



# Referenzhandbuch CANopen

## Schrittmotorsteuerungen und Plug & Drive Motoren

NANOTEC ELECTRONIC GmbH & Co. KG  
Kapellenstraße 6  
D-85622 Feldkirchen bei München

Tel. +49 (0)89-900 686-0  
Fax +49 (0)89-900 686-50  
[info@nanotec.de](mailto:info@nanotec.de)

## Impressum

© 2012

**Nanotec<sup>®</sup> Electronic GmbH & Co. KG**

Kapellenstraße 6

D-85622 Feldkirchen bei München

Tel.: +49 (0)89-900 686-0

Fax: +49 (0)89-900 686-50

Internet: [www.nanotec.de](http://www.nanotec.de)

Alle Rechte vorbehalten!

MS-Windows 2000/XP/Vista sind eingetragene Warenzeichen der Microsoft Corporation.

### Originalbetriebsanleitung

### Version/Änderungsübersicht

Version	Datum	Änderungen
2.2	12.10.2010	Überarbeitung C+P
2.3	03.11.2011	Überarbeitung C+P
2.4	12.04.2012	Aktualisierung der Objektbeschreibungen. Neu: ENCODER_TYPE (0x2011)

## Zu diesem Handbuch

### Inhalt

Dieses Dokument enthält eine kurze Übersicht über die wichtigsten Funktionen der Nanotec Schrittmotorsteuerungen und Plug & Drive Motoren sowie deren Verwendung über CANopen.

### Wichtige Hinweise

Nanotec<sup>®</sup> behält sich im Interesse seiner Kunden das Recht vor, technische Änderungen und Weiterentwicklungen von Hard- und Software zur Verbesserung der Funktionalität dieses Produktes ohne besondere Ankündigung vorzunehmen.

Für Kritik, Anregungen und Verbesserungsvorschläge wenden Sie sich bitte an die oben angegebene Adresse oder per E-Mail an: [info@nanotec.de](mailto:info@nanotec.de)

### Weitere Handbücher

Bitte beachten Sie auch folgende Handbücher von Nanotec:

<p><b>NanoCAN Benutzerhandbuch</b></p>	<p>Konfiguration der CAN-Kommunikation für CANopen-fähige Steuerungen mit der Software NanoCAN</p>	
<p><b>Programmierhandbuch</b></p>	<p>Programmierung von Steuerungen</p> <ul style="list-style-type: none"> <li>• Befehlsreferenz</li> <li>• NanoJ</li> <li>• COM-Schnittstelle</li> </ul>	
<p><b>Technische Handbücher</b></p>	<p>Anschluss und Inbetriebnahme von Schrittmotorsteuerungen oder Plug &amp; Drive Motoren</p>	

Die Handbücher stehen auf [www.nanotec.de](http://www.nanotec.de) zum Download zur Verfügung.

## Inhalt

<b>1</b>	<b>Inbetriebnahme .....</b>	<b>9</b>
1.1	Konfiguration der CAN-Kommunikation .....	9
1.1.1	Drehschalter für CAN-Node-ID .....	9
1.1.2	Konfiguration mit der Software NanoCAN .....	10
1.1.3	Konfiguration über CAN-Objekte .....	10
1.2	Motorkonfiguration .....	11
<b>2</b>	<b>Erste Schritte im CANopen-Modus .....</b>	<b>14</b>
2.1	Überblick .....	14
2.2	CANopen-Node hochfahren.....	14
2.3	CANopen-Node-Status abfragen.....	15
2.4	Leistungsteil einschalten.....	15
2.5	Operationsmodus auswählen .....	17
2.6	Fahrt starten.....	17
2.7	Neue Endposition vorgeben (0x12345) .....	18
<b>3</b>	<b>Control- und Statusword .....</b>	<b>19</b>
3.1	Einführung.....	19
3.2	Controlword (SDO 0x6040).....	19
3.3	Statusword (SDO 0x6041).....	20
<b>4</b>	<b>Profile Position Modus (PP).....</b>	<b>22</b>
4.1	Einführung.....	22
4.2	Controlword (SDO 0x6040).....	22
4.3	Statusword (SDO 0x6041).....	23
4.4	Quick Stop Option Code (SDO 0x605A).....	24
4.5	Position Demand Value (SDO 0x6062) .....	24
4.6	Target Position (SDO 0x607A) .....	24
4.7	Home Offset (SDO 0x607C) .....	24
4.8	Profile Velocity (SDO 0x6081) .....	25
4.9	End Velocity (SDO 0x6082) .....	25
4.10	Profile Acceleration (SDO 0x6083).....	25
4.11	Profile Deceleration (SDO 0x6084).....	25
4.12	Quick Stop Deceleration (SDO 0x6085) .....	26
4.13	Motion profile Type (0x6086) .....	26
<b>5</b>	<b>Homing Mode .....</b>	<b>27</b>
5.1	Controlword (0x6040).....	27
5.2	Statusword (0x6041).....	27
5.3	Homing Method (0x6098) .....	28
5.4	Homing Speeds (0x6099) .....	30
5.5	Homing Acceleration (0x609A) .....	31

<b>6</b>	<b>Velocity Mode</b> .....	<b>32</b>
6.1	Controlword (0x6040).....	32
6.2	Statusword (0x6041).....	32
6.3	VL target velocity (0x6042) .....	33
6.4	VL velocity demand (0x6043) .....	33
6.5	VL velocity actual value (0x6044) .....	33
6.6	VL velocity min max amount (0x6046).....	34
6.7	VL velocity acceleration (0x6048).....	34
6.8	VL velocity deceleration (0x6049).....	35
6.9	VL velocity quick stop (0x604A).....	35
6.10	VL dimension factor (0x604C) .....	36
<b>7</b>	<b>Interpolated Position Mode</b> .....	<b>37</b>
7.1	Einführung .....	37
7.2	Zeitlicher Ablauf .....	37
7.3	Controlword (SDO 0x6040).....	39
7.4	Statusword (SDO 0x6041).....	39
7.5	Interpolation Sub-Mode Select (SDO 0x60C0).....	39
7.6	Interpolation Data Record (SDO 0x60C1) .....	40
7.7	Interpolation Time Period (0x60C2).....	40
7.8	Interpolation Data Configuration (0x60C4) .....	41
7.9	Empfehlungen zum PDO-Mapping .....	42
7.10	Hinweis zur Inbetriebnahme .....	43
<b>8</b>	<b>Torque Mode</b> .....	<b>44</b>
8.1	Einführung .....	44
8.2	Controlword (SDO 0x6040).....	44
8.3	Statusword (SDO 0x6041).....	45
8.4	Target Torque (SDO 0x6071) .....	45
8.5	Max Profile Velocity (SDO 0x607F) .....	45
<b>9</b>	<b>Nicht Modusspezifische SDOs</b> .....	<b>46</b>
9.1	Digital-Eingänge (SDO 0x60FD).....	46
9.2	Digital-Ausgänge (SDO 0x60FE Subindex 1).....	46
9.3	Digital-Ausgänge Bitmaske (SDO 0x60FE Subindex 2).....	47
9.4	Polarity (0x607E).....	47
9.5	Producer Heartbeat Time (0x1017) .....	47
9.6	DEVICE TYPE (0x1000) .....	48
9.7	ERROR REGISTER (0x1001) .....	48
9.8	COB_ID_EMERGENCY_MESSAGE (0x1014) .....	49
9.9	IDENTITY_OBJECT_VENDOR_ID (0x1018).....	49
9.10	RPDO1_COMMUNICATION_PARAMETER (0x1400).....	50
9.11	RPDO2_COMMUNICATION_PARAMETER (0x1401).....	50

9.12	RPDO3_COMMUNICATION_PARAMETER (0x1402).....	51
9.13	RPDO4_COMMUNICATION_PARAMETER (0x1403) #.....	52
9.14	RPDO1_MAPPING (0x1600).....	53
9.15	RPDO2_MAPPING (0x1601).....	53
9.16	RPDO3_MAPPING (0x1602).....	54
9.17	RPDO4_MAPPING (0x1603).....	54
9.18	TPDO1_COMMUNICATION_PARAMETER (0x1800).....	55
9.19	TPDO2_COMMUNICATION_PARAMETER (0x1801).....	56
9.20	TPDO3_COMMUNICATION_PARAMETER (0x1802).....	57
9.21	TPDO4_COMMUNICATION_PARAMETER (0x1803).....	58
9.22	TPDO1_MAPPING (0x1A00).....	60
9.23	TPDO2_MAPPING (0x1A01).....	60
9.24	TPDO3_MAPPING (0x1A02).....	61
9.25	TPDO4_MAPPING (0x1A03).....	61
9.26	ERROR_CODE (0x603F) .....	62
9.27	VL_DIMENSION_FACTOR (0x604C).....	62
9.28	MODES_OF_OPERATION (0x6060) .....	62
9.29	MODES_OF_OPERATION_DISPLAY (0x6061) .....	63
9.30	POSITION_ACTUAL_INTERNAL_VALUE (0x6063).....	63
9.31	POSITION_ACTUAL_VALUE (0x6064).....	64
9.32	FOLLOWING_ERROR_WINDOW (0x6065) .....	64
9.33	FOLLOWING_ERROR_TIME_OUT (0x6066).....	64
9.34	POSITION_WINDOW (0x6067).....	64
9.35	POSITION_WINDOW_TIME (0x6068) .....	65
9.36	POS_RANGE_LIMIT (0x607B).....	65
9.37	SW_POS_LIMIT (0x607D).....	65
9.38	POSITION_ENCODER_RESOLUTION (0x608F).....	66
9.39	GEAR_RATIO (0x6091).....	66
9.40	FEED_CONSTANT (0x6092) .....	67
9.41	MAX_ACCEL (0x60C5) .....	67
9.42	MAX_DECEL (0x60C6) .....	68
9.43	SUPPORTED_DRIVE_MODES (0x6502).....	68
9.44	Dummy Objekt Boolean (0x0001).....	69
9.45	Dummy Objekt Integer8 (0x0002).....	69
9.46	Dummy Objekt Integer16 (0x0003).....	69
9.47	Dummy Objekt Integer32 (0x0004).....	69
9.48	Dummy Objekt Unsigned8 (0x0005).....	70
9.49	Dummy Objekt Unsigned16 (0x0006).....	70
9.50	Dummy Objekt Unsigned32 (0x0007).....	70
9.51	ERROR CODE LOG .....	70

<b>10</b>	<b>Analogeingang .....</b>	<b>71</b>
10.1	Überblick .....	71
10.2	Analog Input (0x6401).....	71
10.3	Analog Global Interrupt Enable (0x6423).....	72
10.4	Analog Input Interrupt Upper Limit (0x6424).....	72
10.5	Analog Input Interrupt Lower Limit (0x6425).....	73
10.6	Analog Input Interrupt Delta (0x6426).....	73
10.7	Analog Input Interrupt negative Delta (0x6427).....	74
10.8	Analog Input Interrupt positive Delta (0x6428) .....	74
<b>11</b>	<b>Herstellerspezifische CAN-Objekte.....</b>	<b>76</b>
11.1	STEP_MODE (0x2000).....	76
11.2	ENABLE_CL (0x2001).....	76
11.3	CL_CONFIGURATION (0x2002).....	77
11.4	A/D Converter (0x2003).....	80
11.5	Current control (0x2004).....	80
11.6	CAN enable und Baudrate (0x2005).....	83
11.7	Motor Pole Pairs (0x2006) .....	83
11.8	Brake Wait Time (0x2007) .....	84
11.9	Milliseconds Input Debounce Time (0x2008).....	85
11.10	Node-ID (0x2009).....	85
11.11	CL is enabled (0x200A) .....	85
11.12	CL POSCNT Offset (0x200B) .....	86
11.13	CL Lastwinkel Kurve (0x200C) .....	86
11.14	Encoder Drehrichtung (0x200D).....	87
11.15	Stromregler Parameter DSPdrive (0x200E) .....	88
11.16	Geschwindigkeitsmodus Reglertyp (0x200F) .....	89
11.17	Externe Referenzfahrt IO (0x2010).....	89
11.18	ENCODER_TYPE (0x2011) .....	90
11.19	COB-ID SYNC message (0x1005).....	90
11.20	Hardware Version (0x1009) .....	91
11.21	Software Version (0x100A) .....	91
11.22	Guard Time (0x100C) .....	91
11.23	Life Time Factor (0x100D) .....	92
11.24	Store Parameters (0x1010).....	92
11.25	Restore Parameters (0x1011).....	92
11.26	Acceleration notation index (0x608D).....	93
11.27	Acceleration dimension index (0x608E).....	93
<b>12</b>	<b>Prozessdatenobjekte (PDO).....</b>	<b>94</b>
12.1	Zweck der PDOs .....	94
12.2	PDO-Mapping .....	94

---

12.3	Dummy Objekte .....	96
<b>13</b>	<b>Inbetriebnahme über RS485 .....</b>	<b>97</b>
<b>14</b>	<b>Anhang: Verwendete Abkürzungen .....</b>	<b>98</b>
<b>15</b>	<b>Anhang: Mögliche Fehlermeldungen.....</b>	<b>99</b>
<b>16</b>	<b>Anhang: Mögliche Error Codes .....</b>	<b>100</b>
<b>17</b>	<b>Anhang: Motordaten .....</b>	<b>101</b>
17.1	Default Werte für Schrittmotoren .....	101
17.2	Default Werte für BLDC Motoren .....	101
17.3	Schrittmotoren der Serie STxxxx .....	101
17.4	BLDC Motoren der Serie DB22.....	102
17.5	BLDC Motoren der Serie DB28.....	102
17.6	BLDC Motoren der Serie DB33.....	103
17.7	BLDC Motoren der Serie DB42.....	103
17.8	BLDC Motoren der Serie DB57.....	107
17.9	BLDC Motoren der Serie DB87.....	108

# 1 Inbetriebnahme

## 1.1 Konfiguration der CAN-Kommunikation

### 1.1.1 Drehschalter für CAN-Node-ID

#### Funktion

Bei Steuerungen mit Drehschalter kann durch diesen ausgewählt werden, wie die Steuerung ihre Baudrate und Node-ID ermittelt. Mit den Drehschaltern wird eine hexadezimale Zahl eingestellt, die sich aus der 1-er Stelle (vom linken Drehschalter) und der 16-er Stelle (vom rechten Drehschalter) zusammensetzt.

Die Node-ID eines CANopen-Nodes kann zwischen 1 und 127 eingestellt werden. Sie wird von der Steuerung bei Anlegen der Spannung von den Drehschaltern oder aus dem EEPROM gelesen.

Näheres zum Schreiben der Node-ID und der Baudrate über CAN-Objekte siehe Abschnitt 1.1.3.

#### Einstellungen

In der folgenden Tabelle sind die Abhängigkeit der Node-ID und der Baudrate vom eingestellten Wert der Drehschalter aufgeführt.

<b>Wert Drehschalter (dezimal, SMC147-S)</b>	<b>0</b>	<b>1-127</b>	<b>128</b>	<b>129-255</b>
<b>Wert Drehschalter (hex, SMC147-S)</b>	<b>0x00</b>	<b>0x01-0x7F</b>	<b>0x80</b>	<b>0x81-0xFF</b>
<b>Wert Drehschalter (dezimal, PD4-N)</b>	<b>0</b>	<b>1-7</b>	<b>8</b>	<b>9-15</b>
<b>Wert Drehschalter (hex, PD4-N)</b>	<b>0x0</b>	<b>0x1-0x7</b>	<b>0x8</b>	<b>0x9-0xF</b>
Node-ID von Wert Drehschalter		X		X-128 bzw. X-8
Node-ID aus EEPROM	X		X	
Baudrate fest auf 1 Mbaud	X	X		
Baudrate aus EEPROM			X	X

#### Beispiel

Ist der linke Drehschalter auf 1 und der rechte Drehschalter auf 2 eingestellt, so ergibt sich die Zahl  $16 \cdot 2 + 1 = 33$ .

Hier ist die Baudrate also auf 1 Mbaud festgelegt und die Node-ID ist 33.

#### PD4-N

Die PD4-N verfügt nur über einen Drehschalter. Die Node-ID kann daher nur im Bereich von 1 bis 7 eingestellt werden. Andere Node-ID können nur über SDO eingestellt werden.

#### SMC112 und PD2-N

Die SMC112 und PD2-N verfügen aufgrund der beengten Platzverhältnisse über keinen Drehschalter. Die Node-ID und Baudrate können nur über SDO eingestellt werden.

## 1.1.2 Konfiguration mit der Software NanoCAN

**Hinweis:**  
Informationen zur Konfiguration finden Sie im NanoCAN Benutzerhandbuch.

## 1.1.3 Konfiguration über CAN-Objekte

### Vorgehensweise

Gehen Sie wie folgt vor, um eine Motorsteuerung über CAN-Objekte in Betrieb zu nehmen:

Schritt	Tätigkeit														
1	<p>Mit den beiden Drehschaltern den Wert 1 einstellen (linker Schalter auf 0, rechter Schalter auf 1).</p> <p>Damit ist die Baudrate fest auf 1Mbaud und die Node-ID auf 1 eingestellt. Ungültige Werte für die Baudrate im EEPROM werden so ignoriert.</p>														
2	<p>Standardeinstellungen der Firmware herstellen. Dazu den String „load“ in das SDO „restore all default parameters“ (0x1011 Subindex 1) schreiben.</p> <table border="1"> <thead> <tr> <th>COB-ID</th> <th>Datenbytes</th> <th>Beschreibung</th> </tr> </thead> <tbody> <tr> <td>601</td> <td>23 11 10 01 6C 6F 61 64</td> <td>Lade Defaults</td> </tr> <tr> <td>581</td> <td>60 11 10 01 00 00 00 00</td> <td>Antwort: OK (dauert bis zu 5 Sekunden)</td> </tr> </tbody> </table> <p>Alle Einstellungen werden auf die im EDS definierten Voreinstellungen zurückgesetzt, mit Ausnahme des Objekts zum Einstellen der Baudrate (siehe Schritt 4).</p>	COB-ID	Datenbytes	Beschreibung	601	23 11 10 01 6C 6F 61 64	Lade Defaults	581	60 11 10 01 00 00 00 00	Antwort: OK (dauert bis zu 5 Sekunden)					
COB-ID	Datenbytes	Beschreibung													
601	23 11 10 01 6C 6F 61 64	Lade Defaults													
581	60 11 10 01 00 00 00 00	Antwort: OK (dauert bis zu 5 Sekunden)													
3	<p>Node-ID mit Drehschaltern einstellen.</p> <p>Die Node-ID eines CANopen-Nodes kann zwischen 1 und 127 eingestellt werden. Diese Node-ID wird von der Steuerung bei Anlegen der Spannung von den Drehschaltern gelesen.</p>														
4	<p>Ergibt der eingestellte Wert der Drehschalter eine Zahl zwischen 1 und 127, ist die CAN-Baudrate auf 1 Mbaud festgelegt.</p> <p>Um eine andere Baudrate einzustellen: Gewünschte Baudrate über eine CAN-Karte mit Hilfe eines Service-Daten-Objekts (SDO) einstellen.</p> <p>Das zu schreibende Objekt hat die SDO-ID 0x2005 Sub-Index 0x0, Datentyp unsigned8.</p> <p>Folgende Tabelle gibt Aufschluss über die einstellbaren Baudraten:</p> <table border="1"> <thead> <tr> <th>Baudrate</th> <th>Wert für SDO 0x2005 Sub 0x0, Typ unsigned8</th> </tr> </thead> <tbody> <tr> <td>20 kbaud</td> <td>130</td> </tr> <tr> <td>50 kbaud</td> <td>131</td> </tr> <tr> <td>125 kbaud</td> <td>132</td> </tr> <tr> <td>250 kbaud</td> <td>133</td> </tr> <tr> <td>500 kbaud</td> <td>134</td> </tr> <tr> <td>1000 kbaud</td> <td>135</td> </tr> </tbody> </table>	Baudrate	Wert für SDO 0x2005 Sub 0x0, Typ unsigned8	20 kbaud	130	50 kbaud	131	125 kbaud	132	250 kbaud	133	500 kbaud	134	1000 kbaud	135
Baudrate	Wert für SDO 0x2005 Sub 0x0, Typ unsigned8														
20 kbaud	130														
50 kbaud	131														
125 kbaud	132														
250 kbaud	133														
500 kbaud	134														
1000 kbaud	135														
5	<p>Wert 0x65766173 (unsigned32) in das SDO 0x1010 Subindex 0x1 schreiben, um den Wert im internen EEPROM der Steuerung zu speichern.</p> <p>Es dauert ca. 5 Sekunden, bis die Firmware alle Einstellungen ins EEPROM übertragen hat und die SDO-Antwort sendet (siehe Schritt 2).</p>														
6	<p>Steuerung von der Stromversorgung trennen.</p>														

Schritt	Tätigkeit									
7	<p>Drehschalter auf den Wert Node-ID + 128 einstellen, damit der gespeicherte Wert als Baudrate übernommen wird.</p> <p>Sind für die Baudrate andere Werte als in der Tabelle aufgeführt eingestellt, antwortet die Steuerung zwar über die CAN-Schnittstelle, wird aber keine Fahraufträge akzeptieren.</p>									
8	<p>Einstellungen im EEPROM speichern.</p> <p>Dazu den Wert (u32)0x65766173 in das Objekt 0x1010 Subindex 1 schreiben (der Wert entspricht dem String „save“).</p> <table border="1"> <thead> <tr> <th>COB-ID</th> <th>Datenbytes</th> <th>Beschreibung</th> </tr> </thead> <tbody> <tr> <td>601</td> <td>23 10 10 01 73 61 76 65</td> <td>Speichere alle Einstellungen</td> </tr> <tr> <td>581</td> <td>60 10 10 01 00 00 00 00</td> <td>Antwort: OK</td> </tr> </tbody> </table>	COB-ID	Datenbytes	Beschreibung	601	23 10 10 01 73 61 76 65	Speichere alle Einstellungen	581	60 10 10 01 00 00 00 00	Antwort: OK
COB-ID	Datenbytes	Beschreibung								
601	23 10 10 01 73 61 76 65	Speichere alle Einstellungen								
581	60 10 10 01 00 00 00 00	Antwort: OK								

## 1.2 Motorkonfiguration

### Allgemeines

Vor der Inbetriebnahme der Motorsteuerung müssen die Motorparameter in der Steuerung an den jeweils angeschlossenen Motor angepasst werden, wenn diese von den Standardeinstellungen abweichen.

Zu diesen Einstellungen gehören:

- Phasen- und Ruhestrom des Motors (0x2004)
- Anzahl der Polpaare des Motors (0x2006)
- Auflösung des Drehgebers (0x608F)
- Getriebefaktor (0x6091)

### Phasen- und Ruhestrom

Mit dem Objekt 0x2004 Subindex 1 wird der Phasenstrom in Prozent des Maximalstroms eingestellt. Der Strom wird als prozentualer Anteil vom Nennstrom (Effektivwert) angegeben. Die Defaulteinstellung ist steuerungsabhängig.

Mit Subindex 2 wird der Ruhestrom ebenso wie der Phasenstrom in Prozent des Maximalstroms eingestellt. Mit diesem Strom wird die Motorwicklung beaufschlagt, wenn sich dieser im Stillstand befindet. Die Defaulteinstellung ist 20%.

