das erste Wort des in RPDO Achse 1 empfangenen Telegramms auf das Steuerwort des b maXX<sup>®</sup> geschrieben. In Subindex 0x00 muss die Anzahl der zu mappenden Objekte (Anzahl der mit gültigen Objekten belegten Subindizes) eingetragen werden. Ein Beispiel für das Mapping ist im [►Beispiel für das PDO-Mapping◄](#) ab Seite 104 erläutert.

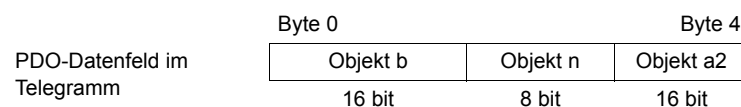
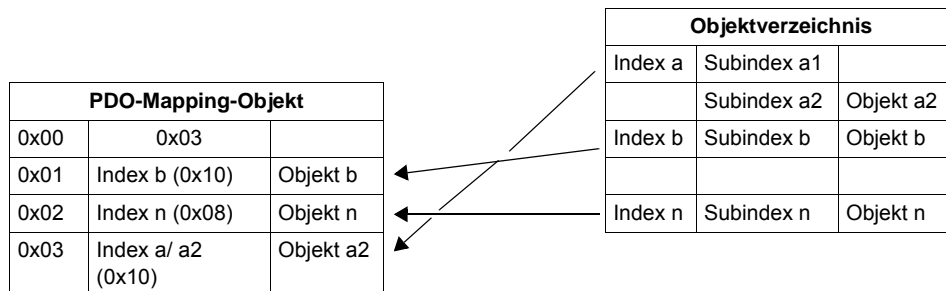


Abbildung 21: Mapping

Um ein bestehendes Mapping zu löschen, können die Werte in den Subindizes mit neuen Objekten überschrieben oder zu Null gesetzt werden. Mit dem Schreiben des Wertes „0“ auf den Subindex 0x00 der entsprechenden PDO (0x1600, 0x1A00) deaktiviert man die PDO.



### HINWEIS!

Bei der Einstellung des Mappings in den PDOs (0x1600, 0x1A00) ist jeweils der Subindex 0x00 mit der richtigen Anzahl der gemappten Objekte zuletzt zu beschreiben.

### Sollwerte

Die zulässigen zyklische Sollwerte sind in einer Tabelle mit der Spalte „PDO-Mapping“ als „RX“ gekennzeichnet. Die Tabelle ist im Anhang B.2 (für die 6000'er Objektnummern) zu finden.

### Istwerte

Die zulässigen zyklische Istwerte sind in einer Tabelle mit der Spalte „PDO-Mapping“ als „TX“ gekennzeichnet. Die Tabelle ist im Anhang B.2 (für die 6000'er Objektnummern) zu finden. Eine detaillierte Beschreibung der b maXX<sup>®</sup>-Parameter finden Sie ebenfalls im Parameterhandbuch b maXX<sup>®</sup> 5000 (5.09022) bzw. Parameterhandbuch b maXX<sup>®</sup> 3300 (5.12001).

Fehlerhafte Mapping-Konfigurationen (unzulässige Objekte in 0x1600, 0x1A00) werden durch Abort Codes über SDO gemeldet.

Die zyklischen Soll-/Istwerte werden lückenfrei in der Prozessdaten-Konfiguration initialisiert, d. h. der erste Sollwert von PDO Achse 1 steht an erster Stelle, der zweite Sollwert von PDO Achse 1 an zweiter Stelle usw. Anschließend folgen die Sollwerte der PDO Achse 2. Analog gilt für die Istwert-Initialisierung der erste Istwert von PDO Achse 1 steht an erster Stelle, der zweite Istwert von PDO Achse 1 an zweiter Stelle usw.

**Dummy-Mapping** Das Optionsmodul CoE-Slave stellt 2 Dummy-Objekte bereit: ein 1 Byte Dummy-Objekt und ein 2 Byte Dummy-Objekt, welches ebenfalls in eine PDO gemappt werden kann. Diese Objekte haben die Indices 0x0005 (1 Byte Dummy) und 0x0006 (2 Byte Dummy). Das Dummy-Objekt dient als Platzhalter, um nur bestimmte Objekte innerhalb eines Telegramms zu verwenden (siehe auch [►Beispiel für das PDO-Mapping◄](#) ab Seite 104).

**HINWEIS!**

Das aktuell eingestellte Mapping geht nach dem Ausschalten verloren.

### 9.8.2 Synchronisation (SYNC)

Zum Synchronisieren des Reglers kann auf zwei Synchronisiermechanismen zurück gegriffen werden. Erstens Synchronisieren auf SM2 (RPDO) und zweitens mit den Distributed Clocks (DC). Die DC sind im [►Grundlagen EtherCAT◄](#) ab Seite 81 kurz vorgestellt worden.

Beide Arten lösen auf dem Optionsmodul CoE einen Interrupt aus, der an den b maXX<sup>®</sup> Regler weitergeleitet wird. Damit kann dieses Signal zur Synchronisation des b maXX<sup>®</sup> Reglers verwendet werden.

#### Einstellung der Zykluszeit

Die Einstellung der Zykluszeit sollte vorzugsweise über das FBO 0x1C32 Subindex 0x02 erfolgen. Dabei wird die Zykluszeit in ns angegeben. Eingabe 1 000 000 entspricht z. B. dann einer Zykluszeit von 1 ms. Die Eingabe über das FBO wird vom Slave vorgezogen und kann die im Datensatz abgespeicherte Zykluszeit ändern. Wird eine Zykluszeit ungleich den zulässigen Zykluszeiten 8 ms, 4 ms, 2 ms, 1 ms, 500 µs und 250 µs geschrieben, wird die Synchronisation abgeschaltet (in ProDrive unter „Feldbus-Slave“ zu erkennen).

Es gibt noch die Möglichkeit die Zykluszeit über das Visualisierungstool ProDrive des Reglers einzustellen. Auf der Seite „Feldbus-Slave“ wird die zulässige (siehe oben) Zykluszeit eingestellt. Nach der Einstellung muss der Datensatz abgespeichert und der Regler neu gebootet werden. Default wird auf SM2 synchronisiert, wenn die DC (Distributed Clocks) nicht aktiviert wurden. Beim Synchronisieren auf SM2 ist im FPGA der Optionskarte eine Funktion aktiviert, die den Jitter der RPDO vom Master ausgleicht (PLL). Bei Verwendung der DC wird diese Funktion deaktiviert.

Wird keine der beiden Möglichkeiten genutzt, wird das Synchronisieren des Reglers deaktiviert. **Dabei ist zu beachten, dass der Slave nicht mehr nach OPERATIONAL geschaltet werden kann.**

**HINWEIS!**

Wird im laufenden Betrieb die Synchronisationsart gewechselt, muss der Regler neu gebootet werden.

### Master-Einstellungen für die Verwendung von Distributed Clocks (DC)

Damit die DC im ECT-FPGA (ASIC) überhaupt aktiviert werden kann, muss die Register Adresse des ECT ASIC 0x981 (siehe hierzu [3]) vom Master folgend beschrieben werden:

Bit 0  $\Rightarrow$  1 „Activate cyclic operation“

Bit 1  $\Rightarrow$  1 „Activate Sync0“

Die Überprüfung im Slave findet beim Übergang PRE-OPERATIONAL nach SAFE-OPERATIONAL der ECT Zustandsmaschine statt.

Über FBO 0x1C32 Subindex 0x01 wird eingestellt, welche Synchronisierungsart gewünscht ist:

Wert 0x0  $\Rightarrow$  freerun, nicht synchronisiert

Wert 0x2  $\Rightarrow$  DC Sync0, synchronisiert mit DC IRQ Sync0

Wert 0x22  $\Rightarrow$  SyncSM2, synchronisiert mit SyncManager IRQ des SM2 (SyncManager2 RxPDO)

Alle anderen Synchronisierungsarten werden nicht unterstützt. Wird trotzdem versucht darauf zu schreiben wird eine Fehlermeldung (0x06010000 = Fehler im Datenformat) generiert.

Die Zykluszeit muss über FBO 0x1C32 Subindex 0x02 oder über Register 0x9A0 im ECT eingestellt werden (DWORD in ns). Der Master muss dafür sorgen, dass die Sollwert-Telegramme bei eingestellten DC 200  $\mu$ s bis 50  $\mu$ s vor dem Sync-Event nicht gesendet werden. Mit Hilfe des Sync-Offsets in der ProDrive Seite „Feldbus-Slave“ ist es möglich diesen „verbotenen Bereich“ zu verschieben, wenn der Master keine Möglichkeit hat die Sollwert-Telegramme aus dem „verbotenen Bereich“ zu schieben.

Wird in beiden Fällen keine Zykluszeit eingestellt, wird die Zykluszeit entsprechend der Seite Feldbus-Slave in ProDrive übernommen.

Wenn die DC aktiviert werden, wird durch den Slave (im FPGA) die „PLL“ deaktiviert und das Sync0 Signal direkt von den DC zum Regler weitergegeben.

Beim Übergang nach SAFE-OPERATIONAL wird überprüft ob im Register 0x981 die DC aktiviert wurden. Wenn nicht wird FBO 0x1C32 Subindex 0x01 auf den Wert 0x22 (Synchronisierung auf SM2) zurückgesetzt.

### 9.8.3 Beispiel für das PDO-Mapping

Das Optionsmodul CoE-Slave mit der Knotenadresse 1 empfängt vom Master einen Drehzahlsollwert in RPDO Achse 1. Dieser Drehzahlsollwert soll auf den Hochlaufgeber-Eingang geschrieben werden. Weiterhin erhält Knoten 1 in seiner RPDO Achse 1 das Steuerwort vom Master. Als Istwerte sendet der Knoten 1 seinen Drehzahl-Istwert und das Statuswort. Die Zykluszeit soll 1 ms betragen und auf DC oder auf SM2 synchronisiert werden. Der Slave wird mit Hilfe des Masters aus der Init Phase in den Zustand PRE-OPERATIONAL gebracht. Dies erfolgt in CoE mit den definierten AL Event Mechanismus.



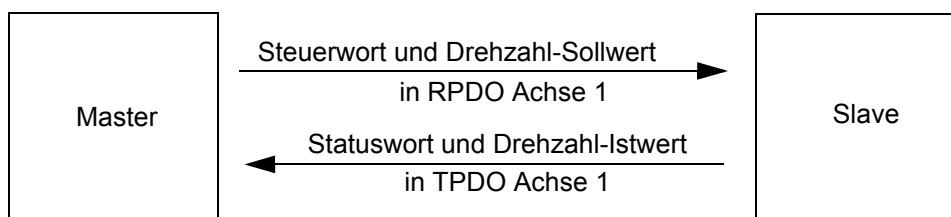


Abbildung 22: Beispiel-Mapping mit einem b maXX®

### 1. Schritt: Bestimmen der notwendigen Objekte

Ermitteln Sie aus der Objektliste (siehe [►Anhang C - Umsetzungstabellen◄](#) ab Seite 143 und [►Verzeichnis der Objekte zur Kommunikationssteuerung◄](#) ab Seite 92) die entsprechende Objekte des Objektverzeichnisses.

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

Parameternummer, z.B. beim b maXX® 5000	CANopen Feldbus-Objekt
<b>108.3</b> Statuswort	☒ 0x6041 Statuswort
<b>108.1</b> Steuerwort	☒ 0x6040 Steuerwort
<b>110.5</b> Sollwertauswahl HLG Eingang	☒ 0x6042 Drehzahl Sollwert am HLG
<b>18.22</b> Drehzahl-Istwert	☒ 0x6044 Control Effort

### 2. Schritt: Mapping konfigurieren

Schreiben des ersten zu mappenden Objektes mit Index (0x6041), Subindex (0x00) und Länge (0x10) auf 0x1A00 Subindex 0x01 (TPDO Achse 1).

Schreiben des zweiten zu mappenden Objektes mit Index (0x6044), Subindex (0x00) und Länge (0x10) auf 0x1A00 Subindex 0x02 (TPDO Achse 1).

Schreiben der Anzahl der gemappten Objekte (0x02) auf 0x1A00 Subindex 0x00 (TPDO Achse 1).

Der Inhalt von Objekt 0x1A00 sieht wie folgt aus:

0x1A00	0x00	0x02
	<b>0x01</b>	0x60410010
	<b>0x02</b>	0x60440010

Schreiben des ersten zu mappenden Objektes mit Index (0x6040), Subindex (0x00) und Länge (0x10) auf 0x1600 Subindex 0x01 (RPDO Achse 1).

Schreiben des zweiten zu mappenden Objektes mit Index (0x6042), Subindex (0x00) und Länge (0x10) auf 0x1600 Subindex 0x02 (RPDO Achse 1).

Schreiben der Anzahl der gemappten Objekte (0x02) auf 0x1600 Subindex 0x00 (RPDO Achse 1).

Der Inhalt von Objekt 0x1600<sub>hex</sub> sieht wie folgt aus:

<b>0x1600</b>	<b>0x00</b>	0x02
	<b>0x01</b>	0x60400010
	<b>0x02</b>	0x60420010

Jetzt muss noch die Zykluszeit von 1 ms eingestellt werden, sowie der Synchronisiermodus (SM2 oder Sync0). Dies geschieht über SDOs mit dem FBO 0x1C32 Subindex 0x01/0x02. Zusätzlich müssen beim Synchronisieren auf DC noch Einstellungen im EtherCAT ASIC vorgenommen werden. Siehe dazu [Synchronisation \(SYNC\)](#) ab Seite 103.

### 3. Schritt: Synchronisierung

- Synchronisieren auf DC

FBO 0x1C32 Subindex 0x01 Type der Synchronisation:

Request

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x2B	0x32	0x1C	0x01	0x02			

CS = 0x2B für 1 Byte;

0x32 und 0x1C wird gekreuzt und setzt sich zusammen aus 0x1C32;

SIX1 = 0x02 für den Sync Type DC.

Die Antwort darauf lautet:

Response

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x60	0x32	0x1C	0x01	0x02			

- Synchronisieren auf SM2

FBO 0x1C32 Subindex 0x01 Type der Synchronisation:

Request

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x2B	0x32	0x1C	0x01	0x22			

CS = 0x2B für 1 Byte;

0x32 und 0x1C wird gekreuzt und setzt sich zusammen aus 0x1C32;

SIX1 = 0x22 für den Sync Type SyncManager2.

Die Antwort darauf lautet:

Response

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x60	0x32	0x1C	0x01	0x22			

#### 9.8.4 Eintrag in die Feldbus-Prozessdaten des Reglers

Es können maximal 16 zyklische Soll- und 16 zyklische Istwerte zwischen dem CoE-Slave und dem b maXX<sup>®</sup>-Regler gleichzeitig ausgetauscht werden. Alle Werte werden in einem Zyklus aktualisiert. Die Soll-/Istwerte können bei CoE auf je eine PDO verteilt sein. Es können 64 Byte je Richtung zyklisch übertragen werden.

Die Aktualisierungszeit für die Verarbeitung der PDOs im Regler ist abhängig von der im b maXX<sup>®</sup>-Regler eingestellten Kommunikationszeit (siehe Kommunikation zum b maXX<sup>®</sup>-Regler). Die Einträge erfolgen lückenfrei, mit dem 1. Objekt von PDO Achse 1 beginnend werden die Inhalte abwechselnd auf ihre Gültigkeit (kein Dummy) abgefragt. Ist das Objekt gültig, dann wird dieses an die nächste freie Stelle in die Feldbus Prozessdatenliste des Reglers eingetragen. Ist das PDO-Mapping fehlerhaft (falsche Parameternummern o.ä.), wird keine zyklische Kommunikation zwischen Slave und b maXX<sup>®</sup>-Regler gestartet.

**HINWEIS!**

Werden in dem vorhandenen PDO gleicher Richtung mehrmals identische Objektnummern gemappt, so erscheint das Objekt mehrmals in der Prozessdaten-Konfiguration.

Hierbei muss beachtet werden, dass sich die Objekte im Mapping unter Umständen gegenseitig beeinflussen.

**HINWEIS!**

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



# GRUNDLAGEN POWERLINK

## 10.1 Literatur zum Thema POWERLINK

---

Für Informationen zum Thema POWERLINK empfehlen wir folgende Literatur:

- Ethernet POWERLINK Communication Profile Specification  
EPSG Draft Standard 301  
Ethernet POWERLINK Standardization Group (EPSG)
- Ethernet POWERLINK XML Device Description  
EPSG Draft Standard 311  
Ethernet POWERLINK Standardization Group (EPSG)
- [www.ethernet-powerlink.org](http://www.ethernet-powerlink.org)  
Ethernet POWERLINK Standardization Group (EPSG)  
Bonsaiweg 6  
D-15370 Fredersdorf

## 10.2 POWERLINK Allgemeines

---

POWERLINK Version 2 (Ethernettyp 0x88ab) ist ein offengelegtes Feldbussystem auf Echtzeit-Ethernetbasis, welches die CANopen-Mechanismen vollständig integriert.

Als physikalische Basis dienen Twisted-Pair-Kabel (100Base-TX).

