



# DME 230 / 400

## CAN Open Schnittstelle

der Servoregler mit der  
Typenbezeichnung

- BN6773
- BN6783

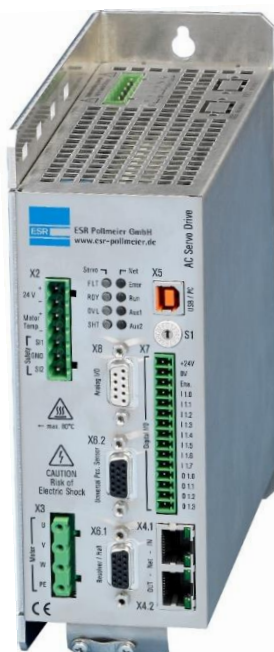
Typ:

- DME 230x4-CO
- DME 230x4-I/O
- DME 400x8-CO

Part No:

- 81703.00100
- 81703.00101
- 81703.00110

Publikation Ref: 160616



Dunkermotoren GmbH | Allmendstraße 11 | D-79848 Bonndorf/ Schwarzwald  
Phone +49 (0) 7703 930-0 | Fax +49 (0) 7703 930-210/ 212 | [info@dunkermotoren.com](mailto:info@dunkermotoren.com)

## **TrioDrive D/CS / MidiDrive D/CS**

**Digitale Servoregler  
für direkten Netzanschluss**

## **CANopen-Schnittstelle**

**Betriebsanleitung 6745.105, V 1.0**

Diese Betriebsanleitung gilt für

- TrioDrive-D/CS-Servoregler, Kompaktbauweise BN 6756 bis BN 6758 mit Einbau-Netzgerät für einphasigen Anschluss an Wechselspannung und integrierter Sicherheitstechnik
- MidiDrive-D/CS-Servoregler, Kompaktbauweise BN 6745 bis BN 6749 mit Einbau-Netzgerät für Drehstromanschluss und integrierter Sicherheitstechnik

Diese Betriebsanleitung gilt zusammen mit

- Betriebsanleitung 6710.101 (Funktionen und Parameter)
- Betriebsanleitung 6755.102 bzw. 6745.102 (Anschluss und Inbetriebnahme)
- Betriebsanleitung 6710.107 (SPP Windows Bedien- und Inbetriebnahmeprogramm)

ESR Pollmeier GmbH  
Lindenstraße 20  
64372 Ober-Ramstadt  
Bundesrepublik Deutschland

Tel. +49 6167 9306-0  
Fax +49 6167 9306-77

E-Mail [info@esr-pollmeier.de](mailto:info@esr-pollmeier.de)  
[www.esr-pollmeier.de](http://www.esr-pollmeier.de)

### Versionen des Textes

2006-04-07	V 0.1, KS/Ri	neu erstellt für TrioDrive D/CS und MidiDrive D/CS auf der Basis von 6710.105
2009-05-07	V 1.0, KS/Ri	Pin-Nummern Abschlusswiderstand korrigiert; Normen angepasst; Variablen hinzugefügt

O:\!DB\6\7\4\6745\_105\_10.wpd

Copyright by ESR Pollmeier GmbH, 64372 Ober-Ramstadt, Germany

Alle Rechte, auch die der Übersetzung, vorbehalten. Ohne vorherige ausdrückliche schriftliche Genehmigung der ESR Pollmeier GmbH darf kein Teil dieser Betriebsanleitung vervielfältigt, reproduziert, in einem Informationssystem gespeichert oder verarbeitet oder in anderer Form weiter übertragen werden.

Diese Betriebsanleitung wurde mit Sorgfalt erstellt. ESR Pollmeier GmbH übernimmt jedoch für eventuelle Irrtümer in dieser Betriebsanleitung und deren Folgen keine Haftung. Ebenso wird keine Haftung für direkte Schäden oder Folgeschäden übernommen, die sich aus dem Missbrauch des Gerätes ergeben.

Bei der Anwendung der Geräte sind die einschlägigen Vorschriften bezüglich Sicherheitstechnik und Funkentstörung zu beachten.

Änderungen vorbehalten.

# Inhalt

Beachten Sie auch das *Stichwortverzeichnis* ab Seite 66.

<b>1</b>	<b>Vorbemerkungen</b>	6
1.1	Zu dieser Beschreibung	6
1.2	Funktionsbausteine	8
<b>2</b>	<b>Sicherheitshinweise</b>	9
2.1	Art der Hinweise	9
<b>3</b>	<b>Technische Daten</b>	10
<b>4</b>	<b>CANopen-Einführung</b>	11
4.1	Begriffe und Abkürzungen	11
<b>5</b>	<b>Anschluss und Inbetriebnahme</b>	15
5.1	Busanschluss	15
5.1.1	Anschlussstecker und Terminierung	16
5.1.2	Busleitungen	16
5.2	Leuchtdioden	16
5.3	Inbetriebnahme	17
5.3.1	Projektierung der CANopen-Schnittstelle	17
5.3.2	Parametrierung der Antriebsfunktionen (Servoregler, -motor)	17
5.3.3	Inbetriebnahme des CANopen-Systems (Master, weitere Geräte)	17
<b>6</b>	<b>Netzwerkmanagement und COB-ID-Verteilung</b>	18
6.1	NMT Node State Diagram	18
6.2	COB-ID-Verteilung	20
<b>7</b>	<b>SDO-Kommunikation (Parameter)</b>	21
<b>8</b>	<b>PDO-Kommunikation (Prozessdaten)</b>	22
<b>9</b>	<b>Synchronisation</b>	25
9.1	Asynchrone Übertragung von PDOs	25
9.2	Synchrone Übertragung von PDOs	25
<b>10</b>	<b>Interpolated Position Mode</b>	26
10.1	Interpolated Position Mode aktivieren	27
10.2	Interpolated Position Mode deaktivieren	28
<b>11</b>	<b>Überwachungsmechanismen</b>	29
11.1	Störungsbehandlung (Error, Emergency)	29
11.2	Node Guarding, Life Guarding	29
11.3	Heartbeat	30

11.4	Sende- und Empfangsüberwachung .....	30
<b>12</b>	<b>Speicherfunktionen .....</b>	<b>31</b>
12.1	Speicherfunktionen Programm SPP Windows .....	31
<b>13</b>	<b>Geräteinformationen .....</b>	<b>31</b>
<b>14</b>	<b>Variablen-Beschreibungen .....</b>	<b>32</b>
14.1	Variablen-Beschreibung Netzwerk-Management .....	35
14.2	Variablen-Beschreibung SDO-Kommunikation .....	36
14.3	Variablen-Beschreibung PDO-Kommunikation .....	37
14.3.1	Abbildungs-PDOs (PDO Mapping) .....	37
14.3.1.1	Receive PDOs Mapping .....	37
14.3.1.2	Transmit PDOs Mapping .....	38
14.3.2	Maschinendaten PDOs (Comm. Parameter) .....	40
14.3.2.1	Receive PDOs Comm. Parameter .....	40
14.3.2.2	Transmit PDOs Comm. Parameter .....	42
14.4	Variablen-Beschreibung Synchronisation .....	45
14.4.1	Maschinendaten Synchronisation .....	45
14.5	Variablen-Beschreibung Interpolated Position Mode .....	45
14.5.1	Prozessdaten-Abbildung Interpolated Position Mode .....	45
14.5.2	Maschinendaten Interpolated Position Mode .....	47
14.6	Variablen-Beschreibung Überwachung .....	48
14.6.1	Status Störungsbehandlung (Error, Emergency) .....	48
14.6.2	Maschinendaten Störungsbehandlung (Emergency) .....	49
14.6.3	Maschinendaten Node Guarding, Life Guarding .....	50
14.7	Variablen-Beschreibungen Speicherfunktionen .....	51
14.7.1	Steuerung Speicherfunktionen .....	51
14.8	Variablen-Beschreibung Geräteinformationen .....	52

## Anhang

<b>Anhang A</b>	<b>Zustandsmaschinen .....</b>	<b>55</b>
<b>Anhang B</b>	<b>Variablenliste .....</b>	<b>57</b>
<b>Anhang C</b>	<b>Projektierungshilfe .....</b>	<b>59</b>
<b>Anhang D</b>	<b>Störungscodes .....</b>	<b>64</b>
<b>Anhang E</b>	<b>Stichwortverzeichnis .....</b>	<b>66</b>

# Abbildungen

Bild 1: NMT Node State Diagram Servoregler .....	18
Bild 2: Elemente von Zustandsmaschinen .....	55

Hinweis: Die in dieser Betriebsanleitung verwendeten Soft- und Hardware-Bezeichnungen und Markennamen der jeweiligen Firmen unterliegen im Allgemeinen warenzeichen-, marken- oder patentrechtlichem Schutz.

# 1 Vorbemerkungen

## 1.1 Zu dieser Beschreibung

Diese Betriebsanleitung 6745.105 erläutert die CANopen-Schnittstelle der digitalen Servoregler TrioDrive D/CS (BN 6756 bis BN 6758) und MidiDrive D/CS (BN 6745 bis BN 6749). Sie befasst sich speziell mit dem Zugriff auf die Funktionen der Servoregler über die Kommunikationsschnittstelle CANopen.

Üblicherweise wird die CANopen-Schnittstelle zum Anschluss der Servoregler an eine übergeordnete Steuerung verwendet. Zur Integration der Servoantriebe in Automatisierungssysteme sind Funktionsbausteine erhältlich, die in Abschnitt 1.2 (Seite 8) näher beschrieben werden.

In Kombination mit einem Bedienterminal oder mit I/O-Erweiterungen können die Servoantriebe dank des integrierten Teileprogramms (optional) auch in Anwendungen eingesetzt werden, bei denen keine übergeordnete Steuerung erforderlich ist. Für nähere Informationen hierzu wenden Sie sich bitte an ESR.

Die vorliegende Betriebsanleitung 6745.105 gilt zusammen mit der

- Betriebsanleitung „Anschluss und Inbetriebnahme“ des Servoreglers (gehört zum Lieferumfang des Servoreglers)
  - Betriebsanleitung 6755.102 (TrioDrive D/CS) bzw.
  - Betriebsanleitung 6745.102 (MidiDrive D/CS)
- Betriebsanleitung „Funktionen und Parameter“ des Servoreglers (gehört zum Lieferumfang des Servoreglers)
  - Betriebsanleitung 6710.101
- Betriebsanleitung „SPP Windows Bedien- und Inbetriebnahmeprogramm“ (wird mit dem optionalen Bedien- und Inbetriebnahmeprogramm SPP Windows ausgeliefert)
  - Betriebsanleitung 6710.107

Bei der Arbeit mit der CANopen-Schnittstelle werden zusätzlich folgende Dokumente benötigt:

- CANopen Communication Profile for Industrial Systems CiA 301 (kurz: CANopen-Kommunikationsprofil CiA 301 oder CiA 301)

Diese Betriebsanleitung basiert auf dem CANopen-Kommunikationsprofil CiA 301, V 4.0.2 vom 13. Februar 2002. Für den Einsatz einer neueren Version wenden Sie sich bitte an ESR.

- Beschreibung(en) der in der Steuerung eingesetzten CAN-Anschaltbaugruppe und der dort verwendeten Software.

Mit der CANopen-Schnittstelle sind Teile des CANopen Device Profile Drives and Motion Control CiA 402 implementiert (kurz: CANopen-Antriebsprofil CiA 402 oder CiA 402, genormt als IEC 61800-7-201 und IEC 61800-7-301). Die

implementierten Teile von CiA 402 sind vollständig in dieser Betriebsanleitung beschrieben, sodass dieses Dokument für die Arbeit mit den Servoreglern mit CANopen-Schnittstelle nicht benötigt wird.