### Motortyp

Mit dem Objekt 0x2004 Subindex 7 wird der Motortyp festgelegt. Diese Einstellung wird nicht von allen Steuerungen unterstützt. Bitte beachten Sie das entsprechende Datenblatt. Mögliche Werte:

- 0: Schrittmotor mit und ohne Quadratur-Encoder
- 1: BLDC mit Hall-Sensor und ohne Quadratur-Encoder
- 2: BLDC mit Hall-Sensor und mit Quadratur-Encoder

Wenn ein BLDC-Motor verwendet wird, muss darüber hinaus der Hall-Sensor konfiguriert werden (Objekt 0x2004 Subindex C).

Für alle Nanotec-Motoren außer der Baureihe DB42 muss für die Hall-Konfiguration der Wert 0x243015 eingetragen werden (Default-Einstellung). Für Motoren der Baureihe DB42 gilt der Wert 0x510342.

### Anzahl der Polpaare

Aus der Anzahl der Polpaare des Schrittmotors ergibt sich dessen Schrittwinkel.  
Die Formel zur Umrechnung lautet:

$$\text{Schrittwinkel} = 360^\circ / (4 * \text{Anzahl der Polpaare})$$

Somit hat ein Motor mit einem Schrittwinkel von  $1,8^\circ$  50 Polpaare (Defaulteinstellung bei der Steuerung) und ein  $0,9^\circ$  Motor hat 100 Polpaare.

Die Anzahl der Polpaare muss korrekt im Objekt 0x2006 Subindex 0 eingetragen werden.

### Auflösung des Drehgebers

Wird ein Drehgeber verwendet, muss dessen Auflösung der Steuerung bekannt sein. Die Auflösung wird in Inkrementen pro Umdrehung angegeben. Aufgrund des Quadraturprinzips hat beispielsweise ein Encoder mit 500 Strichen pro Umdrehung die vierfache Anzahl von Inkrementen pro Umdrehung: 2000.

Dieser Wert ist in das Objekt 0x608F Subindex 1 einzutragen.  
Der Defaultwert ist 2000.

### Getriebefaktor

In Objekt „Gear Ratio“ (0x6091) kann bei Verwendung eines Getriebes dessen Über- oder Untersetzung angegeben werden. Dieses Verhältnis wird als Bruch angegeben: In Subindex 1 steht die Anzahl der Motor-Umdrehungen und im Subindex 2 die Anzahl der Umdrehungen des Getriebes auf der Seite des Abtriebs. Die Defaulteinstellung ist 1/1.

### Umrechnung von Position, Geschwindigkeit und Beschleunigung

Sind die Motorparameter korrekt eingestellt, ist es für den Anwender nicht nötig, beim Wechsel des Schrittmodus oder beim Wechsel vom Open-Loop- in den Closed-Loop-Betrieb die Rampen, Verfahrenswege und Geschwindigkeiten anzupassen.

Die Werte für Geschwindigkeit, Beschleunigung und Position beziehen sich im PP-, HM-, und IP-Modus auf das Objekt „Feed Constant“ (Vorschubkonstante, SDO 0x6092).

Im Subindex 1 dieses Objekts wird der Zähler, im Subindex 2 der Nenner der Vorschubkonstante gespeichert. Der Defaultwert ist 2000/1. Das bedeutet für den Anwender, dass ein Weg von 2000 als Zielposition (SDO 0x607A) unabhängig vom Schrittmodus immer einer Umdrehung auf der abtreibenden Welle entspricht.

Für Geschwindigkeit gilt das entsprechend bezogen auf eine Sekunde, für die Beschleunigung entsprechend pro Sekunde zum Quadrat.

Im VL-Modus wird dementsprechend das Objekt „VL Dimension Factor“ (SDO 0x604C) verwendet. Die Geschwindigkeiten bzw. Beschleunigung beziehen sich hier aber im Gegensatz zu den anderen unterstützten Modi auf eine Minute bzw. eine Minute zum Quadrat.

## Maximale Geschwindigkeiten

Maximale Geschwindigkeiten und Beschleunigungen (PP und HM-Modus) können aus der folgenden Tabelle entnommen werden.

<b>Motortyp</b>	1,8°	0.9°		
<b>Polpaare (0x2006)</b>	50	100		
<b>Max. UPM</b>	4800	2400		
<b>Schrittmodus</b>	<b>Feed (0x6092 sub1) (Schritte pro Umdrehung)</b>	<b>Feed (0x6092 sub1) (Schritte pro Umdrehung)</b>	<b>Max. Profile Velocity (0x6081)</b>	<b>Max. Accel. (0x6083, 0x6084)</b>
Vollschritt	200	400	16000	1600000
Halbschritt	400	800	32000	3200000
Viertelschritt	800	1600	64000	6400000
Fünftelschritt	1000	2000	80000	8000000
Achtelschritt	1600	3200	128000	12800000
Zehntelschritt	2000	4000	160000	16000000
1/16 Schritt	3200	6400	256000	25600000
1/32 Schritt	6400	12800	512000	51200000
1/64 Schritt	12800	25600	1024000	102400000

## 2 Erste Schritte im CANopen-Modus

### 2.1 Überblick

Nach dem Anlegen der Versorgungsspannung ist die Steuerung nicht sofort automatisch betriebsbereit.

Es müssen zunächst folgende Schritte durchgeführt werden:

- CANopen-Node hochfahren
- CANopen-Node-Status abfragen
- Leistungsteil einschalten
- Operationsmodus auswählen
- Fahrt starten
- Neue Endposition vorgeben

Die einzelnen Schritte sind in den folgenden Abschnitten detailliert beschrieben.

### 2.2 CANopen-Node hochfahren

#### Status „Operational“

Um die Funktionalität der Steuerung nutzen zu können, muss nach jedem Einschalten der Steuerung diese in den Operational-Status versetzt werden.

Dies geschieht durch das Senden einer Network-Management-Nachricht mit der COB-ID 0x0 und dem 2 Byte langem Inhalt: <Kommando> und <Node-ID>.

#### Eingabe in IXXAT MiniMon

Die komplette Eingabe in IXXAT MiniMon lautet: „0 1 22“.

- 0 : COB-ID für NMT Nachricht
- 1: Starte Node
- 22 : CANopen Node-ID (hier 0x22 bzw. 34)

#### Kommandos

Die Kommandos sind:

- 0x01: Start Node (Wechsel zu Operational, Status 0x05)
- 0x02: Stop Node (Wechsel zu Stopped, Status 0x04)
- 0x80: Wechsel zu Pre-Operational (Status 0x7F, Zustand nach Anlegen der Betriebsspannung)
- 0x81: Neustart der Firmware, Rücksetzen aller CANopen-Einstellungen auf zuletzt um EEPROM abgelegte Werte
- 0x82: Neustart der Firmware, Rücksetzen aller CANopen-Einstellungen auf zuletzt um EEPROM abgelegte Werte

## 2.3 CANopen-Node-Status abfragen

### Statusabfrage

Der Status kann mit einem Remote Transmission Request (RTR) auf COB-ID 0x700 + Node-ID abgefragt werden.

Ein Motor mit der Node-ID 34(dec) sendet seinen Netzwerkstatus auf der COB-ID 0x700 + 34 = 0x722.

Um diese Nachricht zu Empfangen, muss ein Remote Transmission Request (RTR) für diese COB-ID gesendet werden.

Es ist auch möglich, den Motor diese Nachricht zyklisch senden zu lassen (siehe SDO 0x1017: Dynamic Heartbeat Time).

### Mögliche Status

Es gibt folgende Status:

- Status Pre-operational (Zustand nach Anlegen der Betriebsspannung, nach Neustart und Reset): **0x7F**  
In diesem Zustand können SDOs abgefragt und geschrieben werden aber keine PDOs gelesen oder geschrieben werden.
- Status Stopped: **0x04**  
In diesem Modus können weder SDOs noch PDOs abgefragt werden.
- Status Operational: **0x05**  
In diesem Modus können sowohl SDOs als auch PDOs gelesen und geschrieben werden

## 2.4 Leistungsteil einschalten

### Controlword

Das Einschalten des Leistungsteils geschieht über das Controlword. Dieses ist unter dem Service-Daten-Objekt (SDO) 0x6040 erreichbar.

### Abfrage des Statusword

Nach Senden jedes Kommandos wird empfohlen, durch Abfrage des Statusword zu überprüfen, ob der beabsichtigte Status erreicht wurde, da Statusübergänge verhindert werden (z.B. durch einen Unterspannungsfehler) oder sich verzögern können (z.B. durch die Verzögerungszeit der mechanischen Bremse oder durch die Ausführungszeit internen Übergänge).

### Eingabe in IXXAT MiniMon

Zum Aktivieren sind mehrere Übergänge des Controlwords nötig. Es folgen die Übergänge als Eingabe in IXXAT MiniMon für die CANopen Node-ID 34:

Ein Mehrfachübergang beim Einschalten (z.B. von „Switch on Disabled“ nach „Operation Enabled“) ist nicht möglich. Lediglich ein Sprung von „Ready to Switch On“ nach „Operational“ ist möglich.

COB-ID	Datenbytes	Beschreibung
622	2B 40 60 00 00 00	Switch On Disabled (Grundzustand)
5A2	60 40 60 00 00 00 00 00	Antwort: OK
622	40 41 60 00	Abfrage des Statusword
5A2	4B 41 60 00 60 02 00 00	Antwort: Switch on Disabled

COB-ID	Datenbytes	Beschreibung
622	2B 40 60 00 06 00	Ready to Switch On
5A2	60 40 60 00 00 00 00 00	Antwort: OK
622	40 41 60 00	Abfrage des Statusword
5A2	4B 41 60 00 21 02 00 00	Antwort: Ready to Switch On

COB-ID	Datenbytes	Beschreibung
622	2B 40 60 00 07 00	Switch On (Leistungsteil an)
5A2	60 40 60 00 00 00 00 00	Antwort: OK
622	40 41 60 00	Abfrage des Statusword
5A2	4B 41 60 00 33 02 00 00	Antwort: Switch On, Voltage Enabled

COB-ID	Datenbytes	Beschreibung
622	2B 40 60 00 0F 00	Operation Enabled
5A2	60 40 60 00 00 00 00 00	Antwort: OK
622	40 41 60 00	Abfrage des Statusword
5A2	4B 41 60 00 37 02 00 00	Antwort: Operation Enabled, Voltage Enabled

### Erläuterung der Datenbytes

Erläuterung zu den einzugebenden Daten in Hex:

- 622: Die COB-ID zum Senden von SDOs an den CANopen-Node 34(dec)
- 2B: Befehl zum Schreiben eines SDOs mit 2 Datenbytes
- 40: Niederwertiges Byte der SDO-ID 0x6040
- 60: Höherwertiges Byte der SDO-ID 0x6040
- 00: Sub-Index 0x00
- 00, 06, 07, 0f: Niederwertigstes Byte des Controlwords

Es ist zu beachten, dass die Daten in CANopen immer in der Intel-Notation übertragen werden, d. h. die niederwertigen Bytes zuerst.

### SDO-Adressierung über CAN

Service-Daten-Objekte (SDO) dienen zum Konfigurieren der jeweiligen CAN-Nodes. Da es viele verschiedene mögliche CANopen-Nodes gibt (Motoren, Sensoren usw.) sind im CANopen-Standard maximal  $2^{16} = 65536$  SDOs vorgesehen.

Die Anzahl der SDOs übersteigt die Anzahl der COB-IDs in CAN  $2^{11} = 2048$ . Um die SDOs dennoch über CAN adressieren zu können, wird die SDO-ID als Teil des Datenteils eines COB übertragen.

Zusätzlich sind für jedes SDO noch 256 Subindizes möglich. Ist in einem SDO mehr als nur Subindex 0 belegt, ist im Subindex 0 die Anzahl der vorhandenen Subindizes des jeweiligen SDOs vermerkt.

## 2.5 Operationsmodus auswählen

### Voraussetzung

Änderungen des Modus können im Status „Operation Enabled“ erfolgen.

Es sollte dabei sichergestellt werden, dass sich der Motor beim Kommandieren eines Moduswechsels nicht bewegt.

### Beispiel

Am Beispiel des PP-Modus (Profile Position bzw. Positionier-Modus) wird die Auswahl eines Modus aufgezeigt:

COB-ID	Datenbytes	Beschreibung
622	2F 60 60 00 01	Modus: Profile Position (PP)
5A2	60 60 60 00 00 00 00 00	Antwort: OK

## 2.6 Fahrt starten

### Schritt 1

Setzen des Bits „new setpoint“ im Controlword zum Starten einer Fahrt:

COB-ID	Datenbytes	Beschreibung
622	2b 40 60 00 1F 00	Start einer Fahrt
5A2	60 40 60 00 00 00 00 00	Antwort: OK

### Schritt 2

Rücksetzen des Bits, um Folgefahrten zu ermöglichen:

COB-ID	Datenbytes	Beschreibung
622	2b 40 60 00 0F 00	Rücksetzen des Start-Bits
5A2	60 40 60 00 00 00 00 00	Antwort: OK

Der Motor sollte sich jetzt ein Stück bewegt haben. Auf ein weiteres Toggeln des Bits reagiert der Motor nicht mehr, da er seine Endposition erreicht hat.

## 2.7 Neue Endposition vorgeben (0x12345)

### Schritt 1

COB-ID	Datenbytes	Beschreibung
622	23 7A 60 00 45 23 01 00	Endposition auf 0x12345
5A2	60 7A 60 00 00 00 00 00	Antwort: OK

### Schritt 2

COB-ID	Datenbytes	Beschreibung
622	2b 40 60 00 1F 00	Start einer Fahrt
5A2	60 40 60 00 00 00 00 00	Antwort: OK

### Schritt 3

COB-ID	Datenbytes	Beschreibung
622	2b 40 60 00 0F 00	Rücksetzen des Start-Bits
5A2	60 40 60 00 00 00 00 00	Antwort: OK

## 3 Control- und Statusword

### 3.1 Einführung

Control- und Statusword sind Servicedatenobjekte (SDO) und dienen zum Steuern und Abfragen des Motorstatus. Sie setzen sich aus einzelnen Bits zusammen.

Control- und Statusword sind standardmäßig auf Prozessdatenobjekte (PDO) gemappt.

In den jeweiligen verfügbaren Modi (Positionier-, Drehzahl- und Referenz-Modus) haben einige Bits immer die gleiche Bedeutung, andere sind hingegen spezifisch für den jeweiligen Modus.

Nachfolgend werden nur die Bits beschrieben, die in allen Modi die gleiche Bedeutung haben.

### 3.2 Controlword (SDO 0x6040)

#### Objektbeschreibung

<b>Bezeichnung</b>	Controlword
<b>SDO-ID</b>	0x6040
<b>Typ</b>	u16
<b>Wertebereich</b>	Bits

Dieses Objekt ist in der Standardeinstellung als PDO mit der COB-ID 0x200 + Node-ID verfügbar. Ein RTR ist nicht konfiguriert.

#### Bit 0, 1 und 3

Bit 0: Switch On

Bit 1: Enable Voltage

Bit 3: Enable Operation

Mit Bit 0, 1 und 3 wird der Motor vom Zustand Ausgeschaltet bis Betriebsbereit kommandiert (siehe auch Abschnitt 2).

Die Zustände sind:

- Ausgeschaltet (Switch On Disabled)
- Bereit zum Einschalten (Ready to Switch On)
- Eingeschaltet (Switch On)
- Betriebsbereit (Operation Enabled)

Ab dem Zustand Eingeschaltet (Switch On) ist die Haltebremse gelöst und das elektrische Feld des Motors aktiv. Eine Bewegung des Motors ist nur im Zustand Betriebsbereit (Operation Enabled) möglich.

Zusätzliche Zustände sind:

- Ausführung der Schnellbremsung (Quick Stop Active)
- Reagieren auf Fehler (Fault Reaction Active)
- Fehler (Fault)

Die Zustände müssen von Ausgeschaltet bis Betriebsbereit in der angegebenen Reihenfolge durchlaufen werden. Das geschieht durch das aufeinander folgende

Setzen der Bits 0, 1 und 3. Am Ende des Einschaltvorganges sind alle drei Bits gesetzt.

#### Bit 2

Quick Stop (invertiert: 0 bedeutet Quick Stop aktivieren)

Bit 2 muss immer auf „1“ gesetzt sein, außer es ist eine Schnellbremsung (Quick Stop) gefordert. Wird dieses Bit auf „0“ gesetzt, führt der Motor eine Schnellbremsung aus. Während der Schnellbremsung befindet sich der Motor im Zustand „Quick Stop Active“. Nach der Schnellbremsung geht der Motor automatisch in den Zustand „Switch On Disabled“.

#### Bit 4 bis 15

Bit 4 bis 6: Modusspezifisch.

Bit 7: Fault Reset.

Ist ein Fehler aufgetreten, befindet sich die Firmware nach der Fehlerreaktion im Zustand Fault. Um die Firmware in „Switch On Disabled“ zurück zu versetzen, muss dieses Bit einen Übergang von „0“ nach „1“ ausführen (eine Dauer-„1“ reicht hier nicht).

Bit 8: Halt (Modusspezifisch).

Bit 9: Modusspezifisch.

Bit 10: Reserviert.

Bit 11 bis 15: Herstellerspezifisch.

## 3.3 Statusword (SDO 0x6041)

### Objektbeschreibung

<b>Bezeichnung</b>	Statusword
<b>SDO-ID</b>	0x6041
<b>Typ</b>	u16
<b>Wertebereich</b>	Bits

Dieses Objekt ist in der Standardeinstellung als PDO mit der COB-ID 0x180 + Node-ID verfügbar. Ein RTR ist nicht konfiguriert. Bei jeder Änderung des Statusword wird dieses Objekt automatisch als PDO gesendet.

### Bits

Bit 0: ready to switch on

Bit 1: switched on

Bit 2: operation enabled: Der eingestellt Operationsmodus ist aktiv und nimmt Befehle entgegen (z.B. Profile Position Mode)

Bit 3: fault: Wird im Fehlerfall gesetzt

Bit 4: voltage enabled: Bit ist gesetzt, wenn Motor bestromt wird

Bit 5: quick stop

Bit 6: switch on disabled

Bit 7: warning

Bit 8: PLL sync complete: Wird gesetzt, sobald die Synchronisation mit dem SYNC-Objekt abgeschlossen ist.

Bit 9: remote

Bit 10: target reached: Wird gesetzt, wenn der Motor sein Ziel erreicht hat (Profile Position Mode)

Bit 11: internal limit active: Wird gesetzt, wenn die Sollwerte die Maximalgrenzen überschreiten.

Bit 12, 13: Moduspezifisch

Bit 14, 15: Herstellerspezifisch (nicht genutzt)

## 4 Profile Position Modus (PP)

### 4.1 Einführung

#### Funktion

Der Profile Position Modus dient dazu, Positionen relativ zur letzten Zielposition oder absolut zur letzten Referenzposition unter Vorgabe einer Geschwindigkeit und einer Rampe zu erreichen.

#### Ausführung von Fahraufträgen

Dieser Modus unterstützt auch, neue Fahraufträge an die Steuerung zu schicken, während der aktuelle Fahrauftrag noch nicht beendet ist.

Je nach Einstellung im Controlword wird der Fahrauftrag erst nach dem aktuellen ausgeführt (keine Option) oder der neue Fahrauftrag unterbricht den aktuellen Fahrauftrag („Change Set Immediately“). Es ist auch möglich, dass der aktuelle Fahrauftrag noch bis zur Zielposition ausgeführt wird, der Übergang zum neuen Satz aber „fliegend“ d.h. ohne die Geschwindigkeit zu reduzieren erfolgt (Change on Setpoint).

#### Positionsregler

Ist der Positionsregler (Closed Loop) aktiv, werden die Objekte 0x6067 und 0x6068 verwendet, um zu entscheiden, wann die Fahrt als beendet anzusehen ist. Außerdem werden die SDO 0x6065 und 0x6066 ausgewertet um den Schleppfehler im Statusword anzuzeigen.

Ist der Positionsregler nicht aktiv (Open Loop), kann bei Vorhandensein eines Drehgebers eine Positionsfehler-Korrektur nach einer Fahrt ausgeführt werden. Dazu dienen die Einstellungen in SDO 0x2004 Subindex 5 und 6. Ist der Fehler größer als die dort Eingestellte Toleranz, wird eine Korrekturfahrt gestartet.

### 4.2 Controlword (SDO 0x6040)

#### Objektbeschreibung

<b>Bezeichnung</b>	Controlword
<b>SDO-ID</b>	0x6040
<b>Typ</b>	u16
<b>Wertebereich</b>	Bits

Dieses Objekt ist in der Standardeinstellung als PDO mit der COB-ID 0x200 + Node-ID verfügbar. Ein RTR ist nicht konfiguriert.

Mit diesem Objekt wird der Motor eingeschaltet und es können Fahrbefehle ausgeführt werden.

#### Bits

Bit 0, 1 und 3 dienen zum Hochfahren des Leistungsteils. Diese sind in der allgemeinen Sektion beschrieben.

Bit 2 dient zum Auslösen eines Nothalts. Wird es auf „0“ gesetzt, führt der Motor eine Schnellbremsung mit der in SDO 0x6085 eingestellten Rampe durch. Danach geht der Motor in „Switch On Disabled“ (siehe allgemeine Beschreibung des Controlwords)

Bit 4 startet einen Fahrauftrag. Dieser wird bei einem Übergang von „0“ nach „1“ ausgeführt.

Bit 5: Ist dieses Bit auf „1“ gesetzt, wird ein durch Bit 4 ausgelöster Fahrauftrag sofort ausgeführt. Ist es auf „0“, wird der gerade ausgeführte Fahrauftrag zu Ende gefahren und erst im Anschluss der nächste Fahrauftrag gestartet. Wird im Augenblick des Startens kein Fahrauftrag durchgeführt, wird sofort gestartet.

Bit 6: Bei „0“ ist die Zielposition (SDO 0x607A) absolut und bei „1“ ist die Zielposition relativ zur aktuellen Position.

Bit 7: Fehler Reset (siehe allg. Beschreibung des Controlwords)

Bit 8: Halt:: Ist dieses Bit auf „0“ gesetzt, beschleunigt der Motor mit der eingestellten Rampe bis zur Zielgeschwindigkeit. Ist es auf „1“ gesetzt, bremst der Motor ab und bleibt stehen.

Bit 9: Ist dieses Bit gesetzt und soll der neue Fahrbefehl erst nach dem Ende des aktuellen Fahrbefehls ausgeführt werden, so wird die Geschwindigkeit erst bei Erreichen der ersten Zielposition geändert. Das bedeutet, dass vor Erreichen des ersten Ziels keine Bremsung durchgeführt wird, da der Motor auf dieser Position nicht stehen bleiben soll.

Bit 10 bis 15: Reserviert, auf 0 zu setzen.

## 4.3 Statusword (SDO 0x6041)

### Objektbeschreibung

<b>Bezeichnung</b>	Statusword
<b>SDO-ID</b>	0x6041
<b>Typ</b>	u16
<b>Wertebereich</b>	Bits

Dieses Objekt ist in der Standardeinstellung als PDO mit der COB-ID 0x180 + Node-ID verfügbar. Ein RTR ist nicht konfiguriert. Bei jeder Änderung des Statusword wird dieses Objekt automatisch als PDO gesendet.

### Bits

Bit 0 bis 9, 11, 14, 15: Siehe Abschnitt 3 „Control- und Statusword“.