### Netzwerk

Bei POWERLINK sind beliebige Topologien möglich. Das Netzwerk kann als Linien-, Baum-, Stern- oder Ringstruktur realisiert werden. Es sind auch Kombinationen zulässig. Es besteht die Möglichkeit während der Laufzeit Geräte hinzuzufügen oder zu entfernen (Hot-Plugging).

### Buszugriff

Der Buszugriff erfolgt über das CSMA/CD-Verfahren (Carrier Sense Multiple Access / Collision Detection). Da jeder Teilnehmer das Recht hat, nach Erkennung der notwendigen Busruhe, mit dem Senden seiner Nachricht zu beginnen, können Kollisionen entstehen. In diesem Falle werden die Kollisionen erkannt (Collision Detection) und das Senden nach Ablauf eines zufälligen Zeitintervalls erneut versucht. Somit ist gewährleistet, dass die Übertragung ohne Datenverlust funktioniert. Aus diesem Grund ist es natürlich notwendig, dass jeder Teilnehmer eindeutig im Netzwerk durch die jeweilige MAC-Adresse identifiziert werden kann.

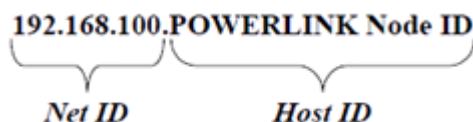
Der Einsatz von Switches kann zu undeterministischem Verhalten im Netzwerk führen.

### MAC-Adressierung

Jeder Teilnehmer kann unaufgefordert Telegramme senden. Daher ist eine eindeutige Sende- und Empfangsadresse notwendig. Dies wird durch die MAC-Adresse realisiert erreicht.

### IP-Adressierung

Als Net ID eines POWERLINK Netzwerkes sollte die Class C IPv4 Adresse 192.168.100.0 verwendet werden. Jedes Netzwerk unterstützt 254 IP-Adressen. Das letzte Byte der IP-Adresse (Host ID) sollte dabei mit der Knotennummer (Node ID) des Teilnehmers übereinstimmen.



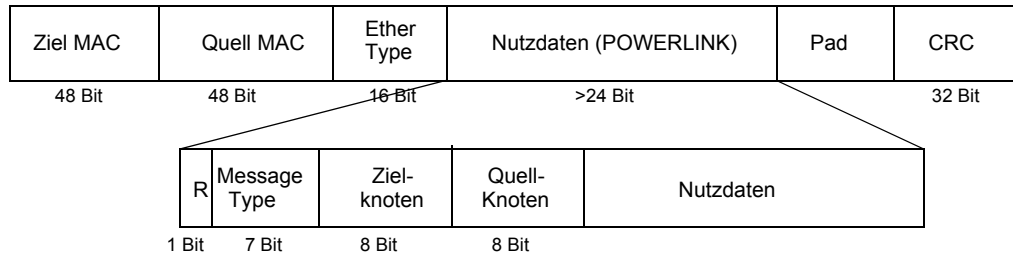
Die folgenden Node-IDs sind reserviert:

Adresse	Beschreibung
0x00	Ungültige Adresse
0xF0	Default Adresse des POWERLINK Managed Node
0xFB	Pseudo-Adresse zur Selbstadressierung eines Knotens
0xFC	POWERLINK Dummy-Adresse
0xFD	Default-Adresse des POWERLINK Diagnosegerätes
0xFE	Default-Adresse des POWERLINK Gateways/ Routers
0xFF	POWERLINK Broadcast-Adresse

Da es sich um ein Class C Netzwerk handelt, ist die Subnetzmaske der POWERLINK Knoten 255.255.255.0.

**Ethernet-Frame**

Das Ethernet-Telegramm besteht aus einem Header und den Nutzdaten. Der Header besteht aus der Ziel- und Quell-MAC-Adresse und dem Typfeld mit Steuerinformationen. Der Block für die Ethernet-Nutzdaten enthält das POWERLINK-Telegramm und nimmt eine Größe von mindestens 46 Byte und höchstens 1500 Byte ein. Abschließend wird die Richtigkeit des Frames anhand einer Prüfsumme sichergestellt.



**POWERLINK Frame**

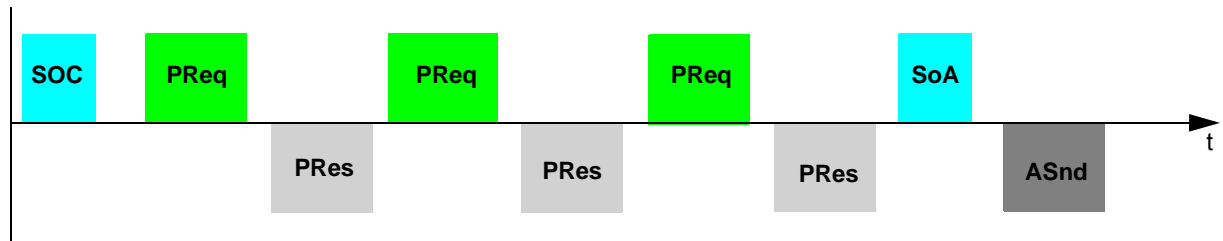
Das POWERLINK-Telegramm besteht ebenfalls aus einem Header und den eigentlichen Nutzdaten. Der Header besteht aus einem Reserved-Bit, dem Nachrichtentyp, dem Ziel- und dem Quellknoten. Folgende Nachrichtentypen sind definiert.

Nachrichtentyp	ID	Bezeichnung	Verwendung	Transfertyp
SoC	0x01	Start of Cycle	Definiert den Start eines neuen Zyklus	Multicast
PReq	0x03	Poll Request	Erfrage zyklische Daten des CN	Unicast
PRes	0x04	Poll Response	Sende aktuelle zyklische Daten des CN	Multicast
SoA	0x05	Start of Asynchronous	Signalisiere den Start der asynchronen Phase	Multicast
ASnd	0x06	Asynchro-nous Send	Senden von asynchronen Daten	Multicast

**Determinismus**

Die einzelnen Teilnehmer im Netzwerk, die Controlled Nodes (CN), werden durch einen speziellen Teilnehmer, dem Managed Node (MN), gesteuert und dürfen nur dann senden, wenn sie durch den MN aufgefordert wurden.

Der POWERLINK-Zyklus ist unterteilt in eine synchrone und eine asynchrone Phase. Zu Beginn der synchronen Phase sendet der MN das SoC Frame. Anschließend fragt er jeden Knoten einzeln mit einem PReq ab, welche daraufhin mit dem PRes antworten. Nach der zyklischen Phase beginnt der MN die asynchrone Phase mit dem Senden des SoA Telegramms. Mittels eines ASnd Frames kann ein vom MN bestimmter Knoten azyklische Daten übertragen.



### Geräteprofil

POWERLINK unterstützt die CANopen Geräteprofile. In diesen Profilen 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. Das Optionsmodul POWERLINK Controlled Node für den b maXX 2500 / 3300 / 5000-Regler ist nach dem Geräteprofil CiA<sup>®</sup> 402 (Drives and Motion Control) implementiert.



# POWERLINK AM B MAXX 2500 / 3300 / 5000

## 11.1 Allgemeines

---

Der b maXX 2500 / 3300 / 5000 POWERLINK Controlled Node verbindet den b maXX 2500 / 3300 / 5000 über den POWERLINK-Bus mit anderen POWERLINK-Knoten (z. B. PC, SPS, weitere b maXX Geräte, I/O-Module).

Informationen zur Installation und zum Umgang mit der Gerätereihe b maXX 3300 / 5000 finden Sie in der Dokumentation 5.11018 / 5.09021.

Informationen zur Programmierung des b maXX 3300 / 5000 Reglers finden Sie im Parameterhandbuch 5.12001 / 5.09022.

## 11.2 Adresseinstellung

---

Die Einstellung der IP-Adresse des b maXX 3300 / 5000 POWERLINK Controlled Node ist in der Betriebsanleitung b maXX 3300 / 5500 (Dokumentation 5.11018 / 5.09021) beschrieben.

## 11.3 XDD-Datei

---

Die XDD-Datei ist eine XML-Datei und dient der Beschreibung des Funktionsumfangs eines POWERLINK-Gerätes. Es ist ein elektronisches Datenblatt des POWERLINK-Gerätes. Die XDD-Datei wird vom POWERLINK-Managed Node bzw. Buskonfigurator genutzt. Die XDD-Datei beinhaltet Informationen über alle vom Controlled Node unterstützten Objekte, das Netzwerkmanagement und weitere Merkmale.

Der Namensweiterung der XDD-Datei ist \*.xdd.

Die Datei ist im Downloadbereich der Baumüller Homepage erhältlich.

### 11.4 Diagnose

Ethernet POWERLINK folgt dem Ethernet-Standard IEEE 802.3. Somit können für die Systemdiagnose Standardtools wie Wireshark (Freeware) oder OmniPeek und Standardgeräte wie Hubs bzw. Switches und PC-Netzwerkschnittstellen verwendet werden.

### 11.5 Datenaustausch und Parametrierung

Der Zugriff auf Daten oder Parameter erfolgt bei POWERLINK **immer** über CANopen Objekte. Entsprechend der Profilstruktur werden Objekte zur Kommunikationssteuerung (Indizes 0x1XXX) und anwendungs- oder gerätespezifische Objekte unterschieden. Die letzteren gliedern sich in Objekte nach Profil CiA<sup>®</sup> 402 (Indizes 0x6XXX) und herstellerspezifische Objekte (Indizes 0x2XXX bzw. 0x4XXX). Eine Auflistung der 6XXX'er und 0x2XXXer bzw. 0x4XXXer Objekte finden Sie in [▶Anhang B - Kurzreferenz◀](#) ab Seite 137.



#### HINWEIS!

#### Wichtig:

Die Adressierung der herstellerspezifischen Parameter ist in Anhang [▶B.1 2000er / 4000er Objektnummern \(herstellerspezifische Objekte\)◀](#) ab Seite 137 beschrieben.

### 11.6 Verzeichnis der Objekte zur Kommunikationssteuerung

In diesem Abschnitt finden Sie alle vom Baumüller POWERLINK Controlled Node unterstützten Objekte des kommunikationsspezifischen Bereiches des Objektverzeichnisses nach EPSG DS301.

Name	Index	Subindex	Datentyp	Defaultwert
NMT_DeviceType_U32	0x1000	0x00	UINT32	0x00020192

Dieses Objekt kann nur gelesen werden und enthält die Information, um welches Gerät es sich handelt (Drive nach CiA<sup>®</sup> 402), nur lesbar.

Name	Index	Subindex	Datentyp	Defaultwert
NMT_CycleLen_U32	0x1006	0x00	UINT32	1000

Falls das Sync-Telegramm aktiviert ist, muss das Sync-Interval auf die Zeit des Sync-Telegramms (500 µs, 1000 µs, 2000 µs, 4000 ms oder 8000 µs) eingestellt werden. Die eingestellte Zeit wirkt sich auf die Parameter 131.18 (FeldbusZykluszeit) und 156.4 (Sync-Offset) des b maXX Reglers aus.

Name	Index	Subindex	Datentyp	Defaultwert
NMT_ManufactDevName_VS	0x1008	0x00	VAR	-

Dieses Objekt ist nur lesbar. Es enthält die Zeichenfolge: "b maXX 5000".

Name	Index	Subindex	Datentyp	Defaultwert
NMT_ManufactHwVers_VS	0x1009	0x00	VAR	-

Dieses Objekt ist nur lesbar. Es enthält die aktuelle Hardware-Version des Optionsmoduls, z. B. die Zeichenfolge: "01.00".

Name	Index	Subindex	Datentyp	Defaultwert
NMT_ManufactSwVers_VS	0x100A	0x00	VAR	-

Dieses Objekt ist nur lesbar. Es enthält die aktuelle POWERLINK Stack Version des Optionsmoduls, z. B. die Zeichenfolge: "EPL V2 V1.8 r1".

Name	Index	Subindex	Datentyp	Defaultwert
NMT_IdentityObject_REC	0x1018	0x00	UINT8	0x04
VendorId_U32		0x01	UINT32	0x0000015A
ProductCode_U32		0x02	UINT32	0x00010000
RevisionNo_U32		0x03	UINT32	0x00010000
SerialNo_U32		0x04	UINT32	0x00000000

In diesem Objekt sind einige Informationen über das Gerät enthalten.

RevisionNo\_U32 enthält den aktuellen Stand der Firmware z. B. 00010002 für FW Version 01.02.

Name	Index	Subindex	Datentyp	Defaultwert
CFM_VerifyConfiguration_REC	0x1020	0x00	UINT8	0x02
ConfDate_U32		0x01	UINT32	0x00000000
ConfTime_U32		0x02	UINT32	0x00000000

In diesem Objekt sind die Informationen zur lokalen Konfigurationszeit des Gerätes enthalten. ConfDate\_U32 enthält die Konfigurationszeit. Diese beinhaltet die Tage seit 01.01.1984.

Name	Index	Subindex	Datentyp	Defaultwert
NMT_InterfaceGroup_Xh_REC	0x1030	0x00	UINT8	0x09
InterfaceIndex_U16		0x01	UINT16	0x0001
InterfaceDescription_VSTR		0x02	VISIBLE_STRING	"Interface 1"
InterfaceType_U8		0x03	UINT8	0x06
InterfaceMtu_U16		0x04	UINT16	0x1500
InterfacePhysAddress_OSTR		0x05	OCTET_STRING6	0x06
InterfaceName_VSTR		0x06	VISIBLE_STRING	"Interface 1"
InterfaceOperStatus_U8		0x07	UINT8	0x01
InterfaceAdminState_U8		0x08	UINT8	0x01
Valid_BOOL		0x09	BOOL	0x1
HubPortEnableMask_U64		0x0A	UINT64	0x00000000 00000000

Anhand dieses Objektes werden Parameter des Netzwerk Interface über SDO konfiguriert.

Name	Index	Subindex	Datentyp	Defaultwert
SDO_SequLayerTimeout_U32	0x1300	0x00	VAR	-

In diesem Objekt ist der Timeout-Wert in [ms] für die Erkennung eines SDO Abbruchs enthalten.

Name	Index	Subindex	Datentyp	Defaultwert
PDO_RxCommParam_00h_REC	0x1400	0x00	UINT8	0x02
Node_ID_U8		0x01	UINT8	0x00
MappingVersion_U8		0x02	UINT8	0x00

Dieses Objekt enthält Informationen zur Receive-PDO.

Name	Index	Subindex	Datentyp	Defaultwert
PDO_RxMappParam_00h_REC	0x1600	0x00	UINT8	0x01
		0x01	UINT64	0x00100000 0006040
		:	:	
		n	UINT64	

Dieses Objekt enthält den Inhalt der Receive-PDO. In Subindex 0x00 steht die Gesamtzahl der nachfolgenden Einträge. Die Gesamtzahl der gemappten Objekte darf den Regler Sollwertrahmen von max. 16 Objekten nicht überschreiten. (siehe auch [PDO-Mapping](#) ab Seite 125).

Name	Index	Subindex	Datentyp	Defaultwert
PDO_TxCommParam_00h_REC	0x1800	0x00	UINT8	0x02
Node_ID_U8		0x01	UINT8	0x00
MappingVersion_U8		0x02	UINT8	0x00

Dieses Objekt enthält Informationen zur Transmit-PDO.

Name	Index	Subindex	Datentyp	Defaultwert
PDO_TxMappParam_00h_REC	0x1A00	0x00	UINT8	0x01
		0x01	UINT64	0x001000000 0006041
		:	:	
		n	UINT64	

Dieses Objekt enthält den Inhalt der Transmit-PDO. In Subindex 0x00 steht die Gesamtzahl der nachfolgenden Einträge. Die Gesamtzahl der gemappten Objekte darf den Regler Istwertrahmen von max. 16 Objekten nicht überschreiten. (siehe auch [PDO-Mapping](#) ab Seite 125).

## 11.7 Netzwerkmanagement (NMT)

Kommandos des Netzwerkmanagements dienen vorwiegend der Steuerung der Kommunikationszustände im POWERLINK Netz

Hier ist das Zustandsdiagramm der Kommunikation des POWERLINK Controlled Node dargestellt.