Über die CANopen-Schnittstelle wird auf die Funktionen der Servoregler mittels so genannter Variablen zugegriffen. Für die Zwecke dieser Betriebsanleitung ist es dabei sinnvoll, zwischen zwei Arten von Variablen zu unterscheiden, den Antriebsvariablen und den CANopen-Variablen:

- Antriebsvariablen



Auf die Antriebsfunktionen des Servoreglers wird unabhängig von der Kommunikationsschnittstelle (serielle Schnittstelle, CANopen- oder eine andere Feldbus-Schnittstelle) über Antriebsvariablen zugegriffen (z. B. *Achsen-Betriebsart*, *Lageziel*, *Geschwindigkeits-Istwert*). Die Antriebsfunktionen und Antriebsvariablen der Servoregler sind in der Betriebsanleitung 6710.101 „Funktionen und Parameter“ beschrieben. Der Abschnitt „Variablen-Beschreibungen“ und der Anhang „Variablenliste“ in jener Gerätebeschreibung enthalten alle Informationen über die Variablen, die für den Zugriff über eine der Kommunikationsschnittstellen benötigt werden.



Diese Antriebsfunktionen und Antriebsvariablen der Servoregler können Sie bereits erproben, ohne ein Programm für Ihren Rechner oder Ihre Steuerung zu schreiben oder den CAN-Bus in Betrieb zu nehmen. Verwenden Sie dazu einen PC mit dem Bedien- und Inbetriebnahmeprogramm SPP Windows. Erst wenn Sie die Funktionen der Servoregler mit Hilfe des PC kennen gelernt haben und wissen, über welchen Variablenzugriff Sie welche Funktion ausführen oder abfragen können, sollten Sie Programme schreiben, die über die CANopen-Schnittstelle auf diese Variablen zugreifen.

- CANopen-Variablen

Zusätzlich zu den o. a. Antriebsvariablen werden weitere Variablen zur Verwaltung der CANopen-Schnittstelle (CANopen-Variablen) verwendet (z. B. *Node-ID*, *Guard Time*). Die CANopen-Variablen der Servoregler sind in dieser Betriebsanleitung beschrieben. Abschnitt 14 (Variablen-Beschreibungen) und Anhang B (Variablenliste) dieser Betriebsanleitung enthalten alle Informationen über die Variablen, die für den Zugriff über die CANopen-Schnittstelle benötigt werden.



Bevor entsprechend dieser Betriebsanleitung über die CANopen-Schnittstelle auf die Servoregler zugegriffen wird, sollte der Servoantrieb (Servoregler und Servomotor) in Betrieb genommen sein. Für die Inbetriebnahme der Servoantriebe wird ein PC mit dem Bedien- und Inbetriebnahmeprogramm SPP Windows benötigt. Bitte prüfen Sie, ob diese Voraussetzungen erfüllt sind.



## 1.2 Funktionsbausteine

Für eine einfache Integration der Servoantriebe in Automatisierungssysteme sind Funktionsbausteine erhältlich.

Diese sind verfügbar für verschiedene Steuerungen nach IEC 61131-3. Bei Interesse wenden Sie sich bitte an ESR.

Die Kommunikation erfolgt als PDO/SDO-Kommunikation über CANopen.

Unterstützte Funktionen:

- Parametrierung der Servoantriebe durch die Steuerung (z. B. nach Einschalten)
- Auslösen von Bewegungen (relativ/absolut positionieren, Referenzfahrt, Geschwindigkeitsvorgabe ...)
- Beeinflussung der im Antrieb integrierten Positioniersteuerung (Teileprogramm)
- Ein- und Ausgabe von Binärsignalen (Software-Ein-/Ausgänge)
- Beispielprogramme zur Benutzung der Funktionsbibliothek als Ausgangsbasis für die Entwicklung eigener Programme

Die Funktionsbausteine orientieren sich an der PLCopen-Spezifikation „Function blocks for motion control“, die wiederum auf IEC 61131-3 basiert.

Weitere Informationen finden Sie im Datenblatt 6710.160.

Die Funktionsbausteine übernehmen viele der in dieser Betriebsanleitung beschriebenen Funktionen. Welche Schritte zur Parametrierung der CANopen-Schnittstelle Sie selbst durchführen müssen, ist in der Betriebsanleitung zu den Funktionsbausteinen beschrieben.

## 2 Sicherheitshinweise

Beachten Sie unbedingt die Sicherheitshinweise in der jeweiligen Betriebsanleitung „Anschluss und Inbetriebnahme“ (6755.102 bzw. 6745.102) sowie die Warnungen und Hinweise in den Randspalten aller Betriebsanleitungen.



Der Zugriff auf die Servoregler über die CANopen-Schnittstelle kann Antriebsbewegungen auslösen. Wenn der Antrieb und/oder die Maschine nicht vorschriftsmäßig aufgebaut und gesichert sind, können dabei Gesundheit und Leben von Personen gefährdet werden.



Der Zugriff über die CANopen-Schnittstelle ist deshalb solange untersagt, bis die Anforderungen der Maschinenrichtlinie erfüllt sind.



Beim Einsatz von Bussystemen besteht allgemein die Gefahr einer nicht sichtbaren Beeinflussung eines Busteilnehmers von außen. Dies kann zu unerwartetem (nicht kontrolliertem) Systemverhalten führen. Nehmen Sie den Bus erst in Betrieb, nachdem Sie sich vergewissert haben, dass alle Teilnehmer vorschriftsmäßig angeschlossen und konfiguriert sind.

### 2.1 Art der Hinweise

Beachten Sie unbedingt die Warnungen und Hinweise am Rand:



- **Gefahr** für Gesundheit und Leben durch elektrischen Schlag oder Bewegung des Antriebs.



- **Achtung:** Nichtbeachtung verstößt gegen Sicherheitsvorschriften oder gesetzliche Vorgaben und kann zu Personen- oder Sachschäden führen.



- **Prüfen:** Überprüfen Sie vor Inbetriebnahme, bei Störungen oder auftretenden Problemen zuerst diese Punkte.



- **Tipp**, nützlicher Hinweis.

### 3 Technische Daten

Die CANopen-Schnittstelle ist als Modul in die Servoregler eingebaut. Sie besteht aus zwei RJ-45-Steckverbindern, die sich an der Frontseite des Geräts befinden. Die Steckverbinder sind gemäß CANopen Cabling and Connector Pin Assignment DR 303-1 belegt. Die Signale entsprechen der Norm ISO 11898. Zwischen Busanschluss und CAN-Controller besteht eine galvanische Trennung.

Unterstützte Übertragungsraten

- 1000, 800, 500, 250, 125, 50, 20, 10 kBit/s

Funktionen der CANopen-Schnittstelle entsprechend CANopen-Kommunikationsprofil CiA 301 V 4.0.2:

- NMT-Master: nein (Master-Funktionen auf Anfrage)
- NMT-Slave: ja
- Extended Boot-up: nein
- Minimum Boot-up: ja
- COB-ID Distribution: SDO, Default
- Node-ID Distribution: Software
- No. of PDOs: 4 Rx/4 Tx
- PDO Modes: synchron (zyklisch, azyklisch), asynchron
- Variables PDO Mapping: ja
- Emergency Message: ja
- Life Guarding: ja
- No. of SDOs: 2
- Device Profile: DRIVECOM 22 mit herstellerspezifischen Erweiterungen und Teile von CiA 402

## 4 CANopen-Einführung



Zum Verständnis dieser Betriebsanleitung wird vorausgesetzt, dass der Leser mit der Begriffswelt von CANopen vertraut ist, insbesondere mit den Begriffen aus dem CANopen-Kommunikationsprofil CiA 301. Dieser Abschnitt gibt einen kurzen Überblick über die wichtigsten Begriffe und Abkürzungen. Er kann die Originaldokumente und ggf. eine entsprechende Schulung nicht ersetzen.

Bei Benutzung von Begriffen aus den CANopen-Profilen wurde weitgehend auf eine Übersetzung aus dem Englischen verzichtet, um unnötige Missverständnisse zu vermeiden.

### 4.1 Begriffe und Abkürzungen

**CAL** (CAN Application Layer = CAN-Anwendungsschicht)

Kommunikationsschicht oberhalb des eigentlichen CAN-Busses, die für Anwendungen des CAN-Busses in offenen Kommunikationssystemen geschaffen wurde. Sie besteht aus den Elementen NMT, DBT, LMT und CMS. Da CAL sehr umfangreich ist und bei der Nutzung viele Freiheitsgrade bestehen, wurde mit dem CANopen-Kommunikationsprofil CiA 301 eine Reihe von CAL-Funktionen speziell für automatisierungstechnische Anwendungen definiert.

**CAN** (Controller Area Network)

Serielles Bussystem (auch CAN-Bus genannt), das ursprünglich für die Verwendung in Automobilen konzipiert wurde und inzwischen auch in der Automatisierungstechnik eingesetzt wird. CANopen erweitert die Protokolle des CAN-Busses um zusätzliche Schichten.

**CAN-Controller**

Elektronischer Baustein, der die Abwicklung des CAN-Protokolls übernimmt.

**CANopen**

Von CiA definiertes Kommunikationsmodell, das auf CAN-Bus und CAL basiert. Um den Einsatz von Geräten unterschiedlicher Hersteller an einem Bus zu vereinfachen, wurde mit dem CANopen-Kommunikationsprofil CiA 301 eine Reihe von CAL-Funktionen speziell für automatisierungstechnische Anwendungen definiert. Darauf aufbauend werden für bestimmte Gerätearten weitere Profile definiert, wie z. B. CiA 402 für Antriebe.

**CiA** (CAN in Automation International Users and Manufacturers Group)

Verein von Herstellern und Anwendern von Geräten mit CAN-Schnittstelle.

**CMS** (CAN Message Specification)

Teil von CAL, definiert verschiedene Mechanismen für die Übertragung von Daten.

**COB** (Communication Object = Kommunikationsobjekt)

Auf dem CAN-Bus werden alle Daten in Paketen übertragen, die COB genannt werden (ein anderer Name ist CAN-Message = CAN-Meldung). Die am CAN-Bus angeschlossenen Geräte können COBs senden und empfangen.

Diese COBs sind nicht zu verwechseln mit den in dieser Betriebsanleitung beschriebenen Kommunikationsobjekten. Die in dieser Betriebsanleitung beschriebenen Kommunikationsobjekte sind keine Datenpakete, sondern Variablen.

**COB-ID** (COB-Identifizier)

Jedes COB wird über einen Identifizier, der Bestandteil des COB ist, eindeutig identifiziert. Der CAN-Bus unterstützt bis zu 2032 COBs, die über 11 Bit lange Identifizier identifiziert werden. Eine Erweiterung von CAN sieht auch 29 Bit lange Identifizier vor, diese werden von den Servoreglern nicht unterstützt. COB-IDs werden in dieser Betriebsanleitung immer als Hexadezimalwerte angegeben.

**DBT** (Distributor = Verteiler)

Teil von CAL, regelt die Verteilung von COB-IDs. Servoregler nutzen wie die meisten CANopen-Geräte einfachere Möglichkeiten der Zuordnung von COB-IDs zu einem Gerät: Sie werden über Standardwerte (Standardeinstellungen) in Abhängigkeit von der Node-ID gewählt und können ggf. über SDO verändert werden.