Bit 10: Target reached: Diese Bit ist auf 1 gesetzt, wenn das letzte Ziel erreicht wurde (Motor im Stillstand)

Bit 11: Internal limit active: Dieses Bit wird gesetzt, wenn eine der Sollvorgaben interne Bereichsgrenzen überschreitet. (z.B. Geschwindigkeitsvorgabe größer als maximal mögliche Geschwindigkeit)

Bit 12: Set-point acknowledge: Dieses Bit bestätigt den Erhalt eines neuen Zielpunktes. Es wird synchron zu dem Bit „New set-point“ im Controlword gesetzt und zurückgesetzt.

Eine Ausnahme besteht, wenn eine neue Fahrt gestartet wird, während eine andere noch nicht abgeschlossen ist und die nächste Fahrt erst nach dem Abschluss der ersten Fahrt ausgeführt werden soll. In diesem Fall wird das Bit erst zurückgesetzt, wenn der Befehl angenommen wurde und die Steuerung bereit ist, neue Fahrbefehle auszuführen. Wird ein neuer Fahrauftrag gesendet, obwohl dieses Bit noch gesetzt ist, wird der neueste Fahrauftrag überschrieben.

Bit 13: Following error: Dieses Bit wird im Closed-Loop Betrieb gesetzt, wenn der Schleppfehler größer als die eingestellten Grenzen ist (siehe Kapitel 9.32 und 9.33)

## 4.4 Quick Stop Option Code (SDO 0x605A)

### Objektbeschreibung

<b>Bezeichnung</b>	Quick Stop Option Code
<b>SDO-ID</b>	0x605A
<b>Typ</b>	s16, ro
<b>Wertebereich</b>	3

Gibt das Verhalten bei einem Quick Stop an. Derzeit wird nur bremsen mit Maximalstrom mit anschließendem Wechsel auf „Switch On Disabled“ unterstützt.

## 4.5 Position Demand Value (SDO 0x6062)

### Objektbeschreibung

<b>Bezeichnung</b>	Position Demand Value
<b>SDO-ID</b>	0x6062
<b>Typ</b>	s32 ro
<b>Wertebereich</b>	s32

Gibt die aktuelle Sollposition an.

## 4.6 Target Position (SDO 0x607A)

### Objektbeschreibung

<b>Bezeichnung</b>	Target Position
<b>SDO-ID</b>	0x607A
<b>Typ</b>	s32 rw
<b>Wertebereich</b>	-100000000 bis 100000000

Gibt die Zielposition an.

Je nach Kommandierung des Controlwords wird die Endposition relativ zur aktuellen Position oder absolut zur Referenzposition interpretiert.

Mit dem Objekt 0x607E (Polarity) kann die Richtung umgekehrt werden.

## 4.7 Home Offset (SDO 0x607C)

### Objektbeschreibung

<b>Bezeichnung</b>	Home Offset
<b>SDO-ID</b>	0x607C
<b>Typ</b>	s32 rw
<b>Wertebereich</b>	s32

Gibt die Differenz zwischen Null-Position der Applikation und dem Referenzpunkt der Maschine an.

## 4.8 Profile Velocity (SDO 0x6081)

### Objektbeschreibung

<b>Bezeichnung</b>	Profile Velocity
<b>SDO-ID</b>	0x6081
<b>Typ</b>	u32 rw
<b>Wertebereich</b>	Je nach Feed Consant

Gibt die maximale Fahrgeschwindigkeit in Schritten pro Sekunde an.

## 4.9 End Velocity (SDO 0x6082)

### Objektbeschreibung

<b>Bezeichnung</b>	End Velocity
<b>SDO-ID</b>	0x6082
<b>Typ</b>	u32 rw
<b>Wertebereich</b>	Je Nach Feed Constant

Gibt die minimale Fahrgeschwindigkeit bei einer Trapezrampe in Schritten pro Sekunde an.

## 4.10 Profile Acceleration (SDO 0x6083)

### Objektbeschreibung

<b>Bezeichnung</b>	Profile Acceleration
<b>SDO-ID</b>	0x6083
<b>Typ</b>	u32
<b>Wertebereich</b>	1 bis 100000

Gibt die Beschleunigungsrampe in Schritten / s<sup>2</sup> an.

## 4.11 Profile Deceleration (SDO 0x6084)

### Objektbeschreibung

<b>Bezeichnung</b>	Profile Deceleration
<b>SDO-ID</b>	0x6084
<b>Typ</b>	u32
<b>Wertebereich</b>	1 bis 100000

Gibt die Bremsrampe in Schritten / s<sup>2</sup> an.

## 4.12 Quick Stop Deceleration (SDO 0x6085)

### Objektbeschreibung

<b>Bezeichnung</b>	Quick Stop Deceleration
<b>SDO-ID</b>	0x6085
<b>Typ</b>	u32
<b>Wertebereich</b>	1 bis 100000

Gibt die Nothalt Bremsrampe in Schritten / s<sup>2</sup> an.

## 4.13 Motion profile Type (0x6086)

### Objektbeschreibung

<b>Bezeichnung</b>	Motion profile type
<b>SDO-ID</b>	0x6086
<b>Typ</b>	s16
<b>Wertebereich</b>	0-1

Gibt den Rampentyp an.

Derzeit wird nur eine Sin2 (Wert=1) sowie eine lineare / trapezförmige Rampe unterstützt (Wert = 0).

## 5 Homing Mode

### 5.1 Controlword (0x6040)

#### Objektbeschreibung

<b>Bezeichnung</b>	Controlword
<b>SDO-ID</b>	0x6040
<b>Typ</b>	u16
<b>Wertebereich</b>	Bits

Dieses Objekt ist in der Standardeinstellung als PDO mit der COB-ID 0x200 + Node-ID verfügbar. Ein RTR ist nicht konfiguriert.

Mit diesem Objekt wird der Motor eingeschaltet und es können Fahrbefehle gestartet werden.

#### Bits

Bit 0, 1 und 3 dienen zum Hochfahren des Leistungsteils. Siehe Abschnitt 3 „Control- und Statusword“.

Bit 2 dient zum Auslösen eines Nothalts. Wird es auf „0“ gesetzt, führt der Motor eine Schnellbremsung mit der in SDO 0x6085 eingestellten Rampe durch. Danach geht der Motor in „Switch On Disabled“ (Siehe Abschnitt 3 „Control- und Statusword“.)

Bit 4 startet die Referenzierung. Diese wird solange ausgeführt, bis entweder die Referenzposition erreicht wurde oder Bit 4 wieder auf „0“ gesetzt wird.

Bit 5 bis 6: Reserviert.

Bit 7: Siehe Abschnitt 3 „Control- und Statusword“.

Bit 8: Wird dieses Bit gesetzt, stoppt der Motor.

Bit 9 bis 15: Siehe Abschnitt 3 „Control- und Statusword“.

### 5.2 Statusword (0x6041)

#### Objektbeschreibung

<b>Bezeichnung</b>	Statusword
<b>SDO-ID</b>	0x6041
<b>Typ</b>	u16
<b>Wertebereich</b>	Bits

Dieses Objekt ist in der Standardeinstellung als PDO mit der COB-ID 0x180 + Node-ID verfügbar. Ein RTR ist nicht konfiguriert. Bei jeder Änderung des Statusword wird dieses Objekt automatisch als PDO gesendet.

#### Bits

Bit 0 bis 9, 11, 14, 15: Siehe Abschnitt 3 „Control- und Statusword“.

Bit 10: Target reached: Ist auf „1“ gesetzt, wenn der Motor im Stillstand ist

Bit 11: Ungenutzt.

Bit 12: Homing attained: Auf „1“ gesetzt, wenn Referenzposition erreicht.

Bit 13: Auf „1“ gesetzt, wenn ein Fehler aufgetreten ist.

## 5.3 Homing Method (0x6098)

### Objektbeschreibung

<b>Bezeichnung</b>	Homing Method
<b>SDO-ID</b>	0x6098
<b>Typ</b>	u8
<b>Wertebereich</b>	19, 20, 21, 22 ,33, 34, 35, -2 bis -7

Dieses Objekt wählt den Homing-Modus aus.

Modus 21 und 22 sind erst ab Firmware 15-12-2008 (SMCI47-S) verfügbar.

#### Modus 19: Externe Referenzfahrt – Schalter als Öffner

- Suche des Schalters
- Motor dreht im Uhrzeigersinn
- Geschwindigkeit aus Objekt 0x6099\_1 (Search for switch)
- Solange Eingang 6 Highpegel hat
- Sobald Lowpegel an Eingang 6 anliegt (Schalter erreicht) wird die Richtung umgekehrt
- Motor dreht gegen den Uhrzeigersinn
- Geschwindigkeit aus Objekt 0x6099\_2 (Search for zero)
- Bis Eingang 6 wieder Highpegel hat (Schalter wieder frei)
- Motor hält an

#### Modus 20: Externe Referenzfahrt – Schalter als Schließer

- Suche des Schalters
- Motor dreht gegen den Uhrzeigersinn
- Geschwindigkeit aus Objekt 0x6099\_1 (Search for switch)
- Solange Eingang 6 Lowpegel hat
- Bei Erreichen des Schalters (Highpegel an Eingang 6) wird die Richtung umgekehrt
- Motor dreht im Uhrzeigersinn
- Geschwindigkeit aus Objekt 0x6099\_2 (Search for zero)
- Bis Eingang 6 wieder Lowpegel erreicht
- Motor hält an

#### Modus 21: Externe Referenzfahrt – Schalter als Öffner

- Suche des Schalters
- Motor dreht gegen den Uhrzeigersinn
- Geschwindigkeit aus Objekt 0x6099\_1 (Search for switch)
- Solange Eingang 6 Highpegel hat
- Bei Erreichen des Schalters (Lowpegel an Eingang 6) wird die Richtung umgekehrt
- Motor dreht im Uhrzeigersinn
- Geschwindigkeit aus Objekt 0x6099\_2 (Search for zero)
- Bis Eingang 6 wieder Highpegel erreicht
- Motor hält an

**Modus 22: Externe Referenzfahrt – Schalter als Schließer**

- Suche des Schalters
- Motor dreht im Uhrzeigersinn
- Geschwindigkeit aus Objekt 0x6099\_1 (Search for switch)
- Solange Eingang 6 Lowpegel hat
- Bei Erreichen des Schalters (Highpegel an Eingang 6) wird die Richtung umgekehrt
- Motor dreht gegen den Uhrzeigersinn
- Geschwindigkeit aus Objekt 0x6099\_2 (Search for zero)
- Bis Eingang 6 wieder Lowpegel erreicht
- Motor hält an

**Modus 33: Interne Referenzfahrt**

- Suche des Index-Strichs des internen Drehgebers
- Motor dreht im Uhrzeigersinn
- Geschwindigkeit aus Objekt 0x6099\_2 (Search for zero)
- Bis Indexstrich erreicht
- Bei Erreichen des Index-Strichs wird die Richtung umgekehrt
- Motor dreht gegen den Uhrzeigersinn
- Motor fährt vom Index-Strich herunter
- Motor hält an

**Modus 34: Interne Referenzfahrt**

- Suche des Index-Strichs des internen Drehgebers
- Motor dreht gegen den Uhrzeigersinn
- Geschwindigkeit aus Objekt 0x6099\_2 (Search for zero)
- Bis Index-Strich erreicht
- Bei Erreichen des Index-Striches wird die Richtung umgekehrt
- Motor dreht im Uhrzeigersinn
- Motor fährt vom Index-Strich herunter
- Motor hält an

**Modus 35: Positionsreset**

- Setzt die aktuelle Position auf Home-Offset, ohne dass die Welle bewegt

**Modus -2: Referenzfahrt auf Blockierung**

- Modus funktioniert nur mit Drehgeber (OL und CL)
- Erster Durchgang:  
Motor dreht im Uhrzeigersinn mit Geschwindigkeit aus Objekt 0x6099\_1 (Search for switch) bis Welle blockiert.  
Hierbei werden die Objekte „Following Error Window“ und „Following Error Timeout“ ausgewertet.
- Motor fährt eine elektrische Umdrehung rückwärts
- Zweiter Durchgang:  
Motor dreht im Uhrzeigersinn mit Geschwindigkeit aus Objekt 0x6099\_2 (Search for zero) bis Welle blockiert.  
Hierbei werden die Objekte „Following Error Window“ und „Following Error Timeout“ ausgewertet.

- Motor fährt eine elektrische Umdrehung rückwärts
- Motor fährt genau auf blockierte Position des zweiten Durchgangs und setzt die Position auf „Home Offset“

#### Modus -3: Referenzfahrt auf Blockierung

- Wie Modus -2, nur gegen den Uhrzeigersinn.

#### Modus -4: Referenzfahrt auf externes IO-Node

- Wie Modus 19, nur dass statt Eingang 6 ein externes IO-Node als Endschalter verwendet wird. (Siehe auch SDO 0x2010)

#### Modus -5: Referenzfahrt auf externes IO-Node

- Wie Modus 20, nur dass statt Eingang 6 ein externes IO-Node als Endschalter verwendet wird. (Siehe auch SDO 0x2010)

#### Modus -6: Referenzfahrt auf externes IO-Node

- Wie Modus 21 nur dass statt Eingang 6 ein externes IO-Node als Endschalter verwendet wird. (Siehe auch SDO 0x2010)

#### Modus -7: Referenzfahrt auf externes IO-Node

- Wie Modus 22, nur dass statt Eingang 6 ein externes IO-Node als Endschalter verwendet wird. (Siehe auch SDO 0x2010)

## 5.4 Homing Speeds (0x6099)

### Objektbeschreibung

<b>Bezeichnung</b>	Homing Speeds
<b>SDO-ID</b>	0x6099
<b>Typ</b>	2x u32
<b>Wertebereich</b>	1 bis 25000

Gibt die Geschwindigkeiten für den Homing-Modus in Schritten / s an.

In Subindex 1 wird die Geschwindigkeit für die Suche nach dem Schalter angegeben.

In Subindex 2 wird die (niedrigere) Geschwindigkeit für die Suche nach der Referenzposition angegeben.

### Hinweise

Die Geschwindigkeit in Subindex 2 ist gleichzeitig die Anfangsgeschwindigkeit beim Start der Beschleunigungsrampe. Wird diese zu hoch eingestellt, verliert der Motor Schritte bzw. dreht sich überhaupt nicht. Auch führt eine zu hohe Einstellung dazu, dass die Indexmarkierung übersehen wird. Diese Geschwindigkeit sollte daher unter 1000 Schritten pro Sekunde sein.

Des Weiteren muss die Geschwindigkeit in Subindex 1 größer als die in Subindex 2 sein.

Siehe auch Homing Method (SDO 0x6098, Abschnitt 5.3)

## 5.5 Homing Acceleration (0x609A)

### Objektbeschreibung

<b>Bezeichnung</b>	Homing Acceleration
<b>SDO-ID</b>	0x609A
<b>Typ</b>	u32
<b>Wertebereich</b>	1 bis 100000

Gibt die Beschleunigungsrampe für den Homing Mode in Schritten / s<sup>2</sup> an. Die Rampe wird nur beim Losfahren verwendet. Beim Erreichen des Schalters wird sofort auf die niedrigere Geschwindigkeit umgeschaltet und beim Erreichen der Endposition wird sofort gestoppt.

## 6 Velocity Mode

### 6.1 Controlword (0x6040)

#### Objektbeschreibung

<b>Bezeichnung</b>	Controlword
<b>SDO-ID</b>	0x6040
<b>Typ</b>	u16
<b>Wertebereich</b>	Bits

Dieses Objekt ist als statisches PDO mit der COB-ID 0x200 + Node-ID verfügbar. Ein RTR ist nicht möglich.

Mit diesem Objekt wird der Motor eingeschaltet und es können Fahrbefehle gestartet werden.

#### Bits

Bit 0, 1 und 3 dienen zum Hochfahren des Leistungsteils. Diese sind in der allgemeinen Sektion beschrieben.

Bit 2 dient zum Auslösen eines Nothalts. Wird es auf „0“ gesetzt, führt der Motor eine Schnellbremsung mit der in SDO 0x604A eingestellten Rampe durch. Danach geht der Motor in „Switch On Disabled“ (siehe allgemeine Beschreibung des Controlwords)

Bit 8: Halt: Bei einem Übergang von „1“ auf „0“ beschleunigt der Motor mit der eingestellten Rampe bis zur Zielgeschwindigkeit.

Bei einem Übergang von „0“ auf „1“ bremst der Motor ab und bleibt stehen.

### 6.2 Statusword (0x6041)

#### Objektbeschreibung

<b>Bezeichnung</b>	Statusword
<b>SDO-ID</b>	0x6041
<b>Typ</b>	u16
<b>Wertebereich</b>	Bits

Dieses Objekt ist als statisches PDO mit der COB-ID 0x180 + Node-ID verfügbar. Ein RTR ist möglich. Bei jeder Änderung des Statusword wird dieses Objekt automatisch als PDO gesendet.

#### Bit 11

Internal Limit Active: Dieses Bit wird gesetzt, wenn eine Sollvorgabe interne Grenzen über- oder unterschreitet (z.B. Zielgeschwindigkeit (SDO 0x6082) größer als 2500 Schritte / s).

## 6.3 VL target velocity (0x6042)

### Objektbeschreibung

<b>Bezeichnung</b>	VL target velocity
<b>SDO-ID</b>	0x6042
<b>Typ</b>	s16
<b>Wertebereich</b>	-25000 bis 25000

Gibt die Zielgeschwindigkeit in Schritten / s an.

Werte größer 0 stehen für Rechtslauf, Werte kleiner 0 für Linkslauf.

Die Richtung kann mit dem Objekt „Polarity“ umgekehrt werden.

## 6.4 VL velocity demand (0x6043)

### Objektbeschreibung

<b>Bezeichnung</b>	VL velocity demand
<b>SDO-ID</b>	0x6043
<b>Typ</b>	s16
<b>Wertebereich</b>	-25000 bis 25000

Gibt die aktuelle Zielgeschwindigkeit in Schritten / s an.

Dieses Objekt ist nur lesbar.

## 6.5 VL velocity actual value (0x6044)

### Objektbeschreibung

<b>Bezeichnung</b>	VL velocity actual value
<b>SDO-ID</b>	0x6044
<b>Typ</b>	s16
<b>Wertebereich</b>	-25000 bis 25000

Gibt die aktuelle Istgeschwindigkeit in Schritten / s an.

Nur bei aktiviertem Closed Loop wird ein Wert ausgegeben.

Dieses Objekt ist nur lesbar.

## 6.6 VL velocity min max amount (0x6046)

### Objektbeschreibung

<b>Bezeichnung</b>	VL velocity min max amount
<b>SDO-ID</b>	0x6046
<b>Typ</b>	2x u32
<b>Wertebereich</b>	1 bis 25000

Mit diesem Objekt können Minimalgeschwindigkeit und Maximalgeschwindigkeit in Schritten / s eingestellt werden.

Subindex 1 enthält die Minimalgeschwindigkeit.

Subindex 2 enthält die Maximalgeschwindigkeit.

### Hinweise

Wird eine Zielgeschwindigkeit (SDO 0x6042) vom Betrag her kleiner als die Minimalgeschwindigkeit angegeben, gilt die Minimalgeschwindigkeit. Ist die Zielgeschwindigkeit 0, hält der Motor an.

Eine Zielgeschwindigkeit größer als die Maximalgeschwindigkeit setzt die Geschwindigkeit auf die Maximalgeschwindigkeit und setzt das Bit 11 (internal limit active) im Statusword (SDO 0x6041).

## 6.7 VL velocity acceleration (0x6048)

### Objektbeschreibung

<b>Bezeichnung</b>	VL velocity acceleration
<b>SDO-ID</b>	0x6048
<b>Typ</b>	u32, u16
<b>Wertebereich</b>	u32, u16, beide ungleich 0

Setzt die Beschleunigungsrampe im VL-Modus.

Die Beschleunigung wird als Bruch angegeben:  
Geschwindigkeitsänderung pro Zeitänderung.

Subindex 1 enthält die Geschwindigkeitsänderung in Schritten / s (u32).

Subindex 2 enthält die Zeitänderung in s (u16).

### Hinweis

Weder Zähler noch Nenner dürfen auf 0 gesetzt werden.

## 6.8 VL velocity deceleration (0x6049)

### Objektbeschreibung

<b>Bezeichnung</b>	VL velocity deceleration
<b>SDO-ID</b>	0x6049
<b>Typ</b>	u32, u16
<b>Wertebereich</b>	u32, u16, beide ungleich 0

Setzt die Bremsrampe im VL-Modus.

## 6.9 VL velocity quick stop (0x604A)

### Objektbeschreibung

<b>Bezeichnung</b>	VL velocity quick stop
<b>SDO-ID</b>	0x604A
<b>Typ</b>	u32, u16
<b>Wertebereich</b>	u32, u16, beide ungleich 0

Setzt die Bremsrampe für die Schnellbremsung im VL-Modus.

Die Beschleunigung wird als Bruch angegeben:  
 Geschwindigkeitsänderung pro Zeitänderung.

Subindex 1 enthält die Geschwindigkeitsänderung in Schritten / s (u32).

Subindex 2 enthält die Zeitänderung in s (u16).

### Hinweis

Weder Zähler noch Nenner dürfen auf 0 gesetzt werden.

## 6.10 VL dimension factor (0x604C)

### Objektbeschreibung

<b>Bezeichnung</b>	VL dimension factor
<b>SDO-ID</b>	0x604C
<b>Typ</b>	2x s32
<b>Wertebereich</b>	s32, s32 ,beide größer 0

Bei diesem Objekt handelt es sich um einen Bruch, mit dem die Soll-Vorgaben des Nutzers im Velocity-Modus in UPM umgerechnet werden.  
(Im VL-Modus stellt die Motorsteuerung die Drehgeschwindigkeit intern in Umdrehungen pro Minute dar.)

Im Subindex 1 steht hierbei der Zähler und im Subindex 2 der Nenner dieses Bruchs.

### Objekte

Die Umrechnung wird auf folgende Objekte angewendet:

- 0x6042 (VL Target Velocity)
- 0x6043 (VL Velocity Demand)
- 0x6044 (VL Velocity Actual Value)
- 0x6046 (VL Velocity min/max Amount)
- 0x6048 (VL Velocity Acceleration)
- 0x6049 (VL Velocity Deceleration)
- 0x604A (VL Velocity Quick Stop)

### Umrechnung

Die Umrechnung folgt dem folgenden Prinzip:

**Benutzereinheit \* Umrechnungsfaktor = Einheit in UPM**

### Beispiel

Die Benutzerspezifische Einheit soll bei einem Wert von 2000 zu einer Umdrehung pro Sekunde führen (Default in der Firmware).

Der Umrechnungsfaktor lautet dann:  
60/2000 (SDO 0x604C Sub1/Sub2)

Die Umrechnung führt die Firmware dann wie folgt aus:  
 $2000 * 60/2000 = 60\text{UPM}$  (= 1 Umdrehung pro Sekunde)

Für Werte, die von der Firmware aktualisiert werden (SDO 0x6043 und 0x6044), läuft die Umrechnung entsprechend, so dass der Nutzer den Wert in der nutzerspezifischen Einheit aus dem SDO lesen kann.

## 7 Interpolated Position Mode

### 7.1 Einführung

#### Firmware

Dieser Modus ist ab der Firmware-Version 06-04-2009 verfügbar.

#### Funktion

Der Interpolated Position Mode dient zum Synchronisieren mehrerer Achsen. Hierzu übernimmt eine übergeordnete Steuerung die Rampen- bzw. Bahnberechnung und überträgt die jeweilige Sollposition, bei denen sich die Achse zu einem bestimmten Zeitpunkt befinden soll, zur Steuerung. Zwischen diesen Positions-Stützstellen interpoliert die Steuerung.

