



# CANopen SERVOSTAR<sup>®</sup> 400/600

## Kommunikationsprofil

**Feldbus Schnittstelle**

Ausgabe 08/2007

Bewahren Sie das Handbuch als Produktbestandteil während der Lebensdauer des Produktes auf. Geben Sie das Handbuch an nachfolgende Benutzer oder Besitzer des Produktes weiter.

Datei s400600can\_d.\*\*\*

 **DANAHER**  
MOTION

## **Bisher erschienene Ausgaben**

<b>Ausgabe</b>	<b>Bemerkung</b>
01 / 1999	Erstausgabe, gültig ab Software-Versionen 1.57
08 / 1999	Layout, kleinere Korrekturen
03 / 2001	Ergänzung des Objektverzeichnisses, gültig ab Firmwareversion 4.20
08 / 2001	Detail-Korrekturen, Index erweitert, Objektverzeichnis erweitert, gültig ab Firmwareversion 4.80
09 / 2002	neues Layout, TPDO 34 neu, diverse Korrekturen
09 / 2006	Neues Design, Ergänzung des Objektverzeichnisses, gültig ab Firmwareversion 6.57
08 / 2007	Kleinere Korrekturen, Produktbrand, Symbole, Normen

SERVOSTAR ist ein eingetragenes Warenzeichen von Danaher Motion

**Technische Änderungen, die der Verbesserung der Geräte dienen, vorbehalten!**

Gedruckt in der BRD

Alle Rechte vorbehalten. Kein Teil des Werkes darf in irgendeiner Form (Druck, Fotokopie, Mikrofilm oder in einem anderen Verfahren) ohne schriftliche Genehmigung der Firma Danaher Motion reproduziert oder unter Verwendung elektronischer Systeme verarbeitet, vervielfältigt oder verbreitet werden.

<b>1</b>	<b>Allgemeines</b>	
1.1	Über dieses Handbuch	7
1.2	Bestimmungsgemäße Verwendung des CANopen Interfaces	7
1.3	Verwendete Symbole	8
1.4	Verwendete Kürzel	8
1.5	Realisierte Grundfunktionen über CANopen	9
1.6	Verhalten bei Kommunikationsstörungen BUSOFF	9
1.7	Systemvoraussetzungen	9
1.8	Übertragungsgeschwindigkeit und -verfahren	9
<b>2</b>	<b>Installation / Inbetriebnahme</b>	
2.1	Montage, Installation	10
2.1.1	Einstellen von Stationsadresse und Baudrate	10
2.2	CANopen Interface	11
2.3	CAN Busleitung	11
2.4	Leitfaden zur Inbetriebnahme	12
2.5	Wichtige Konfigurationsparameter für den CAN-Bus - Betrieb	13
<b>3</b>	<b>CANopen Kommunikationsprofil</b>	
3.1	Allgemeine Erläuterungen zu CAN	15
3.2	Aufbau des COB-Identifiers	16
3.3	Definition der verwendeten Datentypen	16
3.3.1	Basisdatentypen	17
3.3.1.1	Unsigned Integer	17
3.3.1.2	Signed Integer	17
3.3.2	Zusammengesetzte Datentypen	17
3.3.3	Erweiterte Datentypen	18
3.3.3.1	Octet String	18
3.3.3.2	Visible String	18
3.4	Kommunikationsobjekte	18
3.4.1	Network Management Objects (NMT)	19
3.4.2	Synchronisation Object (SYNC)	19
3.4.3	Time Stamp Object (TIME)	19
3.4.4	Emergency Object (EMCY)	19
3.4.4.1	Verwendung des Emergency Object	20
3.4.4.2	Aufteilung des Emergency Object	20
3.4.5	Service Data Objects (SDO)	21
3.4.5.1	Aufbau eines Service Data Object	21
3.4.5.2	Initiate SDO Download Protocol	22
3.4.5.3	Download SDO Segment Protocol	22
3.4.5.4	Initiate SDO Upload Protocol	22
3.4.5.5	Upload SDO Segment Protocol	22
3.4.5.6	Abort SDO Protocol	23
3.4.6	Process Data Object (PDO)	23
3.4.6.1	Übertragungsmodi	24
3.4.6.2	Triggermodi	25
3.4.7	Nodeguard	25

**4 CANopen Antriebsprofil**

4.1	Emergency Messages .....	27
4.2	Allgemeine Definitionen .....	29
4.2.1	Allgemeine Objekte .....	29
4.2.1.1	Objekt 1000h: Device Type (DS301) .....	29
4.2.1.2	Objekt 1001h: Error register (DS301) .....	29
4.2.1.3	Objekt 1002h: Manufacturer Status Register (DS301) .....	30
4.2.1.4	Objekt 1003h: Pre-Defined Error Field (DS301) .....	31
4.2.1.5	Objekt 1004h: Zahl der unterstützten PDOs (DS301) .....	32
4.2.1.6	Objekt 1005h: COB-ID der SYNC - Message (DS301) .....	33
4.2.1.7	Objekt 1006h: Communication Cycle Period (DS301) .....	33
4.2.1.8	Objekt 1007h: Synchronous window length (DS301) .....	34
4.2.1.9	Objekt 1008h: Manufacturer Device Name (DS301) .....	34
4.2.1.10	Objekt 100Ah: Manufacturer Software Version (DS301) .....	34
4.2.1.11	Objekt 100Bh: Node-ID (DS301) .....	35
4.2.1.12	Objekt 100Ch: Guard Time (DS301) .....	35
4.2.1.13	Objekt 100Dh: Life Time Factor (DS301) .....	35
4.2.1.14	Objekt 100Eh: COB - ID Nodeguarding (DS301) .....	36
4.2.1.15	Objekt 100Fh: Zahl der unterstützten Objekte (DS301) .....	36
4.2.1.16	Objekt 1010h: Store Parameters (DS301) .....	37
4.2.1.17	Objekt 1012h: COB - ID für Time - Stamp Message (DS301) .....	38
4.2.1.18	Objekt 1013h: High Resolution Time Stamp (i.V.) (DS301) .....	38
4.2.1.19	Objekt 1014h: COB - ID für Emergency Message (DS301) .....	38
4.2.1.20	Objekt 1018h: Identity Object (DS301) .....	39
4.3	PDO Mapping .....	40
4.3.1	Empfangs-PDOs (RPDO) .....	41
4.3.1.1	Beschreibung der vordefinierten Empfangs-PDOs .....	41
4.3.1.1.1	PDO controlword (1) (DS402) .....	41
4.3.1.1.2	PDO ASCII-Kanal empfangen (21) .....	42
4.3.1.1.3	PDO Strom- / Drehzahlsollwert (22) .....	42
4.3.1.1.4	PDO Sollwert 2 (32) .....	42
4.3.1.1.5	PDO Trajektorie (33) .....	43
4.3.1.1.6	PDO Fahrsatz (34) .....	43
4.3.1.1.7	PDO Starte Fahrsatz (35) .....	44
4.3.1.1.8	PDO Frei definierbar (37 bis 40) .....	44
4.3.1.1.9	PDO Masterposition für CAN-Slaves (41) .....	44
4.3.1.2	Objektbeschreibung .....	45
4.3.1.2.1	Objekt 1400-1403h: 1st -4th Receive - PDO Kommunikationsparameter (DS301) .....	45
4.3.1.2.2	Objekt 1600-1603h: 1st -4th Receive - PDO Mappingparameter (DS301) .....	45
4.3.1.2.3	Objekt 2600-2603h: 1st -4th Receive PDO select .....	45
4.3.1.2.4	Objekt 2721h: Konfiguration Receive PDO 33 .....	45
4.3.2	Sende-PDOs (TPDO) .....	46
4.3.2.1	Beschreibung der vordefinierten Sende-PDOs .....	46
4.3.2.1.1	PDO statusword (1) (DS402) .....	46
4.3.2.1.2	PDO ASCII-Kanal senden (21) .....	47
4.3.2.1.3	PDO Ist-Lage (22) .....	47
4.3.2.1.4	PDO Erweiterter Status (23) .....	47
4.3.2.1.5	PDO Ist-Lage 2 (32) .....	47
4.3.2.1.6	PDO Inkrementeller Positionswert (33) .....	48
4.3.2.1.7	PDO Positionsschwellen (34) .....	48
4.3.2.1.8	PDO Frei definierbar (37 bis 40) .....	48
4.3.2.1.9	PDO Interne Master-Sollposition (41) .....	48
4.3.2.2	Objektbeschreibung .....	49
4.3.2.2.1	Objekt 1800-1803h: 1st -4th Transmit - PDO Kommunikationsparam. (DS301) .....	49
4.3.2.2.2	Objekt 1A00-1A03h: 1st -4th Transmit - PDO Mappingparameter (DS301) .....	49
4.3.2.2.3	Objekt 2A00-2A03h: 1st -4th Transmit PDO select .....	49
4.3.2.2.4	Objekt 2014-2017h: Maske 1-4 für Sende-PDOs .....	50

4.4	Gerätesteuerung (dc) . . . . .	50
4.4.1	Zustandsmaschine (DS402) . . . . .	51
4.4.1.1	Zustände der Zustandsmaschine . . . . .	51
4.4.1.2	Übergänge der Zustandsmaschine . . . . .	52
4.4.2	Objektbeschreibung . . . . .	53
4.4.2.1	Objekt 6040h: Controlword (DS402) . . . . .	53
4.4.2.2	Objekt 6041h: Statusword (DS402) . . . . .	54
4.4.2.3	Objekt 6060h: modes_of_operation (DS402) . . . . .	56
4.4.2.4	Objekt 6061h: mode_of_operation_display (DS402) . . . . .	57
4.5	Factor Groups (fg) (DS402) . . . . .	57
4.5.1	Allgemeine Informationen . . . . .	57
4.5.1.1	Faktoren . . . . .	57
4.5.1.2	Beziehung zwischen physikalischen und antriebsinternen Einheiten . . . . .	57
4.5.2	Objektbeschreibung . . . . .	58
4.5.2.1	Objekt 608Bh: velocity_notation_index (DS402) . . . . .	58
4.5.2.2	Objekt 608Ch: velocity_dimension_index (DS402) . . . . .	58
4.5.2.3	Objekt 6093h: position_factor (DS402) . . . . .	59
4.5.2.4	Objekt 6094h: velocity_encoder_factor (DS402) . . . . .	60
4.5.2.5	Objekt 6097h: acceleration_factor (DS402) . . . . .	62
4.6	Herstellerspezifischer Strom- und Drehzahlmode . . . . .	63
4.6.1	Objekt 2060h: Digitaler Strom- oder Drehzahlsollwert . . . . .	63
4.6.2	Objekt 2061h: Strombegrenzung . . . . .	63
4.7	Einrichtdaten für den herstellersiz. Tipp- und Referenziermode . . . . .	64
4.7.1	Objekt 2024h: Einrichtbetrieb für den Mode Large (SERVOSTAR) . . . . .	64
4.8	Positionierdaten für den Mode Lage (SERVOSTAR) . . . . .	66
4.8.1	Objekt 2020h: Lageregler . . . . .	66
4.8.2	Objekt 2022h: Positionierdaten für den Mode Lage . . . . .	69
4.9	Objekt 2050h: Hilfsvariable für digitale Eingänge . . . . .	75
4.10	Latchfunktion . . . . .	76
4.10.1	Objekt 2026h: Latchfreigabe . . . . .	76
4.10.2	Objekt 2082h: 32/24-bit Latch positive . . . . .	76
4.10.3	Objekt 2083h: 32/24-bit Latch negative . . . . .	77
4.10.4	Objekt 2084h: 16-bit Latch positive . . . . .	77
4.10.5	Objekt 2085h: 16-bit Latch negative . . . . .	78
4.10.6	Objekt 2087h: Latch Positionen Digitaleingang 1 . . . . .	79
4.11	Herstellerspezifische Istwerte . . . . .	80
4.11.1	Objekt 2070h: Istwerte . . . . .	80
4.11.2	Objekt 6077h: Torque actual value . . . . .	87
4.11.3	Objekt 60C2h: Interpolation time period . . . . .	87
4.12	Frei verfügbare, mappbare SPS – Variablen, Objekte 2030h / 2090h . . . . .	88
4.12.1	Objekt 2030h: DP-Ram-Variablen 9-16 (nur schreibbar) . . . . .	88
4.12.2	Objekt 2090h: DP-Ram-Variablen 1-8 (nur lesbar) . . . . .	88
4.13	Dummy - Variablen, Objekte 2031h / 2071h . . . . .	89
4.13.1	Objekt 2031h: Dummy-Variablen für Mappingzwecke (WO) . . . . .	89
4.13.2	Objekt 2071h: Dummy-Variablen für Mappingzwecke (RO) . . . . .	90
4.14	Profile Velocity Mode (pv) (DS402) . . . . .	91
4.14.1	Allgemeine Informationen . . . . .	91
4.14.2	Objekte, die in diesem Kapitel definiert werden . . . . .	91
4.14.3	Objekte, die in anderen Kapiteln definiert werden . . . . .	91
4.14.4	Objektbeschreibung . . . . .	91
4.14.4.1	Objekt 606Ch: velocity_actual_value* (DS402) . . . . .	91
4.14.4.2	Objekt 60FFh: target_velocity (DS402) . . . . .	92
4.15	Position Control Function (pc) (DS402) . . . . .	92
4.15.1	Allgemeine Informationen . . . . .	92
4.15.2	Objekte, die in diesem Kapitel definiert werden . . . . .	92
4.15.3	Objekte, die in anderen Kapiteln definiert werden . . . . .	92
4.15.4	Objektbeschreibung . . . . .	93
4.15.4.1	Objekt 6063h: position_actual_value* (DS402) . . . . .	93
4.15.4.2	Objekt 6064h: position_actual_value (DS402) . . . . .	93

---

4.16	Homing Mode (hm) (DS402) . . . . .	94
4.16.1	Allgemeine Informationen . . . . .	94
4.16.2	Objekte, die in diesem Kapitel definiert werden . . . . .	94
4.16.3	Objekte, die in anderen Kapiteln definiert werden . . . . .	94
4.16.4	Objektbeschreibung . . . . .	94
4.16.4.1	Objekt 607Ch: home_offset (DS402) . . . . .	94
4.16.4.2	Objekt 6098h: homing_method (DS402) . . . . .	95
4.16.4.2.1	Beschreibung der Referenziermethoden . . . . .	96
4.16.4.3	Objekt 6099h: homing_speeds (DS402) . . . . .	96
4.16.4.4	Objekt 609Ah: homing_acceleration (DS402) . . . . .	96
4.16.5	Homing Mode Sequence . . . . .	97
4.17	Profile Position Mode (pp) . . . . .	97
4.17.1	Allgemeine Informationen . . . . .	97
4.17.2	Objekte, die in diesem Kapitel definiert werden . . . . .	97
4.17.3	Objekte, die in anderen Kapiteln definiert werden . . . . .	98
4.17.4	Objektbeschreibung . . . . .	98
4.17.4.1	Objekt 607Ah: target_position (DS402) . . . . .	98
4.17.4.2	Objekt 607Bh: position_range_limit (DS402) . . . . .	99
4.17.4.3	Objekt 6081h: profile_velocity (DS402) . . . . .	99
4.17.4.4	Objekt 6083h: profile_acceleration (DS402) . . . . .	100
4.17.4.5	Objekt 6084h: profile_deceleration (DS402) . . . . .	100
4.17.4.6	Objekt 6086h: motion_profile_type (DS402) . . . . .	101
4.17.5	Funktionsbeschreibung . . . . .	101
<b>5</b>	<b>Der Objektkanal</b>	
5.1	Objektbeschreibung . . . . .	105
5.1.1	Objekt >3500h: Herstellerspezifischer Objektkanal . . . . .	105
<b>6</b>	<b>Anhang</b>	
6.1	Beispiele für die Inbetriebnahme . . . . .	115
6.1.1	Prinzipielle Prüfung der Verbindung Steuerung <-> SERVOSTAR . . . . .	115
6.1.2	Beispiel für die Bedienung der Zustandsmaschine . . . . .	116
6.1.3	Beispiel für PDO-Bedienung . . . . .	117
6.1.4	Beispiel für die Referenzierung . . . . .	119
6.1.5	Beispiel für Fahrsatzverarbeitung . . . . .	120
6.1.6	Beispiel für Bedienung des Profile-Position Modes . . . . .	121
6.1.7	ASCII - Kommunikation . . . . .	123
6.1.8	Test für Synctelegramme . . . . .	124
6.1.9	SYNC-Objekt . . . . .	124
6.1.10	Emergency-Objekt . . . . .	124
6.2	Spezielle Applikationen . . . . .	125
6.2.1	Externe Trajektorie . . . . .	125
6.2.1.1	Lageregler im Servoverstärker . . . . .	125
6.2.1.2	Lageregler in der Steuerung . . . . .	128
6.3	Beschreibung Objektverzeichnis . . . . .	129
6.4	Neukonfiguration des SERVOSTAR 400/600 . . . . .	135
6.5	Stichwortverzeichnis . . . . .	136

# 1 Allgemeines

## 1.1 Über dieses Handbuch

Dieses Handbuch beschreibt Inbetriebnahme, Funktionsumfang und Softwareprotokoll des Kommunikationsprofils CANopen bei Servoverstärkern der Serie SERVOSTAR® 400/600. Es ist Bestandteil der Gesamtdokumentation der digitalen Servoverstärker-Familie SERVOSTAR 400/600.

Installation und Inbetriebnahme der Servoverstärker, sowie alle Standardfunktionen werden im zugehörigen Produkthandbuch beschrieben.

### Sonstige Bestandteile der Gesamtdokumentation der digitalen Servoverstärker-Familien:

Titel	Herausgeber
Produkthandbuch SERVOSTAR 400/600	Danaher Motion
Online-Hilfe mit Objekt Referenz (Inbetriebnahme-Software)	Danaher Motion

### Weiterführende Dokumentation:

Titel	Herausgeber
CAN Application Layer (CAL) for Industrial Applications	CiA e.V.
Draft standards 301 (ab Version 4.0), 401	CiA e.V.
CAN Specification Version 2.0	CiA e.V.
ISO 11898 ...Controller area network (CAN) for high-speed communication	



### Dieses Handbuch richtet sich mit folgenden Anforderungen an Fachpersonal:

**Verdrahtung:** Fachleute mit elektrotechnischer Ausbildung  
**Programmierung:** Softwareentwickler, CAN-BUS Projektoren

Wir bieten auf Anfrage Schulungs- und Einarbeitungskurse an.

Konkrete Beispiele zu einzelnen Kapiteln sind im Applikationsteil (⇒ Kap. 6.1 und Kap. 6.2) dieses Handbuches zu finden.

## 1.2 Bestimmungsgemäße Verwendung des CANopen Interfaces

Beachten Sie das Kapitel "Bestimmungsgemäße Verwendung" im Produkthandbuch des Servoverstärkers.

Das Interface ist Bestandteil der digitalen Servoverstärker der Serie SERVOSTAR 400/600. Das CANopen Interface dient allein dem Anschluss des Servoverstärkers an einen Master mit CAN BUS Anbindung.

Die Servoverstärker werden als Komponenten in elektrischen Anlagen oder Maschinen eingebaut und dürfen nur als integrierte Komponenten der Anlage in Betrieb genommen werden.



### Wir garantieren nur bei Verwendung der von uns genannten Komponenten und Einhaltung der Installationsvorschriften die Konformität der Servoverstärker zu folgenden Normen im Industriebereich:

EG-EMV-Richtlinie	2004/108/EG
EG-Niederspannungs-Richtlinie	2006/95/EG

### 1.3 Verwendete Symbole

	Gefahr für Personen durch Elektrizität und ihre Wirkung		Gefahr für Maschinen, allgemeine Warnung		Wichtige Hinweise
⇒ S.	siehe Seite	●	Hervorhebung		

### 1.4 Verwendete Kürzel

In der Tabelle unten werden die in diesem Handbuch verwendeten Abkürzungen erklärt.

Kürzel	Bedeutung
BTB/RTO	Betriebsbereit
COB	Communication Object
COB-ID	Communication Object Identifier
EEPROM	Elektrisch löschbarer Festspeicher
EMV	Elektromagnetische Verträglichkeit
ISO	International Standardization Organisation
LED	Leuchtdiode
MB	Megabyte
NSTOP	Endschalteneingang Drehrichtung links
PC	Personal Computer
PDO	Prozessdatenobjekt
PSTOP	Endschalteneingang Drehrichtung rechts
RAM	flüchtiger Speicher
ROD	Inkrementelle Positionsausgabe
RXPDO	Receive- (Empfangs-) PDO
SDO	Servicadatenobjekt
TXPDO	Transmit- (Sende-) PDO

## 1.5 Realisierte Grundfunktionen über CANopen

In Zusammenhang mit dem im digitalen Servoverstärker SERVOSTAR 400/600 integrierten Lage-regler werden folgende Funktionen bereitgestellt:

### **Einricht- und Allgemeine Funktionen:**

- Referenzfahren, Referenzpunkt setzen
- Tippen mit variabler Geschwindigkeit
- Vorgabe digitaler Sollwerte für die Drehzahl- und Momentenregelung

### **Positionierfunktionen:**

- Ausführen eines Fahrauftrages aus dem Fahrsatzspeicher des Servoverstärkers
- Ausführen eines Direktfahrauftrages
- Trajektorie absolut

### **Datentransferfunktionen:**

- Übertragen eines Fahrauftrages in den Fahrsatzspeicher des Servoverstärkers  
Ein Fahrauftrag besteht aus folgenden Elementen:
  - » Positionssollwert (Absolutauftrag) oder Wegsollwert (Relativauftrag)
  - » Geschwindigkeitssollwert
  - » Beschleunigungszeit, Bremszeit, bei Bedarf Ruckbegrenzungszeiten (i.V.)
  - » Fahrauftragsart (absolut/relativ)
  - » Nummer eines Folgefahrauftrags (mit oder ohne Zwischenstopp)
- Lesen eines Fahrauftrages aus dem Fahrsatzspeicher des Servoverstärkers
- Lesen von Istwerten
- Lesen der Fehlerregister
- Lesen der Statusregister
- Lesen / Schreiben von Regelparametern

## 1.6 Verhalten bei Kommunikationsstörungen BUSOFF

Die Kommunikationsstörung BUSOFF wird direkt von der Schicht 2 (CAN controller) überwacht und gemeldet. Diese Meldung kann unterschiedliche Ursachen haben.

Hier einige Beispiele:

- Telegramm wird gesendet, obwohl kein weiterer CAN – Knoten angeschlossen ist
- CAN Knoten weisen unterschiedliche Baudraten auf
- Busleitung defekt
- Reflexionen auf den Leitungen aufgrund fehlerhafter Leitungsabschlüsse.

Ein BUSOFF wird vom SERVOSTAR 400/600 nur gemeldet, wenn ein weiterer CAN Knoten angeschlossen ist und mindesten ein Objekt zu Beginn erfolgreich abgesetzt werden konnte. Der Zustand BUSOFF wird mit der Fehlermeldung F23 signalisiert. Sollte beim Auftreten dieses Fehlers die Endstufe freigegeben (enable) sein und eine Fahrfunktion ausgeführt werden, wird der Antrieb mit der Notbremsrampe angehalten und die Endstufe gesperrt (disable).

## 1.7 Systemvoraussetzungen

- Servoverstärker SERVOSTAR 400/600
- Masterstation mit CAN-Bus Anbindung (z.B. PC mit CAN-Interface)

## 1.8 Übertragungsgeschwindigkeit und -verfahren

- Busankopplung und Busmedium: CAN-Standard ISO 11898 (CAN-Highspeed)
- Übertragungsgeschwindigkeit: max. 1MBit/s  
Einstellmöglichkeiten des Servoverstärkers:  
10, 20, 50, 100, 125, 250, 333, 500(default), 666, 800, 1000kBaud

## 2 Installation / Inbetriebnahme

### 2.1 Montage, Installation



Installieren und verdrahten Sie die Geräte immer in spannungsfreiem Zustand. Weder die Leistungsversorgung, noch die 24V-Hilfsspannung, noch die Betriebsspannung eines anderen anzuschliessenden Gerätes darf eingeschaltet sein.

Sorgen Sie für eine sichere Freischaltung des Schaltschranks (Sperrleiste, Warnschilder etc.). Erst bei der Inbetriebnahme werden die einzelnen Spannungen eingeschaltet.

Lösen Sie die elektrischen Anschlüsse der Servoverstärker nie unter Spannung. Es könnte zu Zerstörungen der Elektronik kommen. Restladungen in den Kondensatoren können auch mehrere Minuten nach Abschalten der Netzspannung gefährliche Werte aufweisen. Messen Sie die Spannung im Zwischenkreis und warten Sie, bis die Spannung unter 40V abgesunken ist. Steuer- und Leistungsanschlüsse können Spannung führen, auch wenn sich der Motor nicht dreht.



Montieren Sie den Servoverstärker wie im Produkthandbuch SERVOSTAR 400/600 beschrieben. Beachten Sie alle Sicherheitshinweise in der zum Servoverstärker gehörenden Produkthandbuch. Beachten Sie alle Hinweise zu Einbaulage, Umgebungsbedingungen und Verdrahtung, sowie Absicherung.

#### 2.1.1 Einstellen von Stationsadresse und Baudrate

##### Einstellen der Stationsadresse

Bei der Inbetriebnahme ist es sinnvoll, die Stationsadressen der einzelnen Verstärker und die Baudrate für die Kommunikation vorab über die Frontplattentastatur einzustellen (siehe Kapitel "Inbetriebnahme" im Produkthandbuch).

Die Stationsadresse (Geräteadresse am CAN-Bus) des Servoverstärkers können Sie auf drei Arten einstellen:

- Mit der Tastatur in der Frontplatte (siehe Produkthandbuch SERVOSTAR 400/600)
- In der Inbetriebnahme-Software DRIVE.EXE auf der Bildschirmseite "Basiseinstellungen"
- Über die serielle Schnittstelle mit der Abfolge der ASCII-Kommandos:  
ADDR nn ⇒ SAVE ⇒ COLDSTART (mit nn = Adresse)



**Nach Verändern der Stationsadresse und Baudrate müssen Sie die 24V-Hilfsspannungs-Versorgung der Servoverstärker aus- und wieder einschalten.**

Der Adressbereich kann mit Hilfe des ASCII – Objektes MDRV von 1..63 auf 1..127 erweitert werden.

##### Einstellen der Baudrate

Die CAN - Übertragungsgeschwindigkeit (Baudrate) können Sie auf drei Arten einstellen:

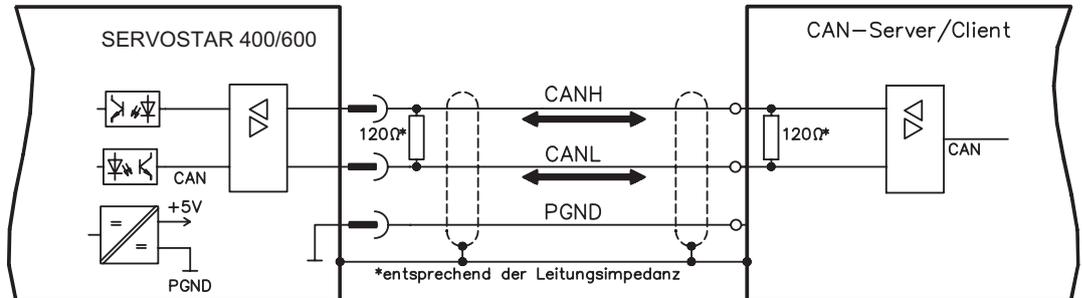
- Mit der Tastatur in der Frontplatte (siehe Produkthandbuch SERVOSTAR 400/600)
- In der Inbetriebnahme-Software DRIVE.EXE auf der Bildschirmseite "Basiseinstellungen"
- Über die serielle Schnittstelle mit der Abfolge der ASCII - Kommandos:  
CBAUD bb ⇒ SAVE ⇒ COLDSTART (mit bb = Baudrate in kB)

Mögliche Baudraten sind 10, 20, 50, 100, 125, 250, 333, 500(default), 666, 800, 1000 kBaud.

## 2.2 CANopen Interface

Interface zum Anschluss an den CAN Bus (default : 500 kBaud). Die Schnittstelle liegt auf dem gleichen Potential wie das RS232-Interface. Die analogen Sollwerteingänge sind weiterhin nutzbar.

Wir liefern auf Anfrage spezielle Klemmbuchsenstecker, die für den Busbetrieb leicht konfektioniert werden können. Darüber hinaus kann auch ein CAN-Bus – Adapter für den Optionsschacht verwendet werden (Option -2CAN-), der die Möglichkeit der Durchverdrahtung und das optionale Zuschalten des 120Ω Terminierungswiderstandes bietet. Die Pinbelegung entspricht ISO 11898 (CAN).



**Zum Potentialausgleich muss AGND mit der Steuerung verbunden werden! Weitere Hinweise siehe Produkthandbuch des Servoverstärkers.**

## 2.3 CAN Busleitung

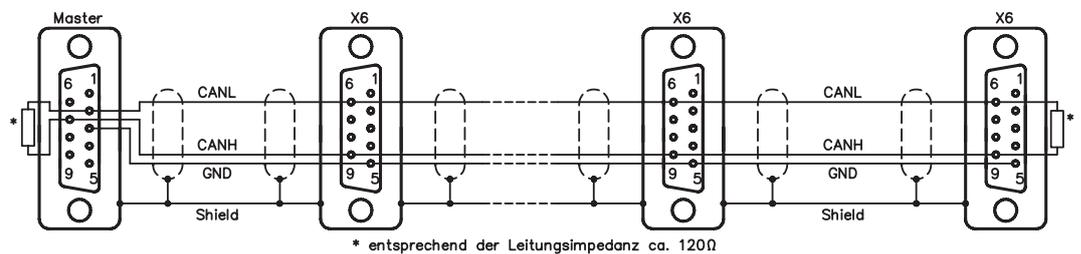
Nach ISO 11898 sollten Sie eine Busleitung mit einem Wellenwiderstand von 120 Ω verwenden. Die verwendbare Leitungslänge für eine sichere Kommunikation nimmt mit zunehmender Übertragungsrate ab. Als Anhaltspunkte können folgende bei uns gemessenen Werte dienen, die allerdings nicht als Grenzwerte zu verstehen sind:

<b>Leistungsdaten:</b>	Wellenwiderstand	100..120 Ω
	Leitungskapazität	max. 60 nF/km
	Leiterwiderstand (Schleife)	159,8 Ω/km

### Leitungslängen in Abhängigkeit von der Übertragungsrate

Übertragungsrate / kBaud	max. Leitungslänge / m
1000	20
500	70
250	115

Mit geringerer Leitungskapazität (max. 30 nF/km) und geringerem Leiterwiderstand (Schleife, 115 Ω/km) können größere Übertragungsweiten erreicht werden. (Wellenwiderstand 150 ± 5Ω ⇒ Abschlusswiderstand 150 ± 5Ω).



An das SubD-Steckergehäuse sind aus EMV-Gründen folgende Anforderung gestellt:

- metallisches oder metallisch beschichtetes Gehäuse
- Anschlussmöglichkeit für den Leitungsschirm im Gehäuse, großflächige Verbindung

Sie können von uns spezielle Klemmbuchsenstecker (Best.Nr. DE-90650) beziehen, die für den Busbetrieb leicht konfektioniert werden können. Darüber hinaus kann auch ein CAN – Bus – Adapter für den Optionsschacht verwendet werden (Option -2CAN-), der die Möglichkeit der Durchverdrahtung und das optionale Zuschalten des 120Ω Terminierungswiderstandes bietet. Die Pinbelegung entspricht dem Standard CANopen DS301.

## 2.4

## Leitfaden zur Inbetriebnahme



Nur Fachpersonal mit fundierten Kenntnissen in Regelungstechnik und Antriebstechnik darf den Servoverstärker in Betrieb nehmen.

**Montage / Installation prüfen**

Prüfen Sie, ob alle Sicherheitshinweise im Produkthandbuch des Servoverstärkers und im vorliegenden Handbuch beachtet und umgesetzt wurden. Kontrollieren Sie Stationsadresse und Baudrate.

**PC anschließen, Inbetriebnahmesoftware starten**

Zum Parametrieren des Servoverstärkers verwenden Sie die Inbetriebnahmesoftware DRIVE.EXE.

**Grundfunktionen in Betrieb nehmen**

Nehmen Sie nun die Grundfunktionen des Servoverstärkers in Betrieb und optimieren Sie Strom- und Drehzahlregler. Dieser Teil der Inbetriebnahme ist im Produkthandbuch und in der Onlinehilfe der Inbetriebnahmesoftware genauer beschrieben.

**Parameter speichern**

Speichern Sie die Parameter nach erfolgter Optimierung im Servoverstärker.

**Buskommunikation in Betrieb nehmen**

Die geänderten Parameter werden erst nach einem Software-Reset (Warmboot) wirksam. Wechseln Sie hierzu auf die Bildschirmseite "Status" und betätigen Sie den Reset-Button.  
Voraussetzung: das in Kapitel 4 beschriebene Software-Protokoll ist auf dem Master realisiert.  
Passen Sie die Baudrate des SERVOSTAR 400/600 an die des Masters an.

**Test der Kommunikation**

Vorschlag: Fordern Sie das Emergency-Objekt an.



Stellen Sie sicher, dass auch bei ungewollter Bewegung des Antriebs keine maschinelle oder personelle Gefährdung eintreten kann.

**Lageregler in Betrieb nehmen**

Nehmen Sie den Lageregler in Betrieb, wie in der Online-Hilfe der Inbetriebnahmesoftware beschrieben.

## 2.5 Wichtige Konfigurationsparameter für den CAN-Bus - Betrieb

Die folgenden Parameter sind für den CAN - Betrieb wichtig:

**1. CBAUD (Objekt 3515<sub>h</sub> Subindex 01<sub>h</sub>):** Übertragungsrate für den CAN-Bus

**2. ADDR (Objekt 3505<sub>h</sub> Subindex 01<sub>h</sub>):** Mit dem Kommando ADDR wird die Feldbus-Adresse des Verstärkers definiert. Nach der Änderung der Adresse sollten alle Parameter im EEPROM abgespeichert werden und der Verstärker aus- und eingeschaltet werden.

**3. AENA (Objekt 3506<sub>h</sub> Subindex 01<sub>h</sub>):** Hiermit kann der Zustand des Software-Enable beim Einschalten des Verstärkers definiert werden. Mit dem Software-Enable wird einer externen Steuerung die Möglichkeit gegeben die Endstufe softwaremässig zu sperren bzw. freizugeben. Bei Geräten, die mit einem analogen Sollwert arbeiten (OPMODE=1,3) wird beim Einschalten des Verstärkers das Software-Enable automatisch gesetzt, so dass diese Geräte sofort betriebsbereit sind (Hardware-Enable vorausgesetzt). Bei allen anderen Geräten wird beim Einschalten das Software-Enable auf den Wert von AENA gesetzt. Die Variable AENA hat auch eine Funktion beim Resetieren des Verstärkers nach einem Fehler (über digitalen Eingang 1 bzw. mit dem ASCII-Kommando CLRFAULT). Bei Fehlern, die softwaremässig resettiert werden können, wird, nachdem der Fehler gelöscht wurde, der Software-Enable auf den Zustand von AENA gesetzt. Auf diese Weise ist das Verhalten des Verstärkers beim Software-Reset analog zu dem Einschaltverhalten.

**4. DRVCNFG (Objekt 3672<sub>h</sub> Subindex 01<sub>h</sub>):** Mit der Konfigurationsvariable DRVCNFG können unterschiedliche CANopen-Zusatzfunktionen des Verstärkers aktiviert werden.

Bit0 =1 CANopen Einschalt-Telegramm 0 Bytes lang  
 =0 CANopen Einschalt-Telegramm 8 Bytes lang

Bit1 =1 Beeinflussung der CANopen-Zustandsmaschine durch Enable/Disable. Die CANopen-Zustandsmaschine richtet sich nach dem internen Zustand des Servoverstärkers. Bei einer Änderung dieses Zustandes (z.B. Hardware-Disable) wird die CANopen-Zustandsmaschine automatisch aktualisiert (mit entsprechender Emergency-Meldung).  
 =0 Die CANopen-Zustandsmaschine wird nicht beeinflusst.

Bit2 =1 Überprüfung der Objekt-Länge, Generierung eines Emergency-Objektes bei fehlerhafter Objekt-Länge  
 =0 die Objekt-Länge wird nicht überprüft.

Bit3 =1 Bei dieser Konfiguration können alle PDO-Mappings und Kommunikationsparameter abgespeichert werden. Dies erfolgt mit dem Objekt 1010 sub 2 (⇒ Kap. 4.2.1.16).  
 Bei Einschalten des Geräts wird dieses gespeicherte Mapping eingestellt.  
 =0 Beim Einschalten des SERVOSTAR 400/600 wird die Defaultkonfiguration der PDO - Mappings eingestellt.

**5. MDRV (Objekt 3639<sub>h</sub>, Subindex 01<sub>h</sub>):**

Mit der Konfigurationsvariablen MDRV kann der Multidrive - Modus (MDRV = 1) für die Setup-Software eingestellt werden. Dann sind nur die ersten drei TPDOs und RPDOs nutzbar. Falls alle vier TPDOs und RPDOs genutzt werden sollen, muss diese Konfigurationsvariable auf 0 gesetzt sein.

Diese Seite wurde bewusst leer gelassen.

### 3 CANopen Kommunikationsprofil

Dieses Kapitel beschreibt die grundlegenden Dienste und Kommunikationsobjekte des CANopen Kommunikationsprofils DS 301, die vom SERVOSTAR 400/600 verwendet werden.



**Es wird vorausgesetzt, dass das Kommunikationsprofil in seiner grundlegenden Funktionsweise bekannt ist und als Referenzdokumentation zur Verfügung steht.**

Da die Geschäftssprache der CiA englisch ist, werden feststehende Begriffe im Original verwendet (z.B. Objekt 1000<sub>h</sub>: Device Type).

#### 3.1 Allgemeine Erläuterungen zu CAN

Das hier verwendete Übertragungsverfahren ist in der ISO 11898 (Controller Area Network [CAN] for high-speed communication) definiert.

Das in allen CAN-Bausteinen implementierte Schicht 1/2-Protokoll (Physical Layer/Data Link Layer) stellt u. a. die Anforderung von Daten zur Verfügung.

Datentransport bzw. Datenanforderung erfolgen über ein Datentelegramm (Data Frame) mit bis zu 8 Byte Nutzdaten bzw. über ein Datenanforderungstelegramm (Remote Frame).

Kommunikationsobjekte werden durch einen 11 Bit Identifier (ID) gekennzeichnet, der auch die Priorität von Objekten bestimmt.

Um die Applikation von der Kommunikation zu entkoppeln, wurde ein Schicht-7-Protokoll (Anwendungsschicht) entwickelt. Die von der Anwendungsschicht bereitgestellten Dienstelemente ermöglichen die Realisierung einer über das Netzwerk verteilten Applikation. Diese Dienstelemente sind im "CAN Applikation Layer (CAL) for Industrial Applications" beschrieben.

Auf die CAL ist das Kommunikationsprofil CANopen und das Antriebsprofil aufgesetzt.

Das folgende Diagramm zeigt die grundlegende Struktur eines KommunikationsObjekte:

S O M	COB-ID	R T R	CTRL	Data Segment	CRC	A C K	EOM
-------------	--------	-------------	------	--------------	-----	-------------	-----

SOM	Start of message
COB-ID	COB-Identifier (11 Bit)
RTR	Remote Transmission Request
CTRL	Control Field (u.a. Data Length Code)
Data Segment	0..8 Byte (Data-COB) 0 Byte (Remote-COB)
CRC	Prüfsequenz
ACK	Acknowledge Slot
EOM	End of message

### 3.2 Aufbau des COB-Identifiers

Die folgende Grafik zeigt den Aufbau des COB – Identifiers (COB - ID). Der “Function Code” legt die Bedeutung und die Priorität des jeweiligen Objekte fest.

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

Bit 0..6 Modul ID (Stationsnummer, Bereich 1..63; wird in der Inbetriebnahmesoftware oder am Servoverstärker eingestellt, => 2.1.1)

Bit 7..10 Function Code (Nummer des im Server definierten Kommunikationsobjektes)



**Achtung: Wird eine ungültige Stationsnummer eingestellt (=0 oder >63), so wird die Modul-ID intern auf 1 gesetzt. Mit dem ASCII Objekt MDRV ist es möglich, den Adressbereich von 63 auf 127 zu expandieren.**

Die folgenden Tabellen zeigen die Default - Belegung der COB - Identifier nach dem Einschalten des Servoverstärkers. Die Objekte, die mit einem Index (Communication Parameters at Index) versehen sind, können nach der Initialisierungsphase mit einem neuen Identifier versehen werden. Die Indizes in Klammern sind optional.

Vordefinierte “Broadcast” Objekte (Senden an alle)

Objekt	Function code (binary)	Resulting COB-IDs		Communication parameters at index
		Dez.	Hex.	
NMT	0000	0	0h	—
SYNC	0001	128	80h	(1005h)
TIME	0010	256	100h	—

Vordefinierte “Peer to Peer” Objekte (Senden Station zu Station)

Objekt	Function code (binary)	Resulting COB-IDs		Communication parameters at index	Priorität
		Dez.	Hex.		
EMERGENCY	0001	129...255	81h...FFh	—	hoch  niedrig
TPDO 1	0011	385...511	181h...1FFh	1800h	
RPDO 1	0100	513...639	201h...27Fh	1400h	
TPDO 2	0101	641...767	281h...2FFh	1801h	
RPDO 2	0110	769...895	301h...37Fh	1401h	
TPDO 3	0110	897...1023	381h...3FFh	1802h	
RPDO 3	1000	1025...1151	401h...47Fh	1402h	
TPDO 4	1001	1153...1279	481h...4FFh	1803h	
RPDO 4	1010	1281...1407	501h...57Fh	1403h	
SDO (tx*)	1011	1409...1535	581h...5FFh		
SDO (rx*)	1100	1537...1663	601h...67Fh		
Nodeguard	1110	1793...1919	701h...77Fh	(100Eh)	

\* tx = Senderichtung SERVOSTAR => Master  
 rx = Senderichtung Master => SERVOSTAR

### 3.3 Definition der verwendeten Datentypen

Dieses Kapitel definiert die verwendeten Datentypen. Jeder Datentyp kann mit Hilfe von Bit -Sequenzen beschrieben werden. Diese Bit - Sequenzen werden in “Octets” (Bytes) zusammengefasst. Für numerische Datentypen wird das “Little – Endian” – Format (Intel) verwendet (siehe auch DS301 Application Layer “General Description of Data Types and Encoding Rules”).

### 3.3.1 Basisdatentypen

#### 3.3.1.1 Unsigned Integer

Daten vom Basis Datentyp UNSIGNEDn definiert ausschließlich positive Integer.

Der Wertebereich ist  $0..2^n-1$ . Die Bitsequenz  $b = b_0..b_{n-1}$  definiert den Wert

$$\text{UNSIGNED}_n(b) = b_{n-1} 2^{n-1} + \dots + b_1 2^1 + b_0 2^0$$

Beispiel: Der Wert 266 = 10A<sub>h</sub> wird mit dem Datentyp UNSIGNED16 mit Hilfe von zwei "octets" übertragen (1. octet = 0A<sub>h</sub>, 2. octet = 01<sub>h</sub>).

**Übertragungssyntax für den Datentyp UNSIGNEDn:**

octet Nummer	1.	2.	3.	4.	5.	6.	7.	8.
UNSIGNED8	b7..b0							
UNSIGNED16	b7..b0	b15..b8						
UNSIGNED24	b7..b0	b15..b8	b23..b16					
UNSIGNED32	b7..b0	b15..b8	b23..b16	b31..b24				
UNSIGNED40	b7..b0	b15..b8	b23..b16	b31..b24	b39..b32			
UNSIGNED48	b7..b0	b15..b8	b23..b16	b31..b24	b39..b32	b47..b40		
UNSIGNED56	b7..b0	b15..b8	b23..b16	b31..b24	b39..b32	b47..b40	b55..b48	
UNSIGNED64	b7..b0	b15..b8	b23..b16	b31..b24	b39..b32	b47..b40	b55..b48	b63..b56

#### 3.3.1.2 Signed Integer

Daten vom Basis Datentyp INTEGERn definiert positive und negative Integer.

Der Wertebereich ist  $-2^{n-1}..2^{n-1}-1$ . Die Bitsequenz  $b = b_0..b_{n-1}$  definiert den Wert

$$\text{INTEGER}_n(b) = b_{n-2} 2^{n-2} + \dots + b_1 2^1 + b_0 2^0 \text{ mit } b_{n-1} = 0$$

Negative Zahlen werden im zweier Komplement dargestellt - somit gilt:

$$\text{INTEGER}_n(b) = - \text{INTEGER}_n(b) - 1 \text{ mit } b_{n-1} = 1$$

Beispiel: Der Wert -266 = FEF6<sub>h</sub> wird mit dem Datentyp INTEGER16 mit Hilfe von zwei "octets" übertragen (1. octet = F6<sub>h</sub>, 2. octet = FE<sub>h</sub>).

**Übertragungssyntax für den Datentyp INTEGERn:**

octet Nummer	1.	2.	3.	4.	5.	6.	7.	8.
INTEGER8	b7..b0							
INTEGER16	b7..b0	b15..b8						
INTEGER24	b7..b0	b15..b8	b23..b16					
INTEGER32	b7..b0	b15..b8	b23..b16	b31..b24				
INTEGER40	b7..b0	b15..b8	b23..b16	b31..b24	b39..b32			
INTEGER48	b7..b0	b15..b8	b23..b16	b31..b24	b39..b32	b47..b40		
INTEGER56	b7..b0	b15..b8	b23..b16	b31..b24	b39..b32	b47..b40	b55..b48	
INTEGER64	b7..b0	b15..b8	b23..b16	b31..b24	b39..b32	b47..b40	b55..b48	b63..b56

### 3.3.2 Zusammengesetzte Datentypen

Zusammengesetzte Datentypen nutzen Basisdatentypen (INTEGERn, UNSIGNEDn, REAL). Es werden zwei Datentypen unterschieden:

- STRUCT  
Dieser Datentyp setzt sich aus unterschiedlichen Basisdatentypen zusammen.
- ARRAY  
Dieser Datentyp setzt sich aus gleichen Basisdatentypen zusammen.

### 3.3.3 Erweiterte Datentypen

Erweiterte Datentypen werden aus Basisdatentypen und zusammengesetzten Datentypen abgeleitet. Im Folgenden werden die unterstützten Datentypen definiert.

#### 3.3.3.1 Octet String

Der Datentyp "OCTET\_STRING" definiert sich aus dem Datentyp ARRAY. "Length" ist die Länge des "octet string".

ARRAY[length] OF UNSIGNED8            OCTET\_STRINGlength

#### 3.3.3.2 Visible String

Der Datentyp "VISIBLE\_" kann sich aus dem Datentyp UNSIGNED8 oder aus dem Datentyp ARRAY definieren. Die zulässigen Werte sind 00<sub>h</sub> und der Bereich von 20<sub>h</sub> bis 7E<sub>h</sub>. Die Daten werden als 7 Bit ASCII Code interpretiert (ISO 646-1973(E)). "Length" ist die Länge des "visible string".

UNSIGNED8                                VISIBLE\_CHAR  
ARRAY[length] OF VISIBLE\_CHAR        VISIBLE\_STRINGlength

### 3.4 Kommunikationsobjekte

Die Kommunikationsobjekte werden mit Hilfe von Dienstelementen und Protokollen beschrieben. Es können im wesentlichen zwei Dienstelemente unterschieden werden:

- Unbestätigte Dienste
- Bestätigte Dienste

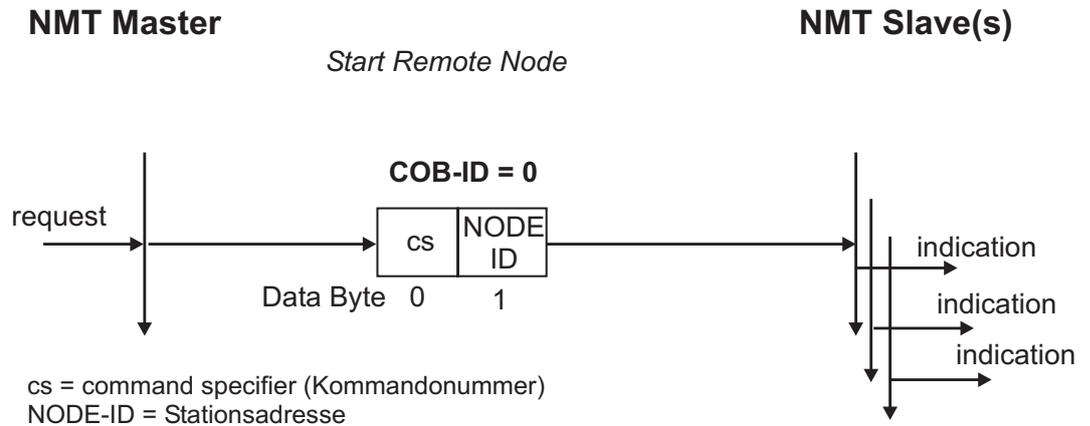
Sämtliche Dienste setzen voraus, das "Data Link" und "Physical Layer" fehlerfrei arbeiten.