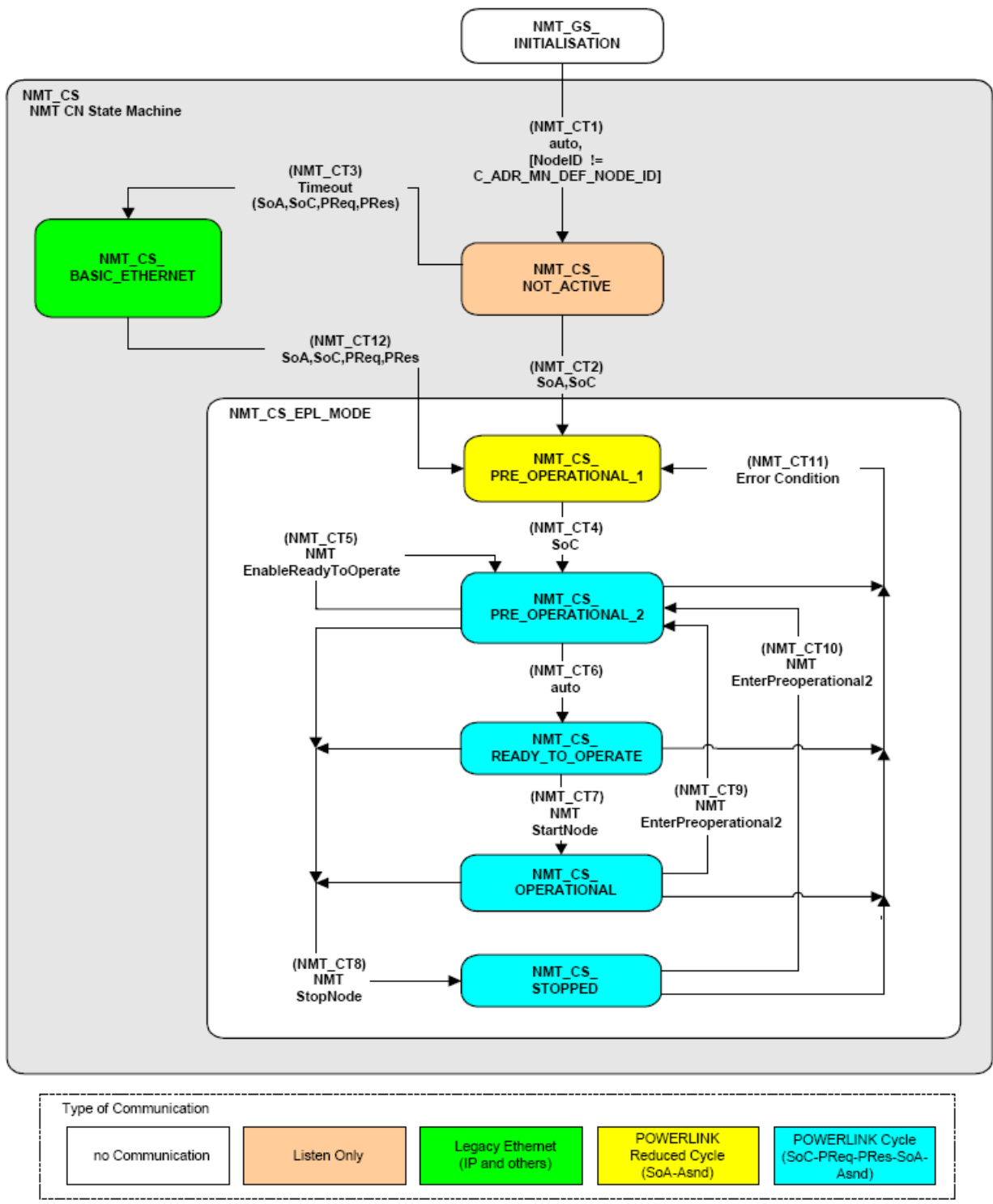


Abbildung 23: Zustandsmaschine der Kommunikation

Zustand	Beschreibung
NMT_CS_NOT_ACIVE	Übergangszustand zum Starten des CN und zur Identifizierung im Netzwerk
NMT_CS_PRE_OPERATIONAL_1	Identifizierung des Knotens vom MN über IdentRequest
NMT_CS_PRE_OPERATIONAL_2	PDO Konfiguration durch den MN; Konfigurationsüberprüfung
NMT_CS_READY_TO_OPERATE	Signalisierung der Betriebsbereitschaft zum MN; Starten der PDO Kommunikation
NMT_CS_OPERATIONAL	Normaler Betriebszustand des CN; CN nimmt am zyklischen Datenverkehr teil
NMT_CS_STOPPED	CN ist passiv; Kontrolliertes Herunterfahren des CN; CN nimmt nicht an zyklischen Datenverkehr teil
NMT_CS_BASIC_ETHERNET	CN nimmt an herkömmlicher Ethernet Kommunikation nach IEEE 802.3 teil

## 11.8 Bedarfsdaten (SDO)

Service Daten Objekte (SDO) dienen dem Austausch von Nachrichten ohne Echtzeitanforderungen. Die SDOs werden zur Parametrierung der CNs 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 123.

SDOs folgen dem Client-Server-Modell. Der Client (MN) initiiert die Kommunikation und der Server (CN) antwortet darauf. Ein Server kann eine SDO Kommunikation nicht beginnen. Das Baumüller POWERLINK Modul unterstützt eine Server SDO und keine Client SDO.

Der asynchrone Zyklus wird durch Senden des Start of Asynchronous (SoA) Telegramms durch den MN gestartet. Der SDO Transfer wird über ASnd durch den CN beantwortet.

## 11.8.1 Telegrammaufbau SoA

Das Start of Asynchronous (SoA) Telegramm hat folgenden Aufbau:

Byte Offset	Bit Offset							
	7	6	5	4	3	2	1	0
0	Message Type = SoA (0x05)							
1	Destination = Broadcast Adresse (0xFF)							
2	Source = Node ID des MN (0xF0)							
3	NMTStatus							
4	res	res	res	res	res	EA/res	ER/res	res
5	reserviert							
6	RequestedServiceID (siehe nachfolgende Tabelle)							
7	RequestedServiceTarget							
8	EPLVersion							
9 ... 45	reserviert							

EA (Exception Acknowledge)

ER (Exception Reset)

RequestedService	ID	Beschreibung
NoService	0x00	Asynchroner Slot wird durch keinen Knoten bedient
IdentRequest	0x01	Identifizierung inaktiver Knoten
StatusRequest	0x02	Abfrage von Status und Fehlerinformationen einzelner Knoten
NMTRequestInvite	0x03	Aufforderung eines Knotens, der ein NMT Kommando gestellt hat, dieses zu senden
Manufacturer specific	0xA0 .. 0xFE	Herstellerspezifischer Service
UnspecificInvite	0xFF	Aufforderung eines Knotens, der eine Sendeanfrage gestellt hat, diese zu senden



### 11.8.2 Telegrammaufbau ASnd

Das Asynchronous Send (ASnd) Telegramm hat folgenden Aufbau:

Byte Offset	Bit Offset							
	7	6	5	4	3	2	1	0
0	Message Type = ASnd (0x06)							
1	Destination (Node ID des adressierten Knoten)							
2	Source (Node ID des sendenden Knoten)							
3	Service ID (siehe nachfolgende Tabelle)							
4 .. 7	Sequence Layer Protocol							
8 .. k-1	Command Layer Protocol							
k .. 1472	SDO Payload Data							

Service ID	ID	Beschreibung
IdentResponse	0x01	Antwort eines Knotens auf ein IdentRequest über SoA
StatusResponse	0x02	Antwort eines Knotens auf ein StatusRequest über SoA
NMTRquest	0x03	Antwort eines CN auf ein NMTRquestInvite über SoA
NMTCommand	0x04	Antwort des MN auf eine interne Aufforderung oder eine externe Aufforderung über NMTRquest
SDO	0x05	Antwort eines CN auf ein UnspecificInvite über SoA
Manufacturer specific	0xA0 .. 0xFE	Herstellerspezifischer Service

### 11.8.3 Fehlerreaktionen

Fehlerhafte SDO Zugriffe werden mit Abort Codes abgewiesen. Der Aufbau dieser Abort Telegramme ist identisch zu dem in [▶ Telegrammaufbau SoA](#) auf Seite 120 dargestellten SDO Telegramm.

Das Datenfeld enthält einen Abort Code mit 4 Bytes.

Bei fehlerhaften SDO Zugriffen werden folgende Meldungen unterschieden:

Abort Code	Beschreibung
0x05040000	Zeitüberschreitung des SDO Protokolls
0x05040001	Unbekannte Kommando ID
0x05040002	Ungültige Blockgröße
0x05040003	Ungültige Sequenznummer
0x05040005	Speicherplatz nicht ausreichend
0x06010000	Nicht Unterstützter Zugriff auf Objekt
0x06010001	Lesen eines lesegeschützten Objektes
0x06010002	Schreiben auf schreibgeschütztes Objektes
0x06020000	Objekt existiert nicht
0x06040041	Objekt kann nicht als PDO gemappt werden
0x06040042	Die Anzahl und Länge der gemappten Objekte überschreitet die PDO Länge
0x06040043	Allgemeine Parameterinkompatibilität
0x06040044	Ungültige Heartbeat-Festlegung
0x06040047	Allgemeine Geräteinkompatibilität
0x06060000	Zugriffsverletzung auf Hardware
0x06070010	Inkorrekte Datenwertlänge
0x06070012	Inkorrekter Datenwert, Datenlänge zu groß
0x06070013	Inkorrekter Datenwert, Datenlänge zu kurz
0x06090011	Subindex existiert nicht
0x06090030	Wertebereich überschritten (bei Schreibzugriffen)
0x06090031	Wert zu groß (bei Schreibzugriffen)
0x06090032	Wert zu klein (bei Schreibzugriffen)
0x06090036	Maximaler Wert kleiner als minimaler Wert
0x08000000	Allgemeiner Fehler
0x08000020	Daten können nicht zur Applikation übertragen oder abgespeichert werden
0x08000021	Daten im Moment nicht verfügbar
0x08000022	Daten können aufgrund des derzeitigen Kommunikationszustandes nicht gemappt werden
0x08000023	Objektverzeichnis ist nicht verfügbar
0x08000024	Konfigurationsdatensatz beinhaltet keine Daten

## 11.9 Synchronisation (SYNC)

Zur Synchronisation der CNs wird das Start of Cycle (SoC) Telegramm verwendet. Dieses Telegramm ist unbestätigt (Multicast) und wird vom MN gesendet. Es enthält keine Daten. Der POWERLINK Knoten kann SoC-Telegramme empfangen.

Der Empfang eines SoC Telegramms erzeugt auf dem POWERLINK Knoten einen Interrupt, der an den b maXX<sup>®</sup> Regler weitergeleitet wird. Damit kann dieses Signal zur Synchronisation des b maXX<sup>®</sup> Reglers verwendet werden. Innerhalb eines Kommunikationszyklus müssen alle entsprechenden Telegramme an alle projektierten Slaves gesendet werden. Die Anzahl der Knoten sowie die Verarbeitungszeiten sind dabei zu berücksichtigen. Die Einstellung der Zykluszeit für das SoC Telegramm wird in Objekt 0x1006 vorgenommen. Siehe hierzu [▶Verzeichnis der Objekte zur Kommunikationssteuerung◀](#) ab Seite 114. Weiterhin muss die Kommunikationszykluszeit im Datensatz des Reglers abgespeichert werden.

Durch das sogenannte Multiplexing ist es im POWERLINK möglich einzelne Knoten nicht in jedem Zyklus abzufragen. Es können sich somit mehrere Knoten einen Zeitabschnitt in der Übertragungsphase teilen. Damit lässt sich die Bandbreite der synchronen Phase auf dem POWERLINK Bus optimal ausnutzen. Die Konfiguration dieser Zuteilung erfolgt durch den Managed Node.

### 11.9.1 Telegrammaufbau SoC

Das Start of Cycle (SoC) Telegramm hat folgenden Aufbau:

Byte Offset	Bit Offset							
	7	6	5	4	3	2	1	0
0	Message Type = SoC (0x01)							
1	Destination = Broadcast Adresse (0xFF)							
2	Source = Node ID des MN (0xF0)							
3	reserviert							
4	MC	PS	res	res	res	res	res	res
5	reserviert							
6 ... 13	NetTime (optional)							
14 ... 21	RelativeTime (optional)							
22 ... 45	reserviert							

MC (Multiplexed Cycle Completed)

PS (Prescaled Slot)

## 11.10 Prozessdaten (PDO)

Prozess Daten Objekte sind unbestätigte Telegramme, die auf den Austausch von Daten mit Echtzeitanforderungen optimiert sind. Es existieren zwei Arten von PDOs, die aufgrund der Richtung der Datenübertragung aus Gerätesicht unterschieden werden. Der POWERLINK Controlled Node für b maXX<sup>®</sup> Regler unterstützt sowohl eine Transmit PDO (TPDO) als auch eine Receive PDO (RPDO). In jeder PDO können bis zu 16 Objekte übertragen werden.

Die PDO Kommunikation in POWERLINK erfolgt anhand synchroner PReq bzw. PRes Telegramme. In der synchronen Phase sendet der MN das PollRequest (PReq) als Unicast-Telegramm. Der jeweilige CN sendet das PollResponse (PRes) als Broadcast.

Das Format des Datenaustauschs muss vor Beginn der Kommunikation zwischen Sender und Empfänger definiert sein (Mapping). Das Senden und Empfangen von PDOs kann auf unterschiedliche Weise ausgelöst werden.



### HINWEIS!

Alle in den PDOs konfigurierten Objekte werden zwischen dem POWERLINK Controlled Node und dem b maXX<sup>®</sup> Regler als zyklische Daten übertragen (siehe auch [► Kommunikationsablauf ◀](#) ab Seite 25). Da die zyklische Datenübertragung nur im Zustand NMT\_CS\_OPERATIONAL stattfindet, darf auch nur in diesem Zustand die Kommunikationsüberwachung in ProDrive aktiviert werden.

### 11.10.1 Telegrammaufbau PReq und PRes

Das PollRequest (PReq) Telegramm hat folgenden Aufbau:

Byte Offset	Bit Offset							
	7	6	5	4	3	2	1	0
0	Message Type = PReq (0x03)							
1	Destination = Node ID des CN							
2	Source = Node ID des MN (0xF0)							
3	reserviert							
4	res	res	MS	res	res	EA	res	RD
5	reserviert							
6	PDOVersion							
7	reserviert							
8 ... 9	Size (Größe der Nutzdaten in Byte)							
10 ... n	Payload							

MS (Multiplexed Slot)  
 EA (Exception Acknowledge)  
 RD (Ready)

Das PollResponse (PRes) Telegramm hat folgenden Aufbau:

Byte Offset	Bit Offset							
	7	6	5	4	3	2	1	0
0	Message Type = PRes (0x04)							
1	Destination = Broadcast Adresse (0xFF)							
2	Source = Node ID des CN							
3	NMT Status							
4	res	res	MS	EN	res	res	res	RD
5	res	res	PR			RS		
6	PDOVersion							
7	reserviert							
8 ... 9	Size (Größe der Nutzdaten in Byte)							
10 ... n	Payload							

MS (Multiplexed Slot)  
 EN (Exception New)  
 RD (Ready)  
 PR (Priority)  
 RS (RequestToSend)

### 11.10.2 PDO-Mapping

Mapping ist ein Verfahren zur Zuordnung von Variablen / Objekten an PDOs. Mit den PDOs werden diese Variablen / Objekte über POWERLINK transportiert. Durch das Mapping wird der zyklische Datenaustausch projiziert. Für diese Parametrierung werden SDOs genutzt. Das Mapping wird über im Objektverzeichnis adressierbare Objekte eingestellt.

Für die TPDO und die RPDO existiert jeweils zwei solche Objekte (siehe auch [►Verzeichnis der Objekte zur Kommunikationssteuerung◄](#) ab Seite 114). Eines der Objekte bestimmt den Inhalt der PDO, das zweite die Kommunikationsbeziehung bzw. Triggung.

Prozess-Daten-Objekt	Objekt für den Inhalt	Objekt für die Kommunikationsbeziehung
TPDO	0x1A00	0x1800
RPDO	0x1600	0x1400



**HINWEIS!**

Das Mapping kann im Zustand NMT\_CS\_OPERATIONAL **nicht** geändert werden. Erst beim Übergang nach NMT\_CS\_READY\_TO\_OPERATE wird ein neues Mapping aktiviert.

Zur Nutzdatenübertragung stellt das PReq bzw. PRes Datentelegramm maximal 1490 Byte zur Verfügung. Der POWERLINK Controlled Node kann hierbei den Inhalt von bis zu 16 Variablen / Objekten in jede Richtung übertragen. Durch das Mapping wird der logische Inhalt der Nutzdaten festgelegt.

Für diese Festlegung werden bestimmte Angaben über die zu mappende Objekte benötigt:

Objektindex, Subindex und die Länge des Datums und die Reihenfolge der zu mappenden Objekte. Aus dem Objektverzeichnis werden die zu mappenden Objekte an die Subindizes beginnend mit 0x01 des Mapping-Objekt (0x1600, 0x1A00) geschrieben, z. B. wird auf Objekt 0x1600 Subindex 0x01 der Wert 0x0010.0000.0000.6040 eingetragen. Dies bedeutet, die ersten beiden Bytes (Länge 0x0010, Offset 0x0000) der in RXPD empfangenen Daten werden auf das Steuerwort (Objekt 0x6040, Subindex 0x00) geschrieben. Das Objekt 0x6040 ist auf den b maXX<sup>®</sup> Parameter **108.1** Steuerwort umgesetzt (siehe auch [►Anhang C - Umsetzungstabellen◄](#) ab Seite 143). Damit wird das erste Wort des in RPDO empfangenen Telegramms auf das Steuerwort des b maXX<sup>®</sup> geschrieben. In Subindex 0x00 muss die Anzahl der zu mappenden Objekte (Anzahl der mit gültigen Objekten belegten Subindizes) eingetragen werden.