**DRIVECOM**

Verein von Antriebsherstellern, der Standards für die Vernetzung von Antrieben (Profile) entwickelt hat. Das DRIVECOM-Profil 22 für positionierende Antriebe, das im Servoregler implementiert ist, war Grundlage für die Entwicklung des CANopen-Antriebsprofils CiA 402 durch CiA.

**EDS** (Electronic Data Sheet)

Beschreibung der CANopen-Funktionen und Variablen als Datei auf einem Datenträger. Das EDS kann mit SPP Windows generiert werden und ermöglicht einen einfachen Zugriff auf ein Gerät.

**EMCY** (Emergency)

Emergency-Funktion zur Übermittlung von Störungen einschließlich Störungscode an den Master.

**NMT** (Network Management = Netzwerkmanagement)

Teil von CAL, für Initialisierung, Konfiguration und Fehlerbehandlung zuständig.

**Node-ID** (Node identification)

Identifiziert ein Gerät eindeutig im CANopen-Netzwerk, d. h. alle Geräte müssen unterschiedliche Node-IDs haben. Bei Servoreglern wird die Node-ID über die Software eingestellt. Von der Node-ID werden die COB-IDs abgeleitet. Node-IDs werden in dieser Betriebsanleitung immer als Hexadezimalwerte angegeben.

**PDO** (Process Data Object = Prozessdaten-Objekt)

Wird für den schnellen Echtzeit-Zugriff auf ausgewählte Daten eingesetzt. Für bestimmte Variablen oder Variablengruppen sind Abbildungen (Mappings) auf bestimmte PDOs vorkonfiguriert.

Für den Zugriff auf alle übrigen Variablen ist das SDO vorgesehen.

**Profil**

Auf dem Gebiet der Kommunikation mit Bus-Systemen sind Profile Dokumente, die der Geräte-Standardisierung dienen. Dabei werden entweder Kommunikationsfunktionen (in einem Kommunikationsprofil) oder Gerätefunktionen (in einem Geräteprofil) aus Sicht der Kommunikationsschnittstelle beschrieben.

**RPDO** (Receive PDO = Empfangs-Prozessdaten-Objekt)

PDO, das vom Servoregler empfangen wird (enthält z. B. Lageziel).

**SDO** (Service Data Object = Parameter-Objekt)

Das SDO erlaubt den Zugriff auf alle Variablen in einem CANopen-Gerät; bei Servoreglern sind dies die Antriebs- und CANopen-Variablen.

Das SDO wird im Allgemeinen zur Konfiguration eingesetzt. Für den schnellen Echtzeitzugriff auf ausgewählte Variablen werden PDOs verwendet.

**SYNC** (Synchronisation)

Mit Hilfe der SYNC-Message können PDOs auch synchron übertragen werden.

**TPDO** (Transmit PDO = Sende-Prozessdaten-Objekt)

PDO, das vom Servoregler gesendet wird (enthält z. B. *Lage-Istwert*)

## **Variable**

Auf alle Antriebs- und CANopen-Funktionen kann der Anwender über Variablen zugreifen. Variablen können dabei nur ein Wort groß (z. B. *Lageregler Kp*) oder auch sehr umfangreich sein (z. B. Teileprogramm mit 71 Elementen zu jeweils 128 Byte). Auf diese Variablen kann wiederum über SDOs oder PDOs zugegriffen werden. Die CANopen-Variablen für den Servoregler sind in dieser Betriebsanleitung *kursiv* gedruckt und in einem separaten Abschnitt beschrieben.

## 5 Anschluss und Inbetriebnahme

Für den Anschluss und zur Statusanzeige sind die Servoregler mit folgenden Elementen ausgestattet:

- Busanschluss
- Leuchtdioden

Diese Elemente befinden sich auf der Frontseite der TrioDrive-D/CS- und MidiDrive-D/CS-Servoregler.

Bei TrioDrive D/CS und MidiDrive D/CS sind keine Drehcodierschalter vorhanden, die entsprechenden Einstellungen werden über die Software vorgenommen.

### 5.1 Busanschluss

X4.1 CAN-In: RJ-45-Steckverbinder an der Frontplatte

X4.2 CAN-Out: RJ-45-Steckverbinder an der Frontplatte

Pin	Signal	Belegung
1	CAN_H	Busleitung (dominant high)
2	CAN_L	Busleitung (dominant low)
3	CAN_GND	Ground / 0 V / V-
4	–	reserviert
5	–	reserviert
6	CAN_SHLD	Schirmanschluss für Sonderanwendungen (nur nach Rücksprache mit dem Werk benutzen)
7	CAN_GND	Ground / 0 V / V-
8	–	nicht belegt
Geh.	Schirm	Der Kabelschirm muss mit dem Steckergehäuse verbunden sein. Verwenden Sie Standard-CAT5-Kabel.

Die beiden RJ-45-Steckverbinder sind gemäß CiA CANopen Cabling and Connector Pin Assignment DR 303-1 belegt. Die Signale entsprechen der Norm ISO 11898. Zwischen Busanschluss und CAN-Controller besteht eine galvanische Trennung.

Pin 8 von X4.1 und X4.2 ist für eine externe Einspeisung der CAN-Versorgungsspannung durchverbunden, falls der am Bus angeschlossene Master keine Spannungsversorgung bereitstellt.



### 5.1.1 Anschlussstecker und Terminierung



Die Ausführung des Busanschlusses mit einem CAN-In- und CAN-Out- Steckverbinder ermöglicht den Anschluss des Geräts an den Bus ohne Benutzung von T-Verbindern und Stichleitungen.



Falls der Servoregler als erster oder letzter Teilnehmer an den Bus geschaltet ist, muss die korrekte Terminierung des Busses gewährleistet werden. Dies kann z. B. durch Aufstecken eines Terminierungssteckers auf den CAN-Out-Anschluss erfolgen (Abschlusswiderstand von 120  $\Omega$  zwischen Pin 1 und Pin 2).

### 5.1.2 Busleitungen



Maximal zulässige Buslängen (Summe aller Busleitungen):

Übertragungsrate kBit/s	1000	500	125	50
Max. Buslänge m	40	100	500	1000



Es ist der Einsatz von Leitungen nach ISO 11898 Teil 2 vorgeschrieben.

## 5.2 Leuchtdioden

Auf der Vorderseite der Servoregler befinden sich unterhalb der Aufschrift Bus die Zustands-LEDs Error, Run, Aux1 und Aux2 des CANopen-Moduls.

Diese Leuchtdioden zeigen die Zustände der CANopen-Schnittstelle an (Kommunikation).

LED Error (rot) zeigt den Zustand der Sende- und Empfangsüberwachung des CAN-Controllers im Servoregler an:

- Aus: OK (CAN-Begriff: Error Active)
- Blinken 0,5 Hz: Warnung (CAN-Begriff: Error Passive)  
Sende- und/oder Empfangsfehler gehäuft aufgetreten;  
Gerät, Bus laufen weiter
- Ein: Störung (CAN-Begriff: Bus-Off)  
Sende- und/oder Empfangsfehler zu stark aufgetreten;  
Gerät vom Bus getrennt

LED Run (grün) zeigt den Zustand im NMT Node State Diagram an:

- Blinken ca. 2 Hz: Pre-Operational
- Dauernd ein: Operational

Aux1 und Aux2 sind derzeit nicht belegt.

## 5.3 Inbetriebnahme



Beachten Sie bei der Inbetriebnahme unbedingt die Sicherheitshinweise in Abschnitt 2 (Seite 9).

Die Inbetriebnahme eines Servoreglers mit CANopen-Schnittstelle muss in 3 Phasen vorgenommen werden, die im Folgenden beschrieben werden.

### 5.3.1 Projektierung der CANopen-Schnittstelle

Nehmen Sie Einstellungen der CANopen-Schnittstelle mit Hilfe der Projektierungshilfe in Anhang C vor. Kopieren Sie sich diese Projektierungshilfe je Servoregler einmal. Füllen Sie die Blätter aus und nehmen Sie erst dann die Einstellungen am Gerät vor. Zuerst müssen die Hardware-Einstellungen vorgenommen werden, dann können die Maschinendaten der CANopen-Schnittstelle eingestellt werden.

Achten Sie darauf, dass Sie die Einstellungen der Maschinendaten der CANopen-Schnittstelle mit der CANopen-Konfigurationssoftware auf Diskette/Festplatte und im Servoregler speichern, siehe Abschnitt 12 (Seite 31) und/oder die Betriebsanleitung der verwendeten CANopen-Konfigurationssoftware.



Die Projektierung der CANopen-Schnittstelle mit Einstellung des Geräts kann gegen Berechnung auch im Werk vorgenommen werden. Bitte wenden Sie sich bei Bedarf an ESR.

### 5.3.2 Parametrierung der Antriebsfunktionen (Servoregler, -motor)



Für die Inbetriebnahme der ESR-Servoantriebe wird ein PC mit dem Bedien- und Inbetriebnahmeprogramm SPP Windows benötigt. Weitere Informationen hierzu finden Sie in der Betriebsanleitung 6710.101 „Funktionen und Parameter“.

### 5.3.3 Inbetriebnahme des CANopen-Systems (Master, weitere Geräte)

Die Inbetriebnahme des CANopen-Systems mit dem Master und evtl. weiteren Geräten ist der letzte Inbetriebnahmeschritt.

Zur Vereinfachung der Inbetriebnahme können mit SPP Windows so genannte elektronische Datenblätter (EDS) erstellt werden, die einen einfachen Zugriff auf den Servoregler ermöglichen.

## 6 Netzwerkmanagement und COB-ID-Verteilung

### 6.1 NMT Node State Diagram

Das Kommunikationsverhalten der CANopen-Schnittstelle eines Geräts wird über so genannte Netzwerkmanagement-Funktionen (NMT-Services) beeinflusst. Das Kommunikationsverhalten des Servoreglers und die Beeinflussungsmöglichkeiten werden über das folgende Zustandsdiagramm (NMT Node State Diagram) dargestellt. Grundsätzliche Informationen zu Zustandsmaschinen finden Sie in Anhang A (Seite 55).