Der SERVOSTAR 400/600 unterstützt die Kommunikationsobjekte, die in den folgenden Kapiteln detailliert beschrieben werden:

- Network Management Objects (NMT)
- Synchronisation Object (SYNC)
- Time Stamp Object (TIME)
- Emergency Object (EMCY)
- Process Data Object (PDO)
- Service Data Object (SDO)
- Nodeguard

### 3.4.1 Network Management Objects (NMT)

Das NMT - Telegramm sieht wie folgt aus:



Der Antrieb unterstützt folgende Funktionen des Netzwerkmanagements:

- cs = 129, reset node:** führt zu einem Kaltstart des Antriebs, damit werden sämtliche im RAM abgelegte Parameter gelöscht und wieder die im EEPROM gespeicherten Werte bzw. Defaultwerte eingestellt.
- cs = 1, start remote node:** startet den CAN-Knoten, d.h. gibt die PDOs des Antriebs zur Bedienung frei. Ab diesem Zeitpunkt werden Sende-PDOs ereignisgesteuert gesendet, bzw. kann der zyklische Prozessdatenbetrieb starten.
- cs = 2, Stop remote node:** stoppt den CAN-Knoten, d.h. der Antrieb reagiert nicht mehr auf empfangene PDOs und sendet keine mehr.

### 3.4.2 Synchronisation Object (SYNC)

Das SYNC ist ein periodisches "Broadcast" – Objekt und gibt den Basis - Bustakt vor. Um einen zeitlich äquidistanten Abstand zu ermöglichen, besitzt das SYNC eine hohe Priorität. Die Anwendung dieses Protokolls wird im Applikationsteil dieses Handbuchs veranschaulicht.

### 3.4.3 Time Stamp Object (TIME)

Dieses Kommunikationsobjekt wird vom SERVOSTAR nicht unterstützt.

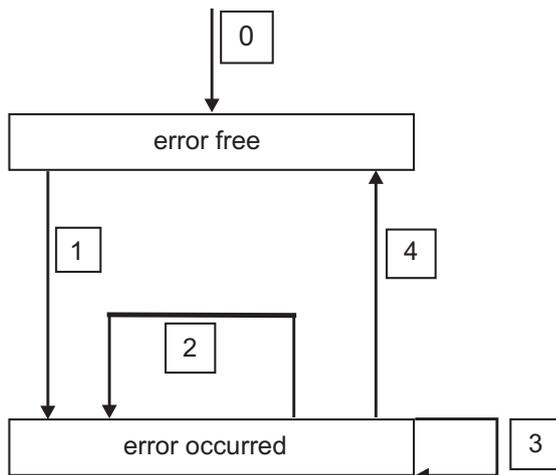
### 3.4.4 Emergency Object (EMCY)

Das EMCY wird ereignisgetriggert aufgrund einer internen Fehlersituation generiert. Für jeden Fehler wird dieses Objekt erneut übertragen. Da die Fehlercodes geräteabhängig sind, werden sie im Kapitel "CANopen Antriebsprofil" (⇒ Kap.4) beschrieben.

### 3.4.4.1 Verwendung des Emergency Object

Das Verhalten im Fehlerfall ist von der Fehlerart abhängig und somit unterschiedlich. Aus diesem Grund wird das Verhalten mit Hilfe eine Fehlerzustandsmaschine beschrieben. Es werden die Fehlerzustände "error free" und "error occurred" unterschieden. Es sind die folgenden Übergänge definiert:

0. Nach der Initialisierung wird der Zustand "error free" eingenommen, falls kein Fehler erkannt wurde. In diesem Zustand wird keine Fehlermeldung generiert.
1. Der SERVOSTAR erkennt einen internen Fehler und zeigt diesen in den ersten drei Bytes des "emergency" Telegramms an ("error code" Byte 0,1 und "error register" Byte 2). Da der SERVOSTAR mehrere Fehlerarten unterscheidet, wird Byte 3 aus dem herstellenspezifischen Fehlerfeld dazu genutzt, um die Fehlerkategorie anzuzeigen.
2. Einer, aber nicht alle Fehler sind zurückgesetzt worden. Das EMCY Telegramm beinhaltet den "Error Code" 0000<sub>h</sub>, das "Error Register" zeigt die restlichen noch anstehenden Fehler an. Der herstellenspezifische Bereich ist zu Null gesetzt.
3. Ein neuer Fehler ist aufgetreten. Der SERVOSTAR verbleibt im Zustand "Error State" und überträgt ein EMCY Objekt mit dem entsprechenden "Error Code". Der neue "Error Code" wird in den Bytes 0 und 1 eingetragen.
4. Alle Fehler sind zurückgesetzt worden. Das EMCY Telegramm beinhaltet den "Error Code" 0000<sub>h</sub>, das "Error Register" zeigt keine Fehler an. Der herstellenspezifische Bereich ist zu Null gesetzt.



### 3.4.4.2 Aufteilung des Emergency Object

Das Emergency - Objekt setzt sich aus 8 Byte zusammen und ist folgendermaßen aufgeteilt:

Byte	0	1	2	3	4	5	6	7
Inhalt	Emergency error code (=> Kap. 4.1)		Error register (Objekt 1001 <sub>h</sub> )	Kategorie	Reserviert			

Wurde ein Emergency Objekt generiert, wird im Anschluss daran der Fehlerzustand der Zustandsmaschine gemeldet (error free / error occurred), indem ein zweites Emergency - Objekt generiert wird. Hier sind nur die ersten 4 Bytes relevant (Emergency ErrorCode , Error register, Kategorie). Byte 0/1 enthält den "Error Reset Code" (0000<sub>h</sub>) und Byte 2 zeigt an, dass eventuell noch ein Fehler ansteht. Ist das Error Register 00<sub>h</sub>, ist der Fehlerzustand "error free".

Byte 3 enthält die Kategorie. Die Bedeutung der Fehlernummern (Error Code) und der Fehlerkategorie werden im Kapitel "Emergency Message" (=> Kap. 4.1) beschrieben. Das Fehlerregister wird über das Objekt 1001<sub>h</sub> "Error register" definiert.

### 3.4.5 Service Data Objects (SDO)

Mit Hilfe der SDOs wird der Zugriff auf das Objektverzeichnis realisiert. Die SDOs werden zur Parametrierung und zur Statusabfrage verwendet. Der Zugriff auf ein einzelnes Objekt erfolgt mit einem Multiplexer über Index und Subindex des Objektverzeichnisses. Es werden die folgenden Kommunikationsprotokolle vom SERVOSTAR 400/600 unterstützt:

- Initiate SDO Download Protocol
- Download SDO Segment Protocol
- Initiate SDO Upload Protocol
- Upload SDO Segment Protocol
- Abort SDO Transfer Protocol

Die Definition der einzelnen Kommunikationsdienste und der Protokolle sind dem DS301 zu entnehmen. Beispiele zur Handhabung von SDOs sind im Applikationsteil dieses Handbuches zu finden.



**Da es sich bei einem SDO um einen bestätigten Dienst handelt, muss immer auf das SDO - Antworttelegramm gewartet werden, bevor ein neues Telegramm gesendet werden darf.**

#### 3.4.5.1 Aufbau eines Service Data Object

Ein SDO setzt sich aus folgenden Komponenten zusammen:

Byte	1	2	3	4	5	6	7	8
Inhalt	rw	Index		Subindex	Daten			

1. Das Control Byte (Byte 1):  
 Mit Hilfe des Control Bytes wird bestimmt, ob über das SDO schreibend oder lesend auf den Objektverzeichniseintrag zugegriffen wird. Eine Darstellung des gesamten Objektverzeichnis des SERVOSTAR 400/600 finden Sie in Kapitel 6.3.  
 Der Datenaustausch mit dem SERVOSTAR 400/600 hält sich an den Standard "CMS multiplexed domain protocols", wie er im CAN-Standard DS 202 beschrieben wird.  
 Um Daten zu lesen, muss das Control Byte entsprechend folgender Darstellung beschrieben werden:

Bit	7	6	5	4	3	2	1	0
Inhalt	ccs*=2			0	0	0	0	0

\* ccs ⇒ client command specifier (ccs = 2 ⇒ initiate download request)

Der Wert 0100 0000 (binary) oder 40h muss im Control-Byte übertragen werden.

Der Servoverstärker antwortet entsprechend mit einem Antwortbyte:

Bit	7	6	5	4	3	2	1	0
Inhalt	scs*=2			X	n		e	s

\* scs ⇒ server command specifier (scs = 2 ⇒ initiate upload response)

n ⇒ nur gültig bei e = s = 1, Ist dies der Fall, so steht in n die Zahl der Bytes, die keine Daten enthalten.

X ⇒ beliebig

Wenn der Lesezugriff erfolgreich war, sind im Antwortbyte immer die Bits 0 und 1 auf 1 gesetzt (e=s=1)

Kodierte Byte-Länge in der SDO Antwort::

0x43 - 4 Bytes

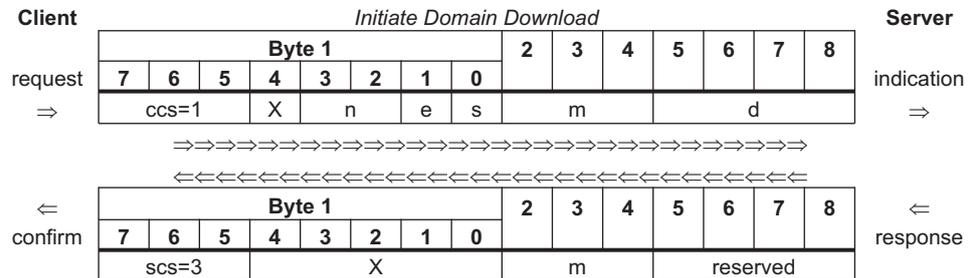
0x47 - 3 Bytes

0x4B - 2 Bytes

0x4F - 1 Byte.

Wenn ein Fehler auftritt, wird scs auf 4 gesetzt, das Antwort-Byte ist 0x80 und die Fehlerinformation ist im 4-Byte Datenfeld enthalten. Die Aufschlüsselung des Fehlers ist in Kapitel 3.4.5.6 zu finden.

Um Daten zu schreiben, muss das Control Byte entsprechend folgender Darstellung beschrieben werden:



- Der Index (Bytes 2 und 3):  
Der Index ist der Haupteintrag im Objektverzeichnis und teilt die Parameter in Gruppen ein (Beispiel: Index 2022<sub>h</sub> – Positionierdaten für den Mode Lage). Wie bei allen CAN – Daten wird der Index in umgekehrter Reihenfolge in den Bytes abgelegt, z.B.: Index 6040<sub>h</sub> -> Byte 2 = 40<sub>h</sub>, Byte 3 = 60<sub>h</sub>)
- Der Subindex (Byte 4):  
Der Subindex unterteilt die Parameter innerhalb einer Parametergruppe.
- Datenfeld (Bytes 5 – 8):  
Über diese Komponenten werden die eigentlichen Nutzdaten ausgetauscht. In den Telegrammen zur Leseaufforderung an den SERVOSTAR 400/600 sind diese auf 0 zu setzen, in der Schreibbestätigung vom SERVOSTAR 400/600 haben diese Daten bei erfolgreichem Transfer keinen Inhalt, bei fehlerhaftem Schreiben enthalten sie einen Fehlercode (⇒ Kap. 3.4.5.6).

### 3.4.5.2 Initiate SDO Download Protocol

Das "Initiate SDO Download" – Protokoll wird für Schreibzugriffe auf Objekte mit bis zu 4 Byte Nutzdaten verwendet ("expedited transfer") oder zum Einleiten eines Segmenttransfers ("normal transfer").

Der "expedited transfer" wird auch für Objekte genutzt, die nur eine Kommandoeseigenschaft aufweisen (z.B. ASCII: SAVE) und somit keine weiteren Nutzdaten benötigen.

### 3.4.5.3 Download SDO Segment Protocol

Das "Download SDO Segment" – Protokoll wird für Schreibzugriffe auf Objekte mit mehr als 4 Byte Nutzdaten verwendet ("normal transfer"). Dieser Dienst wird vom SERVOSTAR 400/600 z.Zt. nicht unterstützt, da keine Objekte existieren, die mehr als 4 Byte Nutzdaten verarbeiten.

### 3.4.5.4 Initiate SDO Upload Protocol

Das "Initiate SDO Upload" – Protokoll wird für Lesezugriffe auf Objekte mit bis zu 4 Byte Nutzdaten ("expedited transfer") oder zum Einleiten eines Segmenttransfers ("normal transfer") verwendet .

### 3.4.5.5 Upload SDO Segment Protocol

Das "Upload SDO Segment" – Protokoll wird für Lesezugriffe auf Objekte mit mehr als 4 Byte Nutzdaten verwendet ("normal transfer"). Dieser Dienst wird vom SERVOSTAR 400/600 z.Zt. nicht unterstützt, da keine Objekte existieren, die mehr als 4 Byte Nutzdaten verarbeiten.

### 3.4.5.6 Abort SDO Protocol

Das "Abort SDO" Protokoll wird bei einer fehlerhaften SDO-Übertragung vom Servostar verwendet und zeigt mit Hilfe des "abort code" (Fehlercode) den Fehler an, der zum Abbruch der Übertragung führte. Der Fehlercode ist als UNSIGNED32-Wert kodiert. Die folgende Tabelle zeigt die möglichen Ursachen für einen Abbruch:

Abort Code	Beschreibung
0503 0000 <sub>h</sub>	"Toggle Bit" wurde nicht alterniert
0504 0000 <sub>h</sub>	SDO Protokoll "time out"
0504 0001 <sub>h</sub>	Client/Server Kommando - Bezeichner nicht gültig oder unbekannt
0504 0002 <sub>h</sub>	Unbekannte Blockgröße (block mode only)
0504 0003 <sub>h</sub>	Unbekannte Blocknummer (block mode only)
0504 0004 <sub>h</sub>	CRC Fehler (block mode only)
0504 0005 <sub>h</sub>	Außerhalb des Speichers
0601 0000 <sub>h</sub>	Zugriff auf dieses Objekt wird nicht unterstützt
0601 0001 <sub>h</sub>	Versuchter Lesezugriff auf ein Objekt, das nur geschrieben werden kann
0601 0002 <sub>h</sub>	Versuchter Schreibzugriff auf ein Objekt, das nur gelesen werden kann
0602 0000 <sub>h</sub>	Objekt existiert im Objektverzeichnis nicht
0604 0041 <sub>h</sub>	Objekt kann nicht in ein PDO "gemappt" werden
0604 0042 <sub>h</sub>	Größe und Anzahl der "gemappten" Objekte übersteigt die mögliche PDO Länge
0604 0043 <sub>h</sub>	Allgemeine Parameter - Inkompatibilität
0604 0047 <sub>h</sub>	Allgemeine Inkompatibilität im Gerät
0606 0000 <sub>h</sub>	Zugriffsverletzung aufgrund eines Hardwarefehlers
0607 0010 <sub>h</sub>	Datentyp passt nicht, Länge des Service Parameters passt nicht
0607 0012 <sub>h</sub>	Datentyp passt nicht, Länge des Service Parameters zu groß
0607 0013 <sub>h</sub>	Datentyp passt nicht, Länge des Service Parameters zu klein
0609 0011 <sub>h</sub>	Subindex existiert nicht
0609 0030 <sub>h</sub>	Wertebereich des Parameters verlassen (nur für Schreibzugriffe)
0609 0031 <sub>h</sub>	Wert des Parameters zu groß
0609 0032 <sub>h</sub>	Wert des Parameters zu klein
0609 0036 <sub>h</sub>	Maximaler Wert ist kleiner als minimaler Wert
0800 0000 <sub>h</sub>	Allgemeiner Fehler
0800 0020 <sub>h</sub>	Datum kann nicht übertragen oder gespeichert werden
0800 0021 <sub>h</sub>	Datum kann aufgrund lokaler Geräteführung nicht übertragen / gespeichert werden
0800 0022 <sub>h</sub>	Datum kann aufgrund des Gerätezustandes nicht übertragen / gespeichert werden
0800 0023 <sub>h</sub>	Dynamisches Generieren des Objektverzeichnisses nicht möglich oder bereits vorhanden (Objektverzeichnis wird aus einer Datei erstellt und ein Fehler aufgrund einer defekten Datei tritt auf)

Nicht aufgeführte "Abort Codes" sind reserviert.

### 3.4.6 Process Data Object (PDO)

PDOs werden für die Echtzeit - Datenkommunikation verwendet. Zum Beispiel lassen sich über PDOs ähnliche Regler aufbauen wie sie bei analog realisierten Antrieben verwendet werden. Statt der dort verwendeten +/- 10V Sollwertvorgabe und ROD-Istwertausgabe können hier digitaler Drehzahlswert und Istpositionsangabe über PDOs realisiert werden. Die PDO-Daten werden dazu ohne Protokoll-Overhead übertragen und der Empfang nicht bestätigt.

Dieses Kommunikationsobjekt nutzt einen unbestätigten Kommunikationsdienst.

PDOs definieren sich über das Objektverzeichnis des SERVOSTAR 400/600. Hierbei können vordefinierte PDOs ausgewählt ("mapping" vordefinierter PDOs) oder vom Anwender selbst zusammengestellt (variables "mapping") werden. Das "Mapping" erfolgt mit Hilfe von SDOs während der Konfigurationsphase. Die Länge und die "Mapping" Nummer des PDOs werden über das Antriebsprofil DS 402 definiert.

Die Definition des PDO Dienstes und des Protokolls sind dem DS301 zu entnehmen. Beispiele zur Handhabung von PDOs sind im Applikationsteil dieser Dokumentation zu finden. Grundsätzlich können zwei Arten von PDOs in Abhängigkeit der Übertragungsrichtung unterschieden werden:

- Transmit PDOs (TPDOs) (SERVOSTAR ⇒ Master)  
Die TPDOs übertragen Daten vom SERVOSTAR zur Steuerung (z.B. Istwertobjekte, Gerätezustände).
- Receive PDOs (RPDOs) (Master ⇒ SERVOSTAR)  
Die RPDOs übertragen Daten von der Steuerung zum SERVOSTAR (z.B. Sollwerte)

Der SERVOSTAR 400/600 unterstützt für jede Übertragungsrichtung zwei unabhängige PDO - Kanäle. Die Kanäle sind mit den Kanalnummern 1 bis 4 gekennzeichnet. Zur Konfiguration der vier möglichen PDOs stehen jeweils drei Parameter zur Verfügung, die über entsprechende SDOs eingestellt werden können:

1. Auswahlparameter, mit denen aus einer Anzahl möglicher PDO-Konfigurationen die zwei PDOs in jeder Richtung ausgewählt werden, die im Prozessdatenbetrieb verwendet werden sollen:  
(Objekte 2600<sub>h</sub> bis 2603<sub>h</sub>, 2A00<sub>h</sub> bis 2A03<sub>h</sub>)
2. Mappingparameter, über die zum einen ermittelt werden kann, welche Daten im ausgewählten PDO vorhanden (= gemappt) sind und über die zum anderen bei den frei konfigurierbaren PDOs (PDO. Nr. 37 bis 40) das Mapping der PDOs vorgegeben werden kann
3. Kommunikationsparameter, die festlegen, ob die PDOs im synchronisierten Betrieb oder ereignisgesteuert arbeiten sollen: (Objekte 1400<sub>h</sub> bis 1403<sub>h</sub>, 1800<sub>h</sub> bis 1803<sub>h</sub>)

Darüber hinaus können einzelne Bits der TPDOs maskiert werden, um so bitweise das selbständige Generieren des Sendetriggers auf einzelne "Bitereignisse" von TPDOs zu steuern.

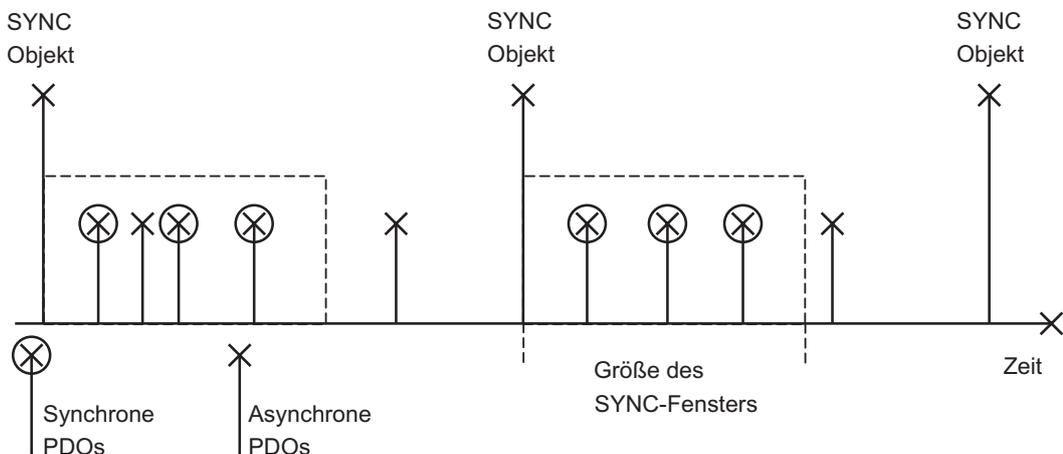
### 3.4.6.1 Übertragungsmodi

Es werden die folgenden PDO Übertragungsmodi (transmission mode) unterschieden:

- Synchroner Übertragung
- Asynchroner Übertragung

Um Antriebe zu synchronisieren, wird periodisch das vordefinierte SYNC – Objekt übertragen (Bus-takt). Synchroner PDOs werden innerhalb eines vordefinierten Zeitfensters im Anschluss an das SYNC Objekt übertragen.

Die Einstellung der Übertragungsmodi wird mit Hilfe der PDO Kommunikationsparameter vorgenommen.



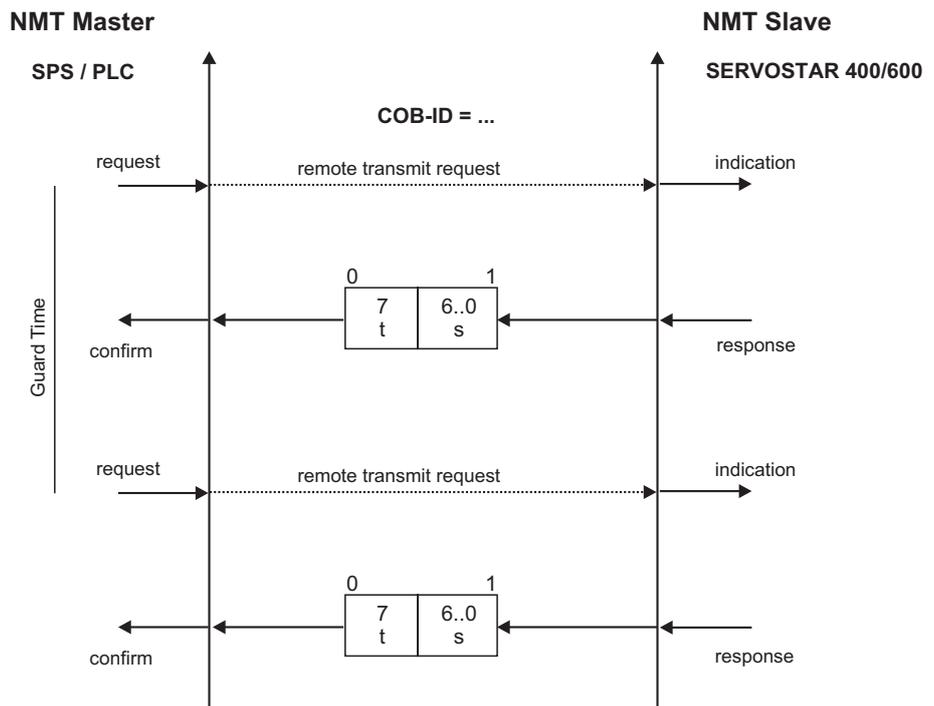
### 3.4.6.2 Triggermodi

Es werden drei Triggermodi (Triggering Modes) unterschieden:

- **Ereignisgesteuert (Event Driven)**  
Die Übertragung der Telegramme wird über ein objektpezifisches Ereignis getriggert. Zusätzlich besteht die Möglichkeit einzelne Bits objektunabhängig zu maskieren, um so das Selbständige generieren von Telegrammen einzuschränken und somit die Buslast zu senken (⇒ Kap. 4.3.2.2.4 f).
- **Zeitgesteuert (Time Driven)**  
Bei ereignisgesteuerten Signalen, die eine zu hohe Buslast erzeugen würden, kann mit Hilfe der "inhibit time" (Kommunikationsparameter, Subindex 3) bestimmt werden, nach welcher Mindestzeit wieder ein PDO gesendet werden darf.

### 3.4.7 Nodeguard

Das Node Guarding Protokoll dient der Funktionsüberwachung des Antriebs. Dazu wird der Antrieb in äquidistanten Zeitabständen vom CANopen - Master angesprochen. Der maximale zeitliche Abstand, der zwischen zwei Nodeguard - Telegrammen liegen darf, wird durch das Produkt der Guard Time (Objekt 100C<sub>h</sub>, ⇒ Kap. 4.2.1.12) und des Life Time Factors (Objekt 100D<sub>h</sub>, ⇒ Kap. 4.2.1.13) bestimmt. Ist einer dieser beiden Werte 0, ist die Ansprechüberwachung deaktiviert. Das Nodeguarding wird nur bei enableter Endstufe aktiviert. Wird der Antrieb innerhalb der Zeit, die durch die Objekte 100C<sub>h</sub> und 100D<sub>h</sub> definiert ist, nicht angesprochen, tritt die Warnung N04 (Ansprechüberwachung) am Antrieb auf, der Antrieb wird mittels Quickstop - Rampe abgebremst und jede weitere Bewegung wird verhindert. Die zeitliche Folge des Nodeguardings sieht wie folgt aus:



t = toggle Bit, ändert seinen Zustand mit jedem Slave-Telegramm  
s = Zustand der NMT Slave - Zustandsmaschine

Das Nodeguarding wird vom Master mit RTR - Telegrammen mit dem COB-ID ausgeführt, der über das Objekt 100E<sub>h</sub> (⇒ Kap. 4.2.1.14) beim Slave eingestellt werden kann. Der Defaultwert hierfür ist 700<sub>h</sub> + Slave - Knotenadresse.

Diese Seite wurde bewusst leer gelassen.

## 4 CANopen Antriebsprofil

### 4.1 Emergency Messages

“Emergency Messages” werden durch interne Gerätefehler ausgelöst. Sie haben eine hohe ID – Priorität, um einen schnellen Buszugriff sicherzustellen. Die “Emergency Message” beinhaltet ein Fehlerfeld mit vordefinierten Fehlernummern (2 Byte), Fehlerregister (1Byte) und die Fehlerkategorie (1Byte) und zusätzlichen Informationen (⇒ Kap. 3). Das höherwertige Byte der Fehlernummer gibt die Fehlerklasse und das niederwertige Byte die Fehlernummer der Klasse an. Fehlernummern von xx00<sub>h</sub> bis xx7F<sub>h</sub> sind im Kommunikations- oder Antriebsprofil definiert. Fehlernummern von xx80<sub>h</sub> bis xxFF<sub>h</sub> sind herstellerspezifische Definitionen. Mit Hilfe der Fehlerkategorie können auftretende Fehler in ihrer Bedeutung klassifiziert werden. Es sind die folgenden Fehlerkategorien definiert:

- 1: Fehler, die nur durch einen Reset (Kommando “COLDSTART” oder Bit 7 im controlword ⇒ 4.4.2.1) gelöscht werden können. Diese Fehler werden auch durch Blinken der LED - Anzeige in der Frontplatte angezeigt. (Fxx, xx = Fehlernummer)
- 2: Fehler, die durch Bit 11 im controlword (⇒ Kap. 4.4.2.1) gelöscht werden können.
- 3: Fehlermeldungen, die bei der Verarbeitung eines PDOs auftreten können.
- 4: Fehler, die **nicht** durch den Anwender behoben werden können.
- 5: Bedienungsfehler / Warnungen

Die folgende Tabelle beschreibt die verschiedenen Error Codes:

Error Code	Kategorie	Beschreibung
0000h	—	Error reset or no error (mandatory)
1000h	—	Generic error (mandatory)
1080h	5	Kein BTB (Zustand "not ready for operation")
2330h	2	Erdschluß (F22)
3100h	1	Kein Netz-BTB (F16)
3110h	1	Überspannung Zwischenkreis (F02)
3120h	1	Unterspannung Zwischenkreis (F05)
3130h	1	Eine Netzphase fehlt (bei PMODE = 2) (F19)
4110h	1	Umgebungstemperatur überschritten (F13)
4210h	1	Kühlkörpertemperatur überschritten (F01)
4310h	1	Motortemperatur überschritten (F06)
5111h	1	Fehler ±15VHilfsspannung (F07)
5380h	1	Fehler A/D-Konverter (F17)
5400h	1	Endstufenfehler (F14)
5420h	1	Bremse (Chopper) (F18)
5441h	1	Fehler bei Bedienung der AS - Option (F27)
5530h	1	Seriellles EEPROM (F09)
5581h	1	Flash-EEPROM (F10)
6010h	4	Watchdog (Softwarereset, F32)
6181h	4	BCC Fehler (Table)
6182h	4	BCC Fehler (System Makro)
6183h	4	BCC Fehler (EEPROM seriell)
6184h	4	FPGA Fehler
6185h	4	Fehler (Table)
6281h	4	User Software BCC (Makro, F32)
6282h	4	Fehlerhafte User Software (Makro, F32)
6320h	3	Parameterfehler
7111h	1	Bremsenfehler (F11)
7122h	1	Kommutierungsfehler (F25)
7181h	5	SERVOSTAR konnte nicht enabled werden
7182h	5	Kommando nur im Zustand "disabled" möglich
7303h	1	Fehler Rückführeinheit (F04)
8053h	1	Handlingfehler (F21)
8181h	2	Ansprechüberwachung aktiv
8182h	1	CAN - Busoff (F23)
8281h	5	Zustandsmaschine ist nicht im "Operational Enable"-Zustand
8282h	5	Fehlerhafte Modeeinstellung
8331h	1	$I^2t$ (torque fault, F15)
8480h	1	Überdrehzahl (overspeed, F08)
8611h	2	Schleppfehler
8681h	5	Ungültige Fahrauftragsnummer
8682h	2	Fehler externe Trajektorie (F28) (nur Sercos)
FF01h	4	Schwerwiegender Ausnahmefehler (F32)
FF02h	3	Fehler in PDO-Komponente
FF03h	5	Betriebsmode
FF04	1	Slotfehler (F20)
FF06	2	Warnungsanzeige als Fehler (F24)
FF07	2	Fehler Referenzfahrt (HW-Endschalter erreicht) (F26)
FF08	2	Sercos - Fehler (F29)
FF10	1	Motorphasen-Fehler (F12)
FF11	1	Sercos

## 4.2 Allgemeine Definitionen

In diesem Kapitel werden allgemeingültige Objekte beschrieben (z.B. Objekt 1000<sub>h</sub> "Device Type"). Im Anschluss daran wird die freie Konfiguration von Prozessdatenobjekten ("free mapping") erläutert.

### 4.2.1 Allgemeine Objekte

#### 4.2.1.1 Objekt 1000h: Device Type (DS301)

Dieses Objekt beschreibt den Gerätetyp (Servoantrieb) und die Gerätefunktionalität (DS402 Antriebsprofil). Es setzt sich folgendermaßen zusammen:

MSB		LSB	
zusätzliche Informationen		Geräte-Profil-Nummer	
Modebits	Typ	402 <sub>d</sub>	
31	24 23	16 15	0

Nummer des Geräteprofils: 402<sub>d</sub> (Antriebsprofil)  
 Typ: 02 (Servoantrieb)  
 Modebits: 0 (Herstellerspezifisch)

<b>Index</b>	1000 <sub>h</sub>
<b>Name</b>	Gerätetyp
<b>Objektcode</b>	VAR
<b>Datentyp</b>	UNSIGNED32
<b>Kategorie</b>	M

<b>Zugriff</b>	ro
<b>PDO Mapping</b>	nicht möglich
<b>Wertebereich</b>	UNSIGNED32
<b>Defaultwert</b>	nein

#### 4.2.1.2 Objekt 1001h: Error register (DS301)

Wird ein Fehlerbit im herstellerunabhängigen Fehlerregister gesetzt, so werden detailliertere Informationen im Objekt 1003<sub>h</sub> zur Verfügung gestellt. Dieses Objekt ist Teil des FehlerObjekte (Emergency Message).

<b>Index</b>	1001 <sub>h</sub>
<b>Name</b>	Error register
<b>Objektcode</b>	VAR
<b>Datentyp</b>	UNSIGNED8
<b>Kategorie</b>	M

<b>Zugriff</b>	ro
<b>PDO Mapping</b>	nicht möglich
<b>Wertebereich</b>	UNSIGNED8
<b>Defaultwert</b>	nein

Im Folgenden wird die Bitbelegung des Fehlerregisters beschrieben. Ein gesetztes Bit zeigt ein Fehlerereignis an.

Bit	Beschreibung	Bit	Beschreibung
0	generic error	4	communication error
1	current	5	device profile specific
2	voltage	6	reserved
3	temperature	7	manufacturer specific

### 4.2.1.3 Objekt 1002h: Manufacturer Status Register (DS301)

Dieses Objekt beinhaltet das herstellerspezifische Statusregister, in dem auch die SERVOSTAR Warnungen enthalten sind (s. auch ASCII Objekt DRVSTAT).

<b>Index</b>	1002h
<b>Name</b>	Manufacturer Status Register
<b>Objektcode</b>	VAR
<b>Datentyp</b>	UNSIGNED8
<b>Kategorie</b>	M

<b>Zugriff</b>	ro
<b>PDO Mapping</b>	nicht möglich
<b>Wertebereich</b>	UNSIGNED8
<b>Defaultwert</b>	nein

Die folgende Tabelle zeigt die Bitbelegung des Statusregisters:

Bit	Beschreibung
0	Warnung 1: I <sup>2</sup> t-Meldeschwelle überschritten
1	Warnung 2: Bremsleistung erreicht
2	Warnung 3: Schleppfehler
3	Warnung 4: Ansprechüberwachung aktiv
4	Warnung 5: Netzphase fehlt
5	Warnung 6: Software-Endschalter 1 hat angesprochen
6	Warnung 7: Software-Endschalter 2 hat angesprochen
7	Warnung 8: Fehlerhafter Fahrauftrag gestartet
8	Warnung 9: Beim Fahrauftrag-Start war kein Referenzpunkt gesetzt
9	Warnung 10: PSTOP aktiv
10	Warnung 11: NSTOP aktiv
11	Warnung 12: Motordefaultwerte wurden geladen
12	Warnung 13: Erweiterungskarte arbeitet nicht ordnungsgemäß
13	Warnung 14: Motorphase
14	Warnung 15: fehlerhafter VCT-Eintrag angewählt
15	Warnung 16: Reserve
16	Fahrauftrag aktiv
17	Referenzpunkt gesetzt
18	Aktuelle Position = Home Position
19	In Position
20	Position-Latch erfolgt (positive Flanke)
21	Position 0 erreicht
22	Position 1 erreicht
23	Position 2 erreicht
24	Position 3 erreicht
25	Position 4 erreicht
26	Initialisierung beendet
27	Position 5 erreicht
28	Drehzahl = 0
29	Sicherheitsrelais hat angesprochen
30	Endstufe freigegeben
31	Fehler steht an

#### 4.2.1.4 Objekt 1003h: Pre-Defined Error Field (DS301)

Dieses Objekt beinhaltet das letzte Fehlerereignis, das über ein Fehlertelegramm gemeldet wurde. Es wird nur der letzte Fehler angezeigt.

1. Der Wert unter Subindex 00<sub>h</sub> zeigt die Anzahl der aufgezeichneten Fehler an.
2. Der aktuellste Fehler steht unter Subindex 01<sub>h</sub>.
3. Wird der Wert 0 in Subindex 02<sub>h</sub> geschrieben, wird der Fehler aus der Fehlerliste gelöscht (Achtung: Fehler wird nicht quittiert!).
4. Die Fehlernummer ist vom Datentyp UNSIGNED32 und setzt sich aus einer 16 Bit Fehlernummer und einem herstellerspezifischen Informationsfeld zusammen. Das herstellerspezifische Informationsfeld wird vom SERVOSTAR 400/600 nicht benutzt. Die möglichen Fehlernummern sind im Kapitel "Error Message" (⇒ Kap. 4.1) beschrieben. Das Datenfeld baut sich folgendermaßen auf:

MSB	LSB
Informationsfeld	Fehlernummer

<b>Index</b>	1003 <sub>h</sub>
<b>Name</b>	Error Field
<b>Objektcode</b>	ARRAY
<b>Datentyp</b>	UNSIGNED32
<b>Kategorie</b>	O

Beschreibung der Subindizes:

<b>Subindex</b>	00 <sub>h</sub>
<b>Beschreibung</b>	Anzahl der Fehler
<b>Kategorie</b>	M
<b>Zugriff</b>	rw
<b>PDO Mapping</b>	nicht möglich
<b>Wertebereich</b>	0..254
<b>Defaultwert</b>	nein

<b>Subindex</b>	01 <sub>h</sub>
<b>Beschreibung</b>	Standard Fehlerfeld (⇒ Kap. 4.1)
<b>Kategorie</b>	M
<b>Zugriff</b>	ro
<b>PDO Mapping</b>	nicht möglich
<b>Wertebereich</b>	UNSIGNED32
<b>Defaultwert</b>	nein

#### 4.2.1.5 Objekt 1004h: Zahl der unterstützten PDOs (DS301)

Dieses Objekt wird nur aus Kompatibilitätsgründen zu älteren Ausgaben des CANopen-Standards DS 301 unterstützt.

Parameterstruktur:

Subindex	MSB	LSB
0	Zahl der unterstützten RPDOs	Zahl der unterstützten TPDOs
1	Zahl der unterstützten synchronen RPDOs	Zahl der unterstützten synchronen TPDOs
2	Zahl der unterstützten asynchronen RPDOs	Zahl der unterstützten asynchronen TPDOs

<b>Index</b>	1004 <sub>h</sub>
<b>Name</b>	Zahl der unterstützten PDOs
<b>Objektcode</b>	ARRAY
<b>Datentyp</b>	UNSIGNED32
<b>Kategorie</b>	0

Beschreibung der Subindizes:

<b>Subindex</b>	00 <sub>h</sub>
<b>Beschreibung</b>	Zahl der unterstützten PDOs
<b>Kategorie</b>	0
<b>Zugriff</b>	ro
<b>PDO Mapping</b>	nicht möglich
<b>Wertebereich</b>	00 <sub>h</sub> - 1FF01FF <sub>h</sub>
<b>Defaultwert</b>	00020002 <sub>h</sub>

<b>Subindex</b>	01 <sub>h</sub>
<b>Beschreibung</b>	Zahl der synchronen PDOs
<b>Kategorie</b>	0
<b>Zugriff</b>	ro
<b>PDO Mapping</b>	nicht möglich
<b>Wertebereich</b>	00 <sub>h</sub> - 1FF01FF <sub>h</sub>
<b>Defaultwert</b>	00 <sub>h</sub>

<b>Subindex</b>	02 <sub>h</sub>
<b>Beschreibung</b>	Zahl der asynchronen PDOs
<b>Kategorie</b>	0
<b>Zugriff</b>	ro
<b>PDO Mapping</b>	nicht möglich
<b>Wertebereich</b>	00 <sub>h</sub> - 1FF01FF <sub>h</sub>
<b>Defaultwert</b>	00020002 <sub>h</sub>

Beschrieben ist hier nur die Verteilung nach dem Hochlauf des SERVOSTAR 400/600. Danach können die PDOs über die Kommunikationsparameter 1400<sub>h</sub> bis 1403<sub>h</sub>, 1800<sub>h</sub> bis 1803<sub>h</sub> konfiguriert werden.

#### 4.2.1.6 Objekt 1005h: COB-ID der SYNC - Message (DS301)

Mit diesem Objekt kann der COB - ID für die SYNC - Message geändert werden.

<b>Index</b>	1005h
<b>Name</b>	COB - ID der SYNC message
<b>Objektcode</b>	VAR
<b>Datentyp</b>	UNSIGNED32
<b>Kategorie</b>	O
<b>Zugriff</b>	rw
<b>PDO Mapping</b>	nicht möglich
<b>Wertebereich</b>	UNSIGNED32
<b>Defaultwert</b>	80000080h

Parameterstruktur:

MSB										LSB					
X	0/1	0	00h							11 Bit identifier					
31	30	29	28						11	10	0				

Nur die Bits 0 - 10, 29 und 30 haben dabei eine Bedeutung:

Bit	Wert	Bedeutung
31	X	—
30	0	Gerät erzeugt keine SYNC-Message
	1	Gerät erzeugt SYNC-Message
29	0	11 Bit ID
	1	29 Bit ID
28..11	X	Bit 11..28 von 29 Bit SYNC COB-ID
10..0	X	Bit 0..10 von SYNC COB-ID

#### 4.2.1.7 Objekt 1006h: Communication Cycle Period (DS301)

Mit diesem Objekt kann die Periode des Sendens des SYNC - Telegramms in  $\mu$ s festgelegt werden.

<b>Index</b>	1006h
<b>Name</b>	Periode des Kommunikationszyklus
<b>Objektcode</b>	VAR
<b>Datentyp</b>	UNSIGNED32
<b>Kategorie</b>	O
<b>Zugriff</b>	rw
<b>PDO Mapping</b>	nicht möglich
<b>Wertebereich</b>	UNSIGNED32
<b>Defaultwert</b>	00h

**4.2.1.8 Objekt 1007h: Synchronous window length (DS301)**

Dieses Objekt legt die Länge des Zeitfensters für synchrone PDOs in  $\mu\text{s}$  fest.

<b>Index</b>	1007h
<b>Name</b>	Zeitfenster für synchrone Nachrichten
<b>Objektcode</b>	VAR
<b>Datentyp</b>	UNSIGNED32
<b>Kategorie</b>	O
<b>Zugriff</b>	rw
<b>PDO Mapping</b>	nicht möglich
<b>Wertebereich</b>	UNSIGNED32
<b>Defaultwert</b>	00h

**4.2.1.9 Objekt 1008h: Manufacturer Device Name (DS301)**

Der Gerätenamen besteht aus vier ASCII-Zeichen und hat die Buchstaben "SDxx" zum Inhalt, wobei xx für die Stromstärke der Endstufe steht (z.B. SD06).

<b>Index</b>	1008h
<b>Name</b>	Manufacturer Device Name
<b>Objektcode</b>	VAR
<b>Datentyp</b>	Visible String
<b>Kategorie</b>	Optional
<b>Zugriff</b>	const
<b>PDO Mapping</b>	nicht möglich
<b>Wertebereich</b>	nein
<b>Defaultwert</b>	nein

**4.2.1.10 Objekt 100Ah: Manufacturer Software Version (DS301)**

Die Interface-Softwareversion besteht aus vier ASCII-Zeichen (z.B. 3.00).

<b>Index</b>	100Ah
<b>Name</b>	Manufacturer Software Version
<b>Objektcode</b>	VAR
<b>Datentyp</b>	Visible String
<b>Kategorie</b>	Optional
<b>Zugriff</b>	const
<b>PDO Mapping</b>	nicht möglich
<b>Wertebereich</b>	nein
<b>Defaultwert</b>	nein

#### 4.2.1.11 Objekt 100Bh: Node-ID (DS301)

Über das Objekt "Node-ID" kann die Teilnehmeradresse des SERVOSTAR 400/600 ausgelesen werden. Die Teilnehmeradresse kann unter anderem über das ASCII-Objekt "ADDR" verändert werden. Der einzustellende Adressbereich (Wertebereich) ist abhängig vom ASCII – Objekt "MDRV". Die Teilnehmeradresse kann von 1..63 oder von 1..127 eingestellt werden.



**Wird der expandierte Adressbereich verwendet, kann die Inbetriebnahmesoftware nicht mehr über den CAN – Bus vernetzt werden.**

Index	100B <sub>h</sub>
Name	Node-ID
Objektcode	VAR
Datentyp	UNSIGNED32
Kategorie	Optional

Zugriff	ro
PDO Mapping	nicht möglich
Wertebereich	1..127 (1..63)
Defaultwert	1

#### 4.2.1.12 Objekt 100Ch: Guard Time (DS301)

Das Produkt der Objekte 100C<sub>h</sub> "Guard Time" und 100D<sub>h</sub> "Life Time Factor" ergibt die Ansprechüberwachungszeit. Die "Guard Time" wird in Millisekunden angegeben. Die Ansprechüberwachung wird erst mit dem ersten "Nodeguard - Objekt" aktiv (⇒ Kap. 3.4.7). Wird der Wert des Objekte "Guard Time" zu Null gesetzt, ist die Ansprechüberwachung inaktiv.

Index	100C <sub>h</sub>
Name	Guard Time
Objektcode	VAR
Datentyp	UNSIGNED16
Kategorie	Mandatory

Zugriff	ro
PDO Mapping	nicht möglich
Wertebereich	UNSIGNED16
Defaultwert	0

#### 4.2.1.13 Objekt 100Dh: Life Time Factor (DS301)

Das Produkt der Objekte "Guard Time" und "Life Time Factor" ergibt die Ansprechüberwachungszeit. Die Ansprechüberwachung wird erst mit dem ersten "Nodeguard - Objekt" aktiv (⇒ Kap.3.4.7). Wird der Wert des Objekte "Life Time Factor" zu Null gesetzt, ist die Ansprechüberwachung inaktiv.

Index	100D <sub>h</sub>
Name	Life Time Factor
Objektcode	VAR
Datentyp	UNSIGNED8
Kategorie	Mandatory

Zugriff	rw
PDO Mapping	nicht möglich
Wertebereich	0..255
Defaultwert	0

**4.2.1.14 Objekt 100Eh: COB - ID Nodeguarding (DS301)**

Dieses Objekt legt den COB - ID für das Nodeguarding - Protokoll fest.

<b>Index</b>	100E <sub>h</sub>
<b>Name</b>	Node guarding identifier
<b>Objektcode</b>	VAR
<b>Datentyp</b>	UNSIGNED32
<b>Kategorie</b>	O
<b>Zugriff</b>	rw
<b>PDO Mapping</b>	nicht möglich
<b>Wertebereich</b>	UNSIGNED32
<b>Defaultwert</b>	700 + node - ID

**4.2.1.15 Objekt 100Fh: Zahl der unterstützten Objekte (DS301)**

Dieses Objekt beschreibt die Zahl der vom Gerät unterstützten Objekte.

<b>Index</b>	100F <sub>h</sub>
<b>Name</b>	Zahl der unterstützten Objekte
<b>Objektcode</b>	VAR
<b>Datentyp</b>	UNSIGNED32
<b>Kategorie</b>	O
<b>Zugriff</b>	ro
<b>PDO Mapping</b>	nicht möglich
<b>Wertebereich</b>	UNSIGNED32
<b>Defaultwert</b>	01 <sub>h</sub>

#### 4.2.1.16 Objekt 1010h: Store Parameters (DS301)

Mit Hilfe dieses Objekte können die Kommunikations- und Mappingparameter abgespeichert werden.

<b>Index</b>	1010h
<b>Name</b>	store parameters
<b>Objektcode</b>	ARRAY
<b>Datentyp</b>	UNSIGNED32
<b>Kategorie</b>	O

Beschreibung der Subindizes:

<b>Subindex</b>	0h
<b>Beschreibung</b>	Anzahl der Einträge
<b>Zugriff</b>	ro
<b>PDO Mapping</b>	nicht möglich
<b>Wertebereich</b>	2
<b>Defaultwert</b>	2

<b>Subindex</b>	1h
<b>Beschreibung</b>	save all parameters
<b>Zugriff</b>	rw
<b>PDO Mapping</b>	nicht möglich
<b>Wertebereich</b>	UNSIGNED32
<b>Defaultwert</b>	0

Dieser Subindex ist reserviert.

<b>Subindex</b>	2h
<b>Beschreibung</b>	save communication parameters
<b>Zugriff</b>	rw
<b>PDO Mapping</b>	nicht möglich
<b>Wertebereich</b>	UNSIGNED32
<b>Defaultwert</b>	0

Zum Abspeichern der Kommunikations- und Mappingparameter muss Subindex 2 verwendet werden. Es muss ein schreibender Zugriff mit der Schreibsignatur "save" erfolgen, d.h. Sie müssen folgenden Wert schreiben:

<b>MSB</b>			<b>LSB</b>
e	v	a	s
65h	76h	61h	73h

Ein Abspeichern der Parameter kann nur erfolgen, wenn diese Funktion über den Parameter DRVCNFG (Objekt 6372, Subindex 1) Bit 3 freigegeben ist.

**4.2.1.17 Objekt 1012h: COB - ID für Time - Stamp Message (DS301)**

Mit diesem Objekt kann der COB - ID für die Time - stamp - Message definiert werden.

<b>Index</b>	1012h
<b>Name</b>	COB-ID time stamp message
<b>Objektcode</b>	VAR
<b>Datentyp</b>	UNSIGNED32
<b>Kategorie</b>	O
<b>Zugriff</b>	rw
<b>PDO Mapping</b>	nicht möglich
<b>Wertebereich</b>	UNSIGNED32
<b>Defaultwert</b>	100h

**4.2.1.18 Objekt 1013h: High Resolution Time Stamp (i.V.) (DS301)**

Mit diesem Objekt kann ein Zeitwert gelesen oder geschrieben werden.