Die Subindizes für die Mapping Objekte 0x1600 und 0x1A00 haben folgenden Aufbau:

Byte Offset	Name	Beschreibung
0 ... 1	Index	Index des zu mappenden Objekts
2	Subindex	Subindex des zu mappenden Objekts
3	reserviert	
4 ... 5	Offset	Versatz zum Start der PDO Nutzdaten in Bit
6 ... 7	Länge	Länge des zu mappenden Objekts in Bit



**HINWEIS!**

Bei der Einstellung des Mappings in den Mapping Parametern (0x1600, 0x1A00) ist jeweils der Subindex 0x00 mit der richtigen Anzahl der gemappten Objekte zu beschreiben.

**Sollwerte:**

Die zulässigen zyklische Sollwerte sind in einer Tabelle mit der Spalte "PDO-Mapping" als "RX" gekennzeichnet. Die Tabelle ist im Anhang B.2 (für die 6000'er Objektnummern) zu finden.

**Istwerte:**

Die zulässigen zyklische Istwerte sind in einer Tabelle mit der Spalte "PDO-Mapping" als "TX" gekennzeichnet. Die Tabelle ist im Anhang B.2 (für die 6000'er Objektnummern) zu finden. Eine detaillierte Beschreibung der b maXX<sup>®</sup>-Parameter finden Sie im Parameterhandbuch b maXX<sup>®</sup> 3300 (5.12001) bzw. im Parameterhandbuch b maXX<sup>®</sup> 5000 (5.09022).

Fehlerhafte Mapping-Konfigurationen (unzulässige Objekte in 0x1600, 0x1A00) werden durch Abort Codes über SDO gemeldet.

Die zyklischen Soll-/Istwerte werden lückenlos in der Feldbus Prozessdatenliste des Reglers initialisiert, d. h. der erste Sollwert der RPDO steht an erster Stelle in der Prozessdatenliste, der zweite Sollwert an zweiter Stelle usw. Analog gilt für die Istwert-Initialisierung der erste Istwert der TPDO steht an erster Stelle in der Prozessdatenliste, der zweite Istwert von TPDO an zweiter Stelle usw.

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

In der Regel überschreiben PD Schreibzugriffe zyklisch SD Schreibzugriffe auf das gleiche FBO. In einzelnen Fällen kann es vorkommen, dass ein Schreibzugriff über SD erfolgreich gewesen ist. Dies ist aber nicht zuverlässig.

**HINWEIS!**

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

### 11.11 Konfigurationsbeispiel mit einer B&R X20 SPS

Im folgenden Abschnitt wird die Konfiguration des POWERLINK Controlled Node für b maXX 2500 / 3300 / 5000 Regler an einer Steuerung X20 von B&R mittels Automation Studio (V4.0.16.81) beschrieben.

Zum Einbinden des POWERLINK Controlled Nodes muss die entsprechende XDD-Datei in das Automation Studio Projekt importiert werden. Dazu muss im Menü unter:

*Extras → Feldbus Gerät importieren...*

die Gerätebeschreibungsdatei ausgewählt werden. Die Datei ist für die unterschiedlichen Gerätereihen im Downloadbereich der Baumüller Homepage erhältlich.

Da die Gerätebeschreibung in der Automation Studio Projektdatei gespeichert ist, muss dieser Vorgang bei der Erstellung eines neuen Projektes wiederholt werden.



Abbildung 24: Konfiguration - Feldbus Gerät importieren

Das importierte Gerät wird nun unter dem Netzwerktyp *POWERLINK* im *Hardware Katalog der Toolbox* angezeigt.



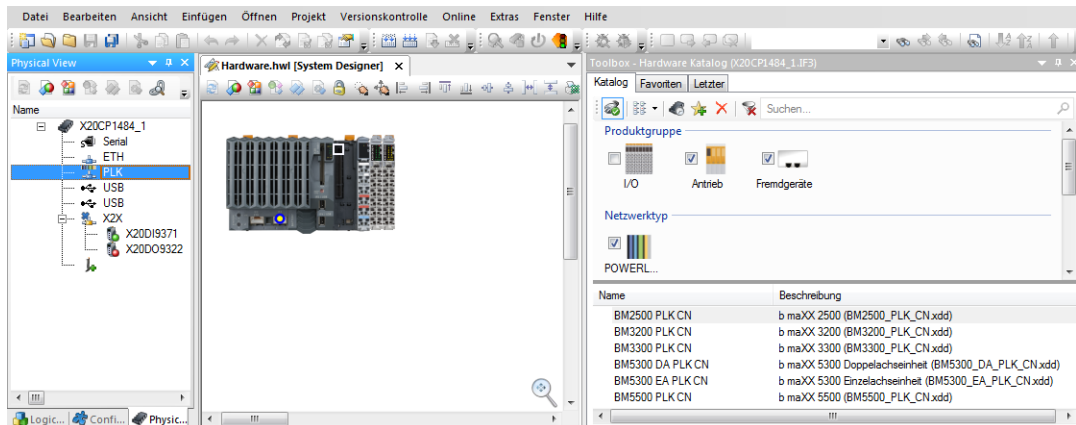


Abbildung 25: Konfiguration - Hardware Katalog

Der entsprechende Controlled Node wird im *Hardware Katalog* mit folgenden Bezeichnungen angezeigt:

BM2500_PLK_CN	b maXX 25xx
BM3300_PLK_CN	b maXX 32xx / 33xx
BM5000_NWR_PLK_CN	b maXX 51xx Einspeiseeinheit Netzwechselrichter
BM5000_EA_PLK_CN	b maXX 52xx / 53xx Einzelachseinheit
BM5000_DA_PLK_CN	b maXX 52xx / 53xx Doppelachseinheit
BM5000_MA_PLK_CN	b maXX 54xx / 55xx Monoeinheit

Das Gerät kann mittels Drag und Drop oder durch Doppelklick zum *System Designer* hinzugefügt und verbunden werden.

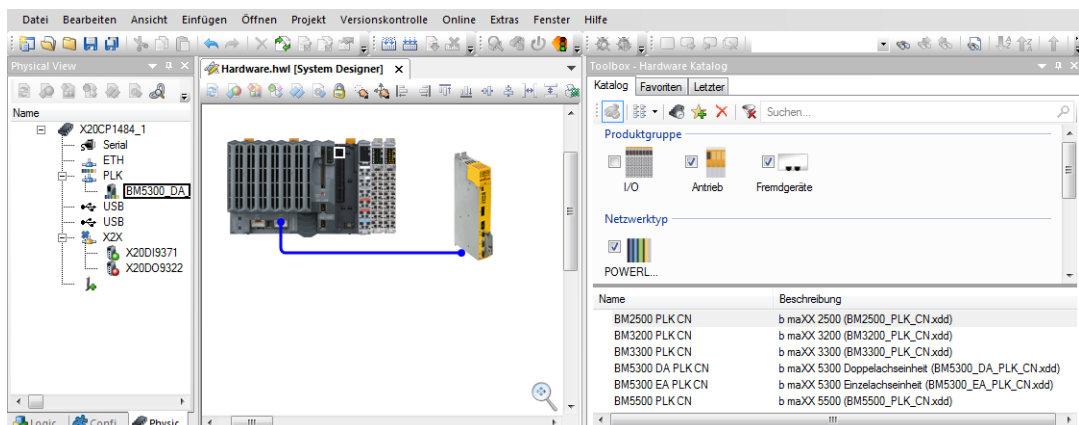


Abbildung 26: Konfiguration - System Designer

Nach erfolgreichem Importieren erscheint der Controlled Node in der Automation Studio *Physical View*.

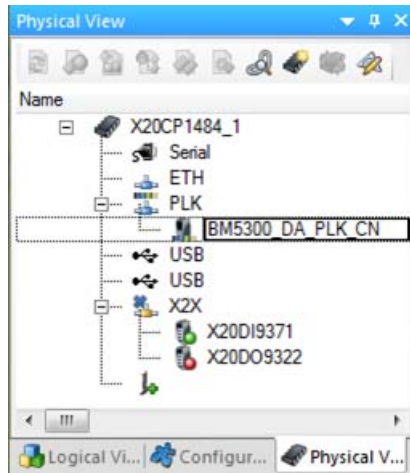


Abbildung 27: Konfiguration - Controlled Node in Physical View

### 11.12 PollResponse Chaining (ab Version 1.10)

PollResponse Chaining wird verwendet, um die Telegrammanzahl im POWERLINK Netzwerk zu reduzieren und somit die Performance zu erhöhen. Anstelle der üblichen PollRequest/PollResponse Telegrammsequenz werden die einzelnen PollRequest Telegramme an die Controlled Nodes (CN) in einem PollResponse Telegramm des Managing Node (PResMN) zusammengefasst. Dieses wird über Broadcast an alle Knoten versendet. Das Verfahren verringert vor allem die Durchlaufzeiten.



Abbildung 28: Ohne PollResponse Chaining:



Abbildung 29: Mit PollResponse Chaining:

Es ist ebenfalls möglich konventionelle PollRequest/PollResponse Teilnehmer mit PollResponse Chaining Teilnehmern in einem Netzwerk zu kombinieren.

### 11.13 Ethernet Kommunikation (ab Version 1.10)

Mit der Bediensoftware ProDrive kann der POWERLINK Controlled Node für b maXX 2500 / 3300 / 5000 konfiguriert werden. Der Zugriff mit ProDrive kann über Ethernet erfolgen. Dabei kann der Controlled Node (CN) direkt vom PC aus oder indirekt über einen Ethernet Gateway angesprochen werden.

#### 11.13.1 Direkte Ethernet Kommunikation

Bei einer direkten Verbindung mit einem PC muss sich die IP Adresse des PCs im gleichen Subnetz befinden wie die POWERLINK Controlled Nodes. Bei Linientopologie kann die Verbindung zum PC sowohl am Ausgangsport des letzten Slaves als auch am Eingangsport des ersten Slaves erfolgen.

##### 11.13.1.1 Konfiguration der IP Einstellungen mit ProDrive

Mit ProDrive kann man die IP Einstellungen der POWERLINK Controlled Node für b maXX 2500 / 3300 / 5000 auslesen und konfigurieren.

Im Menü unter *Werkzeuge* → *Ethernet Konfiguration* ist es möglich, nach vorhandenen POWERLINK Controlled Nodes für b maXX 2500 / 3300 / 5000 zu suchen.

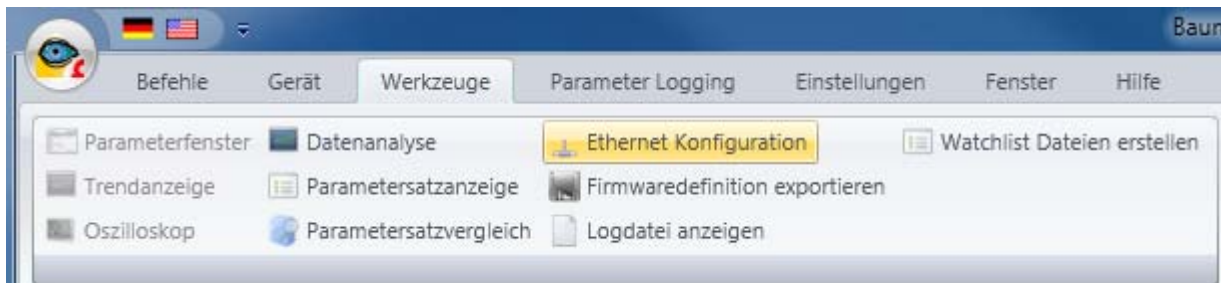


Abbildung 30: Ethernet Konfiguration

Die gefunden Geräte können hinsichtlich der Einstellungen für IP-Adresse, Gateway-Adresse und Subnetzmaske geändert werden.



Abbildung 31: Gefundene Geräte



Abbildung 32: Gerät konfigurieren

### 11.13.2 Kommunikation mit ProDrive

Mit der Bediensoftware ProDrive kann mittels einer direkten TCP/IP Verbindung der POWERLINK Controlled Node für b maXX 2500 / 3300 / 5000 konfiguriert werden. Der Zugriff der Ethernet-Kommunikation über TCP/IP erfolgt über Port 20547.

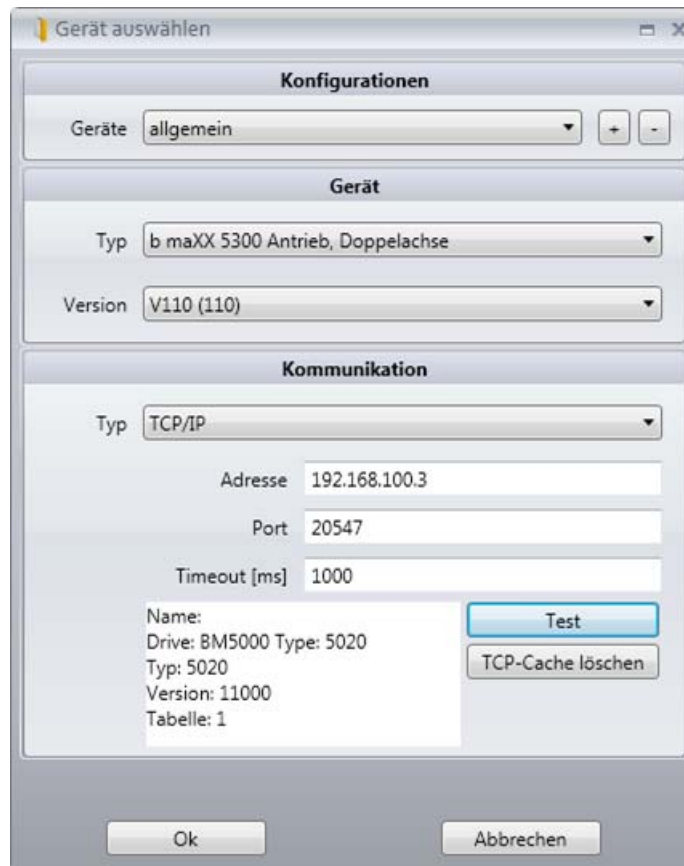


Abbildung 33: Gerät auswählen

### 11.13.3 Indirekte Ethernet Kommunikation

Der Ethernet Zugriff auf den POWERLINK Controlled Node kann auch über einen Ethernet Gateway erfolgen. Dieser Gateway kann zum Beispiel ein Managing Node (MN) sein, falls diese Funktion unterstützt wird. Dabei werden die Ethernet-Telegramme in den vorgesehenen asynchronen Zeitintervallen getunnelt. Die Größe des Nettodatenbereichs der asynchronen POWERLINK Telegramme (Maximum Transmission Unit - MTU) sollte dabei 988 Bytes betragen. Dies entspricht der Größe der ProDrive Telegrammsequenzen.

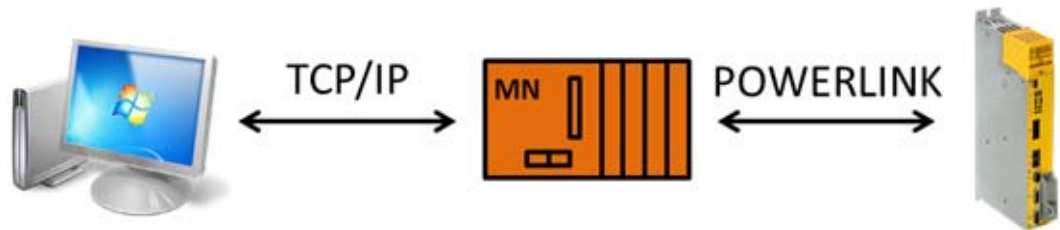
Für eine indirekte Ethernet Kommunikation müssen die korrekten IP Einstellungen des POWERLINK Controlled Nodes und des Gateways beachtet werden. Für das Routing muss die Ethernet Adresse des Gateway in die Routing Tabelle des PCs zum Beispiel über das Kommandofenster eingetragen werden.

Zum Beispiel:

	<code>ROUTE ADD [host] MASK [subnet] [gateway]</code>
host	Hostadresse des POWERLINK Netzwerkes z. B. 192.168.100.0
subnet	Subnetzmaske für Routingeintrag z. B. 255.255.255.0
gateway	Adresse des Gateways z. B. 192.168.1.2

## 11.14 Automatische Knotennummernvergabe (Dynamic Node Allocation) (ab Version 1.10)

Beispiel eines MN als Gateway:



IP Adresse: 192.168.1.1	IP Adresse TCP/IP: 192.168.1.2	IP Adresse: 192.168.100.1
Netzmaske: 255.255.255.0	Netzmaske TCP/IP: 255.255.255.0	Netzmaske: 255.255.255.0
Route: ROUTE ADD 192.168.100.0 MASK 255.255.255.0 192.168.1.2	POWERLINK NAT Sub- netz: 192.168.100.0	Gateway: 192.168.100.240
	Netzmaske POWERLINK: 255.255.255.0	
	MTU Größe: 988	

Die Adresse des MN im POWERLINK Netzwerk ist 192.168.100.240.