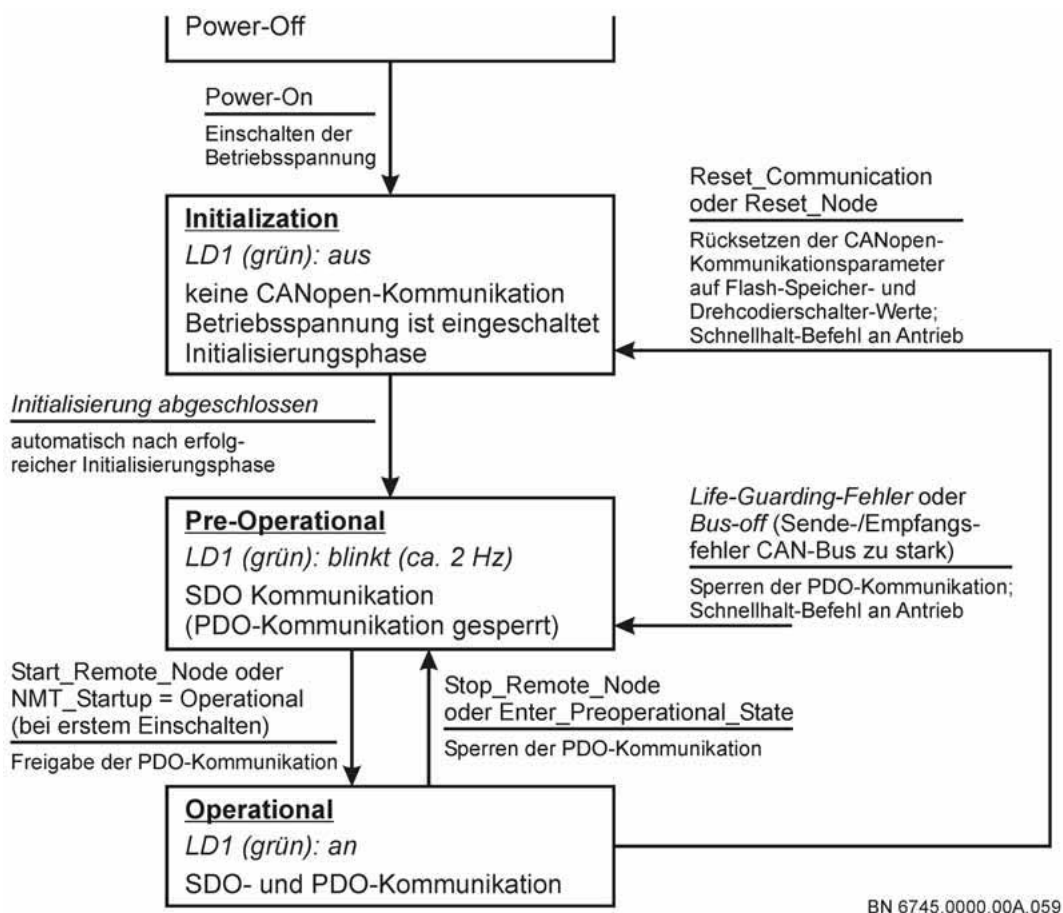


Bild 1: NMT Node State Diagram Servoregler

Das Bild zeigt die vom Servoregler unterstützten Zustände und Zustandsübergänge des im CANopen-Kommunikationsprofil CiA 301 angegebenen NMT Node State Diagram.

Nach dem Einschalten der Steuerspannung und abgeschlossener interner Initialisierung der Firmware des Servoreglers erreicht das Gerät automatisch

den Zustand „Pre-Operational“. Mithilfe der Variablen *NMT Startup* ist automatisches Schalten nach „Operational“ möglich.

Die übrigen Zustandsübergänge werden im Allgemeinen durch NMT-Services ausgelöst, die vom NMT-Master gesendet werden:

- Start\_Remote\_Node
- Stop\_Remote\_Node
- Enter\_Preoperational\_State
- Reset\_Communication
- Reset\_Node



Im Zustand „Pre-Operational“ ist nur eine SDO-Kommunikation möglich, die PDO-Kommunikation ist gesperrt. Im Zustand „Operational“ findet schließlich auch eine PDO-Kommunikation statt.



Der momentan aktive Zustand wird über die grüne Leuchtdiode Run angezeigt, siehe Abschnitt 5.2 (Seite 16).

## 6.2 COB-ID-Verteilung

Die Verteilung der COB-IDs für die vom Gerät unterstützten CANopen-Kommunikationsobjekte erfolgt standardmäßig durch den Hersteller in Abhängigkeit von der eingestellten Node-ID. Diese Verteilung wird „Predefined Connection Set“ genannt.

COB-ID nach dem „Predefined Connection Set“ (CANopen COBs)		
CANopen-Kommunikationsobjekt	COB-ID (hexadezimal)	COB-ID veränderbar über Variable
NMT	0	–
SYNC	80	COB-ID Sync Message
EMCY	80 + Node-ID	–
RPDO 1	200 + Node-ID	Receive PDO 1 Comm. Parameter
RPDO 2	300 + Node-ID	Receive PDO 2 Comm. Parameter
RPDO 3	400 + Node-ID	Receive PDO 3 Comm. Parameter
RPDO 4	500 + Node-ID	Receive PDO 4 Comm. Parameter
TPDO 1	180 + Node-ID	Transmit PDO 1 Comm. Parameter
TPDO 2	280 + Node-ID	Transmit PDO 2 Comm. Parameter
TPDO 3	380 + Node-ID	Transmit PDO 3 Comm. Parameter
TPDO 4	480 + Node-ID	Transmit PDO 4 Comm. Parameter
SDO (Tx)	580 + Node-ID	–
SDO (Rx)	600 + Node-ID	–
Node Guarding	700 + Node-ID	COB-ID Guarding Protocol

Soweit angegeben können die COB-IDs über Variablen geändert werden.

## 7 SDO-Kommunikation (Parameter)

Über die SDO-Kommunikation kann auf alle Antriebsvariablen und alle CANopen-Variablen des Servoreglers zugegriffen werden.



Die CANopen-Variablen sind in dieser Betriebsanleitung in Abschnitt 14 (Seite 32ff) beschrieben, die Antriebsvariablen in der Betriebsanleitung 6710.101 „Funktionen und Parameter“.

Bei der SDO-Kommunikation ist der Servoregler ein „Server“ im Sinne des CANopen-Kommunikationsprofils CiA 301. Das heißt der Servoregler ermöglicht anderen Geräten, die „Clients“ genannt werden, den lesenden oder schreibenden Zugriff auf seine Variablen.



Eine SDO-Kommunikation ist in den Zuständen „Pre-Operational“ und „Operational“ des NMT Node State Diagram möglich. Diese Zustände werden durch Blinken bzw. Leuchten der grünen Leuchtdiode Run angezeigt.

Die SDO-Kommunikation wird entsprechend dem CANopen-Kommunikationsprofil CiA 301 mit dem „multiplexed domain protocol“ abgewickelt, wobei wie dort beschrieben für Variablen bis zu 4 Byte Länge die „expedited“-Variante des Protokolls verwendet wird. Die SDO-Kommunikation verwendet 2 COB-IDs, jeweils eine für jede Übertragungsrichtung des Protokolls. Die verwendeten COB-IDs hängen von der Node-ID des Servoreglers ab. Über die Variablen *SDO 1 Server Parameter* und *SDO 2 Server Parameter* können auch die COB-IDs abgefragt werden.



Die für die SDO-Kommunikation verwendeten COB-IDs hängen nur von der Node-ID ab. Sie können durch Parametrierung nicht verändert werden. Damit kann unabhängig von der weiteren Projektierung der CANopen-Schnittstelle auf alle Variablen zugegriffen werden, wenn nur die Node-ID des Geräts und die Übertragungsrate bekannt sind. Um eine maximale Flexibilität beim Austausch von Geräten zu ermöglichen ist es daher sinnvoll, die Maschinendaten CANopen nach jedem Einschalten des Systems vom CANopen-Master per SDO-Kommunikation zum Servoregler zu übertragen. Beim Austausch von Geräten sind dann nur noch Node-ID und Übertragungsrate einzustellen, um die CANopen-Schnittstelle zu konfigurieren. Entsprechendes gilt auch für die Übertragung der übrigen Gerätedaten.

Eine Beschreibung der für die SDO-Kommunikation verwendeten Variablen *SDO 1 Server Parameter* und *SDO 2 Server Parameter* finden Sie in Abschnitt 14.2 (Seite 36).

## 8 PDO-Kommunikation (Prozessdaten)

Für den schnellen Echtzeit-Zugriff auf ausgewählte Variablen steht die PDO-Kommunikation zur Verfügung.

Entsprechend der folgenden Tabelle sind jeweils sieben Abbildungen von Variablen auf PDOs (PDO-Mappings) in Sende- und Empfangsrichtung vordefiniert; das PDO-Mapping ist darüber hinaus frei konfigurierbar (variables PDO-Mapping).

Die nachfolgende Tabelle zeigt das werkseitig eingestellte Mapping:

PDO-Bezeichnung	Beschreibung				
	Name	Index	Subindex	Länge (Byte)	Zustand
RPDO1	Achsen-Steuerwort	6040	0	2	aktiv
RPDO2	Achsen-Steuerwort	6040	0	2	inaktiv
	Achsen-Betriebsart	6060	0	2	inaktiv
RPDO3	Achsen-Steuerwort	6040	0	2	inaktiv
	Lageziel	607a	0	4	inaktiv
RPDO4	Achsen-Steuerwort	6040	0	2	inaktiv
	Verfahrgeschwindigkeit	6081	0	4	inaktiv
TPDO1	Achsen-Statuswort	6041	0	2	aktiv
TPDO2	Achsen-Statuswort	6041	0	2	inaktiv
	Betriebsarten-Auswahlcode	6061	0	2	inaktiv
TPDO3	Achsen-Statuswort	6041	0	2	inaktiv
	Lage-Istwert (in LE)	6064	0	4	inaktiv
TPDO4	Achsen-Statuswort	6041	0	2	inaktiv
	Geschwindigkeits-Istwert	606c	0	4	inaktiv

Die Namensgebung der PDOs entspricht den Konventionen im CANopen-Antriebsprofil CiA 402.



PDO-Kommunikation ist nur im Zustand „Operational“ des NMT Node State Diagram möglich. Dieser Zustand wird durch Leuchten der grünen Leuchtdiode Run angezeigt.

Die auf ein PDO abgebildeten Variablen, wie sie in der vorstehenden Tabelle beschrieben sind, können über *Receive PDO 1 Mapping* bis *Transmit PDO 4 Mapping* gelesen und geschrieben werden. Über *Receive PDO 1 Comm. Parameter* bis *Transmit PDO 4 Comm. Parameter* kann das Verhalten der Receive- bzw. der Transmit-PDOs abgefragt und eingestellt werden.



Standardmäßig ist je ein PDO in Sende- und Empfangsrichtung aktiv (valid): RPDO1 und TPDO1. Alle übrigen PDOs sind inaktiv. Über die Kommunikationsparameter kann jedes PDO einzeln aktiviert oder deaktiviert werden.

Die COB-IDs der PDOs leiten sich standardmäßig von der Node-ID ab.

Die Übertragungsart der PDOs ist standardmäßig asynchron. Das Ereignis, welches eine Übertragung des PDO seitens des Servoreglers auslöst, ist eine Änderung des Wertes einer der auf das PDO abgebildeten Variablen. Über die Kommunikationsparameter kann bei Bedarf eine synchrone Übertragungsart gewählt werden.

Damit Transmit PDOs mit Variablen, deren Wert sich häufig ändert (z. B. Lage-Istwert), den CAN-Bus nicht überlasten, kann die Häufigkeit der Übertragung über die so genannte *Inhibit Time* begrenzt werden. Die gemäß CiA 301 festgelegte Einheit der *Inhibit Time* beträgt 100 µs (d. h. ein Wert von 100 entspricht 10 ms). Die Auswertung im Servoregler erfolgt mit einer Auflösung von ca. 3 ms, sodass nur Änderungen der *Inhibit-Time*-Einstellung um jeweils mindestens den Wert 30 eine Wirkung haben. Über die Kommunikationsparameter kann die *Inhibit Time* für jedes *Transmit PDO* einzeln eingestellt werden.

Eine Beschreibung der Variablen *Receive PDO 1* bis *4* und *Transmit PDO 1* bis *4* finden Sie in Abschnitt 14.3 (Seite 37ff).