<b>Index</b>	1013h
<b>Name</b>	high resolution time stamp
<b>Objektcode</b>	VAR
<b>Datentyp</b>	UNSIGNED32
<b>Kategorie</b>	O
<b>Zugriff</b>	rw
<b>PDO Mapping</b>	nicht möglich
<b>Wertebereich</b>	UNSIGNED32
<b>Defaultwert</b>	00h

**4.2.1.19 Objekt 1014h: COB - ID für Emergency Message (DS301)**

Mit diesem Objekt kann der COB - ID für die Emergency definiert werden.

<b>Index</b>	1014h
<b>Name</b>	COB-ID emergency message
<b>Objektcode</b>	VAR
<b>Datentyp</b>	UNSIGNED32
<b>Kategorie</b>	O
<b>Zugriff</b>	rw
<b>PDO Mapping</b>	nicht möglich
<b>Wertebereich</b>	UNSIGNED32
<b>Defaultwert</b>	80h + Node - ID

#### 4.2.1.20 Objekt 1018h: Identity Object (DS301)

Das "Identity Objekt" beinhaltet allgemeine Geräteinformationen.

<b>Index</b>	1018h
<b>Name</b>	Identity Object
<b>Objektcode</b>	RECORD
<b>Datentyp</b>	Identity

Beschreibung der Subindizes:

<b>Subindex</b>	00h
<b>Beschreibung</b>	Anzahl der Einträge
<b>Zugriff</b>	ro
<b>PDO Mapping</b>	nicht möglich
<b>Wertebereich</b>	1..4
<b>Defaultwert</b>	4

<b>Subindex</b>	01h
<b>Beschreibung</b>	Vendor ID*
<b>Zugriff</b>	ro
<b>PDO Mapping</b>	nicht möglich
<b>Wertebereich</b>	UNSIGNED32
<b>Defaultwert</b>	6Ah

\* 00 00 00 6Ah

<b>Subindex</b>	02h
<b>Beschreibung</b>	Product Code*
<b>Zugriff</b>	ro
<b>PDO Mapping</b>	nicht möglich
<b>Wertebereich</b>	UNSIGNED32
<b>Defaultwert</b>	nein

\* Stromstärke des Verstärkers

<b>Subindex</b>	03h
<b>Beschreibung</b>	Revision Number*
<b>Zugriff</b>	ro
<b>PDO Mapping</b>	nicht möglich
<b>Wertebereich</b>	UNSIGNED32
<b>Defaultwert</b>	nein

\* Bit 16..31 Firmware Version, Bit 15..0 CANopen Version

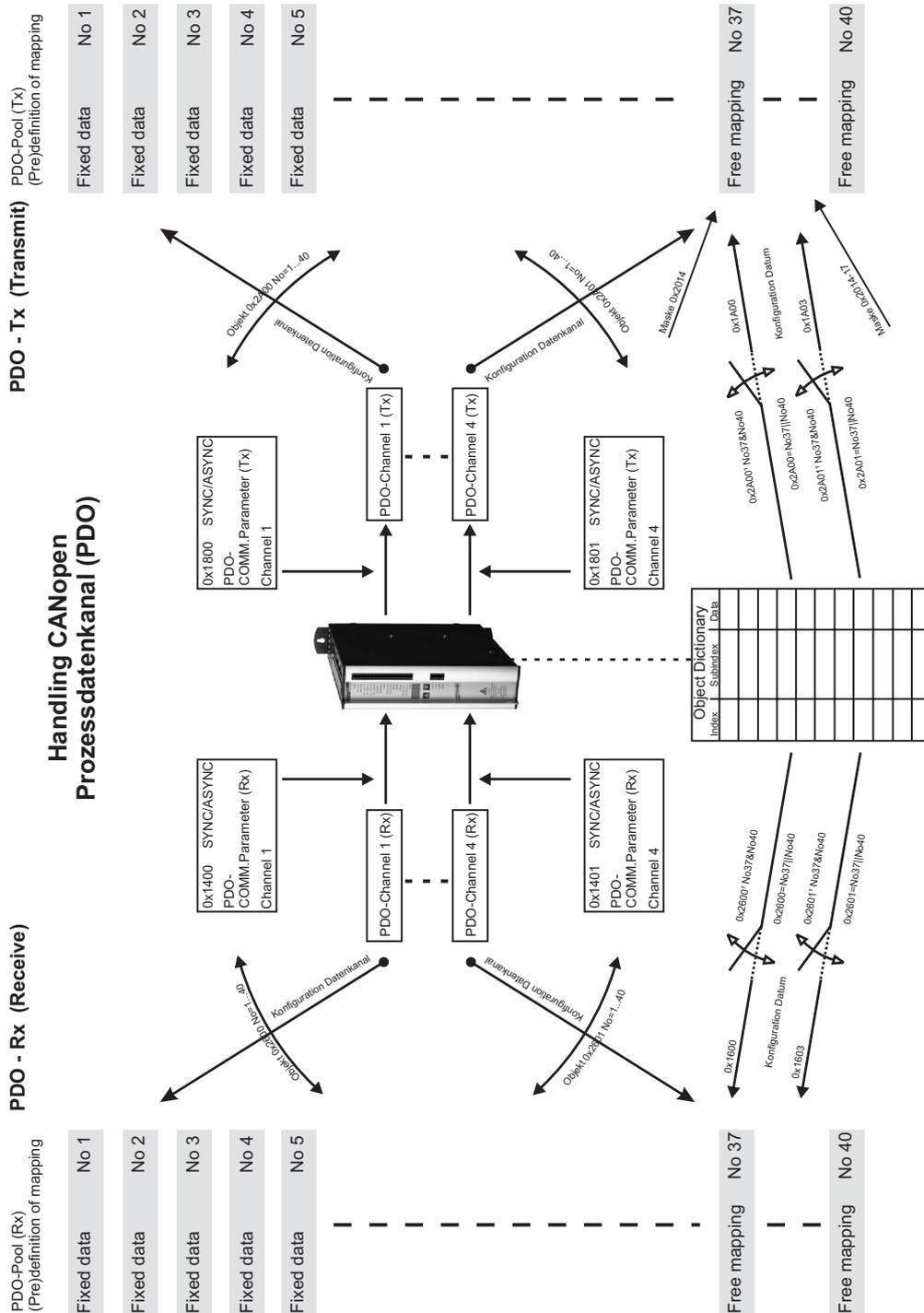
<b>Subindex</b>	04h
<b>Beschreibung</b>	Serial Number
<b>Zugriff</b>	ro
<b>PDO Mapping</b>	nicht möglich
<b>Wertebereich</b>	UNSIGNED32
<b>Defaultwert</b>	nein

### 4.3 PDO Mapping

Da der SERVOSTAR 400/600 mehr als einen Betriebsmode unterstützt, werden unterschiedlich konfigurierte PDOs für die Send- und Empfangsrichtung - in Abhängigkeit der Anwendung - benötigt.

Die erste Lösung für die Handhabung von frei konfigurierbaren PDOs im SERVOSTAR war das "quasi freie Mapping". Bei dieser Art von freiem Mapping werden vordefinierte PDOs mit Hilfe eines Index aus der PDO Bibliothek ausgewählt. Diese PDOs können inhaltlich nicht mehr umkonfiguriert werden. Das "quasi freie Mapping" hat sich jedoch als zu statisch für die wachsende Anzahl unterschiedlichster Anwendungen gezeigt. Daher wurde das "vollständig freie Mapping" für PDOs zusätzlich eingeführt. Hierbei kann der Anwender das PDO auch inhaltlich verändern (⇒ Kap.3.4.6).

Die folgende Abbildung "Handling CANopen Prozessdatenkanal" veranschaulicht die Handhabung von PDOs.



### 4.3.1 Empfangs-PDOs (RPDO)

Mit den Objekte 2600<sub>h</sub> bis 2603<sub>h</sub> können RPDOs (R=receive) ausgewählt werden. Die Konfiguration der RPDOs erfolgt mit den Objekte 1400<sub>h</sub> bis 1403<sub>h</sub> (Kommunikationsparameter) und 1600<sub>h</sub> bis 1603<sub>h</sub>(Mapping - Parameter). Diese Objekte arbeiten kanalbezogen d. h.:

- RPDO Kanal 1: 2600<sub>h</sub>, 1400<sub>h</sub>, 1600<sub>h</sub>
- RPDO Kanal 2: 2601<sub>h</sub>, 1401<sub>h</sub>, 1601<sub>h</sub>
- RPDO Kanal 3: 2602<sub>h</sub>, 1402<sub>h</sub>, 1602<sub>h</sub>
- RPDO Kanal 4: 2603<sub>h</sub>, 1403<sub>h</sub>, 1603<sub>h</sub>

Die Objekte 1600<sub>h</sub> bis 1603<sub>h</sub> können nur gelesen und so die Objektkonfiguration innerhalb des RPDOs ermittelt werden.



#### Ausnahme:

**Wurden einer oder mehrere der RPDOs 37 bis 40 ausgewählt, muss die Objektkonfiguration für den jeweiligen RPDO – Kanal mit den Objekte 1600<sub>h</sub> bis 1603<sub>h</sub> (Schreibzugriff) erfolgen.**

Es sind die folgenden PDOs definiert:

PDO Nummer	Mapping Objekt Index	Subindex	Mapping Object Name
1 (01h)	6040 <sub>h</sub>	—	Controlword
2..20 (02h..14h)	—	—	reserviert
21 (15h)	3100 <sub>h</sub>	01 <sub>h</sub> ..08 <sub>h</sub>	ASCII-Objekt
22 (16h)	6040 <sub>h</sub>	—	controlword
	2060 <sub>h</sub>	00 <sub>h</sub>	Strom- oder Drehzahlsollwert
23..31 (17h..1Fh)	—	—	reserviert
32 (20h)	2060 <sub>h</sub>	00 <sub>h</sub>	Strom- oder Drehzahlsollwert
33 (21h)	2022 <sub>h</sub>	04 <sub>h</sub>	Inkrementeller Positionssollwert
	2022 <sub>h</sub>	04 <sub>h</sub>	Inkrementeller Positionssollwert
34 (22h)	2022 <sub>h</sub>	01 <sub>h</sub>	Position
	2022 <sub>h</sub>	02 <sub>h</sub>	Geschwindigkeit
	2022 <sub>h</sub>	03 <sub>h</sub>	Fahrauftragsart
35 (23h)	2022 <sub>h</sub>	05 <sub>h</sub>	Fahrsatznummer (Achtung: Im Gerätezustand "Operation Enable" wird sofort gestartet)
36 (24h)	—	—	reserviert
37..40 (25h..28h)	xxxx <sub>h</sub>	xx <sub>h</sub>	Frei definierbare RPDOs
41 (29h)	2022 <sub>h</sub>	04 <sub>h</sub>	Master Position für CAN-Slaves
42..64 (30h..40h)	—	—	reserviert

#### 4.3.1.1 Beschreibung der vordefinierten Empfangs-PDOs

##### 4.3.1.1.1 PDO controlword (1) (DS402)

Das PDO controlword (Default - PDO) besteht aus dem controlword (UNSIGNED16). Mit diesem PDO kann am Einschalten ist dieses PDO auf RPDO 1 gemappt. Die Tabelle beschreibt das Mapping des PDO controlword:

Subindex	Wert	Beschreibung
00 <sub>h</sub>	1 <sub>h</sub>	Anzahl der Einträge
01 <sub>h</sub>	60400010 <sub>h</sub>	Kontrollwort

4.3.1.1.2 PDO ASCII-Kanal empfangen (21)

Mit Hilfe des ASCII-Kanals (Default-PDO) können sämtliche Parameter und Kommandos zum SERVOSTAR übertragen werden. Es können bis zu 8 ASCII-Zeichen in einem PDO gesendet werden. Kommandos oder Parameter, die mehr als 8 Zeichen benötigen, müssen segmentiert werden. Alle Kommandos und Parameter werden mit dem ASCII-Code "CR LF" (0xD<sub>h</sub>, 0xA<sub>h</sub>) abgeschlossen. Die nicht benötigten Bytes im PDO werden mit dem ASCII-Code "NUL" (0x0<sub>h</sub>) aufgefüllt, da sonst jedes überflüssige Zeichen als neues Kommando erkannt würde. Nach dem Einschalten ist dieses PDO auf RPDO2 gemappt.

Die Tabelle beschreibt das Mapping des PDO ASCII-Kanal empfangen:

Subindex	Wert	Beschreibung
00 <sub>h</sub>	1 <sub>h</sub>	Anzahl der Einträge
01..08 <sub>h</sub>	31000208 <sub>h</sub>	1. bis 8. ASCII-Zeichen

Dieses Objekt unterstützt **nur** den Übertragungstyp 255 (asynchron).

4.3.1.1.3 PDO Strom- / Drehzahlsollwert (22)

Das PDO Strom- bzw. Drehzahlsollwert setzt sich aus dem controlword (UNSIGNED16) und dem Sollwert (Signed16) zusammen. Dieses PDO darf nur in den Modi "Drehzahl digital" oder "Strom digital" benutzt werden. Anhand des eingestellten Mode (digitaler Strom- bzw. Drehzahlsollwert) wird erkannt, ob es sich um einen Drehzahl- oder Stromsollwert handelt. Das PDO wird sofort ausgeführt. Mehrfaches Senden des PDO mit verschiedenen Sollwerten erfordert kein zwischenzeitliches Anhalten des Antriebs.

**Stromnormierung:** 3280 = Spitzenstrom des Reglers  
1640 = Nennstrom

z.B. Nennstrom = 3A, Sollwert = 1,0A ⇒ 547 Inkremente

**Drehzahlnormierung:**  
z.B. Drehzahl = 3000min<sup>-1</sup> ⇒ Sollwert = 419430

Die Tabelle beschreibt das Mapping des PDO Sollwert:

Subindex	Wert	Beschreibung
00 <sub>h</sub>	1 <sub>h</sub>	Anzahl der Einträge
01 <sub>h</sub>	60400010 <sub>h</sub>	Controlword
02 <sub>h</sub>	20600020 <sub>h</sub>	Strom- / Drehzahlsollwert

4.3.1.1.4 PDO Sollwert 2 (32)

Das PDO Sollwert\_2 ist ein zeit- und datenoptimiertes PDO. Es beinhaltet ausschließlich einen 32 Bit Sollwert. Dieses PDO darf nur in den Modi "Drehzahl digital" oder "Strom digital" benutzt werden. Anhand des eingestellten Mode (digitaler Strom- bzw. Drehzahlsollwert) wird erkannt, ob es sich um einen Drehzahl- oder Stromsollwert handelt. Das PDO wird sofort ausgeführt. Mehrfaches Senden des PDO mit verschiedenen Sollwerten erfordert kein zwischenzeitliches Anhalten des Antriebs.

Die Tabelle beschreibt das Mapping des PDO Sollwert2:

Subindex	Wert	Beschreibung
00 <sub>h</sub>	1 <sub>h</sub>	Anzahl der Einträge
01 <sub>h</sub>	20600020 <sub>h</sub>	Strom- / Drehzahlsollwert

#### 4.3.1.1.5 PDO Trajektorie (33)

Das PDO Trajektorie ist ein zeit- und datenoptimiertes PDO. Dieses PDO darf nur im Mode "Trajektorie" benutzt werden. Das PDO Trajektorie muss in immer gleichen Zeitabständen (einstellbar mit dem Kommando "PTBASE") übertragen werden, da es sonst zu Einbrüchen im Drehzahlverlauf kommt. Dieses PDO besteht aus zwei Komponenten, den inkrementellen Istwerten für zwei Achsen. Die Zuordnung der Daten für die Achsen, die beide auf den gleichen COB-ID für dieses RPDO eingestellt werden müssen (Subindex 01<sub>h</sub> der entsprechenden Kommunikationsparameter), erfolgt über das Objekt 2721<sub>h</sub>.

**Beispiel zur Berechnung der absoluten Position:**

$$\text{Position} = \frac{\text{inkrementeller Positionswert}}{2^{20}}$$

Die max. Differenz zwischen zwei inkrementellen Positionen wird durch die eingestellte Enddrehzahl (ASCII - Kommando VLIM) vorgegeben (s. Beispiel).

**Beispiel zur maximalen inkrementellen Positionsdifferenz:**

$$\begin{aligned} \text{max. erreichbare Enddrehzahl} / 1000 \frac{\text{Umd.}}{\text{min}} &= 0,016667 \frac{\text{Umd.}}{\text{ms}} \\ |\text{ink. Pos.}(t_2) - \text{ink. Pos.}(t_1)| &\leq 2^{20} \times 0,016667 = 17475 \end{aligned}$$

In Abhängigkeit der eingestellten Verstärkerparameter kann es zu einem mehr oder weniger großen Schleppfehler kommen. Tritt die Fehlermeldung "Schleppfehler" auf und die Achse wird mit der Notbremsrampe angehalten, so kann das mehrere Fehlerursachen haben:

- Die inkrementelle Positionsdifferenz ist zu groß gewählt worden (s.o.).
- Das Schleppfehlerfenster ist zu klein eingestellt worden (Objekt 2020<sub>h</sub> Subindex 03<sub>h</sub>)
- Verstärkerparameter sind nicht optimal eingestellt

Die Tabelle beschreibt das Mapping des PDO Trajektorie:

Subindex	Wert	Beschreibung
00 <sub>h</sub>	2 <sub>h</sub>	Anzahl der Einträge
01 <sub>h</sub>	20220420 <sub>h</sub>	Inkrementelle Position
02 <sub>h</sub>	20220420 <sub>h</sub>	Inkrementelle Position

Dieses Objekt unterstützt **nicht** den Übertragungstyp 255 (asynchron).

#### 4.3.1.1.6 PDO Fahrsatz (34)

Das PDO Fahrsatz setzt sich aus Position (SIGNED32, gewichtet), Geschwindigkeit (UNSIGNED16) und der Fahrauftragsart (UNSIGNED16) zusammen.

Das PDO startet einen Fahrsatz aus dem flüchtigen Fahrsatzspeicher (Fahrsatznummer = 0) und ist ausschließlich im Mode "Lage" verwendbar.

Die Tabelle beschreibt das Mapping des PDO Fahrsatz:

Subindex	Wert	Beschreibung
00 <sub>h</sub>	3 <sub>h</sub>	Anzahl der Einträge
01 <sub>h</sub>	20220120 <sub>h</sub>	Position
02 <sub>h</sub>	20220210 <sub>h</sub>	Geschwindigkeit
03 <sub>h</sub>	20220310 <sub>h</sub>	Fahrauftragsart (abs. / rel.)

Dieses Objekt unterstützt **nur** den Übertragungstyp 255 (asynchron).

#### 4.3.1.1.7 PDO Starte Fahrsatz (35)

Das PDO Starte Fahrsatz besteht aus der Fahrauftragsnummer (UNSIGNED16).  
Das PDO startet einen Fahrsatz aus dem flüchtigen (Fahrsatznummer = 0, 192..255) oder permanenten (Fahrsatznummer = 1..180) Fahrsatzspeicher.  
Dieses PDO kann ausschließlich im Mode "Lage" verwendet werden.

Die Tabelle beschreibt das Mapping des PDO starte Fahrsatz:

Subindex	Wert	Beschreibung
00 <sub>h</sub>	1 <sub>h</sub>	Anzahl der Einträge
01 <sub>h</sub>	20220510 <sub>h</sub>	Fahrauftragsnummer

Dieses Objekt unterstützt **nur** den Übertragungstyp 255 (asynchron).

#### 4.3.1.1.8 PDO Frei definierbar (37 bis 40)

Sind diese PDOs angewählt, so können die Objekte frei hinzugefügt werden. Hierzu werden die Objekte 1600<sub>h</sub> bis 1603<sub>h</sub> (Mapping Parameter) verwendet. Es können bis zu 8 einzelne Objekte in ein PDO "gemappt" werden.

Ein Beispiel zum PDO Mapping und Kommunikationskonfiguration finden Sie im Anhang (⇒ Kap. 6.1).

#### 4.3.1.1.9 PDO Masterposition für CAN-Slaves (41)

Über das PDO 41 erhält der CAN-Slave Sollpositionen eines Masters, der er wie beim elektronischen Getriebe folgt. Diese Sollpositionen müssen in einem festen Zeitraster gesendet werden.

Die folgenden Einstellungen sind Voraussetzung für diesen Betrieb:

- Über das Objekt 60C2<sub>h</sub> muss der Takt eingestellt sein (-> PTBASE).
- Synchronisationsquelle ist der CAN-Bus -> SYNCSRC = 3
- FPGA = 3 (nur SERVOSTAR 600)
- CAMMCTRL = 2
- CANopen-Mode 0xFA (Trajektorienmodus)

Die Tabelle beschreibt das Mapping des Receive PDOs „Masterposition über CAN“

Subindex	Wert	Beschreibung
00 <sub>h</sub>	1 <sub>h</sub>	Anzahl der Einträge
01 <sub>h</sub>	20220420 <sub>h</sub>	Interner Positionssollwert

### 4.3.1.2 Objektbeschreibung

#### 4.3.1.2.1 Objekt 1400-1403h: 1st -4th Receive - PDO Kommunikationparameter (DS301)

Subindex	Beschreibung
00 <sub>h</sub>	Anzahl der Einträge
01 <sub>h</sub>	PDO COB-ID*
02 <sub>h</sub>	transmission type**
03 <sub>h</sub>	inhibit time
04 <sub>h</sub>	CMS priority group

\* Nach der Selektion eines PDOs über 2600-2603<sub>h</sub> kann hier der COB-Identifizier benannt werden, auf den der Verstärker reagieren soll (falls ein abweichender Wert vom Defaultwert erforderlich ist)

\*\* Zur Anwahl SYNC, event-gesteuert etc.

#### 4.3.1.2.2 Objekt 1600-1603h: 1st -4th Receive - PDO Mappingparameter (DS301)

Subindex	Beschreibung
00 <sub>h</sub>	Anzahl der konfigurierten Objekte
01 <sub>h</sub> ..08 <sub>h</sub>	Beschreibung der konfigurierten Objekte

#### 4.3.1.2.3 Objekt 2600-2603h: 1st -4th Receive PDO select

Mit Hilfe dieses Objekte wird ein vordefiniertes Empfangs - PDO ausgewählt. Mit den Objekte 1400-1403<sub>h</sub> "1<sup>st</sup> -4<sup>th</sup> Receive PDO parameter" und 1600-1603<sub>h</sub> "1<sup>st</sup> -4<sup>th</sup> Receive PDO mapping" können dann die Eigenschaften dieses PDOs festgelegt werden.

Dieses Objekt ermöglicht ein variables Mapping von vordefinierten PDOs. Die möglichen anwählbaren PDOs sind in der Tabelle in Kapitel 4.3.1 aufgeführt.

<b>Index</b>	2600 <sub>h</sub> ... 2603 <sub>h</sub>
<b>Name</b>	1 <sup>st</sup> -4 <sup>th</sup> Receive PDO select
<b>Objektcode</b>	VAR
<b>Datentyp</b>	UNSIGNED8

<b>Zugriff</b>	rw
<b>PDO Mapping</b>	nicht möglich
<b>Wertebereich</b>	0..255
<b>Defaultwert</b>	0

#### 4.3.1.2.4 Objekt 2721h: Konfiguration Receive PDO 33

Dieses Objekt beeinflusst die Verarbeitung der empfangenen acht Datenbytes des PDOs Trajektorie (Nr. 33). Mit dem Wert „LOW“ (LOW = 0) werden die Bytes 0 – 3 (Subindex 01<sub>h</sub> des PDO-Mappings) und mit dem Wert „HIGH“ (HIGH = 1) die Bytes 4 – 7 (Subindex 02<sub>h</sub> des PDO-Mappings) des PDOs als inkrementeller Sollwert für die nächste Bewegung interpretiert.

<b>Index</b>	2721 <sub>h</sub>
<b>Name</b>	PDO 33 data select
<b>Objektcode</b>	VAR
<b>Datentyp</b>	UNSIGNED8

<b>Zugriff</b>	rw
<b>PDO Mapping</b>	nicht möglich
<b>Einheit</b>	—
<b>Wertebereich</b>	0, 4
<b>EEPROM</b>	nein
<b>Defaultwert</b>	0

4.3.2 **Sende-PDOs (TPDO)**

Mit den Objekte 2A00<sub>h</sub> bis 2A03<sub>h</sub> können TPDOs (t=transmit) ausgewählt werden. Die Konfiguration der TPDOs erfolgt mit den Objekte 1800<sub>h</sub> bis 1803<sub>h</sub> (Kommunikationsparameter) und 1A00<sub>h</sub> bis 1A03<sub>h</sub> (Mapping - Parameter). Diese Objekte arbeiten kanalbezogen d. h.:

- TPDO Kanal 1: 2A00<sub>h</sub>, 1800<sub>h</sub>, 1A00<sub>h</sub>
- TPDO Kanal 2: 2A01<sub>h</sub>, 1801<sub>h</sub>, 1A01<sub>h</sub>
- TPDO Kanal 3: 2A02<sub>h</sub>, 1802<sub>h</sub>, 1A02<sub>h</sub>
- TPDO Kanal 4: 2A03<sub>h</sub>, 1803<sub>h</sub>, 1A03<sub>h</sub>

Die Objekte 1A00<sub>h</sub> bis 1A03<sub>h</sub> können nur gelesen werden und so die Objektkonfiguration innerhalb des PDOs ermittelt werden.



**Ausnahme:**

**Wurden einer oder mehrere der TPDOs 37 bis 40 ausgewählt, muss die Objektkonfiguration für den jeweiligen TPDO – Kanal mit den Objekte 1A00<sub>h</sub> bis 1A03<sub>h</sub> (Schreibzugriff) erfolgen.**

Es sind die folgenden PDOs definiert:

PDO Nummer	Mapping Objekt Index	Subindex	Mapping Object Name
1 (01 <sub>h</sub> )	6041 <sub>h</sub>	—	statusword
2..20 (02 <sub>h</sub> ..14 <sub>h</sub> )	—	—	reserviert
21 (15 <sub>h</sub> )	3102 <sub>h</sub>	01 <sub>h</sub> ..08 <sub>h</sub>	ASCII-Objekt
22 (16 <sub>h</sub> )	6041 <sub>h</sub>	—	statusword
	2070 <sub>h</sub>	01 <sub>h</sub>	Istlage (20 Bit / Umdrehung)
	2070 <sub>h</sub>	02 <sub>h</sub>	Drehzahl (1 Bit = 1875/262144 [Umdr./min])
23 (17 <sub>h</sub> )	6041 <sub>h</sub>	—	statusword
	1002 <sub>h</sub>	—	herstellerspezifisches Statusregister
24..31 (18 <sub>h</sub> ..1F <sub>h</sub> )	—	—	reserviert
32 (20 <sub>h</sub> )	2070 <sub>h</sub>	01 <sub>h</sub>	Istlage (20 Bit / Umdrehung)
	2070 <sub>h</sub>	02 <sub>h</sub>	Drehzahl (1 Bit = 1875/262144 [Umdr./min])
33 (21 <sub>h</sub> )	2070 <sub>h</sub>	03 <sub>h</sub>	Inkrementeller Positionswert
	2070 <sub>h</sub>	11 <sub>h</sub>	2. erw. Statusregister
34..35 (22 <sub>h</sub> ..23 <sub>h</sub> )	—	—	—
36 (24 <sub>h</sub> )	—	—	reserviert
37..40 (25 <sub>h</sub> ..28 <sub>h</sub> )	xxx <sub>h</sub>	xx <sub>h</sub>	Frei definierbare TPDOs
41 (29 <sub>h</sub> )	2070 <sub>h</sub>	19 <sub>h</sub>	Interne Master Sollposition
42..64 (30 <sub>h</sub> ..40 <sub>h</sub> )	—	—	reserviert

4.3.2.1 **Beschreibung der vordefinierten Send-PDOs**

4.3.2.1.1 **PDO statusword (1) (DS402)**

Das PDO statusword (Default - PDO) besteht aus dem statusword (UNSIGNED16).

Mit diesem PDO kann ausschließlich der Zustand der Zustandsmaschine (⇒ Kap. 4.4.1) ermittelt werden. Dieses PDO ist modeunabhängig.

Nach dem Einschalten ist dieses PDO auf TPDO 1 gemappt.

Die Tabelle beschreibt das Mapping des PDO statusword

Subindex	Wert	Beschreibung
00 <sub>h</sub>	1 <sub>h</sub>	Anzahl der Einträge
01 <sub>h</sub>	60410010 <sub>h</sub>	statusword

#### 4.3.2.1.2 PDO ASCII-Kanal senden (21)

Sobald ASCII-Zeichen in den ASCII Sendepuffer übertragen werden, werden diese mit Hilfe dieses PDOs (Default-PDO) zum Master (Steuerung) übertragen. Das erfolgt immer dann, wenn mit Hilfe des PDO ASCII-Kanal empfangen ( $\Rightarrow$  Kap. 4.3.1.1.2) Kommandos bzw. Parameter übertragen worden sind. Nach dem Einschalten ist dieses PDO auf TPDO2 gemappt.

Die Tabelle beschreibt das Mapping des PDO ASCII-Kanal senden

Subindex	Wert	Beschreibung
00 <sub>h</sub>	8 <sub>h</sub>	Anzahl der Einträge
01 <sub>h</sub> ..08 <sub>h</sub>	31020208 <sub>h</sub>	1. bis 8. ASCII-Zeichen

Dieses Objekt unterstützt **nur** den Übertragungstyp 255 (asynchron).

#### 4.3.2.1.3 PDO Ist-Lage (22)

Das PDO Ist-Lage besteht aus statusword (UNSIGNED16), Istlage (UNSIGNED24) und Umdrehung pro Minute (UNSIGNED24). Mit diesem PDO kann die Position im Mode "Drehzahl digital" oder "Strom digital" ermittelt werden.

Die Tabelle beschreibt das Mapping des PDO Ist-Lage:

Subindex	Wert	Beschreibung
00 <sub>h</sub>	3 <sub>h</sub>	Anzahl der Einträge
01 <sub>h</sub>	60410010 <sub>h</sub>	statusword
02 <sub>h</sub>	20700118 <sub>h</sub>	Istlage (20 Bit / Umdrehung)
03 <sub>h</sub>	20700218 <sub>h</sub>	Drehzahl*

\* Auflösung: 1 Bit = 1875 / 262144 min<sup>-1</sup>

#### 4.3.2.1.4 PDO Erweiterter Status (23)

Das PDO Erweiterter Status besteht aus dem statusword (UNSIGNED16) und dem herstellereigenen Statusregister (UNSIGNED32). Dieses PDO kann zusätzlich auf ein Ereignis im Bereich des Statusregisters getriggert werden. Dieses PDO ist modeunabhängig.

Die Tabelle beschreibt das Mapping des PDO Erweiterter Status:

Subindex	Wert	Beschreibung
00 <sub>h</sub>	2 <sub>h</sub>	Anzahl der Einträge
01 <sub>h</sub>	60410010 <sub>h</sub>	statusword
02 <sub>h</sub>	10020020 <sub>h</sub>	Herstellerspezifisches Statusregister

#### 4.3.2.1.5 PDO Ist-Lage 2 (32)

Das PDO Ist-Lage 2 ist ein zeit- und datenoptimiertes PDO (vgl. PDO 21). Es beinhaltet die Istlage (UNSIGNED24) und die Umdrehung pro Minute (UNSIGNED 24). Mit diesem PDO kann die Position im Mode "Drehzahl digital" oder "Strom digital" ermittelt werden.

Dieses PDO kann **ausschließlich** mit dem **SYNC - Objekt** angefordert werden.

Die Tabelle beschreibt das Mapping des PDO Ist-Lage 2:

Subindex	Wert	Beschreibung
00 <sub>h</sub>	2 <sub>h</sub>	Anzahl der Einträge
01 <sub>h</sub>	20700118 <sub>h</sub>	Istlage (20 Bit / Umdrehung)
02 <sub>h</sub>	20700218 <sub>h</sub>	Drehzahl *

\* Auflösung: 1 Bit = 1875 / 262144 min<sup>-1</sup>

Dieses Objekt unterstützt **nur** die Übertragungstypen 1 bis 240 (zyklisch synchron).

#### 4.3.2.1.6 PDO Inkrementeller Positionswert (33)

Das PDO Inkrementeller Positionswert ist ein datenoptimiertes Objekt, das **ausschließlich** mit einem **SYNC - Objekt** angefordert werden kann.

**Berechnung der absoluten Position:**

$$\text{Position} = \frac{\text{inkrementeller Positionswert}}{2^{20}}$$

Die Tabelle beschreibt das Mapping des PDO Inkrementeller Positionswert:

Subindex	Wert	Beschreibung
00h	2h	Anzahl der Einträge
01h	20700320h	Inkrementeller Positionswert
02h	20701120h	erweitertes Statusregister für PDO 33

Dieses Objekt unterstützt **nur** die Übertragungstypen 1 bis 240 (zyklisch synchron).

#### 4.3.2.1.7 PDO Positionsschwellen (34)

Das PDO Positionsschwellen wird verwendet, um ein Überschreiten der schnellen Positionsregister P1..P16 zu erkennen. Sie werden über die Objekte 2051 - 2053 konfiguriert

Die Tabelle beschreibt das Mapping des PDO Positionsschwellen:

Subindex	Wert	Beschreibung
00h	1h	Anzahl der Einträge
01h	20860010h	Positionsschwellenanzeige

#### 4.3.2.1.8 PDO Frei definierbar (37 bis 40)

Sind dies PDOs angewählt können die Objekte frei hinzugefügt werden. Hierzu werden die Objekte 1A00h bis 1A03h (Mapping Parameter) verwendet. Es können bis 8 einzelne Objekte in ein PDO "gemappt" werden.



**Auch wenn Objekte "event getriggert" sind und nicht über den CAN Bus gemeldet werden, werden diese dennoch überwacht. Das führt dazu, dass bei 16 gemappten Objekten die CPU Last sehr stark steigen kann. Um das zu verhindern, sollten wirklich nur die Objekte gemappt werden, die auch wirklich benötigt werden, oder die "inhibit Zeit" muss entsprechend hoch gesetzt werden (z.B. 400 ms). Die Folge einer zu hohen CPU – Last kann eine lange Reaktionszeit bei einem Objekt Datendienst (Antwortzeit im Mittel > 40 ms) sein.**

#### 4.3.2.1.9 PDO Interne Master-Sollposition (41)

Über das Transmit-PDO 41 kann der CAN-Master seine interne Trajektorien Sollposition (z.B. aus dem Fahrsatzbetrieb) auf den Bus geben. Diese Werte werden in einem durch PTBASE bestimmten Zeitraster ausgegeben.

Die folgenden Einstellungen sind Voraussetzung für diesen Betrieb:

- Über das Objekt 60C2 muss der Takt eingestellt werden (->PTBASE)
- SYNCSRC = 3
- FPGA = 3 (nur SERVOSTAR 600)
- CAMMCTRL = 1

Die Tabelle beschreibt das Mapping des PDO „Interne Master-Sollposition“

Subindex	Wert	Beschreibung
00h	1h	Anzahl der Einträge
01h	20701920h	Interner Positionssollwert

### 4.3.2.2 Objektbeschreibung

#### 4.3.2.2.1 Objekt 1800-1803h: 1st -4th Transmit - PDO Kommunikationsparam. (DS301)

Subindex	Beschreibung
00 <sub>h</sub>	Anzahl der Einträge
01 <sub>h</sub>	PDO COB-ID*
02 <sub>h</sub>	transmission type
03 <sub>h</sub>	inhibit time**
04 <sub>h</sub>	CMS priority group

\* Nach der Selektion eines PDOs über 2A00<sub>h</sub> bis 2A03<sub>h</sub> kann hier der COB-Identifizier benannt werden, auf den der Verstärker reagieren soll (falls ein abweichender Wert vom Defaultwert erforderlich ist)

\*\* Hiermit wird eine Ruhezeit eingestellt, die nach dem Senden eines PDOs eingehalten werden muss, bevor es wieder gesendet werden kann (nur wichtig bei event-gesteuerten PDOs).

#### 4.3.2.2.2 Objekt 1A00-1A03h: 1st -4th Transmit - PDO Mappingparameter (DS301)

Subindex	Beschreibung
00 <sub>h</sub>	Anzahl der konfigurierten Objekte
01 <sub>h</sub> ..08 <sub>h</sub>	Beschreibung der konfigurierten Objekte

#### 4.3.2.2.3 Objekt 2A00-2A03h: 1st -4th Transmit PDO select

Mit Hilfe dieses Objekte wird ein vordefiniertes Sende - PDO ausgewählt.

Mit den Objekte 1800<sub>h</sub>-1803<sub>h</sub> "1<sup>st</sup> -4<sup>th</sup> Transmit PDO parameter" und 1A00<sub>h</sub>-1A03<sub>h</sub> "1<sup>st</sup> -4<sup>th</sup> Transmit PDO mapping" können dann die Eigenschaften dieses PDOs festgelegt werden.

Dieses Objekt ermöglicht ein variables Mapping von vordefinierten PDOs. Die möglichen anwählbaren PDOs sind in der Tabelle in Kap. 4.3.2 aufgeführt.

<b>Index</b>	2A00 <sub>h</sub> ...2A03 <sub>h</sub>
<b>Name</b>	1 <sup>st</sup> -4 <sup>th</sup> Transmit PDO select
<b>Objektcode</b>	VAR
<b>Datentyp</b>	UNSIGNED8

<b>Zugriff</b>	rw
<b>PDO Mapping</b>	nicht möglich
<b>Wertebereich</b>	0..255
<b>Defaultwert</b>	0

#### 4.3.2.2.4 Objekt 2014-2017h: Maske 1-4 für Sende-PDOs

Um die Buslast bei ereignisgesteuerten PDOs zu senken, kann die Überwachung für einzelne Bits im PDO mit Hilfe der Masken abgeschaltet werden. Beispielsweise kann so auch erreicht werden, dass Positionswerte immer nur pro Umdrehung gemeldet werden.

Diese Objekte maskieren den PDO - Kanal 1 für TPDOs mit den Nummern 37 bis 40. Sollten nur 2 Byte in einem PDO definiert sein, überlagert die Maske auch nur zwei Byte obwohl 4 Byte Maskeninformation übertragen worden ist.

Ein aktiviertes Bit in der Maske bedeutet, dass die Überwachung für das korrespondierende Bit im PDO aktiv ist, d. h. überwacht wird.

<b>Index</b>	0x2014h...0x2017h.
<b>Name</b>	tx_mask 1-4
<b>Objektcode</b>	ARRAY
<b>Anzahl Elemente</b>	2
<b>Datentyp</b>	UNSIGNED32

Beschreibung der Subindizes:

<b>Subindex</b>	01h
<b>Kurzbeschreibung</b>	tx_mask1-4_low
<b>Mode</b>	unabhängig
<b>Zugriff</b>	rw
<b>PDO Mapping</b>	nicht möglich
<b>Einheit</b>	—
<b>Wertebereich</b>	—
<b>EEPROM</b>	nein
<b>Defaultwert</b>	FFFFFFFFh

<b>Subindex</b>	02h
<b>Kurzbeschreibung</b>	tx_mask1-4_high
<b>Mode</b>	unabhängig
<b>Zugriff</b>	rw
<b>PDO Mapping</b>	nicht möglich
<b>Einheit</b>	—
<b>Wertebereich</b>	—
<b>EEPROM</b>	nein
<b>Defaultwert</b>	FFFFFFFFh

## 4.4 Gerätesteuerung (dc)

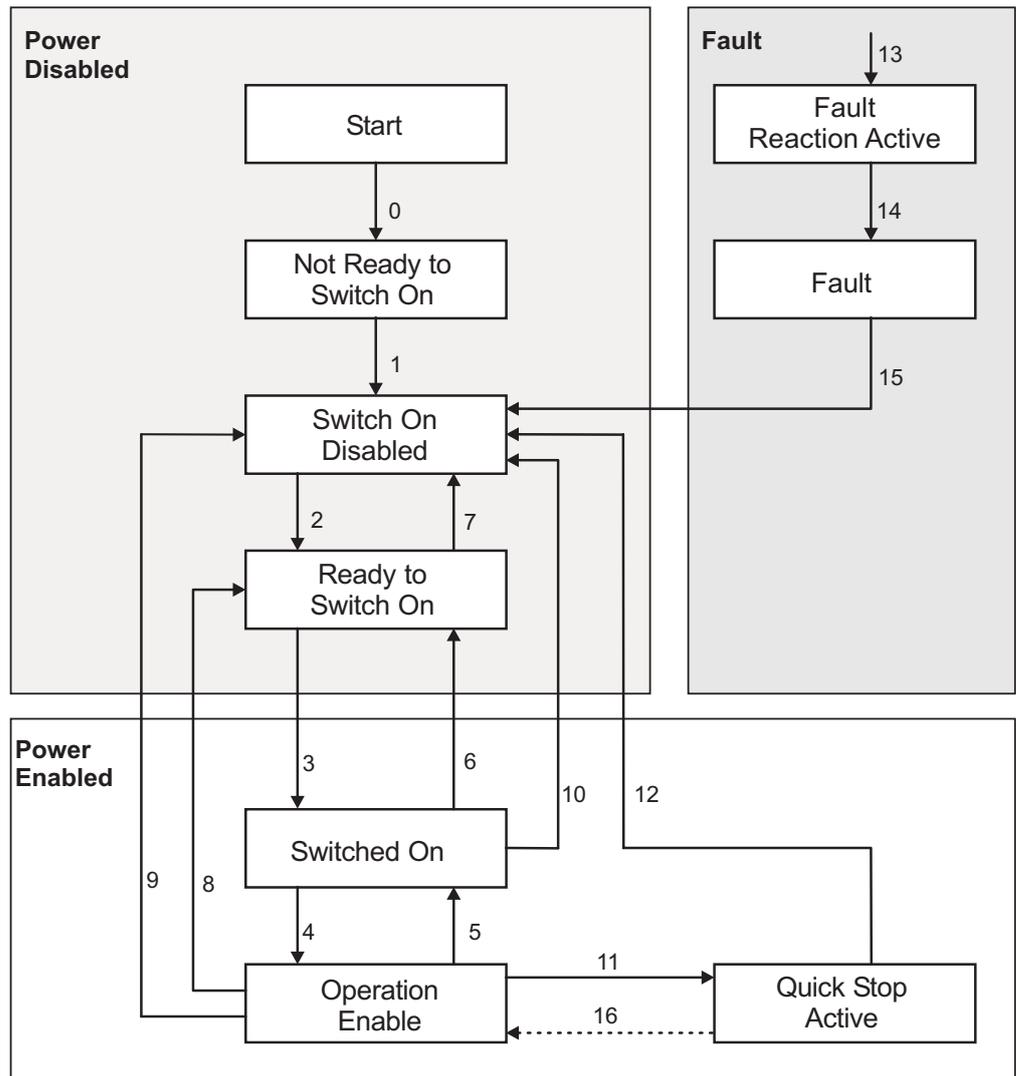
Mit Hilfe der SERVOSTAR Gerätesteuerung können sämtliche Fahrfunktionen in den entsprechenden Modi ausgeführt werden. Die Steuerung des SERVOSTAR 400/600 ist über eine modeabhängige Zustandsmaschine realisiert.

Zur Steuerung der Zustandsmaschine dient das controlword (⇒ Kap.4.4.2.1).

Die Modeeinstellung erfolgt über das Objekt "Modes of Operation" (⇒ Kap.6.3). Die Zustände der Zustandsmaschine können mit dem statusword ermittelt werden (⇒ .Kap.4.4.2.2).

4.4.1 Zustandsmaschine (DS402)

4.4.1.1 Zustände der Zustandsmaschine



Zustand	Beschreibung
Not Ready to Switch On	SERVOSTAR ist nicht einschaltbereit, es wird keine Betriebsbereitschaft (BTB) vom Reglerprogramm gemeldet.
Switch On Disable	SERVOSTAR ist einschaltbereit, Parameter können übertragen werden, Zwischenkreisspannung kann eingeschaltet werden, Fahrfunktionen können noch nicht ausgeführt werden
Ready to Switch On	Zwischenkreisspannung kann angeschaltet werden. Parameter können übertragen werden, Fahrfunktionen können noch nicht ausgeführt werden
Switched On	Zwischenkreisspannung muss angeschaltet sein, Parameter können übertragen werden, Fahrfunktionen können noch nicht ausgeführt werden, Endstufe ist eingeschaltet (Enabled)
Operation Enable	kein Fehler steht an, Endstufe ist eingeschaltet (Enabled), Fahrfunktionen sind "enabled"
Quick Stop Active	Antrieb ist mit der Notbremsrampe gestoppt worden, Endstufe ist eingeschaltet (Enabled), Fahrfunktionen sind "enabled", Verhalten abhängig von Objekt 605A <sub>h</sub> (⇒ Kap.6.3)
Fault Reaction Active	wird z.Zt. nicht unterstützt
Fault	wird z.Zt. nicht unterstützt

#### 4.4.1.2 Übergänge der Zustandsmaschine

Die Zustandsübergänge werden durch interne Ereignisse (z.B. Ausschalten der Zwischenkreisspannung) und durch die Flags im controlword (Bits 0,1,2,3,7) beeinflusst.

Übergang	Ereignis	Aktion
0	Reset	Initialisierung
1	Initialisierung erfolgreich abgeschlossen. SERVOSTAR betriebsbereit	keine
2	Bit 1 (disable Voltage) und Bit 2 (Quick Stop) im controlword gesetzt ('Shutdown' command). Zwischenkreisspannung kann anliegen.	keine
3	Bit 0 wird zusätzlich gesetzt ('Switch On' command)	Endstufe wird eingeschaltet (enable), sofern Hardware-enable anliegt (UND-Verknüpfung). Antrieb hat Drehmoment.
4	Bit 3 wird zusätzlich gesetzt ('Enable Operation' command)	Fahrfunktion in Abhängigkeit des eingestellten Mode wird freigegeben
5	Bit 3 wird gelöscht ('Disable Operation' command)	Fahrfunktion wird gesperrt. Antrieb wird mit der relevanten Rampe (modeabhängig) gestoppt. Aktuelle Position wird gehalten
6	Bit 0 wird gelöscht ('Shutdown' command)	Endstufe wird abgeschaltet (disable). Motor wird drehmomentfrei
7	Bit 1/2 werden gelöscht ('Quickstop' / 'Disable Voltage' command)	keine
8	Bit 0 wird gelöscht ('Shutdown' command)	Endstufe wird abgeschaltet (disable) Motor wird drehmomentfrei
9	Bit 1 wird gelöscht ('Disable Voltage' command)	Endstufe wird abgeschaltet (disable) Motor wird drehmomentfrei
10	Bit 1/2 werden gelöscht ('Quickstop' / 'Disable Voltage' command)	Fahrfunktion in Abhängigkeit des eingestellten Mode wird freigegeben.
11	Bit 2 wird gelöscht ('Quickstop' command)	Antrieb wird mit der Notbremsrampe angehalten. Die Endstufe bleibt "enabled". Sollwerte werden gelöscht (Fahrsatznummer, digitaler Sollwert, Geschwindigkeit für Tippbetrieb oder Referenzfahrt). Vor dem erneuten Ausführen einer Fahrfunktion muss Bit 2 wieder gesetzt werden
12	Bit 1 wird gelöscht ('Disable Voltage' command)	Endstufe wird abgeschaltet (disable) Motor wird drehmomentfrei
13	Wird zur Zeit nicht unterstützt	keine
14	Wird zur Zeit nicht unterstützt	keine
15	Wird zur Zeit nicht unterstützt	keine
16	Bit 2 wird gesetzt	Fahrfunktion ist wieder freigegeben.



**Wenn der Servoverstärker über das controlword / statusword bedient wird, dürfen keine Steuerkommandos über einen anderen Kommunikationskanal (RS232, CANopen, ASCII-Kanal, Optionskarte) gesendet werden.**

## 4.4.2 Objektbeschreibung

### 4.4.2.1 Objekt 6040h: Controlword (DS402)

Aus der logischen Verknüpfung der Bits im controlword und externen Signalen (Freigabe der Endstufe) ergeben sich die Steuerkommandos. Die Definition der Bits wird im Folgenden beschrieben:

<b>Index</b>	0x6040h
<b>Name</b>	Controlword
<b>Objektcode</b>	VAR
<b>Datentyp</b>	UNSIGNED16

<b>Mode</b>	all
<b>Zugriff</b>	rw
<b>PDO Mapping</b>	möglich
<b>Einheit</b>	—
<b>Wertebereich</b>	0..65535
<b>EEPROM</b>	nein
<b>Defaultwert</b>	0

#### Bitbelegung im controlword

Bit	Name	Bit	Name
0	Switch on	8	Halt (Zwischenstopp)
1	Disable Voltage	9	reserved
2	Quick Stop	10	reserved
3	Enable Operation	11	Schleppfehler und Ansprechüberwachung quittieren
4	Operation mode specific	12	Rücksetzen der Position
5	Operation mode specific	13	Manufacturer specific
6	Operation mode specific	14	Manufacturer specific
7	Reset Fault (Nur wirksam bei Fehlern)	15	Manufacturer specific

#### Kommandos im controlword

Command	Bit 7 Fault Reset	Bit 3 Enable Operation	Bit 2 Quick Stop	Bit 1 Disable Voltage	Bit 0 Switch on	Übergänge
Shutdown	X	X	1	1	0	2, 6, 8
Switch on	X	X	1	1	1	3
Disable Voltage	X	X	X	0	X	7, 9, 10, 12
Quick Stop	X	X	0	1	X	7, 10, 11
Disable Operation	X	0	1	1	1	5
Enable Operation	X	1	1	1	1	4, 16
Fault Reset	1	X	X	X	X	15

Mit X gekennzeichnete Bits sind irrelevant.