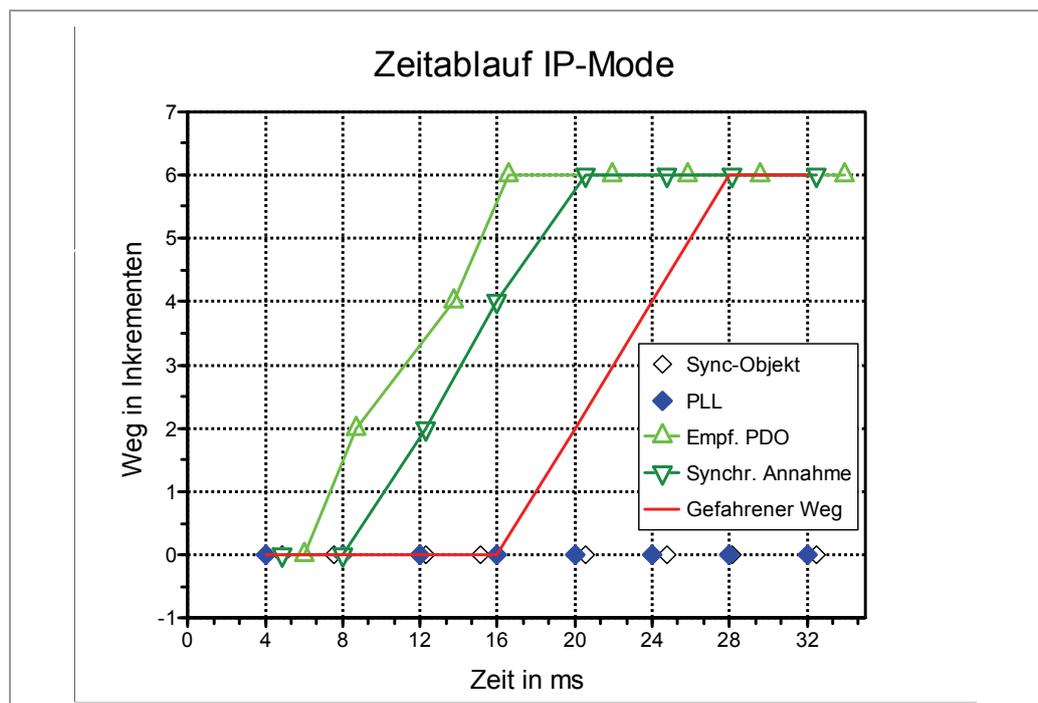
#### Synchronisierung zum SYNC-Objekt

Für den Interpolated Position Mode ist es notwendig, dass sich die Steuerung auf das SYNC-Objekt aufsynchronisiert. Dieses SYNC-Objekt ist in regelmäßigen Zeitabständen von der übergeordneten Steuerung zu senden. Die Synchronisation erfolgt, sobald die Steuerung in den NMT-Modus „Operational“ geschaltet wird (siehe Abschnitt 2.2).

### 7.2 Zeitlicher Ablauf

#### Darstellung

Die folgende Abbildung zeigt den zeitlichen Ablauf im Interpolated Position Mode. Auf der x-Achse ist die Zeit und auf der y-Achse die Position aufgetragen.



## Erläuterungen zur Legende

### Sync-Objekt

Die entlang der x-Achse aufgetragenen, nicht ausgefüllten Rauten markieren die Zeitpunkte, an denen die Steuerung ein Sync-Objekt von der übergeordneten Steuerung empfängt.

Die Zeitabstände der Sync-Objekte sind in diesem Fall nicht konstant. Die Abweichung von den Zeitpunkten, die sich ergeben, wenn das Sync-Objekt in gleichen Zeitabständen empfangen würde, nennt man Jitter. Ein Jitter wird beispielsweise durch eine hohe CAN-Buslast verursacht.

### PLL

Die Steuerung gleicht den Jitter des Sync-Objekts aus, indem sie sich mit Hilfe einer sogenannten „Phase Locked Loop“ (PLL) zu den Sync-Objekten synchronisiert. Die sich ergebenden Zeitpunkte (ausgefüllte Rauten auf der x-Achse) sind nahezu frei von Jitter.

### Empf. PDO

Die nach oben zeigenden Dreiecke kennzeichnen die Zeitpunkte, zu denen eine neue Sollposition empfangen wird. Die Höhe der Dreiecke auf der y-Achse gibt den Wert der Sollposition an. Es ist erkennbar, dass die sich ergebende Kurve zwischen  $t=4\text{ms}$  und  $t=20\text{ms}$  keine Gerade ist, obwohl die Position immer um zwei Inkremente vom vorherigen Wert erhöht wird. Das passiert aufgrund des zeitlichen Jitters, der auch den Objekten für die Sollposition anhafet.

### Synchr. Annahme

Zu den Zeitpunkten, die mit dem nach unten zeigenden Dreiecken markiert sind, übernimmt die Steuerung das PDO mit der Sollposition in seinen internen Puffer. Dieser Zeitpunkt ist genau der, an dem das Sync-Objekt empfangen wird. Ab diesem Zeitpunkt wäre dieser Wert auch über das SDO-Protokoll auslesbar. Auch diese Kurve ist aufgrund des Jitters der empfangenen Sync-Objekte keine gerade Linie.

### Gefahrener Weg

Die durchgezogene Linie stellt die vom Interpolated Position Mode gefahrene Strecke dar. Da die Interpolationszeitpunkte von der PLL abgeleitet werden, hat der Jitter des Sync-Objekts keinen Einfluss auf die Form der Kurve.

Die zeitliche Verzögerung zwischen der Sollposition und dem gefahrenen Weg ist notwendig, da schon am Anfang des Interpolations-Zeitraums (hier  $2 \cdot 4\text{ms}$ ) der Steuerung die Anfangs- und Zielposition des aktuellen Intervalls bekannt sein müssen, um die Geschwindigkeit dieses Intervalls berechnen zu können.

## 7.3 Controlword (SDO 0x6040)

### Objektbeschreibung

<b>Bezeichnung</b>	Controlword
<b>SDO-ID</b>	0x6040
<b>Typ</b>	u16
<b>Wertebereich</b>	Bits

Mit diesem Objekt kann der Leistungsteil hochgefahren und der IP-Modus aktiviert werden.

### Bits

Bit 0, 1 und 3 dienen zum Hochfahren des Leistungsteils. Siehe Abschnitt 3 „Control- und Statusword“.

Bit 2 dient zum Auslösen eines Nothalts. Wird es auf „0“ gesetzt, bleibt der Motor abrupt stehen. Danach geht der Motor in „Switch On Disabled“ (siehe Abschnitt 3 „Control- und Statusword“).

Bit 4 aktiviert den IP-Modus.

## 7.4 Statusword (SDO 0x6041)

### Objektbeschreibung

<b>Bezeichnung</b>	Statusword
<b>SDO-ID</b>	0x6041
<b>Typ</b>	u16
<b>Wertebereich</b>	Bits

Bit12: IP mode active: Diese Bit ist gesetzt, wenn der IP-Modus aktiv ist.

Restliche Bits: Siehe Abschnitt 3 „Control- und Statusword“.

## 7.5 Interpolation Sub-Mode Select (SDO 0x60C0)

### Objektbeschreibung

<b>Bezeichnung</b>	Interpolation Sub-Mode Select
<b>SDO-ID</b>	0x60C0
<b>Typ</b>	s16
<b>Wertebereich</b>	0

Mit diesem Objekt wird der Interpolations-Modus ausgewählt. Es wird nur eine lineare Interpolation (Wert 0) unterstützt. Andere Werte sind nicht erlaubt und werden ignoriert.

## 7.6 Interpolation Data Record (SDO 0x60C1)

### Objektbeschreibung

<b>Bezeichnung</b>	Interpolation Data Record
<b>SDO-ID</b>	0x60C1
<b>Typ</b>	Array
<b>Anzahl der Einträge</b>	1

Dieses Objekt beinhaltet die nächste anzufahrende Zielposition.

### Subindex 1

<b>Bezeichnung</b>	1st Setpoint
<b>SDO-ID</b>	0x60C1
<b>Subindex</b>	1
<b>Typ</b>	s32
<b>Wertebereich</b>	s32

In diesen Subindex ist die nächste Zielposition zu schreiben. Es wird empfohlen, bei Verwendung des Interpolated Position Mode dieses Objekt in ein synchrones RPDO zu mappen.

## 7.7 Interpolation Time Period (0x60C2)

### Objektbeschreibung

In der Steuerung sind diese Objekte aus Kompatibilitätsgründen vorhanden. Eingestellte Werte werden ignoriert. Die Interpolationszeit wird bei der Steuerung von den Zeitabständen des Synchron-Objekts abgeleitet.

<b>Bezeichnung</b>	Interpolation Time Period
<b>SDO-ID</b>	0x60C2
<b>Typ</b>	Array
<b>Anzahl der Einträge</b>	2

### Subindex 1

<b>Bezeichnung</b>	Interpolation time period value
<b>SDO-ID</b>	0x60C2
<b>Subindex</b>	1
<b>Typ</b>	u8
<b>Wertebereich</b>	u8

### Subindex 2

<b>Bezeichnung</b>	Interpolation time index
<b>SDO-ID</b>	0x60C2
<b>Subindex</b>	2
<b>Typ</b>	s8
<b>Wertebereich</b>	s8

## 7.8 Interpolation Data Configuration (0x60C4)

### Objektbeschreibung

Dieses Objekt dient als Platzhalter für zukünftige Modi. Seine Werte haben derzeit keinen Einfluss auf das Verhalten des Interpolated Position Mode.

<b>Bezeichnung</b>	Interpolation Data Configuration
<b>SDO-ID</b>	0x60C4
<b>Typ</b>	Array
<b>Anzahl der Einträge</b>	6

### Subindex 1

<b>Bezeichnung</b>	Maximum Buffer Size
<b>SDO-ID</b>	0x60C4
<b>Subindex</b>	1
<b>Typ</b>	u32
<b>Wertebereich</b>	1

Maximale Länge der Puffers

### Subindex 2

<b>Bezeichnung</b>	Actual Buffer Size
<b>SDO-ID</b>	0x60C4
<b>Subindex</b>	2
<b>Typ</b>	u32
<b>Wertebereich</b>	1

Tatsächliche Länge des Puffers

### Subindex 3

<b>Bezeichnung</b>	Buffer Organization
<b>SDO-ID</b>	0x60C4
<b>Subindex</b>	3
<b>Typ</b>	u8
<b>Wertebereich</b>	0

Wählt zwischen Ring- oder FIFO-Puffer aus.

### Subindex 4

<b>Bezeichnung</b>	Buffer Position
<b>SDO-ID</b>	0x60C4
<b>Subindex</b>	4
<b>Typ</b>	u16
<b>Wertebereich</b>	1

Aktuelle Buffer-Position. (Bei der Steuerung immer 1; wird nicht abgefragt oder aktualisiert)

### Subindex 5

<b>Bezeichnung</b>	Size of data record
<b>SDO-ID</b>	0x60C4
<b>Subindex</b>	5
<b>Typ</b>	u8
<b>Wertebereich</b>	4

Hat eine Bedeutung, wenn beispielsweise eine Spline-Interpolation verwendet wird. Für diesen Fall sind pro Interpolationsabschnitt mehrere Werte nötig. Dieses Objekt gibt die Größe eines Records an.

### Subindex 6

<b>Bezeichnung</b>	Buffer Clear
<b>SDO-ID</b>	0x60C4
<b>Subindex</b>	6
<b>Typ</b>	u8
<b>Wertebereich</b>	0

Mit diesem Objekt könnte man den Buffer löschen. Da der Buffer der Steuerung lediglich eine Länge von eins hat, macht ein Löschen keinen Sinn.

## 7.9 Empfehlungen zum PDO-Mapping

### Controlword (RPDO1)

Das PDO-Mapping des Controlwords (RPDO1) könnte derart geändert werden, dass zusätzlich noch das Objekt „Modes of Operation“ (SDO 0x6060) mit diesem PDO von der Steuerung empfangen wird. Der Transmission Type kann bei 255 (Asynchron) belassen werden. Es ist aber auch ohne Weiteres möglich, dieses Objekt als synchrones PDO (Transmission Type 0 bis 240) zu konfigurieren. Ist der IP-Modus einmal in Betrieb, ist es nicht nötig, dieses Objekt ständig zu aktualisieren, da das eine unnötige CAN-Buslast verursachen würde.

### Statuswords (TPDO1)

Das PDO-Mapping des Statuswords (TPDO1) könnte derart geändert werden, dass zusätzlich noch das Objekt „Modes of Operation Display“ (SDO 0x6061) mit diesem PDO von der Steuerung gesendet wird. Der Transmission Type kann bei 255 (Asynchron) belassen werden. Es ist aber auch ohne Weiteres möglich, dieses Objekt als synchrones azyklisches PDO (Transmission Type 0) zu konfigurieren. Es wird dann synchron zu dem Sync-Objekt gesendet, wenn sich Statusword oder „Modes of Operation Display“ ändert, was im normalen Betrieb des Interpolated Position Mode nicht vorkommt. Ist eine Protokollierung der Position gewünscht, kann an dieses TPDO zusätzlich noch das Objekt „Position actual Value“ (SDO 0x6064) angehängt werden. Dann sollte dieses PDO aber unbedingt als synchron und zyklisch konfiguriert werden (Transmission Type 1 bis 240). Es wäre dann auch denkbar, den Transmission Type auf Werte größer 1 einzustellen, damit nicht bei jedem Sync-Objekt die aktuelle Position gesendet wird. Hier muss der Anwender selbst entscheiden, was für ihn sinnvoll ist.

### Interpolation Data Record (SDO 0x60C1)

Das hauptsächliche Objekt für den IP-Modus ist der „Interpolation Data Record“ (SDO 0x60C1). Dieses Objekt besitzt kein Standardmapping. Sinnvoll wäre ein Mapping auf das RPDO2, Transmission Type 0 (zyklisch synchron). Die übergeordnete Steuerung muss nach jedem Sync-Objekt die nächste Sollposition in diesem PDO senden. Beim

darauf folgenden Sync-Objekt wird dann die Sollposition übernommen (siehe Abbildung 3). Bei Verwendung mehrerer Steuerungen macht es Sinn, die Sollposition für jeweils zwei Steuerungen in ein PDO zu packen und in der jeweiligen Steuerung die Sollposition der anderen Steuerung durch ein s32-Dummy Objekt (SDO 0x0004) zu maskieren. So kann die benötigte CAN-Bandbreite reduziert und die eingesparte Bandbreite für eine erhöhte Interpolationsfrequenz verwendet werden.

## 7.10 Hinweis zur Inbetriebnahme

Wenn vor der Verwendung des IP-Modus andere Modi verwendet wurden, ist es sehr wahrscheinlich, dass die aktuelle Position des Rotors von Null abweicht. Wird dann der IP-Modus ausgewählt, muss der Anwender dafür Sorge tragen, dass die Sollposition im Objekt „Interpolation Data Record“ (SDO 0x60C1) der tatsächlichen Position entspricht, bevor die Interpolation mit Bit 4 des Controlwords gestartet wird.

Wird dies nicht gemacht, kommt es beim Einschalten der Interpolation zu einer Bewegung zur Nullposition innerhalb eines Interpolationszeitraums (zeitlicher Abstand zweier Sync-Objekte). Im Open-Loop-Betrieb äussert sich das durch ein Rucken der Welle oder ein kurzes Pfeifen (hierbei gehen Schritte verloren). Im Closed-Loop-Betrieb regelt die Steuerung mit Maximalstrom, bis die Zielposition erreicht wird.

Um das zu verhindern, muss für den Open-Loop-Betrieb das Objekt „Position Demand value“ (SDO 0x6062) und für den Closed-Loop-Betrieb das Objekt „Position actual value“ (SDO 0x6064) als Anfangsposition für den Interpolation Data Record (SDO 0x60C1) übernommen werden.

## 8 Torque Mode

### 8.1 Einführung

Der Torque Mode (Drehmomentmodus) dient dazu, entsprechend eines vorgegebenen Zielwertes das Drehmoment einzustellen.

Da die sich ergebende Drehzahl von den Eigenschaften des Motors und des Lastmoments abhängt, ist es möglich, über das SDO 0x607F die maximale Drehzahl zu begrenzen.

Der Torque Mode wird mit dem Wert ,4' im SDO 0x6060 (modes of operation) ausgewählt.

Für den Torque Mode muss der Closed-Loop-Modus aktiv sein.

### 8.2 Controlword (SDO 0x6040)

Bezeichnung	Controlword
SDO-ID	0x6040
Typ	u16
Wertebereich	Bits

Dieses Objekt ist in der Standardeinstellung als PDO mit der COB-ID 0x200 + Node-ID verfügbar. Ein RTR ist nicht konfiguriert.

Mit diesem Objekt wird der Motor eingeschaltet und es können Fahrbefehle ausgeführt werden.

Bit 0, 1 und 3 dienen zum Hochfahren des Leistungsteils. Diese sind in der allgemeinen Sektion beschrieben.

Bit 2 dient zum Auslösen eines Nothalts. Wird es auf „0“ gesetzt, führt der Motor eine Schnellbremsung mit der in SDO 0x6085 eingestellten Rampe durch. Danach geht der Motor in „Switch On Disabled“ (siehe allgemeine Beschreibung des Controlwords)

Bit 8 (,Halt-Bit') dient zum Starten und Stoppen des Motors. Ist der Wert auf ,1', wird der Motor angehalten. Bei einem Übergang von ,1' auf ,0' fängt der Motor an zu drehen und das Drehmoment wird entsprechend SDO 0x6071 eingestellt.

Bit 4, 6, 7, 9-15 werden im Torque Mode nicht verwendet und sind auf 0 zu setzen.

### 8.3 Statusword (SDO 0x6041)

Bezeichnung	Statusword
SDO-ID	0x6041
Typ	u16
Wertebereich	Bits

Bit 10 dient zur Statusanzeige des Drehmoments.

Bit 10 Statusword	Bit 8 Controlword	Bedeutung
0	0	Vorgegebenes Drehmoment nicht erreicht
0	1	Motor bremst
1	0	Vorgegebenes Drehmoment erreicht
1	1	Motor steht

Bit 0-9, 11-15 siehe allgemeine Beschreibung.

### 8.4 Target Torque (SDO 0x6071)

Bezeichnung	Target Torque
SDO-ID	0x6071
Typ	s16
Wertebereich	-1000 bis 1000

Dieses Objekt enthält den Sollwert für das einzustellende Drehmoment. Das Drehmoment ist direkt proportional zum Strom, weshalb der Wert in Tausendstel vom maximal eingestellten Strom vorgegeben wird.

Bsp.:

- Steuerung SMCI47-S mit Nennstrom 7,5 A
- SDO 0x2004 Sub 1 auf 20 (20% von 7,5 A)
- SDO 0x6071 auf 500

→  $500 / 1000 * 0,2 * 7,5A = 0,75 A$

### 8.5 Max Profile Velocity (SDO 0x607F)

Bezeichnung	Max Profile Velocity
SDO-ID	0x607F
Typ	u32
Wertebereich	u32

Dieses Objekt enthält die maximal zulässige Drehzahl als Betrag für beide Drehrichtungen, die sich im Torque Modus einstellen lassen.

Der Wert wird in RPM angegeben.

## 9 Nicht Moduspezifische SDOs

### 9.1 Digital-Eingänge (SDO 0x60FD)

#### Objektbeschreibung

<b>Bezeichnung</b>	Digital Inputs
<b>SDO-ID</b>	0x60FD
<b>Typ</b>	u32
<b>Wertebereich</b>	32 Bit Bitmaske

Mit diesem SDO können die Digital-Eingänge des Motors gelesen werden.

Dieses SDO ist nur lesbar.

#### Bits

Bit 0 bis 1: unbelegt („0“)

Bit 2: Eingang 6 (Referenzschalter)

Bit 3 bis 15: unbelegt („0“)

Bit 16 bis 21: Eingang 1 bis Eingang 6

Bit 22 bis 31: unbelegt („0“).

### 9.2 Digital-Ausgänge (SDO 0x60FE Subindex 1)

#### Objektbeschreibung

<b>Bezeichnung</b>	Digital Outputs
<b>SDO-ID</b>	0x60FE
<b>Subindex</b>	1
<b>Typ</b>	u32
<b>Wertebereich</b>	32 Bit Bitmaske

Mit diesem SDO können die Digital-Ausgänge des Motors geschrieben werden.

Das Schreiben der Ausgänge über dieses SDO ist nur dann möglich, wenn diese auch für den Benutzer mit der Ausgangs-Maske (SDO 0x60FE Subindex 2) freigegeben sind. Andernfalls geben die Ausgänge den Motor-Status an (siehe auch Handbuch).

#### Bits

Bit 0 bis 15 unbelegt („0“)

Bit 16 bis 18: Ausgang 1 bis Ausgang 3

Bit 19 bis 31: unbelegt („0“).

## 9.3 Digital-Ausgänge Bitmaske (SDO 0x60FE Subindex 2)

### Objektbeschreibung

<b>Bezeichnung</b>	Bitmask
<b>SDO-ID</b>	0x60FE
<b>Subindex</b>	2
<b>Typ</b>	u32
<b>Wertebereich</b>	32 Bit Bitmaske

Mit diesem SDO können die Digital-Ausgänge des Motors für den Nutzer reserviert werden.

Eine „0“ bedeutet, dass der jeweilige Ausgang für den Nutzer reserviert ist und somit mit SDO 0x60FE Subindex 1 geändert werden kann. Eine „1“ in der Maske bedeutet, dass der Status der Firmware Einfluss auf die Ausgänge nimmt. Diese sind dann nicht mit SDO 0x60FE Subindex 1 beeinflussbar.

### Bits

Bit 0 bis 15 unbelegt („0“)

Bit 16 bis 18: Maske für Ausgang 1 bis Ausgang 3

Bit 19 bis 31: unbelegt („0“)

Werden unbelegte Bits mit „1“ beschrieben, wird die gesamte Maske verworfen.

## 9.4 Polarity (0x607E)

### Objektbeschreibung

<b>Bezeichnung</b>	Polarity
<b>SDO-ID</b>	0x607E
<b>Typ</b>	u8
<b>Wertebereich</b>	8 Bit Bitmaske, Bit 6 und 7 verwendbar, Bit 0 bis 5 reserviert (gleich 0)

### Bits

Bit 6: Umkehr der Drehrichtung im Velocity Modus.

Bit 7: Umkehr der Drehrichtung im Profile Position Modus.

(„1“ bedeutet Umkehrung ist aktiviert, „0“ bedeutet Drehrichtung wie im jeweiligen Modus beschrieben).

## 9.5 Producer Heartbeat Time (0x1017)

### Objektbeschreibung

<b>Bezeichnung</b>	Producer Heartbeat Time
<b>SDO-ID</b>	0x1017
<b>Typ</b>	u16
<b>Wertebereich</b>	u16

Stellt die Heartbeat Time in ms (Millisekunden) ein.

Defaulteinstellung ist 0 (deaktiviert).

Ist der Wert auf 0 gesetzt, dann ist der Heartbeat ausgeschaltet.

Bei Werten ungleich 0 wird im eingestellten Intervall auf der COB-ID 0x700 + Node-ID der CANopen Netzwerkstatus gesendet.