Die Standardbelegung der PDOs kann über die *PDO Mapping Parameter* geändert werden (variables PDO-Mapping). Es stehen jeweils 4 RPDOs und 4 TPDOs mit jeweils maximal 8 Einträgen pro PDO zur Verfügung. Folgende Parameter können in PDOs abgebildet werden:

Index	Objekt	Typ	PDO-Richtung
1001	Error Register	U8	TPDO
5e93	I2t-Auslastung	U16	TPDO
5e95	Lagegeber-Messwert2	I32	TPDO
5e96	Lagegeber-Messwert2 LE	I32	TPDO
5e9f	Lagegeber-Messwert1 LE	I32	TPDO
5f03	Widerstand Motor-Temperatursensor	U32	TPDO
5f0b	Kühlertemperatur	I16	TPDO
5f0c	Motortemperatur	I16	TPDO
5f1c	Lagegeber-Messwert	I32	TPDO
603f	Störungscode	U16	TPDO
6041	Statuswort	U16	TPDO
6061	Betriebsarten-Auswahlcode Anzeige	I16	TPDO
6063	Lagegeber-Istwert	I32	TPDO
6064	Lage-Istwert	I32	TPDO
606c	Geschwindigkeits-Istwert	I32	TPDO
6078	Strom-Istwert	I16	TPDO



<b>Index</b>	<b>Objekt</b>	<b>Typ</b>	<b>PDO-Richtung</b>
6079	Zwischenkreisspannung	U16	TPDO
5f54	Digitale Ausgänge	U8 (Array)	TPDO, RPDO
5f56	Digitale Eingänge	U8 (Array)	TPDO, RPDO
5f5e	Programm-Variablen	I32 (Array)	TPDO, RPDO
5e9c	Strom-Max-Betrag2	U16	RPDO
5ef4	Beschleunigungs-Zeit	U32	RPDO
5ef5	Verzögerungs-Zeit	U32	RPDO
5ef6	Schnellhalt-Zeit	U32	RPDO
5ef7	Geschwindigkeits-Bezugswert	U32	RPDO
5fc1	Alternativer Interpol. Data Record	I16/I8	RPDO
6040	Steuerwort	U16	RPDO
6060	Achsen-Betriebsart	I16	RPDO
6071	Momentensollwert-extern	I16	RPDO
6073	Strom-Max-Betrag	U16	RPDO
607a	Lageziel	I32	RPDO
6081	Verfahrgeschwindigkeit	I32	RPDO
6086	Rampenform-Geschwindigkeit	I16	RPDO
60b1	Geschwindigkeits-Offset	I32	RPDO
60c1	Interpolation Data Record	I32	RPDO
6fc2	Alternative für Velocity Offset	I16/I8	RPDO

## 9 Synchronisation

### 9.1 Asynchrone Übertragung von PDOs

Standardmäßig sind die PDOs (Prozessdaten) der Servoregler über die jeweiligen Kommunikationsparameter so konfiguriert, dass ihre Übertragung asynchron erfolgt (*Transmission Type* = 255). Das bedeutet:

- *Receive PDOs* (PDOs, die der Servoregler empfängt) werden bei ihrem Eintreffen vom Servoregler übernommen. Das sendende Gerät legt fest, wann es die Daten sendet (typischerweise bei jeder Änderung des Wertes einer der auf das PDO abgebildeten Variablen).
- *Transmit PDOs* (PDOs, die der Servoregler sendet) werden vom Servoregler gesendet, wenn sich der Wert einer der auf das PDO abgebildeten Variablen geändert hat. Eine Überlastung des Busses durch Variablen, die sich oft ändern, kann verhindert werden, indem die Häufigkeit der Übertragung durch eine *Inhibit Time* begrenzt wird. Mithilfe des *Event Timer* wird eingestellt, wann ein PDO spätestens gesendet wird, auch wenn sich der Wert der Variablen nicht geändert hat.

### 9.2 Synchroner Übertragung von PDOs

Die PDOs können auch synchron zu einer SYNC-Message übertragen werden (*Transmission Type* = 1 .. 240). Die SYNC-Message wird von genau einem der Geräte am CAN-Bus, dem so genannten SYNC-Master, in festen Abständen gesendet. Ist kein SYNC-Master (keine SYNC-Message) vorhanden, können nur asynchrone PDOs übertragen werden.

Die Übertragung erfolgt zyklisch, der Wert *n* des *Transmission Type* gibt die Anzahl der SYNC-Message zwischen zwei PDO-Übertragungen an.

- *Receive PDOs* (PDOs, die der Servoregler empfängt) werden bei ihrem Eintreffen vom Servoregler zunächst zwischengespeichert. Die Daten werden jeweils mit dem Eintreffen der *n*-ten SYNC-Message übernommen, *n* entspricht dabei dem Wert des *Transmission Type* (Wertebereich 1 .. 240).
- Die Daten für *Transmit PDOs* (PDOs, die der Servoregler sendet) werden vom Servoregler beim Eintreffen der *n*-ten SYNC-Message ermittelt und gesendet.

Hinweis: Die Time Stamp von CANopen wird nicht unterstützt. *Inhibit Time* und *Event Timer* sind bei der synchronen Übertragung von PDOs ohne Bedeutung.

Eine Beschreibung der Variablen *COB-ID SYNC Message* finden Sie in Abschnitt 14.4 (Seite 45).

## 10 Interpolated Position Mode

Die Achsen-Betriebsart „Interpolated Position Mode“ dient zum Betrieb einer oder mehrerer Achsen, für die eine Zeitinterpolation der Sollwerte erforderlich ist (z. B. zur Bahnsteuerung). Die übergeordnete Steuerung sendet hierzu über ein RPDO Lageziele in festen Zeitintervallen an den Antrieb. Zusätzlich zu diesen Positionen berechnet der Feininterpolator des Antriebs weitere Lageziele im Zeitintervall des Lageregelkreises (1 ms). Die Übermittlung der Lageziele erfolgt im Interpolated Position Mode nur in Lagegeber-Schritten (LgS), nicht in Längeneinheiten (LE).

Im Interpolated Position Mode werden die PDOs synchron zu einer SYNC-Message übertragen. Bei einer Störung oder einem Ausfall der SYNC-Übertragung behält der Antrieb die letzte Sollposition so lange bei, bis er eine neue SYNC-Message erhält.



Die Überwachung der Hardware-Endschalter und die Funktion „Lage messen“ sind in der Betriebsart Interpolated Position Mode nicht aktiv.

Für den Interpolated Position Mode sind im Achsen-Statuswort keine betriebsartenabhängigen Bits vorhanden.

Im Achsen-Steuerwort hat Bit 4 die folgende spezielle Bedeutung:

- Interpolation freigeben

Über eine positive Flanke im Bit „Interpolation freigeben“ wird die Interpolation gestartet.

Die Achsen-Betriebsart kann nur gewechselt werden, wenn sich die Achsen-Zustandsmaschine im Zustand „Einschaltsperr“, „Einschaltbereit“ oder „Eingeschaltet“ befindet.



Im Interpolated Position Mode wird nur der Streckentyp Linearachse unterstützt, ein Betrieb als Rundachse (Endlosachse) ist nicht möglich.

Die CANopen-Kommunikation wird in der Betriebsart Interpolated Position Mode durch folgende Parameter beeinflusst:

- Die Abbildungs-PDOs (PDO Mapping)
  - dienen zur Übertragung von Achsen-Statuswort und -Steuerwort sowie der Lageziele (im *Interpolation Data Record*) und Lage-Istwerte.

Sie sind in Abschnitt 14.3.1 ab Seite 37 beschrieben.

- Die Maschinendaten PDOs (Comm. Parameter)
  - steuern die synchrone Übertragung der oben genannten Abbildungs-PDOs.

Sie sind in Abschnitt 14.3.2 ab Seite 40 beschrieben.

- Die Maschinendaten Interpolated Position Mode mit *Interpolation Time Period*
  - bestimmen das Zeitintervall für die Vorgabe der Lageziele in Millisekunden.

Eine Beschreibung der Maschinendaten Interpolated Position Mode finden Sie in Abschnitt 14.5.2 ab Seite 47.

Durch das variable PDO-Mapping sind für den Interpolated Position Mode keine PDOs mehr vorgegeben. Beim Mapping ist zu beachten, dass die Objekte *GeraetSteuer* (Index 6040, Subindex 0) und *IpolDataRecord* (Index 60c1, Subindex 1) in ein RPDO gemappt werden müssen, die Objekte *GeraetStatus* (Index 6041, Subindex 0) und *PosIstLg* (Index 6063, Subindex 1) müssen in ein TPDO gemappt werden. Die Übertragung erfolgt synchron, das heißt der Wert  $n$  des *Transmission Type* = 1.

In der Achsen-Betriebsart Interpolated Position Mode arbeiten der Feininterpolator als Sollwertgenerator und der Lageregler zusammen; dem Lageregler sind dabei der Drehzahl- und der Stromregler unterlagert.

Die Servoregler können mit Steuerungen unterschiedlicher Hersteller in der Achsen-Betriebsart „Interpolated Position Mode“ betrieben werden. Wenden Sie sich bei Bedarf bitte an ESR.

## 10.1 Interpolated Position Mode aktivieren



Als Voraussetzung für die Steuerung der Achsen-Zustandsmaschine müssen die entsprechenden PDOs auf „valid“ gesetzt sein.

1. Wählen Sie die Achsen-Betriebsart Interpolated Position Mode, sofern sie nicht bereits über das Maschinendatum *Achsen-Betriebsart Vorwahl* eingestellt ist.
2. Mit der Freigabe der PDO-Kommunikation durch den NMT-Master wird die Zustandsmaschine der NMT-Services auf „Operational“ geschaltet.
3. Den aktuellen Lage-Istwert auslesen und zur Initialisierung des Interpolated Position Mode in den *Interpolation Data Record* schreiben.
4. Die Achsen-Zustandsmaschine auf „Betrieb freigegeben“ schalten.
5. Über „Interpolation freigegeben“ (Bit 4 im Achsen-Steuerwort) die Interpolation aktivieren.

## 10.2 Interpolated Position Mode deaktivieren

Zum Deaktivieren des Interpolated Position Mode muss die Achsen-Zustandsmaschine in den Zustand „Einschaltsperr“, „Einschaltbereit“ oder „Eingeschaltet“ geschaltet werden. Gegebenenfalls die NMT-Zustandsmaschine auf „Pre-Operational“ schalten.

Danach kann vom Interpolated Position Mode in eine andere Achsen-Betriebsart gewechselt werden.

# 11 Überwachungsmechanismen

## 11.1 Störungsbehandlung (Error, Emergency)

Jede Störung wird mit einem Emergency-Telegramm an den Master übermittelt. Dieses Emergency-Telegramm ist wie folgt aufgebaut:

- Byte 0, 1: Pre-Defined Error Field, Subindex 1 (Störungscode)
- Byte 2: Error Register
- Byte 3 .. 7: nicht genutzt (0)

Die Bytes 0 und 1 enthalten den Störungscode, wie er in der CANopen-Variablen *Pre-Defined Error Field*, Subindex 1, gemeldet wird.

Byte 2 enthält den Inhalt der CANopen-Variablen *Error Register*.

Die Bytes 3 bis 7 des Emergency-Telegramms sind nicht genutzt.

Eine Beschreibung der für die Überwachung verwendeten Variablen *Error Register*, *Pre-Defined Error Field* und *COB-ID Emergency Message* finden Sie in Abschnitt 14.6.2 (Seite 49ff).

## 11.2 Node Guarding, Life Guarding

Der Servoregler unterstützt die im CANopen-Kommunikationsprofil CiA 301 definierte Prozedur des Node Guarding/Life Guarding.

Wenn die Überwachung des Masters freigegeben ist (CANopen-Variablen *Guard Time* und *Life Time Factor* sind ungleich Null), reagiert der Servoregler mit einer Störung, wenn der Master innerhalb einer bestimmten Zeit (Life Time) keine Guarding-Message gesendet hat:

- wenn der *Abort Connection Option Code* den Wert 1 enthält, wird der Antrieb mit einem Schnellhalt-Befehl angehalten, ein eventuell laufendes Teilprogramm wird gestoppt,
- Störungscode 8130 wird angezeigt,
- im NMT Node State Diagram erfolgt ein Übergang in den Zustand „Pre-Operational“.