**Modeabhängige Bits im controlword**

Die folgende Tabelle beschreibt die modeabhängigen Bits im controlword. Es werden z. Zt. nur herstellerspezifische Modi unterstützt. Die einzelnen Modi werden mit dem Objekt 6060<sub>h</sub> "Modes of operation" eingestellt.

Operation mode	Nr.	Bit 4	Bit 5	Bit 6
Lage	FF <sub>h</sub>	reserviert	reserviert	reserviert
Drehzahl digital	FE <sub>h</sub>	reserviert	reserviert	reserviert
Strom digital	FD <sub>h</sub>	reserviert	reserviert	reserviert
Drehzahl analog	FC <sub>h</sub>	reserviert	reserviert	reserviert
Strom analog	FB <sub>h</sub>	reserviert	reserviert	reserviert
Trajektorie	FA <sub>h</sub>	reserviert	reserviert	reserviert
Referenzierung	F9 <sub>h</sub>	Referenzfahrt starten	reserviert	reserviert
Tippbetrieb	F8 <sub>h</sub>	reserviert	reserviert	reserviert
Profile Position Mode (pp)	01 <sub>h</sub>	new_set_point	change_set_immediately	absolut / relativ
Profile Velocity Mode (pv)	03 <sub>h</sub>	reserviert	reserviert	reserviert
Homing Mode (hm)	06 <sub>h</sub>	homing_operation_start	reserviert	reserviert

**Beschreibung der restlichen Bits im controlword**

Im Folgenden werden die restlichen Bits des controlword beschrieben.

**Bit 8 Zwischenstopp** Ist Bit 8 gesetzt, wird in sämtlichen Modi der Antrieb gestoppt (Zwischenstopp). Die Sollwerte (Geschwindigkeit für Referenzfahrt oder Tippbetrieb, Fahrauftragsnummer, Sollwerte für Digitalmode) der einzelnen Modi bleiben erhalten

**Bit 9,10** Diese Bits sind durch das Antriebsprofil (DS402) reserviert.

**Bit 11 Fehler quittieren** Durch Setzen von Bit 11 wird die Ansprechüberwachung und/oder der Schleppfehler quittiert.

**Bit 12** Rücksetzen der Position unter Berücksichtigung des Referenzoffsets. (siehe auch Referenzfahrtart Nr. 6 in Objekt 2024<sub>h</sub>, Subindex 01<sub>h</sub>)

**Bit 13, 14, 15** Diese Bits sind herstellerspezifisch und z.Zt. reserviert.

**4.4.2.2 Objekt 6041h: Statusword (DS402)**

Der momentane Zustand der Zustandsmaschine kann mit Hilfe des statusword (⇒ Kap.4.3.2.1.1) abgefragt werden.

<b>Index</b>	0x6041 <sub>h</sub>
<b>Name</b>	Statusword
<b>Objektcode</b>	VAR
<b>Datentyp</b>	UNSIGNED16

<b>Mode</b>	all
<b>Zugriff</b>	rw
<b>PDO Mapping</b>	möglich
<b>Einheit</b>	—
<b>Wertebereich</b>	0..65535
<b>EEPROM</b>	ja
<b>Defaultwert</b>	0

**Bitbelegung im statusword**

Bit	Name	Bit	Name
0	Ready to switch on	8	Manufacturer specific (reserviert)
1	Switched on	9	Remote (nicht unterstützt)
2	Operation enable	10	Target reached
3	Fault (i.V.)	11	Internal limit active
4	Disable voltage	12	Operation mode specific (reserviert)
5	Quick stop	13	Operation mode specific (reserviert)
6	Switch on disabled	14	Manufacturer specific (reserviert)
7	Warning	15	Manufacturer specific (reserviert)

**Zustände der Statusmaschine**

State	Bit 6 switch on disable	Bit 5 quick stop	Bit 3 fault	Bit 2 operation enable	Bit 1 switched on	Bit 0 ready to switch on
Not ready to switch on	0	X	0	0	0	0
Switch on disabled	1	X	0	0	0	0
Ready to switch on	0	1	0	0	0	1
Switched on	0	1	0	0	1	1
Operation enabled	0	1	0	1	1	1
Fault	Wird z. Zt. nicht unterstützt					
Fault reaction active	Wird z. Zt. nicht unter- stützt	X	X	X	X	15
Quick stop active	0	0	0	1	1	1

Die mit X gekennzeichneten Bits sind irrelevant

**Beschreibung der restlichen Bits im statusword**

**Bit 4: voltage\_disable** Die Zwischenkreisspannung liegt an, wenn das Bit gelöscht ist.

**Bit 7: warning** Für das Setzen von Bit 7 kann es mehrere Gründe geben, die zu dieser Warnung geführt haben. Der Grund für diese Warnung kann anhand des Objekt 1002<sub>h</sub> "herstellerspezifisches Statusregister" ermittelt werden. (⇒ Kap. 4.2.1.3)

**Bit 8: Toggle – Bit Mode** Tippbetrieb, Referenzierung ,Lage, Positionierung (pp), Referenzierung (hm) Das Toggle - Bit wird immer dann verändert (setzen bzw. rücksetzen), wenn ein Fahrsatz erfolgreich ausgeführt wurde (Inkrement genau in Zielposition!). Das Bit wird nicht "getoggelt", wenn ein Fahrsatz abgebrochen wird (z.B.: mit dem STOP – Kommando oder beim Schleppfehler). Die Auswertung des Toggle - Bits kann in Kombination mit dem Bit 10 "target reached" (Objekt 6041<sub>h</sub>) und "Fahrsatz aktiv" Bit 16 (Objekt 1002<sub>h</sub>) erfolgen. Die Auswertung des Bits ist dann sinnvoll, wenn aufgrund der Fahrsatzdaten keine Änderung des Bits 10 oder Bits 16 sichtbar werden (sehr kurze oder gleiche Fahrsätze).

**Bit 9: remote** Wird nicht unterstützt

**Bit 10: target\_reached** Wird gesetzt, wenn der Antrieb die Zielposition erreicht hat.

**Bit 11: internal\_limit\_active** Ist im Parameter DRVCFG das Bit 20 gesetzt, wird dieses Bit bei den folgenden anstehenden Warnungen gesetzt:

- Ansprechüberwachung (n04)
- SW-Endschalter 1 (n06)
- SW-Endschalter 2 (n07)
- nicht gesetzter Referenzpunkt (n09)
- P-STOP (n10)
- N-STOP (n11)
- Slot-Warnung (n13)

**Bit 12: profile position mode** setpoint acknowledge  
**digital speed modes** velocity 0 detection (1 = speed = 0)

### 4.4.2.3 Objekt 6060h: modes\_of\_operation (DS402)

Mit diesem Objekt wird der Mode eingestellt und mit dem Objekt 6061<sub>h</sub> kann der eingestellte Mode gelesen werden. Es können zwei Arten von Operationsmodi unterschieden werden:

#### herstellerspezifische Operationsmodi

Diese Operationsmodi sind auf die Gerätefunktionalitäten optimierte Betriebszustände.

#### Operationsmodi nach CANopen Antriebsprofil DSP402

Diese Operationsmodi nach CANopen Antriebsprofil DSP402 definiert.

Nach einem Modewechsel muss der entsprechende Sollwert neu gesetzt werden (z.B.: Mode Referenzierung \_ Sollwert Referenzfahrtgeschwindigkeit). Wurde der Mode Lage oder Tippbetrieb gespeichert, so ist der Mode Referenzierung nach einem RESET des Servoverstärkers eingestellt.

**Ein Operationmode ist erst gültig, wenn er mit Objekt 6061<sub>h</sub> gelesen werden konnte.**

**Niemals den Modus bei drehendem Motor umschalten!**

**Das Umschalten der Modi ist bei "enabletem" Verstärker grundsätzlich nur bei Drehzahl 0 erlaubt. Setzen Sie vor dem Umschalten den Sollwert auf 0.**



<b>Index</b>	0x6060 <sub>h</sub>
<b>Name</b>	mode_of_operation
<b>Objektcode</b>	VAR
<b>Datentyp</b>	INTEGER8

<b>Mode</b>	all
<b>Zugriff</b>	wo
<b>PDO Mapping</b>	möglich
<b>Einheit</b>	—
<b>Wertebereich</b>	—
<b>EEPROM</b>	nein
<b>Defaultwert</b>	—

Mode	mode dez.	mode hex.	Kommentar
	-10..-128	F8 <sub>h</sub> ..80 <sub>h</sub>	reserviert
elektr. Getriebe	-9	F7 <sub>h</sub>	—
Tippbetrieb	-8	F8 <sub>h</sub>	—
Referenzierung	-7	F9 <sub>h</sub>	—
Trajektorie	-6	FA <sub>h</sub>	—
Strom analog	-5	FB <sub>h</sub>	—
Drehzahl analog	-4	FC <sub>h</sub>	—
Strom digital	-3	FD <sub>h</sub>	—
Drehzahl digital	-2	FE <sub>h</sub>	—
Lage	-1	FF <sub>h</sub>	Notwendiger Mode für Fahraufträge
—	0	0	reserviert nach DSP402
Positionierung (pp)	1	1 <sub>h</sub>	nach DSP402
Drehzahl (vl)	2	2 <sub>h</sub>	wird nicht unterstützt
Drehzahl (pv)	3	3 <sub>h</sub>	nach DSP 402
Drehmoment (tq)	4	4 <sub>h</sub>	nach DSP 402 (wird z. Zt. nicht unterstützt)
—	5	5 <sub>h</sub>	reserviert
Referenzierung (hm)	6	6 <sub>h</sub>	nach DSP 402
Interpolation	7	7 <sub>h</sub>	nach DSP 402 (wird nicht unterstützt)
—	8..127	8 <sub>h</sub> ..7F <sub>h</sub>	reserviert

#### 4.4.2.4 Objekt 6061h: mode\_of\_operation\_display (DS402)

Mit diesem Objekt kann der über das Objekt 6060<sub>h</sub> eingestellte Mode gelesen werden. Ein Operati-  
onmode ist erst gültig, wenn er mit Objekt 6061<sub>h</sub> gelesen werden konnte (s. auch Objekt 6060<sub>h</sub>).

Index	0x6061 <sub>h</sub>
Name	mode_of_operation_display
Objektcode	VAR
Datentyp	INTEGER8

Mode	all
Zugriff	wo
PDO Mapping	möglich
Einheit	—
Wertebereich	—
EEPROM	nein
Defaultwert	—

## 4.5 Factor Groups (fg) (DS402)

### 4.5.1 Allgemeine Informationen

#### 4.5.1.1 Faktoren

Es gibt eine Umrechnungsmöglichkeit für die physikalischen Dimensionen und Größen auf die antriebsinternen Einheiten (Inkrementen). Dafür können einige Faktoren implementiert werden. Dieses Kapitel beschreibt den Einfluss dieser Faktoren auf das System, die Berechnungsvorschriften für diese Faktoren und die Daten, die dazu erforderlich sind.

**Normierte Parameter sind mit \* gekennzeichnet.**

#### 4.5.1.2 Beziehung zwischen physikalischen und antriebsinternen Einheiten

Die in den factor groups festgelegten Faktoren definieren die Beziehung zwischen antriebsinternen Einheiten (Inkrementen) und physikalischen Einheiten.

Die Faktoren sind das Ergebnis einer Berechnung von zwei Parametern: des dimension index und des notation index. Der dimension index beschreibt dabei die physikalische Dimension, der notation index die physikalische Einheit und den Dezimalpunkt für Werte. Diese Faktoren werden zur Normierung der physikalischen Werte verwendet.

Der notation index kann auf zwei Arten verwendet werden:

- Für eine Einheit mit dezimaler Skalierung und notation index < 64 definiert der notation index den Dezimalpunkt der Einheit.
- Für eine Einheit mit nicht-dezimaler Skalierung und notation index > 64 definiert der notation index den Subindex der physikalischen Dimension der Einheit.

## 4.5.2 Objektbeschreibung

## 4.5.2.1 Objekt 608Bh: velocity\_notation\_index (DS402)

<b>Index</b>	0x608B <sub>h</sub>
<b>Name</b>	velocity_notation_index
<b>Objektcode</b>	VAR
<b>Datentyp</b>	INTEGER8

<b>Mode</b>	all
<b>Zugriff</b>	rw
<b>PDO Mapping</b>	nicht möglich
<b>Einheit</b>	—
<b>Wertebereich</b>	-128..127
<b>EEPROM</b>	nein
<b>Defaultwert</b>	0

Dieses Objekt legt die Notation für das Objekt 6081<sub>h</sub> "profile\_velocity" fest. In Kombination mit dem Objekt 608C<sub>h</sub> "velocity\_dimension\_index" können die folgenden Basiseinheiten dargestellt werden:

physical dimension	units	velocity_dimension_index	velocity_notation_index
herstellerspezifisch	inc/s	0	0
Umdrehungen	U/min	11	73

## 4.5.2.2 Objekt 608Ch: velocity\_dimension\_index (DS402)

<b>Index</b>	0x608C <sub>h</sub>
<b>Name</b>	velocity_dimension_index
<b>Objektcode</b>	VAR
<b>Datentyp</b>	UNSIGNED8

<b>Mode</b>	all
<b>Zugriff</b>	rw
<b>PDO Mapping</b>	nicht möglich
<b>Einheit</b>	—
<b>Wertebereich</b>	0..255
<b>EEPROM</b>	nein
<b>Defaultwert</b>	0

Dieses Objekt legt die Dimension für das Objekt 6081<sub>h</sub> "profile\_velocity" fest. In Kombination mit dem Objekt 608B<sub>h</sub> "velocity\_notation\_index" können die folgenden Basiseinheiten dargestellt werden:

physical dimension	Einheit	velocity_dimension_index	velocity_notation_index
herstellerspezifisch	inkr/s	0	0
Umdrehungen	U/min	11	73

### 4.5.2.3 Objekt 6093h: position\_factor (DS402)

Der Positionsfaktor (position\_factor) konvertiert die gewünschte Position in das interne Datenformat des SERVOSTAR 400/600 (Inkremente).

Die Auflösung des Lagereglers kann mit 20 Bit/Umd oder 16 Bit/Umd betrieben werden (s. Objekt 35D1<sub>h</sub> und ASCII – Kommando PRBASE). Mit Hilfe des Numerators und der Vorschubkonstante kann jede beliebige Skalierung eingestellt werden.

$$\text{position\_factor} = \frac{\text{position\_encoder\_resolution} \times \text{gear\_ratio}}{\text{feed\_constant}}$$

- **position\_encoder\_resolution**  
Auflösung des Lagereglers  $2^{20}$  oder  $2^{16}$
- **gear\_ratio**  
Übersetzungsverhältnis des eingesetzten Getriebes
- **feed\_constant**  
Vorschubkonstante des Abtriebes

<b>Index</b>	0x6093 <sub>h</sub>
<b>Name</b>	position_factor
<b>Objektcode</b>	ARRAY
<b>Anzahl Elemente</b>	2
<b>Datentyp</b>	UNSIGNED32

Beschreibung der Subindizes:

<b>Subindex</b>	01 <sub>h</sub>
<b>Kurzbeschreibung</b>	numerator
<b>Mode</b>	pp
<b>Zugriff</b>	rw
<b>PDO Mapping</b>	nicht möglich
<b>Einheit</b>	Inkrement [inc]
<b>Wertebereich</b>	$0..(2^{32}-1)$
<b>EEPROM</b>	nein
<b>Defaultwert</b>	$2^{20}$

<b>Subindex</b>	02 <sub>h</sub>
<b>Kurzbeschreibung</b>	feed_constant
<b>Mode</b>	pp
<b>Zugriff</b>	rw
<b>PDO Mapping</b>	nicht möglich
<b>Einheit</b>	—
<b>Wertebereich</b>	$0..(2^{32}-1)$
<b>EEPROM</b>	nein
<b>Defaultwert</b>	10000

**Beispiel:** Eine Umdrehung soll 10000 Inkrementen entsprechen. Übersetzungsverhältnis ist 1.

$$\text{position\_factor} = \frac{2^{20} \text{Inkr.}}{10000 \text{Inkr.}}$$

- ⇒ **Numerator:**  $2^{20}$       **Vorschubkonstante:** 10000  
 ⇒ Sollwertvorgabe in [Inkr / Umd.] für das Objekt 607A<sub>h</sub> (target\_position)

Der Numerator entspricht im Gerät dem ASCII - Parameter PGEARO, die feed\_constant dem Parameter PGEARI.

#### 4.5.2.4 Objekt 6094h: velocity\_encoder\_factor (DS402)

Der velocity\_encoder\_factor konvertiert die gewünschte Drehzahl (Umd/min) oder Geschwindigkeit (inc/s) in das interne Datenformat des SERVOSTAR 400/600 (Inkremente).

Für die Berechnung des velocity\_encoder\_factor gilt:

$$\text{velocity\_encoder\_factor} = \frac{\text{velocity\_encoder\_resolution} \times \text{gear\_ratio} \times \text{position\_unit} \times \text{Fvelocity}(\text{notation\_index})}{\text{feed\_constant} \times \text{velocity\_unit} \times \text{sec} \times \text{Fposition}(\text{notation\_index})}$$

- **velocity\_encoder\_resolution** Auflösung des Drehzahlreglers  $2^{20}$
- **gear ratio** Übersetzungsverhältnis des eingesetzten Getriebes
- **position unit** in Meter
- **Fposition(notation\_index)** in dimension\_index = 1, notation\_index = 0 [m]
- **feed\_constant** Vorschubkonstante des Abtriebes
- **velocity\_unit** in [m/s]
- **Fvelocity (notation\_index)** in dimension\_index = 13, notation\_index = 0 [m / s]

(s. auch Objekt 606Bh "velocity\_notation\_index" und Objekt 606Ch "velocity\_dimension\_index")

<b>Index</b>	0x6094h
<b>Name</b>	velocity_encoder_factor
<b>Objektcode</b>	ARRAY
<b>Anzahl Elemente</b>	2
<b>Datentyp</b>	UNSIGNED32

Beschreibung der Subindizes:

<b>Subindex</b>	01h
<b>Kurzbeschreibung</b>	numerator
<b>Mode</b>	pv
<b>Zugriff</b>	rw
<b>PDO Mapping</b>	nicht möglich
<b>Einheit</b>	Inkremente [inc]
<b>Wertebereich</b>	$0..(2^{32}-1)$
<b>EEPROM</b>	nein
<b>Defaultwert</b>	0

<b>Subindex</b>	02h
<b>Kurzbeschreibung</b>	divisor
<b>Mode</b>	pv
<b>Zugriff</b>	rw
<b>PDO Mapping</b>	nicht möglich
<b>Einheit</b>	Sekunde [s]
<b>Wertebereich</b>	$0..(2^{32}-1)$
<b>EEPROM</b>	nein
<b>Defaultwert</b>	0

**Beispiel:** Die Drehzahl - Sollwertvorgabe soll in Umdrehungen pro Minute erfolgen. Das Übersetzungsverhältnis und die Vorschubkonstante ist 1.

$$\text{velocity\_encoder\_factor} = \frac{2^{20}}{1} \cdot \frac{1}{1} \cdot \frac{\text{Inkr.}}{\text{s}}$$

Erweiterung auf Umdrehung / min:

$$\text{velocity\_encoder\_factor} = \frac{2^{20}}{1} \cdot \frac{1}{1} \cdot \frac{[\text{m}]}{[\text{m/s}]} \cdot \frac{\text{Inkr.}}{\text{s}} \times \frac{1}{60[\text{s/min}]} = \frac{2^{20}}{60} \cdot \frac{\text{Inkr.}}{\text{s}}$$

- ⇒ **Numerator:**  $2^{20}$       **Divisor:** 60
- ⇒ Sollwertvorgabe in [Umd / min] für das Objekt 60FF<sub>h</sub> "target\_velocity / velocity units"
- ⇒ Sollwertvorgabe in [inc. / s] für das Objekt 6081<sub>h</sub> "Profile\_velocity / speed units"

**Anmerkung:** Da der Drehzahlregler intern, unabhängig von der Auflösung des Gebersystems mit einer Auflösung von  $2^{20}$  Bit/Umdrehung arbeitet, ergibt sich folgende Berechnung für den Operationsmode pv (für Umdrehung pro Minute):

$$\text{Inkmente} = \frac{262144}{1875} \times \text{Drehzahlsollwert}[\text{min}^{-1}]$$

Diese inkrementelle Sollwertvorgabe sollte bei zyklischen Anwendungen (z.B. Lageregelung, 4ms Zyklus) verwendet werden (Vorteile: keine Rundungsfehler, geringere CPU - Last).

Die o. g. Umrechnung ist gültig, wenn **Divisor oder Numerator zu 0** gesetzt werden.

Der velocity\_encoder\_factor beeinflusst auch das Objekt 6081<sub>h</sub> "profile\_velocity". Um diesen Faktor auch für den Positioniermode (pp) verwenden zu können, müssen die internen Getriebefaktoren PGEAR1 und PGARO gleich sein (PGEAR1 = PGARO; Objekt 2020<sub>h</sub> Subindex: 08<sub>h</sub>, 09<sub>h</sub>). Werden **Divisor oder Numerator zu 0** gesetzt, wird die interne Skalierung "Inkmente pro Abtastzyklus (250 µs)" verwendet.

#### 4.5.2.5 Objekt 6097h: acceleration\_factor (DS402)

Der Beschleunigungsfaktor (acceleration\_factor) konvertiert die Beschleunigung [Einheit /s<sup>2</sup>] in das interne Format des SERVOSTAR 400/600.

Zur Zeit sind Numerator und Divisor nur lesbar. Die Werte sind zu 1 gesetzt. Ist der Beschleunigungsfaktor 1, werden die Rampenvorgaben (Objekt 6083h "profile\_acceleration" und Objekt 6084h "profile\_deceleration") als Beschleunigungszeit [ms] bezogen auf die Zielgeschwindigkeit (Objekt 6081h "profile\_velocity") vorgegeben.

<b>Index</b>	0x6097h
<b>Name</b>	acceleration_factor
<b>Objektcode</b>	ARRAY
<b>Anzahl Elemente</b>	2
<b>Datentyp</b>	UNSIGNED32

Beschreibung der Subindizes:

<b>Subindex</b>	01h
<b>Kurzbeschreibung</b>	numerator
<b>Mode</b>	pp, pv
<b>Zugriff</b>	ro
<b>PDO Mapping</b>	nicht möglich
<b>Einheit</b>	Inkrementen [inc]
<b>Wertebereich</b>	0..(2 <sup>32</sup> -1)
<b>EEPROM</b>	nein
<b>Defaultwert</b>	0

<b>Subindex</b>	02h
<b>Kurzbeschreibung</b>	divisor
<b>Mode</b>	pp, pv
<b>Zugriff</b>	ro
<b>PDO Mapping</b>	nicht möglich
<b>Einheit</b>	—
<b>Wertebereich</b>	0..(2 <sup>32</sup> -1)
<b>EEPROM</b>	nein
<b>Defaultwert</b>	0

## 4.6 Herstellerspezifischer Strom- und Drehzahlmode

### 4.6.1 Objekt 2060h: Digitaler Strom- oder Drehzahlsollwert

Index	2060h
Kurzbeschreibung	Digitaler Strom- oder Drehzahlsollwert
Einheit	A bzw. min <sup>-1</sup>
Zugriff	rw
PDO mapping	möglich (vorgemappt auf wählbares RPDO 22)
Datentyp	INTEGER32
Wertebereich	-(2 <sup>31</sup> -1)..(2 <sup>31</sup> -1)
Defaultwert	0

Dieses Objekt dient zur Übergabe digitaler Sollwerte, die abhängig vom eingestellten Digitalmode (Mode FD<sub>h</sub> = Strom digital, Mode FE<sub>h</sub> = Drehzahl digital, einstellbar über Objekt 6060<sub>h</sub>) ausgewertet werden. Die Normierungen sind dabei wie folgt festgelegt:

$$\text{Strom: } I[\text{A}] = \frac{\text{digitaler Stromsollwert}}{1640} \times I_{\text{max}}$$

$$\text{Drehzahl: } n[\text{min}^{-1}] = \frac{1875}{262144} \times \text{digitaler Drehzahlsollwert}$$

Ein neuer Sollwert wird immer erst nach einem erneuten Enable Operation gültig. (über Objekt 6040<sub>h</sub>, Controlword)



**Der SERVOSTAR 400/600 - Lageregler ist bei aktiver Drehzahl- oder Stromregelung abgeschaltet.**

### 4.6.2 Objekt 2061h: Strombegrenzung

Dieses Objekt dient zur schnellen Strombegrenzung im Drehzahlmode (0x3, 0xFE, 0xFC). Ein Wert von 3280 entspricht dabei dem maximalen Gerätestrom, der über DIPEAK abfragbar ist.

Index	2061h
Kurzbeschreibung	Strombegrenzung
Einheit	A (bezogen auf DIPEAK)
Zugriff	wo
PDO mapping	möglich
Datentyp	UNSIGNED16
Wertebereich	0...3280
Defaultwert	0

Dieses Objekt wirkt auf den ASCII-Parameter DPRILIMIT. Damit es wirksam werden kann, muss der Konfigurationsparameter DILIM auf 1 stehen.

## 4.7 Einrichtdaten für den herstellersiz. Tipp- und Referenziermode

### 4.7.1 Objekt 2024h: Einrichtbetrieb für den Mode Large (SERVOSTAR)

Index	2024h
Kurzbeschreibung	Parameter für Referenzfahrt und Tippen
Objektcode	RECORD
Anzahl Elemente	7

Über diesen Index werden Parameter eingegeben, die für die Betriebsarten Referenzieren und Tippen wichtig sind.

Beschreibung der Subindizes:

Subindex	01h
Kurzbeschreibung	Referenzfahrtart
Einheit	—
Zugriff	rw
PDO mapping	nicht möglich
Datentyp	UNSIGNED8
Wertebereich	0..6
Defaultwert	0

Einstellen der Art der Referenzfahrt. Folgenden Einstellungen sind möglich:

Wert	Bedeutung
0	Der Referenzpunkt wird an der Ist - Position gesetzt. Die dann gemeldete Istposition ist gleich dem eingestellten Referenzoffset.
1	Referenzfahrt auf Referenzschalter mit anschliessender Suche des Resolvnullpunkts.
2	Referenzfahrt auf Endschalter mit anschliessender Suche des Resolvnullpunkts.
3	Referenzfahrt auf Referenzschalter ohne anschliessende Suche des Resolvnullpunkts.
4	Referenzfahrt auf Endschalter ohne anschliessende Suche des Resolvnullpunkts.
5	Referenzfahrt innerhalb einer Motorumdrehung bis zum Resolvnullpunkt. Die Fahrtrichtung wird dabei durch den Subindex 02h vorgegeben. Dabei bedeuten: 0: Fahrtrichtung negativ 1: Fahrtrichtung positiv 2: Motordrehung in Richtung der kürzesten Entfernung zum Resolvnullpunkt innerhalb einer Umdrehung.
6	Der Referenzpunkt wird an der aktuellen Lageregler-Sollposition auf den Wert des Referenzoffsets gesetzt. Die neue Istposition behält den selben Abstand zur Sollposition wie zuvor.



**Für die Referenzfahrten 1 und 3 muss ein digitaler Eingang als Nullposition-Eingang (Home - Position) konfiguriert sein.**

**Für die Referenzfahrten 2 und 4 muss ein digitaler Eingang als Hardware - Endschalter konfiguriert werden.**

**Bei den Referenzfahrtarten 1 – 5 wird die Einstellung des Nullimpulsoffsets für die ROD - Ausgabe berücksichtigt (ASCII - Kommando ENCZERO) d.h. der Nullpunkt wird so gelegt, dass sowohl die Ausgabe des Nullimpulses als auch die Anzeige der 0-Position an der Stelle des Nullimpulsoffsets erfolgen.**

Subindex	02h
Kurzbeschreibung	Referenzfahrtrichtung
Einheit	—
Zugriff	rw
PDO mapping	nicht möglich
Datentyp	UNSIGNED8
Wertebereich	0..2
Defaultwert	0

Über diesen Subindex wird die Fahrtrichtung für die Referenzfahrtarten 1 bis 5 vorgegeben. Dabei haben die Werte die folgende Bedeutung:

0: Fahrtrichtung negativ

1: Fahrtrichtung positiv

2: Motordrehung in Richtung der kürzesten Entfernung zum Resolvnullpunkt innerhalb einer Umdrehung. (Nur relevant bei Referenzfahrtart 5).

<b>Subindex</b>	03h
<b>Kurzbeschreibung</b>	Geschwindigkeit der Referenzfahrt
<b>Einheit</b>	µm/s
<b>Zugriff</b>	rw
<b>PDO mapping</b>	nicht möglich
<b>Datentyp</b>	INTEGER32
<b>Wertebereich</b>	$-(2^{31}-1)..(2^{31}-1)$
<b>Defaultwert</b>	0

Vorgabe der Referenzfahrtgeschwindigkeit.

<b>Subindex</b>	04h
<b>Kurzbeschreibung</b>	Beschleunigungsrampe Tippen / Referenzfahrt
<b>Einheit</b>	ms
<b>Zugriff</b>	rw
<b>PDO mapping</b>	nicht möglich
<b>Datentyp</b>	UNSIGNED16
<b>Wertebereich</b>	1..32767
<b>Defaultwert</b>	10

Einstellen der Beschleunigungsrampe für Referenzfahrt und Tipbetrieb. Sie wird als Trapezrampe ausgeführt. Die eingestellte Zeit bezieht sich auf die eingestellten Geschwindigkeiten für Referenzfahrt und Tippen.

<b>Subindex</b>	05h
<b>Kurzbeschreibung</b>	Bremsrampe für Tippen / Referenzfahrt
<b>Einheit</b>	ms
<b>Zugriff</b>	rw
<b>PDO mapping</b>	nicht möglich
<b>Datentyp</b>	UNSIGNED16
<b>Wertebereich</b>	1..32767
<b>Defaultwert</b>	10

Einstellen der Bremsrampe für Referenzfahrt und Tipbetrieb. Sie wird als Trapezrampe ausgeführt. Die eingestellte Zeit bezieht sich auf die eingestellten Geschwindigkeiten für Referenzfahrt und Tippen. Bei den Referenzfahrten auf Hardwareendschalter wird als Bremsrampe die Notrampe (ASCII – Parameter DECSTOP) verwendet.

<b>Subindex</b>	06h
<b>Kurzbeschreibung</b>	Referenzoffset
<b>Einheit</b>	µm
<b>Zugriff</b>	rw
<b>PDO mapping</b>	nicht möglich
<b>Datentyp</b>	INTEGER32
<b>Wertebereich</b>	$-(2^{31}-1)..(2^{31}-1)$
<b>Defaultwert</b>	0

Einstellen des Referenzoffset, d.h. der angezeigte Positionswert nach erfolgter Referenzierung (Objekt 2070<sub>h</sub>, Subindex 06<sub>h</sub>) wird eingestellt.

<b>Subindex</b>	07h
<b>Kurzbeschreibung</b>	Geschwindigkeit Tipbetrieb
<b>Einheit</b>	µm/s
<b>Zugriff</b>	rw
<b>PDO mapping</b>	nicht möglich
<b>Datentyp</b>	INTEGER32
<b>Wertebereich</b>	$-(2^{31}-1)..(2^{31}-1)$
<b>Defaultwert</b>	0

Vorgabe der Tippgeschwindigkeit.

## 4.8 Positionierdaten für den Mode Lage (SERVOSTAR)

### 4.8.1 Objekt 2020h: Lageregler

Index	2020h
Kurzbeschreibung	Parameter für den Lageregler
Objektcode	RECORD
Anzahl Elemente	10

Über diesen Index werden alle allgemeinen Parameter für den Mode Lage vorgegeben.

Beschreibung der Subindizes:

Subindex	01h
Kurzbeschreibung	Achstyp
Einheit	—
Zugriff	rw
PDO mapping	nicht möglich
Datentyp	UNSIGNED8
Wertebereich	0, 1
Defaultwert	0

Beschreibt die Art der mechanischen Achse.

**Wert 0:** Linearachse. Es gibt einen definierten Referenzpunkt, ab dem Positionen gezählt werden. Dieser muss durch eine Referenzfahrt oder durch Setzen eines Referenzpunktes definiert werden. Falls konfiguriert, wird die Bewegung der Achse durch Software-Endschalter begrenzt.

**Wert 1:** Rundachse. Es muss nicht referenziert werden. Beim Starten von Fahrsätzen oder des Tippbetriebes wird die Position jeweils auf 0 gesetzt. Software-Endschalter begrenzen die Bewegung nicht.

Subindex	02h
Kurzbeschreibung	In-Positionsfenster
Einheit	µm
Zugriff	rw
PDO mapping	nicht möglich
Datentyp	INTEGER32
Wertebereich	$-(2^{31}-1)..(2^{31}-1)$
Defaultwert	4000h

Festlegung eines Zielfensters für Positionierungen. Wird die Zielfenstergrenze erreicht, wird im herstellereigenen Statusregister das Bit 19 gesetzt und bei entsprechender Ausgangskonfiguration der eingestellte Ausgang auf High gelegt.

Subindex	03h
Kurzbeschreibung	Maximaler Schleppfehler
Einheit	µm
Zugriff	rw
PDO mapping	nicht möglich
Datentyp	INTEGER32
Wertebereich	$-(2^{31}-1)..(2^{31}-1)$
Defaultwert	40000h

Festlegung eines Maximalwerts des Schleppfehlers. Überschreitet der auftretende Schleppfehler diesen Wert, wird der Antrieb gestoppt. Über das herstellereigene Statusregister wird im Bit 2 die Schleppfehlerüberschreitung angezeigt. Bei einem eingestellten Wert von 0 wird der Schleppfehler nicht überwacht.

<b>Subindex</b>	04 <sub>h</sub>
<b>Kurzbeschreibung</b>	Positionsregister 1
<b>Einheit</b>	µm
<b>Zugriff</b>	rw
<b>PDO mapping</b>	nicht möglich
<b>Datentyp</b>	INTEGER32
<b>Wertebereich</b>	$-(2^{31}-1)..(2^{31}-1)$
<b>Defaultwert</b>	0

Je nach Konfiguration wird bei Über- bzw. Unterschreitung der eingestellten Position ein Schwellwertbit (Bit 22 herstellerspezifisches Statusregister) gesetzt oder die Achse gestoppt. (Softwareend-schalter 1 unterschritten = herstellerspezifisches Statusregister Bit 5 = 1).

<b>Subindex</b>	05 <sub>h</sub>
<b>Kurzbeschreibung</b>	Positionsregister 2
<b>Einheit</b>	µm
<b>Zugriff</b>	rw
<b>PDO mapping</b>	nicht möglich
<b>Datentyp</b>	INTEGER32
<b>Wertebereich</b>	$-(2^{31}-1)..(2^{31}-1)$
<b>Defaultwert</b>	0

Je nach Konfiguration wird bei Über- bzw. Unterschreitung der eingestellten Position ein Schwellwertbit (Bit 23 herstellerspezifisches Statusregister) gesetzt oder die Achse gestoppt. (Softwareend-schalter 2 überschritten = herstellerspezifisches Statusregister Bit 6 = 1).

<b>Subindex</b>	06 <sub>h</sub>
<b>Kurzbeschreibung</b>	Positionsregister 3
<b>Einheit</b>	µm
<b>Zugriff</b>	rw
<b>PDO mapping</b>	nicht möglich
<b>Datentyp</b>	INTEGER32
<b>Wertebereich</b>	$-(2^{31}-1)..(2^{31}-1)$
<b>Defaultwert</b>	0

Je nach Konfiguration wird bei Über- bzw. Unterschreitung der eingestellten Position ein Schwellwertbit (Bit 24 herstellerspezifisches Statusregister) gesetzt.

<b>Subindex</b>	07 <sub>h</sub>
<b>Kurzbeschreibung</b>	Positionsregister 4
<b>Einheit</b>	µm
<b>Zugriff</b>	rw
<b>PDO mapping</b>	nicht möglich
<b>Datentyp</b>	INTEGER32
<b>Wertebereich</b>	$-(2^{31}-1)..(2^{31}-1)$
<b>Defaultwert</b>	0

Je nach Konfiguration wird bei Über- bzw. Unterschreitung der eingestellten Position ein Schwellwertbit (Bit 25 herstellerspezifisches Statusregister) gesetzt.

<b>Subindex</b>	08 <sub>h</sub>
<b>Kurzbeschreibung</b>	Auflösung / Nenner des Übersetzungsfaktors
<b>Einheit</b>	Umdrehungen
<b>Zugriff</b>	rw
<b>PDO mapping</b>	nicht möglich
<b>Datentyp</b>	UNSIGNED32
<b>Wertebereich</b>	$1..(2^{32}-1)$
<b>Defaultwert</b>	1

Über das Verhältnis der Werte der Subindizes 08<sub>h</sub> und 09<sub>h</sub> wird die mechanische Auflösung der Achse in µm / Umdrehung vorgegeben.

<b>Subindex</b>	09 <sub>h</sub>
<b>Kurzbeschreibung</b>	Auflösung / Zähler des Übersetzungsfaktors
<b>Einheit</b>	µm
<b>Zugriff</b>	rw
<b>PDO mapping</b>	nicht möglich
<b>Datentyp</b>	UNSIGNED32
<b>Wertebereich</b>	1..(2 <sup>32</sup> -1)
<b>Defaultwert</b>	1

Über das Verhältnis der Werte der Subindizes 08<sub>h</sub> und 09<sub>h</sub> wird die mechanische Auflösung der Achse in µm / Umdrehung vorgegeben.

<b>Subindex</b>	0A <sub>h</sub>
<b>Kurzbeschreibung</b>	Zählrichtung
<b>Einheit</b>	—
<b>Zugriff</b>	rw
<b>PDO mapping</b>	nicht möglich
<b>Datentyp</b>	UNSIGNED8
<b>Wertebereich</b>	0, 1
<b>Defaultwert</b>	1

Der Wert gibt die Zählrichtung für Strom-, Drehzahl- und Lageregelung vor. Bei einem Wert von 1 ist die positive Zählrichtung angewählt. Positive Sollwertvorgaben bewirken eine Drehung der Motorwelle im Uhrzeigersinn (bei Draufsicht).

## 4.8.2

## Objekt 2022h: Positionierdaten für den Mode Lage

<b>Index</b>	2022 <sub>h</sub>
<b>Kurzbeschreibung</b>	Fahrauftragsparameter
<b>Objektcode</b>	RECORD
<b>Anzahl Elemente</b>	16

Über diesen Index werden alle Parameter eingegeben, die einen Bezug zu im Regler gespeicherten oder direkt ausführbaren Fahraufträgen haben. (Siehe ASCII-Kommando "ORDER")

Beschreibung der Subindizes:

<b>Subindex</b>	01 <sub>h</sub>
<b>Kurzbeschreibung</b>	Position
<b>Mode</b>	Lage
<b>Zugriff</b>	rw
<b>PDO mapping</b>	möglich (vorgemappt auf wählbares RPDO 34)
<b>Datentyp</b>	INTEGER32
<b>Einheit</b>	Inkmente oder $\mu\text{m}$
<b>Wertebereich</b>	$-(2^{31}-1)..(2^{31}-1)$
<b>EEPROM</b>	nein
<b>Defaultwert</b>	0

Über diesen Index wird die Zielposition (Absolutfahrauftrag) oder die zu fahrende Strecke (Relativfahrauftrag) von Fahraufträgen vorgegeben. Dies wird durch Bit 0 der Fahrauftragsart festgelegt. Bit 13 der Fahrauftragsart legt fest, ob der angegebene Wert als Inkrement oder SI-Wert interpretiert werden soll.

<b>Subindex</b>	02 <sub>h</sub>
<b>Kurzbeschreibung</b>	Sollgeschwindigkeit gewichtet
<b>Mode</b>	Lage
<b>Zugriff</b>	rw
<b>PDO mapping</b>	möglich (vorgemappt auf wählbares RPDO 34)
<b>Datentyp</b>	INTEGER16
<b>Einheit</b>	Inkmente/s oder $\mu\text{m/s}$
<b>Wertebereich</b>	-32768..32767
<b>EEPROM</b>	nein
<b>Defaultwert</b>	0

Über diesen Index wird die Sollgeschwindigkeit für Fahraufträge vorgegeben. Er wird über Subindex 0D<sub>h</sub> gewichtet. Wird der Wert durch Wahl von Fahrauftragsartbit 13 = 1 als SI-Einheit definiert, ergibt sich die Inkrementalgeschwindigkeit  $v_i$  als

$$v_i = v_{\text{SI}} \times \frac{\text{PGEARO}}{\text{PGEARI} \times 4000}$$

wobei PGEARO (Objekt 2020<sub>h</sub>, Subindex 08<sub>h</sub>) die Anzahl der Inkmente enthält, die gefahren werden, wenn die zu fahrende Strecke PGEARI (Objekt 2020<sub>h</sub>, Subindex 09<sub>h</sub>) beträgt. Zu beachten ist, dass dabei eine Motorumdrehung einer Inkrementzahl von  $2^{20} = 1048567$  entspricht.

<b>Subindex</b>	03h
<b>Kurzbeschreibung</b>	Fahrauftragsart
<b>Mode</b>	pp
<b>Zugriff</b>	rw
<b>PDO mapping</b>	nicht möglich
<b>Datentyp</b>	UNSIGNED8
<b>Einheit</b>	—
<b>Wertebereich</b>	0..FFFFh
<b>EEPROM</b>	nein
<b>Defaultwert</b>	0

Über diesen Index werden Fahrparameter des Fahrauftrags eingestellt. Die Bedeutung der Bits wird in den folgenden Tabellen erklärt.

Bit	Wert	Bedeutung
0	0x0001h	Bit für die Art des relativen/absoluten Fahrsatzes (s. Tabelle 2)
1	0x0002h	Bit für die Art des relativen Fahrsatzes (s. Tabelle 2)
2	0x0004h	Bit für die Art des relativen Fahrsatzes (s. Tabelle 2)
3	0x0008h	=0 kein Folgefahrstanz vorhanden, nach dem Erreichen der Zielposition bleibt der Antrieb stehen =1 Folgefahrstanz vorhanden, nach dem Erreichen der Zielposition wird automatisch der Folgefahrstanz gestartet. Die Nummer des Folgefahrstanzes wird mit dem Kommando O_FN vorgegeben.
4	0x0010h	Bit für die Art des Folgefahrstanzes (s. Tabelle 3)
5	0x0020h	Bit für die Art des Folgefahrstanzes (s. Tabelle 3)
6	0x0040h	Bit für die Art des Folgefahrstanzes (s. Tabelle 3)
7	0x0080h	Bit für die Art des Folgefahrstanzes (s. Tabelle 3)
8	0x0100h	Bit für die Art des Folgefahrstanzes (s. Tabelle 3)
9	0x0200h	reserviert
10	0x0400h	reserviert
11	0x0800h	reserviert
12	0x1000h	=0 die Brems- und Anfahrbeschleunigung wird als Brems-/Anfahrzeit von 0 auf die Zielgeschwindigkeit (in msec) vorgegeben. =1 die Brems- und Anfahrbeschleunigung wird in mm/sek <sup>2</sup> vorgegeben. (s. auch Kommandos O_ACC1, O_ACC2, O_DEC1, O_DEC2).
13	0x2000h	=0 die Zielposition und die Zielgeschwindigkeit des Fahrsatzes werden als Inkremente interpretiert. Es findet keine Umrechnung statt. =1 die Zielposition und die Zielgeschwindigkeit werden vor dem Start des Fahrsatzes in Inkremente umgerechnet. Für die Umrechnung werden die Parameter PGEARI und PGEARO benutzt. (s. auch Kommandos O_S_O_V, PGEARI, PGEARO)
14	0x4000h	=0 die Geschwindigkeit des Fahrsatzes wird beim Fahrsatzstart als die Zielgeschwindigkeit übernommen. =1 die Zielgeschwindigkeit wird beim Fahrsatzstart analog vorgegeben (SW1). Beim Start des Fahrsatzes wird der analoge SW1-Wert eingelesen und als die Zielgeschwindigkeit übernommen (Skalierung: 10V=VSCALE1). Das Vorzeichen der SW1-Spannung wird ignoriert.
15	0x8000h	Bit 3 für die Art des relativen Fahrsatzes (s. separate Tabelle)

Art des relativen / absoluten Fahrsatzes

Bit 0	Bit 1	Bit 2	Bit 15	Bedeutung
0	x	x	x	Absoluter Fahrsatz, die Positionsangabe innerhalb des Fahrsatzes wird als die Zielposition gewertet.
1	0	0	x	Relativer Fahrsatz, die Positionsangabe innerhalb des Fahrsatzes wird als Verfahrstrecke gewertet. Die Zielposition wird abhängig vom Zustand der IN-POSITION Meldung berechnet: IN-POSITION=1: neue Zielposition=letzte Zielposition + Verfahrstrecke IN-POSITION=0: neue Zielposition=aktuelle Position + Verfahrstrecke
1	1	0	x	Relativer Fahrsatz, die Positionsangabe innerhalb des Fahrsatzes wird als Verfahrstrecke gewertet. neue Zielposition=letzte Zielposition+Verfahrstrecke
1	0	1	x	Relativer Fahrsatz, die Positionsangabe innerhalb des Fahrsatzes wird als Verfahrstrecke gewertet. neue Zielposition=aktuelle Position+Verfahrstrecke
1	1	1	0	Relativer Fahrsatz, die Positionsangabe innerhalb des Fahrsatzes wird als Verfahrstrecke gewertet. neue Zielposition=positive Latchposition+Verfahrstrecke (s. auch Kommando LATCH32)
1	1	1	1	Relativer Fahrsatz, die Positionsangabe innerhalb des Fahrsatzes wird als Verfahrstrecke gewertet. neue Zielposition=negative Latchposition+Verfahrstrecke (s. auch Kommando LATCH32N)

Art des Folgefahrsatzes

Bit 4 NOBRAKE	Bit 5 FOL IO	Bit 6 HI/LO	Bit 7 FTIME	Bit 8 VTARG	Bedeutung
0	0	0	0	0	Umschalten auf Folgefahrauftrag mit Abbremsen. Der Antrieb bremsst in die Zielposition des ersten Fahrsatzes. Anschließend wird der Folgefahrsatz gestartet.
1	0	0	0	0	Fliegendes Umschalten auf den Folgeauftrag in der Zielposition. Der Antrieb fährt mit der Zielgeschwindigkeit bis in die Zielposition des ersten Fahrsatzes. Anschließend wird bei voller Geschwindigkeit auf den Folgefahrsatz umgeschaltet.
1	0	0	0	1	Fliegendes Umschalten auf den Folgeauftrag in der Zielposition. Der Umschaltzeitpunkt auf den Folgefahrauftrag wird so vorverlegt, dass in der Zielposition des ersten Fahrsatzes die Zielgeschwindigkeit des Folgefahrsatzes bereits erreicht wird.
0	1	0	0	0	Umschalten auf Folgefahrauftrag mit Abbremsen. Der Antrieb bremsst in die Zielposition des ersten Fahrsatzes. Der Folgefahrsatz wird gestartet, wenn der dazu definierte Eingang (Funktion INxMODE=15) den Zustand LOW erreicht hat.
0	1	1	0	0	Umschalten auf Folgefahrauftrag mit Abbremsen. Der Antrieb bremsst in die Zielposition des ersten Fahrsatzes. Der Folgefahrsatz wird gestartet, wenn der dazu definierte Eingang (Funktion INxMODE=15) den Zustand HIGH erreicht hat.
0	0	0	1	0	Umschalten auf Folgefahrauftrag mit Abbremsen. Der Antrieb bremsst in die Zielposition des ersten Fahrsatzes. Der Folgefahrsatz wird gestartet, nachdem die programmierte Verzögerungszeit (O_FT) abgelaufen ist.
0	1	0	1	0	Umschalten auf Folgefahrauftrag mit Abbremsen. Der Antrieb bremsst in die Zielposition des ersten Fahrsatzes. Der Folgefahrsatz wird gestartet, wenn der dazu definierte Eingang (Funktion INxMODE=15) den Zustand LOW erreicht hat oder nachdem die programmierte Verzögerungszeit (O_FT) abgelaufen ist.
0	1	1	1	0	Umschalten auf Folgefahrauftrag mit Abbremsen. Der Antrieb bremsst in die Zielposition des ersten Fahrsatzes. Der Folgefahrsatz wird gestartet, wenn der dazu definierte Eingang (Funktion INxMODE=15) den Zustand HIGH erreicht hat oder nachdem die programmierte Verzögerungszeit (O_FT) abgelaufen ist.