**Achtung:**

Durch Einstellen kleiner Intervalle kann der CAN-Bus stark ausgelastet werden, so dass anderweitige Kommunikation kaum noch möglich ist.

### Netzwerkstatus anfordern

Ist der Heartbeat ausgeschaltet, kann der Netzwerkstatus durch Senden eines Remote Transmission Request auf der COB-ID 0x700 + Node-ID angefordert werden.

Der Netzwerkstatus ist ein u8 und kann folgende Werte annehmen:

- 0x7F: Pre-Operational (Status nach Anlegen der Betriebsspannung)
- 0x04: Node Disabled
- 0x05: Node Operational.

## 9.6 DEVICE TYPE (0x1000)

### Objektbeschreibung

<b>Bezeichnung</b>	Device Type
<b>SDO-ID</b>	0x1000
<b>Typ</b>	u16, ro
<b>Wertebereich</b>	u16

Beschreibt den Steuerungstyp. Hier: 0x00008 (Schrittmotor)

## 9.7 ERROR REGISTER (0x1001)

### Objektbeschreibung

<b>Bezeichnung</b>	Error Register
<b>SDO-ID</b>	0x1001
<b>Typ</b>	u8
<b>Wertebereich</b>	u8, ro

Fehlerregister: Im Fehlerfall wird das entsprechende Fehlerbit gesetzt.

Bit 0: generic error

Bit 1: current (0)

Bit 2: voltage (0)

Bit 3: temperature (0)

Bit 4: communication error (overrun, error state)

Bit 5: device profile specific (0)

Bit 6: reserved (0)

Bit 7: manufacturer specific (0)

## 9.8 COB\_ID\_EMERGENCY\_MESSAGE (0x1014)

### Objektbeschreibung

<b>Bezeichnung</b>	Emergency Message
<b>SDO-ID</b>	0x1014
<b>Typ</b>	u32, ro
<b>Wertebereich</b>	u32

Default Value 80h + Node-ID

### Bits

Bit 31 (MSB): 0 = EMCY exists, 1 = EMCY does not exist

Bit 30: 0 = reserved (always 0)

Bit 29: 0 = 11-Bit-ID (CAN 2.0A), 1 = 29-Bit-ID (CAN 2.0B)

Bit 28-11: 0 = wenn Bit 29=0; Bits 28-11 der 29-Bit-COB-ID wenn Bit 29=1:

Bit 10-0 (LSB): Bits 10-0 der COB-ID

## 9.9 IDENTITY\_OBJECT\_VENDOR\_ID (0x1018)

### Objektbeschreibung

<b>Bezeichnung</b>	Identity Object
<b>SDO-ID</b>	0x1018
<b>Typ</b>	record

#### Sub-Index 0h: Anzahl der Einträge

<b>Typ</b>	u8
<b>Wertebereich</b>	1 ... 4, ro

#### Sub-Index 1h: Vendor ID

<b>Typ</b>	u32, ro
------------	---------

#### Sub-Index 2h: Product code

<b>Typ</b>	u32, ro
------------	---------

#### Sub-Index 3h: Revision number

<b>Typ</b>	u32, ro
------------	---------

#### Sub-Index 4h: Serial number

<b>Typ</b>	u32, ro
------------	---------

## 9.10 RPDO1\_COMMUNICATION\_PARAMETER (0x1400)

### Objektbeschreibung

<b>Bezeichnung</b>	Receive PDO1 Communication Parameter
<b>SDO-ID</b>	0x1400
<b>Typ</b>	record
<b>Anzahl Einträge</b>	2

Enthält die Kommunikationsparameter für das RPDO 1.

### Sub-Index 01h: COB-ID

<b>Typ</b>	u32, rw
<b>Defaultwert</b>	0x200+Node-ID Der Defaultwert wird nicht durch Ändern der Node-ID wirksam, sondern nur durch einen EEPROM-Reset (siehe Abschnitt 1.1.3).

#### Bits

Bit 31: 1 = PDO deaktiviert

Bit 30: 1 = RTR nicht erlaubt

Bit 29: 1 = 29-Bit-ID, 0 = 11-Bit-ID

Bit 28-11: Bit 28-11 der 29-bit ID, sonst 0

Bit 10-0: Bits 10-0 der COB-ID

### Sub-Index 02h: Transmission Type

<b>Typ</b>	u8, rw
<b>Defaultwert</b>	255

#### Werte

0 bis 240: Update des Objekts nach Empfangen des nächsten Sync-Objekts.

255: Sofortiges Update des Objekts.

## 9.11 RPDO2\_COMMUNICATION\_PARAMETER (0x1401)

### Objektbeschreibung

<b>Bezeichnung</b>	Receive PDO2 Communication Parameter
<b>SDO-ID</b>	0x1401
<b>Typ</b>	record
<b>Anzahl Einträge</b>	2

Enthält die Kommunikationsparameter für das RPDO 2.

**Sub-Index 01h: COB-ID**

<b>Typ</b>	u32, rw
<b>Defaultwert</b>	0x300 + Node-ID Der Defaultwert wird nicht durch Ändern der Node-ID wirksam, sondern nur durch einen EEPROM-Reset (siehe Abschnitt 1.1.3).

**Bits**

- Bit 31: 1 = PDO deaktiviert
- Bit 30: 1 = RTR nicht erlaubt
- Bit 29: 1 = 29-Bit-ID, 0 = 11-Bit-ID
- Bit 28-11: Bit 28-11 der 29-bit ID, sonst 0
- Bit 10-0: Bits 10-0 der COB-ID

**Sub-Index 02h: Transmission Type**

<b>Typ</b>	u8, rw
<b>Defaultwert</b>	255

**Werte**

- 0 bis 240: Update des Objekts nach Empfangen des nächsten Sync-Objekts.
- 255: Sofortiges Update des Objekts.

## 9.12 RPDO3\_COMMUNICATION\_PARAMETER (0x1402)

**Objektbeschreibung**

<b>Bezeichnung</b>	Receive PDO3 Communication Parameter
<b>SDO-ID</b>	0x1403
<b>Typ</b>	record
<b>Anzahl Einträge</b>	2

Enthält die Kommunikationsparameter für das RPDO 3.

**Sub-Index 01h: COB-ID**

<b>Typ</b>	u32, rw
<b>Defaultwert</b>	0x400 + Node-ID Der Defaultwert wird nicht durch Ändern der Node-ID wirksam, sondern nur durch einen EEPROM-Reset (siehe Abschnitt 1.1.3).

**Bits**

- Bit 31: 1 = PDO deaktiviert
- Bit 30: 1 = RTR nicht erlaubt
- Bit 29: 1 = 29-Bit-ID, 0 = 11-Bit-ID
- Bit 28-11: Bit 28-11 der 29-bit ID, sonst 0
- Bit 10-0: Bits 10-0 der COB-ID

#### Sub-Index 02h: Transmission Type

<b>Typ</b>	u8, rw
<b>Defaultwert</b>	255

##### Werte

0 bis 240: Update des Objekts nach Empfangen des nächsten Sync-Objekts.

255: Sofortiges Update des Objekts.

## 9.13 RPDO4\_COMMUNICATION\_PARAMETER (0x1403) #

#### Objektbeschreibung

<b>Bezeichnung</b>	Receive PDO4 Communication Parameter
<b>SDO-ID</b>	0x1403
<b>Typ</b>	record
<b>Anzahl Einträge</b>	2

Enthält die Kommunikationsparameter für das RPDO 4.

#### Sub-Index 01h: COB-ID

<b>Typ</b>	u32, rw
<b>Defaultwert</b>	0x500 + Node-ID Der Defaultwert wird nicht durch Ändern der Node-ID wirksam, sondern nur durch einen EEPROM-Reset (siehe Abschnitt 1.1.3).

##### Bits

Bit 31: 1 = PDO deaktiviert

Bit 30: 1 = RTR nicht erlaubt

Bit 29: 1 = 29-Bit-ID, 0 = 11-Bit-ID

Bit 28-11: Bit 28-11 der 29-bit ID, sonst 0

Bit 10-0: Bits 10-0 der COB-ID

#### Sub-Index 02h: Transmission Type

<b>Typ</b>	u8, rw
<b>Defaultwert</b>	255

##### Werte

0 bis 240: Update des Objekts nach Empfangen des nächsten Sync-Objekts.

255: Sofortiges Update des Objekts.

## 9.14 RPDO1\_MAPPING (0x1600)

### Objektbeschreibung

<b>Bezeichnung</b>	Receive PDO Mapping Parameter
<b>SDO-ID</b>	0x1600
<b>Typ</b>	record
<b>Anzahl gemappter Objekte</b>	0-8

Beschreibt, welches Objekt auf die im Objekt 0x1400 definierte COB-ID gemappt wird.

### Subindex 1-8: gemappte Objekte

<b>Bezeichnung</b>	Receive PDO1 Mapping Parameter
<b>Typ</b>	u32, rw

Beschreibt jeweils ein gemapptes Objekt.

#### Bits

Bit 31-16: index

Bit 15-8: sub-index

Bit 7-0: Objektlänge in Bit

## 9.15 RPDO2\_MAPPING (0x1601)

### Objektbeschreibung

<b>Bezeichnung</b>	Receive PDO2 Mapping Parameter
<b>SDO-ID</b>	0x1601
<b>Typ</b>	record
<b>Anzahl gemappter Objekte</b>	0-8

Beschreibt, welches Objekt auf die im Objekt 0x1401 definierte COB-ID gemappt wird.

### Subindex 1-8: gemappte Objekte

<b>Bezeichnung</b>	Receive PDO Mapping Parameter
<b>Typ</b>	u32, rw

Beschreibt jeweils ein gemapptes Objekt.

#### Bits

Bit 31-16: index

Bit 15-8: sub-index

Bit 7-0: Objektlänge in Bit

## 9.16 RPDO3\_MAPPING (0x1602)

### Objektbeschreibung

<b>Bezeichnung</b>	Receive PDO3 Mapping Parameter
<b>SDO-ID</b>	0x1602
<b>Typ</b>	record
<b>Anzahl gemappter Objekte</b>	0-8

Beschreibt, welches Objekt auf die im Objekt 0x1402 definierte COB-ID gemappt wird.

### Subindex 1-8: gemappte Objekte

<b>Bezeichnung</b>	Receive PDO Mapping Parameter
<b>Typ</b>	u32, rw

Beschreibt jeweils ein gemapptes Objekt.

#### Bits

Bit 31-16: index

Bit 15-8: sub-index

Bit 7-0: Objektlänge in Bit

## 9.17 RPDO4\_MAPPING (0x1603)

### Objektbeschreibung

<b>Bezeichnung</b>	Receive PDO4 Mapping Parameter
<b>SDO-ID</b>	0x1603
<b>Typ</b>	record
<b>Anzahl gemappter Objekte</b>	0-8

Beschreibt, welches Objekt auf die im Objekt 0x1403 definierte COB-ID gemappt wird.

### Subindex 1-8: gemappte Objekte

<b>Bezeichnung</b>	Receive PDO Mapping Parameter
<b>Typ</b>	u32, rw

Beschreibt jeweils ein gemapptes Objekt.

#### Bits

Bit 31-16: index

Bit 15-8: sub-index

Bit 7-0: Objektlänge in Bit

## 9.18 TPDO1\_COMMUNICATION\_PARAMETER (0x1800)

### Objektbeschreibung

<b>Bezeichnung</b>	Transmit PDO1 Communication Parameter
<b>SDO-ID</b>	0x1800
<b>Typ</b>	record
<b>Anzahl Einträge</b>	5

Enthält die Kommunikationsparameter für das TPDO 1.

### Sub-Index 01h: COB-ID

<b>Typ</b>	u32, rw
<b>Defaultwert</b>	0x180 + Node-ID Der Defaultwert wird nicht durch Ändern der Node-ID wirksam, sondern nur durch einen EEPROM-Reset (siehe Abschnitt 1.1.3).

#### Bits

Bit 31: 1 = PDO deaktiviert

Bit 30: 1 = RTR nicht erlaubt

Bit 29: 1 = 29-Bit-ID, 0 = 11-Bit-ID

Bit 28-11: Bit 28-11 der 29-bit ID, sonst 0

Bit 10-0: Bits 10-0 der COB-ID

### Sub-Index 02h: Transmission Type

<b>Typ</b>	u8, rw
<b>Defaultwert</b>	255

#### Werte

0: Senden des Objekts nach dem nächsten Sync-Objekt, wenn es seinen Wert geändert hat.

1 bis 240: Senden des Objekts nach der eingestellten Anzahl von Sync-Objekten (1 bis 240), unabhängig davon, ob sich das Objekt geändert hat.

255: Senden des Objektes nach einer Änderung, unabhängig vom Sync-Objekt.

### Sub-Index 03h: inhibit time

<b>Typ</b>	u16, rw
<b>Defaultwert</b>	1000

Bei Verwendung des Transmission Types 255 (Asynchrones Senden bei Änderung) gibt dieser Wert die minimale Zeit zwischen dem Senden zweier zeitlich benachbarter Objekte in 100µs-Schritten an. Somit kann verhindert werden, dass beispielsweise die Istposition, die sich während der Fahrt kontinuierlich ändert, den CAN-Bus blockiert.

### Sub-Index 04h: reserved

<b>Typ</b>	u8, rw
<b>Defaultwert</b>	0

Keine Bedeutung.

**Sub-Index 05h: event timer**

<b>Typ</b>	u16, rw
<b>Defaultwert</b>	0

Gibt beim Transmission Type 255 (Asynchrones Senden bei Änderung) den maximalen Zeitabstand zwischen zwei gesendeten Objekten gleichen Types an. Mit dieser Einstellung können Objekte, die sich selten ändern, zusätzlich noch zyklisch gesendet werden. Ein Wert von „0“ in dieser Einstellung deaktiviert dieses Verhalten (Default).

## 9.19 TPDO2\_COMMUNICATION\_PARAMETER (0x1801)

**Objektbeschreibung**

<b>Bezeichnung</b>	Transmit PDO2 Communication Parameter
<b>SDO-ID</b>	0x1801
<b>Typ</b>	record
<b>Anzahl Einträge</b>	5

Enthält die Kommunikationsparameter für das TPDO 2.

**Sub-Index 01h: COB-ID**

<b>Typ</b>	u32, rw
<b>Defaultwert</b>	0x280 + Node-ID Der Defaultwert wird nicht durch Ändern der Node-ID wirksam, sondern nur durch einen EEPROM-Reset (siehe Abschnitt 1.1.3).

**Bits**

- Bit 31: 1 = PDO deaktiviert
- Bit 30: 1 = RTR nicht erlaubt
- Bit 29: 1 = 29-Bit-ID, 0 = 11-Bit-ID
- Bit 28-11: Bit 28-11 der 29-bit ID, sonst 0
- Bit 10-0: Bits 10-0 der COB-ID

**Sub-Index 02h: Transmission Type**

<b>Typ</b>	u8, rw
<b>Defaultwert</b>	255

**Werte**

- 0: Senden des Objekts nach dem nächsten Sync-Objekt, wenn es seinen Wert geändert hat.
- 1 bis 240: Senden des Objekts nach der eingestellten Anzahl von Sync-Objekten (1 bis 240), unabhängig davon, ob sich das Objekt geändert hat.
- 255: Senden des Objektes nach einer Änderung, unabhängig vom Sync-Objekt.

**Sub-Index 03h: inhibit time**

<b>Typ</b>	u16, rw
<b>Defaultwert</b>	1000

Bei Verwendung des Transmission Types 255 (Asynchrones Senden bei Änderung) gibt dieser Wert die minimale Zeit zwischen dem Senden zweier zeitlich benachbarter Objekte in 100µs-Schritten an. Somit kann verhindert werden, dass beispielsweise die Istposition, die sich während der Fahrt kontinuierlich ändert, den CAN-Bus blockiert.

**Sub-Index 04h: reserved**

<b>Typ</b>	u8, rw
<b>Defaultwert</b>	0

Keine Bedeutung

**Sub-Index 05h: event timer**

<b>Typ</b>	u16, rw
<b>Defaultwert</b>	0

Gibt beim Transmission Type 255 (Asynchrones Senden bei Änderung) den maximalen Zeitabstand zwischen zwei gesendeten Objekten gleichen Types an. Mit dieser Einstellung können Objekte, die sich selten ändern, zusätzlich noch zyklisch gesendet werden. Ein Wert von „0“ in dieser Einstellung deaktiviert dieses Verhalten (Default).

## 9.20 TPDO3\_COMMUNICATION\_PARAMETER (0x1802)

**Objektbeschreibung**

<b>Bezeichnung</b>	Transmit PDO3 Communication Parameter
<b>SDO-ID</b>	0x1802
<b>Typ</b>	record
<b>Anzahl Einträge</b>	5

Enthält die Kommunikationsparameter für das TPDO 3.

**Sub-Index 01h: COB-ID**

<b>Typ</b>	u32, rw
<b>Defaultwert</b>	0x380 + Node-ID Der Defaultwert wird nicht durch Ändern der Node-ID wirksam, sondern nur durch einen EEPROM-Reset (siehe Abschnitt 1.1.3).

**Bits**

Bit 31: 1 = PDO deaktiviert

Bit 30: 1 = RTR nicht erlaubt

Bit 29: 1 = 29-Bit-ID, 0 = 11-Bit-ID

Bit 28-11: Bit 28-11 der 29-bit ID, sonst 0

Bit 10-0: Bits 10-0 der COB-ID

**Sub-Index 02h: Transmission Type**

<b>Typ</b>	u8, rw
<b>Defaultwert</b>	255

**Werte**

0: Senden des Objekts nach dem nächsten Sync-Objekt, wenn es seinen Wert geändert hat.

1 bis 240: Senden des Objekts nach der eingestellten Anzahl von Sync-Objekten (1 bis 240), unabhängig davon, ob sich das Objekt geändert hat.

255: Senden des Objektes nach einer Änderung, unabhängig vom Sync-Objekt.

**Sub-Index 03h: inhibit time**

<b>Typ</b>	u16, rw
<b>Defaultwert</b>	1000

Bei Verwendung des Transmission Types 255 (Asynchrones Senden bei Änderung) gibt dieser Wert die minimale Zeit zwischen dem Senden zweier zeitlich benachbarter Objekte in 100µs-Schritten an. Somit kann verhindert werden, dass beispielsweise die Istposition, die sich während der Fahrt kontinuierlich ändert, den CAN-Bus blockiert.

**Sub-Index 04h: reserved**

<b>Typ</b>	u8, rw
<b>Defaultwert</b>	0

Keine Bedeutung.

**Sub-Index 05h: event timer**

<b>Typ</b>	u16, rw
<b>Defaultwert</b>	0

Gibt beim Transmission Type 255 (Asynchrones Senden bei Änderung) den maximalen Zeitabstand zwischen zwei gesendeten Objekten gleichen Types an. Mit dieser Einstellung können Objekte, die sich selten ändern, zusätzlich noch zyklisch gesendet werden. Ein Wert von „0“ in dieser Einstellung deaktiviert dieses Verhalten (Default).

## 9.21 TPDO4\_COMMUNICATION\_PARAMETER (0x1803)

**Objektbeschreibung**

<b>Bezeichnung</b>	Transmit PDO4 Communication Parameter
<b>SDO-ID</b>	0x1803
<b>Typ</b>	record
<b>Anzahl Einträge</b>	5

Enthält die Kommunikationsparameter für das TPDO 4.

**Sub-Index 01h: COB-ID**

<b>Typ</b>	u32, rw
<b>Defaultwert</b>	0x480 + Node-ID Der Defaultwert wird nicht durch Ändern der Node-ID wirksam, sondern nur durch einen EEPROM-Reset (siehe Abschnitt 1.1.3).

**Bits**

Bit 31: 1 = PDO deaktiviert

Bit 30: 1 = RTR nicht erlaubt

Bit 29: 1 = 29-Bit-ID, 0 = 11-Bit-ID

Bit 28-11: Bit 28-11 der 29-bit ID, sonst 0

Bit 10-0: Bits 10-0 der COB-ID

**Sub-Index 02h: Transmission Type**

<b>Typ</b>	u8, rw
<b>Defaultwert</b>	255

**Werte**

0: Senden des Objekts nach dem nächsten Sync-Objekt, wenn es seinen Wert geändert hat.

1 bis 240: Senden des Objekts nach der eingestellten Anzahl von Sync-Objekten (1 bis 240), unabhängig davon, ob sich das Objekt geändert hat.

255: Senden des Objektes nach einer Änderung, unabhängig vom Sync-Objekt.

**Sub-Index 03h: inhibit time**

<b>Typ</b>	u16, rw
<b>Defaultwert</b>	1000

Bei Verwendung des Transmission Types 255 (Asynchrones Senden bei Änderung) gibt dieser Wert die minimale Zeit zwischen dem Senden zweier zeitlich benachbarter Objekte in 100µs-Schritten an. Somit kann verhindert werden, dass beispielsweise die Istposition, die sich während der Fahrt kontinuierlich ändert, den CAN-Bus blockiert.

**Sub-Index 04h: reserved**

<b>Typ</b>	u8, rw
<b>Defaultwert</b>	0

Keine Bedeutung

**Sub-Index 05h: event timer**

<b>Typ</b>	u16, rw
<b>Defaultwert</b>	0

Gibt beim Transmission Type 255 (Asynchrones Senden bei Änderung) den maximalen Zeitabstand zwischen zwei gesendeten Objekten gleichen Types an. Mit dieser Einstellung können Objekte, die sich selten ändern, zusätzlich noch zyklisch gesendet werden. Ein Wert von „0“ in dieser Einstellung deaktiviert dieses Verhalten (Default).

## 9.22 TPDO1\_MAPPING (0x1A00)

### Objektbeschreibung

<b>Bezeichnung</b>	Transmit PDO1 Mapping Parameter
<b>SDO-ID</b>	0x1A00
<b>Typ</b>	record
<b>Anzahl gemappter Objekte</b>	0-8

Beschreibt, welches Objekt auf die im Objekt 0x1800 definierte COB-ID gemappt wird.

### Subindex 1-8: gemappte Objekte

<b>Bezeichnung</b>	Transmit PDO Mapping Parameter
<b>Typ</b>	u32, rw

Beschreibt jeweils ein gemapptes Objekt.

#### Bits

31-16: index

15-8: sub-index

7-0: Objektlänge in Bit

## 9.23 TPDO2\_MAPPING (0x1A01)

### Objektbeschreibung

<b>Bezeichnung</b>	Transmit PDO2 Mapping Parameter
<b>SDO-ID</b>	0x1A01
<b>Typ</b>	record
<b>Anzahl gemappter Objekte</b>	0-8

Beschreibt, welches Objekt auf die im Objekt 0x1801 definierte COB-ID gemappt wird.

### Subindex 1-8: gemappte Objekte

<b>Bezeichnung</b>	Transmit PDO Mapping Parameter
<b>Typ</b>	u32, rw

Beschreibt jeweils ein gemapptes Objekt.

#### Bits

31-16: index

15-8: sub-index

7-0: Objektlänge in Bit

## 9.24 TPDO3\_MAPPING (0x1A02)

### Objektbeschreibung

<b>Bezeichnung</b>	Transmit PDO2 Mapping Parameter
<b>SDO-ID</b>	0x1A02
<b>Typ</b>	record
<b>Anzahl gemappter Objekte</b>	0-8

Beschreibt, welches Objekt auf die im Objekt 0x1802 definierte COB-ID gemappt wird.

### Subindex 1-8: gemappte Objekte

<b>Bezeichnung</b>	Transmit PDO Mapping Parameter
<b>Typ</b>	u32, rw

Beschreibt jeweils ein gemapptes Objekt.