Eine Beschreibung der für die Überwachung verwendeten Variablen *Guard Time* und *Life Time Factor* finden Sie in Abschnitt 14.6.3 (Seite 50).

## 11.3 Heartbeat

Der Servoregler unterstützt die im CANopen-Kommunikationsprofil CiA 301 definierte Prozedur des Heartbeat-Producers.

Wenn der Heartbeat-Producer freigegeben ist (CANopen-Variable *Producer Heartbeat Time* ungleich 0), wird zyklisch entsprechend der dort vorgegebenen Zeit ein Heartbeat-Telegramm mit COB-ID 700 + Node-ID mit folgendem Inhalt gesendet:

- Byte 0: 127 (Pre-Operational) bzw. 5 (Operational)

Eine Beschreibung der Variablen *Producer Heartbeat Time* finden Sie in Abschnitt 14.6.3 (Seite 50).

## 11.4 Sende- und Empfangsüberwachung

Jeder CAN-Controller überwacht automatisch die Sende- und Empfangsdaten auf Störungen. Wenn Sende- oder Empfangsfehler auftreten, erfolgt die Meldung in zwei Stufen:

- Falls gehäuft Sende- oder Empfangsfehler auftauchen, wird zuerst zur Warnung der Störungscode 8100 angezeigt (CAN-Begriff: Error Passive). Das Gerät arbeitet anschließend weiter wie vor dieser Warnung.
- Wenn sich die Anzahl der Sende- und Empfangsfehler weiter erhöht und eine bestimmte Grenze überschreitet, schaltet sich der CAN-Controller selbstständig vom Bus ab (CAN-Begriff: Bus-Off). Damit gekoppelt ist eine Störreaktion:
  - der Antrieb wird mit einem Schnellhalt-Befehl angehalten, ein eventuell laufendes Teileprogramm wird gestoppt,
  - Störungscode 8180 wird angezeigt,
  - im NMT Node State Diagram erfolgt ein Übergang in den Zustand Pre-Operational.



Die o. a. Warnungen und Störungen werden durch Blinken bzw. Leuchten der roten Leuchtdiode Error angezeigt.



Überprüfen Sie im Fall der o. a. Warnungen und Störungen bei der Fehlersuche auch den Anschluss aller übrigen Geräte an den CAN-Bus.

## 12 Speicherfunktionen

Die Speicherfunktionen ermöglichen es, die Maschinendaten CANopen (und weitere Gerätedaten) im nichtflüchtigen Speicher des Servoreglers (Servo-EEPROM) zu speichern.



Um maximale Flexibilität beim Austausch von Geräten zu erreichen, ist es sinnvoll, die Maschinendaten CANopen nach jedem Einschalten des Systems vom CANopen-Master per SDO-Kommunikation zum Servoregler zu übertragen.

Eine Beschreibung der für die Speicherfunktion verwendeten Variablen *Store Parameters* und *Restore Default Parameters* finden Sie in Abschnitt 14.7 (Seite 51ff).

### 12.1 Speicherfunktionen Programm SPP Windows

Die Maschinendaten CANopen werden üblicherweise über den Zugriff auf die Variable *Store Parameters* im nichtflüchtigen Speicher des Servoreglers (Servo-EEPROM) gesichert. Sie können außerdem über den Menüpunkt „Kommunikation/im Gerät speichern“ des Programms SPP Windows vom RAM des Servoreglers in das Servo-EEPROM gesichert werden.

## 13 Geräteinformationen

Informationen über die angeschlossenen Geräte wie Gerätetyp und Hersteller sind ebenfalls in Variablen enthalten.

Eine Beschreibung der für Geräteinformationen verwendeten Variablen *Device Type*, *Manufacturer Device Name*, *Manufacturer Hardware Version*, *Manufacturer Software Version* und *Identity* finden Sie in Abschnitt 14.8 (Seite 52ff).



## 14 Variablen-Beschreibungen

In den folgenden Abschnitten finden Sie Informationen zu den in dieser Betriebsanleitung beschriebenen CANopen-Variablen. Die Variablen-Beschreibungen sind in diesem Teil nach Funktionen sortiert. Ein Zugriff nach dem Namen ist über das Stichwortverzeichnis (Seite 66) möglich. In Anhang B finden Sie alle CANopen-Variablen nach Index sortiert aufgelistet.



Die Variablen-Beschreibungen sind nach einem standardisierten Schema aufgebaut. Dieses Schema wird entsprechend angepasst auch für den Aufbau der Variablenliste in Anhang B (Seite 57) sowie für die Variablen-Beschreibungen in der Betriebsanleitung 6710.101 „Funktionen und Parameter“ verwendet.

Grundinformationen über eine Variable sind in einer Tabelle der folgenden Form zusammengefasst. Die in dieser Tabelle angegebenen Werte sind immer Standardwerte.

<b>Name</b>	Index: 1234, Kurzname: VarName
<b>SPP Windows</b> ...	
<b>Typ</b> ...	
<b>Zugriff</b> ...	
<b>Art</b> ...	
<b>Einheit</b> ...	
<b>Standardwert</b> ...	

### Name

Der Variablenname steht fett gedruckt links oben in jeder Variablen-Beschreibung. Im beschreibenden Text ist der Variablenname kursiv gedruckt.

### Index

Der Index ist eine Hexadezimalzahl, über die bei der SDO-Kommunikation über den CAN-Bus auf die Variable zugegriffen werden kann.

### Kurzname

Dies ist ein Kurzname, der für die spätere Verwendung mit speziellen Hochsprachen-Treiberprogrammen vorgesehen ist.

### SPP Windows

Im Bedien- und Inbetriebnahmeprogramm SPP Windows wird die Variable wie hier angegeben dargestellt. Die Zeile enthält als Beispielwert die Standardeinstellung ab Werk (unterstrichen), die über einen entsprechenden Zugriff auf die Variable *Restore Default Parameters* auch zu einem späteren Zeitpunkt wie-

derhergestellt werden kann. Wenn diesem Wert eine Maßeinheit zugeordnet ist, wird diese zusammen mit dem Wert angegeben.

## Typ

Diese Zeile ist aus mehreren Informationen zusammengesetzt:

- Objekt-Code:

- Simple Variable

Die Simple Variable enthält jeweils nur ein Element des angegebenen Datentyps. Beim Zugriff auf die Simple Variable ist immer der Subindex 0 anzugeben.

- Array

Das Array enthält mehrere Elemente des angegebenen Datentyps. Beim Zugriff auf das Array ist immer der Subindex des gewünschten Elements anzugeben (1 oder größer). Speziell für CANopen-Variablen gilt, dass über Subindex 0 die Anzahl der Elemente des Array ausgelesen werden kann.

- Record

Ein Record enthält mehrere Elemente unterschiedlicher Datentypen. Beim Zugriff auf den Record ist immer der Subindex des gewünschten Elements anzugeben (1 oder größer). Speziell für CANopen-Variablen gilt, dass über Subindex 0 die Anzahl der Elemente des Record ausgelesen werden kann.

- Datentyp:

Mögliche Werte:

- Boolean (Bool)

- Integer8 (i8) = Byte in 2er-Komplement-Darstellung

- Integer16 (i16) = Wort in 2er-Komplement-Darstellung

- Integer32 (i32) = Doppelwort in 2er-Komplement-Darstellung

- Unsigned8 (u8) = Byte, vorzeichenlos

- Unsigned16 (u16) = Wort, vorzeichenlos

- Unsigned32 (u32) = Doppelwort, vorzeichenlos

- Visible-String (VisStr) mit Länge = eine Folge von Bytes, die Text enthält

- Octet-String (OctStr) mit Länge = eine Folge von Bytes, die binär codierte Informationen enthält

- PDO CommPar (PDOCoP), siehe Abschnitt 14.3.2 (Seite 40ff) PDO Mapping (PDOMap), siehe Abschnitt 14.3.1 (Seite 37ff)

- SDO Parameter (SDOPar), siehe Abschnitt 14.2 (Seite 36)

**Zugriff, R/W**

Mögliche Werte:

- Lesen und Schreiben, kurz R W für Read und Write
- nur Lesen, kurz R für Read

**Art**

Mögliche Werte (mit Buchstaben als Abkürzung):

- Konstante (Variablen-Wert ändert sich nicht), kurz „F“
- Maschinendaten CANopen (alle Maschinendaten, die in dieser Betriebsanleitung beschrieben sind), kurz „MC“
- Maschinendaten, kurz „M“
- Status-Information, kurz „S“
- Steuer-Information, kurz „S“
- Veränderlicher Wert, kurz „V“

**Array-Element n**

Falls in einem Array die einzelnen Elemente bestimmte Namen, Einheiten und/oder Standardwerte haben, werden diese hier angegeben.

**Record-Element n**

Falls in einem Record die einzelnen Elemente Namen, Einheiten und/oder Standardwerte haben, werden diese hier angegeben.

**Beschreibung**

Je nach Variable erscheint vor und/oder nach der Tabelle ein kurzer beschreibender Text.

Beispiel für eine Variablen-Beschreibung:

<b>Receive PDO 1 Mapping</b>		Index: 1600, Kurzname: RPD01Map
<b>SPP Windows</b>	RPD01 Mapping 1 <u>60400010</u>	
<b>Typ</b>	Record, 1 Element (Subindex 0 .. 1), Typ PDO Mapping	
<b>Record-Element 1</b>	Default: Achsen-Steuerwort (Index 6040, Subindex 0, Länge 16 Bit)	
<b>Zugriff</b>	nur Lesen	
<b>Art</b>	Konstante (Variablenwert ändert sich nicht)	

Diese Variable zeigt an, welche Variable auf *Receive PDO 1* abgebildet ist.

## 14.1 Variablen-Beschreibung Netzwerk-Management

<b>Node-ID</b>	Index: 5fb4, Kurzname: CANNodeId
<b>SPP Windows</b>	Node-Id SDO Server 1 <u>01</u> Node-Id SDO Server 2 <u>00</u>
<b>Typ</b>	Array, 2 Elemente (Subindex 0 .. 2), U8
<b>Array-Element 1</b>	Node-ID SDO Server 1
<b>Array-Element 2</b>	Node-ID SDO Server 2
<b>Zugriff</b>	Lesen und Schreiben
<b>Art</b>	veränderlicher Wert

Über diese Variable kann die eingestellte Node-ID gelesen und geschrieben werden. Node-IDs werden in dieser Betriebsanleitung immer als Hexadezimalwerte angegeben. Wird *Node-ID SDO Server 1* mit gesetztem Bit 7 beschrieben, wird der Wert des „Predefined Connection Set“ für diese Node-ID voreingestellt.

Änderungen bei den Variablen *Node-ID SDO Server 1* und *CAN Baudrate* werden erst nach dem nächsten Einschalten des Geräts wirksam, *Node-ID Server 2* wird sofort wirksam.

Wert 0 bei Node-ID2 deaktiviert Server 2.

<b>NodeID</b>	Index: 100b, Kurzname: NodeId
<b>SPP Windows</b>	Node-Id SDO Server 1 <u>01</u> Node-Id SDO Server 2 <u>00</u>
<b>Typ</b>	Simple Variable 32 Bit
<b>Zugriff</b>	nur Lesen
<b>Art</b>	veränderlicher Wert

Diese Variable ist nur zur Kompatibilität mit der alten DS 301-Version 3.0 und CodeSys vorhanden. Ihre Funktion entspricht der Variablen *Node-ID*, Index 5fb4.