<b>Subindex</b>	04 <sub>h</sub>
<b>Kurzbeschreibung</b>	Trajektorie
<b>Einheit</b>	—
<b>Zugriff</b>	rw
<b>PDO mapping</b>	möglich (vorgemappt auf wählbares RPDO 33)
<b>Datentyp</b>	INTEGER32
<b>Wertebereich</b>	$-(2^{31}-1)..(2^{31}-1)$
<b>Defaultwert</b>	0

In Vorbereitung

<b>Subindex</b>	05 <sub>h</sub>
<b>Kurzbeschreibung</b>	Fahrauftragsnummer
<b>Einheit</b>	—
<b>Zugriff</b>	rw
<b>PDO mapping</b>	möglich (vorgemappt auf wählbares RPDO 35)
<b>Datentyp</b>	UNSIGNED16
<b>Wertebereich</b>	1..180, 129..255
<b>Defaultwert</b>	0

Über diesen Index wird die Nummer des angewählten Fahrauftrags vorgegeben. Dabei sind die Fahraufträge 1 bis 180 EEPROM – Fahrsätze, die Fahraufträge 192 bis 255 sind RAM – Fahraufträge. Die RAM – Fahraufträge werden beim Einschalten oder bei einem Reset des Servoverstärkers mit den ersten 64 EEPROM – Fahraufträgen gefüllt. Der Fahrauftrag 0 ist ebenfalls ein RAM – Fahrauftrag, der zum einen als Kopierpuffer für Fahraufträge zum anderen zum Eintragen von Fahrauftragsdaten für den Direktfahrauftrag (RPDO 34) Verwendung findet.

<b>Subindex</b>	06 <sub>h</sub>
<b>Kurzbeschreibung</b>	Anfahrzeit (Beschleunigung)
<b>Einheit</b>	ms
<b>Zugriff</b>	rw
<b>PDO mapping</b>	nicht möglich
<b>Datentyp</b>	UNSIGNED16
<b>Wertebereich</b>	1..65535
<b>Defaultwert</b>	0

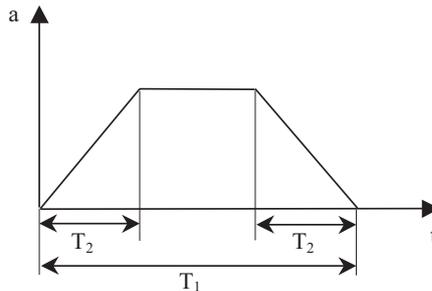
Über diesen Index wird die Gesamtzeit angegeben, um auf die Zielgeschwindigkeit des Fahrauftrags zu kommen. Über die Wahl des Wertes des Subindex 08<sub>h</sub> wird die Form der Beschleunigungsrampe eingestellt.

<b>Subindex</b>	07 <sub>h</sub>
<b>Kurzbeschreibung</b>	Bremszeit (Verzögerung)
<b>Einheit</b>	ms
<b>Zugriff</b>	rw
<b>PDO mapping</b>	nicht möglich
<b>Datentyp</b>	UNSIGNED16
<b>Wertebereich</b>	1..65535
<b>Defaultwert</b>	0

Über diesen Index wird die Gesamtzeit angegeben, um die Geschwindigkeit zur Zielposition auf 0 zu bringen. Über die Wahl des Wertes des Subindex 09<sub>h</sub> wird die Form der Beschleunigungsrampe eingestellt.

<b>Subindex</b>	08 <sub>h</sub>
<b>Kurzbeschreibung</b>	Ruckbegrenzung (Beschleunigung) (i.V)
<b>Einheit</b>	ms
<b>Zugriff</b>	rw
<b>PDO mapping</b>	nicht möglich
<b>Datentyp</b>	UNSIGNED16
<b>Wertebereich</b>	1..65535
<b>Defaultwert</b>	0

Über diesen Index wird die Form der Beschleunigungsrampe vorgegeben. Der Wert muss kleiner als die Hälfte der Anfahrzeit (Subindex 06<sub>h</sub>) gewählt werden. Folgende Darstellung dient der Veranschaulichung:



Dabei entspricht  $T_1$  dem Subindex 06<sub>h</sub>,  $T_2$  dem Subindex 08<sub>h</sub>. Bei  $T_2 = 0$  wird eine Trapezrampe gefahren, bei  $T_2 = \frac{T_1}{2}$  wird in etwa eine  $\sin^2$  – Rampe gefahren.

<b>Subindex</b>	09 <sub>h</sub>
<b>Kurzbeschreibung</b>	Ruckbegrenzung (Verzögerung)(i.V.)
<b>Einheit</b>	ms
<b>Zugriff</b>	rw
<b>PDO mapping</b>	nicht möglich
<b>Datentyp</b>	UNSIGNED16
<b>Wertebereich</b>	0..65535
<b>Defaultwert</b>	0

Über diesen Index wird die Form der Bremsrampe vorgegeben. Der Wert muss kleiner als die Hälfte der Bremszeit (Subindex 07<sub>h</sub>) gewählt werden. Die Ruckbegrenzung wirkt hier sinngemäß wie bei der Begrenzung für das Anfahren.

<b>Subindex</b>	0A <sub>h</sub>
<b>Kurzbeschreibung</b>	Nummer des Folgefahrauftrages
<b>Einheit</b>	—
<b>Zugriff</b>	rw
<b>PDO mapping</b>	nicht möglich
<b>Datentyp</b>	UNSIGNED16
<b>Wertebereich</b>	0..180, 192..255
<b>Defaultwert</b>	0

Über diesen Index wird die Nummer eines Folgefahrauftrags eingestellt. Ob mit diesem fortgefahren wird, wird über den Subindex 03<sub>h</sub>, Bit 3 eingestellt.

<b>Subindex</b>	0B <sub>h</sub>
<b>Kurzbeschreibung</b>	Startverzögerung Folgefahrauftrag
<b>Einheit</b>	ms
<b>Zugriff</b>	rw
<b>PDO mapping</b>	nicht möglich
<b>Datentyp</b>	UNSIGNED16
<b>Wertebereich</b>	1..65535
<b>Defaultwert</b>	0

Über dieses Objekt wird eine Verzögerungszeit eingestellt, mit der der Folgefahrauftrag gestartet wird. Dazu muss diese Funktion über Subindex 03<sub>h</sub>, Bit 7 freigegeben werden.

<b>Subindex</b>	0C <sub>h</sub>
<b>Kurzbeschreibung</b>	Kopieren eines Fahrauftrages
<b>Einheit</b>	—
<b>Zugriff</b>	wo
<b>PDO mapping</b>	nicht möglich
<b>Datentyp</b>	2 x UNSIGNED16
<b>Wertebereich</b>	Jeweils 0..180, 192..255
<b>Defaultwert</b>	0, 0

Über dieses Objekt können Fahraufträge kopiert werden. Die Nummer, die zuerst im CAN – Telegramm erscheint, beschreibt dabei den Quell – Fahrauftrag, die folgende Nummer den Ziel – Fahrauftrag.

<b>Subindex</b>	0D <sub>h</sub>
<b>Kurzbeschreibung</b>	Wichtungsfaktor Geschwindigkeit
<b>Einheit</b>	—
<b>Zugriff</b>	rw
<b>PDO mapping</b>	nicht möglich
<b>Datentyp</b>	UNSIGNED16
<b>Wertebereich</b>	0..65535
<b>Defaultwert</b>	1

Über dieses Objekt wird ein Multiplikator für die im RPDO Fahrsatz angegebene Geschwindigkeit eingestellt.

<b>Subindex</b>	0E <sub>h</sub>
<b>Kurzbeschreibung</b>	Geschwindigkeit Direktfahrauftrag
<b>Einheit</b>	Inkmente / 250µs, bzw. Auflösungsabhängig
<b>Zugriff</b>	rw
<b>PDO mapping</b>	nicht möglich
<b>Datentyp</b>	INTEGER32
<b>Wertebereich</b>	$-(2^{31}-1)..(2^{31}-1)$
<b>Defaultwert</b>	0

Über dieses Objekt wird die Geschwindigkeit für den Direktfahrauftrag (Fahrsatz 0) vorgegeben. Die Fahrauftragsart bestimmt dann, ob die Geschwindigkeit inkrementell oder als SI-Einheit ausgewertet wird.

<b>Subindex</b>	0F <sub>h</sub>
<b>Kurzbeschreibung</b>	Sollposition CAN-Master-Slave
<b>Einheit</b>	Inkmente / 250µs, bzw. Auflösungsabhängig
<b>Zugriff</b>	rw
<b>PDO mapping</b>	möglich
<b>Datentyp</b>	INTEGER32
<b>Wertebereich</b>	$-(2^{31}-1)..(2^{31}-1)$
<b>Defaultwert</b>	0

Über dieses Objekt wird die Zielposition in Lagereglerinkrementen ( $2^{20}$  Inkmente / Motorumdrehung) im Mode elektronisches Getriebe vorgegeben. Dieses Objekt ist im fest vorgegebenen Rx-PDO 41 eingetragen.

## 4.9 Objekt 2050h: Hilfsvariable für digitale Eingänge

<b>Index</b>	2050h
<b>Kurzbeschreibung</b>	Hilfsvariablen der digitalen Eingänge
<b>Objektcode</b>	ARRAY
<b>Anzahl Elemente</b>	6

Beschreibung der Subindizes:

<b>Subindex</b>	01h...04h
<b>Kurzbeschreibung</b>	Triggervariable Eingang 1...4
<b>Einheit</b>	—
<b>Zugriff</b>	rw
<b>PDO mapping</b>	nicht möglich
<b>Datentyp</b>	INTEGER32
<b>Wertebereich</b>	INTEGER32
<b>Defaultwert</b>	0

<b>Subindex</b>	05h
<b>Kurzbeschreibung</b>	Trigger-Hilfsvariable für elektronisches Getriebe
<b>Einheit</b>	—
<b>Zugriff</b>	wo
<b>PDO mapping</b>	möglich
<b>Datentyp</b>	INTEGER32
<b>Wertebereich</b>	INTEGER32
<b>Defaultwert</b>	0

Über dieses Objekt wird im Mode elektronisches Getriebe (0xF7) die Wegstrecke für das Aufsynchronisieren des Slave-Antriebs auf die Mastergeschwindigkeit angegeben, wenn der Parameter ENGAGE einen Wert von 3 hat. Die Angabe erfolgt in Inkrementen (Eine Motorumdrehung entspricht  $2^{20}$  Inkrementen).

## 4.10 Latchfunktion

Objekt 2026 gibt die Latchfunktion frei, mit den Objekten 2082 bis 2085 können die gelatchten Positionen (positive oder negative Flanke) des digitalen Eingangs 2 in verschiedenen Auflösungen gelesen werden.

### 4.10.1 Objekt 2026h: Latchfreigabe

<b>Index</b>	2026 <sub>h</sub>
<b>Kurzbeschreibung</b>	Freigabe der Latchfunktion für CAN
<b>Objektcode</b>	RECORD
<b>Anzahl Elemente</b>	2

Subindizes:

<b>Subindex</b>	01 <sub>h</sub>
<b>Kurzbeschreibung</b>	Freigabe der Latchfunktion für CAN
<b>Einheit</b>	—
<b>Zugriff</b>	rw
<b>PDO mapping</b>	nicht möglich
<b>Datentyp</b>	UNSIGNED8
<b>Wertebereich</b>	0, 1
<b>Defaultwert</b>	1

Der SERVOSTAR 400/600 bietet die Möglichkeit über einen Latch - Eingang (Eingang 2, IN2MODE 26, auch konfigurierbar über Objekt 3565<sub>h</sub>, Subindex 01<sub>h</sub>) eine Istposition hochgenau (Erfassungszeit < 1 µs) zu erfassen. Über das Objekt 2026<sub>h</sub> kann eingestellt werden, ob ein Latchimpuls über CAN gemeldet werden soll. Ein Wert von 0 bedeutet Sperren.

### 4.10.2 Objekt 2082h: 32/24-bit Latch positive

Diese Objekte liefern die gelatchte Position bei positiver Flanke am Digitaleingang 1 im 32-bit Format.

<b>Index</b>	2082 <sub>h</sub>
<b>Kurzbeschreibung</b>	32/24-bit Latch positive
<b>Objektcode</b>	RECORD
<b>Anzahl Elemente</b>	3

Subindizes:

<b>Subindex</b>	01 <sub>h</sub>
<b>Kurzbeschreibung</b>	32 Bit Latch positiv
<b>Einheit</b>	—
<b>Zugriff</b>	rw
<b>PDO mapping</b>	nicht möglich
<b>Datentyp</b>	INTEGER32
<b>Wertebereich</b>	INTEGER32
<b>Defaultwert</b>	0

<b>Subindex</b>	02 <sub>h</sub>
<b>Kurzbeschreibung</b>	24 Bit Latch positiv
<b>Einheit</b>	—
<b>Zugriff</b>	rw
<b>PDO mapping</b>	nicht möglich
<b>Datentyp</b>	INTEGER24
<b>Wertebereich</b>	INTEGER24
<b>Defaultwert</b>	0

### 4.10.3 Objekt 2083h: 32/24-bit Latch negative

Diese Objekte liefern die gelatchte Position bei negativer Flanke am Digitaleingang 1 im 32-bit Format.

<b>Index</b>	2083 <sub>h</sub>
<b>Kurzbeschreibung</b>	32/24-bit Latch negative
<b>Objektcode</b>	RECORD
<b>Anzahl Elemente</b>	3

Subindizes:

<b>Subindex</b>	01 <sub>h</sub>
<b>Kurzbeschreibung</b>	32 Bit Latch negativ
<b>Einheit</b>	—
<b>Zugriff</b>	rw
<b>PDO mapping</b>	nicht möglich
<b>Datentyp</b>	INTEGER32
<b>Wertebereich</b>	INTEGER32
<b>Defaultwert</b>	0

<b>Subindex</b>	02 <sub>h</sub>
<b>Kurzbeschreibung</b>	24 Bit Latch negativ
<b>Einheit</b>	—
<b>Zugriff</b>	rw
<b>PDO mapping</b>	nicht möglich
<b>Datentyp</b>	INTEGER24
<b>Wertebereich</b>	INTEGER24
<b>Defaultwert</b>	0

### 4.10.4 Objekt 2084h: 16-bit Latch positive

Diese Objekte liefern die gelatchte Position bei positiver Flanke am Digitaleingang 1 im 16-bit Format.

<b>Index</b>	2084 <sub>h</sub>
<b>Kurzbeschreibung</b>	16-bit Latch positiv
<b>Objektcode</b>	RECORD
<b>Anzahl Elemente</b>	2

Subindizes:

<b>Subindex</b>	01 <sub>h</sub>
<b>Kurzbeschreibung</b>	16 Bit Latch positiv
<b>Einheit</b>	—
<b>Zugriff</b>	rw
<b>PDO mapping</b>	nicht möglich
<b>Datentyp</b>	INTEGER16
<b>Wertebereich</b>	INTEGER16
<b>Defaultwert</b>	0

**4.10.5 Objekt 2085h: 16-bit Latch negative**

Diese Objekte liefern die gelatchte Position bei negativer Flanke am Digitaleingang 1 im 16-bit Format.

<b>Index</b>	2085h
<b>Kurzbeschreibung</b>	16-bit Latch negativ
<b>Objektcode</b>	RECORD
<b>Anzahl Elemente</b>	2

Subindizes:

<b>Subindex</b>	01h
<b>Kurzbeschreibung</b>	16-bit Latch negativ
<b>Einheit</b>	—
<b>Zugriff</b>	rw
<b>PDO mapping</b>	nicht möglich
<b>Datentyp</b>	INTEGER16
<b>Wertebereich</b>	INTEGER16
<b>Defaultwert</b>	0

#### 4.10.6 Objekt 2087h: Latch Positionen Digitaleingang 1

Die Objekte unter Index 2087 dienen zum Auslesen der Latchpositionen des Digitaleingangs 1 und zur Freigabe der Latchfunktionen der Digitaleingänge 1 und 2.

<b>Index</b>	2087h
<b>Kurzbeschreibung</b>	Latch Positionen Digitaleingang 1
<b>Objektcode</b>	RECORD
<b>Anzahl Elemente</b>	4

Subindizes:

<b>Subindex</b>	01h
<b>Kurzbeschreibung</b>	Latchposition Digitaleingang 1, positiv
<b>Einheit</b>	—
<b>Zugriff</b>	ro
<b>PDO mapping</b>	möglich
<b>Datentyp</b>	INTEGER32
<b>Wertebereich</b>	INTEGER32
<b>Defaultwert</b>	0

<b>Subindex</b>	02h
<b>Kurzbeschreibung</b>	Latchposition Digitaleingang 1, negativ
<b>Einheit</b>	—
<b>Zugriff</b>	ro
<b>PDO mapping</b>	möglich
<b>Datentyp</b>	INTEGER32
<b>Wertebereich</b>	INTEGER32
<b>Defaultwert</b>	0

<b>Subindex</b>	03h
<b>Kurzbeschreibung</b>	Latchfreigaben für Eingänge 1 und 2
<b>Einheit</b>	—
<b>Zugriff</b>	rw
<b>PDO mapping</b>	möglich
<b>Datentyp</b>	UNSIGNED8
<b>Wertebereich</b>	0...15
<b>Defaultwert</b>	0

Dieses mappbare Objekt dient der Freigabe der positiven bzw. negativen Flanken der Latcheingänge 1 bzw. 2. Dazu muss für diese Eingänge die Latchfunktion (INxMODE = 26) konfiguriert sein. Die folgenden Zuordnungen zwischen Freigabebits und Funktion bestehen:

Bit	Freigabe
0	Freigabe positives Latchereignis Digitaleingang 1
1	Freigabe negatives Latchereignis Digitaleingang 1
2	Freigabe positives Latchereignis Digitaleingang 2
3	Freigabe negatives Latchereignis Digitaleingang 2

## 4.11 Herstellerspezifische Istwerte

### 4.11.1 Objekt 2070h: Istwerte

Über diesen Index werden relevante Istwerte des SERVOSTAR 400/600 zur Verfügung gestellt.

<b>Index</b>	2070 <sub>h</sub>
<b>Kurzbeschreibung</b>	Istwerte
<b>Objektcode</b>	RECORD
<b>Anzahl Elemente</b>	25

Subindizes:

<b>Subindex</b>	01 <sub>h</sub>
<b>Kurzbeschreibung</b>	Istlage
<b>Einheit</b>	—
<b>Zugriff</b>	ro
<b>PDO mapping</b>	möglich (vorgemappt auf wählbare TPDO 22, TPDO 32)
<b>Datentyp</b>	UNSIGNED32
<b>Wertebereich</b>	0..16777215
<b>Defaultwert</b>	0

Über diesen Index kann die Motorposition innerhalb von 16 Umdrehungen eingelesen werden. Eine Umdrehung wird dabei mit einer Inkrementzahl von 20 Bit aufgelöst. Es gilt also:

$$1 \text{ Umdrehung} \Rightarrow 2^{20} \text{ Inkremente} \Rightarrow 1048576 \text{ Inkremente}$$

<b>Subindex</b>	02 <sub>h</sub>
<b>Kurzbeschreibung</b>	Istdrehzahl
<b>Einheit</b>	min <sup>-1</sup>
<b>Zugriff</b>	ro
<b>PDO mapping</b>	möglich (vorgemappt auf wählbare TPDO 22, TPDO 32)
<b>Datentyp</b>	UNSIGNED32
<b>Wertebereich</b>	0..1677215
<b>Defaultwert</b>	0

Einlesen der Motordrehzahl. Der Drehzahlwert entspricht dabei:

$$n[\text{min}^{-1}] = \frac{1875}{262144} \times \text{eingeliesener Drehzahlwert}$$

<b>Subindex</b>	03 <sub>h</sub>
<b>Kurzbeschreibung</b>	Inkrementeller Positionswert
<b>Einheit</b>	—
<b>Zugriff</b>	ro
<b>PDO mapping</b>	möglich (vorgemappt auf wählbares TPDO 33)
<b>Datentyp</b>	INTEGER32
<b>Wertebereich</b>	-(2 <sup>31</sup> -1)..(2 <sup>31</sup> -1)
<b>Defaultwert</b>	0

Lesen des Inkrementalwerts der Istposition. Eine Umdrehung wird dabei mit einer Inkrementzahl von 20 Bit aufgelöst. Es gilt also:

$$1 \text{ Umdrehung} \Rightarrow 2^{20} \text{ Inkremente} = 1048576 \text{ Inkremente}$$

<b>Subindex</b>	04 <sub>h</sub>
<b>Kurzbeschreibung</b>	Lesen des 16 Bit Positionslatch
<b>Einheit</b>	—
<b>Zugriff</b>	ro
<b>PDO mapping</b>	möglich
<b>Datentyp</b>	INTEGER16
<b>Wertebereich</b>	-(2 <sup>15</sup> )..(2 <sup>15</sup> -1)
<b>Defaultwert</b>	0

Lesen der gespeicherten (gelatchten) 16 Bit Position. Die Position wird in Inkrementen innerhalb einer Umdrehung ausgegeben. Die Ausgabe wird nicht von den Getriebefaktoren oder den Factor Groups beeinflusst.

<b>Subindex</b>	05h
<b>Kurzbeschreibung</b>	Lesen des 32 Bit Positions latch
<b>Einheit</b>	—
<b>Zugriff</b>	ro
<b>PDO mapping</b>	möglich
<b>Datentyp</b>	INTEGER32
<b>Wertebereich</b>	$-(2^{31})..(2^{31}-1)$
<b>Defaultwert</b>	0

Lesen der gespeicherten (gelatchten) 32 Bit Position. Die Position wird in Inkrementen innerhalb einer Umdrehung ausgegeben. Die Ausgabe wird nicht von den Getriebefaktoren oder den Factor Groups beeinflusst.

<b>Subindex</b>	06h
<b>Kurzbeschreibung</b>	SI Positionswert
<b>Einheit</b>	$\mu\text{m}$
<b>Zugriff</b>	ro
<b>PDO mapping</b>	möglich
<b>Datentyp</b>	INTEGER32
<b>Wertebereich</b>	$-(2^{31}-1)..(2^{31}-1)$
<b>Defaultwert</b>	0

Lesen der Istposition als SI – Einheit. Das Verhältnis von real zurückgelegter Strecke zu Motorumdrehungen wird durch

$$S_{SI} = S_{Inkr} \times \frac{PGEARI}{PGEARO} \quad \text{bestimmt, wobei PGEARO (Objekt 2020h, Subindex 08h)}$$

die Anzahl der Inkremente enthält, die gefahren werden, wenn die zu fahrende Strecke PGEARI (Objekt 2020h, Subindex 09h) beträgt. Zu beachten ist, dass dabei eine Motorumdrehung einer Inkrementzahl von  $2^{20} = 1048576$  entspricht.

<b>Subindex</b>	07h
<b>Kurzbeschreibung</b>	SI Geschwindigkeitswert
<b>Einheit</b>	$\mu\text{m/s}$
<b>Zugriff</b>	ro
<b>PDO mapping</b>	nicht möglich
<b>Datentyp</b>	INTEGER32
<b>Wertebereich</b>	$-(2^{31}-1)..(2^{31}-1)$
<b>Defaultwert</b>	0

Lesen der Istgeschwindigkeit in SI – Einheiten gelesen

<b>Subindex</b>	08h
<b>Kurzbeschreibung</b>	Schleppfehler
<b>Einheit</b>	$\mu\text{m}$
<b>Zugriff</b>	ro
<b>PDO mapping</b>	nicht möglich
<b>Datentyp</b>	INTEGER32
<b>Wertebereich</b>	$-(2^{31}-1)..(2^{31}-1)$
<b>Defaultwert</b>	0

Lesen des momentan gemessenen Schleppfehlers in SI – Einheiten.

<b>Subindex</b>	09h
<b>Kurzbeschreibung</b>	Effektivstrom
<b>Einheit</b>	mA
<b>Zugriff</b>	ro
<b>PDO mapping</b>	nicht möglich
<b>Datentyp</b>	UNSIGNED32
<b>Wertebereich</b>	$0..2 * \text{Nennstrom [mA]}$
<b>Defaultwert</b>	0

Lesen des momentan gemessenen Effektivstroms.

<b>Subindex</b>	0A <sub>h</sub>
<b>Kurzbeschreibung</b>	Drehzahl
<b>Einheit</b>	min <sup>-1</sup>
<b>Zugriff</b>	ro
<b>PDO mapping</b>	nicht möglich
<b>Datentyp</b>	INTEGER32
<b>Wertebereich</b>	-(2 <sup>31</sup> -1)..(2 <sup>31</sup> -1)
<b>Defaultwert</b>	0

Lesen der momentan gemessenen Drehzahl.

<b>Subindex</b>	0B <sub>h</sub>
<b>Kurzbeschreibung</b>	Kühlkörpertemperatur
<b>Einheit</b>	°C
<b>Zugriff</b>	ro
<b>PDO mapping</b>	nicht möglich
<b>Datentyp</b>	INTEGER32
<b>Wertebereich</b>	-(2 <sup>31</sup> -1)..(2 <sup>31</sup> -1)
<b>Defaultwert</b>	0

Lesen der Temperatur des Kühlkörpers.

<b>Subindex</b>	0C <sub>h</sub>
<b>Kurzbeschreibung</b>	Innentemperatur
<b>Einheit</b>	°C
<b>Zugriff</b>	ro
<b>PDO mapping</b>	nicht möglich
<b>Datentyp</b>	INTEGER32
<b>Wertebereich</b>	-(2 <sup>31</sup> -1)..(2 <sup>31</sup> -1)
<b>Defaultwert</b>	0

Lesen der Innentemperatur im Servoverstärker.

<b>Subindex</b>	0D <sub>h</sub>
<b>Kurzbeschreibung</b>	Zwischenkreisspannung
<b>Einheit</b>	V
<b>Zugriff</b>	ro
<b>PDO mapping</b>	nicht möglich
<b>Datentyp</b>	INTEGER32
<b>Wertebereich</b>	-(2 <sup>31</sup> -1)..(2 <sup>31</sup> -1)
<b>Defaultwert</b>	0

Lesen der momentan gemessenen Zwischenkreisspannung.

<b>Subindex</b>	0E <sub>h</sub>
<b>Kurzbeschreibung</b>	Bremsleistung
<b>Einheit</b>	W
<b>Zugriff</b>	ro
<b>PDO mapping</b>	nicht möglich
<b>Datentyp</b>	INTEGER32
<b>Wertebereich</b>	-(2 <sup>31</sup> -1)..(2 <sup>31</sup> -1)
<b>Defaultwert</b>	0

Lesen der momentan gemessenen Brems(Ballast)leistung.

<b>Subindex</b>	0F <sub>h</sub>
<b>Kurzbeschreibung</b>	I <sup>2</sup> T-Belastung
<b>Einheit</b>	%
<b>Zugriff</b>	ro
<b>PDO mapping</b>	nicht möglich
<b>Datentyp</b>	INTEGER32
<b>Wertebereich</b>	-(2 <sup>31</sup> -1)..(2 <sup>31</sup> -1)
<b>Defaultwert</b>	0

Lesen der I<sup>2</sup>t – Belastung.

<b>Subindex</b>	10 <sub>h</sub>
<b>Kurzbeschreibung</b>	Betriebsdauer
<b>Einheit</b>	min
<b>Zugriff</b>	ro
<b>PDO mapping</b>	nicht möglich
<b>Datentyp</b>	INTEGER32
<b>Wertebereich</b>	$-(2^{31}-1)..(2^{31}-1)$
<b>Defaultwert</b>	0

Lesen des Betriebsstundenzählers des Servoverstärkers.

<b>Subindex</b>	11 <sub>h</sub>
<b>Kurzbeschreibung</b>	erweiterter Status für TPDO 33
<b>Einheit</b>	—
<b>Zugriff</b>	ro
<b>PDO mapping</b>	möglich
<b>Datentyp</b>	UNSIGNED32
<b>Wertebereich</b>	$0..(2^{32}-1)$
<b>Defaultwert</b>	—

Beschreibung:

Bit	Wert	Beschreibung
0	0x00000001	Status Eingang 1
1	0x00000002	Status Eingang 2
2	0x00000004	Status Eingang 3
3	0x00000008	Status Eingang 4
4	0x00000010	reserviert
5	0x00000020	reserviert
6	0x00000040	reserviert
7	0x00000080	reserviert
8	0x00000100	reserviert
9	0x00000200	reserviert
10	0x00000400	reserviert
11	0x00000800	reserviert
12	0x00001000	reserviert
13	0x00002000	reserviert
14	0x00004000	reserviert
15	0x00008000	reserviert
16	0x00010000	Auftrag aktiv (Lageregelung)
17	0x00020000	Referenzpunkt gesetzt
18	0x00040000	Home-Position
19	0x00080000	In-Position
20	0x00100000	Positionslatch erfolgt
21	0x00200000	frei
22	0x00400000	Meldung Position 1
23	0x00800000	Meldung Position 2
24	0x01000000	Meldung Position 3
25	0x02000000	Meldung Position 4
26	0x04000000	Initialisierung beendet
27	0x08000000	frei
28	0x10000000	Motorstillstand
29	0x20000000	Sicherheitsrelais
30	0x40000000	Endstufe freigegeben
31	0x80000000	Fehler steht an

<b>Subindex</b>	12h
<b>Kurzbeschreibung</b>	Absolutwertgeberposition in SI-Einheiten
<b>Einheit</b>	Benutzereinheiten
<b>Zugriff</b>	ro
<b>PDO mapping</b>	möglich
<b>Datentyp</b>	INTEGER32
<b>Wertebereich</b>	$-(2^{31}-1)..(2^{31}-1)$
<b>Defaultwert</b>	—

Falls ein Absolutwertgeber im Gerät konfiguriert wurde (EXTPOS 1, 2 oder 3), kann über dieses Objekt die Geberposition ermittelt werden. Die Skalierung wird über die Parameter PGEARI, PGEARO, ENCIN und EXTMUL festgelegt.

<b>Subindex</b>	13h
<b>Kurzbeschreibung</b>	Interner Positionssollwert
<b>Einheit</b>	Interne Lagecounts
<b>Zugriff</b>	ro
<b>PDO mapping</b>	möglich
<b>Datentyp</b>	INTEGER32
<b>Wertebereich</b>	$-(2^{31}-1)..(2^{31}-1)$
<b>Defaultwert</b>	—

Lesen des für die interne Lageregelung verwendeten Positionssollwert. Der Wert wird in internen Lagecounts ausgegeben.

<b>Subindex</b>	14h
<b>Kurzbeschreibung</b>	Istwert analoger Eingang 1
<b>Einheit</b>	mV (skaliert)
<b>Zugriff</b>	ro
<b>PDO mapping</b>	möglich
<b>Datentyp</b>	INTEGER16
<b>Wertebereich</b>	$-(2^{15}-1)..(2^{15}-1)$
<b>Defaultwert</b>	—

Liefert einen skalierten Wert der Eingangsspannung am analogen Eingang 1. 10 V entsprechen dabei 8192 Inkrementen.

<b>Subindex</b>	15h
<b>Kurzbeschreibung</b>	Istwert analoger Eingang 2
<b>Einheit</b>	mV (skaliert)
<b>Zugriff</b>	ro
<b>PDO mapping</b>	möglich
<b>Datentyp</b>	INTEGER16
<b>Wertebereich</b>	$-(2^{15}-1)..(2^{15}-1)$
<b>Defaultwert</b>	—

Liefert einen skalierten Wert der Eingangsspannung am analogen Eingang 2. 10 V entsprechen dabei 8192 Inkrementen.

<b>Subindex</b>	16h
<b>Kurzbeschreibung</b>	Fehlerregister
<b>Einheit</b>	—
<b>Zugriff</b>	ro
<b>PDO mapping</b>	möglich
<b>Datentyp</b>	UNSIGNED32
<b>Wertebereich</b>	0..(2 <sup>32</sup> -1)
<b>Defaultwert</b>	0

Liefert den Inhalt des Fehlerregisters ERRCODE. Folgende Fehlermeldungen sind vorhanden:

Bit	Nummer	Bezeichnung	Erklärung
0	F01*	Kühlkörpertemperatur	Kühlkörpertemperatur zu hoch. Grenzwert vom Hersteller auf 80°C eingestellt
1	F02*	Überspannung	Überspannung im Zwischenkreis. Grenzwert abhängig von der Netzspannung
2	F03*	Schleppfehler	Meldung des Lagereglers
3	F04	Rückführung	Kabelbruch, Kurzschluss, Erdschluss
4	F05*	Unterspannung	Unterspannung im Zwischenkreis. Grenzwert vom Hersteller auf 100V eingestellt
5	F06	Motortemperatur	Temperaturfühler defekt oder Motortemperatur zu hoch. Grenzwert vom Hersteller auf 145°C eingestellt
6	F07	Reserve	Reserve
7	F08*	Überdrehzahl	Motor geht durch, Drehzahl unzulässig hoch
8	F09	EEPROM	Checksummenfehler
9	F10	Flash-EEPROM	Checksummenfehler
10	F11	Bremse	Kabelbruch, Kurzschluss, Erdschluss
11	F12	Motorphase	Motorphase fehlt (Leitungsbruch o.ä.)
12	F13*	Innentemperatur	Innentemperatur zu hoch
13	F14	Endstufe	Fehler in der Leistungsendstufe
14	F15	I <sup>2</sup> t max.	I <sup>2</sup> t-Maximalwert überschritten
15	F16*	Netz-BTB	Fehlen von 2 oder 3 Phasen der Einspeisung
16	F17	A/D-Konverter	Fehler in der analog-digital-Wandlung, oft hervorgerufen durch sehr starke elektromagnetische Störungen
17	F18	Bremsschaltung	Bremsschaltung defekt oder Einstellung fehlerhaft
18	F19*	Netzphase	Fehlen von einer Phase der Einspeisung (Abschaltbar für den Betrieb an zwei Phasen)
19	F20	Slotfehler	Slotfehler (Hardwarefehler der Erweiterungskarte)
20	F21	Handlingfehler	Softwarefehler der Erweiterungskarte
21	F22	reserviert	reserviert
22	F23	CAN Bus aus	Schwerwiegender CAN Bus Kommunikationsfehler
23	F24	Warnung	Warnungsanzeige wird als Fehler gewertet
24	F25	Kommutierungsfehler	Kommutierungsfehler
25	F26	Endschalter	Referenzfahrt-Fehler (Hardware-Endschalter erreicht)
26	F27	Option AS	Fehler bei der Bedienung der Option -AS-, Eingänge AS-ENABLE und ENABLE wurden gleichzeitig gesetzt
27	F28	Reserve	Reserve
28	F29	Feldbus Sync	Feldbus nicht synchronisiert
29	F30	Emergency Timeout	Timeout Not-Stopp
30	F31	Reserve	Reserve
31	F32	Systemfehler	Systemsoftware reagiert nicht korrekt

\* Diese Fehlermeldungen können ohne Reset mit dem ASCII-Commando CLRFAULT zurückgesetzt werden.

Wenn nur einer dieser Fehler anliegt und der RESET-Button oder die I/O-Funktion RESET verwendet wird, wird ebenfalls nur das Kommando CLRFAULT ausgeführt. Werden die anderen Fehler resetiert, wird der Antrieb neu gestartet (COLDSTART).

<b>Subindex</b>	17h
<b>Kurzbeschreibung</b>	Interner Geschwindigkeitssollwert
<b>Einheit</b>	—
<b>Zugriff</b>	ro
<b>PDO mapping</b>	möglich
<b>Datentyp</b>	INTEGER32
<b>Wertebereich</b>	$-(2^{31}-1)..(2^{31}-1)$
<b>Defaultwert</b>	—

Auslesen des internen Geschwindigkeitssollwerts. Ein interner Geschwindigkeitscount entspricht dabei einem Positionscout / 250 Mikrosekunden.

Die Geschwindigkeit in rpm ergibt sich daraus zu:

$$v_{rpm} = \frac{v_{incr} \cdot 4000}{2^{20}} \cdot 60$$

<b>Subindex</b>	18h
<b>Kurzbeschreibung</b>	Mastergeschwindigkeit (elektronisches Getriebe)
<b>Einheit</b>	—
<b>Zugriff</b>	ro
<b>PDO mapping</b>	möglich
<b>Datentyp</b>	UNSIGNED32
<b>Wertebereich</b>	$0..(2^{32}-1)$
<b>Defaultwert</b>	—

Dieses Objekt liefert die Geschwindigkeit, die sich aus den eingelesenen Mastergeberimpulsen im elektronischen Getriebe (Mode 0xF7) ergibt.

<b>Subindex</b>	19h
<b>Kurzbeschreibung</b>	Steuervariable (elektronisches Getriebe)
<b>Einheit</b>	—
<b>Zugriff</b>	ro
<b>PDO mapping</b>	möglich
<b>Datentyp</b>	INTEGER8
<b>Wertebereich</b>	$-(2^7-1)..(2^7-1)$
<b>Defaultwert</b>	0

Mit diesem Objekt wird der Zustand bzw. der Übergang beim Aufsynchronisieren/Abkoppeln im elektronischen Getriebe angezeigt. Folgende Werte sind definiert (nur bei ENGAGE = 1 oder 3):

Wert	Bedeutung
0	Keine Synchronisation
1	Aufsynchronisieren auf Mastergeschwindigkeit mit Rampe ACCR
2	Auskoppeln mit Rampe DECR
3	Synchronfahrt zum Master
4	Aufsynchronisieren über die Strecke, die über Objekt 2050 sub 5 vorgegeben wurde

#### 4.11.2 Objekt 6077h: Torque actual value

Das aktuelle Drehmoment bezieht sich auf das augenblickliche Drehmoment im Motor. Die Skalierung ist 1/1000 des Nenndrehmoments.

<b>Index</b>	6077h
<b>Kurzbeschreibung</b>	Torque actual value
<b>Objektcode</b>	VAR
<b>Anzahl Elemente</b>	1
<b>Zugriff</b>	ro
<b>PDO mapping</b>	möglich
<b>Wertebereich</b>	INTEGER16
<b>Defaultwert</b>	0

#### 4.11.3 Objekt 60C2h: Interpolation time period

Der Interpolationszeitraum wird für den PLL (phase locked loop) synchronisierten Positionierbetrieb verwendet.

Die Einheit (Subindex 1) der Zeit ist mit  $10^{\text{interpolation time index}_s}$  gegeben, nur Vielfache von 1ms sind erlaubt. Die zwei Werte definieren den internen ASCII-Parameter PTBASE (Vielfache von 250  $\mu$ s).

Voraussetzung für den PLL-synchronisierten Betrieb ist die Einstellung von SYNC SRC = 3 (Synchronisierung über CAN) bzw. FPGA = 3 (nur SERVOSTAR 600)

<b>Index</b>	60C2h
<b>Kurzbeschreibung</b>	Interpolation time period
<b>Objektcode</b>	RECORD
<b>Anzahl Elemente</b>	3

Beschreibung der Subindizes:

<b>Subindex</b>	01h
<b>Beschreibung</b>	Interpolation time units
<b>Zugriff</b>	rw
<b>PDO mapping</b>	möglich
<b>Wertebereich</b>	UNSIGNED8
<b>Defaultwert</b>	1

<b>Subindex</b>	02h
<b>Beschreibung</b>	Interpolation time index
<b>Zugriff</b>	rw
<b>PDO mapping</b>	möglich
<b>Wertebereich</b>	-123 ... 63
<b>Defaultwert</b>	-3

## 4.12 Frei verfügbare, mappbare SPS – Variablen, Objekte 2030h / 2090h

Über die Parameter DPRVAR1 .. DPRVAR16 können Zustandsvariablen zwischen CAN-Bus und internem SPS-Ablaufprogramm ausgetauscht werden. Dazu gibt es je einen Satz von acht Objekten, die in Nur-Schreib- (WO) bzw. Nur-Lese-Richtung, auch in PDOs gemappt werden können.

### 4.12.1 Objekt 2030h: DP-Ram-Variablen 9-16 (nur schreibbar)

<b>Index</b>	2030 <sub>h</sub>
<b>Kurzbeschreibung</b>	DP-Ram Variablen, nur schreibbar (PDO)
<b>Objektcode</b>	ARRAY
<b>Anzahl Elemente</b>	9

Beschreibung der Subindizes:

<b>Subindex</b>	01 <sub>h</sub> ...08 <sub>h</sub>
<b>Kurzbeschreibung</b>	DP-Ram Variable 9...16
<b>Einheit</b>	—
<b>Zugriff</b>	rw
<b>PDO mapping</b>	möglich
<b>Datentyp</b>	INTEGER32
<b>Wertebereich</b>	INTEGER32
<b>Defaultwert</b>	0

### 4.12.2 Objekt 2090h: DP-Ram-Variablen 1-8 (nur lesbar)

<b>Index</b>	2090 <sub>h</sub>
<b>Kurzbeschreibung</b>	DP-Ram Variablen, nur lesbar (PDO)
<b>Objektcode</b>	ARRAY
<b>Anzahl Elemente</b>	9

Beschreibung der Subindizes:

<b>Subindex</b>	01 <sub>h</sub> ...08 <sub>h</sub>
<b>Kurzbeschreibung</b>	DP-Ram Variable 1...8
<b>Einheit</b>	—
<b>Zugriff</b>	ro
<b>PDO mapping</b>	möglich
<b>Datentyp</b>	INTEGER32
<b>Wertebereich</b>	INTEGER32
<b>Defaultwert</b>	0

## 4.13 Dummy - Variablen, Objekte 2031h / 2071h

Mit Hilfe dieser Objekte können Lücken in Prozeßdatenobjekten (PDOs) realisiert werden. Dies kann z.B. genutzt werden für ein Sollwert-PDO an mehrere Knoten oder für Istwerte, die an bestimmten Stellen im PDO liegen müssen, wobei die anderen Teile des PDOs schnell abgearbeitet werden sollen. Mit Hilfe der Objekte können 1 bis 4 Byte breite Lücken gefüllt werden.

### 4.13.1 Objekt 2031h: Dummy-Variablen für Mappingzwecke (WO)

<b>Index</b>	2031h
<b>Kurzbeschreibung</b>	Dummy-Variablen für Mappingzwecke
<b>Objektcode</b>	RECORD
<b>Anzahl Elemente</b>	5

Beschreibung der Subindizes:

<b>Subindex</b>	01h
<b>Kurzbeschreibung</b>	Write-only dummy 8 bit
<b>Einheit</b>	—
<b>Zugriff</b>	wo
<b>PDO mapping</b>	möglich
<b>Datentyp</b>	UNSIGNED8
<b>Wertebereich</b>	UNSIGNED8
<b>Defaultwert</b>	0

<b>Subindex</b>	02h
<b>Kurzbeschreibung</b>	Write-only dummy 16 bit
<b>Einheit</b>	—
<b>Zugriff</b>	wo
<b>PDO mapping</b>	möglich
<b>Datentyp</b>	UNSIGNED16
<b>Wertebereich</b>	UNSIGNED16
<b>Defaultwert</b>	0

<b>Subindex</b>	03h
<b>Kurzbeschreibung</b>	Write-only dummy 24 bit
<b>Einheit</b>	—
<b>Zugriff</b>	wo
<b>PDO mapping</b>	möglich
<b>Datentyp</b>	UNSIGNED24
<b>Wertebereich</b>	UNSIGNED24
<b>Defaultwert</b>	0

<b>Subindex</b>	04h
<b>Kurzbeschreibung</b>	Write-only dummy 32 bit
<b>Einheit</b>	—
<b>Zugriff</b>	wo
<b>PDO mapping</b>	möglich
<b>Datentyp</b>	UNSIGNED32
<b>Wertebereich</b>	UNSIGNED32
<b>Defaultwert</b>	0

## 4.13.2

## Objekt 2071h: Dummy-Variablen für Mappingzwecke (RO)

<b>Index</b>	2027h
<b>Kurzbeschreibung</b>	Dummy-Variablen für Mappingzwecke
<b>Objektcode</b>	RECORD
<b>Anzahl Elemente</b>	5

Beschreibung der Subindizes:

<b>Subindex</b>	01h
<b>Kurzbeschreibung</b>	Read-only dummy 32 bit
<b>Einheit</b>	—
<b>Zugriff</b>	ro
<b>PDO mapping</b>	möglich
<b>Datentyp</b>	UNSIGNED8
<b>Wertebereich</b>	UNSIGNED8
<b>Defaultwert</b>	0

<b>Subindex</b>	02h
<b>Kurzbeschreibung</b>	Read-only dummy 32 bit
<b>Einheit</b>	—
<b>Zugriff</b>	ro
<b>PDO mapping</b>	möglich
<b>Datentyp</b>	UNSIGNED16
<b>Wertebereich</b>	UNSIGNED16
<b>Defaultwert</b>	0

<b>Subindex</b>	03h
<b>Kurzbeschreibung</b>	Read-only dummy 32 bit
<b>Einheit</b>	—
<b>Zugriff</b>	ro
<b>PDO mapping</b>	möglich
<b>Datentyp</b>	UNSIGNED24
<b>Wertebereich</b>	UNSIGNED24
<b>Defaultwert</b>	0

<b>Subindex</b>	04h
<b>Kurzbeschreibung</b>	Read-only dummy 32 bit
<b>Einheit</b>	—
<b>Zugriff</b>	ro
<b>PDO mapping</b>	möglich
<b>Datentyp</b>	UNSIGNED32
<b>Wertebereich</b>	UNSIGNED32
<b>Defaultwert</b>	0

## 4.14 Profile Velocity Mode (pv) (DS402)

### 4.14.1 Allgemeine Informationen

Der Profile Velocity Mode ermöglicht die Verarbeitung von Geschwindigkeitssollwerten und den zugehörigen Beschleunigungen.

### 4.14.2 Objekte, die in diesem Kapitel definiert werden

Index	Objekt	Name	Type	Zugriff
606Ch	VAR	velocity_actual_value	INTEGER32	ro
60FFh	VAR	target_velocity	INTEGER32	rw

### 4.14.3 Objekte, die in anderen Kapiteln definiert werden

Index	Objekt	Name	Type	Kapitel
6040h	VAR	controlword	INTEGER16	dc (⇒ Kap. 4.4)
6041h	VAR	statusword	UNSIGNED16	dc (⇒ Kap. 4.4)
6063h	VAR	position_actual_value*	INTEGER32	pc (⇒ Kap. 4.15)
6083h	VAR	profile_acceleration	UNSIGNED32	pp (⇒ Kap. 4.17)
6084h	VAR	profile_deceleration	UNSIGNED32	pp (⇒ Kap. 4.17)
6086h	VAR	motion_profile_type	INTEGER16	pp (⇒ Kap. 4.17)
6094h	ARRAY	velocity_encoder_factor	UNSIGNED32	fg (⇒ Kap. 4.5)

### 4.14.4 Objektbeschreibung

#### 4.14.4.1 Objekt 606Ch: velocity\_actual\_value\* (DS402)

Das Objekt "velocity\_actual\_value" repräsentiert die aktuelle Drehzahl. Die Skalierung des Wertes hängt von dem Faktor "velocity\_encoder\_resolution" (Objekt 6096h) ab.

<b>Index</b>	0x606Ch
<b>Name</b>	velocity_actual_value
<b>Objektcode</b>	VAR
<b>Datentyp</b>	INTEGER32
<b>Mode</b>	pv
<b>Zugriff</b>	ro
<b>PDO Mapping</b>	möglich
<b>Einheit</b>	velocity units
<b>Wertebereich</b>	$(-2^{31})..(2^{31}-1)$
<b>Defaultwert</b>	—
<b>EEPROM</b>	nein

#### 4.14.4.2 Objekt 60FFh: target\_velocity (DS402)

Die Soll Drehzahl (target\_velocity) repräsentiert den Sollwert für den Rampengenerator. Die Skalierung des Wertes hängt von dem Faktor "velocity\_encoder\_resolution" (Objekt 6096h) ab.

<b>Index</b>	0x60FFh
<b>Name</b>	target_velocity
<b>Objektcode</b>	VAR
<b>Datentyp</b>	INTEGER32
<b>Mode</b>	pv
<b>Zugriff</b>	rw
<b>PDO Mapping</b>	möglich
<b>Einheit</b>	Inkmente
<b>Wertebereich</b>	$(-2^{31})..(2^{31}-1)$
<b>Defaultwert</b>	—
<b>EEPROM</b>	nein

### 4.15 Position Control Function (pc) (DS402)

#### 4.15.1 Allgemeine Informationen

In diesem Kapitel werden die Positionswerte beschrieben, die im Zusammenhang mit dem Lage-regler des Antriebs stehen. Sie finden Verwendung im Profile Position Mode.

#### 4.15.2 Objekte, die in diesem Kapitel definiert werden

Index	Objekt	Name	Type	Zugriff
6063h	VAR	position_actual_value*	INTEGER32	ro
6064h	VAR	position_actual_value	INTEGER32	ro

#### 4.15.3 Objekte, die in anderen Kapiteln definiert werden

Index	Objekt	Name	Type	Kapitel
607Ah	VAR	target_position	INTEGER32	pp (⇒ Kap. 4.17)
607Bh	VAR	position_range_limit	INTEGER32	pp (⇒ Kap. 4.17)
607Ch	VAR	home-offset	INTEGER32	hm (⇒ Kap. 4.16)
6093h	VAR	position_factor	UNSIGNED32	fg (⇒ Kap. 4.5)
6094h	ARRAY	velocity_encoder_factor	UNSIGNED32	fg (⇒ Kap. 4.5)
6096h	ARRAY	acceleration_factor	UNSIGNED32	fg (⇒ Kap. 4.5)
6040h	VAR	controlword	INTEGER16	dc (⇒ Kap. 4.4)
6041h	VAR	statusword	UNSIGNED16	dc (⇒ Kap. 4.4)

#### 4.15.4 Objektbeschreibung

##### 4.15.4.1 Objekt 6063h: position\_actual\_value\* (DS402)

Das Objekt position\_actual\_value liefert die aktuelle Istposition in Inkremente. Die Auflösung pro Umdrehung kann 16 Bit oder 20 Bit betragen (s. Kommando PRBASE).