#### Bits

31-16: index

15-8: sub-index

7-0: Objektlänge in Bit

## 9.25 TPDO4\_MAPPING (0x1A03)

### Objektbeschreibung

<b>Bezeichnung</b>	Transmit PDO4 Mapping Parameter
<b>SDO-ID</b>	0x1A03
<b>Typ</b>	record
<b>Anzahl gemappter Objekte</b>	0-8

Beschreibt, welches Objekt auf die im Objekt 0x1803 definierte COB-ID gemappt wird.

### Subindex 1-8: gemappte Objekte

<b>Bezeichnung</b>	Transmit PDO Mapping Parameter
<b>Typ</b>	u32, rw

Beschreibt jeweils ein gemapptes Objekt.

#### Bits

31-16: index

15-8: sub-index

7-0: Objektlänge in Bit

## 9.26 ERROR\_CODE (0x603F)

### Objektbeschreibung

<b>Bezeichnung</b>	Error code
<b>SDO-ID</b>	0x603F
<b>Typ</b>	u16, ro
<b>Wertebereich</b>	u16

Enthält den Error-Code des zuletzt aufgetretenen Fehlers.

## 9.27 VL\_DIMENSION\_FACTOR (0x604C)

### Objektbeschreibung

<b>Bezeichnung</b>	vl dimension factor
<b>SDO-ID</b>	0x604C
<b>Typ</b>	record
<b>Anzahl Einträge</b>	2

Umrechnungsfaktor der benutzerspezifischen Geschwindigkeit (z.B. 1/s, Inkremente/s, m/s,...) in Umdrehungen pro Minute.

Geschwindigkeit [upm] = Geschwindigkeit [benutzerspezifische Einheit] /  
Umrechnungsfaktor [upm/benutzerspezifische Einheit] .

#### Sub-Index 01h: vl dimension factor numerator

<b>Typ</b>	u32, rw
<b>Defaultwert</b>	1

#### Sub-Index 02h: vl dimension factor denominator

<b>Typ</b>	u32, rw
<b>Defaultwert</b>	1

## 9.28 MODES\_OF\_OPERATION (0x6060)

### Objektbeschreibung

<b>Bezeichnung</b>	Modes of operation
<b>SDO-ID</b>	0x6060
<b>Typ</b>	s8, rw
<b>Wertebereich / Default</b>	-128 - 10 / 0

Enthält den aktuellen Betriebsmodus.

## Werte

Modus	Beschreibung
-128 to -1	Manufacturer-specific operation modes
-2	Kurzer Closed-Loop Testlauf (Alignment)
-1	Closed-Loop Testlauf
0	No mode change/no mode assigned
+1	Profile Position Mode
+2	Velocity Mode
+3	Profile Velocity Mode
+4	Torque Profile Mode
+5	Reserved
+6	Homing Mode
+7	Interpolated Position Mode
+11 bis +127	Reserved

## 9.29 MODES\_OF\_OPERATION\_DISPLAY (0x6061)

### Objektbeschreibung

<b>Bezeichnung</b>	Modes of operation display
<b>SDO-ID</b>	6061
<b>Typ</b>	s8, ro
<b>Wertebereich</b>	s8

Enthält den aktuellen Betriebsmodus. -> siehe MODES\_OF\_OPERATION (SDO 0x6060)

## 9.30 POSITION\_ACTUAL\_INTERNAL\_VALUE (0x6063)

<b>Bezeichnung</b>	Position actual internal value
<b>SDO-ID</b>	0x6063
<b>Typ</b>	s32, ro
<b>Wertebereich</b>	s32

Enthält die aktuelle Drehgeberposition.

## 9.31 POSITION\_ACTUAL\_VALUE (0x6064)

### Objektbeschreibung

<b>Bezeichnung</b>	Position actual value
<b>SDO-ID</b>	0x6064
<b>Typ</b>	s32, ro
<b>Wertebereich</b>	s32

Enthält die aktuelle Istposition (Drehgeberposition umgerechnet laut Feed Constant und Gear Ratio)

## 9.32 FOLLOWING\_ERROR\_WINDOW (0x6065)

### Objektbeschreibung

<b>Bezeichnung</b>	Following error window
<b>SDO-ID</b>	0x6065
<b>Typ</b>	U32, rw
<b>Wertebereich</b>	u32

Gibt den maximalen Schleppfehler symmetrisch zur Sollposition an. Weicht die Istposition zu stark von der Sollposition ab, wird ein following error erzeugt.

## 9.33 FOLLOWING\_ERROR\_TIME\_OUT (0x6066)

### Objektbeschreibung

<b>Bezeichnung</b>	Following error timeout
<b>SDO-ID</b>	0x6066
<b>Typ</b>	u16, rw
<b>Wertebereich</b>	u16

Zeit in Millisekunden bis ein zu großer Schleppfehler zu einer Fehlermeldung führt.

## 9.34 POSITION\_WINDOW (0x6067)

### Objektbeschreibung

<b>Bezeichnung</b>	Position window
<b>SDO-ID</b>	0x6067
<b>Typ</b>	u32, rw
<b>Wertebereich</b>	u32

Gibt relativ zur Zielposition einen symmetrischen Bereich an, innerhalb dem das Ziel als erreicht gilt.

## 9.35 POSITION\_WINDOW\_TIME (0x6068)

### Objektbeschreibung

<b>Bezeichnung</b>	Position window time
<b>SDO-ID</b>	0x6068
<b>Typ</b>	u16, rw
<b>Wertebereich</b>	u16

Die Istposition muss sich für diese Zeit innerhalb des position window befinden, damit die Zielposition als erreicht gilt.

## 9.36 POS\_RANGE\_LIMIT (0x607B)

### Objektbeschreibung

<b>Bezeichnung</b>	Position range limit
<b>SDO-ID</b>	0x607B
<b>Typ</b>	record
<b>Anzahl Einträge</b>	2

Enthält die Minimal- und Maximalposition. Wird dieser Bereich über- oder unterschritten, erfolgt ein Überlauf. Um diesen Überlauf zu verhindern siehe auch SW\_POS\_LIMIT (SDO 0x607D).

#### Sub-Index 01h: Min position range limit

<b>Typ</b>	s32, rw
<b>Defaultwert</b>	-100000000

#### Sub-Index 02h: Max position range limit

<b>Typ</b>	s32, rw
<b>Defaultwert</b>	100000000

## 9.37 SW\_POS\_LIMIT (0x607D)

<b>Bezeichnung</b>	Software position limit
<b>SDO-ID</b>	0x607D
<b>Typ</b>	record
<b>Anzahl Einträge</b>	2

Die Zielposition muss innerhalb der hier gesetzten Grenzen liegen. Vor der Überprüfung wird jeweils der home offset (SDO 0x607C) abgezogen:

corrected min position limit = min position limit - home offset

corrected max position limit = max position limit - home offset.

**Sub-Index 01h: Min position limit**

<b>Typ</b>	s32, rw
<b>Defaultwert</b>	-100000000

**Sub-Index 02h: Max position limit**

<b>Typ</b>	s32, rw
<b>Defaultwert</b>	100000000

## 9.38 POSITION\_ENCODER\_RESOLUTION (0x608F)

**Objektbeschreibung**

<b>Bezeichnung</b>	Position encoder resolution
<b>SDO-ID</b>	0x608F
<b>Typ</b>	record
<b>Anzahl Einträge</b>	2

Encoder Inkremente pro Umdrehung:

position encoder resolution = encoder increments / motor revolutions.

**Sub-Index 01h: encoder increments**

<b>Typ</b>	u32, rw
<b>Defaultwert</b>	2000

**Sub-Index 02h: motor revolutions**

<b>Typ</b>	u32, rw
<b>Defaultwert</b>	1

## 9.39 GEAR\_RATIO (0x6091)

**Objektbeschreibung**

<b>Bezeichnung</b>	Gear ratio
<b>SDO-ID</b>	0x6091
<b>Typ</b>	record
<b>Anzahl Einträge</b>	2

Anzahl der Motorumdrehungen pro Umdrehung der Abtriebsachse:

gear ratio = motor shaft revolutions / driving shaft revolutions.

**Sub-Index 01h: Motor revolutions**

<b>Typ</b>	u32, rw
<b>Defaultwert</b>	1

**Sub-Index 02h: Shaft revolutions**

<b>Typ</b>	u32, rw
<b>Defaultwert</b>	1

## 9.40 FEED\_CONSTANT (0x6092)

**Objektbeschreibung**

<b>Bezeichnung</b>	Feed constant
<b>SDO-ID</b>	0x6092
<b>Typ</b>	record
<b>Anzahl Einträge</b>	2

Gibt im Falle eines Linearantriebes den Vorschub pro Umdrehung an:

Feed constant = feed / revolutions.

**Sub-Index 01h: Feed**

<b>Typ</b>	u32, rw
<b>Defaultwert</b>	2000

**Sub-Index 02h: Shaft revolutions**

<b>Typ</b>	u32, rw
<b>Defaultwert</b>	1

## 9.41 MAX\_ACCEL (0x60C5)

**Objektbeschreibung**

<b>Bezeichnung</b>	Max acceleration
<b>SDO-ID</b>	0x60C5
<b>Typ</b>	u32, rw
<b>Wertebereich</b>	u32

Enthält die maximal zulässige Beschleunigungsrampe. → siehe auch MAX\_DECEL(SDO 0x60C6).

## 9.42 MAX\_DECEL (0x60C6)

### Objektbeschreibung

<b>Bezeichnung</b>	Max deceleration
<b>SDO-ID</b>	0x60C6
<b>Typ</b>	u32, rw
<b>Wertebereich</b>	u32

Enthält die maximale Bremsrampe. → siehe auch MAX\_ACCEL (0x60C5).

## 9.43 SUPPORTED\_DRIVE\_MODES (0x6502)

### Objektbeschreibung

<b>Bezeichnung</b>	Supported drive modes
<b>SDO-ID</b>	0x6502
<b>Typ</b>	record, ro
<b>Wertebereich</b>	u32

Das Objekt beschreibt die unterstützten drive modi. Derzeit werden im CANopen Betrieb nur PP, HM, VL, IP und TQ unterstützt.

### Bits

Bit	Bedeutung	unterstützt
31-16	not used (manufacturerspecific)	0
15-20	reserved	0
9	CST	0
8	CSV	0
7	CSP	0
6	IP (Interpolated Position Mode)	1 (ab Firmware Version 06042009)
5	HM (Homing Mode)	1
4	R	0
3	TQ (Torque Mode)	1
2	PV	0
1	VL (Velocity Mode)	1
0	PP (Profile Position Mode)	1

## 9.44 Dummy Objekt Boolean (0x0001)

### Objektbeschreibung

<b>Bezeichnung</b>	Boolean
<b>SDO-ID</b>	0x0001
<b>Typ</b>	Bit, rw
<b>Wertebereich</b>	Bit

Gedacht als Dummy-Objekt beim PDO-Mapping.

Kann nicht als PDO gemapt werden.

## 9.45 Dummy Objekt Integer8 (0x0002)

### Objektbeschreibung

<b>Bezeichnung</b>	Signed Integer 8
<b>SDO-ID</b>	0x0002
<b>Typ</b>	Integer 8, rw
<b>Wertebereich</b>	-128 bis +128

Kann als Dummy-Objekt für RxPDO verwendet werden.

## 9.46 Dummy Objekt Integer16 (0x0003)

### Objektbeschreibung

<b>Bezeichnung</b>	Signed Integer 16
<b>SDO-ID</b>	0x0003
<b>Typ</b>	Integer 16, rw
<b>Wertebereich</b>	-32768 bis +32767

Kann als Dummy-Objekt für RxPDO verwendet werden.

## 9.47 Dummy Objekt Integer32 (0x0004)

### Objektbeschreibung

<b>Bezeichnung</b>	Signed Integer 32
<b>SDO-ID</b>	0x0004
<b>Typ</b>	Integer 32, rw
<b>Wertebereich</b>	-2147483648 bis +2147483647

Kann als Dummy-Objekt für RxPDO verwendet werden.

## 9.48 Dummy Objekt Unsigned8 (0x0005)

### Objektbeschreibung

<b>Bezeichnung</b>	Signed Integer 8
<b>SDO-ID</b>	0x0005
<b>Typ</b>	Integer 8, rw
<b>Wertebereich</b>	0 bis 255

Kann als Dummy-Objekt für RxPDO verwendet werden.

## 9.49 Dummy Objekt Unsigned16 (0x0006)

### Objektbeschreibung

<b>Bezeichnung</b>	Signed Integer 16
<b>SDO-ID</b>	0x0006
<b>Typ</b>	Integer 16, rw
<b>Wertebereich</b>	0 bis 65535

Kann als Dummy-Objekt für RxPDO verwendet werden.

## 9.50 Dummy Objekt Unsigned32 (0x0007)

### Objektbeschreibung

<b>Bezeichnung</b>	Signed Integer 32
<b>SDO-ID</b>	0x0007
<b>Typ</b>	Integer 32, rw
<b>Wertebereich</b>	0 bis 4294967295

Kann als Dummy-Objekt für RxPDO verwendet werden.

## 9.51 ERROR CODE LOG

### Objektbeschreibung

<b>Bezeichnung</b>	Error Code Log
<b>SDO-ID</b>	1003
<b>Subindex</b>	1 – 5
<b>Typ</b>	u32, ro
<b>Wertebereich</b>	u32

Dieses SDO enthält eine Historie der zuletzt aufgetretenen Fehler. Dabei enthält Subindex 0 die Anzahl der gespeicherten Fehler und Subindex 1 den zuletzt aufgetretenen Fehler.

Jeder Eintrag enthält dabei im niederwertigen Wort den aufgetretenen Error Code und im höherwertigen Wort eine firmwareinterne Fehlernummer.

# 10 Analogeingang

## 10.1 Überblick

### Allgemein

Die Steuerung SMCI47-S verfügt über einen Analogeingang, der Triggerbedingungen unterstützt, wie sie in CiA CANopen 401 beschrieben sind.

### Auflösung

Der Eingang verfügt über eine Auflösung von 10 Bit bei einer Wandelrate von 1kHz. Die volle Auflösung bezieht sich auf einen Eingangsspannungsbereich von -10V bis +10V.

### Werte

Die ausgegebenen Werte sind vom Typ s16, wobei nur positive Werte verwendet werden. Das entsprechende SDO (0x6401,1) wird laufend aktualisiert. Die Triggerbedingungen für entsprechend gemappte TPDOs können in den Objekten 0x6423 bis 0x6428 eingestellt werden. Eine Skalierung oder Offset-Rechnung wird nicht unterstützt.

<b>Bezeichnung</b>	Read analog input 16-bit
<b>SDO-ID</b>	0x6401 Subindex 1
<b>Typ</b>	s16, ro
<b>Wertebereich</b>	0x0000 bis 0x03FF
Enthält den letzten gewandelten Wert des Analogeingangs. Dieses Objekt kann als PDO gemappt werden. Für das PDO werden gesonderte Triggerbedingungen unterstützt	

## 10.2 Analog Input (0x6401)

### Objektbeschreibung

Dieses Objekt enthält den gewandelten Wert des Analogeingangs und ist als TxPDO mapbar.

<b>Bezeichnung</b>	Analog Input
<b>SDO-ID</b>	0x6401
<b>Typ</b>	Array
<b>Anzahl der Einträge</b>	1

<b>Bezeichnung</b>	Anzahl der Analogeingänge
<b>Subindex</b>	0
<b>Typ</b>	u8, ro
<b>Wertebereich</b>	1

<b>Bezeichnung</b>	Wanderwert Analogeingang 1
<b>Subindex</b>	1
<b>Typ</b>	u16, ro
<b>Wertebereich</b>	0x0 bis 0x3FF (10 Bit)

## 10.3 Analog Global Interrupt Enable (0x6423)

### Objektbeschreibung

Erst wenn dieses Objekt auf 1 gesetzt wurde, kann eine Änderung des Analogwertes ein Sendes des entsprechenden PDO bewirken.

<b>Bezeichnung</b>	Analog Global Interrupt Enable
<b>SDO-ID</b>	0x6423
<b>Typ</b>	U8, rw
<b>Wertebereich</b>	0 oder 1
<b>Default</b>	0

## 10.4 Analog Input Interrupt Upper Limit (0x6424)

### Objektbeschreibung

Ändert sich der Analogwert und ist größer als die in diesem Objekt eingestellte Schwelle, wird ein PDO gesendet.

Ist das Upper Limit kleiner als das Lower Limit, ist der Bereich zwischen Upper und Lower Limit aktiv.

Ist das Upper Limit größer als das Lower Limit, sind die Bereiche zwischen 0 und Lower Limit und zwischen Upper Limit und unendlich aktiv.

<b>Bezeichnung</b>	Analog Input Interrupt Upper Limit
<b>SDO-ID</b>	0x6424
<b>Typ</b>	Array
<b>Anzahl der Einträge</b>	1

<b>Bezeichnung</b>	Anzahl der Analogeingänge
<b>Subindex</b>	0
<b>Typ</b>	u8, ro
<b>Wertebereich</b>	1

<b>Bezeichnung</b>	Oberer Schwellwert Analog 1
<b>Subindex</b>	1
<b>Typ</b>	s32, rw
<b>Wertebereich</b>	0x0 bis 0x3FF (10 Bit)

## 10.5 Analog Input Interrupt Lower Limit (0x6425)

### Objektbeschreibung

Ändert sich Analogwert und ist kleiner als die in diesem Objekt eingestellte Schwelle, wird ein PDO gesendet.

Ist das Upper Limit kleiner als das Lower Limit, ist der Bereich zwischen Upper und Lower Limit aktiv.

Ist das Upper Limit größer als das Lower Limit, sind die Bereiche zwischen 0 und Lower Limit und zwischen Upper Limit und unendlich aktiv.

<b>Bezeichnung</b>	Analog Input Interrupt Lower Limit
<b>SDO-ID</b>	0x6425
<b>Typ</b>	Array
<b>Anzahl der Einträge</b>	1

<b>Bezeichnung</b>	Anzahl der Analogeingänge
<b>Subindex</b>	0
<b>Typ</b>	u8, ro
<b>Wertebereich</b>	1

<b>Bezeichnung</b>	Unterer Schwellwert Analog 1
<b>Subindex</b>	1
<b>Typ</b>	s32, rw
<b>Wertebereich</b>	0x0 bis 0x3FF (10 Bit)

## 10.6 Analog Input Interrupt Delta (0x6426)

### Objektbeschreibung

Ändert sich Analogwert und der Unterschied zum letzten gesendeten Wert ist größer als der in diesem Objekt eingestellte Wert, wird ein PDO gesendet. Zusätzlich werden die Grenzen aus Upper und Lower Limit ausgewertet.

<b>Bezeichnung</b>	Analog Input Delta
<b>SDO-ID</b>	0x6426
<b>Typ</b>	Array
<b>Anzahl der Einträge</b>	1

<b>Bezeichnung</b>	Anzahl der Analogeingänge
<b>Subindex</b>	0
<b>Typ</b>	u8, ro
<b>Wertebereich</b>	1

<b>Bezeichnung</b>	Änderungsgrenze Analog 1
<b>Subindex</b>	1
<b>Typ</b>	u32, rw
<b>Wertebereich</b>	0x0 bis 0x3FF (10 Bit)

## 10.7 Analog Input Interrupt negative Delta (0x6427)

### Objektbeschreibung

Ändert sich Analogwert ins negative und der Unterschied zum letzten gesendeten Wert ist größer als der in diesem Objekt eingestellte Wert, wird ein PDO gesendet. Zusätzlich werden die Grenzen aus Upper und Lower Limit ausgewertet. Ist dieser Wert kleiner als Analog Input Delta, wird er nicht ausgewertet.

<b>Bezeichnung</b>	Analog Input negative Delta
<b>SDO-ID</b>	0x6427
<b>Typ</b>	Array
<b>Anzahl der Einträge</b>	1

<b>Bezeichnung</b>	Anzahl der Analogeingänge
<b>Subindex</b>	0
<b>Typ</b>	u8, ro
<b>Wertebereich</b>	1

<b>Bezeichnung</b>	Negative Änderungsgrenze Analog 1
<b>Subindex</b>	1
<b>Typ</b>	u32, rw
<b>Wertebereich</b>	0x0 bis 0x3FF (10 Bit)

## 10.8 Analog Input Interrupt positive Delta (0x6428)

### Objektbeschreibung

Ändert sich Analogwert ins positive und der Unterschied zum letzten gesendeten Wert ist größer als der in diesem Objekt eingestellte Wert, wird ein PDO gesendet. Zusätzlich werden die Grenzen aus Upper und Lower Limit ausgewertet. Ist dieser Wert kleiner als Analog Input Delta, wird er nicht ausgewertet.

<b>Bezeichnung</b>	Analog Input positive Delta
<b>SDO-ID</b>	0x6428
<b>Typ</b>	Array
<b>Anzahl der Einträge</b>	1

<b>Bezeichnung</b>	Anzahl der Analogeingänge
<b>Subindex</b>	0
<b>Typ</b>	u8, ro
<b>Wertebereich</b>	1

<b>Bezeichnung</b>	Positive Änderungsgrenze Analog 1
<b>Subindex</b>	1
<b>Typ</b>	u32, rw
<b>Wertebereich</b>	0x0 bis 0x3FF (10 Bit)

# 11 Herstellerspezifische CAN-Objekte

## 11.1 STEP\_MODE (0x2000)

### Objektbeschreibung

<b>Bezeichnung</b>	Schrittmodus
<b>SDO-ID</b>	0x2000
<b>Typ</b>	u8, rw
<b>Wertebereich / Defaultwert</b>	1,2,4,5,8,10,16,32,64,255

Dieses Objekt ist obsolet. Der Schrittmodus, der hier eingestellt werden konnte, ergibt sich aus den Objekten „Feed Constant“ und „Gear Ratio“. Ein weiches Fahren ist durch die Mikroschritte-Interpolation gegeben, die immer aktiv ist.

## 11.2 ENABLE\_CL (0x2001)

### Objektbeschreibung

<b>Bezeichnung</b>	Enable closed loop
<b>SDO-ID</b>	0x2001
<b>Typ</b>	u8, rw
<b>Wertebereich</b>	0, 1, 2, 3

Wird der Wert auf '1', '2' oder '3' gesetzt, wird die Firmware angewiesen, den Regelkreis zu aktivieren. Dieser wird aber erst dann aktiviert, wenn gewisse Voraussetzungen erfüllt sind:

Wert	Beschreibung
0	Der Regelkreis wird sofort deaktiviert.
1	Closed Loop wird aktiviert, sobald der Index erkannt wurde und die Steuerung wieder im Status „Bereit“ ist („Auto-Enable nach der Fahrt“).
2	Closed Loop wird aktiviert, sobald der Index erkannt wurde („Auto-Enable während der Fahrt“).
3	Closed Loop wird aktiviert, sobald ein kurzer CL-Testlauf durchgeführt wurde (Modes of Operation SDO 0x6060 = -2). Diese Einstellung ist ab Firmware Version 24-10-2011 verfügbar.

### Voraussetzungen

Vor dem erstmaligen Nutzen des Closed-Loop-Modus ist es nötig, die Steuerung an die Motor-Drehgeber-Kombination anzupassen. Das geschieht über den Rotorposition-Testlauf (Modes of Operation (SDO 0x6060) = -1). Andernfalls wird der Regler mit hoher Wahrscheinlichkeit mit maximalem Strom in die falsche Richtung drehen.