<b>CAN Baudrate</b>	Index: 5fb5, Kurzname: CANBaudrate
<b>SPP Windows</b>	CAN Baudrate <u>500</u> Kbps
<b>Typ</b>	Simple-Variable, U8
<b>Zugriff</b>	Lesen und Schreiben
<b>Art</b>	veränderlicher Wert

Über diese Variable kann die Baudrate eingestellt werden.

<b>NMT Startup</b>		Index: 1f80, Kurzname: NMTStartup
<b>SPP Windows</b>	NMT Startup	<u>Pre-Operational</u>
<b>Typ</b>	Simple-Variable, U32	
<b>Zugriff</b>	Lesen und Schreiben	
<b>Art</b>	veränderlicher Wert	

Über diese Variable wird bestimmt, in welchen Zustand des NMT Node State Diagrams der Servoantrieb nach Einschalten wechseln soll.

Zulässige Werte sind:

Wert	Zustand
12	Pre-Operational
8	Operational

Änderungen in dieser Variablen werden erst mit dem nächsten Einschalten des Geräts wirksam.

## 14.2 Variablen-Beschreibung SDO-Kommunikation

<b>Server SDO 1 Parameter</b>		Index: 1200, Kurzname: SSD01
<b>SPP Windows</b>	–	
<b>Typ</b>	Record, 2 Elemente (Subindex 0 .. 2), Typ SDO Parameter	
<b>Record-Element 1</b>	COB-ID client → server, 600 + Node-ID, U32	
<b>Record-Element 2</b>	COB-ID server → client, 580 + Node-ID, U32	
<b>Zugriff</b>	nur Lesen	
<b>Art</b>	veränderlicher Wert	

<b>Server SDO 2 Parameter</b>		Index: 1201, Kurzname: SSD02
<b>SPP Windows</b>	–	
<b>Typ</b>	Record, 3 Elemente (Subindex 0 .. 3), Typ SDO Parameter	
<b>Record-Element 1</b>	COB-ID client → server, 600 + Node-ID, U32	
<b>Record-Element 2</b>	COB-ID server → client, 580 + Node-ID, U32	
<b>Record-Element 3</b>	SDO2 Node-ID, U8	
<b>Zugriff</b>	Lesen und Schreiben	
<b>Art</b>	veränderlicher Wert	

*Server SDO 1 Parameter* und *Server SDO 2 Parameter* zeigen für Server 1 und Server 2 die COB-IDs an, die für die SDO-Kommunikation verwendet werden.

Die Werte hängen ab von der Einstellung der Node-ID. Die Record-Elemente sind wie folgt aufgebaut:

<b>Record-Element 1, 2 (u32)</b>	Bit 31:	0 = SDO valid
	Bit 30 .. 12:	reserviert (immer 0)
	Bit 11 .. 0:	COB-ID SDO je Richtung

COB-IDs werden in dieser Betriebsanleitung immer als Hexadezimalwerte angegeben.

## 14.3 Variablen-Beschreibung PDO-Kommunikation

### 14.3.1 Abbildungs-PDOs (PDO Mapping)

#### 14.3.1.1 Receive PDOs Mapping

Die nachfolgend beschriebenen PDOs finden Sie in SPP Windows im Fenster „Parametrierung“ unter „CANopen/Receive PDO Mapping“.

Die Variablenbeschreibungen der *Receive-PDO...Mapping* zeigen an, auf welche Funktion über das jeweilige PDO zugegriffen wird (z. B. *Achsen-Steuerwort*). Diese Variablen sind Records, die für jede Variable, die auf das PDO abgebildet ist, ein Element vom Typ unsigned32 haben. Dort sind für die abgebildete Variable jeweils eingetragen:

<b>Record-Element x (u32)</b>	Bit 31 .. 16:	Index (16 Bit)
	Bit 15 .. 8:	Subindex (8 Bit)
	Bit 7 .. 0:	Länge in Bit (8 Bit)

Die Anzahl der abgebildeten Variablen (entspricht der Anzahl der Elemente des Records) kann über den Subindex 0 gelesen und geschrieben werden.

<b>Receive PDO 1 Mapping</b>		Index: 1600, Kurzname: RPD01Mapping
<b>SPP Windows</b>	RPD01 Mapping 1	60400010
<b>Typ</b>	Record, max. 8 Elemente (Subindex 0 .. 8), Typ PDO Mapping	
<b>Record-Element 0</b>	Anzahl Elemente (0-8), Default = 1	
<b>Record-Element 1</b>	Default: Achsen-Steuerwort (Index 6040, Subindex 0, Länge 16 Bit)	
<b>Zugriff</b>	Lesen und Schreiben	
<b>Art</b>	Maschinendaten CANopen	

<b>Receive PDO 2 Mapping</b>		Index: 1601, Kurzname: RPD02Mapping
<b>SPP Windows</b>	RPD02 Mapping 1 <u>60400010</u> RPD02 Mapping 2 <u>60600010</u>	
<b>Typ</b>	Record, max. 8 Elemente (Subindex 0 .. 8), Typ PDO-Mapping	
<b>Record-Element 0</b>	Anzahl Elemente (0-8), Default = 2	
<b>Record-Element 1</b>	Default: Achsen-Steuerwort (Index 6040, Subindex 0, Länge 16 Bit)	
<b>Record-Element 2</b>	Default: Achsen-Betriebsart (Index 6060, Subindex 0, Länge 16 Bit)	
<b>Zugriff</b>	Lesen und Schreiben	
<b>Art</b>	Maschinendaten CANopen	

<b>Receive PDO 3 Mapping</b>		Index: 1602, Kurzname: RPD03Mapping
<b>SPP Windows</b>	RPD03 Mapping 1 <u>60400010</u> RPD03 Mapping 2 <u>607A0020</u>	
<b>Typ</b>	Record, max. 8 Elemente (Subindex 0 .. 8), Typ PDO Mapping	
<b>Record-Element 0</b>	Anzahl Elemente (0-8), Default = 2	
<b>Record-Element 1</b>	Default: Achsen-Steuerwort (Index 6040, Subindex 0, Länge 16 Bit)	
<b>Record-Element 2</b>	Default: Lageziel (Index 607A, Subindex 0, Länge 32 Bit)	
<b>Zugriff</b>	Lesen und Schreiben	
<b>Art</b>	Maschinendaten CANopen	

<b>Receive PDO 4 Mapping</b>		Index: 1603, Kurzname: RPD04Mapping
<b>SPP Windows</b>	RPD04 Mapping 1 <u>60400010</u> RPD04 Mapping 1 <u>60810020</u>	
<b>Typ</b>	Record, max. 8 Elemente (Subindex 0 .. 8), Typ PDO Mapping	
<b>Record-Element 0</b>	Anzahl Elemente (0-8), Default = 2	
<b>Record-Element 1</b>	Default: Achsen-Steuerwort (Index 6040, Subindex 0, Länge 16 Bit)	
<b>Record-Element 2</b>	Default: Verfahrensgeschwindigkeit (Index 6081, Subindex 0, Länge 32 Bit)	
<b>Zugriff</b>	Lese und Schreiben	
<b>Art</b>	Maschinendaten CANopen	

Das oben beschriebene Standard-Mapping kann überschrieben werden (variables Mapping). Dazu ist das entsprechende Record-Element nach oben beschriebenem Schema neu zu setzen. Um das Mapping zu verändern, setzen Sie die Anzahl der abgebildeten Variablen (Subindex 0) zunächst auf 0. Nachdem Sie das Mapping geändert haben, muss die Anzahl der Variablen auf die entsprechende Anzahl gemappter Elemente (0–8) gesetzt werden.

### 14.3.1.2 Transmit PDOs Mapping

Die nachfolgend beschriebenen PDOs finden Sie in SPP Windows im Fenster „Parametrierung“ unter „CANopen/Transmit PDO Mapping“.

Die Variablenbeschreibungen der *Transmit-PDO...Mapping* zeigen an, auf welche Funktion über das jeweilige PDO zugegriffen wird. Diese Variablen sind Records, die für jede Variable, die auf das PDO abgebildet ist, ein Element vom

Typ unsigned32 haben. Dort sind für die abgebildete Variable jeweils eingetragen:

<b>Record-Element x (u32)</b>	Bit 31 .. 16:	Index (16 Bit)
	Bit 15 .. 8:	Subindex (8 Bit)
	Bit 7 .. 0:	Länge in Bit (8 Bit)

Die Anzahl der abgebildeten Variablen (entspricht der Zahl der Elemente des Records) kann über den Subindex 0 gelesen und geschrieben werden.

---

<b>Transmit PDO 1 Mapping</b>		Index: 1a00, Kurzname: TPD01Mapping
<b>SPP Windows</b>	TPD01 Mapping 1 <a href="#">60410010</a>	
<b>Typ</b>	Record, max. 8 Elemente (Subindex 0 .. 8), Typ PDO Mapping	
<b>Record-Element 0</b>	Anzahl Elemente (0-8), Default = 1	
<b>Record-Element 1</b>	Default: Achsen-Statuswort (Index 6041, Subindex 0, Länge 16 Bit)	
<b>Zugriff</b>	Lesen und Schreiben	
<b>Art</b>	Maschinendaten CANopen	

---



---

<b>Transmit PDO 2 Mapping</b>		Index: 1a01, Kurzname: TPD02Mapping
<b>SPP Windows</b>	TPD02 Mapping 1 <a href="#">60410010</a> TPD02 Mapping 2 <a href="#">60610010</a>	
<b>Typ</b>	Record, max. 8 Elemente (Subindex 0 .. 8), Typ PDO Mapping	
<b>Record-Element 0</b>	Anzahl Elemente (0-8), Default = 2	
<b>Record-Element 1</b>	Default: Achsen-Statuswort (Index 6041, Subindex 0, Länge 16 Bit)	
<b>Record-Element 2</b>	Default: Betriebsarten-Auswahlcode Anzeige (Index 6061, Subindex 0, Länge 16 Bit)	
<b>Zugriff</b>	Lesen und Schreiben	
<b>Art</b>	Maschinendaten CANopen	

---



---

<b>Transmit PDO 3 Mapping</b>		Index: 1a02, Kurzname: TPD03Mapping
<b>SPP Windows</b>	TPD03 Mapping 1 <a href="#">60410010</a> TPD03 Mapping 2 <a href="#">60640020</a>	
<b>Typ</b>	Record, max. 8 Elemente (Subindex 0 .. 8), Typ PDO Mapping	
<b>Record-Element 0</b>	Anzahl Elemente (0-8), Default = 2	
<b>Record-Element 1</b>	Default: Achsen-Statuswort (Index 6041, Subindex 0, Länge 16 Bit)	
<b>Record-Element 2</b>	Default: Lage-Istwert (Index 6064, Subindex 0, Länge 32 Bit)	
<b>Zugriff</b>	Lesen und Schreiben	
<b>Art</b>	Maschinendaten CANopen	

---



<b>Transmit PDO 4 Mapping</b>		Index: 1a03, Kurzname: TPD04Mapping
<b>SPP Windows</b>	TPD04 Mapping 1 <u>60410010</u> TPD04 Mapping 2 <u>606C0020</u>	
<b>Typ</b>	Record, max. 8 Elemente (Subindex 0 .. 8), Typ PDO Mapping	
<b>Record-Element 0</b>	Anzahl Elemente (0-8), Default = 2	
<b>Record-Element 1</b>	Default: Achsen-Statuswort (Index 6041, Subindex 0, Länge 16 Bit)	
<b>Record-Element 2</b>	Default: Geschwindigkeits-Istwert (Index 606c, Subindex 0, Länge 32 Bit)	
<b>Zugriff</b>	Lesen und Schreiben	
<b>Art</b>	Maschinendaten CANopen	

Das oben beschriebene Standard-Mapping kann überschrieben werden (variables Mapping). Dazu ist das entsprechende Record-Element nach oben beschriebenem Schema neu zu setzen. Um das Mapping zu verändern, setzen Sie die Anzahl der abgebildeten Variablen (Subindex 0) zunächst auf 0. Nachdem Sie das Mapping geändert haben, muss die Anzahl der Variablen auf die entsprechende Anzahl gemappter Elemente (0–8) gesetzt werden.