<b>Index</b>	0x6063h
<b>Name</b>	position_actual_value
<b>Objektcode</b>	VAR
<b>Datentyp</b>	INTEGER32
<b>Mode</b>	pc, pp
<b>Zugriff</b>	rw
<b>PDO Mapping</b>	möglich
<b>Einheit</b>	Inkremente (1 Umdr. = 16Bit / 20Bit)(s. PRBASE)f
<b>Wertebereich</b>	$(-2^{31})..(2^{31}-1)$
<b>Defaultwert</b>	—
<b>EEPROM</b>	nein

##### 4.15.4.2 Objekt 6064h: position\_actual\_value (DS402)

Das Objekt position\_actual\_value liefert die aktuelle Istposition. Die Auflösung (herstellerspez. Einheiten s. Objekte 2020h 08h/09h oder nach Antriebsprofil DSP402 s. Objekt 607Ah) kann mit den Getriebefaktoren des Lageregler geändert werden.

**Hinweis: Dieses Objekt sollte nicht in ein synchrones TPDO definiert (mapped) werden.**

<b>Index</b>	0x6064h
<b>Name</b>	position_actual_value
<b>Objektcode</b>	VAR
<b>Datentyp</b>	INTEGER32
<b>Mode</b>	pc, pp
<b>Zugriff</b>	rw
<b>PDO Mapping</b>	möglich
<b>Einheit</b>	position units
<b>Wertebereich</b>	$(-2^{31})..(2^{31}-1)$
<b>Defaultwert</b>	—
<b>EEPROM</b>	nein

## 4.16 Homing Mode (hm) (DS402)

### 4.16.1 Allgemeine Informationen

Dieses Kapitel beschreibt die verschiedenen Parameter, die zur Definition einer Referenzierung benötigt werden.

### 4.16.2 Objekte, die in diesem Kapitel definiert werden

Index	Objekt	Name	Type	Zugriff
607Ch	VAR	home_offset	INTEGER32	rw
6098h	VAR	homing_method	INTEGER8	rw
6099h	ARRAY	homing_speeds	UNSIGNED32	rw
609Ah	VAR	homing_acceleration	UNSIGNED32	rw

### 4.16.3 Objekte, die in anderen Kapiteln definiert werden

Index	Objekt	Name	Type	Kapitel
6040h	VAR	controlword	INTEGER16	dc (⇒ Kap. 4.4)
6041h	VAR	statusword	UNSIGNED16	dc (⇒ Kap. 4.4)

### 4.16.4 Objektbeschreibung

#### 4.16.4.1 Objekt 607Ch: home\_offset (DS402)

Der Referenzoffset ist die Differenz zwischen der Nullposition der Anwendung und des Maschinen-nullpunktes. Alle nachfolgenden absoluten Fahraufträge berücksichtigen den Referenzoffset.

<b>Index</b>	0x607Ch
<b>Name</b>	home_offset
<b>Objektcode</b>	VAR
<b>Datentyp</b>	INTEGER32
<b>Mode</b>	hm
<b>Zugriff</b>	rw
<b>PDO Mapping</b>	möglich
<b>Einheit</b>	benutzerdefiniert
<b>Wertebereich</b>	$(-2^{31})..(2^{31}-1)$
<b>Defaultwert</b>	0
<b>EEPROM</b>	ja

## 4.16.4.2 Objekt 6098h: homing\_method (DS402)

<b>Index</b>	0x6098h
<b>Name</b>	homing_method
<b>Objektcode</b>	VAR
<b>Datentyp</b>	INTEGER8
<b>Mode</b>	hm
<b>Zugriff</b>	rw
<b>PDO Mapping</b>	möglich
<b>Einheit</b>	position units
<b>Wertebereich</b>	-128..127
<b>Defaultwert</b>	0
<b>EEPROM</b>	ja

Die folgenden Referenzfahrtarten werden unterstützt:

Methodenach DSP402	Kurzbeschreibung Referenz	ASCII-Kommando
-128..-4	reserviert	—
-3	Fahren auf mechanischen Anschlag, mit Nullpunktsuche	NREF = 7
-2	Setzen des Referenzpunktes auf die aktuelle Position unter Berücksichtigung des Schleppabstandes	NREF = 6
-1	Referenzfahrt innerhalb einer Umdrehung (Drehrichtung entfernungsabhängig)	NREF = 5, DREF= 2
0	reserviert	—
1	Referenzfahrt auf negativen Endschalter, mit Nullpunktsuche, Fahrtrichtung negativ	NREF = 2, DREF= 0
2	Referenzfahrt auf positiven Endschalter, mit Nullpunktsuche, Fahrtrichtung positiv	NREF = 2, DREF= 1
3..7	werden nicht unterstützt	—
8	Referenzfahrt mit Referenzschalter, mit Nullpunktsuche, Fahrtrichtung positiv	NREF = 1, DREF= 1
9..11	werden nicht unterstützt	—
12	Referenzfahrt mit Referenzschalter, mit Nullpunktsuche, Fahrtrichtung negativ	NREF = 1, DREF= 0
13..14	werden nicht unterstützt	—
15..16	reserviert	—
17	Referenzfahrt auf negativen Endschalter, ohne Nullpunktsuche, Fahrtrichtung negativ	NREF = 4, DREF= 0
18	Referenzfahrt auf negativen Endschalter, ohne Nullpunktsuche, Fahrtrichtung positiv	NREF = 4, DREF= 1
19..23	werden nicht unterstützt	—
24	Referenzfahrt auf Referenzschalter, ohne Nullpunktsuche, Fahrtrichtung positiv	NREF = 3, DREF= 1
25..27	werden nicht unterstützt	—
28	Referenzfahrt auf Referenzschalter, ohne Nullpunktsuche, Fahrtrichtung negativ	NREF = 3, DREF= 0
29..30	werden nicht unterstützt	—
31..32	reserviert	—
33	Referenzfahrt innerhalb einer Umdrehung, Fahrtrichtung negativ	NREF = 5, DREF= 0
34	Referenzfahrt innerhalb einer Umdrehung, Fahrtrichtung positiv	NREF = 5, DREF= 1
35	Setzen des Referenzpunktes an die aktuelle Position	NREF = 0
36..127	reserviert	—

#### 4.16.4.2.1 Beschreibung der Referenziermethoden

Durch die Auswahl einer Referenzfahrtart durch Beschreiben des Parameters homing\_method (Objekt 6098<sub>h</sub>) wird folgendes bestimmt:

- das Referenzsignal (PStop, NStop, Referenzschalter)
- die Richtung der Referenzfahrt  
und, wenn passend
- die Position des Nullimpulses

Die Referenzposition wird durch den Referenzoffset (Objekt 607C<sub>h</sub>) festgelegt. Zur Anpassung der Grundeinstellung der Motorlage bei Referenzierung auf den Nullimpuls kann der herstellerspezifische Parameter ENCZERO (Objekt 3537<sub>h</sub>, Subindex 01<sub>h</sub>) verwendet werden.

Eine ausführliche Beschreibung der Referenzfahrtarten finden Sie in der Beschreibung der Inbetriebnahmesoftware DRIVE.EXE.

#### 4.16.4.3 Objekt 6099h: homing\_speeds (DS402)

<b>Index</b>	0x6099 <sub>h</sub>
<b>Name</b>	homing_speeds
<b>Objektcode</b>	ARRAY
<b>Anzahl Elemente</b>	2
<b>Datentyp</b>	UNSIGNED32

<b>Subindex</b>	01 <sub>h</sub>
<b>Kurzbeschreibung</b>	speed_during_search_for_switch
<b>Mode</b>	hm
<b>Zugriff</b>	rw
<b>PDO Mapping</b>	möglich
<b>Einheit</b>	velocity units
<b>Wertebereich</b>	0..(2 <sup>32</sup> -1)
<b>Defaultwert</b>	2 <sup>20</sup>
<b>EEPROM</b>	ja

#### 4.16.4.4 Objekt 609Ah: homing\_acceleration (DS402)

<b>Index</b>	0x609A <sub>h</sub>
<b>Name</b>	homing_acceleration
<b>Objektcode</b>	VAR
<b>Datentyp</b>	UNSIGNED32
<b>Mode</b>	hm
<b>Zugriff</b>	rw
<b>PDO Mapping</b>	möglich
<b>Einheit</b>	acceleration units
<b>Wertebereich</b>	0..(2 <sup>32</sup> -1)
<b>Defaultwert</b>	0
<b>EEPROM</b>	ja

### 4.16.5 Homing Mode Sequence

Mit Setzen des Bit 4 (positive Flanke) wird die Referenzfahrt gestartet. Der erfolgreiche Abschluss wird mit Bit 12 im Zustandswort angezeigt (s. Objekt 6041<sub>h</sub>). Bit 13 zeigt einen Fehler an, der sich während der Referenzfahrt ereignet hat. Hier ist der Fehlercode auszuwerten:  
 Error register (Objekte 1001<sub>h</sub>, 1003<sub>h</sub>), manufacturer status (Objekt1002<sub>h</sub>)

Bit 4	Bedeutung
0	Referenzfahrt inaktiv
0 ⇒ 1	Referenzfahrt starten
1	Referenzfahrt aktiv
0 ⇒ 1	Unterbrechung der Referenzfahrt

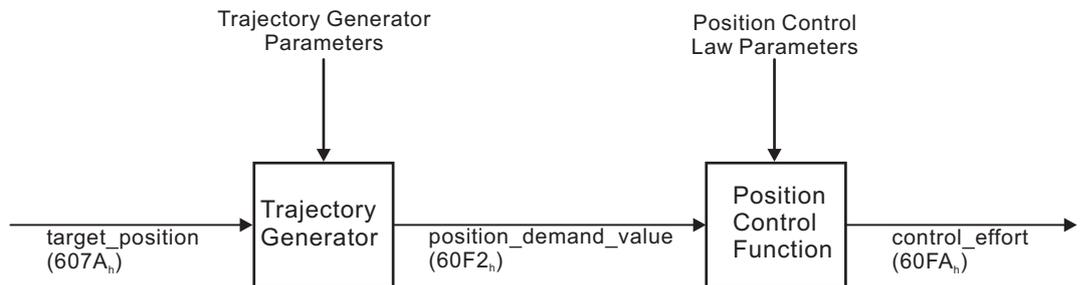
Bit 13	Bit 12	Bedeutung
0	0	Referenzpunkt nicht gesetzt bzw. Referenzfahrt noch nicht abgeschlossen
0	1	Referenzpunkt gesetzt bzw. Referenzfahrt erfolgreich abgeschlossen
1	0	Referenzfahrt konnte nicht erfolgreich abgeschlossen werden. (Schleppfehler)
1	1	kein erlaubter Zustand

## 4.17 Profile Position Mode (pp)

### 4.17.1 Allgemeine Informationen

Die Struktur dieser Betriebsart ist hier dargestellt:

Die spezielle Handshake-Verarbeitung von Controlword und Statusword wird in Kap.4.17.4.1 beschrieben.



### 4.17.2 Objekte, die in diesem Kapitel definiert werden

Index	Objekt	Name	Type	Zugriff
607A <sub>h</sub>	VAR	target_position	INTEGER32	rw
607B <sub>h</sub>	ARRAY	position_range_limit	INTEGER32	rw
6081 <sub>h</sub>	VAR	profile_velocity	UNSIGNED32	rw
6083 <sub>h</sub>	VAR	profile_acceleration	UNSIGNED32	rw
6084 <sub>h</sub>	VAR	profile_deceleration	UNSIGNED32	rw
6085 <sub>h</sub>	VAR	quick_stop_deceleration	UNSIGNED32	rw
6086 <sub>h</sub>	VAR	motion_profile_type	INTEGER16	rw

### 4.17.3 Objekte, die in anderen Kapiteln definiert werden

Index	Objekt	Name	Type	Kapitel
6040 <sub>h</sub>	VAR	controlword	INTEGER16	dc (⇒ Kap. 4.4)
6041 <sub>h</sub>	VAR	statusword	UNSIGNED16	dc (⇒ Kap. 4.4)
605A <sub>h</sub>	VAR	quick_stop_option_code	INTEGER16	dc (⇒ Kap. 4.4)
6093 <sub>h</sub>	ARRAY	position_factor	UNSIGNED32	fg (⇒ Kap. 4.5)
6094 <sub>h</sub>	ARRAY	velocity_encoder_factor	UNSIGNED32	fg (⇒ Kap. 4.5)
6097 <sub>h</sub>	ARRAY	acceleration_factor	UNSIGNED32	fg (⇒ Kap. 4.5)

### 4.17.4 Objektbeschreibung

#### 4.17.4.1 Objekt 607Ah: target\_position (DS402)

Das Objekt target\_position definiert die Zielposition des Antriebes. Abhängig vom Bit 6 im controlword wird die Zielposition als relativer Weg oder als absolute Position interpretiert. Dabei kann die Art der Relativfahrt durch den herstellerspezifischen Parameter 2022<sub>h</sub> Subindex 03<sub>h</sub> weiter aufgeschlüsselt werden.

Die mechanische Auflösung wird über die Getriebefaktoren Objekt 6093<sub>h</sub> Subindex 01<sub>h</sub> und 02<sub>h</sub> eingestellt.

<b>Index</b>	0x607A <sub>h</sub>
<b>Name</b>	target_position
<b>Objektcode</b>	VAR
<b>Datentyp</b>	INTEGER32
<b>Mode</b>	pp
<b>Zugriff</b>	rw
<b>PDO Mapping</b>	möglich
<b>Einheit</b>	benutzerdefiniert
<b>Wertebereich</b>	$-(2^{31}-1)..(2^{31}-1)$
<b>Defaultwert</b>	—
<b>EEPROM</b>	nein

#### 4.17.4.2 Objekt 607Bh: position\_range\_limit (DS402)

Mit dem Objekt "position\_range\_limit" wird der Anfang und das Ende des Verfahrbereiches für eine Modulo - Achse festgelegt. Der Anfang des Bereiches wird mit Subindex 01h "min\_position\_range\_limit" (ASCII SRND) und das Ende mit Subindex 02h "max\_position\_range\_limit" (ASCII ERND) definiert. Diese Funktionalität kann erst nach einer Neukonfiguration des Geräts verwendet werden. Dazu muss das Objekt 2020h Subindex 01h mit dem Wert 2 versehen werden und danach die Konfigurationsprozedur gestartet werden (⇒ Kap. 6.4).

<b>Index</b>	0x607Bh
<b>Name</b>	position_range_limit
<b>Objektcode</b>	ARRAY
<b>Anzahl Elemente</b>	3
<b>Datentyp</b>	INTEGER32

Beschreibung der Subindizes:

<b>Subindex</b>	01h
<b>Kurzbeschreibung</b>	min_position_range_limit
<b>Mode</b>	pp, pc
<b>Zugriff</b>	rw
<b>PDO Mapping</b>	nicht möglich
<b>Einheit</b>	position units
<b>Wertebereich</b>	$(-2^{31})..(2^{31}-1)$
<b>EEPROM</b>	ja
<b>Defaultwert</b>	$-2^{31}$

<b>Subindex</b>	02h
<b>Kurzbeschreibung</b>	max_position_range_limit
<b>Mode</b>	pp, pc
<b>Zugriff</b>	rw
<b>PDO Mapping</b>	nicht möglich
<b>Einheit</b>	position units
<b>Wertebereich</b>	$(-2^{31})..(2^{31}-1)$
<b>EEPROM</b>	ja
<b>Defaultwert</b>	$2^{31}-1$

#### 4.17.4.3 Objekt 6081h: profile\_velocity (DS402)

Die profile\_velocity ist die Endgeschwindigkeit, die nach der Beschleunigungsphase eines Fahrauftrages erreicht werden soll. Die verwendete Skalierung ist abhängig von dem eingestellten "velocity\_encoder\_factor" (Objekt 6094h).

Der Sollwert wird in Abhängigkeit des eingestellten Operationsmode (pp, pv) benutzt.

<b>Index</b>	0x6081h
<b>Name</b>	profile_velocity
<b>Objektcode</b>	VAR
<b>Datentyp</b>	UNSIGNED32
<b>Mode</b>	pp, pv
<b>Zugriff</b>	rw
<b>PDO Mapping</b>	möglich
<b>Einheit</b>	speed units
<b>Wertebereich</b>	$0..(2^{32}-1)$
<b>Defaultwert</b>	10
<b>EEPROM</b>	nein

#### 4.17.4.4 Objekt 6083h: profile\_acceleration (DS402)

Die Beschleunigungsrampe (profile\_acceleration) wird in Einheiten, die der Anwender definiert hat, vorgegeben. Die Verarbeitung bzw. Interpretation des Beschleunigungswertes kann auf zwei Arten erfolgen:

- **Interne Getriebefaktoren PGEARI = PGEARO (s. Objekte 2020<sub>h</sub> 08<sub>h</sub>/09<sub>h</sub>, s. ASCII - Kommando PGEARI/PGEARO)**  
Die Beschleunigungsrampe wird als Beschleunigungszeit [ms] oder Beschleunigungssteigung [Inkr./s<sup>2</sup>] bezogen auf die Zielgeschwindigkeit (Objekt 6081<sub>h</sub> "profile\_velocity") interpretiert. Die Skalierung des Wertes ist abhängig vom Beschleunigungsfaktor (Objekt 6097<sub>h</sub> "acceleration\_factor").  
**Hinweis:** Z. Zt. wird nur die Einheit Beschleunigungszeit unterstützt!
- **Interne Getriebefaktoren PGEARI <> PGEARO (s. Objekte 2020<sub>h</sub> 08<sub>h</sub>/09<sub>h</sub>, s. ASCII - Kommando PGEARI/PGEARO)**  
Die Beschleunigungsrampe wird als Beschleunigungszeit [ms] oder Beschleunigungssteigung [Längeneinheit/s<sup>2</sup>] bezogen auf die Zielgeschwindigkeit interpretiert. Die Skalierung des Wertes ist abhängig von den eingestellten Getriebefaktoren (s. Beschreibung der ASCII - Kommandos PGEARI und PGEARO) und der eingestellten Basiseinheiten [ms] oder [Längeneinheit/s<sup>2</sup>]. Die Vorwahl der Basiseinheit erfolgt über Bit 12 im controlword des Fahrauftrages (Objekt 2022<sub>h</sub>03<sub>h</sub>, ASCII - Kommando O\_C).

Die Art der Beschleunigungsrampe kann als lineare Rampe oder als Sin<sup>2</sup> Rampe ausgewählt werden (s. Objekt 6086<sub>h</sub>).

Index	0x6083 <sub>h</sub>
Name	profile_acceleration
Objektcode	VAR
Datentyp	UNSIGNED32
Mode	pp
Zugriff	rw
PDO Mapping	möglich
Einheit	acceleration units
Wertebereich	0..(2 <sup>15</sup> -1)
Defaultwert	0

#### 4.17.4.5 Objekt 6084h: profile\_deceleration (DS402)

Die Bremsrampe wird analog zur Beschleunigungsrampe behandelt. (s. Objekt 6083<sub>h</sub>)

Index	0x6084 <sub>h</sub>
Name	profile_deceleration
Objektcode	VAR
Datentyp	UNSIGNED32
Mode	pp
Zugriff	rw
PDO Mapping	möglich
Einheit	acceleration units
Wertebereich	0..(2 <sup>15</sup> -1)
Defaultwert	0

4.17.4.6 **Objekt 6086h: motion\_profile\_type (DS402)**

Die Art der Beschleunigungsrampe kann mit diesem Objekt als lineare Rampe oder als Sin<sup>2</sup> Rampe ausgewählt werden.

<b>Index</b>	0x6086h
<b>Name</b>	motion_profile_type
<b>Objektcode</b>	VAR
<b>Datentyp</b>	INTEGER16
<b>Mode</b>	pp
<b>Zugriff</b>	rw
<b>PDO Mapping</b>	möglich
<b>Einheit</b>	keine
<b>Wertebereich</b>	$(-2^{15})..(2^{15}-1)$
<b>Defaultwert</b>	—
<b>EEPROM</b>	ja

profile code	profile type
-32768..-1	herstellerspez. (wird nicht unterstützt)
0	linear (trapez)
1	sin <sup>2</sup>
2..32767	profilspez. Erweiterungen (wird nicht unterstützt)

4.17.5 **Funktionsbeschreibung**

In diesem Profil werden zwei Wege der Positionssollwertübergabe an den Antrieb unterstützt.

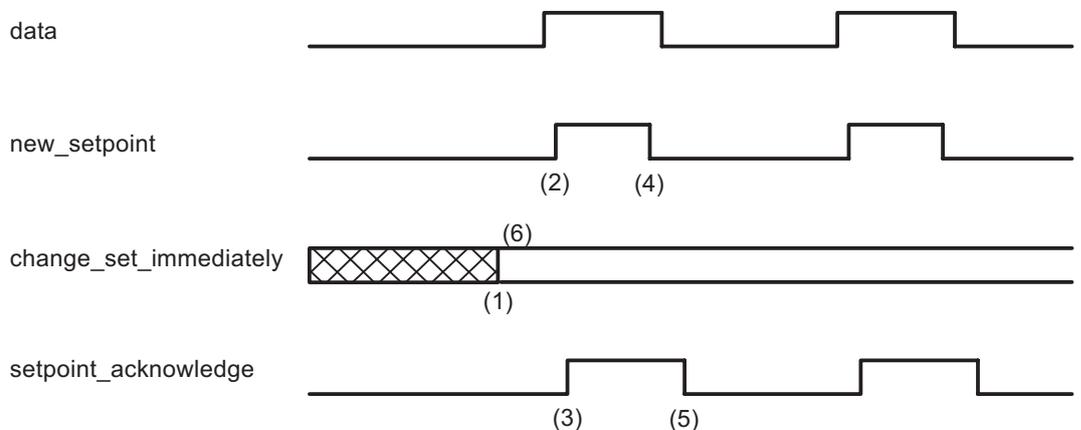
Eine Folge von Sollwerten:

Nach Erreichen der Zielposition berechnet der Antrieb sofort die Bewegung zur vorher übergebenen neuen Zielposition. Dies führt zu einer kontinuierlichen Bewegung, ohne dass der Antrieb zwischendurch auf Geschwindigkeit 0 abbremst. Beim SERVOSTAR 400/600 ist dies nur bei Verwendung von Trapezrampen möglich.

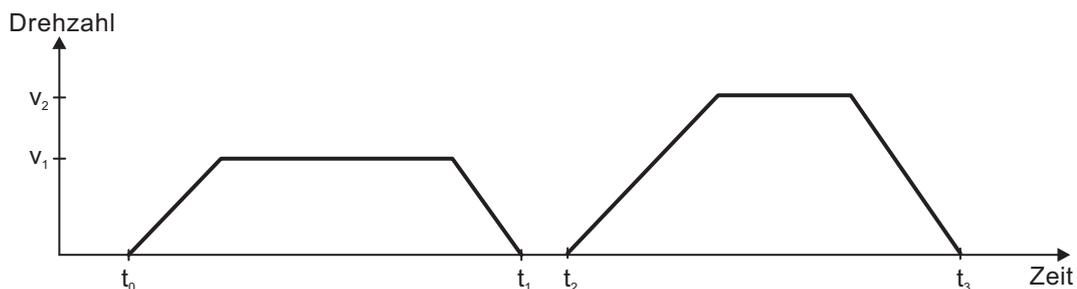
Einzelne Sollwerte:

Nach Erreichen der Zielposition signalisiert der Antrieb an den Master, dass er das Ziel erreicht hat und erhält dann einen neuen Sollwert. Nach Erreichen der Zielposition ist die Geschwindigkeit normalerweise 0 bevor die Achse zu einer neuen Zielposition fährt.

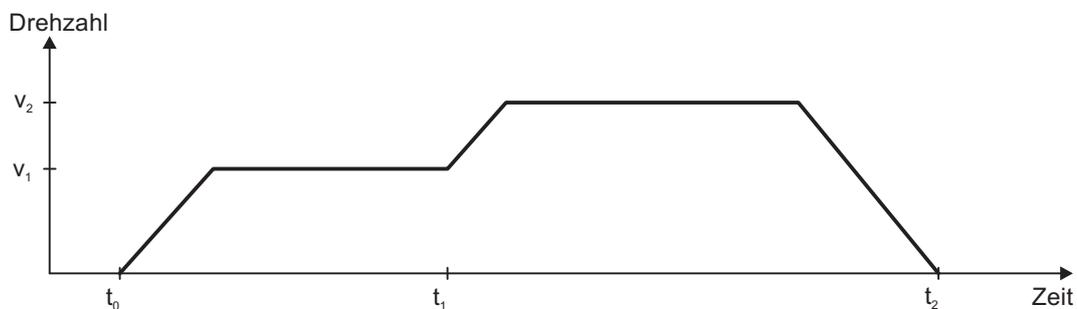
Die zwei Arten werden über das Timing der Bits new\_setpoint, change\_set\_immediately des Steuerworts und das Bit setpoint\_acknowledge des Zustandswort gesteuert. Diese Bits ermöglichen einen Handshake - Mechanismus, der es ermöglicht einen neuen Sollwert vorzubereiten während ein alter Fahrauftrag ausgeführt wird. Das verringert Reaktionszeiten innerhalb eines Steuerungsprogramms in einer Steuerung.



Die Bilder zeigen den Unterschied zwischen einer "Folge von Sollwerten" und einzelnen Sollwerten. Der Startwert des Bits `change_set_immediatly` im Steuerwort entscheidet über die verwendete Art. Die betrachteten Beispiele gelten nur für Trapezrampen. Wenn das Bit `change_set_immediatly` auf 0 ist (durchgehender Strich in Bild 1) wird ein einzelner Sollwert vom Antrieb erwartet (1). Nachdem Daten an den Antrieb übergeben wurden, signalisiert der Master durch den Wechsel am Bit `new_setpoint` im Steuerwort auf "1", dass die Daten gültig sind (2). Der Antrieb antwortet mit dem `setpoint_acknowledge` Bit = 1 im Zustandswort nachdem er den Wert erkannt und gespeichert hat (3). Nun kann der Master das Bit `new_setpoint` auf 0 setzen (4) woraufhin der Antrieb durch Rücksetzen des Bits `setpoint_acknowledge` signalisiert, dass er wieder neue Sollwerte entgegennehmen kann (5). In Bild 2 führt dieser Mechanismus zu einer Geschwindigkeit von 0 nachdem eine Rampe gefahren wurde, um eine Zielposition  $X_1$  zur Zeit  $t_1$  zu erreichen. Nach dem Signal an den Master, dass das Ziel erreicht wurde, wird die neue Zielposition zum Zeitpunkt  $t_2$  verarbeitet und zum Zeitpunkt  $t_3$  erreicht.



Mit dem Bit `change_set_immediatly` = 1 (symbolisiert durch die gestrichelte Linie in Bild 1) weist der Master den Antrieb an direkt nach dem Erreichen des letzten Sollwerts einen neuen Sollwert zu verarbeiten. Das Timing der Signale bleibt das gleiche. Diese Vorgehensweise bewirkt beim Antrieb, dass er schon den nächsten Sollwert  $X_2$  verarbeitet und Geschwindigkeit behält, wenn er die Zielposition  $X_1$  zum Zeitpunkt  $t_1$  erreicht. Danach fährt der Antrieb zur schon berechneten Zielposition  $X_2$ .



#### Bits im controlword:

- Bit 4 `new_set_point` (positive Flanke!)
- Bit 5 `change_set_immediatly`
- Bit 6 `absolut / relativ`

#### Bits im statusword:

- Bit 12 `setpoint acknowledge`
- Bit 13 `following error`

**Hinweis zur Fahrauftragsart "relativ":**

Wird das Bit 6 gesetzt, ist die Fahrauftragsart "relativ in Abhängigkeit zur letzten Zielposition oder Istposition" aktiviert. Sollten andere Relativarten gewünscht sein, sind diese im Vorfeld mit dem Objekt 2022<sub>h</sub> Subindex 03<sub>h</sub> "Positionierdaten für den Mode Lage" zu aktivieren (s. auch Objekt 2022<sub>h</sub> Subindex 03<sub>h</sub> oder ASCII Objekt O\_C).

**Hinweis zum Mode Profile Position Mode:**

Funktionale Beschreibung für den Mode: Profile Position Mode

Das Antriebsprofil DSP402 unterscheidet zwei Methoden Zielpositionen anzufahren. Diese beiden Methoden werden über die Bits "new\_setpoint" und "change\_set\_immediatly" im controlword und "setpoint\_acknowledge" im statusword gesteuert. Mit Hilfe dieser Bits kann ein Fahrauftrag aufgebaut werden, während ein anderer gerade ausgeführt wird (Handshake).

**● Anfahren von mehreren Zielpositionen ohne Zwischenstopp**

Nachdem die Zielposition erreicht wurde, wird sofort die nächste Zielposition angefahren. Voraussetzung ist, dass dem Antrieb neue Sollwerte signalisiert werden. Dies wird mit einer positiven Flanke (Bit "new\_setpoint") realisiert. Das Bit "setpoint\_acknowledge" darf hierbei nicht gesetzt sein (siehe auch Handshake DSP402).

Die Geschwindigkeit wird, nach Erreichen des ersten Zielpunktes, nicht auf Null reduziert.

**● Anfahren von einer einzelnen Zielpositionen**

Der Antrieb fährt in die Zielposition, wobei die Geschwindigkeit auf Null reduziert wird. Das Erreichen der Zielposition wird mit dem Bit "target\_reached" im statusword signalisiert.

Diese Seite wurde bewusst leer gelassen.

## 5 Der Objektkanal

### 5.1 Objektbeschreibung

#### 5.1.1 Objekt >3500h: Herstellerspezifischer Objektkanal

Das Objektverzeichnis ist ab Index 3500<sub>h</sub> (reservierter Objektbereich 3500<sub>h</sub>– 3900<sub>h</sub>) um sämtliche Geräteobjekte erweitert worden, die mit bis zu 4 Byte Nutzdaten darstellbar sind. Dieser Bereich ist dynamisch erweiterbar d.h., neue Geräteparameter, die das o. g. Datenformat erfüllen, werden bei Erweiterungen in der "Kernfirmware" **automatisch** der Tabelle angehängt. Mit Hilfe des Objekt 3500<sub>h</sub> (Subindex 01<sub>h</sub>, read) kann die Gesamtanzahl der Objekte im Objektkanal bestimmt werden (⇒ Kap. 6.3).

Jedes Objekt in diesem Bereich wird mit Hilfe von 8 Subindizes beschrieben. Diese Struktur baut sich folgendermaßen auf:

<b>Index</b>	> 3500 <sub>h</sub>
<b>Name</b>	Objektabhängig
<b>Objektcode</b>	VAR
<b>Datentyp</b>	RECORD

Beschreibung der Subindizes:

<b>Subindex</b>	00 <sub>h</sub>
<b>Beschreibung</b>	Anzahl der Einträge
<b>Einheit</b>	—
<b>Zugriff</b>	—
<b>PDO Mapping</b>	nicht möglich
<b>Datentyp</b>	UNSIGNED8
<b>Wertebereich</b>	0...2 <sup>8</sup> -1
<b>EEPROM</b>	—
<b>Defaultwert</b>	—

<b>Subindex</b>	01 <sub>h</sub>
<b>Beschreibung</b>	Lesen / Schreiben eines Parameters
<b>Einheit</b>	s. jeweiliges ASCII-Kommando
<b>Zugriff</b>	s. jeweiliges ASCII-Kommando
<b>PDO Mapping</b>	nicht möglich
<b>Datentyp</b>	s. jeweiliges ASCII-Kommando
<b>Wertebereich</b>	s. jeweiliges ASCII-Kommando
<b>EEPROM</b>	s. Subindex 04 <sub>h</sub>
<b>Defaultwert</b>	s. jeweiliges ASCII-Kommando

<b>Subindex</b>	02 <sub>h</sub>
<b>Beschreibung</b>	Lesen des unteren Grenzwertes
<b>Einheit</b>	s. jeweiliges ASCII-Kommando
<b>Zugriff</b>	ro
<b>PDO Mapping</b>	nicht möglich
<b>Datentyp</b>	s. jeweiliges ASCII-Kommando
<b>Wertebereich</b>	s. jeweiliges ASCII-Kommando
<b>EEPROM</b>	—
<b>Defaultwert</b>	—

<b>Subindex</b>	03h
<b>Beschreibung</b>	Lesen des oberen Grenzwertes
<b>Einheit</b>	s. jeweiliges ASCII-Kommando
<b>Zugriff</b>	ro
<b>PDO Mapping</b>	nicht möglich
<b>Datentyp</b>	s. jeweiliges ASCII-Kommando
<b>Wertebereich</b>	s. jeweiliges ASCII-Kommando
<b>EEPROM</b>	—
<b>Defaultwert</b>	—

<b>Subindex</b>	04h
<b>Beschreibung</b>	Lesen des Defaultwertes
<b>Einheit</b>	s. jeweiliges ASCII-Kommando
<b>Zugriff</b>	ro
<b>PDO Mapping</b>	nicht möglich
<b>Datentyp</b>	s. jeweiliges ASCII-Kommando
<b>Wertebereich</b>	s. jeweiliges ASCII-Kommando
<b>EEPROM</b>	—
<b>Defaultwert</b>	—

<b>Subindex</b>	05h
<b>Beschreibung</b>	Lesen des Parameterformats
<b>Einheit</b>	—
<b>Zugriff</b>	ro
<b>PDO Mapping</b>	nicht möglich
<b>Datentyp</b>	s. jeweiliges ASCII-Kommando
<b>Wertebereich</b>	s. jeweiliges ASCII-Kommando
<b>EEPROM</b>	—
<b>Defaultwert</b>	—

Die folgenden Parameterformate sind möglich:

0	Funktion (keine Parameter)
1	Funktion (INTEGER32 Parameter)
2	Funktion (INTEGER32 Parameter mit Wichtung 3)
3	INTEGER8
4	unsigned8
5	INTEGER16
6,13	unsigned16
7	INTEGER32
8,12	unsigned32
9,10	INTEGER32 (Wichtung 3)



**Auf Parameter mit dem Parameterformat 0 darf nur lesend zugegriffen werden!**

<b>Subindex</b>	06h
<b>Beschreibung</b>	Lesen der Parameter-Kontrolldaten
<b>Einheit</b>	—
<b>Zugriff</b>	ro
<b>PDO Mapping</b>	nicht möglich
<b>Datentyp</b>	UNSIGNED32
<b>Wertebereich</b>	0..2 <sup>32</sup> -1
<b>EEPROM</b>	—
<b>Defaultwert</b>	—

0x00010000 Nach einer Änderung muss die Variable abgespeichert, und der Regler resettiert werden.

0x00020000 Variable wird im seriellen EEPROM abgespeichert.

0x00200000 Variable ist read-only, darf nicht über den Bus geschrieben werden.

<b>Subindex</b>	07h / 08h
<b>Beschreibung</b>	reserviert
<b>Einheit</b>	—
<b>Zugriff</b>	ro
<b>PDO Mapping</b>	nicht möglich
<b>Datentyp</b>	UNSIGNED32
<b>Wertebereich</b>	0..2 <sup>32</sup> -1
<b>EEPROM</b>	—
<b>Defaultwert</b>	—

Nummer	ASCII-Kommando	Datenformat	Wichtung	Status	EEPROM
3500	MAXSDO	INTEGER32	—	—	—
3501	ACC	INTEGER16	—	—	Ja
3502	ACCR	INTEGER16	—	—	Ja
3503	ACTFAULT	INTEGER8	—	Disabled + Reset	Ja
3504	ACTIVE	INTEGER8	—	—	Nein
3505	ADDR	UNSIGNED8	—	—	Ja
3506	AENA	INTEGER8	—	—	Ja
3507	ANCNFG	INTEGER8	—	Disabled + Reset	Ja
3508	ANDB	INTEGER32	<input checked="" type="checkbox"/>	—	Ja
3509	ANIN1	INTEGER32	—	—	Nein
350A	ANIN2	INTEGER32	—	—	Nein
350B	ANOFF1	INTEGER16	—	—	Ja
350C	ANOFF2	INTEGER16	—	—	Ja
350D	ANOUT1	INTEGER8	—	Disabled + Reset	Ja
350E	ANOUT2	INTEGER8	—	Disabled + Reset	Ja
350F	ANZERO1	—	—	—	—
3510	ANZERO2	—	—	—	—
3511	AVZ1	INTEGER32	<input checked="" type="checkbox"/>	—	Ja
3512	CALCHP	—	—	Enabled	—
3513	CALCRK	—	—	Enabled	—
3514	CALCRP	—	—	Disabled + Reset	—
3515	CBAUD	INTEGER16	—	Disabled + Reset	Ja
3516	reserviert				
3517	CDUMP	—	—	—	—
3518	CLRFAULT	—	—	—	—
3519	CLRHR	—	—	—	—
351A	CLRORDER	INTEGER16	—	Disabled	—
351B	CLRWARN	UNSIGNED8	—	Disabled + Reset	Ja
351C	CONFIG	—	—	—	—
351D	CONTINUE	—	—	Enabled	—
351E	CTUNE	—	—	Enabled	—
351F	CUPDATE	—	—	Disabled	—
3520	DAOFFSET1	INTEGER16	—	—	Ja
3521	DAOFFSET2	INTEGER16	—	—	Ja

Nummer	ASCII-Kommando	Datenformat	Wichtung	Status	EEPROM
3522	DEC	INTEGER16	—	—	Ja
3523	DECDIS	INTEGER16	—	—	Ja
3524	DECR	INTEGER16	—	—	Ja
3525	DECSTOP	INTEGER16	—	—	Ja
3526	DEVICE	—	—	—	Nein
3527	DICONT	INTEGER32	☒	—	Nein
3528	DIFVAR	—	—	—	—
3529	DIPEAK	INTEGER32	☒	—	Nein
352A	DIR	INTEGER8	—	Disabled + Reset	Ja
352B	DIS	—	—	Enabled	—
352C	DREF	INTEGER8	—	—	Ja
352D	DRVSTAT	INTEGER32	—	—	Nein
352E	DR_TYPE	INTEGER16	—	—	Nein
352F	DUMP	—	—	—	—
3530	EN	—	—	Disabled	—
3531	ENCCAPT	INTEGER8	—	Disabled	Ja
3532	ENCIN	INTEGER32	—	Disabled + Reset	Ja
3533	ENCLINES	INTEGER16	—	Disabled + Reset	Ja
3534	ENCMODE	INTEGER8	—	—	Ja
3535	ENCOUT	INTEGER16	—	—	Ja
3536	reserviert				
3537	ENCZERO	INTEGER16	—	—	Ja
3538	EXTMUL	INTEGER16	—	—	Ja
3539	EXTPOS	INTEGER8	—	Disabled + Reset	Ja
353A	EXTWD	INTEGER32	—	—	Ja
353B	FBTYPE	INTEGER8	—	Disabled + Reset	Ja
353C	FILTMODE	UNSIGNED8	—	Disabled + Reset	Ja
353D	FOLDMODE	INTEGER8	—	Disabled + Reset	Ja
353E	GEARI	INTEGER16	—	—	Ja
353F	GEARMODE	INTEGER8	—	Disabled + Reset	Ja
3540	GEARO	INTEGER16	—	—	Ja
3541	GET	—	—	—	—
3542	GP	INTEGER32	☒	—	Ja
3543	GPFBT	INTEGER32	☒	—	Ja
3544	GPFFT	INTEGER32	☒	—	Ja
3545	GPFFV	INTEGER32	☒	—	Ja
3546	GPTN	INTEGER32	☒	—	Ja
3547	GPV	INTEGER32	☒	—	Ja
3548	GV	INTEGER32	☒	—	Ja
3549	GVFBT	INTEGER32	☒	—	Ja
354A	GVFILT	INTEGER8	—	—	Ja
354B	GVFR	INTEGER32	☒	—	Ja
354C	GVT2	INTEGER32	☒	—	Ja
354D	GVTN	INTEGER32	☒	—	Ja
354E	HACOFFS	INTEGER16	—	—	Encoder
354F	HFACT1	INTEGER16	—	—	Encoder
3550	HASOFFS	INTEGER16	—	—	Encoder
3551	HDUMP	—	—	—	—
3552	HICOFFS	INTEGER16	—	—	Ja
3553	HIFACT1	INTEGER16	—	—	Encoder
3554	HISOFFS	INTEGER16	—	—	Encoder
3555	HRESET	—	—	—	—
3556	HSAVE	—	—	—	—
3557	HVER	—	—	—	Nein
3558	I	INTEGER32	☒	—	Nein
3559	I2T	INTEGER32	—	—	Nein
355A	I2TLIM	INTEGER8	—	—	Ja
355B	ICMD	INTEGER32	☒	—	Nein
355C	ICONT	INTEGER32	☒	—	Ja
355D	ID	INTEGER32	☒	—	Nein

Nummer	ASCII-Kommando	Datenformat	Wichtung	Status	EEPROM
355E	IDUMP	—	—	—	—
355F	IMAX	INTEGER32	<input checked="" type="checkbox"/>	—	Nein
3560	IN	—	—	—	—
3561	IN1	INTEGER8	—	—	—
3562	IN1MODE	INTEGER8	—	Disabled + Reset	Ja
3563	IN1TRIG	INTEGER32	—	—	Ja
3564	IN2	INTEGER8	—	—	Nein
3565	IN2MODE	INTEGER8	—	Disabled + Reset	Ja
3566	IN2TRIG	INTEGER32	—	—	Ja
3567	IN3	INTEGER8	—	—	Nein
3568	IN3MODE	INTEGER8	—	Disabled + Reset	Ja
3569	IN3TRIG	INTEGER32	—	—	Ja
356A	IN4	INTEGER8	—	—	Nein
356B	IN4MODE	INTEGER8	—	Disabled + Reset	Ja
356C	IN4TRIG	INTEGER32	—	—	Ja
356D	INPOS	—	—	—	—
356E	IPEAK	INTEGER32	<input checked="" type="checkbox"/>	—	Ja
356F	IPEAKN	INTEGER32	<input checked="" type="checkbox"/>	—	Ja
3570	IQ	INTEGER32	<input checked="" type="checkbox"/>	—	Nein
3571	ISCALE1	INTEGER32	<input checked="" type="checkbox"/>	—	Ja
3572	ISCALE2	INTEGER32	<input checked="" type="checkbox"/>	—	Ja
3573	K	—	—	Enabled	Nein
3574	KC	INTEGER32	<input checked="" type="checkbox"/>	—	Ja
3575	KEYLOCK	INTEGER8	—	—	Ja
3576	reserviert				
3577	L	INTEGER32	<input checked="" type="checkbox"/>	—	Ja
3578	LATCH16	INTEGER16	—	—	—
3579	LATCH16N	INTEGER16	—	—	Nein
357A	LATCH32	INTEGER32	—	—	—
357B	LATCH32N	INTEGER32	—	—	Nein
357C	LATCHX32	INTEGER32	—	—	Nein
357D	LATCHX32N	INTEGER32	—	—	Nein
357E	LED1	INTEGER8	—	—	—
357F	LED2	INTEGER8	—	—	—
3580	LED3	INTEGER8	—	—	—
3581	LEDSTAT	INTEGER16	—	—	—
3582	LIST	—	—	—	Nein
3583	LOAD	—	—	—	Nein
3584	MAXTEMPE	INTEGER16	—	—	Ja
3585	MAXTEMPH	INTEGER16	—	—	Ja
3586	MAXTEMPM	INTEGER32	<input checked="" type="checkbox"/>	—	Ja
3587	MBRAKE	INTEGER8	—	Disabled + Reset	Ja
3588	MDBCNT	—	—	—	—
3589	MDBGET	—	—	—	—
358A	MDBSET	INTEGER16	—	—	—
358B	MDUMP	—	—	—	—
358C	reserviert				
358D	MH	—	—	Enabled	—
358E	MICONT	INTEGER32	<input checked="" type="checkbox"/>	—	Ja
358F	MIPEAK	INTEGER32	<input checked="" type="checkbox"/>	—	Ja
3590	reserviert				
3591	MJOG	—	—	Enabled	—
3592	MVANGLP	INTEGER16	—	—	Ja
3593	MKT	INTEGER32	<input checked="" type="checkbox"/>	—	Ja
3594	reserviert				
3595	MLGC	INTEGER32	<input checked="" type="checkbox"/>	—	Ja
3596	MLGD	INTEGER32	<input checked="" type="checkbox"/>	—	Ja
3597	MLGP	INTEGER32	<input checked="" type="checkbox"/>	—	Ja
3598	MLGQ	INTEGER32	<input checked="" type="checkbox"/>	—	Ja
3599	MNUMBER	INTEGER16	—	Disabled	Ja

Nummer	ASCII-Kommando	Datenformat	Wichtung	Status	EEPROM
359A	MONITOR1	INTEGER16	—	—	Nein
359B	MONITOR2	INTEGER16	—	—	Nein
359C	MPHASE	INTEGER16	—	Disabled	Ja
359D	MPOLES	INTEGER8	—	Disabled	Ja
359E	MRD	—	—	Enabled	—
359F	reserviert				
35A0	MRESBW	INTEGER16	—	—	Ja
35A1	MRESPOLES	INTEGER8	—	Disabled	Ja
35A2	MSG	INTEGER8	—	—	Ja
35A3	MSPEED	INTEGER32	<input checked="" type="checkbox"/>	—	Ja
35A4	reserviert				
35A5	MTANGLP	INTEGER16	—	—	Ja
35A6	reserviert				
35A7	MVANGLB	INTEGER32	—	—	Ja
35A8	MVANGLF	INTEGER16	—	—	Ja
35A9	M_RESET	—	—	Disabled	—
35AA	NONBTB	INTEGER8	—	—	Ja
35AB	NOTCHBW	INTEGER32	<input checked="" type="checkbox"/>	—	Ja
35AC	NOTCHHZ	INTEGER32	<input checked="" type="checkbox"/>	—	Ja
35AD	NREF	INTEGER8	—	—	Ja
35AE	O1	INTEGER8	—	—	Nein
35AF	O1MODE	INTEGER8	—	Disabled + Reset	Ja
35B0	O1TRIG	INTEGER32	—	—	Ja
35B1	O2	INTEGER8	—	—	Nein
35B2	O2MODE	INTEGER8	—	Disabled + Reset	Ja
35B3	O2TRIG	INTEGER32	—	—	Ja
35B4	OPMODE	INTEGER8	—	—	Ja
35B5	OPTION	INTEGER16	—	—	Nein
35B6	OVERRIDE	INTEGER8	—	—	Ja
35B7	O_ACC1	INTEGER16	—	—	Nein
35B8	O_ACC2	INTEGER16	—	—	Nein
35B9	O_C	INTEGER16	—	—	Nein
35BA	O_DEC1	INTEGER16	—	—	Nein
35BB	O_DEC2	INTEGER16	—	—	Nein
35BC	O_FN	INTEGER16	—	—	Nein
35BD	O_FT	INTEGER16	—	—	Nein
35BE	O_P	INTEGER32	—	—	Nein
35BF	O_V	INTEGER32	—	—	Nein
35C0	PBAL	INTEGER32	—	—	Nein
35C1	PBALMAX	INTEGER32	—	—	Ja
35C2	PBALRES	INTEGER8	—	—	Ja
35C3	PBAUD	INTEGER32	<input checked="" type="checkbox"/>	—	Nein
35C4	PDUMP	—	—	—	—
35C5	PE	INTEGER32	—	—	Nein
35C6	PEINPOS	INTEGER32	—	—	Ja
35C7	PEMAX	INTEGER32	—	—	Ja
35C8	PFB	INTEGER32	—	—	Nein
35C9	PFB0	INTEGER32	—	—	Nein
35CA	PGEARI	INTEGER32	—	Disabled + Reset	Ja
35CB	PGEARO	INTEGER32	—	Disabled + Reset	Ja
35CC	PIOBUF	—	—	—	Nein
35CD	PMODE	INTEGER32	—	Disabled + Reset	Ja
35CE	PNOID	INTEGER32	—	—	Nein
35CF	POSCNFG	INTEGER8	—	Disabled + Reset	Ja
35D0	PPOTYP	INTEGER8	—	—	Ja
35D1	PRBASE	INTEGER8	—	Disabled + Reset	Ja
35D2	PRD	INTEGER32	—	—	Nein
35D3	PROMPT	INTEGER16	—	—	Nein
35D4	PSTATE	—	—	—	Nein
35D5	PTBASE	INTEGER8	—	—	Ja
35D6	PTMIN	INTEGER16	—	—	Ja