## 11.3 CL\_CONFIGURATION (0x2002)

### Objektbeschreibung

<b>Bezeichnung</b>	Closed loop Configuration
<b>SDO-ID</b>	0x2002
<b>Typ</b>	record
<b>Anzahl Einträge</b>	24

Enthält die Regelparameter für den Closed-Loop-Betrieb.

### Sub-Index 01h: KP\_V\_Z

<b>Typ</b>	u16, rw
<b>Defaultwert</b>	2

Zähler des Proportionalteils des Geschwindigkeitsreglers.

### Sub-Index 02h: KP\_V\_N

<b>Typ</b>	u16, rw
<b>Defaultwert</b>	0

Nenner des Proportionalteils des Geschwindigkeitsreglers als Zweierpotenz.

### Sub-Index 03h KI\_V\_Z

<b>Typ</b>	u16, rw
<b>Defaultwert</b>	1

Zähler des Integralteils des Geschwindigkeitsreglers.

### Sub-Index 04h: KI\_V\_N

<b>Typ</b>	u16, rw
<b>Defaultwert</b>	6

Nenner des Integralteils des Geschwindigkeitsreglers als Zweierpotenz.

### Sub-Index 05h KD\_V\_Z

<b>Typ</b>	u16, rw
<b>Defaultwert</b>	0

Zähler des Differenzialteils des Geschwindigkeitsreglers.

### Sub-Index 06h: KD\_V\_N

<b>Typ</b>	u16, rw
<b>Defaultwert</b>	0

Nenner des Differenzialteils des Geschwindigkeitsreglers als Zweierpotenz.

### Sub-Index 07h KP\_S\_Z

<b>Typ</b>	u16, rw
<b>Defaultwert</b>	100

Zähler des Proportionalteils des Positionsreglers.

**Sub-Index 08h: KP\_S\_N**

<b>Typ</b>	u16, rw
<b>Defaultwert</b>	0

Nenner des Proportionalteils des Positionsreglers als Zweierpotenz.

**Sub-Index 09h KI\_S\_Z**

<b>Typ</b>	u16, rw
<b>Defaultwert</b>	2

Zähler des Integralteils des Positionsreglers.

**Sub-Index 0Ah: KI\_S\_N**

<b>Typ</b>	u16, rw
<b>Defaultwert</b>	0

Nenner des Integralteils des Positionsreglers als Zweierpotenz.

**Sub-Index 0Bh KD\_S\_Z**

<b>Typ</b>	u16, rw
<b>Defaultwert</b>	300

Zähler des Differenzialteils des Positionsreglers.

**Sub-Index 0Ch: KD\_S\_N**

<b>Typ</b>	u16, rw
<b>Defaultwert</b>	0

Nenner des Differenzialteils des Positionsreglers als Zweierpotenz.

**Sub-Index 0Dh KP\_CSV\_Z**

<b>Typ</b>	u16, rw
<b>Defaultwert</b>	50

Zähler des Proportionalteils des Geschwindigkeits-Kaskadenreglers.

**Sub-Index 0Eh: KP\_CSV\_N**

<b>Typ</b>	u16, rw
<b>Defaultwert</b>	0

Nenner des Proportionalteils des Geschwindigkeits-Kaskadenreglers.

**Sub-Index 0Fh KI\_CSV\_Z**

<b>Typ</b>	u16, rw
<b>Defaultwert</b>	2

Zähler des Integralteils des Geschwindigkeits-Kaskadenreglers.

**Sub-Index 10h: KI\_CSV\_N**

<b>Typ</b>	u16, rw
<b>Defaultwert</b>	0

Nenner des Integralteils des Geschwindigkeits-Kaskadenreglers.

**Sub-Index 11h KD\_CSV\_Z**

<b>Typ</b>	u16, rw
<b>Defaultwert</b>	0

Zähler des Differentialteils des Geschwindigkeits-Kaskadenreglers.

**Sub-Index 12h: KD\_CSV\_N**

<b>Typ</b>	u16, rw
<b>Defaultwert</b>	0

Nenner des Differentialteils des Geschwindigkeits-Kaskadenreglers.

**Sub-Index 13h KP\_CSS\_Z**

<b>Typ</b>	u16, rw
<b>Defaultwert</b>	2

Zähler des Proportionalteils des Positions-Kaskadenreglers.

**Sub-Index 14h: KP\_CSS\_N**

<b>Typ</b>	u16, rw
<b>Defaultwert</b>	0

Nenner des Proportionalteils des Positions-Kaskadenreglers.

**Sub-Index 15h KI\_CSS\_Z**

<b>Typ</b>	u16, rw
<b>Defaultwert</b>	1

Zähler des Integralteils des Positions-Kaskadenreglers.

**Sub-Index 16h: KI\_CSS\_N**

<b>Typ</b>	u16, rw
<b>Defaultwert</b>	6

Nenner des Integralteils des Positions-Kaskadenreglers.

**Sub-Index 17h KD\_CSS\_Z**

<b>Typ</b>	u16, rw
<b>Defaultwert</b>	0

Zähler des Differentialteils des Positions-Kaskadenreglers.

**Sub-Index 18h: KD\_CSS\_N**

<b>Typ</b>	u16, rw
<b>Defaultwert</b>	0

Zähler des Differentialteils des Positions-Kaskadenreglers.

## 11.4 A/D Converter (0x2003)

### Objektbeschreibung

<b>Bezeichnung</b>	Istwert der A/D Konverter
<b>SDO-ID</b>	0x2003
<b>Typ</b>	record
<b>Anzahl Einträge</b>	4

Enthält die Istwerte der A/D Wandler (Raw Werte)

### Sub-Index 01h: A/D 1 – Temperature Channel

<b>Typ</b>	u16, ro
<b>Defaultwert</b>	-

### Sub-Index 02h: A/D 2 – Power Channel

<b>Typ</b>	u16, ro
<b>Defaultwert</b>	-

### Sub-Index 03h A/D 3 – Analog Input Channel

<b>Typ</b>	u16, ro
<b>Defaultwert</b>	-

### Sub-Index 04h: A/D 4 – Auxiliary Power Channel

<b>Typ</b>	u16, ro
<b>Defaultwert</b>	-

## 11.5 Current control (0x2004)

### Objektbeschreibung

<b>Bezeichnung</b>	Strombegrenzung
<b>SDO-ID</b>	0x2004
<b>Typ</b>	record
<b>Anzahl Einträge</b>	14

Enthält die eingestellten Strom-Werte.

### Sub-Index 01h: drive current

<b>Typ</b>	u8, rw
<b>Defaultwert</b>	20

Strom der für eine normale Fahrt verwendet wird. Angabe in %.

### Sub-Index 02h: current reduction

<b>Typ</b>	u8, rw
<b>Defaultwert</b>	20

Strom der für die Reduzierung verwendet wird. Angabe in %.

**Sub-Index 03h: Stromreduzierzeit**

<b>Typ</b>	u16, rw
<b>Defaultwert</b>	80

Zeit in Millisekunden ab Stillstand des Motors im Open-Loop Betrieb, bis der Strom reduziert wird.

**Sub-Index 04h: Strom für Block-Referenzfahrt**

<b>Typ</b>	u8, rw
<b>Defaultwert</b>	0

Strom für die Block-Referenzfahrt in Prozent.  
 Bei einem Wert von 0 wird der Phasenstrom verwendet.

**Sub-Index 05h: Ausschwingzeit**

<b>Typ</b>	u16, rw
<b>Defaultwert</b>	80

Zeit in Millisekunden zwischen Stillstand der Achse und Positionsreset nach einer Block-Referenzfahrt und zwischen Stillstand der Achse und einer Bereitmeldung im PP-Modus

**Sub-Index 06h: Positionstoleranz**

<b>Typ</b>	u32, rw
<b>Defaultwert</b>	0xFFFFFFFF

Toleranz nach einer Fahrt im PP-Modus (Open Loop).

**Sub-Index 07h: Motor Type**

<b>Typ</b>	u8, rw
<b>Defaultwert</b>	0

Motor Typ:

- 0: Schrittmotor mit und ohne Quadratur-Encoder
- 1: BLDC mit Hall-Sensor und ohne Quadratur-Encoder
- 2: BLDC mit Hall-Sensor und mit Quadratur-Encoder

**Sub-Index 08h: Lastwinkel Stützabstand**

<b>Typ</b>	u16, rw
<b>Defaultwert</b>	4096

Stützstellenabstand der Lastwinkelkurve in SDO 0x200C.  
 Ein Wert von 8192 entspricht 1000UPM.

**Sub-Index 09h: Kaskadenregler Einschaltgeschwindigkeit**

<b>Typ</b>	u32, rw
<b>Defaultwert</b>	327680

Ab dieser Geschwindigkeit wird der Kaskadenregler aktiviert.  
 Ein Wert von 8192 entspricht 1000UPM.

### Sub-Index 0Ah: Kaskadenregler Ausschaltgeschwindigkeit

<b>Typ</b>	u32, rw
<b>Defaultwert</b>	512

Ab dieser Geschwindigkeit wird der Kaskadenregler deaktiviert.  
Ein Wert von 8192 entspricht 1000UPM.

### Sub-Index 0Bh: Kaskadenregler Status

<b>Typ</b>	u8, ro
<b>Defaultwert</b>	0

Gibt an, ob der Kaskadenregler aktiv ist.

### Sub-Index 0Ch: Hall-Sensor-Modus

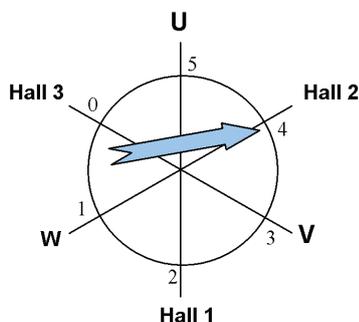
<b>Typ</b>	u32, rw
<b>Defaultwert</b>	2371605 (0x243015)

Dient bei BLDC-Motoren zum Anpassen der Hall-Sensor-Abschnitte an den jeweiligen Motor. Hierbei stellen die niedrigsten 6 Ziffern in der Hexadezimal-Schreibweise jeweils einen Quadranten der Hall-Sensoren dar.

Der Wert ergibt sich aus dem Motortyp sowie der Folge der Hall-Signale. Der Index wird wie folgt berechnet:  $(Hall1 * 1) + (Hall2 * 2) + (Hall3 * 4)$ .

#### Motorzuordnung

Index	1	2	3	4	5	6
Typ 1 (nicht DB47)	1	3	0	2	5	4
Typ 2 (DB47)	4	0	3	5	2	1



#### Beispiel: Motor DB57

Hall-Folge laut Datenblatt:

Hall 1	0	0	0	1	1	1
Hall 2	0	1	1	1	0	0
Hall 3	1	1	0	0	0	1
Index	4	6	2	3	1	5
Wert (Typ1)	2	4	3	0	1	5

Aus den Hall-Zuständen kann der Index berechnet werden. Anhand der Tabelle für die Motorzuordnung ergeben sich dann die Werte 2, 4, 3, 0, 1, 5. Diese werden als Hex 0x243015 in die Steuerung eingetragen.

**Sub-Index 0Dh: Spitzenstrom**

<b>Typ</b>	u32, rw
<b>Defaultwert</b>	0 (0x0)

Dient bei BLDC-Motoren zum Anpassen des maximal zulässigen Spitzenstroms. Ist dieser Wert kleiner als der Motorstrom (Subindex 1), wird er ignoriert. Ist der Wert größer, kann der Motorstrom in Abhängigkeit der Zeitkonstante kurzzeitig überschritten werden.

**Sub-Index 0Eh: Spitzenstrom Zeitkonstante**

<b>Typ</b>	u32, rw
<b>Defaultwert</b>	0 (0x0)

Dient bei BLDC-Motoren zum Anpassen der Zeitkonstante für den Spitzenstrom. Ermöglicht eine kurzzeitige Überschreitung des eingestellten Motorstroms (Subindex 1).

## 11.6 CAN enable und Baudrate (0x2005)

**Objektbeschreibung**

Mit diesem Objekt wird die CAN-Baudrate verstellt. Für eine genauere Beschreibung siehe Abschnitt 1.1.3.

<b>Bezeichnung</b>	CAN enable und Baudrate
<b>SDO-ID</b>	0x2005
<b>Typ</b>	u8
<b>Wertebereich</b>	130,131,132,133,134,135
<b>Defaultwert</b>	135 (wird nicht bei einem EEPROM Reset zurückgesetzt)

## 11.7 Motor Pole Pairs (0x2006)

**Objektbeschreibung**

<b>Bezeichnung</b>	Anzahl der Polpaare des Motors
<b>SDO-ID</b>	0x2006
<b>Typ</b>	u16
<b>Wertebereich</b>	50 und 100
<b>Defaultwert</b>	50

Gibt die Anzahl der Polpaare des Motors an. Nach dem Ändern dieses Parameters muss die Steuerung neu bestromt werden. Für die Verwendung im Closed-Loop muss der Motor mit Hilfe des Rotorposition-Testlaufs neu justiert werden.

Polpaarzahl	Schrittwinkel
50	1,8°
100	0,9°

## 11.8 Brake Wait Time (0x2007)

### Objektbeschreibung

<b>Bezeichnung</b>	Wartezeit für mech. Bremse
<b>SDO-ID</b>	0x2007
<b>Typ</b>	record
<b>Anzahl Einträge</b>	3

Die SMC147-S verfügt über einen Ausgang zum Anschluss einer federbelasteten, elektromagnetisch lösbaren Bremse. Mit diesem SDO können Verzögerungszeiten für diese Bremse eingestellt werden.

Während eine Brems-Wartezeit ausgeführt wird, werden keine Kommandos zum Moduswechsel ausgeführt.

### Subindex 01h: Milliseconds Power on to Brake off

<b>Typ</b>	u16, rw
<b>Defaultwert</b>	0
Gibt die Wartezeit zwischen Einschalten des Leistungsteils und Lösen der Bremse in Millisekunden an.	

### Subindex 02h: Milliseconds Brake off to Operational

<b>Typ</b>	u16, rw
<b>Defaultwert</b>	0
Gibt die Wartezeit zwischen Lösen der Bremse und Erreichen des Zustands „Operational“ in Millisekunden an. Diese Einstellung verhindert, dass der Motor sich aufgrund eines Kommandos dreht, während die Bremse noch nicht vollständig gelöst ist.	

### Subindex 03h: Milliseconds Brake on to Power off

<b>Typ</b>	u16, rw
<b>Defaultwert</b>	0
Gibt die Wartezeit zwischen Festsetzen der Bremse und Abschalten des Leistungsteils in Millisekunden an. Diese Einstellung verhindert, dass sich der Motor aufgrund eines mechanischen Moments bewegt, wenn sich die Bremse noch nicht vollständig festgesetzt hat.	

## 11.9 Milliseconds Input Debounce Time (0x2008)

### Objektbeschreibung

Dient zum Entprellen der digitalen Eingänge der Steuerung. Nach einer Flanke an einem Eingang wird innerhalb der in diesem Objekt eingestellten Zeit in Millisekunden keine weitere Flanke verarbeitet. Erst nach Ablauf der Entprellzeit wird wieder eine Flanke erkannt. Eine laufende Entprellzeit eines Eingangs hat keinen Einfluss auf die Erkennung von Flanken auf den anderen Eingängen.

<b>Bezeichnung</b>	Entprellzeit
<b>SDO-ID</b>	0x2008
<b>Typ</b>	u8, rw
<b>Wertebereich</b>	0 bis 255
<b>Defaultwert</b>	20

## 11.10 Node-ID (0x2009)

### Objektbeschreibung

Mit diesem Objekt wird die Node-ID verstellt, wenn mit den Hex-Schaltern die Node-ID 0 eingestellt ist. Ist eine Node-ID mit den Hex-Schaltern eingestellt, kann dieses Objekt zwar beschrieben und im EEPROM gespeichert werden, aber nach einem Neustart enthält es wieder den Wert der Hex-Schalter.

<b>Bezeichnung</b>	Node-ID
<b>SDO-ID</b>	0x2009
<b>Typ</b>	u8, rw
<b>Wertebereich</b>	1 bis 127
<b>Defaultwert</b>	1 (wird nicht bei einem EEPROM Reset zurückgesetzt)

## 11.11 CL is enabled (0x200A)

### Objektbeschreibung

Mit Hilfe dieses Objektes kann ermittelt werden, ob der Closed-Loop-Modus aktiv ist. Liefert das Objekt den Wert 1, ist der Closed-Loop-Modus aktiv.

<b>Bezeichnung</b>	CL is enabled
<b>SDO-ID</b>	0x200A
<b>Typ</b>	u8, ro
<b>Wertebereich</b>	0 und 1
<b>Defaultwert</b>	0

## 11.12 CL POSCNT Offset (0x200B)

### Objektbeschreibung

Gibt den Abstand zwischen Encoder-Index und Rotor-Ausrichtung an. Ein Wert von 65536 entspricht 360° bzw. 0° elektrischer Winkel. Dieser Wert wird vom CL-Testlauf belegt und ist bei jedem einzelnen Motor unterschiedlich.

<b>Bezeichnung</b>	CL POSCNT Offset
<b>SDO-ID</b>	0x200B
<b>Typ</b>	u16, rw
<b>Wertebereich</b>	0 bis 65535
<b>Defaultwert</b>	0

## 11.13 CL Lastwinkel Kurve (0x200C)

### Objektbeschreibung

Gibt die von der Drehzahl abhängigen Lastwinkel an. Ein Wert von 65536 entspricht 360° bzw. 0° elektrischer Winkel. Dieser Wert wird vom CL-Testlauf belegt und ist Motoren der gleichen Baureihe identisch. Jeder der Subindezes von 1 bis 7 enthält einen Lastwinkel für einen Drehzahlbereich. Der Drehzahl-Abstand zwischen den einzelnen Werten kann in SDO 2004 Subindex 8 eingestellt werden.

<b>Bezeichnung</b>	CL Lastwinkel Kurve
<b>SDO-ID</b>	0x200C
<b>Typ</b>	Array
<b>Anzahl Einträge</b>	10

### Sub-Index 00h Anzahl der Einträge

<b>Typ</b>	u8, ro
<b>Defaultwert</b>	10

### Sub-Index 01h: Lastwinkel im Stillstand

<b>Typ</b>	u16, rw
<b>Defaultwert</b>	16384

### Sub-Index 02h Lastwinkel bei Drehzahl = Lastwinkel Stützabstand \* 1

<b>Typ</b>	u16, rw
<b>Defaultwert</b>	17000

### Sub-Index 03h Lastwinkel bei Drehzahl = Lastwinkel Stützabstand \* 2

<b>Typ</b>	u16, rw
<b>Defaultwert</b>	17500

### Sub-Index 04h Lastwinkel bei Drehzahl = Lastwinkel Stützabstand \* 3

<b>Typ</b>	u16, rw
<b>Defaultwert</b>	17750

**Sub-Index 05h Lastwinkel bei Drehzahl = Lastwinkel Stützabstand \* 4**

<b>Typ</b>	u16, rw
<b>Defaultwert</b>	18000

**Sub-Index 06h Lastwinkel bei Drehzahl = Lastwinkel Stützabstand \* 5**

<b>Typ</b>	u16, rw
<b>Defaultwert</b>	18000

**Sub-Index 07h Lastwinkel bei Drehzahl = Lastwinkel Stützabstand \* 6**

<b>Typ</b>	u16, rw
<b>Defaultwert</b>	18000

**Sub-Index 08h wird nicht verwendet**

<b>Typ</b>	u16, rw
<b>Defaultwert</b>	18000

**Sub-Index 09h wird nicht verwendet**

<b>Typ</b>	u16, rw
<b>Defaultwert</b>	18000

**Sub-Index 0Ah wird nicht verwendet**

<b>Typ</b>	u16, rw
<b>Defaultwert</b>	18000

## 11.14 Encoder Drehrichtung (0x200D)

**Objektbeschreibung**

Wird dieses Objekt auf 1 gesetzt, wird die Richtung des Quadratur-Encoders umgedreht.

<b>Bezeichnung</b>	Encoder Drehrichtung
<b>SDO-ID</b>	0x200D
<b>Typ</b>	u8, rw
<b>Wertebereich</b>	0 oder 1
<b>Defaultwert</b>	0

## 11.15 Stromregler Parameter DSPdrive (0x200E)

### Objektbeschreibung

Bei Steuerungen mit DSPdrive (PD2-N, PD4-N, SMCI12, SMCI36) können die Stromregler-Parameter verändert werden.

<b>Bezeichnung</b>	Stromregler Parameter DSPdrive
<b>SDO-ID</b>	0x200E
<b>Typ</b>	Array
<b>Anzahl Einträge</b>	6

### Sub-Index 00h Anzahl der Einträge

<b>Typ</b>	u8, ro
<b>Defaultwert</b>	6

### Sub-Index 01h: KP low (wird nicht verwendet)

<b>Typ</b>	u16, rw
<b>Defaultwert</b>	1

### Sub-Index 02h KP high

<b>Typ</b>	u16, rw
<b>Defaultwert</b>	10

KP-Wert im Stillstand.

### Sub-Index 03h KP scale

<b>Typ</b>	u16, rw
<b>Defaultwert</b>	58

KP-Wert wird proportional zur Drehzahl erhöht.

### Sub-Index 04h KI low (wird nicht verwendet)

<b>Typ</b>	u16, rw
<b>Defaultwert</b>	1

### Sub-Index 05h KI high

<b>Typ</b>	u16, rw
<b>Defaultwert</b>	10

KI-Wert im Stillstand

### Sub-Index 06h KI scale

<b>Typ</b>	u16, rw
<b>Defaultwert</b>	200

KI-Wert wird proportional zur Drehzahl erhöht.

## 11.16 Geschwindigkeitsmodus Reglertyp (0x200F)

### Objektbeschreibung

Wird dieses Objekt auf 1 gesetzt, wird im VL-Modus der Positionsregler statt dem Geschwindigkeitsregler verwendet. Bei sehr geringen Geschwindigkeiten kann dies von Vorteil sein.

<b>Bezeichnung</b>	Geschwindigkeitsmodus Reglertyp
<b>SDO-ID</b>	0x200F
<b>Typ</b>	u8, rw
<b>Wertebereich</b>	0 oder 1
<b>Defaultwert</b>	0

## 11.17 Externe Referenzfahrt IO (0x2010)

### Objektbeschreibung

Statt Eingang 6 der Steuerung kann auch ein externes IO-Node als Referenzschalter verwendet werden.

<b>Bezeichnung</b>	Externe Referenzfahrt IO
<b>SDO-ID</b>	0x2010
<b>Typ</b>	Array
<b>Anzahl Einträge</b>	2

#### Sub-Index 00h Anzahl der Einträge

<b>Typ</b>	u8, ro
<b>Defaultwert</b>	2

#### Sub-Index 01h: Bit Nummer

<b>Typ</b>	u8, rw
<b>Defaultwert</b>	0

Wählt das Bit in Subindex 2 aus, das als Referenzschalter interpretiert werden soll.

#### Sub-Index 02h Referenzschalter

<b>Typ</b>	u8, rw
<b>Defaultwert</b>	-

Dieses Objekt dient dazu, als RxPDO gemapt zu werden. Ein Bit aus diesem Objekt stellt den Zustand des Endschalters dar, auf den die Steuerung in der entsprechenden Referenzfahrt reagiert.