## 11.14 Automatische Knotennummernvergabe (Dynamic Node Allocation) (ab Version 1.10)

Falls Dynamic Node Allocation (DNA) vom MN unterstützt wird, können die Knotennummern der POWERLINK Controlled Nodes für b maXX 2500 / 3300 / 5000 vom MN automatisch vergeben werden.

Die Knoten aller Controlled Nodes, die ihre Nummer durch DNA erhalten, müssen auf die Knotennummer 0 gesetzt werden. Die Knotennummer entspricht der letzten Ziffer der IP Adresse. Die Einstellung der IP Adresse ist in der Betriebsanleitung b maXX 3300 / 5500 (Dokumentation 5.11018 / 5.09021) beschrieben.

Die korrekte Verkabelung von Ein- und Ausgangsport während DNA ist zu beachten, da die Ausgangsports (Port 2) der CNS während der Vergabe zeitweise deaktiviert werden.

DNA wird nur komplett bei Linientopologie unterstützt. Die Knotennummern werden dabei der Reihe nach inkrementiert. Für andere Topologien müssen die Anfangsknoten einer Linie als Kopfstation eine feste Knotennummer zugewiesen bekommen.

Bei der Verwendung von DNA verzögert sich der Einschaltvorgang mit der Anzahl der Knoten, die ihre Nummer über den MN erhalten.



## ANHANG A - ABKÜRZUNGEN

CD	Collision Detection
CN	Controlled Node
CoE	CAN application protocol over EtherCAT
CSMA	Carrier Sense Multiple Access
DC	Distributed Clocks
DSP	Draft Standard Proposal
EMCY	Emergency Telegramm
FB	Funktionsbaustein
FBO	Feldbusobjekt
FMMU	Fieldbus Memory Management Unit
ID	Ident-Nummer
MAC	Media Access Control
MN	Managed Node
NMT	Netzwerk-Management
PC	Personal Computer
PDO	Prozess Daten Objekt
SDO	Service Daten Objekt
SIX	Subindex
SPS	Speicherprogrammierbare Steuerung
SYNC	Synchronisation
XDD	XML Device Description







## ANHANG B - KURZREFERENZ

Die folgende Kurzreferenz zeigt den Zusammenhang zwischen CANopen-Objektnummer und dem b maXX<sup>®</sup> Regler-Parameternummern (siehe Parameterhandbuch b maXX 5000 (5.09022) bzw. Parameterhandbuch b maXX 3300 (5.12001)).

### B.1 2000er / 4000er Objektnummern (herstellerspezifische Objekte)

Für die Zuordnung der herstellerspezifischen Objekte ergibt sich folgende Regel:

Objekt-ID =  $0x2000 + \text{AchseOffset} + \text{FbNummer} + (0x200 * \text{Instanz})$

mit: AchseOffset Achse 1:  $0x0000$

AchseOffset Achse 2:  $0x2000$

Objekt Subindex = Parameternummer

CAN Index: Bit's	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	x	B1	B2	Instanz			Res	FB-Funktionsblock								
		0	1	Achse 1												
		1	0	Achse 2												

Für Array- und Struct-Parameter sind spezielle Objekte im reservierten Bereich festgelegt.

#### Beispiele: Fehleranzahl Achse 1 — Parameternummer 100.5.0.0

FbNummer =  $100 = 0x64$

Instanz =  $0$

AchseOffset =  $0$

Objekt-ID =  $0x2000 + 0 + 0x64 + 0 = 0x2064$

Subindex =  $5$

→ Objekt-ID =  $0x2064$ , Subindex  $5$

### Isq-Istwert Achse 2 — Parameternummer 47.3.0.0

FbNummer = 47 = 0x2F  
Instanz = 0  
Achsoffset = 0x2000  
Objekt-ID =  $0x2000 + 0x2000 + 0x2F + 0 = 0x402F$   
Subindex = 3  
→ Objekt-ID = 0x402F, Subindex 3

### Lageistwert Winkel 32 Bit Instanz 1 — Parameternummer 106.10.1.0

FbNummer = 106 = 0x6A  
Instanz = 1  
Achsoffset = 0  
Objekt-ID =  $0x2000 + 0 + 0x6A + (0x200 * 1) = 0x226A$   
Subindex = 10  
→ Objekt-ID = 0x226A, Subindex 10

Die Fehlerliste des Antriebs ist als eigenes Objekt angelegt.

Objekt-ID: 0x2120 für Achse 1 bzw. 0x4120 für Achse 2

Mit dem Subindex (von 1 gezählt) wird der jeweilige Eintrag in der Fehlerliste adressiert.

**B.2 6000er Objektnummern (Device Profile CiA® 402)**

Auf zahlreiche Parameter ist es möglich sowohl über 2000er / 4000er als auch über ein oder mehrere 6000er Objekte zuzugreifen.



**HINWEIS!**

Zwischen den 6000er und 2000er / 4000er Objekten können unterschiedliche Normierungen auftreten!

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

CANopen-Objekt-Nummer		Parameter Nr.	PDO-Mapping	Access - Typ	Betriebsart nach CiA® 402
Index	Subindex				
0x6007	0x00	-	TX / RX	rw	Common Entries
0x603F	0x00	<b>100.5</b>	TX / RX	rw	Common Entries
0x6040	0x00	<b>108.1</b>	TX / RX	rw	Device Control
0x6041	0x00	<b>108.3</b>	TX	ro	Device Control
0x6042	0x00	<b>110.5</b>	TX / RX	rw	Velocity Mode
0x6043	0x00	<b>18.21</b>	TX	ro	Velocity Mode
0x6044	0x00	<b>18.22</b>	TX	ro	Velocity Mode
0x6046	0x01	<b>110.16</b>	TX	ro	Velocity Mode
0x6046	0x02	<b>110.15</b>	TX / RX	rw	Velocity Mode
0x604F	0x00	<b>110.6</b>	TX / RX	rw	Velocity Mode
0x6050	0x00	<b>110.7</b>	TX / RX	rw	Velocity Mode
0x605A	0x00	<b>108.13</b>	TX	rw	Device Control
60x05B	0x00	<b>108.14</b>	TX	rw	Device Control
0x605C	0x00	<b>108.15</b>	TX	rw	Device Control
0x6060	0x00	<b>109.1</b>	TX / RX	rw	Device Control
0x6061	0x00	<b>109.2</b>	TX	ro	Device Control
0x6064	0x00	<b>121.9</b>	TX	ro	Position Control Function
0x6067	0x00	<b>121.5</b>	TX	rw	Position Control Function
0x6068	0x00	<b>121.6</b>	TX / RX	rw	Position Control Function
0x6069	0x00	<b>106.10</b>	TX	ro	Profile Velocity Mode
0x606A	0x00	-	-	ro	Profile Velocity Mode

CANopen-Objekt-Nummer		Parameter Nr.	PDO-Mapping	Access - Typ	Betriebsart nach CiA® 402
Index	Subindex				
0x606B	0x00	<b>18.21</b>	TX	ro	Profile Velocity Mode
0x606C	0x00	<b>18.22</b>	TX	ro	Profile Velocity Mode
0x6071	0x00	<b>18.50</b>	TX / RX	rw	Profile Torque Mode
0x6072	0x00	<b>138.14</b>	TX / RX	rw	Profile Torque Mode
0x6077	0x00	<b>47.3</b>	TX	ro	Profile Torque Mode
0x607A	0x00	<b>118.16 / 136.3</b>	TX / RX	rw	Profile Position Mode
0x607C	0x00	<b>120.3</b>	TX	rw	Homing Mode
0x607D	0x01	<b>121.3</b>	TX	rw	Profile Position Mode
0x607D	0x02	<b>121.4</b>	TX	rw	Profile Position Mode
0x607E	0x00	<b>179.1</b>	-	rw	Factor Group
0x6080	0x00	<b>110.13</b>	TX	rw	Profile Position Mode
0x6081	0x00	<b>118.11</b>	TX	rw	Profile Position Mode
0x6083	0x00	<b>118.12</b>	TX	rw	Profile Position Mode
0x6084	0x00	<b>118.13</b>	TX	rw	Profile Position Mode
0x6085	0x00	<b>110.8</b>	TX	rw	Profile Position Mode
0x6086	0x00	<b>118.2</b>	TX / RX	rw	Profile Position Mode
0x608F	0x01	<b>179.2</b>	-	rw	Factor Group
0x608F	0x02	<b>179.3</b>	-	rw	Factor Group
0x6090	0x01	<b>179.4</b>	-	rw	Factor Group
0x6090	0x02	<b>179.5</b>	-	rw	Factor Group
0x6091	0x01	<b>179.6</b>	-	rw	Factor Group
0x6091	0x02	<b>179.7</b>	-	rw	Factor Group
0x6092	0x01	<b>179.8</b>	-	rw	Factor Group
0x6092	0x02	<b>179.9</b>	-	rw	Factor Group
0x6098	0x00	<b>120.4</b>	TX	rw	Homing Mode
0x6099	0x01	<b>120.5</b>	TX	rw	Homing Mode
0x6099	0x02	<b>120.6</b>	TX	rw	Homing Mode
0x609A	0x00	<b>120.7</b>	TX	rw	Homing Mode
0x60B1	0x00	<b>18.68</b>	TX / RX	rw	Cyclic Synchronous Position Mode

CANopen-Objekt-Nummer		Parameter Nr.	PDO-Mapping	Access - Typ	Betriebsart nach CiA® 402
Index	Subindex				
0x60B8	0x00	124.2, 124.1, 124.30, 124.35	TX	rw	Homing Mode
0x60B9	0x00	124.3	TX	ro	Homing Mode
0x60BA	0x00	124.31	TX	ro	Homing Mode
0x60BB	0x00	124.32	TX	ro	Homing Mode
0x60BC	0x00	124.33	TX	ro	Homing Mode
0x60BD	0x00	124.34	TX	ro	Homing Mode
0x60F4	0x00	18.60	TX	ro	Cyclic Synchronous Position Mode
0x60FB	0x0A	18.76	TX	ro	Position Control Function
0x60FB	0x0D	118.6	TX / RX	rw	Position Control Function
0x60FB	0x15	118.10	TX / RX	rw	Position Control Function
0x60FB	0x17	136.5	TX / RX	rw	Position Control Function
0x60FB	0x18	120.10	TX / RX	rw	Position Control Function
0x60FB	0x19	120.8	TX / RX	rw	Position Control Function
0x60FF	0x00	110.4	TX / RX	rw	Profile Velocity Mode
0x6502	0x00	-	TX	ro	Device Control
0x6510	0x03	102.18	TX	ro	Info
0x6510	0x04	102.19	TX	ro	Info
0x6510	0x05	102.22	TX	ro	Info
0x6510	0x06	122.21	TX	ro	Info
0x6510	0x07	102.10	TX	ro	Info
0x6510	0x08	102.14	TX	ro	Info
0x6510	0x09	102.15	TX	ro	Info
0x6510	0x0A	102.1	TX	ro	Info
0x6510	0x0B	102.2	TX	ro	Info
0x6510	0x0C	102.4	TX	ro	Info
0x6510	0x0D	102.3	TX	ro	Info





## ANHANG C - UMSETZUNGSTABELLEN

Dieses Kapitel beinhaltet die Tabellen, welche die Umsetzung der CANopen-Kommunikationsobjekte in b maXX Regler-Kommunikationsparameter und umgekehrt spezifizieren.

Die Umsetzung erfolgt unter Angabe der Wertebereiche ( $x = x_{\min} \dots x_{\max}$ ) und der Abbildungsfunktion  $x = f(x)$  (im einfachsten Fall wird der Wert nur durchgereicht:  $y = x$ ).

Die Tabellen enthalten folgende Einträge:

<b>CANopen-Objekt:</b>	Bezeichnung des CANopen-Objektes aus CiA <sup>®</sup> 402
<b>Index ▶ P-Nr.:</b>	Abbildung der CANopen-Objektindizes auf b maXX <sup>®</sup> -Regler-Parameter
<b>Regler-Parameter:</b>	Bezeichnung des Regler-Parameters
<b>P-Nr. ▶ Index:</b>	Umsetzung der b maXX <sup>®</sup> -Reglerparameter auf CANopen-Objektindizes

CANopen-Objekt	Index Wertebereich	P-Nr. Normierung	Regler-Parameter	P-Nr. Wertebereich	Index Rück-Normierung	Kommentar
<b>Abort connection option code</b>	0x6007				0x6007	
	$y = -2^{15} \dots 2^{15}-1$				$y = -2^{15} \dots 2^{15}-1$	
	Manufacturer-specific				$y = -2^{15} \dots -1$	
	No action				$y = 0$	
	Fault signal				$y = 1$	
	Disable voltage command				$y = 2$	
	Quick stop command				$y = 3$	
	Reserved				$y = 4 \dots 2^{15}-1$	
<b>Error Code</b>	0x603F /ro		<b>Erster Fehler</b>	<b>100.5</b>	0x603F	Umwandlung der internen Reglerfehler in Emergency Messages nach CiA DSP 402-3
	$x = 0 \dots 0xFFFF$			$x = 0 \dots 0xFFFF$	$y = 0 \dots 0xFFFF$	
<b>Controlword</b>	0x6040	<b>108.1</b>	<b>Steuerwort</b>	<b>108.1</b>	0x6040	In der BA = 9 (Cyclic sync velocity mode) wird Bit 4 ignoriert.
	$x = 0 \dots 0xFFFF$	$y = x$		$x = 0 \dots 0xFFFF$	$y = x$	
Switch on	Bit 0	▶ unverändert	Einschalten	Bit 0	▶ unverändert	
Disable voltage	Bit 1	▶ unverändert	Spannung sperren	Bit 1	▶ unverändert	
Quick stop	Bit 2	▶ unverändert	Schnellhalt	Bit 2	▶ unverändert	
Enable operation	Bit 3	▶ unverändert	Betrieb freigeben	Bit 3	▶ unverändert	
Operation mode specific	Bit 4	▶ unverändert	betriebsartabhängig	Bit 4	▶ unverändert	
Operation mode specific	Bit 5	▶ unverändert	betriebsartabhängig	Bit 5	▶ unverändert	
Operation mode specific	Bit 6	▶ unverändert	betriebsartabhängig	Bit 6	▶ unverändert	
Fault reset	Bit 7	▶ unverändert	Reset Störung	Bit 7	▶ unverändert	
Halt	Bit 8	▶ unverändert	betriebsartabhängig	Bit 8	▶ unverändert	
Operation mode specific	Bit 9	▶ unverändert	betriebsartabhängig	Bit 9	▶ unverändert	
reserved	Bit 10	▶ unverändert	Reserve (immer 0)	Bit 10	▶ unverändert	
Manufacturer specific	Bit 11	▶ unverändert	betriebsartabhängig	Bit 11	▶ unverändert	
Manufacturer specific	Bit 12	▶ unverändert	betriebsartabhängig	Bit 12	▶ unverändert	
Manufacturer specific	Bit 13	▶ unverändert	betriebsartabhängig	Bit 13	▶ unverändert	
Manufacturer specific	Bit 14	▶ unverändert	Reserve (immer 0)	Bit 14	▶ unverändert	
Manufacturer specific	Bit 15	▶ unverändert	Reserve (immer 0)	Bit 15	▶ unverändert	



CANopen-Objekt	Index Wertebereich	P-Nr. Normierung	Regler-Parameter	P-Nr. Wertebereich	Index Rück-Normierung	Kommentar
<b>Statusword</b>	0x6041/ro		<b>Statuswort 1</b>	<b>108.3</b>	0x6041	
	x = 0 .. 0xFFFF			x = 0 .. 0xFFFF	y = x	
Ready to switch on			Einschaltbereit	Bit 0	▶ unverändert	
Switched on			Eingeschaltet	Bit 1	▶ unverändert	
Operation enabled			Betrieb freigegeben	Bit 2	▶ unverändert	
Fault			Störung	Bit 3	▶ unverändert	
Voltage enabled			Spannung liegt an	Bit 4	▶ unverändert	
Quick stop			Schnell-Halt	Bit 5	▶ unverändert	
Switched on disabled			Einschaltsperr	Bit 6	▶ unverändert	
Warning			Warnung	Bit 7	▶ unverändert	
Manufacturer specific			betriebsartabhängig	Bit 8	▶ unverändert	
Remote			Remote	Bit 9	▶ unverändert	
Targeted reached			Sollwert erreicht	Bit 10	▶ unverändert	
Internal limit active			Interne Begrenzung	Bit 11	▶ unverändert	
Operation mode specific			betriebsartabhängig	Bit 12	▶ unverändert	
Operation mode specific			betriebsartabhängig	Bit 13	▶ unverändert	
Manufacturer specific			konf. Statusbits	Bit 14	▶ unverändert	
Manufacturer specific			konf. Statusbits	Bit 15	▶ unverändert	
<b>vl_target_velocity</b>	0x6042	▶ <b>110.5</b>	<b>Eingang 16-Bit</b>	<b>110.5</b>	0x6042	
	y = -2 <sup>15</sup> ... 2 <sup>15</sup> -1	▶ y = x * 0x4000 / MotorMaxSpeed		x = -2 <sup>15</sup> ... 2 <sup>15</sup> -1	y = x * MotorMaxSpeed / 0x4000	
<b>vl_velocity_demand</b>	0x6043 /ro		<b>w2 Drehzahl-Sollwert</b>	<b>18.21</b>	0x6043	
				x = -1000000 ... 1000000	y = x / 6	
<b>vl_control_effort</b>	0x6044 /ro		<b>x2 Drehzahl-Istwert</b>	<b>18.22</b>	0x6044	
				x = -1000000 ... 1000000	y = x / 6	
<b>vl_velocity_min_max_-amount</b>	0x6046				0x6046	
vl_velocity_min_amount	SIX. 0x01	<b>110.16</b>	Betrag min. Eingangswert	<b>110.16</b>	SIX 0x01	SIX 1 ist immer Null, die min. Grenze ist auf Null festgelegt
	y = 0 ... 2 <sup>32</sup> -1	y = x		x = 0 ... 2 <sup>32</sup> -1	y = x	
vl_velocity_max_amount	SIX. 0x02	▶ <b>110.15</b>	Betrag max. Eingangswert	<b>110.15</b>	SIX 0x02	
	y = 0 ... 2 <sup>32</sup> -1	▶ y = x			y = x	