Nummer	ASCII-Kommando	Datenformat	Wichtung	Status	EEPROM
35D7	PV	INTEGER32	—	—	Nein
35D8	PVMAX	INTEGER32	—	—	Ja
35D9	PVMAXN	INTEGER32	—	—	Ja
35DA	reserviert				
35DB	reserviert				
35DC	READNIMP	—	—	—	—
35DD	READY	INTEGER8	—	—	Nein
35DE	RECDONE	INTEGER8	—	—	Nein
35DF	RECING	INTEGER8	—	—	Nein
35E0	RECOFF	—	—	—	—
35E1	RECRDY	INTEGER8	—	—	Nein
35E2	REFIP	INTEGER32	<input checked="" type="checkbox"/>	—	Ja
35E3	REFPOS	INTEGER32	—	—	Nein
35E4	REMOTE	INTEGER8	—	—	Nein
35E5	RESPHASE	INTEGER16	—	—	Ja
35E6	RK	INTEGER16	—	—	Ja
35E7	ROFFS	INTEGER32	—	—	Ja
35E8	RS232T	INTEGER16	—	—	Ja
35E9	RSTVAR	—	—	Disabled	Nein
35EA	S	—	—	—	—
35EB	SAVE	—	—	—	—
35EC	SBAUD	INTEGER8	—	—	Ja
35ED	SCAN	—	—	—	—
35EE	SDUMP	—	—	—	—
35EF	SERIALNO	INTEGER32	—	—	Nein
35F0	SETREF	—	—	—	—
35F1	SETROFFS	—	—	—	—
35F2	SLEN	INTEGER8	—	—	Ja
35F3	SLOTIO	INTEGER32	—	—	Nein
35F4	SPHAS	INTEGER8	—	—	Nein
35F5	SPSET	INTEGER16	—	—	Ja
35F6	SSIGRAY	INTEGER8	—	Disabled	Ja
35F7	SSIINV	INTEGER8	—	Disabled	Ja
35F8	SSIMODE	INTEGER8	—	—	Ja
35F9	SSIOUT	INTEGER8	—	Disabled	Ja
35FA	SSTAT	—	—	—	Nein
35FB	STAT	INTEGER16	—	—	Nein
35FC	STATIO	—	—	—	Nein
35FD	STATUS	—	—	—	Nein
35FE	STOP	—	—	Enabled	—
35FF	STOPMODE	INTEGER8	—	Disabled + Reset	Ja
3600	SWCNFG	UNSIGNED16	—	Disabled + Reset	Ja
3601	SWCNFG2	UNSIGNED16	—	Disabled + Reset	Ja
3602	SWE0	INTEGER32	—	—	Ja
3603	SWE0N	INTEGER32	—	—	Ja
3604	SWE1	INTEGER32	—	—	Ja
3605	SWE1N	INTEGER32	—	—	Ja
3606	SWE2	INTEGER32	—	—	Ja
3607	SWE2N	INTEGER32	—	—	Ja
3608	SWE3	INTEGER32	—	—	Ja
3609	SWE3N	INTEGER32	—	—	Ja
360A	SWE4	INTEGER32	—	—	Ja
360B	SWE4N	INTEGER32	—	—	Ja
360C	SWE5	INTEGER32	—	—	Ja
360D	SWE5N	INTEGER32	—	—	Ja
360E	T	INTEGER32	<input checked="" type="checkbox"/>	Enabled	—
360F	TASK	—	—	—	Nein
3610	TEMPE	INTEGER32	—	—	Nein
3611	TEMPH	INTEGER32	—	—	Nein
3612	TEMPM	INTEGER32	—	—	Nein
3613	TRJSTAT	INTEGER32	—	—	Nein

Nummer	ASCII-Kommando	Datenformat	Wichtung	Status	EEPROM
3614	TRUN	—	—	—	Ja
3615	reserviert				
3616	UID	INTEGER16	—	—	Ja
3617	UVLTMODE	INTEGER8	—	Disabled + Reset	Ja
3618	V	INTEGER32	—	—	Nein
3619	reserviert				
361A	VBUS	INTEGER32	—	—	Nein
361B	VBUSBAL	INTEGER16	—	—	Ja
361C	VBUSMAX	INTEGER32	—	—	Ja
361D	VBUSMIN	INTEGER16	—	—	Ja
361E	VCMD	INTEGER32	<input checked="" type="checkbox"/>	—	Nein
361F	VDUMP	—	—	—	—
3620	VELO	INTEGER32	<input checked="" type="checkbox"/>	—	Ja
3621	VJOG	INTEGER32	—	—	Ja
3622	VLIM	INTEGER32	<input checked="" type="checkbox"/>	—	Ja
3623	VLIMN	INTEGER32	<input checked="" type="checkbox"/>	—	Ja
3624	VMAX	INTEGER32	<input checked="" type="checkbox"/>	—	Nein
3625	VMIX	INTEGER32	<input checked="" type="checkbox"/>	—	Ja
3626	VMUL	INTEGER32	—	—	Ja
3627	VOSPD	INTEGER32	<input checked="" type="checkbox"/>	—	Ja
3628	VREF	INTEGER32	—	—	Ja
3629	VSCALE1	INTEGER16	—	—	Ja
362A	VSCALE2	INTEGER16	—	—	Ja
362B	\	UNSIGNED8	—	—	—
362C	DILIM	INTEGER8	—	Disabled + Reset	Ja
362D	DENA	INTEGER8	—	—	Ja
362E	IN2PM	INTEGER8	—	—	Ja
362F	KTN	INTEGER32	<input checked="" type="checkbox"/>	—	Ja
3630	INPT	INTEGER16	—	—	Ja
3631	UCOMP	INTEGER32	—	—	Ja
3632	COLDSTART	—	—	Disabled	—
3633	reserviert				
3634	UID1	INTEGER32	—	—	Ja
3635	SETVCT	INTEGER16	—	—	Nein
3636	WPOS	INTEGER8	—	Disabled + Reset	Nein
3637	SRND	INTEGER32	—	—	Ja
3638	ERND	INTEGER32	—	—	Ja
3639	MDRV	INTEGER8	—	—	Ja
363A	BCC	INTEGER16	—	—	Nein
363B	FPGA	INTEGER8	—	Disabled + Reset	Ja
363C	REFMODE	INTEGER8	—	—	Ja
363D	VLO	INTEGER32	<input checked="" type="checkbox"/>	—	Ja
363E	WMASK	INTEGER32	—	—	Nein
363F	WPOSE	INTEGER32	—	—	Nein
3640	WPOSP	INTEGER32	—	—	Nein
3641	WPOSX	INTEGER32	—	—	Nein
3642	MOVE	INTEGER16	—	Enabled	—
3643	POSRSTAT	INTEGER32	—	—	Nein
3644	P1	INTEGER32	—	—	Ja
3645	P2	INTEGER32	—	—	Ja
3646	P3	INTEGER32	—	—	Ja
3647	P4	INTEGER32	—	—	Ja
3648	P5	INTEGER32	—	—	Ja
3649	P6	INTEGER32	—	—	Ja
364A	P7	INTEGER32	—	—	Ja
364B	P8	INTEGER32	—	—	Ja
364C	P9	INTEGER32	—	—	Ja
364D	P10	INTEGER32	—	—	Ja
364E	P11	INTEGER32	—	—	Ja
364F	P12	INTEGER32	—	—	Ja
3650	P13	INTEGER32	—	—	Ja

Nummer	ASCII-Kommando	Datenformat	Wichtung	Status	EEPROM
3651	P14	INTEGER32	—	—	Ja
3652	P15	INTEGER32	—	—	Ja
3653	P16	INTEGER32	—	—	Ja
3654	PTARGET	INTEGER32	—	—	Ja
3655	ACTRS232	INTEGER8	—	—	Nein
3656	ROFFS2	INTEGER32	—	—	Ja
3657	FW	INTEGER32	<input checked="" type="checkbox"/>	—	Nein
3658	reserviert				
3659	ACCUNIT	INTEGER32	—	—	Ja
365A	VCOMM	INTEGER32	—	—	Ja
365B	MTMUX	INTEGER16	—	—	Nein
365C	ROFFS0	INTEGER32	—	—	Ja
365D	REFLS	INTEGER32	—	—	Ja
365E	BOOT	—	—	—	Ja
365F	VUNIT	INTEGER32	—	—	Ja
3660	PUNIT	INTEGER32	—	—	Ja
3661	reserviert				
3662	reserviert				
3663	reserviert				
3664	reserviert				
3665	reserviert				
3666	reserviert				
3667	reserviert				
3668	reserviert				
3669	reserviert				
366A	reserviert				
366B	reserviert				
366C	reserviert				
366D	reserviert				
366E	TBRAKE	INTEGER16	—	—	Ja
366F	TBRAKE0	INTEGER16	—	—	Ja
3670	CMDDLY	INTEGER16	—	—	Ja
3671	reserviert				
3672	DRVCNFG	INTEGER32	—	—	Ja

Diese Seite wurde bewusst leer gelassen.

## 6 Anhang

### 6.1 Beispiele für die Inbetriebnahme

Alle Angaben sind hexadezimal. Die achsbezogenen Angaben beziehen sich immer auf die Station1.

#### 6.1.1 Prinzipielle Prüfung der Verbindung Steuerung <-> SERVOSTAR

Beim Einschalten des SERVOSTAR 400/600 wird, abhängig von der Einstellung des Bit 2 des Parameters DRVCNFG, eine Emergency - Message mit 0 oder 8 Datenbytes (mit Inhalt 0) auf den Bus gesendet. Wenn sich im Bussystem kein geeigneter Empfänger findet, wird dieses Telegramm fortlaufend weiter gesendet.

Kann ein CAN - Master diese Nachricht nicht erkennen, können folgende Maßnahmen zur Überprüfung der Kommunikation durchgeführt werden:

- Überprüfung des Buskabels: richtiger Wellenwiderstand, korrekte Abschlusswiderstände an beiden Enden?
- Multimeterüberprüfung des Ruhepegels der Busleitungen CAN-H und CAN-L gegenüber der CAN-GND - Leitung (ca. 2.5 V).
- Oszilloskop - Überprüfung der Ausgangssignale an CAN-H und CAN-L am SERVOSTAR 400/600. Werden Signale auf den Bus gesendet? Spannungsdifferenz zwischen CAN-H und CAN-L bei logischer "0" ca. 2-3 V.
- Werden die Signale bei angeschlossenem Master nicht mehr weitergesendet? Eventuell Masterhardware überprüfen.
- Überprüfung der Master-Software!

## 6.1.2 Beispiel für die Bedienung der Zustandsmaschine

Nach dem Einschalten des SERVOSTAR 400/600 und dem Erkennen der Boot-Up-Message kann die Kommunikation über SDOs aufgenommen werden, z.B. können so Parameter abgefragt oder geschrieben werden oder die Zustandsmaschine des Antriebs gesteuert werden.

In den weiteren Beispielen wird angenommen, dass es Funktionen zum Lesen und Schreiben von SDOs gibt, die wie folgt aussehen:

SDO-Read (UINT Index, USHORT Subindex);

SDO-Write (UINT Index, USHORT Subindex, ULONG Wert);

Der Zustand der Zustandsmaschine kann über folgende Abfrage erkannt werden:

SDO-Read (6041<sub>h</sub>, 00<sub>h</sub>)

Direkt nach dem Einschalten erhält man dann z.B. als Antwortwert ein 0040<sub>h</sub>. Dies entspricht dem Zustand „Switch on disabled“ (⇒ Kap. 4.4.1.1).

Auf dem CAN-Bus würde man als Daten folgendes sehen (der Aufbau des SDO-Telegramms ist in Kapitel 3.4.5.1 beschrieben):

COB-ID	Control-Byte	Index		Subindex	Daten	Kommentar
		Low-Byte	High-Byte			
601	40	41	60	00 <sub>h</sub>	00 00 00 00	
581	4B	41	60	00 <sub>h</sub>	40 00 00 00	Antworttelegramm
	2 Byte Daten				Status	

Ist die Leistungsspannung vorhanden und das Hardware - Enable liegt auf High - Signal (24 V gegen DGND) kann durch SDO-Write (6040<sub>h</sub>, 00<sub>h</sub>, 0x7) versucht werden den Antrieb in den Zustand „Switched on“ zu schalten. Im Erfolgsfall wird dies in der SDO - Antwort positiv quittiert (Controlbyte 0 im Datenfeld = 60<sub>h</sub>).

Switch on

Die Nachrichten sehen dann wie folgt aus:

COB-ID	Control-Byte	Index		Subindex	Daten	Kommentar
		Low-Byte	High-Byte			
601	23	40	60	00 <sub>h</sub>	07 00 00 00	controlword
581	60	40	60	00 <sub>h</sub>	00 00 00 00	Antworttelegramm

controlword = 0x0007

Bedeutung: Bit 0, Bit 1, Bit 2 gesetzt ⇒ Switch On,  
Disable Voltage off, Quick Stop off

Statusabfrage 2

Der neue Zustand kann dann wieder abgefragt werden und liefert das folgende Ergebnis:

COB-ID	Control-Byte	Index		Subindex	Daten	Kommentar
		Low-Byte	High-Byte			
601	40	41	60	00 <sub>h</sub>	—	Status abfragen
581	4B	41	60	00 <sub>h</sub>	23 00 00 00	Antworttelegramm

Status = 0x0023

Bedeutung: Bit 0, Bit 1, Bit 5 gesetzt ⇒ ready to Switch On,  
Switched On, Quick Stop

### 6.1.3 Beispiel für PDO-Bedienung

Es sollen alle vier möglichen PDOs im Betrieb verwendet werden:

1. RPDO: PDO Trajektorie für eine Achse
2. RPDO: PDO Steuerwort und Modeumschaltung
1. TPDO: PDO erweiterter Status
2. TPDO: PDO mit Inkrementeller Istposition, Drehzahl und Betriebsartanzeige

Vorgehensweise:

Da das erste RPDO nicht in der geforderten Form vom Antrieb vordefiniert zur Verfügung gestellt wird ( $\Rightarrow$  Kap. 4.3.1) muss es selbst zusammengestellt werden. Dazu ist vorweg zu prüfen, ob die Einträge für die inkrementelle Sollwertvorgabe mappbar zur Verfügung steht.

Das ist bei Objekt 2022<sub>h</sub> Subindex 04<sub>h</sub> der Fall. Also wird das 1. RPDO selektiert:

COB-ID	Control-Byte	Index		Subindex	Daten	Kommentar
		Low-Byte	High-Byte			
601	2F	00	26	00 <sub>h</sub>	25 00 00 00	PDO 37 als 1. RPDO wählen
581	60	00	26	00 <sub>h</sub>	00 00 00 00	Antworttelegramm

Damit ist das frei mappbare RPDO 37 ausgewählt. Im nächsten Schritt muss dieses PDO mit Daten versehen werden. Dies geschieht über die Mappingparameter für das erste RPDO:

COB-ID	Control-Byte	Index		Subindex	Daten	Kommentar
		Low-Byte	High-Byte			
601	23	00	16	00 <sub>h</sub>	00 00 00 00	Daten für 1. RPDO löschen
581	60	00	16	00 <sub>h</sub>	00 00 00 00	Antworttelegramm
601	23	00	16	01 <sub>h</sub>	20 04 22 20	Daten für ersten Eintrag des 1. RPDO Objekt 2022 <sub>h</sub> Subindex 04 <sub>h</sub> , Datenlänge 32 Bit
581	60	00	16	01 <sub>h</sub>	00 00 00 00	Antworttelegramm

Damit ist dieses erste PDO datenmäßig festgelegt, es enthält vier Byte Nutzdaten.

Jetzt können die Kommunikationsparameter festgelegt werden:

Es soll standardmäßig auf COB-ID 201<sub>h</sub> reagieren. Daher muss der Subindex 01<sub>h</sub> auf seinem Defaultwert bleiben. Aber der Antrieb soll auf jedes SYNC-Objekt reagieren, daher muss bei Subindex 2 ein Wert von 1 vorgegeben werden. Also:

COB-ID	Control-Byte	Index		Subindex	Daten	Kommentar
		Low-Byte	High-Byte			
601	2F	00	14	02 <sub>h</sub>	01 00 00 00	Reaktion auf jedes SYNC
581	60	00	14	02 <sub>h</sub>	00 00 00 00	Antworttelegramm

Das zweite RPDO soll zwei Komponenten enthalten, das CANopen Controlwort (Objekt 6040<sub>h</sub> Subindex 00<sub>h</sub>) und das Objekt zur Änderung der Betriebsart (Objekt 6060<sub>h</sub> Subindex 00<sub>h</sub>).

Die Auswahl des 2. RPDO sieht dann wie folgt aus:

COB-ID	Control-Byte	Index		Subindex	Daten	Kommentar
		Low-Byte	High-Byte			
601	2F	01	26	00 <sub>h</sub>	26 00 00 00	PDO 38 als 2. RPDO wählen
581	60	01	26	00 <sub>h</sub>	00 00 00 00	Antworttelegramm

Daraufhin wird das Mapping festgelegt:

COB-ID	Control-Byte	Index		Subindex	Daten	Kommentar
		Low-Byte	High-Byte			
601	23	01	16	00 <sub>h</sub>	00 00 00 00	Daten für 2. RPDO löschen
581	60	01	16	00 <sub>h</sub>	00 00 00 00	Antworttelegramm
601	23	01	16	01 <sub>h</sub>	10 00 40 60	Daten für ersten Eintrag des 2. RPDO Objekt 6040 <sub>h</sub> Subindex 00 <sub>h</sub> , Datenlänge 16 Bit
581	60	01	16	01 <sub>h</sub>	00 00 00 00	Antworttelegramm
601	23	01	16	02 <sub>h</sub>	08 00 60 60	Daten für zweiten Eintrag des 2. RPDO Objekt 6060 <sub>h</sub> Subindex 00 <sub>h</sub> , Datenlänge 8 Bit
581	60	01	16	02 <sub>h</sub>	00 00 00 00	Antworttelegramm

Dieses Objekt soll sofort ausgewertet werden, daher können die Kommunikationsparameter auf ihren Defaultwerten bleiben.

Das erste TPDO ist im Antrieb schon vorhanden, es muss nur ausgewählt werden:

COB-ID	Control-Byte	Index		Subindex	Daten	Kommentar
		Low-Byte	High-Byte			
601	2F	00	2A	00 <sub>h</sub>	17 00 00 00	PDO 23 als 1. TPDO wählen
581	60	00	2A	00 <sub>h</sub>	00 00 00 00	Antworttelegramm

Das Mapping dazu kann über das Objekt 1A00<sub>h</sub> ausgelesen werden. Das PDO enthält 2 Byte für das CANopen-Zustandswort und 4 Byte für das Herstellerstatusregister.

Das zweite TPDO wird wieder zusammengestellt:

1. Die Anwahl über Objekt 2A01:

COB-ID	Control-Byte	Index		Subindex	Daten	Kommentar
		Low-Byte	High-Byte			
601	2F	01	2A	00 <sub>h</sub>	25 00 00 00	PDO 37 als 2. TPDO wählen
581	60	01	2A	00 <sub>h</sub>	00 00 00 00	Antworttelegramm

2. Das Mapping der drei geforderten Komponenten:

COB-ID	Control-Byte	Index		Subindex	Daten	Kommentar
		Low-Byte	High-Byte			
601	23	01	1A	00 <sub>h</sub>	00 00 00 00	Daten für 2. TPDO löschen
581	60	01	1A	00 <sub>h</sub>	00 00 00 00	Antworttelegramm
601	23	01	1A	01 <sub>h</sub>	20 03 70 20	Daten für ersten Eintrag des 2. TPDO Objekt 2070 <sub>h</sub> Subindex 03 <sub>h</sub> , Datenlänge 32 Bit
581	60	01	1A	01 <sub>h</sub>	00 00 00 00	Antworttelegramm
601	23	01	1A	02 <sub>h</sub>	18 02 70 20	Daten für zweiten Eintrag des 2. TPDO Objekt 2070 <sub>h</sub> Subindex 02 <sub>h</sub> , Datenlänge 24 Bit
581	60	01	1A	02 <sub>h</sub>	00 00 00 00	Antworttelegramm
601	23	01	1A	03 <sub>h</sub>	08 00 61 60	Daten für dritten Eintrag des 2. TPDO Objekt 6061 <sub>h</sub> Subindex 00 <sub>h</sub> , Datenlänge 8 Bit
581	60	01	1A	03 <sub>h</sub>	00 00 00 00	Antworttelegramm

Jetzt können die Kommunikationsparameter festgelegt werden. Der Antrieb soll auf jedes SYNC-Objekt reagieren, daher muss bei Subindex 2 ein Wert von 1 vorgegeben werden. Also:

COB-ID	Control-Byte	Index		Subindex	Daten	Kommentar
		Low-Byte	High-Byte			
601	2F	01	18	02 <sub>h</sub>	01 00 00 00	2. TPDO: Reaktion auf jedes SYNC
581	60	01	18	02 <sub>h</sub>	00 00 00 00	Antworttelegramm

### 6.1.4 Beispiel für die Referenzierung

Beim Betrieb des SERVOSTAR 400/600 als Linearachse muss, bevor Positionierungen durchgeführt werden können, ein Referenzpunkt festgelegt werden. Dies kann über ein einfaches "Referenzpunkt setzen" (Bit 12 Steuerwort = 0 -> 1 -> 0 bzw. Referenzfahrt 35 im Mode Homing) oder über das Starten einer Referenzfahrt entweder im herstellerspezifischen Mode "Referenzierung" (0xF9) oder im Homing Mode (0x6) erfolgen.

Hier wird beispielhaft das Vorgehen im Mode "Referenzierung" aufgezeigt.

Dazu wird zuerst in den Modus "Referenzfahrt" geschaltet:

COB-ID	Control-Byte	Index		Subindex	Daten	Kommentar
		Low-Byte	High-Byte			
601	2F	60	60	00 <sub>h</sub>	F9 00 00 00	Mode Referenzierung
581	60	60	60	00 <sub>h</sub>	00 00 00 00	O.K. - Meldung

Im folgenden werden alle die Referenzfahrt betreffenden Parameter über den Bus eingestellt. Wenn man sicher sein kann, dass niemand die Parameter im Gerät geändert hat, kann dieser Teil entfallen, da die Geräte die Daten nullspannungssicher speichern können. (Vorher müssen die Eingänge als Endschalter konfiguriert sein.)

COB-ID	Control-Byte	Index		Subindex	Daten	Kommentar
		Low-Byte	High-Byte			
601	2F	24	20	01 <sub>h</sub>	17 00 00 00	Referenzfahrt auf Endschalter & Resolvnullpunkt
581	60	24	20	01 <sub>h</sub>	00 00 00 00	OK
601	2F	24	20	02 <sub>h</sub>	00 00 00 00	Negative Fahrtrichtung
581	60	24	20	02 <sub>h</sub>	00 00 00 00	OK
601	2B	24	20	03 <sub>h</sub>	10 27 00 00	v = 10mm/s
581	60	24	20	03 <sub>h</sub>	00 00 00 00	OK
601	2B	24	20	04 <sub>h</sub>	32 00 00 00	Anfahrrampe 50ms
581	60	24	20	04 <sub>h</sub>	00 00 00 00	OK
601	2B	24	20	05 <sub>h</sub>	32 00 00 00	Bremsrampe 50ms
581	60	24	20	05 <sub>h</sub>	00 00 00 00	OK
601	23	24	20	06 <sub>h</sub>	30 75 00 00	Referenzoffset 30000µm
581	60	24	20	06 <sub>h</sub>	00 00 00 00	OK

Zur Überprüfung der Bitsignale, die für den Referenzfahrtablauf wichtig sind, soll hier das PDO erweiterter Status (PDO 23) verwendet werden.

Dazu wird zuerst das PDO 23 als TPDO 1 ausgewählt:

COB-ID	Control-Byte	Index		Subindex	Daten	Kommentar
		Low-Byte	High-Byte			
601	2F	00	2A	00 <sub>h</sub>	17 00 00 00	PDO 23 als 1. TPDO auswählen
601	60	00	2A	00 <sub>h</sub>	00 00 00 00	Antworttelegramm

Das TPDO1 setzt sich jetzt also aus 6 Bytes zusammen, wobei die ersten zwei Bytes das CANopen Zustandswort (Objekt 6041) enthalten, die anderen vier Bytes das herstellerspezifische Statusregister (Objekt 1002).

Diese Belegung kann über die Objekts für das PDO-Mapping (Objekt 1A00<sub>h</sub>, Subindex 00<sub>h</sub>-02<sub>h</sub>) abgefragt werden. Danach werden die PDOs über ein NMT - Objekt freigegeben:

COB-ID	Command specifier (CS)	Node-ID
0	1	1

Ab diesem Zeitpunkt werden Zustandsänderungen am TPDO1 automatisch gemeldet.

Nun kann die Referenzfahrt mit Bit 4 des CANopen - Controlwords gestartet werden:

COB-ID	Control-Byte	Index		Subindex	Daten	Kommentar
		Low-Byte	High-Byte			
601	2B	40	60	00 <sub>h</sub>	1F 00 00 00	Mode
581	60	40	60	00 <sub>h</sub>	00 00 00 00	Referenzfahrt läuft bis Referenzbedingung erfüllt

Mit der steigenden Flanke des Bits "Referenzpunkt gesetzt" im herstellerspezifischen Statusregisters kann erkannt werden, dass der Sevoantrieb sein Positionssystem geeicht hat. Der Vorgang der Referenzierung ist mit Rücksetzen des Bits "Fahrauftrag aktiv" beendet.

Ein TPDO1 könnte also wie folgt aussehen:

COB-ID	Daten	Kommentar
181	05 27 00 00 0A 54	Mode

Der Zustand der Referenzfahrt lässt sich über das erweiterte Statusregister, Bit 17 (Referenzpunkt gesetzt) erkennen.

### 6.1.5 Beispiel für Fahrsatzverarbeitung

Lageregelung einschalten

COB-ID	Control-Byte	Index		Subindex	Daten	Kommentar
		Low-Byte	High-Byte			
601	2F	60	60	00 <sub>h</sub>	FF 00 00 00	Lageregelmode
581	60	60	60	00 <sub>h</sub>	00 00 00 00	Lageregelung eingeschaltet

Zweites Receive-PDO mappen

(Starte Fahrsatz, Fahrsätze für Beispiel schon definiert. 1. Receive-PDO steht standardmäßig auf dem controlword.)

COB-ID	Control-Byte	Index		Subindex	Daten	Kommentar
		Low-Byte	High-Byte			
601	2F	01	26	00 <sub>h</sub>	23 00 00 00	Starte Fahrsatz Objekt
581	60	01	26	00 <sub>h</sub>	00 00 00 00	OK, gemappt

NMT-Zustandsmaschine auf "operational" schalten

COB-ID	Command specifier (CS)	Node-ID
0	1	1

Zweites Receive-Objekt ansprechen

COB-ID	Fahrsatznummer-Low	Fahrsatznummer-High
301	01	00

Antwort: keine, angegebener Fahrsatz 1 wird abgearbeitet

Motor Quick Stop

COB-ID	Control-Low	Control-High
201	03	00

Antwort: keine, Motor wird mit t\_not angehalten

Regler sperren (disable)

COB-ID	Control-Low	Control-High	Fahrsatznummer
201	03	00	1

Antwort: keine, Antrieb wird drehmomentfrei

### 6.1.6 Beispiel für Bedienung des Profile-Position Modes

Dieses Kapitel zeigt die Bedienung des Profile-Position Modes. Dazu werden die PDOs wie folgt voreingestellt:

- Erstes RPDO: PDO Steuerwort (Nr. 1)
- Zweites RPDO: frei mappbares PDO 2 (Nr. 38)
- Erstes TPDO: frei mappbares PDO 1 (Nr. 37)
- Zweites TPDO: frei mappbares PDO 2 (Nr. 38)

Die Telegramme sehen analog zum Beispiel für PDO-Bedienung (⇒ Kap. 6.1.3) aus.

In die frei mappbaren PDOs werden Daten entsprechend der folgenden Beispieltelegramme gelegt:

1.: zweites RPDO:

COB-ID	Control-Byte	Index		Subindex	Daten	Kommentar
		Low-Byte	High-Byte			
601	23	01	16	00h	00 00 00 00	2. RPDO: Mapping löschen
581	60	01	16	00h	00 00 00 00	
601	23	01	16	01h	20 00 7A 60	2. RPDO Eintrag 1: target_position
581	60	01	16	01h	00 00 00 00	
601	23	01	16	02h	20 00 81 60	2. RPDO Eintrag 2: profile_velocity
581	60	01	16	02h	00 00 00 00	

2.: erstes TPDO:

COB-ID	Control-Byte	Index		Subindex	Daten	Kommentar
		Low-Byte	High-Byte			
601	23	00	1A	00h	00 00 00 00	1. TPDO: Mapping löschen
581	60	00	1A	00h	00 00 00 00	
601	23	00	1A	01h	10 00 41 60	1. TPDO Eintrag 1: Profilstandswort
581	60	00	1A	01h	00 00 00 00	
601	23	00	1A	02h	08 09 80 20	1. TPDO Eintrag 2: TRJSTAT 3. Byte = Herst.Status 3. Byte
581	60	00	1A	02h	00 00 00 00	

3.: zweites TPDO:

COB-ID	Control-Byte	Index		Subindex	Daten	Kommentar
		Low-Byte	High-Byte			
601	23	01	1A	00h	00 00 00 00	2. TPDO: Mapping löschen
581	60	01	1A	00h	00 00 00 00	
601	23	01	1A	01h	20 00 64 60	2. TPDO Eintrag 1: position_actual_value
581	60	01	1A	01h	00 00 00 00	
601	23	01	1A	02h	20 00 6C 60	1. TPDO Eintrag 2: velocity_actual_value
581	60	01	1A	02h	00 00 00 00	

Das erste TPDO soll ereignisgesteuert gesendet werden. Da dies dem Defaultwert der Kommunikationsparameter entspricht, braucht hier nichts verändert zu werden. Das zweite TPDO soll mit jedem SYNC vom Antrieb gesendet werden:

COB-ID	Control-Byte	Index		Subindex	Daten	Kommentar
		Low-Byte	High-Byte			
601	23	01	18	02h	01 00 00 00	2. TPDO mit jedem SYNC versenden
581	60	01	18	02h	00 00 00 00	

Nach der Festlegung der PDOs können diese mit dem NMT freigegeben werden:

COB-ID	Command specifier (CS)	Node-ID
0	1	1

Falls die mechanische Auflösung vorgegeben werden soll, kann diese nun über Objekt 6093<sub>h</sub>, Subindex 01<sub>h</sub> und 02<sub>h</sub> geschrieben werden. Die Voreinstellung nach dem Einschalten des Antriebs entspricht den antriebspezifischen Faktoren PGEARI und PGEARO:

COB-ID	Control-Byte	Index		Subindex	Daten	Kommentar
		Low-Byte	High-Byte			
601	23	93	60	01 <sub>h</sub>	00 00 10 00	2 <sup>20</sup> Inkremente
581	60	93	60	01 <sub>h</sub>	00 00 00 00	
601	23	93	60	02 <sub>h</sub>	A0 8C 00 00	3600 Nutzereinheiten
581	60	93	60	02 <sub>h</sub>	00 00 00 00	

Im Beispiel könnte man zum Beispiel einen Rundtisch mit einer Winkelauflösung von 0,1 Grad bedienen.

Nach diesen Einstellungen kann eine Referenzfahrt eingestellt und angestoßen werden:

COB-ID	Control-Byte	Index		Subindex	Daten	Kommentar
		Low-Byte	High-Byte			
601	2B	40	60	00 <sub>h</sub>	0F 00 00 00	Controlword: Operation Enable
581	60	40	60	00 <sub>h</sub>	00 00 00 00	
601	2F	60	60	00 <sub>h</sub>	06 00 00 00	Betriebsart: Homing Mode einstellen
581	60	60	60	00 <sub>h</sub>	00 00 00 00	
601	2F	98	60	00 <sub>h</sub>	0C 00 00 00	Referenzfahrt 12, negative Fahrtrichtung
581	60	98	60	00 <sub>h</sub>	00 00 00 00	
601	23	99	60	01 <sub>h</sub>	40 19 01 00	Referenzfahrtgeschw. 72000 Einh./s = 2 s <sup>-1</sup>
581	60	99	60	01 <sub>h</sub>	00 00 00 00	
601	2B	40	60	00 <sub>h</sub>	1F 00 00 00	Referenzfahrt starten
581	60	40	60	00 <sub>h</sub>	00 00 00 00	

Nach dem Start der Referenzfahrt könnten dann folgende Telegramme des 1. TPDO kommen:

COB-ID	Byte 0	Byte 1	Byte 2
181	27	00	21
181	27	10	27
181	27	10	23
181	27	14	2B
181	27	15	2A

Für die Erkennung des Endes der Referenzfahrt sind folgende Bits des Bytes 2 erforderlich: Bit 0 = 0 Referenzfahrt abgeschlossen, Bit 1 = 1 Referenzpunkt gesetzt, Bit 3 = 1 In Position. Anschließend wird die Referenzfahrt auch über das Steuerwort beendet, diesmal über das 1. RPDO:

COB-ID	Byte 0	Byte 1
201	0F	00

Nun kann in den Profile Position Mode geschaltet werden und Rampen für Positionierungen vorgegeben werden:

COB-ID	Control-Byte	Index		Subindex	Daten	Kommentar
		Low-Byte	High-Byte			
601	23	83	60	00 <sub>h</sub>	32 00 00 00	50ms Beschleunigungszeit
581	60	83	60	00 <sub>h</sub>	00 00 00 00	
601	23	84	60	00 <sub>h</sub>	32 00 00 00	50ms Bremszeit
581	60	84	60	00 <sub>h</sub>	00 00 00 00	
601	2F	60	60	00 <sub>h</sub>	01 00 00 00	Profile Position Mode
581	60	60	60	00 <sub>h</sub>	00 00 00 00	

Eine Positionierung kann nun über Vorgabe der Sollwerte über das 1. RPDO und anschließenden Start über das 2. RPDO. Dabei kommt der Handshake über New Setpoint (Controlword) und Setpoint acknowledge (status word) zum Einsatz.

a.: Sollwert

COB-ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
301	F4	01	00	00	E8	03	00	00

b.: Controlword mit „new setpoint“ – Bit (Bit 4) gesetzt

COB-ID	Byte 0	Byte 1
201	1F	00

c.: Warten, dass das CANopen status word „setpoint acknowledge“ (Bit 12) meldet: (z.B.)

COB-ID	Byte 0	Byte 1	Byte 2
181	27	15	03

d.: Controlword mit „new setpoint“ – Bit (Bit 4) sofort zurückgesetzt

COB-ID	Byte 0	Byte 1
201	0F	00

e.: Antrieb nimmt setpoint acknowledge zurück

COB-ID	Byte 0	Byte 1	Byte 2
181	27	01	03

Auf Abschluß der Positionierung warten

COB-ID	Byte 0	Byte 1	Byte 2
181	27	05	0A

### 6.1.7

### ASCII - Kommunikation

Die ASCII-Kommunikation findet sinnvollerweise über PDOs statt, da sie so effizienter genutzt werden kann. Dazu muss sich die NMT-Zustandmaschine im Zustand "operational" befinden.

Beispiel: Parameter T-Tacho (siehe Online-Hilfede Inbetriebnahme-Software) lesen.  
(Alle Daten Hexadezimal mit ASCII-Entsprechung in eckigen Klammern darunter)

Richtung	COB-ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Master ⇒ SERVOSTAR	301	47h [G]	56h [V]	46h [F]	42h [B]	54h [T]	0Dh [CR]	0Ah [LF]	0h [NUL]
SERVOSTAR ⇒ Master	281	30h [0]	2Eh [.]	36h [6]	0h [NUL]	0h [NUL]	0h [NUL]	0h [NUL]	0h [NUL]
SERVOSTAR ⇒ Master	281	0Dh [CR]	0Ah [LF]	0h [NUL]	0h [NUL]	0h [NUL]	0h [NUL]	0h [NUL]	0h [NUL]
SERVOSTAR ⇒ Master	281	2Dh [-]	2Dh [-]	3Eh [>]	0h [NUL]	0h [NUL]	0h [NUL]	0h [NUL]	0h [NUL]

Erklärung: In Telegramm 1 fragt der Master den Parameter "GVFBT" an, abgeschlossen von den ASCII-Codes "CR LF". Das freie letzte Byte wird mit "NUL" gefüllt.  
Die Antwort des SERVOSTAR erfolgt in Telegramm 2 mit dem Wert "0.6", dem Abschlusscode "CR LF" und dem Prompt für den nächsten Parameter bzw. das nächste Kommando "—>". Die Segmentierung der Antwort in drei Telegramme erfolgt nicht zwingend, sondern hängt von der eingestellten Übertragungsrate und internen Synchronisationsmechanismen ab.

### 6.1.8 Test für Synctelegramme

Aufgabenstellung:

1. PDO mit Starte Fahrsatz belegen (1<sup>st</sup> Receive PDO)
2. PDO mit Ist - Lage (PDO21) belegen (1<sup>st</sup> Transmit PDO), mit jedem 2. SYNC auslösen
3. PDO mit statusword (PDO1) belegen (2<sup>nd</sup> Transmit PDO), mit jedem 3. SYNC auslösen

Telegramme mit jeweiligen Antworten:

COB-ID	Control-Byte	Index		Subindex	Daten	Kommentar
		Low-Byte	High-Byte			
601	2F	00	26	00 <sub>h</sub>	23 00 00 00	PDO Starte Fahrsatz auf
581	60	00	26	00 <sub>h</sub>	00 00 00 00	1. RPDO setzen
601	2F	00	2A	00 <sub>h</sub>	16 00 00 00	PDO Ist-Lage auf 1. TPDO setzen
581	60	00	2A	00 <sub>h</sub>	00 00 00 00	
601	2F	01	2A	00 <sub>h</sub>	17 00 00 00	PDO erweitertes statusword auf 2. TPDO setzen
581	60	01	2A	00 <sub>h</sub>	00 00 00 00	
601	2F	00	18	02 <sub>h</sub>	02 00 00 00	1. TPDO auf Trigger, durch jedes 2. SYNC ersetzen
581	60	00	18	02 <sub>h</sub>	00 00 00 00	
601	2F	01	18	02 <sub>h</sub>	03 00 00 00	2. TPDO auf Trigger, durch jedes 3. SYNC ersetzen
581	60	01	18	02 <sub>h</sub>	00 00 00 00	

### 6.1.9 SYNC-Objekt

COB-ID
080

Bedeutung: Bei jedem zweiten SYNC kommt das Objekt 181 (TPDO 1), bei jedem dritten SYNC kommt das Objekt 281 (TPDO 2).

### 6.1.10 Emergency-Objekt

Zieht man zwischendurch zum Beispiel den Resolverstecker ab, löst man einen schweren Fehler im Regler aus. Dies führt zu einem Emergency - Telegramm.

COB-ID	Emergency error code		Error register	
	Low	High		
081	10	43	08	00 00 00 00
081	00	00	88	00 00 00 00

Motortemperatur, temperature, manufacturer specific

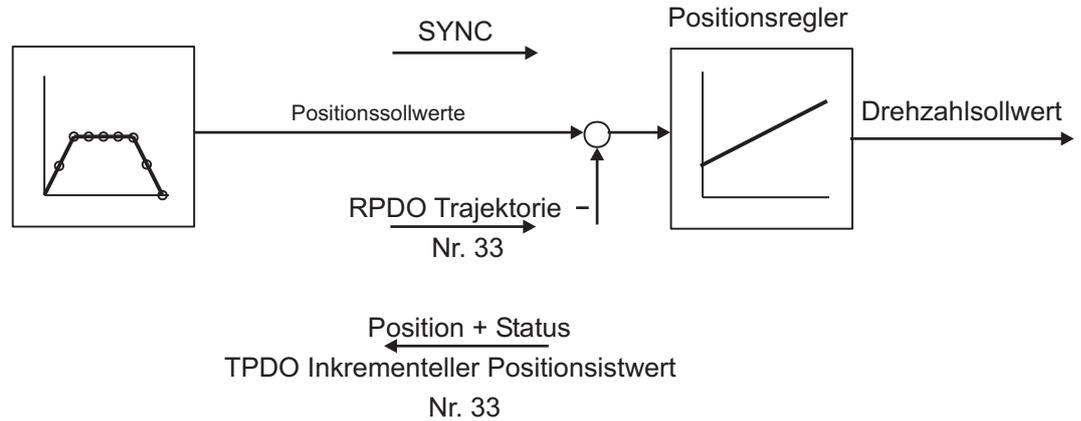
## 6.2 Spezielle Applikationen

### 6.2.1 Externe Trajektorie

#### 6.2.1.1 Lageregler im Servoverstärker

In diesem Beispiel wird der Fall betrachtet, in dem 2 Achsen jeweils Positionssollwerte über das RPDO 33 Trajektorie erhalten.

Reglerstruktur bei Lageregler im Servoverstärker:



#### Beschreibung

Alle Angaben sind hexadezimal. Die beiden Achsen im System haben im Beispiel die Stationsadressen 1 und 2.

Beispieltelegramme und -antworten:

Voraussetzungen:

- Es muss für beide Achsen definiert werden welche Auflösungen innerhalb einer Motorumdrehung verwendet werden sollen: PRBASE auf 16 oder 20 Bit pro Umdrehung.
- Das Zeitraster für Trajektorienvorgabe muss über den Parameter PTBASE eingestellt werden. Dabei entspricht die Einheit einem Wert von 250 Mikrosekunden, d.h. z.B. PTBASE = 8 ergibt ein Trajektorienzeitwert von 2 Millisekunden.
- Die Parameter müssen im EEPROM gespeichert werden.
- Die gespeicherten Werte stehen erst nach einem Neustart zur Verfügung.

Das PDO Trajektorie enthält 2 Trajektorienollwerte und kann an mehrere Stationen gleichzeitig gesendet werden, wobei sich jede Station ihre Trajektorienaten herausholen kann.

Zweites Receive-PDO für beide Achsen auf das RPDO 33 Trajektorie (33<sub>d</sub> = 21<sub>h</sub>) mappen:

COB-ID	Control-Byte	Index		Subindex	Daten	Kommentar
		Low-Byte	High-Byte			
601	2F	01	26	00 <sub>h</sub>	21 00 00 00	PDO Trajektorie für 1.
581	60	01	26	00 <sub>h</sub>	00 00 00 00	Achse eingestellt
602	2F	01	26	00 <sub>h</sub>	21 00 00 00	PDO Trajektorie für 2.
582	60	01	26	00 <sub>h</sub>	00 00 00 00	Achse eingestellt

Nachdem das PDO Trajektorie nun auf zwei Achsen gemappt ist, müssen die Kommunikationsparameter bei beiden so eingestellt werden, dass sie auf denselben Kommunikationsobjekt-Identifizier (COB-ID) reagieren. Der COB-ID für die erste Station kann dabei auf seinem Defaultwert 301 bleiben, der für die zweite Station kann dann auf diesen umgemappt werden:

COB-ID	Control-Byte	Index		Subindex	Daten	Kommentar
		Low-Byte	High-Byte			
602	23	01	14	01 <sub>h</sub>	01 03 00 00	2. RPDO für 2. Achse auf 301 mappen
582	60	01	14	01 <sub>h</sub>	00 00 00 00	

Beide Stationen reagieren jetzt auf denselben COB-ID 301.

Mit dem Objekt 2721<sub>h</sub> Subindex 00<sub>h</sub> kann dann bestimmt werden aus welchem Teil des 8 Byte-Datenfelds jede Achse ihre Trajektorie bezieht. Mit dem Wert 0 werden dabei die Bytes 0..3 der Daten ausgewählt, mit dem Wert 1 die Bytes 4..7:

COB-ID	Control-Byte	Index		Subindex	Daten	Kommentar
		Low-Byte	High-Byte			
601	2F	21	27	00 <sub>h</sub>	00 00 00 00	1. Achse verwendet Byte 0..3 als Wert
581	60	21	27	00 <sub>h</sub>	00 00 00 00	
602	2F	21	27	00 <sub>h</sub>	04 00 00 00	2. Achse verwendet Byte 4..7 als Wert
582	60	21	27	00 <sub>h</sub>	00 00 00 00	

Die Istpositionen der Achsen sollen auch als inkrementelle Istpositionen an die Steuerung zurückkommen. Daher werden die jeweils zweiten Transmit-PDOs auf das TPDO 33 Inkrementeller Positionswert (33<sub>d</sub> = 21<sub>h</sub>) gemappt.:

COB-ID	Control-Byte	Index		Subindex	Daten	Kommentar
		Low-Byte	High-Byte			
601	2F	01	2A	00 <sub>h</sub>	21 00 00 00	PDO Trajektorie für 1. Achse eingestellt
581	60	01	2A	00 <sub>h</sub>	00 00 00 00	
602	2F	01	2A	00 <sub>h</sub>	21 00 00 00	PDO Trajektorie für 2. Achse eingestellt
582	60	01	2A	00 <sub>h</sub>	00 00 00 00	

Es wird hier davon ausgegangen, dass die beiden Geräte mit jedem SYNC-Kommando neue Trajektorienwerte übernehmen und ihre inkrementellen Positionswerte zurücksenden sollen. Also müssen die Kommunikationsparameter entsprechend eingestellt werden:

COB-ID	Control-Byte	Index		Subindex	Daten	Kommentar
		Low-Byte	High-Byte			
601	2F	01	14	02 <sub>h</sub>	01 00 00 00	RPDO 2 Achse 1 Reaktion auf jedes SYNC
581	60	01	14	02 <sub>h</sub>	00 00 00 00	
602	2F	01	14	02 <sub>h</sub>	01 00 00 00	RPDO 2 Achse 2 Reaktion auf jedes SYNC
582	60	01	14	02 <sub>h</sub>	00 00 00 00	
601	2F	01	18	02 <sub>h</sub>	01 00 00 00	TPDO 2 Achse 1 Reaktion auf jedes SYNC
581	60	01	18	02 <sub>h</sub>	00 00 00 00	
602	2F	01	18	02 <sub>h</sub>	01 00 00 00	TPDO 2 Achse 2 Reaktion auf jedes SYNC
582	60	01	18	02 <sub>h</sub>	00 00 00 00	

Damit Trajektorien gefahren werden können, müssen sich beide Servoverstärker im entsprechenden Modus befinden. Dies wird über den Index 6060<sub>h</sub> eingestellt:

COB-ID	Control-Byte	Index		Subindex	Daten	Kommentar
		Low-Byte	High-Byte			
601	2F	60	60	00 <sub>h</sub>	FA 00 00 00	Trajektorienmodus für Achse 1 einstellen
581	60	60	60	00 <sub>h</sub>	00 00 00 00	
602	2F	60	60	00 <sub>h</sub>	FA 00 00 00	Trajektorienmodus für Achse 2 einstellen
582	60	60	60	00 <sub>h</sub>	00 00 00 00	

Um die Achsen zu starten, müssen nun die Servoverstärker in den fahrbereiten Zustand (operational enable) gebracht und die Netzwerkmanagementfunktionen gestartet werden.

Die Netzwerkmanagementfunktionen, die die Verwendung der Prozessdatenobjekte freischalten, werden durch das folgende Telegramm für beide Achsen gestartet:

NMT (Network Management) - Zustandsmaschine auf „operational“ schalten

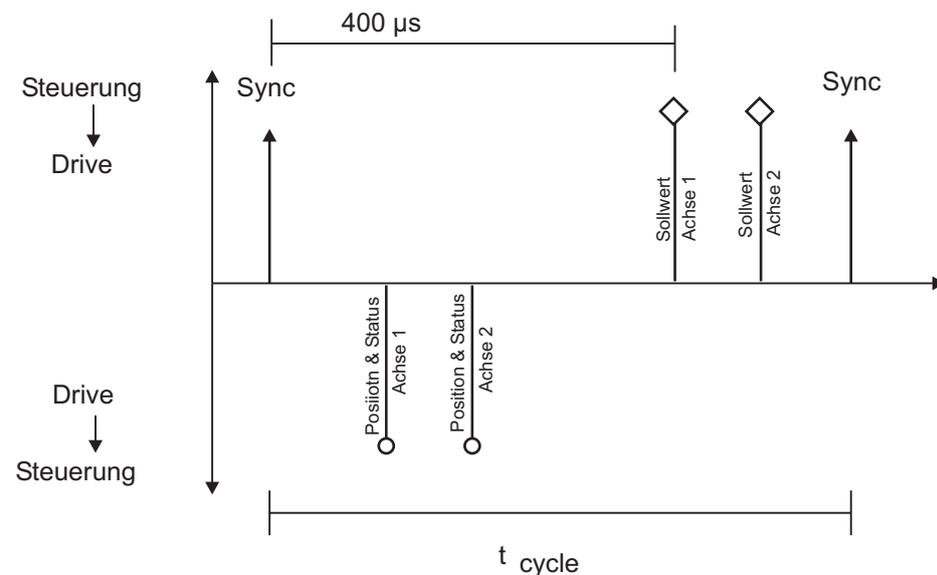
COB-ID	Command specifier (CS)	Node-ID	Kommentar
0	1	1	Alle Achsen NMT freigeben

Danach werden die Servoverstärker einzeln bestromt und in den Zustand „operation enable“ gebracht.

Steuerwort für Enable Operation

COB-ID	Control-Byte	Index		Subindex	Daten	Kommentar
		Low-Byte	High-Byte			
601	2B	40	60	00 <sub>h</sub>	0F 00 00 00	Steuerwort für Achse 1
581	60	40	60	00 <sub>h</sub>	00 00 00 00	
602	2B	40	60	00 <sub>h</sub>	0F 00 00 00	Steuerwort für Achse 2
582	60	40	60	00 <sub>h</sub>	00 00 00 00	

Durch die obige Konfiguration wird jetzt, wie im folgenden Bild dargestellt, ein zyklischer Ablauf ermöglicht:



z.B. 2 Achsen

$t_{cycle}$  1 ms pro Achse bei 1 Mbaud

Nun können über das RPDO 2 Trajektorienwerte für beide Achsen vorgegeben werden, z.B. wie folgt:

COB-ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
301	F4	01	00	00	E8	03	00	00

In diesem Beispiel bekommt die erste Achse einen Trajektorienwert von 500 Inkrementen (Bytes 0..3) und die zweite Achse einen Trajektorienwert von 1000 Inkrementen.