## 11.18 ENCODER\_TYPE (0x2011)

### Objektbeschreibung

<b>Bezeichnung</b>	Encoder type
<b>SDO-ID</b>	0x2011
<b>Typ</b>	u8, rw
<b>Wertebereich</b>	0, 1, 2, 3 (siehe unten)

Setzt den Typ des Drehgebers, der angeschlossen ist. Jeder Typ wird durch einen eindeutigen Wert repräsentiert.

### Werte

Wert	Drehgeber-Typ
0	Kein Drehgeber
1	Inkrementeller Drehgeber mit Index
2	Inkrementeller Drehgeber ohne Index
3	Absoluter Drehgeber, singleturn

Dieser Befehl ist ab Firmware Version 24-10-2011 verfügbar.

## 11.19 COB-ID SYNC message (0x1005)

### Objektbeschreibung

<b>Bezeichnung</b>	COB-ID SYNC message
<b>SDO-ID</b>	0x1005
<b>Typ</b>	u32, rw
<b>Wertebereich</b>	0 bis 0x7FFF (Siehe unten)
<b>Defaultwert</b>	0x80

Mit diesem Objekt wird die COB-ID des Sync-Objekts eingestellt.

### Bits

Die einzelnen Bits haben folgende Bedeutung:

Bit 31: Keine Bedeutung

Bit 30: Gerät erzeugt eine Sync-Nachricht (immer 0)

Bit 29 Verwende 29-Bit COB-ID (immer 0)

Bit 28-11: Erweiterte COB-ID (immer 0)

Bit 10-0: COB-ID

## 11.20 Hardware Version (0x1009)

### Objektbeschreibung

<b>Bezeichnung</b>	Hardware Version
<b>SDO-ID</b>	0x1009
<b>Typ</b>	String (ro)
<b>Wertebereich</b>	-
<b>Defaultwert</b>	-

Dieses Objekt enthält die Hardware Version als Zeichenkette.

In Subindex 0 dieses SDO steht die Länge der Zeichenkette. Ab Subindex 1 sind die einzelnen Zeichen enthalten. Die Zeichenkette ist nicht per Null-Zeichen Terminiert.

## 11.21 Software Version (0x100A)

### Objektbeschreibung

<b>Bezeichnung</b>	Software Version
<b>SDO-ID</b>	0x1009
<b>Typ</b>	String (ro)
<b>Wertebereich</b>	-
<b>Defaultwert</b>	-

Dieses Objekt enthält die Software Version als Zeichenkette.

In Subindex 0 dieses SDO steht die Länge der Zeichenkette. Ab Subindex 1 sind die einzelnen Zeichen enthalten. Die Zeichenkette ist nicht per Null-Zeichen terminiert.

## 11.22 Guard Time (0x100C)

### Objektbeschreibung

<b>Bezeichnung</b>	Guard Time
<b>SDO-ID</b>	0x100C
<b>Typ</b>	u16, rw
<b>Wertebereich</b>	0 bis 65535
<b>Defaultwert</b>	0

Die Guard Time wird in ms eingestellt. Sie gibt an, in welchen zeitlichen Abständen eine Anforderung des NMT-Status (RTR auf COB-ID 0x700 + Node-ID) erwartet wird.

Mit dem Wert 0 kann die Node-Guard Funktion deaktiviert werden.

## 11.23 Life Time Factor (0x100D)

### Objektbeschreibung

<b>Bezeichnung</b>	Life Time Factor
<b>SDO-ID</b>	0x100D
<b>Typ</b>	u8, rw
<b>Wertebereich</b>	0 bis 255
<b>Defaultwert</b>	0

Der Life Time Factor gibt an, nach wie vielen fehlenden NMT-Status Anforderungen (RTR auf 0x700+ Node-ID) ein Fehler erzeugt wird.

Die sich aus den Objekten „Guard Time“ und „Life Time Factor“ ergebende Zeit darf eine Minute nicht überschreiten.

## 11.24 Store Parameters (0x1010)

### Objektbeschreibung

<b>Bezeichnung</b>	Store Parameters
<b>SDO-ID</b>	0x1010
<b>Typ</b>	u32, rw
<b>Wertebereich</b>	0x65766173
<b>Defaultwert</b>	-

Wird 0x65766173 in dieses Objekt geschrieben, wird das gesamte Objektverzeichnis ins EEPROM der Steuerung geschrieben. Nach einem Neustart der Steuerung bleiben so Einstellungen erhalten. Ein Einschalten (Power-Statemachine) ist aber weiterhin von Nöten.

## 11.25 Restore Parameters (0x1011)

### Objektbeschreibung

<b>Bezeichnung</b>	Restore Parameters
<b>SDO-ID</b>	0x1011
<b>Typ</b>	u32, rw
<b>Wertebereich</b>	0x64616F6C
<b>Defaultwert</b>	-

Wird 0x64616F6C in dieses Objekt geschrieben, wird das gesamte Objektverzeichnis auf die Defaultwerte zurückgesetzt. Die Defaultwerte werden auch gleich ins EEPROM übertragen.

## 11.26 Acceleration notation index (0x608D)

### Objektbeschreibung

<b>Bezeichnung</b>	Acceleration notation index
<b>SDO-ID</b>	0x608D
<b>Typ</b>	u8, rw
<b>Wertebereich</b>	u8
<b>Defaultwert</b>	0

Reserved.

## 11.27 Acceleration dimension index (0x608E)

### Objektbeschreibung

<b>Bezeichnung</b>	Acceleration dimension index
<b>SDO-ID</b>	0x608E
<b>Typ</b>	s8, rw
<b>Wertebereich</b>	s8
<b>Defaultwert</b>	0

Reserved.

## 12 Prozessdatenobjekte (PDO)

### 12.1 Zweck der PDOs

Prozessdatenobjekte (PDOs) dienen zum Übertragen von Objekten, die häufig während des Betriebs der Steuerung aktualisiert werden müssen. Sinnvoll ist dies beispielsweise für das Objekt „Position Actual Value“.

Die Vorteile von PDOs (gegenüber SDOs) sind die höhere und einstellbare Priorität, der geringe Overhead sowie Zusatzfunktionen wie automatisches Senden bei Änderung oder zyklisches Senden.

Die höhere Priorität und der geringe Overhead der PDOs ergeben sich, weil die entsprechenden Objekte aus dem Objektverzeichnis ohne Verwendung des SDO-Protokolls einem CAN-Objekt mit einer bestimmten COB-ID zugeordnet werden. Beim Vorgang des PDO-Mappings werden diese Zuordnungen eingestellt.

Bei PDOs unterscheidet man zwischen Empfangs- und Sende-PDOs (RPDO, TPDO). RPDOs werden von der SMC147-S empfangen und die empfangenen Daten in die eingestellten Objekte übernommen. TPDOs werden von der SMC147-S bei bestimmten (einstellbaren) Ereignissen gesendet.

### 12.2 PDO-Mapping

#### Generelle Vorgehensweise

Das Mappen von RPDOs/TPDOs erfolgt in mehreren Schritten:

Schritt	Tätigkeit
1	In den Modus „Pre-Operational“ wechseln.
2	Das zu ändernde PDO deaktivieren und rücksetzen.
3	Geänderte Mapping schreiben und PDO wieder aktivieren.
4	Steuerung wieder in den Modus „Operational“ bringen, um die PDO-Funktionalität zu nutzen.

#### Beispiel: RPDO-Mapping

Nachfolgend wird beispielhaft das Ändern eines RPDO-Mappings beschrieben. Das Ändern eines TPDO-Mappings erfolgt entsprechend. Die Objekte, die zum Einstellen der anderen PDOs nötig sind, sind ab Abschnitt 9.10 beschrieben.

#### Modus Pre-Operational

Damit Objekte gemappt werden können, muss zuerst in den Modus Pre-Operational (0x80) gewechselt werden. Siehe hierzu Kapitel 2.2.

COB-ID	Datenbytes	Beschreibung
0	80 01	Node 1 nach Pre-Operational
701	7F	Antwort Node 1: Pre-Operational

#### PDO deaktivieren

Um ein PDO zu deaktivieren, muss das höchstwertige Bit (MSB) im Objekt „Receive/Transmit PDO Communication Parameter : COB-ID“ gesetzt werden.

→ z.B. Objekt 0x1400 Subindex 1 auf (u32)0x80000000 setzen (deaktiviert RPDO 1).

COB-ID	Datenbytes	Beschreibung
601	23 00 14 01 00 00 00 80	Schreibe COB-ID
581	60 00 14 01 00 00 00 00	Antwort: OK

#### CAN Objekt Identifier (COB-ID) setzen

Für das eigentliche Mapping muss eine COB-ID vergeben werden. Es ist darauf zu achten, dass die COB-ID nur an ein PDO vergeben wird. Je kleiner die COB-ID, desto höher ist die Priorität auf dem CAN-Bus.

→ z.B. Objekt 0x1400 Subindex 1 auf (u32)0x80000201 setzen.

COB-ID	Datenbytes	Beschreibung
601	23 00 14 01 01 02 00 80	Schreibe COB-ID
581	60 00 14 01 00 00 00 00	Antwort: OK

#### Transmission Type setzen

In SDO 0x1400 Subindex 2 wird der Transmission Type angegeben. Wird der Transmission Type auf 255 gesetzt, werden die im PDO abgebildeten Objekte bei Empfang des PDOs sofort aktualisiert. Wird der Transmission Type auf einen Wert zwischen 0 und 240 eingestellt, werden die im PDO abgebildeten Objekte beim Empfang des Sync-Objekts aktualisiert.

-> z.B. Objekt 0x1400 Subindex 2 auf (u8)255 setzen.

COB-ID	Datenbytes	Beschreibung
601	2F 00 14 02 FF 00 00 00	Schreibe Transmission Type
581	60 00 14 02 00 00 00 00	Antwort: OK

An dieser Stelle können dann bei TPDO noch die Subindizes 3 und 5 beschrieben werden. Diese enthalten die Inhibit-Time und die Event-Time.

#### Mapping deaktivieren

Die Anzahl der abgebildeten Objekte auf 0 zurücksetzen.

→ z.B. Objekt 0x1600 Subindex 0 auf (u8)0 setzen.

COB-ID	Datenbytes	Beschreibung
601	2F 00 16 00 00 00 00 00	Schreibe Anzahl der Mappings auf 0
581	60 00 16 00 00 00 00 00	Antwort: OK

#### Mapping ändern

Über das SDO 0x1600 Subindex 1 bis 8 wird festgelegt, welche Objekte aus dem Objektverzeichnis in diesem PDO abgebildet werden. Jeder Subindex kann mit einem u32 beschrieben werden, das Index und Subindex des abzubildenden Objekts enthält. Bit 31 bis 16 enthalten den Index, Bit 15 bis 8 den Subindex. Bit 7 bis 0 geben die Länge des abzubildenden Objekts in Bit an. Bit 7 bis 0 müssen nicht gesetzt werden, dies wird von der Firmware übernommen.

Im folgenden Beispiel wird das Controlword (0x6040 sub 0) in RPDO 1 abgebildet:

→ z.B. Objekt 0x1600 Subindex 1 auf (u32)60400000 setzen.

COB-ID	Datenbytes	Beschreibung
601	23 00 16 01 00 00 40 60	Schreibe Mapping 1
581	60 00 16 01 00 00 00 00	Antwort: OK

Werden mehrere Objekte in ein PDO abgebildet, ist darauf zu achten, dass ein PDO eine maximale Länge von 8 Byte hat. Wird die maximale Länge eines PDOs überschritten, meldet die Firmware einen Fehler beim Mapping.

### Mapping aktivieren

Zum Aktivieren des Mappings ist es nötig, die Anzahl der abzubildenden Objekte in SDO 0x1600 Subindex 0 zu schreiben.

→ z.B. Objekt 0x1600 Subindex 0 auf (u8)1 setzen.

COB-ID	Datenbytes	Beschreibung
601	2F 00 16 00 01 00 00 00	Schreibe Anzahl der Mappings auf 1
581	60 00 16 00 01 00 00 00	Antwort: OK

### PDO aktivieren

Als letzter Schritt muss das Mapping aktiviert werden. Dazu ist das MSB im SDO 0x1400 Subindex 0 zu löschen.

→ z.B. Objekt 0x1400 Subindex 1 auf (u32)0x201 setzen.

COB-ID	Datenbytes	Beschreibung
601	2F 00 14 01 01 02 00 00	Schreibe COB-ID
581	60 00 14 01 00 00 00 00	Antwort: OK

### Aktivieren des Node

Da die PDOs nur im Modus „Operational“ funktionieren, muss nach Abschluss des PDO-Mappings per Netzwerk-Management in diesen Modus gewechselt werden. Siehe Abschnitt 2.2.

COB-ID	Datenbytes	Beschreibung
0	01 01	Node nach Operational
701	5	Antwort Node 1: Operational

### Speichern der Einstellungen im EEPROM

Damit das geänderte PDO-Mapping nicht bei einem Abschalten der Spannungsversorgung verloren geht, können alle Einstellungen im EEPROM abgelegt werden, woraus diese bei jedem Hochfahren geladen werden.

Dazu ist der Wert (u32)0x65766173 in das Objekt 0x1010 Subindex 1 zu schreiben. (Der Wert entspricht dem String „save“)

COB-ID	Datenbytes	Beschreibung
601	23 10 10 01 73 61 76 65	Speichere alle Einstellungen
581	60 10 10 01 00 00 00 00	Antwort: OK

## 12.3 Dummy Objekte

Dummy-Objekte (SDO 0x0002 bis 0x0007) können verwendet werden, um die Teile eines RPDO, die für eine andere Steuerung bestimmt sind, für die eigene Steuerung auszublenden.

Hierzu muss lediglich beim PDO-Mapping an der gewünschten Stelle statt eines Nutzdaten-SDO ein passendes Dummy-SDO gemapt werden. Ein Dummy-Objekt kann mehrfach verwendet werden. Daten, die in ein Dummy-Objekt geschrieben werden haben keine sonstige Auswirkung auf die Steuerung, bis auf dass das Dummy-Objekt beschrieben wurde.

## 13 Inbetriebnahme über RS485

Zum Betrieb der Steuerung über die RS485-Schnittstelle (falls vorhanden) muss die passende Firmware aufgespielt werden. Ein Betreiben der Steuerung im RS485-Modus ist mit der CANopen-Firmware nicht möglich.

## 14 Anhang: Verwendete Abkürzungen

Abkürzung	Bedeutung
COB-ID	CAN Objekt ID
EMCY	Emergency Object
HM	Homing Mode (Referenzmodus)
IP	Interpolated Position Mode
PDO	Process Data Object
PP	Profile Position Mode
ro	read only
RPDO	Receive Process Data Object
rw	read write
SDO	Service Data Object
SYNC	Synchronisation Object
TPDO	Transmit Process Data Object
TQ	Torque Mode
VL	Velocity Mode

## 15 Anhang: Mögliche Fehlermeldungen

Fehlercode	Beschreibung
0503 0000h	Toggle bit not alternated
0504 0000h	SDO protocol timed out.
0504 0001h	Client/server command specifier not valid or unknown.
0504 0002h	Invalid block size (block mode only).
0504 0003h	Invalid sequence number (block mode only).
0504 0004h	CRC error (block mode only).
0504 0005h	Out of memory
0601 0000h	Unsupported access to an object
0601 0001h	Attempt to read a write only object
0601 0002h	Attempt to write a read only object
0602 0000h	Object does not exist in the object dictionary
0604 0041h	Object cannot be mapped to the PDO
0604 0042h	The number and length of the objects to be mapped would exceed
0604 0043h	General parameter incompatibility reason
0604 0047h	General internal incompatibility in the device
0606 0000h	Access failed due to a hardware error
0607 0010h	Data type does not match, length of service parameter does not match
0607 0012h	Data type does not match, length of service parameter too high
0607 0013h	Data type does not match, length of service parameter too low
0609 0011h	Sub-index does not exist
0609 0030h	Value range of parameter exceeded (only for write access)
0609 0031h	Value of parameter written too high
0609 0032h	Value of parameter written too low
0609 0036h	Maximum value is less than minimum value
0800 0000h	general error
0800 0020h	Data cannot be transferred or stored to the application
0800 0021h	Data cannot be transferred or stored to the application because of
0800 0022h	Data cannot be transferred or stored to the application because of the
0800 0023h	Object dictionary dynamic generation fails or no object dictionary is

## 16 Anhang: Mögliche Error Codes

Error Code	Beschreibung
0x0000	CAN_EMERGENCY_ERROR_CODE_NO_ERROR_OR_RESET
0x1000	CAN_EMERGENCY_ERROR_CODE_GENERIC_ERROR
0x2000	CAN_EMERGENCY_ERROR_CODE_CURRENT
0x2100	CAN_EMERGENCY_ERROR_CODE_CURRENT_INPUT
0x2200	CAN_EMERGENCY_ERROR_CODE_CURRENT_INSIDE
0x2300	CAN_EMERGENCY_ERROR_CODE_CURRENT_OUTPUT
0x3000	CAN_EMERGENCY_ERROR_CODE_VOLTAGE
0x3100	CAN_EMERGENCY_ERROR_CODE_VOLTAGE_MAINS
0x3200	CAN_EMERGENCY_ERROR_CODE_VOLTAGE_INSIDE
0x3300	CAN_EMERGENCY_ERROR_CODE_VOLTAGE_OUTPUT
0x4000	CAN_EMERGENCY_ERROR_CODE_TEMPERATURE
0x4100	CAN_EMERGENCY_ERROR_CODE_TEMPERATURE_AMBIENT
0x4200	CAN_EMERGENCY_ERROR_CODE_TEMPERATURE_DEVICE
0x5000	CAN_EMERGENCY_ERROR_CODE_DEVICE_HARDWARE
0x6000	CAN_EMERGENCY_ERROR_CODE_DEVICE_SOFTWARE
0x6100	CAN_EMERGENCY_ERROR_CODE_DEVICE_SOFTWARE_INTERNAL
0x6200	CAN_EMERGENCY_ERROR_CODE_DEVICE_SOFTWARE_USER
0x6300	CAN_EMERGENCY_ERROR_CODE_DEVICE_SOFTWARE_DATA
0x7000	CAN_EMERGENCY_ERROR_CODE_ADDITIONAL_MODULES
0x8000	CAN_EMERGENCY_ERROR_CODE_MONITORING
0x8100	CAN_EMERGENCY_ERROR_CODE_MONITORING_COMM
0x8110	CAN_EMERGENCY_ERROR_CODE_MONITORING_COMM_OVERRUN
0x8120	CAN_EMERGENCY_ERROR_CODE_MONITORING_COMM_PASSIVE
0x8130	CAN_EMERGENCY_ERROR_CODE_MONITORING_COMM_LIFEGUARD
0x8140	CAN_EMERGENCY_ERROR_CODE_MONITORING_COMM_BUSRECOVERY
0x8150	CAN_EMERGENCY_ERROR_CODE_MONITORING_COMM_TXCOBCOLLISION
0x8200	CAN_EMERGENCY_ERROR_CODE_MONITORING_PROT
0x8210	CAN_EMERGENCY_ERROR_CODE_MONITORING_PROT_PDO_NOPROCLENGTH
0x8220	CAN_EMERGENCY_ERROR_CODE_MONITORING_PROT_PDO_LENGTH
0x9000	CAN_EMERGENCY_ERROR_CODE_EXTERNAL
0x9100	CAN_EMERGENCY_ERROR_CODE_EXTERNAL_DOORACCESS
0xF000	CAN_EMERGENCY_ERROR_CODE_ADDITIONAL
0xFF00	CAN_EMERGENCY_ERROR_CODE_DEVICESPEC
0xFFFF	CAN_EMERGENCY_ERROR_CODE_DEVICESPEC_UNKNOWN

## 17 Anhang: Motordaten

### 17.1 Default Werte für Schrittmotoren

Lastwinkel	Wert
1	16384
2	18384
3	20384
4	22384
5	24384
6	26384
7	28384

### 17.2 Default Werte für BLDC Motoren

Lastwinkel	Wert
1	16384
2	16500
3	17000
4	17500
5	18000
6	18500
7	19000

### 17.3 Schrittmotoren der Serie STxxxx

Die folgende Tabelle gilt für Schrittmotoren der Serie ST2018, ST3518, ST4118, ST4209, ST4218, ST5709, ST5909, ST5918, ST6018, ST6318, ST8918, ST11018.

Lastwinkel	Wert
1	16384
2	16500
3	17000
4	17500
5	18000
6	18500
7	19000

## 17.4 BLDC Motoren der Serie DB22

### DB22L01

Lastwinkel	Wert
1	16000
2	16500
3	17000
4	17500
5	18000
6	18500
7	19000

### DB22M01

Lastwinkel	Wert
1	16000
2	16500
3	17000
4	17500
5	18000
6	18500
7	18500

## 17.5 BLDC Motoren der Serie DB28

### DB28M01

Lastwinkel	Wert
1	16000
2	17000
3	17000
4	17000
5	18000
6	18000
7	18000

**DB28S01**

Lastwinkel	Wert
1	16000
2	16500
3	17000
4	17500
5	18000
6	18500
7	18500

**17.6 BLDC Motoren der Serie DB33****DB33S01**

Lastwinkel	Wert
1	16000
2	16000
3	16500
4	16500
5	17000
6	17000
7	17000

**17.7 BLDC Motoren der Serie DB42****DB42C01**

Lastwinkel	Wert
1	16000
2	18000
3	20000
4	20000
5	20000
6	21000
7	20000

**DB42C02**

Lastwinkel	Wert
1	16000
2	18000
3	20000
4	20000
5	20000
6	21000
7	22000

**DB42C03**

Lastwinkel	Wert
1	16000
2	16500
3	16800
4	17100
5	17400
6	17700
7	17800

**DB42L01**

Lastwinkel	Wert
1	16000
2	17000
3	17500
4	17500
5	17700
6	18300
7	18400

**DB42M01**

Lastwinkel	Wert
1	16000
2	16500
3	17000
4	17500
5	18500
6	18750
7	19000

**DB42M02**

Lastwinkel	Wert
1	16000
2	18000
3	20000
4	20000
5	20000
6	21000
7	22000

**DB42M03**

Lastwinkel	Wert
1	16000
2	17000
3	17000
4	17000
5	18000
6	19000
7	19000

**DB42S01**

Lastwinkel	Wert
1	16000
2	16500
3	17000
4	17500
5	18000
6	18000
7	18500

**DB42S02**

Lastwinkel	Wert
1	16000
2	18000
3	18000
4	18000
5	18500
6	19000
7	19000

**DB42S03**

Lastwinkel	Wert
1	16000
2	18000
3	20000
4	20000
5	20000
6	21000
7	22000

## 17.8 BLDC Motoren der Serie DB57

### DB57C01

Lastwinkel	Wert
1	16000
2	16500
3	16500
4	16500
5	17000
6	17000
7	17000

### DB57L01

Lastwinkel	Wert
1	16000
2	17000
3	17000
4	17000
5	17000
6	17000
7	17000

### DB57S01

Lastwinkel	Wert
1	16500
2	17000
3	17000
4	17000
5	17000
6	17500
7	17500

## 17.9 BLDC Motoren der Serie DB87

### DB87L01-S

Lastwinkel	Wert
1	16384
2	17000
3	17000
4	17000
5	17000
6	17000
7	17000

### DB87M01-S

Lastwinkel	Wert
1	16384
2	18384
3	20384
4	22384
5	24384
6	26384
7	28384

### DB87S01-S

Lastwinkel	Wert
1	16000
2	16500
3	17000
4	17250
5	17500
6	17500
7	18000