CANopen-Objekt	Index Wertebereich	P-Nr. Normierung	Regler-Parameter	P-Nr. Wertebereich	Index Rück-Normierung	Kommentar
vl_ramp_function_time	0x604F	▶ 110.6	Hochlaufzeit	110.6	▶ 0x604F	Hochlaufgeber Hochlaufzeit (1 = 1/1000 s ⇒ 1s = 1000). Die Auflösung beträgt 10 ms
	y = 0 ... 2 <sup>31</sup> -1	▶ y = x		x = 0 ... 650000	▶ y = x	
vl_slow_down_time	0x6050	▶ 110.7	Rücklaufzeit	110.7	▶ 0x6050	Hochlaufgeber Hochlaufzeit (1 = 1/1000 s, 1s = 1000). Die Auflösung beträgt 10 ms
	y = 0 ... 2 <sup>31</sup> -1	▶ y = x		x = 0 ... 650000	▶ y = x	
quick_stop_option_code	0x605A	▶ 108.13	Schnellhalt-Reaktion	108.13	▶ 0x605A	
	y = -2 <sup>15</sup> ... 2 <sup>15</sup> -1	▶ y = x		x = 0 .. 8	▶ y = x	
Manufacturer specific	x = -2 <sup>15</sup> ... -1	▶ y = x	unbenutzt	x = -2 <sup>15</sup> .. -1		
Disable drive function	x = 0	▶ y = x	Antrieb sofort sperren	x = 0	▶ y = x	
Slow down on slow down ramp	x = 1	▶ y = x	Rücklauf an Rücklauftrampe	x = 1	▶ y = x	
Slow down on quickstop ramp	x = 2	▶ y = x	Rücklauf an Schnellhalttrampe	x = 2	▶ y = x	
Slow down on current limit	x = 3	▶ y = x	Rücklauf an Stromgrenze	x = 3	▶ y = x	
Slow down on voltage limit	x = 4	▶ y = x	Rücklauf an Spannungsgrenze	x = 4	y = x	
Slow down on slow down ramp and stay in Quick Stop Active	x = 5	▶ y = x	Rücklauf an Rücklauftrampe und im Schnellhalt-Aktiv bleiben	x = 5	y = x	
Slow down on quickstop ramp and stay in Quick Stop Active	x = 6	▶ y = x	Rücklauf an Schnellhalttrampe und im Schnellhalt-Aktiv bleiben	x = 6	y = x	
Slow down on current limit and stay in Quick Stop active	x = 7	▶ y = x	Rücklauf an Stromgrenze und im Schnellhalt-Aktiv bleiben	x = 7	y = x	
Slow down on voltage limit and stay in Quick Stop Active	x = 8	▶ y = x	Rücklauf an Spannungsgrenze und im Schnellhalt-Aktiv bleiben	x = 8	y = x	
reserved	x = 9 .. 2 <sup>15</sup> -1		unbenutzt		y = 9 .. 2 <sup>15</sup> -1	

CANopen-Objekt	Index Wertebereich	P-Nr. Normierung	Regler-Parameter	P-Nr. Wertebereich	Index Rück-Normierung	Kommentar
<b>Shutdown_option_code</b>	0x605B	▶ <b>108.14</b>	<b>Stillsetzen-Reaktion</b>	<b>108.14</b>	▶ 0x605B	
	$x = -2^{15} \dots 2^{15}-1$	▶ $y = x$		$x = 0 \dots 3$	$y = x$	
Manufacturer specific	$x = -2^{15} \dots -1$					
Disable Drive function	$x = 0$	▶ $y = x$	Antrieb sofort sperren	$x = 0$	▶ $y = x$	
Slow down on slow down ramp	$x = 1$	▶ $y = x$	Rücklauf an Rücklauframpe	$x = 1$	▶ $y = x$	
Manufacturer specific	$x = 2$	▶ $y = x$	Rücklauf an Schnellhaltrampe	$x = 2$	▶ $y = x$	
Manufacturer specific	$x = 3$	▶ $y = x$	Rücklauf an Stromgrenze	$x = 3$	▶ $y = x$	
Reserved	$x = 4 \dots 2^{15}-1$					
<b>Disable_operation_option_code</b>	0x605C	▶ <b>108.15</b>	<b>Sperren-Reaktion</b>	<b>108.15</b>	▶ 0x605C	
	$x = -2^{15} \dots 2^{15}-1$	▶ $y = x$		$x = 0 \dots 3$	$y = x$	
Manufacturer specific	$x = -2^{15} \dots -1$					
Manufacturer specific	$x = 0$	▶ $y = x$	Antrieb sofort sperren	$x = 0$	▶ $y = x$	
Manufacturer specific	$x = 1$	▶ $y = x$	Rücklauf an Rücklauframpe	$x = 1$	▶ $y = x$	
Disable drive function	$x = 2$	▶ $y = x$	Rücklauf an Schnellhaltrampe	$x = 2$	▶ $y = x$	
Slow down on slow down ramp	$x = 3$	▶ $y = x$	Rücklauf an Stromgrenze	$x = 3$	▶ $y = x$	
Reserved	$x = 4 \dots 2^{15}-1$					

CANopen-Objekt	Index Wertebereich	P-Nr. Normierung	Regler-Parameter	P-Nr. Wertebereich	Index Rück-Normierung	Kommentar
<b>Modes_of_operation</b>	0x6060	▶ 109.1	<b>Soll-Betriebsart</b>	109.1	▶ 0x6060	
	x = -128 .. 127	▶ y = x		x = -128 .. 127	▶ y = x	
Manufacturer specific	x = -128 ... -12	y = x				
Manufacturer specific	x = -11	y = x	reserviert	x = -11	y = x	
Manufacturer specific	x = -10	y = x	U/f-Betrieb	x = -10	y = x	
Manufacturer specific	x = -9	y = x	Spannungsvorgabe	x = -9	y = x	
Manufacturer specific	x = -8	y = x	Stromvorgabe	x = -8	y = x	
Manufacturer specific	x = -7	▶ y = x	Selbstoptimierung	x = -7	▶ y = x	
Manufacturer specific	x = -6	▶ y = 5	Spindelpositionierung	x = -6	▶ x = -106	
Manufacturer specific	x = -5	▶ y = x	Gleichlauf	x = -5	▶ y = x	
Manufacturer specific	x = -4	▶ y = x	Lageregelung	x = -4	▶ y = x , y = 8	
Manufacturer specific	x = -3	▶ y = x	Drehzahlregelung	x = -3	▶ y = x; y = 3; y = 9	
Manufacturer specific	x = -2	▶ y = x	Stromregelung	x = -2	▶ y = x; y = 4; y = 10	
Manufacturer specific	x = -1	▶ y = x	Rastlagesuche	x = -1	▶ y = x	
No mode change / assigned	x = 0					
Profile position mode	x = 1	▶ y = x	Lagezielvorgabe	x = 1	▶ y = x	
Velocity mode	x = 2	▶ y = x	Geschwindigkeitsvorgabe 1	x = 2	▶ y = x	
Profile velocity mode	x = 3	▶ y = -3				
Torque profile mode	x = 4	▶ y = -2				
Reserved	x = 5	▶ y = x	Handfahrbetrieb	x = 5	▶ y = x	
Homing mode	x = 6	▶ y = x	Referenzfahrbetrieb	x = 6	▶ y = x	
Interpolated position mode	x = 7					
Cyclic sync position mode	x = 8	▶ y = -4				
Cyclic sync velocity mode	x = 9	▶ y = -3				
Cyclic sync torque mode	x = 10	▶ y = -2				
reserved	x = 11 ... 127					

CANopen-Objekt	Index Wertebereich	P-Nr. Normierung	Regler-Parameter	P-Nr. Wertebereich	Index Rück-Normierung	Kommentar
Modes_of_operation_display	0x6061/ro		Ist-Betriebsart	<b>109.2</b>	0x6061	
				x = -11 ... 6	y = x	
			reserviert	x = -11	y = x	
			U/f-Betrieb	x = -10	y = x	
			Spannungsvorgabe	x = -9	y = x	
			Stromvorgabe	x = -8	y = x	
			Selbstoptimierung	x = -7	y = x	
			Spindelpositionierung	x = -6	y = x	
			Gleichlauf	x = -5	y = x	
			Lageregelung	x = -4	y = x; y = 8	
			Drehzahlregelung	x = -3	y = x; y = 3; y = 9	
			Stromregelung	x = -2	y = x; y = 4; y = 10	
			Rastlagesuche	x = -1	y = x	
			Lagezielvorgabe	x = 1	y = x	
	Geschwindigkeitsvorgabe 1	x = 2	y = x			
	Handfahrbetrieb	x = 5	y = x			
	Referenzfahrbetrieb	x = 6	y = x			
Position_actual_internal_value	0x6063 /ro		<b>Lage-Istwert Umdr.+Winkel</b>	<b>18.54</b>	0x6063	UINT32 im Regler wird feldbusseitig mit einem Offset von -2 <sup>31</sup> versehen (UINT32 ⇒ INT32). Nur bei Änderungen Parameter 131.9 Bit 16 = 1, kein Offset von -2 <sup>31</sup> .
			x = 0 .. 2 <sup>32</sup> - 1	y = x - 2 <sup>31</sup>		
Position_actual_value	0x6064 /ro		<b>Positionierung Positions-Istwert</b>	<b>121.9</b>	0x6064	UINT32 im Regler wird feldbusseitig mit einem Offset von -2 <sup>31</sup> versehen (UINT32 ⇒ INT32). Nur bei Änderungen Parameter 131.9 Bit 16 = 1, kein Offset von -2 <sup>31</sup> .
			x = 0 .. 2 <sup>32</sup> - 1	y = x - 2 <sup>31</sup>		
Position_window	0x6067	▶ <b>121.5</b>	<b>Positionier-Fenster</b>	<b>121.5</b>	0x6067	
	x = 0 .. 2 <sup>32</sup> - 1	▶ y = x		x = 0 .. 2 <sup>32</sup> - 1	y = x	
Position_window_time	0x6068	▶ <b>121.6</b>	<b>Positionier-Fensterzeit</b>	<b>121.6</b>	0x6068	
	x = 0 .. 2 <sup>32</sup> - 1	▶ y = x		x = 1 ... 2 <sup>32</sup> - 1	y = x	
Velocity_sensor_actual_value	0x6069 /ro		<b>Lage-Istwert Winkel 32 Bit</b>	<b>106.10</b>	0x6069	
					x = 0 ... 2 <sup>32</sup> - 1	



CANopen-Objekt	Index Wertebereich	P-Nr. Normierung	Regler-Parameter	P-Nr. Wertebereich	Index Rück-Normierung	Kommentar
<b>Sensor_selection_code</b>	0x606A			-		Der Regler unterstützt nur Positiongeber, daher nur Anzeige.
velocity_actual_value_from_position_encoder			-	x = 0	y = x	
velocity_actual_value_from_velocity_encoder			nicht unterstützt			
<b>Velocity_demand_value</b>	0x606B /ro x = -2 <sup>15</sup> ... 2 <sup>15</sup> -1	▶ 18.21	<b>w2 Drehzahl-Sollwert</b>	18.21 x = -1000000 ... 1000000	▶ 0x606B ▶ y = x / 6	
<b>Velocity_actual_value</b>	0x606C /ro		<b>x2 Drehzahl-Istwert</b>	18.22 x = -1000000 ... 1000000	▶ 0x606C ▶ y = x / 6	
<b>Target_torque</b>	0x6071 x = -2 <sup>15</sup> ... 2 <sup>15</sup> -1	▶ 18.50 ▶ y = x*0x4000 / 1000	<b>Isq-Sollwert für Momenten-Regelung</b>	18.50 x = -2 <sup>15</sup> .. 2 <sup>15</sup> -1	▶ 0x6071 ▶ y = x*1000 / 0x4000	1000 entspricht 100,0% bezogen auf den max. verfügbaren Momentenstrom Parameter 19.8
<b>Max_torque</b>	0x6072 x = 0 ... 2 <sup>16</sup> - 1	▶ 138.14 ▶ y = x*0x4000 / 1000	<b>Zyklische Momenten-strom-begrenzung</b>	138.14 x = 0 ... 2 <sup>16</sup> - 1	▶ 0x6072 ▶ y = x*1000 / 0x4000	1000 entspricht 100,0% bezogen auf den max. verfügbaren Momentenstrom Parameter 19.8
<b>Torque_actual_value</b>	0x6077 /ro		<b>Isq-Istwert</b>	47.3 x = -4,096e+3 ... 4,096e+3	▶ 0x6077 ▶ y = x * 1000 / P19.8	1000 entspricht 100,0% bezogen auf den max. verfügbaren Momentenstrom Parameter 19.8
<b>Target_position</b>	0x607A x = -2 <sup>31</sup> ... 2 <sup>31</sup> -1	▶ 118.16 / 136.3 ▶ y = x, y = x + 2 <sup>31</sup>	<b>Relative Zielposition, Ziel-position</b>	118.16 / 136.3 x = -2 <sup>31</sup> .. 2 <sup>31</sup> -1 x = 0 ... 2 <sup>32</sup> - 1	▶ 0x607A ▶ y = x - 2 <sup>31</sup>	UINT32 im Regler wird feldbusseitig mit einem Offset von -2 <sup>31</sup> versehen (UINT32 ⇒ INT32). Nur bei Änderungen Parameter 131.9 Bit 16 = 1, kein Offset von -2 <sup>31</sup> . Bei Normierung der Daten von Regler in Richtung Feldbus wird bei Betriebsart 8 (Cyclic sync position mode) nur Parameter 136.3 und sonst nur Parameter 118.16 verwendet.
<b>Home_offset</b>	0x607C x = -2 <sup>31</sup> ... 2 <sup>31</sup> -1	▶ 120.3 ▶ y = x + 2 <sup>31</sup>	<b>Referenzpunkt</b>	120.3 x = 0 ... 2 <sup>32</sup> - 1	▶ 0x607C ▶ y = x - 2 <sup>31</sup>	UINT32 im Regler wird feldbusseitig mit einem Offset von -2 <sup>31</sup> versehen (UINT32 ⇒ INT32). Nur bei Änderungen Parameter 131.9 Bit 16 = 1, kein Offset von -2 <sup>31</sup> .