Die Werte werden von den Achsen übernommen und mit Erhalt des nächsten SYNC – Telegramms positioniert.

Das SYNC – Telegramm sieht dabei wie folgt aus:

COB-ID
080

Beide Achsen senden anschließend ihre inkrementellen Positionen und ihre Statusregister beim Erhalt des SYNC-Objekts mit der COB-ID des 2. TPDOs zurück:

COB-ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Kommentar
181	23	01	00	00	00	00	03	44	Position und Herstellerstatusregister Achse1
182	A5	02	00	00	00	00	03	44	Position und Herstellerstatusregister Achse2

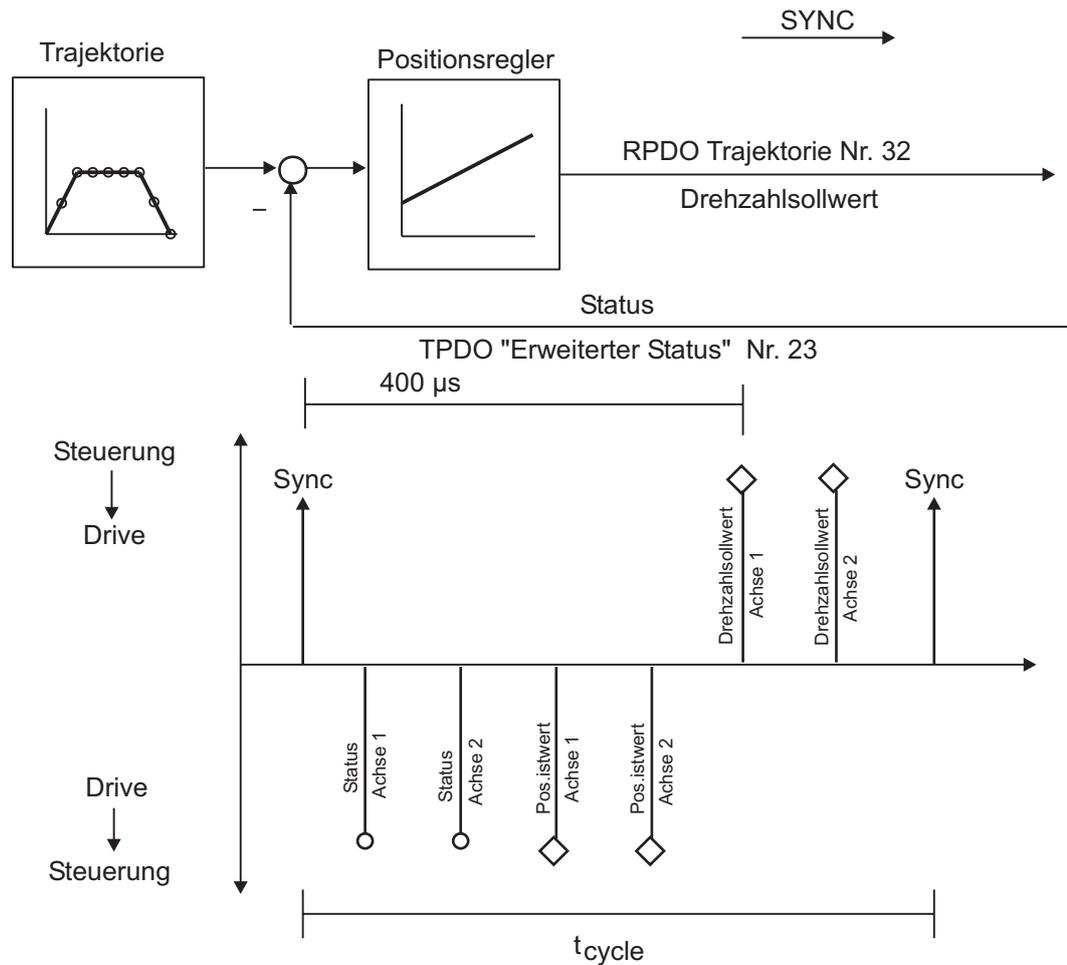
Tritt im Verlauf der Bedienung ein Fehler auf, wird von der betroffenen Achse eine Emergency-Mes- sage gesendet, die z.B. so aussehen kann:

Emergency – Objekt

COB-ID	Emergency error code		Error register	Kategorie	
	Low	High			
081	10	43	08	01	00 00 00 00
081	00	00	08	00	00 00 00 00

6.2.1.2

Lageregler in der Steuerung



z.B. 2 Achsen

$t_{cycle}$  1 ms pro Achse bei 1 MBaud

### 6.3 Beschreibung Objektverzeichnis

Die folgende Tabelle beschreibt das Objektverzeichnis (Object Dictionary).

Die Spalte "Def" weist auf die entsprechende Norm bzw. das herstellerspezifische Profil hin:

**S** = SERVOSTAR                      **3** = DS301                      **4** = DS402

In der Spalte *Zugriff* bedeuten:

**ro** - read only, **wo** - write only

**rwr** - read / write on process input (mappbar für Rx-PDOs)

**rww** - read 7 write on process output (mappbar für Tx-PDOs)

Index	Sub-index	DEF	Datenformat	Zugriff	PDO mappbar	Kurzbeschreibung	ASCII Objekt
1000 <sub>h</sub>	—	3	UNSIGNED32	ro	—	Gerätetyp	—
1001 <sub>h</sub>	—	3	UNSIGNED8	ro	—	Fehlerregister	—
1002 <sub>h</sub>	—	3	UNSIGNED32	ro	☒	Herstellerspezifisches Statusregister	—
1003 <sub>h</sub>	00 <sub>h</sub>	3	UNSIGNED8	ro	—	Pre-defined error field (Anzahl der Einträge)	—
	01 <sub>h</sub>	3	UNSIGNED32	rw	—	Zuletzt gemeldeter Fehler	—
1004 <sub>h</sub>	00 <sub>h</sub>	3	UNSIGNED32	ro	—	Zahl der unterstützten PDOs	—
	01 <sub>h</sub>	3	UNSIGNED32	ro	—	Zahl der synchronen PDOs	—
	02 <sub>h</sub>	3	UNSIGNED32	ro	—	Zahl der asynchronen PDOs	—
1005 <sub>h</sub>	—	3	UNSIGNED32	ro	—	COB-ID SYNC message	—
1006 <sub>h</sub>	—	3	UNSIGNED32	rw	—	Zykluszeit der Kommunikation	—
1007 <sub>h</sub>	—	3	UNSIGNED32	rw	—	Zeitfenster für synchrone CAN-Nachrichten	—
1008 <sub>h</sub>	—	3	Visible String	ro	—	Gerätename	—
100A <sub>h</sub>	—	3	Visible String	ro	—	Softwareversion	VER*
100B <sub>h</sub>	—	3	UNSIGNED32	ro	—	Knotenadresse	ADDR
100C <sub>h</sub>	—	3	UNSIGNED16	rw	—	Guard time	—
100D <sub>h</sub>	—	3	UNSIGNED8	rw	—	Life time factor	—
100E <sub>h</sub>	—	3	UNSIGNED32	rw	—	Node Guarding COB - Identifier	—
100F <sub>h</sub>	—	3	UNSIGNED32	rw	—	Zahl der unterstützten SDOs	—
1010 <sub>h</sub>	00 <sub>h</sub>	3	2	ro	—	Store parameters / Anzahl der Einträge	—
	01 <sub>h</sub>	3	UNSIGNED32	rw	—	Store parameters / save all parameters	—
	02 <sub>h</sub>	3	UNSIGNED32	rw	—	Store parameters / save communication parameters	—
1012 <sub>h</sub>	—	3	UNSIGNED32	rw	—	COB - ID der time stamp message (i.V.)	—
1013 <sub>h</sub>	—	3	UNSIGNED32	rw	—	High resolution time stamp (i.V.)	—
1014 <sub>h</sub>	—	3	UNSIGNED32	rw	—	COB - ID der Emergency message	—
1018 <sub>h</sub>	00 <sub>h</sub>	3	RECORD	ro	—	Identity Object	—
	01 <sub>h</sub>	3	UNSIGNED32	ro	—	Vendor ID	—
	02 <sub>h</sub>	3	UNSIGNED32	ro	—	Product Code	—
	03 <sub>h</sub>	3	UNSIGNED32	ro	—	Revision number	—
	04 <sub>h</sub>	3	UNSIGNED32	ro	—	Serial number	SERIALNO
1400 <sub>h</sub>	00 <sub>h</sub>	3	RECORD	ro	—	Empfangs- PDO 1 Kommunikationsparameter	—
	01 <sub>h</sub>	3	UNSIGNED32	rw	—	PDO COB - ID	—
	02 <sub>h</sub>	3	UNSIGNED8	rw	—	Übertragungstyp	—
	03 <sub>h</sub>	3	UNSIGNED16	rw	—	Inhibit time (bei RPDOs nicht sinnvoll)	—
	04 <sub>h</sub>	3	UNSIGNED8	rw	—	Kompatibilitätseintrag (CMS priority group)	—
1401 <sub>h</sub>	00 <sub>h</sub>	3	RECORD	ro	—	Empfangs- PDO 2 Kommunikationsparameter	—
	01 <sub>h</sub>	3	UNSIGNED32	rw	—	PDO COB - ID	—
	02 <sub>h</sub>	3	UNSIGNED8	rw	—	Übertragungstyp	—
	03 <sub>h</sub>	3	UNSIGNED16	rw	—	Inhibit time (bei RPDOs nicht sinnvoll)	—
	04 <sub>h</sub>	3	UNSIGNED8	rw	—	Kompatibilitätseintrag (CMS priority group)	—
1402 <sub>h</sub>	00 <sub>h</sub>	3	RECORD	ro	—	Empfangs- PDO 3 Kommunikationsparameter	—
	01 <sub>h</sub>	3	UNSIGNED32	rw	—	PDO COB - ID	—
	02 <sub>h</sub>	3	UNSIGNED8	rw	—	Übertragungstyp	—
	03 <sub>h</sub>	3	UNSIGNED16	rw	—	Inhibit time (bei RPDOs nicht sinnvoll)	—
	04 <sub>h</sub>	3	UNSIGNED8	rw	—	Kompatibilitätseintrag (CMS priority group)	—
1403 <sub>h</sub>	00 <sub>h</sub>	3	RECORD	ro	—	Empfangs- PDO 4 Kommunikationsparameter	—
	01 <sub>h</sub>	3	UNSIGNED32	rw	—	PDO COB - ID	—
	02 <sub>h</sub>	3	UNSIGNED8	rw	—	Übertragungstyp	—
	03 <sub>h</sub>	3	UNSIGNED16	rw	—	Inhibit time (bei RPDOs nicht sinnvoll)	—
	04 <sub>h</sub>	3	UNSIGNED8	rw	—	Kompatibilitätseintrag (CMS priority group)	—
1600 <sub>h</sub>	00 <sub>h</sub>	3	RECORD	rw	—	Empfangs- PDO 1 Mappingparameter	—
	01 <sub>h</sub> -08 <sub>h</sub>	3	UNSIGNED32	rw	—	PDO Mapping Parameter für das nte Objekt	—

Index	Sub-index	DEF	Datenformat	Zu-griff	PDO mappbar	Kurzbeschreibung	ASCII Objekt
1601h	00h	3	RECORD	rw	—	Empfangs- PDO 2 Mappingparameter	—
	01h-08h	3	UNSIGNED32	rw	—	PDO Mapping Parameter für das nte Objekt	—
1602h	00h	3	RECORD	rw	—	Empfangs- PDO 3 Mappingparameter	—
	01h-08h	3	UNSIGNED32	rw	—	PDO Mapping Parameter für das nte Objekt	—
1603h	00h	3	RECORD	rw	—	Empfangs- PDO 4 Mappingparameter	—
	01h-08h	3	UNSIGNED32	rw	—	PDO Mapping Parameter für das nte Objekt	—
1800h	00h	3	RECORD	ro	—	Sende- PDO 1 Kommunikationsparameter	—
	01h	3	UNSIGNED32	rw	—	PDO COB - ID	—
	02h	3	UNSIGNED8	rw	—	Übertragungstyp	—
	03h	3	UNSIGNED16	rw	—	Inhibit time	—
	04h	3	UNSIGNED8	rw	—	Kompatibilitätseintrag (CMS priority group)	—
1801h	00h	3	RECORD	ro	—	Sende- PDO 2 Kommunikationsparameter	—
	01h	3	UNSIGNED32	rw	—	PDO COB - ID	—
	02h	3	UNSIGNED8	rw	—	Übertragungstyp	—
	03h	3	UNSIGNED16	rw	—	Inhibit time	—
	04h	3	UNSIGNED8	rw	—	Kompatibilitätseintrag (CMS priority group)	—
1802h	00h	3	RECORD	ro	—	Sende- PDO 3 Kommunikationsparameter	—
	01h	3	UNSIGNED32	rw	—	PDO COB - ID	—
	02h	3	UNSIGNED8	rw	—	Übertragungstyp	—
	03h	3	UNSIGNED16	rw	—	Inhibit time	—
	04h	3	UNSIGNED8	rw	—	Kompatibilitätseintrag (CMS priority group)	—
1803h	00h	3	RECORD	ro	—	Sende- PDO 4 Kommunikationsparameter	—
	01h	3	UNSIGNED32	rw	—	PDO COB - ID	—
	02h	3	UNSIGNED8	rw	—	Übertragungstyp	—
	03h	3	UNSIGNED16	rw	—	Inhibit time	—
	04h	3	UNSIGNED8	rw	—	Kompatibilitätseintrag (CMS priority group)	—
1A00h	00h	3	RECORD	rw	—	Sende- PDO 1 Mappingparameter	—
	01h-08h	3	UNSIGNED32	rw	—	PDO Mapping Parameter für das nte Objekt	—
1A01h	00h	3	RECORD	rw	—	Sende- PDO 2 Mappingparameter	—
	01-08h	3	UNSIGNED32	rw	—	PDO Mapping Parameter für das nte Objekt	—
1A02h	00h	3	RECORD	rw	—	Sende- PDO 3 Mappingparameter	—
	01-08h	3	UNSIGNED32	rw	—	PDO Mapping Parameter für das nte Objekt	—
1A03h	00h	3	RECORD	rw	—	Sende- PDO 4 Mappingparameter	—
	01-08h	3	UNSIGNED32	rw	—	PDO Mapping Parameter für das nte Objekt	—
2014h	00h	S	ARRAY	ro	—	Maske PDOs 37...40 Kanal 1	—
	01h	S	UNSIGNED32	rw	—	Maske (Byte 0..3)	—
	02h	S	UNSIGNED32	rw	—	Maske (Byte 4..7)	—
2015h	00h	S	ARRAY	ro	—	Maske PDOs 37...40 Kanal 2	—
	01h	S	UNSIGNED32	rw	—	Maske (Byte 0..3)	—
	02h	S	UNSIGNED32	rw	—	Maske (Byte 4..7)	—
2016h	00h	S	ARRAY	ro	—	Maske PDOs 37...40 Kanal 3	—
	01h	S	UNSIGNED32	rw	—	Maske (Byte 0..3)	—
	02h	S	UNSIGNED32	rw	—	Maske (Byte 4..7)	—
2017h	00h	S	ARRAY	ro	—	Maske PDOs 37...40 Kanal 4	—
	01h	S	UNSIGNED32	rw	—	Maske (Byte 0..3)	—
	02h	S	UNSIGNED32	rw	—	Maske (Byte 4..7)	—
2020h	00h	S	RECORD	ro	—	Lageregler	—
	01h	S	UNSIGNED8	rw	—	Achstyp	POSCNFG
	02h	S	INTEGER32	rw	—	In-Positionsfenster	PEINPOS
	03h	S	INTEGER32	rw	—	Schleppfehlerfenster	PEMAX
	04h	S	INTEGER32	rw	—	Positionsregister 1	SWE1
	05h	S	INTEGER32	rw	—	Positionsregister 2	SWE2
	06h	S	INTEGER32	rw	—	Positionsregister 3	SWE3
	07h	S	INTEGER32	rw	—	Positionsregister 4	SWE4
	08h	S	UNSIGNED32	rw	—	Auflösung Nenner	PGEARO
	09h	S	UNSIGNED32	rw	—	Auflösung Zähler	PGEAR
0Ah	S	UNSIGNED8	rw	—	Zählrichtung	DIR	

Index	Sub-index	DEF	Datenformat	Zugriff	PDO mappbar	Kurzbeschreibung	ASCII Objekt
2022h	00h	S	RECORD	ro	—	Positionierdaten für den Betriebsmode: Lage	—
	01h	S	INTEGER32	rww	<input checked="" type="checkbox"/>	Position (Fahrsatz 0)	O_P
	02h	S	UNSIGNED16	rww	<input checked="" type="checkbox"/>	Geschwindigkeit gewichtet s. ASCII VMUL (Fahrsatz 0, s.a. Objekt 2022h Subindex0Dh)	O_V
	03h	S	UNSIGNED16	rww	<input checked="" type="checkbox"/>	Fahrauftragsart (Fahrsatz 0)	O_C
	04h	S	INTEGER32	rww	<input checked="" type="checkbox"/>	Position für ext. Trajektorie	—
	05h	S	UNSIGNED16	rww	<input checked="" type="checkbox"/>	Fahrauftragsnummer	—
	06h	S	UNSIGNED16	rww	<input checked="" type="checkbox"/>	Anfahrzeit (Fahrsatz 0)	O_ACC1
	07h	S	UNSIGNED16	rww	<input checked="" type="checkbox"/>	Bremszeit (Fahrsatz 0)	O_DEC1
	08h	S	UNSIGNED16	rww	<input checked="" type="checkbox"/>	Ruckbegrenzung für Anfahrzeit (Fahrsatz 0)	O_ACC2
	09h	S	UNSIGNED16	rww	<input checked="" type="checkbox"/>	Ruckbegrenzung für Bremszeit (Fahrsatz 0)	O_DEC2
	0Ah	S	UNSIGNED16	rww	<input checked="" type="checkbox"/>	Nummer des Folgefahrauftrages	O_FN
	0Bh	S	UNSIGNED16	rww	<input checked="" type="checkbox"/>	Startverzögerung für Folgefahrauftrag	O_FT
	0Ch	S	2 x UNSIGNED16	wo	—	Kopieren eines Fahrauftrages	OCOPY
	0Dh	S	UNSIGNED16	rw	—	Wichtungsfaktor Geschwindigkeit (Objekt 2022h Subindex 02h)	VMUL
	0Eh	S	UNSIGNED32	rww	<input checked="" type="checkbox"/>	Geschwindigkeit (Fahrsatz 0)	O_V
0Fh	S	INTEGER32	rww	<input checked="" type="checkbox"/>	Sollposition CAN-Master-Slave	—	
2024h	00h	S	RECORD	ro	—	Einrichtbetrieb für den Mode „Lage“	—
	01h	S	UNSIGNED8	rw	—	Referenzfahrtart	NREF
	02h	S	UNSIGNED8	rw	—	Referenzfahrtrichtung	DREF
	03h	S	INTEGER32	rw	—	Geschwindigkeit Referenzfahrt	VREF
	04h	S	UNSIGNED16	rw	—	Beschleunigungsrampe [Tippen & Referenzieren]	ACCR
	05h	S	UNSIGNED16	rw	—	Bremsrampe [Tippen & Referenzieren]	DECR
	06h	S	INTEGER32	rw	—	Referenzoffset	RFFS
07h	S	INTEGER32	rw	—	Geschwindigkeit Tippbetrieb	VJOG	
2026h	00h	S	RECORD	ro	—	Sonderobjekte für den Mode Position	—
	01h	S	UNSIGNED8	rw	—	Aktivierung der internen Auswerteeinheit für das Speichern des Positionswertes	—
2030h	00h	S	ARRAY	ro	—	DP-Ram Variablen, nur schreibbar (PDO)	—
	01h	S	INTEGER32	rww	<input checked="" type="checkbox"/>	DP-Ram Variable 9	DPRVAR9
	02h	S	INTEGER32	rww	<input checked="" type="checkbox"/>	DP-Ram Variable 10	DPRVAR10
	03h	S	INTEGER32	rww	<input checked="" type="checkbox"/>	DP-Ram Variable 11	DPRVAR11
	04h	S	INTEGER32	rww	<input checked="" type="checkbox"/>	DP-Ram Variable 12	DPRVAR12
	05h	S	INTEGER32	rww	<input checked="" type="checkbox"/>	DP-Ram Variable 13	DPRVAR13
	06h	S	INTEGER32	rww	<input checked="" type="checkbox"/>	DP-Ram Variable 14	DPRVAR14
	07h	S	INTEGER32	rww	<input checked="" type="checkbox"/>	DP-Ram Variable 15	DPRVAR15
08h	S	INTEGER32	rww	<input checked="" type="checkbox"/>	DP-Ram Variable 16	DPRVAR16	
2031h	00h	S	RECORD	ro	—	Dummy-Variablen für Mappingzwecke	—
	01h	S	UNSIGNED8	wo	<input checked="" type="checkbox"/>	Write-only dummy 8 bit	—
	02h	S	UNSIGNED16	wo	<input checked="" type="checkbox"/>	Write-only dummy 16 bit	—
	03h	S	UNSIGNED24	wo	<input checked="" type="checkbox"/>	Write-only dummy 24 bit	—
2050h	04h	S	UNSIGNED32	wo	<input checked="" type="checkbox"/>	Write-only dummy 32 bit	—
	00h	S	ARRAY	ro	—	Hilfsvariablen der digitalen Eingänge	—
	01h	S	INTEGER32	rw	—	Triggervariable Eingang 1	IN1TRIG
	02h	S	INTEGER32	rw	—	Triggervariable Eingang 2	IN2TRIG
	03h	S	INTEGER32	rw	—	Triggervariable Eingang 3	IN3TRIG
	04h	S	INTEGER32	rw	—	Triggervariable Eingang 4	IN4TRIG
2051h	05h	S	INTEGER32	wo	<input checked="" type="checkbox"/>	Trigger-Hilfsvariable für elektronisches Getriebe	InxTRIG 51
	00h	S	ARRAY	ro	—	Konfiguration für Schwellenregister	—
	01h	S	UNSIGNED32	rw	—	Überwachung (deaktivieren / aktivieren)	WPOSE
	02h	S	UNSIGNED32	rw	—	Meldeart (umlaufend / einmalig)	WPOSX
	03h	S	UNSIGNED32	rw	—	Polarität für Positionsmeldung	WPOSP

Index	Sub-index	DEF	Datenformat	Zugriff	PDO mappbar	Kurzbeschreibung	ASCII Objekt
2052h	00h	S	ARRAY	ro	—	Positionsschwellenregister ABSOLUT (Aktualisierungszeit < 1ms)	POSRSTAT
	01h	S	INTEGER32	rww	<input checked="" type="checkbox"/>	Positionregister	P1
	02h	S	INTEGER32	rww	<input checked="" type="checkbox"/>	Positionregister	P2
	03h	S	INTEGER32	rww	<input checked="" type="checkbox"/>	Positionregister	P3
	04h	S	INTEGER32	rww	<input checked="" type="checkbox"/>	Positionregister	P4
	05h	S	INTEGER32	rww	<input checked="" type="checkbox"/>	Positionregister	P5
	06h	S	INTEGER32	rww	<input checked="" type="checkbox"/>	Positionregister	P6
	07h	S	INTEGER32	rww	<input checked="" type="checkbox"/>	Positionregister	P7
	08h	S	INTEGER32	rww	<input checked="" type="checkbox"/>	Positionregister	P8
	09h	S	INTEGER32	rww	<input checked="" type="checkbox"/>	Positionregister	P9
	0Ah	S	INTEGER32	rww	<input checked="" type="checkbox"/>	Positionregister	P10
	0Bh	S	INTEGER32	rww	<input checked="" type="checkbox"/>	Positionregister	P11
	0Ch	S	INTEGER32	rww	<input checked="" type="checkbox"/>	Positionregister	P12
	0Dh	S	INTEGER32	rww	<input checked="" type="checkbox"/>	Positionregister	P13
	0Eh	S	INTEGER32	rww	<input checked="" type="checkbox"/>	Positionregister	P14
	0Fh	S	INTEGER32	rww	<input checked="" type="checkbox"/>	Positionregister	P15
10h	S	INTEGER32	rww	<input checked="" type="checkbox"/>	Positionregister	P16	
2053h	00h	S	INTEGER8	ro	—	Positionsschwellenregister RELATIV (Aktualisierungszeit < 1ms) ASCII: s.a.	POSRSTAT
	01h	S	INTEGER32	rww	<input checked="" type="checkbox"/>	Positionregister	P1
	02h	S	INTEGER32	rww	<input checked="" type="checkbox"/>	Positionregister	P2
	03h	S	INTEGER32	rww	<input checked="" type="checkbox"/>	Positionregister	P3
	04h	S	INTEGER32	rww	<input checked="" type="checkbox"/>	Positionregister	P4
	05h	S	INTEGER32	rww	<input checked="" type="checkbox"/>	Positionregister	P5
	06h	S	INTEGER32	rww	<input checked="" type="checkbox"/>	Positionregister	P6
	07h	S	INTEGER32	rww	<input checked="" type="checkbox"/>	Positionregister	P7
	08h	S	INTEGER32	rww	<input checked="" type="checkbox"/>	Positionregister	P8
	09h	S	INTEGER32	rww	<input checked="" type="checkbox"/>	Positionregister	P9
	0Ah	S	INTEGER32	rww	<input checked="" type="checkbox"/>	Positionregister	P10
	0Bh	S	INTEGER32	rww	<input checked="" type="checkbox"/>	Positionregister	P11
	0Ch	S	INTEGER32	rww	<input checked="" type="checkbox"/>	Positionregister	P12
	0Dh	S	INTEGER32	rww	<input checked="" type="checkbox"/>	Positionregister	P13
	0Eh	S	INTEGER32	rww	<input checked="" type="checkbox"/>	Positionregister	P14
	0Fh	S	INTEGER32	rww	<input checked="" type="checkbox"/>	Positionregister	P15
10h	S	INTEGER32	rww	<input checked="" type="checkbox"/>	Positionregister	P16	
2060h	00h	S	INTEGER32	rw	—	Drehzahl oder Stromsollwert	—
2061h	00h	S	UNSIGNED16	wo	<input checked="" type="checkbox"/>	Strombegrenzung	DPRILIMIT

Index	Sub-index	DEF	Datenformat	Zugriff	PDO mappbar	Kurzbeschreibung	ASCII Objekt
2070h	00h	S	RECORD	ro	—	herstellerspez. Istwerte	—
	01h	S	INTEGER24	ro	☒	Istlage (Drehwinkel, Auflösung: 20 Bit / Umdrehung)	PRD
	02h	S	INTEGER24	ro	☒	Drehzahl in Inkr. (1[Inkr] = 1875/262144 [Umd/min])	—
	03h	S	INTEGER32	ro	☒	Inkr. Ist.-Position (Auflösung abhängig von PRBASE)	—
	04h	S	UNSIGNED16	ro	☒	gespeicherte Position (HW - LATCH positiv, 16 Bit / Umdrehung)	LATCH16
	05h	S	INTEGER32	ro	☒	gespeicherte Position (HW - LATCH positiv, 32 Bit / Umdrehung)	LATCH32
	06h	S	INTEGER32	ro	☒	Position in Abhängigkeit der Getriebefaktoren (PGEARI, PGEARO)	PFB
	07h	S	INTEGER32	ro	—	Geschwindigkeit in Abhängigkeit der Getriebefaktoren (PGEARI, PGEARO)	PV
	08h	S	INTEGER32	ro	☒	Schleppfehler in Abhängigkeit der Getriebefaktoren (PGEARI, PGEARO)	PE
	09h	S	UNSIGNED32	ro	—	eff. Strom	I
	0Ah	S	INTEGER32	ro	☒	Drehzahl 1/min	V
	0Bh	S	INTEGER32	ro	—	Kühlkörpertemperatur	TEMPH
	0Ch	S	INTEGER32	ro	—	Innentemperatur	TEMPE
	0Dh	S	INTEGER32	ro	—	Zwischenkreisspannung	VBUS
	0Eh	S	INTEGER32	ro	—	Bremsleistung	PBAL
	0Fh	S	INTEGER32	ro	—	I2t - Belastung	I2T
	10h	S	INTEGER32	ro	—	Betriebsdauer	TRUN
	11h	S	INTEGER32	ro	☒	erw. Status für TPDO 33	—
	12h	S	INTEGER32	ro	☒	Absolutwertgeberposition in SI-Einheiten	PFB0
13h	S	INTEGER32	ro	☒	Interner Positionssollwert	S_SETL (user-defined)	
14h	S	INTEGER16	ro	☒	Istwert analoger Eingang 1	ANIN1	
15h	S	INTEGER16	ro	☒	Istwert analoger Eingang 2	ANIN2	
16h	S	UNSIGNED32	ro	☒	Fehlerregister	ERRCODE	
17h	S	INTEGER32	ro	☒	Geschwindigkeitssollwert	PSPEED (user-defined)	
18h	S	UNSIGNED32	ro	☒	Mastergeschwindigkeit (elektronisches Getriebe)	MASTSPEED (user-defined)	
19h	S	INTEGER8	ro	☒	Steuervariable (elektronisches Getriebe)	ENGEDGE (user-defined)	
2071h	00h	S	RECORD	ro	—	Dummy-Variablen für Mappingzwecke	—
	01h	S	UNSIGNED8	ro	☒	Read-only dummy 8 bit	—
	02h	S	UNSIGNED16	ro	☒	Read-only dummy 16 bit	—
	03h	S	UNSIGNED24	ro	☒	Read-only dummy 24 bit	—
	04h	S	UNSIGNED32	ro	☒	Read-only dummy 32 bit	—
2080h	00h	S	ARRAY	ro	☒	Statusinformationen	—
	01h	S	UNSIGNED8	ro	☒	Input/Output/Latch	—
	02h	S	UNSIGNED8	ro	☒	Sammelinfos (Warning/Error/Reserve)	—
	03h	S	UNSIGNED8	ro	☒	DRVSTAT LSB	DRVSTAT
	04h	S	UNSIGNED8	ro	☒	“	DRVSTAT
	05h	S	UNSIGNED8	ro	☒	“	DRVSTAT
	06h	S	UNSIGNED8	ro	☒	DRVSTAT MSB	DRVSTAT
	07h	S	UNSIGNED8	ro	☒	TRJSTAT LSB	TRJSTAT
	08h	S	UNSIGNED8	ro	☒	“	TRJSTAT
	09h	S	UNSIGNED8	ro	☒	“	TRJSTAT
0Ah	S	UNSIGNED8	ro	☒	TRJSTAT MSB	TRJSTAT	
2081h	00h	S	INTEGER8	ro	—	Inkr Ist. - Position (byteweise/24 Bit)	—
	01h	S	INTEGER8	ro	☒	Inkr. Pos LSB	—
	02h	S	INTEGER8	ro	☒	IstIncrpos	—
	03h	S	INTEGER8	ro	☒	IstIncrpos	—
	04h	S	INTEGER8	ro	☒	Inkrementelle Position MSB	—
	05h	S	INTEGER24	ro	☒	Inkrementelle Position 24 Bit	—
2082h	00h	S	INTEGER8	ro	—	Latch 32/24 Bit positiv Latch	—
	01h	S	INTEGER32	ro	☒	Latch 32Bit	LATCH32
	02h	S	INTEGER24	ro	☒	Latch 24Bit	—

Index	Sub-index	DEF	Datenformat	Zu-griff	PDO mappbar	Kurzbeschreibung	ASCII Objekt
2083h	00h	S	INTEGER8	ro	—	Latch 32/24 Bit negativ Latch	—
	01h	S	INTEGER32	ro	☒	Latch 32Bit	LATCH32N
	02h	S	INTEGER24	ro	☒	Latch 24Bit	—
2084h	00h	S	INTEGER8	ro	—	Latch 16 Bit positiv Latch	—
	01h	S	INTEGER16	ro	☒	Latch 16 Bit	LATCH16
2085h	00h	S	INTEGER8	ro	—	Latch 16 Bit negativ Latch	—
	01h	S	INTEGER16	ro	☒	Latch 16 Bit	LATCH16N
2086h	00h	S	UNSIGNED16	ro	☒	Positionstriggerwort	POSRSTAT
2087h	00h	S	RECORD	ro	—	Latch-Positionen Eingang 1	—
	01h	S	INTEGER32	ro	☒	Latchposition Eingang 1, positive Flanke	LATCHX32
	02h	S	INTEGER32	ro	☒	Latchposition Eingang 1, negative Flanke	LATCHX32N
	03h	S	UNSIGNED8	rw	☒	Latchfreigaben für Eingänge 1 und 2	—
2090h	00h	S	ARRAY	ro	☒	DP-Ram Variablen, nur schreibbar (PDO)	-
	01h	S	INTEGER32	ro	☒	DP-Ram Variable 1	DPRVAR1
	02h	S	INTEGER32	ro	☒	DP-Ram Variable 2	DPRVAR2
	03h	S	INTEGER32	ro	☒	DP-Ram Variable 3	DPRVAR3
	04h	S	INTEGER32	ro	☒	DP-Ram Variable 4	DPRVAR4
	05h	S	INTEGER32	ro	☒	DP-Ram Variable 5	DPRVAR5
	06h	S	INTEGER32	ro	☒	DP-Ram Variable 6	DPRVAR6
	07h	S	INTEGER32	ro	☒	DP-Ram Variable 7	DPRVAR7
08h	S	INTEGER32	ro	☒	DP-Ram Variable 8	DPRVAR8	
2600h	00h	S	INTEGER8	rw	—	1 <sup>st</sup> Receive PDO select	—
2601h	00h	S	INTEGER8	rw	—	2 <sup>nd</sup> Receive PDO select	—
2602h	00h	S	INTEGER8	rw	—	3 <sup>rd</sup> Receive PDO select	—
2603h	00h	S	INTEGER8	rw	—	4 <sup>th</sup> Receive PDO select	—
2721h	00h	S	INTEGER8	rw	—	Konfiguration Receive PDO 33	—
2A00h	00h	S	INTEGER8	rw	—	1 <sup>st</sup> Transmit PDO select	—
2A01h	00h	S	INTEGER8	rw	—	2 <sup>nd</sup> Transmit PDO select	—
2A02h	00h	S	INTEGER8	rw	—	3 <sup>rd</sup> Transmit PDO select	—
2A03h	00h	S	INTEGER8	rw	—	4 <sup>th</sup> Transmit PDO select	—
3100h	00h	S	Visible String	rw	—	ASCII-Zeichen Richtung	—
3500h	00h	S	RECORD	ro	—	Beginn des Objektkanals	—
	01h	S	UNSIGNED16	ro	—	Gesamtanzahl der Objekte im Objektkanal	MAXSDO
	02h	S	UNSIGNED16	ro	—	unterer Grenzwert	—
	03h	S	UNSIGNED16	ro	—	oberer Grenzwert	—
	04h	S	UNSIGNED16	ro	—	Defaultwert	—
	05h	S	UNSIGNED8	ro	—	Datenformat des Objektes	—
	06h	S	UNSIGNED32	ro	—	Kontrolldaten	—
	07h	S	—	—	—	reserviert	—
08h	S	—	—	—	reserviert	—	
3501h	00h	S	RECORD	ro	—	Beschleunigungsrampe Drehzahlregelung	—
	01h	S	UNSIGNED16	ro	—	Wert	ACC
	02h	S	UNSIGNED16	ro	—	unterer Grenzwert	—
	03h	S	UNSIGNED16	ro	—	oberer Grenzwert	—
	04h	S	UNSIGNED16	ro	—	Defaultwert	—
	05h	S	UNSIGNED8	ro	—	Datenformat des Objektes	—
	06h	S	UNSIGNED32	ro	—	Kontrolldaten	—
	07h	S	—	—	—	reserviert	—
08h	S	—	—	—	reserviert	—	
6040h	00h	4	INTEGER16	wo	☒	DS402 controlword	—
6041h	00h	4	INTEGER16	ro	☒	DS402 Status	—
605Ah		4	INTEGER16	rw	—	Quickstop option code	—
6060h	00h	4	INTEGER8	wo	☒	Mode of Operation	—
6061h	00h	4	INTEGER8	ro	☒	Display Mode of Operation	—
6063h	00h	4	INTEGER32	ro	☒	Inkr. Istposition	—
6064h	00h	4	INTEGER32	ro	☒	Istposition (unter Berücksichtigung der Getriebefaktoren)	—
606Ch	00h	4	INTEGER32	ro	☒	Istdrehzahl (Mode: pv)	—
6077h	00h	4	INTEGER16	ro	☒	Drehmoment-Istwert	IQ
607Ah	00h	4	INTEGER32	rww	☒	Zielposition (Mode: pp)	—

Index	Sub-index	DEF	Datenformat	Zu-griff	PDO mappbar	Kurzbeschreibung	ASCII Objekt
607B <sub>h</sub>	00 <sub>h</sub>	4	ARRAY	ro	—	position_range_limit	—
	01 <sub>h</sub>	4	INTEGER32	rw	—	min_position_range_limit	—
	02 <sub>h</sub>	4	INTEGER32	rw	—	max_position_range_limit	—
607C <sub>h</sub>	00 <sub>h</sub>	4	INTEGER32	rww	<input checked="" type="checkbox"/>	Referenzoffset	—
6081 <sub>h</sub>	00 <sub>h</sub>	4	UNSIGNED32	rww	<input checked="" type="checkbox"/>	Geschwindigkeit (Mode: pp)	—
6083 <sub>h</sub>	00 <sub>h</sub>	4	UNSIGNED32	rww	<input checked="" type="checkbox"/>	Beschleunigung (Mode: pp pv)	—
6084 <sub>h</sub>	00 <sub>h</sub>	4	UNSIGNED32	rww	<input checked="" type="checkbox"/>	Bremsbeschleunigung (Mode: pp pv)	—
6086 <sub>h</sub>	00 <sub>h</sub>	4	INTEGER16	rww	<input checked="" type="checkbox"/>	Art der Fahrsatzrampen (motion_profil_typ)	—
6089 <sub>h</sub>	00 <sub>h</sub>	4	INTEGER8	rw	—	Notationsindex Position (wird nicht unterstützt)	—
608A <sub>h</sub>	00 <sub>h</sub>	4	UNSIGNED8	rw	—	Dimensionsindex Position (wird nicht unterstützt)	—
608B <sub>h</sub>	00 <sub>h</sub>	4	INTEGER8	rw	—	Notationsindex Drehzahl (wird nicht unterstützt)	—
608C <sub>h</sub>	00 <sub>h</sub>	4	UNSIGNED8	rw	—	Dimensionsindex Drehzahl (wird nicht unterstützt)	—
608D <sub>h</sub>	00 <sub>h</sub>	4	INTEGER8	rw	—	Notationsindex Beschleunigung (wird nicht unterstützt)	—
608E <sub>h</sub>	00 <sub>h</sub>	4	UNSIGNED8	rw	—	Dimensionsindex Beschleunigung (wird nicht unterstützt)	—
6093 <sub>h</sub>	00 <sub>h</sub>	4	UNSIGNED32	rw	—	Positionsfaktor	—
	01 <sub>h</sub>	4	UNSIGNED32	rww	<input checked="" type="checkbox"/>	Numerator	PGEARO
	02 <sub>h</sub>	4	UNSIGNED32	rww	<input checked="" type="checkbox"/>	Vorschubkonstante (feed_constant)	PGEARI
6094 <sub>h</sub>	00 <sub>h</sub>	4	UNSIGNED32	rw	—	Geschwindigkeitsfaktor Encoder	—
	01 <sub>h</sub>	4	UNSIGNED32	rw	—	Numerator	—
	02 <sub>h</sub>	4	UNSIGNED32	rw	—	Divisor	—
6097 <sub>h</sub>	00 <sub>h</sub>	4	UNSIGNED32	rw	—	Beschleunigungsfaktor	—
	01 <sub>h</sub>	4	UNSIGNED32	rw	—	Divisor	—
	02 <sub>h</sub>	4	UNSIGNED32	rw	—	Beschleunigungsfaktor	—
6098 <sub>h</sub>	00 <sub>h</sub>	4	INTEGER8	rww	<input checked="" type="checkbox"/>	Referenzfahrtart	—
6099 <sub>h</sub>	00 <sub>h</sub>	4	ARRAY	ro	—	Referenzfahrtgeschwindigkeit	—
	01 <sub>h</sub>	4	INTEGER32	rw	<input checked="" type="checkbox"/>	Referenzfahrtgeschwindigkeit (bei Suche nach Referenzschalter )	VREF
	02 <sub>h</sub>	4	INTEGER32	ro	<input checked="" type="checkbox"/>	Referenzfahrtgeschwindigkeit (bei Suche nach Resolvernulldpunkt)	VREF
609A <sub>h</sub>	00 <sub>h</sub>	4	UNSIGNED32	rw	<input checked="" type="checkbox"/>	Referenzfahrtbeschleunigung	—
60C2 <sub>h</sub>	00 <sub>h</sub>	4	RECORD	ro	—	Interpolation Zeitspanne	PTBASE
	01 <sub>h</sub>	4	UNSIGNED8	rw	—	Interpolation Zeiteinheiten	-
	02 <sub>h</sub>	4	INTEGER16	rw	—	Interpolation Zeitindex	-
60FF <sub>h</sub>	00 <sub>h</sub>	4	INTEGER32	rw	<input checked="" type="checkbox"/>	Drehzahlsollwert	—

## 6.4 Neukonfiguration des SERVOSTAR 400/600

Im SERVOSTAR 400/600 gibt es eine Anzahl von Parametern, die erst in der Initialisierungsphase beim Einschalten des Geräts in das laufende Firmware-Programm integriert werden und dann ihre Wirkung entfalten. Daher erfordert eine Änderung dieser Parameter ein Speichern aller Parameter und einen anschließenden Neustart des Geräts. Dies kann über den CAN-Bus durch folgende SDOs ausgelöst werden:

1. Schreiben des Objekt 35E<sub>h</sub> Subindex 01<sub>h</sub>, Wert 0, Speichern - Kommando
2. Schreiben des Objekt 3632<sub>h</sub> Subindex 01<sub>h</sub>, Wert 0, Neustart des Geräts

## 6.5 Stichwortverzeichnis

<b>I</b>	1000h . . . . .	29	6077h . . . . .	87
	1001h . . . . .	29	607Ah . . . . .	98
	1002h . . . . .	30	607Bh . . . . .	99
	1003h . . . . .	31	607Ch . . . . .	94
	1004h . . . . .	32	6081h . . . . .	99
	1005h . . . . .	33	6083h . . . . .	100
	1006h . . . . .	33	6084h . . . . .	100
	1007h . . . . .	34	6086h . . . . .	101
	1008h . . . . .	34	608Bh . . . . .	58
	100Ah . . . . .	34	608Ch . . . . .	58
	100Bh . . . . .	35	6093h . . . . .	59
	100Ch . . . . .	35	6094h . . . . .	60
	100Dh . . . . .	35	6097h . . . . .	62
	100Eh . . . . .	36	6098h . . . . .	95
	100Fh . . . . .	36	6099h . . . . .	96
	1010h . . . . .	37	609Ah . . . . .	96
	1012h . . . . .	38	60C2h . . . . .	87
	1013h . . . . .	38	60FFh . . . . .	92
	1014h . . . . .	38	<b>A</b> Abkürzungen . . . . .	8
	1018h . . . . .	39	Abschlusswiderstand . . . . .	11
	1400-1403h . . . . .	45	Ansprechüberwachung . . . . .	35
	1600-1603h . . . . .	45	Ansprechüberwachung quittieren . .	54
	1800-1803h . . . . .	49	Antriebsprofil . . . . .	27
	1A00-1A03h . . . . .	49	<b>B</b> Basisdatentypen . . . . .	17
	2014-2017h . . . . .	50	Baudrate . . . . .	10
	2020h . . . . .	66	Beispiele für die Inbetriebnahme .	115
	2022h . . . . .	69	Bestimmungsgemäße Verwendung .	7
	2024h . . . . .	64	Busleitung . . . . .	11
	2026h . . . . .	76	<b>C</b> COB-ID . . . . .	16
	2030h . . . . .	88	Controlword . . . . .	53
	2031h . . . . .	89	<b>D</b> Data Frame . . . . .	15
	2050h . . . . .	75	Datentransferfunktionen . . . . .	9
	2060h . . . . .	63	Datentypen . . . . .	16
	2061h . . . . .	63	Digitaler Sollwert . . . . .	63
	2070h . . . . .	80	<b>E</b> Einrichtfunktionen . . . . .	9
	2071h . . . . .	90	Emergency Message . . . . .	27
	2082h . . . . .	76	Emergency Object . . . . .	19
	2083h . . . . .	77	Empfangs-PDOs . . . . .	41
	2084h . . . . .	77	Erweiterte Datentypen . . . . .	18
	2085h . . . . .	78	<b>F</b> Factor Groups . . . . .	57
	2087h . . . . .	79	<b>G</b> Gerätesteuerung . . . . .	50
	2090h . . . . .	88	Grundfunktionalitäten . . . . .	9
	2600-2603h . . . . .	45	<b>H</b> Homing Mode . . . . .	94
	2721h . . . . .	45	<b>I</b> Inbetriebnahme . . . . .	12
	2A00-2A03h . . . . .	49	Inhibit time . . . . .	49
	3500h... . . . .	105	Installation . . . . .	10
	6040h . . . . .	53	Istwerte . . . . .	80
	6041h . . . . .	54	<b>K</b> Kommunikationsobjekte . . . . .	18
	6060h . . . . .	56	Kommunikationsprofil . . . . .	15
	6061h . . . . .	57	Konfigurationsparameter . . . . .	13
	6063h . . . . .	93		
	6064h . . . . .	93		
	606Ch . . . . .	91		

<b>L</b>	Latchfunktion . . . . .	76
	Leitungslänge . . . . .	11
<b>M</b>	Mapping. . . . .	40
<b>N</b>	Network Management Object. . . . .	19
	Neukonfiguration . . . . .	135
	Nodeguard . . . . .	25
<b>O</b>	Objektkanal . . . . .	105
	Objektverzeichnis . . . . .	129
	Operationsmodus . . . . .	56
<b>P</b>	Position Control Function . . . . .	92
	Positionierfunktionen . . . . .	9
	Process Data Object . . . . .	23
	Profile Position Mode . . . . .	97
	Profile Velocity Mode . . . . .	91
<b>R</b>	Remote Frame . . . . .	15
<b>S</b>	Schleppfehler quittieren. . . . .	54
	SDO abort codes . . . . .	23
	Sende-PDOs . . . . .	46
	Service Data Object. . . . .	21
	Statusword . . . . .	54
	Symbole . . . . .	8
	Synchronisation Object . . . . .	19
	Systemvoraussetzungen . . . . .	9
<b>T</b>	Time Stamp Object . . . . .	19
	Triggermodus . . . . .	25
<b>U</b>	Übertragungsgeschwindigkeit . . . . .	9
	Übertragungsmodus . . . . .	24
	Übertragungsverfahren. . . . .	9
<b>W</b>	Weiterführende Dokumentation . . . . .	7
<b>Z</b>	Zusammengesetzte Datentypen . . . . .	17
	Zustandsmaschine . . . . .	51

## Vertrieb und Service

Wir bieten Ihnen einen kompetenten und schnellen Service. Wählen Sie die zuständige regionale Vertriebsniederlassung in Deutschland oder kontaktieren Sie den europäischen oder nordamerikanischen Kundenservice.

### Deutschland

#### Vertriebsgebiet **NORD**

Danaher Motion GmbH

Iris Tolusch

Wacholderstr. 40-42

D-40489 Düsseldorf

E-Mail [iris.tolusch@danahermotion.com](mailto:iris.tolusch@danahermotion.com)

Tel.: +49(0)203 - 9979 - 214

Fax: +49(0)203 - 9979 - 3214

#### Vertriebsgebiet **SÜDWEST**

Danaher Motion GmbH

Kerstin Müller

Brückenfeldstr. 26/1

D-75015 Bretten

E-Mail: [kerstin.mueller@danahermotion.com](mailto:kerstin.mueller@danahermotion.com)

Tel.: +49(0)7252 - 96462 - 10

Fax: +49(0)7252 - 96462 - 69

#### Vertriebsgebiet **SÜDOST**

Danaher Motion GmbH

Ursula Koschak

Kiesgräble 7

D-89129 Langenau

E-Mail [ursula.koschak@danahermotion.com](mailto:ursula.koschak@danahermotion.com)

Tel.: +49(0)7471 - 6223 - 23

Fax: +49(0)7471 - 6223 - 26

### Europa

Danaher Motion Kundenservice Europa

Internet [www.DanaherMotion.net](http://www.DanaherMotion.net)

E-Mail [support\\_dus.germany@danahermotion.com](mailto:support_dus.germany@danahermotion.com)

Tel.: +49(0)203 - 99 79 - 0

Fax: +49(0)203 - 99 79 - 216

### Nordamerika

Danaher Motion Customer Support North America

Internet [www.DanaherMotion.com](http://www.DanaherMotion.com)

E-Mail [DMAC@danahermotion.com](mailto:DMAC@danahermotion.com)

Tel.: +1 - 540 - 633 - 3400

Fax: +1 - 540 - 639 - 4162