CANopen-Objekt	Index Wertebereich	P-Nr. Normierung	Regler-Parameter	P-Nr. Wertebereich	Index Rück-Normierung	Kommentar
<b>Software_position_limit</b>	0x607D				0x607D	UINT32 im Regler wird feldbusseitig mit einem Offset von $-2^{31}$ versehen (UINT32 $\Rightarrow$ INT32). Nur bei Änderungen Parameter 131.9 Bit 16 = 1, kein Offset von $-2^{31}$ .
Min position limit	SIX 0x01 $x = -2^{31} \dots 2^{31}-1$	▶ <b>121.3</b> ▶ $y = x + 2^{31}$	<b>Negativer Software-Endschalter</b>	<b>121.3</b> $x = 0 \dots 2^{32} - 1$	▶ SIX 0x01 ▶ $y = x - 2^{31}$	
Max position limit	SIX 0x02 $x = -2^{31} \dots 2^{31}-1$	▶ <b>121.4</b> ▶ $y = x + 2^{31}$	<b>Positiver Software-Endschalter</b>	<b>121.4</b> $x = 0 \dots 2^{32} - 1$	▶ SIX 0x02 ▶ $y = x - 2^{31}$	
<b>Polarity</b>	0x607E $x = 0 \dots 0xFFFF$	▶ <b>179.1</b> ▶ $y = x$	<b>Polarity</b>	<b>179.1</b> $x = 0 \dots 0xFFFF$	▶ 0x607D ▶ $y = x$	
Reserved	Bit 0 ... 5					Nur wirksam bei P131.9.0.0 Bit 14 = 1; wirkt auf 0x607A, 0x607D SIX 1, 0x607D SIX 2, 0x606C, 0x60B1, 0x60FF
Velocity polarity	Bit 6	▶ unverändert	Geschwindigkeitpolarität	Bit 6	▶ unverändert	Nur wirksam in der BA = 3 (Profile velocity mode) und BA = 9 (Cyclic sync velocity mode)
Position polarity	Bit 7	▶ unverändert	Positionspolarität	Bit 7	▶ unverändert	Nur wirksam in der BA = 1 (Profile position mode) und BA = 8 (Cyclic sync position mode)
<b>Max_motor_speed</b>	0x6080 $x = 0 \dots 2^{16}-1$	▶ <b>110.13</b> ▶ $y = x$	<b>Max. Drehzahl des Antriebs</b>	<b>110.13</b> $x = 1 \dots 1.0e+6$	▶ 0x6080 ▶ $y = x$	Die benutzerdefinierte Einheit (speed units) wird im Regler als U/min interpretiert
<b>Profile velocity</b>	0x6081 $x = 0 \dots 2^{32}-1$	▶ <b>118.11</b> ▶ $y = x$	<b>Geschwindigkeit</b>	<b>118.11</b> $x = 1 \dots 65535$	▶ 0x6081 ▶ $y = x$	
<b>Profile acceleration</b>	0x6083 $x = 0 \dots 2^{32}-1$	▶ <b>118.12</b> ▶ $y = x$	<b>Beschleunigung</b>	<b>118.12</b> $x = 7 \dots 65535$	▶ 0x6083 ▶ $y = x$	
<b>Profile deceleration</b>	0x6084 $x = 0 \dots 2^{32}-1$	▶ <b>118.13</b> ▶ $y = x$	<b>Verzögerung</b>	<b>118.13</b> $x = 7 \dots 65535$	▶ 0x6084 ▶ $y = x$	
<b>Quick_stop_deceleration</b>	0x6085 $x = 0 \dots 2^{32}-1$	▶ <b>110.8</b> ▶ $y = x$	<b>Schnellhaltzeit</b>	<b>110.8</b> $x = 0 \dots 650000$	▶ 0x6085 ▶ $y = x$	
<b>Motion profile type</b>	0x6086 $x = -2^{15} \dots 2^{15} - 1$	▶ <b>118.2</b>	<b>Modus</b>	<b>118.2</b>	▶ 0x6086	
<b>Position encoder resolution</b>	0x608F				0x608F	Nur wirksam bei P131.9 Bit 14 = 1; wirkt auf 0x6064, 0x6067, 0x607A, 0x607C, 0x607D SIX 1, 0x607D SIX 2, 0x60BA, 0x60BB, 0x60BC, 0x60BD, 0x60F4
Encoder increments	SIX 0x01 $x = 0 \dots 2^{32}-1$	▶ <b>179.2</b> ▶ $y = x$	<b>Lageauflösung Geber Inkremente</b>	<b>179.2</b> $x = 0 \dots 2^{32}-1$	▶ SIX 0x01 ▶ $y = x$	
Motor revolutions	SIX 0x02 $x = 0 \dots 2^{32}-1$	▶ <b>179.3</b> ▶ $y = x$	<b>Lageauflösung Motor Umdrehungen</b>	<b>179.3</b> $x = 0 \dots 2^{32}-1$	▶ SIX 0x02 ▶ $y = x$	



CANopen-Objekt	Index Wertebereich	P-Nr. Normierung	Regler-Parameter	P-Nr. Wertebereich	Index Rück-Normierung	Kommentar
<b>Velocity encoder resolution</b>	0x6090				0x6090	Nur wirksam bei P131.9 Bit 14 = 1; wirkt auf 0x606C, 0x6081, 0x6083, 0x6084, 0x6085, 0x6099 SIX 1, 0x6099 SIX 2, 0x609A, 0x60B1, 0x60FF
Encoder increments per second	SIX 0x01	▶ <b>179.4</b>	<b>Geschwindigkeitsauflösung Geber Inkremente pro s</b>	<b>179.4</b>	▶ SIX 0x01	
	x = 0 .. 2 <sup>32</sup> -1	▶ y = x		x = 0 .. 2 <sup>32</sup> -1	▶ y = x	
Motor revolutions per second	SIX 0x02	▶ <b>179.5</b>	<b>Geschwindigkeitsauflösung Motor Umdrehungen pro s</b>	<b>179.5</b>	▶ SIX 0x02	
	x = 0 .. 2 <sup>32</sup> -1	▶ y = x		x = 0 .. 2 <sup>32</sup> -1	▶ y = x	
<b>Gear ratio</b>	0x6091				0x6091	Nur wirksam bei P131.9 Bit 14 = 1; wirkt auf 0x6064, 0x6067, 0x606C, 0x607A, 0x607C, 0x607D SIX 1, 0x607D SIX 2, 0x6081, 0x6083, 0x6084, 0x6085, 0x6099 SIX 1, 0x6099 SIX 2, 0x609A, 0x60B1, 0x60BA, 0x60BB, 0x60BC, 0x60BD, 0x60F4, 0x60FF
Motor revolutions	SIX 0x01	▶ <b>179.6</b>	<b>Übersetzungsverhältnis Motorwelle Umdrehungen</b>	<b>179.6</b>	▶ SIX 0x01	
	x = 0 .. 2 <sup>32</sup> -1	▶ y = x		x = 0 .. 2 <sup>32</sup> -1	▶ y = x	
Shaft revolutions	SIX 0x02	▶ <b>179.7</b>	<b>Übersetzungsverhältnis Antriebswelle Umdrehungen</b>	<b>179.7</b>	▶ SIX 0x02	
	x = 0 .. 2 <sup>32</sup> -1	▶ y = x		x = 0 .. 2 <sup>32</sup> -1	▶ y = x	
<b>Feed constant</b>	0x6092					Nur wirksam bei P131.9 Bit 14 = 1; wirkt auf 0x6064, 0x6067, 0x606C, 0x607A, 0x607C, 0x607D SIX 1, 0x607D SIX 2, 0x6081, 0x6083, 0x6084, 0x6085, 0x6099 SIX 1, 0x6099 SIX 2, 0x609A, 0x60B1, 0x60BA, 0x60BB, 0x60BC, 0x60BD, 0x60F4, 0x60FF
Feed	SIX 0x01	▶ <b>179.8</b>	<b>Vorschubkonstante Vorschub</b>	<b>179.8</b>	▶ SIX 0x01	
	x = 0 .. 2 <sup>32</sup> -1	▶ y = x		x = 0 .. 2 <sup>32</sup> -1	▶ y = x	
Shaft revolutions	SIX 0x02	▶ <b>179.9</b>	<b>Vorschubkonstante Antriebswelle Umdrehungen</b>	<b>179.9</b>	▶ SIX 0x02	
	x = 0 .. 2 <sup>32</sup> -1	▶ y = x		x = 0 .. 2 <sup>32</sup> -1	▶ y = x	



CANopen-Objekt	Index Wertebereich	P-Nr. Normierung	Regler-Parameter	P-Nr. Wertebereich	Index Rück-Normierung	Kommentar
Homing_method	0x6098	▶ 120.4	Referenzfahr-Methode	120.4	▶ 0x6098	
	x = -128 ... 127	▶ y = x		x = -10 ... 35	▶ y = x	
Manufacturer specific	x = -128 ... -11					
Manufacturer specific	x = -10	▶ y = x	Anfahren mechanischer Anschlag mit Nullimpuls, Linksdrehung	x = -10	▶ y = x	
Manufacturer specific	x = -9	▶ y = x	Anfahren mechanischer Anschlag mit Nullimpuls, Rechtsdrehung	x = -9	▶ y = x	
Manufacturer specific	x = -8	▶ y = x	Anfahren mechanischer Anschlag, Linksdrehung	x = -8	▶ y = x	
Manufacturer specific	x = -7	▶ y = x	Anfahren mechanischer Anschlag, Rechtsdrehung	x = -7	▶ y = x	
Manufacturer specific	x = -6	▶ y = x	Anfahren des nächsten Gebernulwinkels	x = -6	▶ y = x	
Manufacturer specific	x = -5	▶ y = x	Anfahren des pos. Endschalters	x = -5	▶ y = x	
Manufacturer specific	x = -4	▶ y = x	Anfahren des neg. Endschalters	x = -4	▶ y = x	
Manufacturer specific	x = -3	▶ y = x	Referenzpunkt setzen	x = -3	▶ y = x	
Manufacturer specific	x = -2	▶ y = x	Anfahren des Gebernulwinkels bzw. des Nullimpulses mit Linksdrehung	x = -2	▶ y = x	
Manufacturer specific	x = -1	▶ y = x	Anfahren des Gebernulwinkels bzw. des Nullimpulses mit Rechtsdrehung	x = -1	▶ y = x	
No homing operation	x = 0		reserviert		y = 0	
Homing on the neg. limit switch	x = 1	▶ y = x	neg. Endschalter mit Nullimpuls bzw. Gebernulwinkel	x = 1	▶ y = x	
Homing on the pos. limit switch	x = 2	▶ y = x	pos. Endschalter mit Nullimpuls bzw. Gebernulwinkel	x = 2	▶ y = x	
Homing on the positive home switch & index pulse	x = 3	▶ y = x	pos. Nullpunktumschalter mit Nullimpuls bzw. Gebernulwinkel, Linksdrehung	x = 3	▶ y = x	

CANopen-Objekt	Index Wertebereich	P-Nr. Normierung	Regler-Parameter	P-Nr. Wertebereich	Index Rück-Normierung	Kommentar
Homing on the positive home switch & index pulse	x = 4	▶ y = x	pos. Nullpunktschalter mit Nullimpuls bzw. Gebernullwinkel, Rechtsdrehung	x = 4	▶ y = x	
Homing on the negative Home Switch & Index Pulse	x = 5	▶ y = x	neg. Nullpunktschalter mit Nullimpuls bzw. Gebernullwinkel, Rechtsdrehung	x = 5	▶ y = x	
Homing on the negative Home Switch & Index Pulse	x = 6	▶ y = x	neg. Nullpunktschalter mit Nullimpuls bzw. Gebernullwinkel, Linksdrehung	x = 6	▶ y = x	
Zero reference cam switch, left to pos. edge with Zero pulse; CW move	x = 7	▶ y = x	Nullpunktschalter, links von pos. Flanke mit Nullimpuls; Rechtsdrehung	x = 7	▶ y = x	
Zero reference cam switch, right to pos. edge with Zero pulse; CW move	x = 8	▶ y = x	Nullpunktschalter, rechts von pos. Flanke mit Nullimpuls; Rechtsdrehung	x = 8	▶ y = x	
Zero reference cam switch, left to neg. edge with Zero pulse; CW move	x = 9	▶ y = x	Nullpunktschalter, links von neg. Flanke mit Nullimpuls; Rechtsdrehung	x = 9	▶ y = x	
Zero reference cam switch, right to neg. edge with Zero pulse; CW move	x = 10	▶ y = x	Nullpunktschalter, rechts von neg. Flanke mit Nullimpuls; Rechtsdrehung	x = 10	▶ y = x	
Zero reference cam switch, right to neg. edge with Zero pulse; CCW move	x = 11	▶ y = x	Nullpunktschalter, rechts von neg. Flanke mit Nullimpuls; Linksdrehung	x = 11	▶ y = x	
Zero reference cam switch, right fo pos. edge with Zero pulse; CCW move	x = 12	▶ y = x	Nullpunktschalter, rechts von pos. Flanke mit Nullimpuls; Linksdrehung	x = 12	▶ y = x	
Zero reference cam switch, left to neg. edge with Zero pulse; CCW move	x = 13	▶ y = x	Nullpunktschalter, links von neg. Flanke mit Nullimpuls; Linksdrehung	x = 13	▶ y = x	
Zero reference cam switch, right to neg. edge with Zero pulse; CCW move	x = 14	▶ y = x	Nullpunktschalter, rechts von neg. Flanke mit Nullimpuls; Linksdrehung	x = 14	▶ y = x	
reserved	x = 15, 16		reserviert			
Negative limit switch	x = 17	▶ y = x	negativer Endschalters	x = 17	▶ y = x	
Positive limit switch	x = 18	▶ y = x	positiver Endschalters	x = 18	▶ y = x	

CANopen-Objekt	Index Wertebereich	P-Nr. Normierung	Regler-Parameter	P-Nr. Wertebereich	Index Rück-Normierung	Kommentar
Positive Zero reference switch, CCW move	x = 19	y = x	positiver Nullpunktumschalter; Linksdrehung	x = 19	y = x	
Positive Zero reference switch, CW move	x = 20	y = x	positiver Nullpunktumschalter; Rechtsdrehung	x = 20	y = x	
Negative Zero reference switch, CW move	x = 21	y = x	negativer Nullpunktumschalter; Rechtsdrehung	x = 21	y = x	
Negative Zero reference switch, CCW move	x = 22	y = x	negativer Nullpunktumschalter; Linksdrehung	x = 22	y = x	
Zero reference cam switch, left to pos. edge; CW move	x = 23	y = x	Nullpunktschalter, links von pos.; Rechtsdrehung	x = 23	y = x	
Zero reference cam switch, right to pos. edge; CW move	x = 24	y = x	Nullpunktschalter, rechts von pos. Flanke; Rechtsdrehung	x = 24	y = x	
Zero reference cam switch, left to neg. edge; CW move	x = 25	y = x	Nullpunktschalter, links neg. Flanke; Rechtsdrehung	x = 25	y = x	
Zero reference cam switch, right to neg. edge; CW move	x = 26	y = x	Nullpunktschalter, rechts von neg. Flanke; Rechtsdrehung	x = 26	y = x	
Zero reference cam switch, right to neg. edge; CCW move	x = 27	y = x	Nullpunktschalter, rechts von neg. Flanke; Linksdrehung	x = 27	y = x	
Zero reference cam switch, left to neg. edge; CCW move	x = 28	y = x	Nullpunktschalter, links von neg. Flanke; Linksdrehung	x = 28	y = x	
Zero reference cam switch, right to pos. edge; CCW move	x = 29	y = x	Nullpunktschalter, rechts von pos. Flanke; Linksdrehung	x = 29	y = x	
Zero reference cam switch, left to pos. edge; CCW move	x = 30	y = x	Nullpunktschalter, links von pos. Flanke; Linksdrehung	x = 30	y = x	
reserved	31 ... 32		reserviert	31 ... 32		
Nearest zero pulse; CCW move	x = 33	y = x	nächster Nullimpuls; Linksdrehung	x = 33	y = x	
Nearest zero pulse; CW move	x = 34	y = x	nächster Nullimpuls mit Rechtsdrehung	x = 34	y = x	
Homing on the current Position	x = 35	y = x	Referenzpunkt setzen	x = 35	y = x	
reserved	x = 36 .. 127		unbenutzt			

CANopen-Objekt	Index Wertebereich	P-Nr. Normierung	Regler-Parameter	P-Nr. Wertebereich	Index Rück-Normierung	Kommentar
<b>Homing_speeds</b>	0x6099				0x6099	
Speed_during_search_for_switch	SIX 0x01	▶ <b>120.5</b>	<b>Referenz-Geschwindigkeit</b>	<b>120.5</b>	▶ SIX 0x01	
	x = 0 .. 2 <sup>32</sup> -1	▶ y = x		x = 1 ... 65535	▶ y = x	
Speed_during_search_for_zero	SIX 0x02	▶ <b>120.6</b>	<b>Referenz-Endgeschwindigkeit</b>	<b>120.6</b>	▶ SIX 0x02	
	x = 0 .. 2 <sup>32</sup> -1	▶ y = x		x = 1 ... 65535	▶ y = x	
<b>Homing_acceleration</b>	0x609A	▶ <b>120.7</b>	<b>Referenz-Beschleunigung</b>	<b>120.7</b>	▶ 0x609A	
	x = 0 .. 2 <sup>32</sup> -1	▶ y = x		x = 7 ... 65535	▶ y = x	
<b>Velocity offset</b>	0x60B1	▶ <b>18.68</b>	<b>Drehzahl-Zusatz-Sollwert</b>	<b>18.68</b>	▶ 0x60B1	
	x = -2 <sup>31</sup> ... 2 <sup>31</sup> -1	▶ y = x * 6		x = -150000 ... 150000	▶ y = x / 6	
<b>Torque offset</b>	0x60B2	▶ <b>19.17</b>	<b>Isq-Zusatz-Sollwert</b>	<b>19.17</b>	▶ 0x60B2	
	x = -2 <sup>15</sup> ... 2 <sup>15</sup> -1	▶ y = x * MaxTorqueCurrent / 1000		x = -300 ... 300	▶ y = x * 1000 / MaxTorqueCurrent	
<b>Touch probe pos 1 pos value</b>	0x60BA /ro	▶	<b>DS402 Messtaster 1 pos. Flanke</b>	<b>124.31</b>	▶ 0x60BA	UINT32 im Regler wird feldbusseitig mit einem Offset von -2 <sup>31</sup> versehen (UINT32 → INT32). Nur bei Änderungen Parameter 131.9 Bit 16 = 1, kein Offset von -2 <sup>31</sup> .
		▶		x = 0 ...2 <sup>32</sup> -1	▶ y = x - 2 <sup>31</sup>	
<b>Touch probe pos 1 neg value</b>	0x60BB /ro	▶	<b>DS402 Messtaster 1 neg. Flanke</b>	<b>124.32</b>	▶ 0x60BB	UINT32 im Regler wird feldbusseitig mit einem Offset von -2 <sup>31</sup> versehen (UINT32 → INT32). Nur bei Änderungen Parameter 131.9 Bit 16 = 1, kein Offset von -2 <sup>31</sup> .
		▶		x = 0 ...2 <sup>32</sup> -1	▶ y = x - 2 <sup>31</sup>	
<b>Touch probe pos 2 pos value</b>	0x60BC /ro	▶	<b>DS402 Messtaster 2 pos. Flanke</b>	<b>124.33</b>	▶ 0x60BC	UINT32 im Regler wird feldbusseitig mit einem Offset von -2 <sup>31</sup> versehen (UINT32 → INT32). Nur bei Änderungen Parameter 131.9 Bit 16 = 1, kein Offset von -2 <sup>31</sup> .
		▶		x = 0 ...2 <sup>32</sup> -1	▶ y = x - 2 <sup>31</sup>	
<b>Touch probe pos 2 neg value</b>	0x60BD /ro	▶	<b>DS402 Messtaster 2 neg. Flanke</b>	<b>124.34</b>	▶ 0x60BD	UINT32 im Regler wird feldbusseitig mit einem Offset von -2 <sup>31</sup> versehen (UINT32 → INT32). Nur bei Änderungen Parameter 131.9 Bit 16 = 1, kein Offset von -2 <sup>31</sup> .
		▶		x = 0 ...2 <sup>32</sup> -1	▶ y = x - 2 <sup>31</sup>	
<b>Following error actual value</b>	0x60F4	<b>18.60</b>	<b>Schleppfehler Umdr. + Winkel</b>	<b>18.60</b>	▶ 0x60F4	
	x = -2 <sup>31</sup> ... 2 <sup>31</sup> -1	▶ y = x		x = -2 <sup>31</sup> ... 2 <sup>31</sup> -1	▶ y = x	
<b>position_control_parameter_set</b>	0x60FB					Herstellerspezifisches CANopen-Objekt

CANopen-Objekt	Index Wertebereich	P-Nr. Normierung	Regler-Parameter	P-Nr. Wertebereich	Index Rück-Normierung	Kommentar
Manufacturer specific	SIX 0x0A /ro		<b>Lage-Istwert Winkel</b>	<b>18.76</b>	SIX 0x0A	Default = 0
				$x = 0 \dots 2^{32}-1$	$y = x$	
Manufacturer specific	SIX 0x0D	▶ <b>118.6</b>	<b>Aktuelle Satznummer</b>	<b>118.6</b>	SIX 0x0A	Default = 0
	$x = 0 \dots 16$	$y = x$		$x = 0 \dots 16$	$y = x$	
Manufacturer specific	SIX 0x0F	▶ <b>120.3</b>	<b>Referenzpunkt</b>	<b>120.3</b>	SIX 0x0F	Default = 0x00020000
	$x = 0 \dots 2^{32}-1$	$y = x$		$x = 0 \dots 2^{32}-1$	$y = x$	
Manufacturer specific	SIX 0x15	▶ <b>118.10</b>	<b>Zielmodus</b>	<b>118.10</b>	SIX 0x15	Default = 0
	$x = -2 \dots 13$	$y = x$		$x = -2 \dots 13$	$y = x$	
Manufacturer specific	SIX 0x17	▶ <b>136.5</b>	<b>Ziel-Winkel</b>	<b>136.5</b>	SIX 0x17	Default = 0
	$x = 0 \dots 2^{32} - 1$	$y = x$		$x = 0 \dots 2^{32} - 1$	$y = x$	
Manufacturer specific	SIX 0x18	▶ <b>120.10</b>	<b>Geberoffset Referenzfahrt</b>	<b>120.10</b>	SIX 0x18	Default = 0
	$x = 0 \dots 2^{16}-1$	$y = x$		$x = 0 \dots 2^{16}-1$	$y = x$	
Manufacturer specific	SIX 0x19	<b>120.8</b>	<b>Referenz-Verzögerung</b>	<b>120.8</b>	SIX 0x19	Default = 200
	$x = 7 \dots 2^{16}-1$	$y = x$		$x = 7 \dots 2^{16}-1$	$y = x$	
<b>target_velocity</b>	0x60FF	▶ <b>110.4</b>	<b>Eingang 32-Bit</b>	<b>110.4</b>	0x60FF	Default = 0
	$x = -2^{31} \dots 2^{31}-1$	$y = x * 0x40000000 / \text{MotorMaxSpeed}$		$x = -2^{31} \dots 2^{31}-1$	$y = x * \text{MotorMaxSpeed} / 0x40000000$	
<b>Supported drive modes</b>	0x6502 /ro				0x6502	
				0x000003BF	$y = x$	
<b>Drive_data</b>	0x6510				0x6510	
Manufacturer specific	SIX 0x01/ ro		<b>reserviert</b>			
Manufacturer specific	SIX 0x02 / ro		<b>reserviert</b>			
Manufacturer specific	SIX 0x03 / ro		<b>Feldbus Controller Firmware Nummer</b>	<b>102.18</b>	SIX 0x03	Baumüller interne Feldbus-Firmware-Nummer
				$x = 0 \dots 2^{32} - 1$	$y = x$	
Manufacturer specific	SIX 0x04 / ro		<b>Feldbus Controller Firmware Version</b>	<b>102.19</b>	SIX 0x04	Darstellung im Format: Major[2].Minor[2].Fix[2]
				$x = 0 \dots 2^{32} - 1$	$y = x$	
Manufacturer specific	SIX 0x05 / ro		<b>Feldbus Controller Build Nummer</b>	<b>102.22</b>	SIX 0x05	Nummer für die Zählung von Beta-Ständen, Prototypen oder Nightly-Builds
				$x = 0 \dots 2^{32} - 1$	$y = x$	



CANopen-Objekt	Index Wertebereich	P-Nr. Normierung	Regler-Parameter	P-Nr. Wertebereich	Index Rück-Normierung	Kommentar
Manufacturer specific	SIX 0x06 / ro		<b>Feldbus Controller Firmware Typ</b>	<b>102.21</b>	▶ SIX 0x06	
				$x = 0 \dots 3$	▶ $y = x$	
			Serie	0	▶ $y = x$	
			Beta	1	▶ $y = x$	
			Prototyp	2	▶ $y = x$	
			Nightly Build	3	▶ $y = x$	
Manufacturer specific	SIX 0x07 / ro		<b>FPGA Kennung</b>	<b>102.10</b>	▶ SIX 0x07	Baumüller interne FPGA-Kennung
				$x = 0 \dots 2^{32} - 1$	▶ $y = x$	
Manufacturer specific	SIX 0x08 / ro		<b>FPGA Version</b>	<b>102.14</b>	▶ SIX 0x08	Darstellung im Format: Major[2].Minor[2].Fix[2]
				$x = 0 \dots 2^{32} - 1$	▶ $y = x$	
Manufacturer specific	SIX 0x09 / ro		<b>FPGA Firmware Nummer</b>	<b>102.15</b>	▶ SIX 0x09	Baumüller interne FPGA-Firmware-Nummer
				$x = 0 \dots 2^{32} - 1$	▶ $y = x$	
Manufacturer specific	SIX 0x0A / ro		<b>Regler Firmware Nummer</b>	<b>102.1</b>	▶ SIX 0x0A	Baumüller interne Regler-Firmware-Nummer
				$x = 0 \dots 2^{32} - 1$	▶ $y = x$	
Manufacturer specific	SIX 0x0B / ro		<b>Regler Firmware Version</b>	<b>102.2</b>	▶ SIX 0x0B	Darstellung im Format: Major[2].Minor[2].Fix[2]
				$x = 0 \dots 2^{32} - 1$	▶ $y = x$	
Manufacturer specific	SIX 0x0C / ro		<b>Regler Build Nummer</b>	<b>102.4</b>	▶ SIX 0x0C	Nummer für die Zählung von Beta-Ständen, Prototypen oder Nightly-Builds
				$x = 0 \dots 2^{32} - 1$	▶ $y = x$	
Manufacturer specific	SIX 0x0D / ro		<b>Regler Firmware Typ</b>	<b>102.3</b>	▶ SIX 0x0D	
			Serie	0	▶ $y = x$	
			Beta	1	▶ $y = x$	
			Prototyp	2	▶ $y = x$	
			Nightly Build	3	▶ $y = x$	
			Entwicklerversion	4	▶ $y = x$	



# ANHANG D - TECHNISCHE DATEN: POWERLINK CONTROLLED NODE

In diesem Anhang finden Sie eine Übersicht der Technischen Daten der **CANopen, CoE und POWERLINK** für BM3300/5000.

## D.1 Technische Merkmale

---

CPU	Nios® II
FPGA	Cyclone IV (Fa. Altera)
Speicher	8 kByte DPRAM, 64 MByte DDR2 SDRAM, 8 MByte Flash-Eprom
Baudrate	100 Mbit
Betriebsspannung	+5 V intern
Steckverbinder	2 RJ45 Buchsen, 8-polig

## D.2 Datenkanäle zum b maXX 5000-Regler

---

Für die Datenübertragung vom **b maXX 5000**-Regler zum **CANopen, CoE und POWERLINK** für BM3300/5000 stehen drei Kanäle zur Verfügung:

- zwei Prozessdatenkanäle (1 PDO je Kommunikationsrichtung)
- ein Bedarfsdatenkanal (Server-SDO)

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

Mit dem SDO-Transfer kann über das Objektverzeichnis auf alle **b maXX 5000** Parameter zugegriffen werden (Ausnahme String-Parameter).







## Abbildungsverzeichnis

ProDrive Feldbus-Slave .....	27
PDO Übertragungsarten .....	37
Zustandsmaschine der Kommunikation .....	47
NMT-Telegramm zur Steuerung der Kommunikationszustände .....	48
Node Guarding Protokoll .....	50
Initiate SDO Download Protocol .....	52
Initiate SDO Upload Expedited .....	54
Upload SDO Segmented Protocol .....	57
Kommunikationszyklus .....	61
Beispiel-Mapping mit zwei b maXX® 5000 .....	68
Telegrammaufbau für Beispiel-Mapping .....	77
Flexible Topologie: Linie, Baum oder Stern [1] .....	82
EtherCAT: Standard -IEEE 802.3-Frames [1] .....	83
Device Profile bei EtherCAT[1] .....	84
EtherCAT - Frame [1] .....	84
EtherCAT Telegramm [1] .....	85
EtherCAT Kommunikationsübergänge [1] .....	87
Aufbau der Mailbox .....	97
Mailbox-Header .....	97
CoE-Header .....	97
Mapping .....	102
Beispiel-Mapping mit einem b maXX® .....	105
Zustandsmaschine der Kommunikation .....	118
Konfiguration - Feldbus Gerät importieren .....	128
Konfiguration - Hardware Katalog .....	129
Konfiguration - System Designer .....	129
Konfiguration - Controlled Node in Physical View .....	130
Ohne PollResponse Chaining: .....	130
Mit PollResponse Chaining: .....	130
Ethernet Konfiguration .....	131
Gefundene Geräte .....	132
Gerät konfigurieren .....	132
Gerät auswählen .....	133





## Stichwortverzeichnis

<b>A</b>		Fieldbus Memory Management Unit	100
Abkürzungen	135	FMMU	100
Abort Code	99	FPGA	104
Abort Codes	59, 99	Frame Check Sequence	86
Abort-Telegramm	99	Frame Header	85
Access - Typ	139	Frame-Struktur	84
Adressraum	100	<b>G</b>	
AL Management	87	Garantiebestimmungen	10
Arbitrierung	36	Gefahr	8
<b>B</b>		Geräteprofil	112
Baumtopologie	82	Geräteprofile	84
Bedarfsdaten	119	Grundlagen	81
Betriebsarten	139	Guarding-Zeit	50
Broadcast-Telegramm	89	<b>H</b>	
Buszugriff	36, 110	Haftungsbeschränkung	9
<b>C</b>		Herstellerspezifische Objekte	137
CANopen- Objekt-Nummer	139	Hinweis	8
CANopen-Kommunikationsobjekte	143	Homing Mode	140
CoE-Header	97	<b>I</b>	
CoE-Slave	104	Index	139
Command Specifier	98	Info	141
Common Entries	139	Init	87
Control Register	87	IP-Adresse	110
CSMA/CA	36	Istwerte	102
CSMA/CD-Verfahren	110	<b>K</b>	
<b>D</b>		Knotenanzahl	
Datenbereich	84	maximal	38
Datenkanäle	159	Kommunikationsobjekte	14
DC	103	Kommunikationsrichtung	100
Device Control	139	Kommunikationszeiten	26
Distributed Clocks	103	Einstellung	26
Dummy-Mappin	103	Kommunikationszustände	
<b>E</b>		boot-up	38
Echtzeitanforderungen	100	Konfigurationsbeispiel	128
EMCY-Code	89	Kundendienst	10
ESM	87	Kurzreferenz Objektnummern	137
EtherCAT State Machine	87	<b>L</b>	
EtherCAT-Header	84	Leitung	36
EtherCAT-Telegramm	85	Lichtwellenleiter	82
Ethernet Header	85	Linientopologie	82
Ethernet Technology Group	81	<b>M</b>	
Ethernet-Telegramm	111	MAC-Adresse	110
Event Mechanismus	104	Mailbox-Header	97
Expedited Transfer	99	Mapping	101, 126
<b>F</b>		Mapping konfigurieren	105
fehlerhafte Zugriffe	59, 99	Multiplexor	98
Fehlerreaktionen	99		
Feldbus Standard	81		
Feldbus-Prozessdaten	107		



## Stichwortverzeichnis

---

<b>N</b>		<b>T</b>	
Netzwerk	109	Telegrammaufbau	123, 124
Netzwerkmanagement	46	Telegramm-Header	97
Node Guarding	49	Topologie Daten	82
<b>O</b>		<b>V</b>	
Objektnummer	99	Velocity Mode	139
Objektnummern	139	Vorsicht	8
Objektverzeichnis	13	<b>W</b>	
Offset	32	Warnhinweise	8
Operational	88	Warnung	8
<b>P</b>		Wertebereich	100
Parameternummer	139	<b>Z</b>	
PDO	100	Zugriffe	
PDO-Mapping	61, 101, 139	fehlerhaft	59, 99
fehlerhaft	79	Zustandsmaschine	46
Position Control Function	139		
Positionierung	32		
Varianten	33		
POWERLINK-Telegramm	111		
Pre-Operational	87		
Profile Torque Mode	140		
Profile Velocity Mode	139		
Prozessdaten	83		
Prozess-Daten-Objekte	100		
<b>R</b>			
Rahmenaufbau	83		
Referenzfahrt	32		
Remoteframes	49		
RPDO Achse 1	104		
RTR-Bit	49		
<b>S</b>			
Safe-Operational	87		
SDO	96, 119		
SDO Protokoll	99		
SDO-Zugriffe	99		
Segmented Transfer	99		
Service Daten Objekte	96, 119		
Sollwerte	102		
Speicher	159		
Statuswort	105		
Steckverbinder	159		
Sterntopologie	82		
Steuerwort	105		
Subindex	139		
Symbolerklärung	8		
Synchronisation	60		
Synchronisieren	103		
Synchronisierung	89, 100, 106		
SyncManager	100		
SyncManager Channels	100		



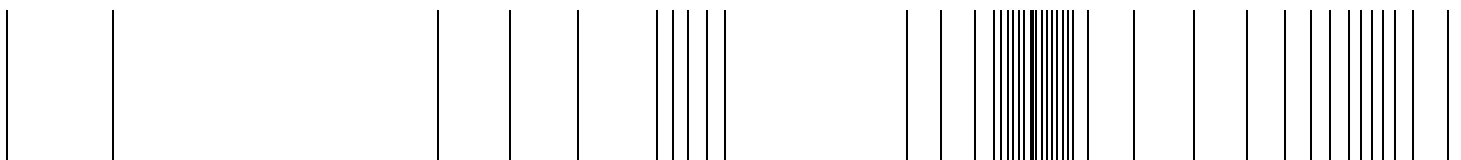
## Revisionsübersicht

Version	Stand	Änderungen
5.14006.01	26.06.2014	Neuerstellung
5.14006.02	09.09.2014	Kap. 3.1 Änderung Kommunikationsprofil Powerlink
5.14006.03	29.07.2015	Kap. 4.2 Abb. ergänzt Anhang B1: Beispiel ergänzt
5.14006.04	03.12.2015	Factor Group ergänzt in Kap. 3.2.1, Kap. 3.2.2, Kap. 5.3 (neu), Anh. B.2 und Anh. C
5.14006.05	07.09.2016	Kap. 3.3 neu, Kap. 11.11 Bilder aktualisiert, Kap. 11.12 bis 11.14 neu
5.14006.06	26.07.2019	Kap. 3.3.2: Nr. 7 bis 10 neu Anh. B2: Zeile 0x60B8 geändert Anh. B2 und Anh C: Zeile 0x60FB geändert





**be in motion**



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

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