## 14.3.2 Maschinendaten PDOs (Comm. Parameter)

### 14.3.2.1 Receive PDOs Comm. Parameter

Die nachfolgend beschriebenen PDOs finden Sie in SPP Windows im Fenster „Parametrierung“ unter „CANopen/Receive PDOs“.

Über die Variablen *Receive PDO...Comm. Parameter* kann das Verhalten der *Receive PDOs* abgefragt und eingestellt werden. Die Variablen enthalten folgende Elemente:

<b>Record-Element 1 (u32)</b>	Bit 31:	1 = PDO not valid 0 = PDO valid
	Bit 30:	0 = RTR allowed
	Bit 20 .. 12:	reserviert (immer 0)
	Bit 11 .. 0:	COB-ID des PDO
<b>Record-Element 2 (u8)</b>	Transmission Type:	0 = synchron, azyklisch
		1 .. 240 = synchron, zyklisch
		255 = asynchron

Bit 31 von Record-Element 1 erlaubt, das PDO zu aktivieren (valid) oder zu deaktivieren (not valid). Wenn ein *Receive PDO* deaktiviert ist, haben COBs (CAN-Messages), die an die COB-ID des PDO übertragen werden, keine Wirkung auf die gemappten Variablen.

Bit 30 von Record-Element 1 zeigt an, ob ein Remote Transmission Request (RTR) zulässig ist, d. h. eine Abfrage des PDO durch Senden eines bestimmten COB. Standardmäßig ist „RTR allowed“ eingestellt.

Bit 11 .. 0 von Record-Element 1 erlauben die Einstellung der COB-ID des PDO. Normalerweise sollte der Standardwert, der von der eingestellten Node-ID abhängt, nicht verändert werden. Für weitere Informationen beachten Sie bitte Abschnitt 6.2 (Seite 20). COB-IDs werden in dieser Betriebsanleitung immer als Hexadezimalwerte angegeben. Die COB-ID darf nur geändert werden, wenn das PDO inaktiv ist (Bit 31 = 1).

Record-Element 2 erlaubt die Einstellung der Übertragungsart (Transmission Type) des PDO. Standardmäßig ist bei allen *Receive PDOs* der *Transmission Type* 255 (asynchrones PDO, profilspezifisches Ereignis) voreingestellt, wobei das Ereignis bei Receive PDOs vom sendenden Gerät abhängt.

Bei *Receive PDOs* (also PDOs, die empfangen werden) ist die Einstellung einer *Inhibit Time* über Record-Element 3 nicht erforderlich, da für die Einstellung dieser Zeit der Sender des PDO verantwortlich ist.

<b>Receive PDO 1 Comm. Parameter</b>		Index: 1400, Kurzname: RPD01ComParm
<b>SPP Windows</b>	RPD01 COB-ID <u>00000200</u> + Node-ID RPD01 Transmission Type <u>255</u>	
<b>Typ</b>	Record, 2 Elemente (Subindex 0 .. 2), Typ PDO CommPar	
<b>Record-Element 1</b>	Default: valid (0) / RTR (0) / COB-ID = 200 + Node-ID	
<b>Record-Element 2</b>	Default: transmission type = asynchron (255)	
<b>Zugriff</b>	Lesen und Schreiben	
<b>Art</b>	Maschinendaten CANopen	

<b>Receive PDO 2 Comm. Parameter</b>		Index: 1401, Kurzname: RPD02ComParm
<b>SPP Windows</b>	RPD02 COB-ID <u>80000300</u> + Node-ID RPD02 Transmission Type <u>255</u>	
<b>Typ</b>	Record, 2 Elemente (Subindex 0 .. 2), Typ PDO CommPar	
<b>Record-Element 1</b>	Default: not valid (1) / RTR (0) / COB-ID = 300 + Node-ID	
<b>Record-Element 2</b>	Default: transmission type = asynchron (255)	
<b>Zugriff</b>	Lesen und Schreiben	
<b>Art</b>	Maschinendaten CANopen	

<b>Receive PDO 3 Comm. Parameter</b>		Index: 1402, Kurzname: RPD03ComParm
<b>SPP Windows</b>	RPD03 COB-ID <u>80000400</u> + Node-ID RPD03 Transmission Type <u>255</u>	
<b>Typ</b>	Record, 2 Elemente (Subindex 0 .. 2), Typ PDO CommPar	
<b>Record-Element 1</b>	Default: not valid (1) / RTR (0) / COB-ID = 400 + Node-ID	
<b>Record-Element 2</b>	Default: transmission type = asynchron (255)	
<b>Zugriff</b>	Lesen und Schreiben	
<b>Art</b>	Maschinendaten CANopen	

<b>Receive PDO 4 Comm. Parameter</b>		Index: 1403, Kurzname: RPD04ComParm
<b>SPP Windows</b>	RPD04 COB-ID <u>80000500</u> + Node-ID RPD04 Transmission Type <u>255</u>	
<b>Typ</b>	Record, 2 Elemente (Subindex 0 .. 2), Typ PDO CommPar	
<b>Record-Element 1</b>	Default: not valid (1) / RTR (0) / COB-ID = 500 + Node-ID	
<b>Record-Element 2</b>	Default: transmission type = asynchron (255)	
<b>Zugriff</b>	Lesen und Schreiben	
<b>Art</b>	Maschinendaten CANopen	

### 14.3.2.2 Transmit PDOs Comm. Parameter

Die nachfolgend beschriebenen PDOs finden Sie in SPP Windows im Fenster „Parametrierung“ unter „CANopen/Transmit PDOs“.

Über die Variablen *Transmit PDO...Comm. Parameter* kann das Verhalten der *Transmit PDOs* abgefragt und eingestellt werden. Die Variablen enthalten folgende Elemente:

<b>Record-Element 1 (u32)</b>	Bit 31:	1 = PDO not valid 0 = PDO valid
	Bit 30:	0 = RTR allowed
	Bit 20 .. 12:	reserviert (immer 0)
	Bit 11 .. 0:	COB-ID des PDO
<b>Record-Element 2 (u8)</b>	Transmission Type:	0 = synchron, azyklisch
		1 .. 240 = synchron, zyklisch
		255 = asynchron
<b>Record-Element 3 (u16)</b>	Inhibit Time (in ms):	0 = no inhibit time
<b>Record-Element 4</b>	nicht implementiert	–
<b>Record-Element 5 (u16)</b>	Event Timer (in ms):	0 = no event timer

Bit 31 von Record-Element 1 erlaubt, das PDO zu aktivieren (valid) oder zu deaktivieren (not valid). Wenn ein *Transmit PDO* aktiviert ist, wird das zugehörige COB (die zugehörige CAN Message) bei bestimmten Ereignissen und/oder zu bestimmten Zeiten auf den CAN-Bus gesendet. Wann gesendet wird, hängt von der Einstellung des *Transmission Type* und gegebenenfalls der *Inhibit Time* und dem *Event Timer* des PDO ab.

Bit 30 von Record-Element 1 zeigt an, ob ein Remote Transmission Request (RTR) zulässig ist, d. h. eine Abfrage des PDO durch Senden eines bestimmten COB. Standardmäßig ist „RTR allowed“ eingestellt.

Bit 11 .. 0 von Record-Element 1 erlauben die Einstellung der COB-ID des PDO. Normalerweise sollte der Standardwert, der von der eingestellten Node-ID

abhängt, nicht verändert werden. Für weitere Informationen beachten Sie bitte Abschnitt 6.2 (Seite 20). COB-IDs werden in dieser Betriebsanleitung immer als Hexadezimalwerte angegeben.

Record-Element 2 erlaubt die Einstellung der Übertragungsart (Transmission Type) des PDO. Standardmäßig ist bei allen *Transmit PDOs* *Transmission Type* 255 (asynchrones PDO, profilspezifisches Ereignis) voreingestellt. Das Ereignis, welches eine Übertragung eines *Transmit PDO* seitens des Servoreglers auslöst, ist eine Änderung des Wertes einer der auf das PDO abgebildeten Variablen.

Bei *Transmit PDOs* könnten Variablen, deren Wert sich häufig ändert, den CAN-Bus überlasten. Um dies zu verhindern, kann die Häufigkeit der Übertragung über die *Inhibit Time* in Record-Element 3 begrenzt werden. Die gemäß CiA 301 festgelegte Einheit der *Inhibit Time* beträgt 100 µs (d. h. einem Wert von 100 entsprechen 10 ms). Die Auswertung im Servoregler erfolgt mit einer Auflösung von ca. 3 ms, sodass nur Änderungen der *Inhibit-Time*-Einstellung um jeweils mindestens den Wert 30 eine Wirkung haben.

Bei asynchronen PDOs wird der *Event Timer* des Record-Element 5 verwendet, um die Zeit (in ms) einzustellen, nach der das PDO spätestens gesendet wird.

<b>Transmit PDO 1 Comm. Parameter</b>		Index: 1800, Kurzname: TPD01ComParm
<b>SPP Windows</b>	TPD01 COB-ID	<u>00000180</u> + Node-ID
	TPD01 Transmission Type	<u>255</u>
	TPD01 Inhibit Time	<u>0,0</u> ms
	TPD01 Event Timer	<u>0</u> ms
<b>Typ</b>	Record, 5 Elemente (Subindex 0 .. 5), Typ PDO CommPar	
<b>Record-Element 1</b>	Default: valid (0), RTR (0), 180 + Node-ID	
<b>Record-Element 2</b>	Default: transmission type = asynchron (255)	
<b>Record-Element 3</b>	Default: inhibit time = 0 ms	
<b>Record-Element 4</b>	nicht implementiert	
<b>Record-Element 5</b>	Default: event timer = 0 ms	
<b>Zugriff</b>	Lesen und Schreiben	
<b>Art</b>	Maschinendaten CANopen	

<b>Transmit PDO 2 Comm. Parameter</b>		Index: 1801, Kurzname: TPD02ComParm
<b>SPP Windows</b>	TPD02 COB-ID	<u>80000280</u> + Node-ID
	TPD02 Transmission Type	<u>255</u>
	TPD02 Inhibit Time	<u>0,0</u> ms
	TPD02 Event Timer	<u>0</u> ms
<b>Typ</b>	Record, 5 Elemente (Subindex 0 .. 5), Typ PDO CommPar	
<b>Record-Element 1</b>	Default: not valid (1), RTR (0), 280 + Node-ID	
<b>Record-Element 2</b>	Default: transmission type = asynchron (255)	
<b>Record-Element 3</b>	Default: inhibit time = 0 ms	
<b>Record-Element 4</b>	nicht implementiert	
<b>Record-Element 5</b>	Default: event timer = 0 ms	
<b>Zugriff</b>	Lesen und Schreiben	
<b>Art</b>	Maschinendaten CANopen	

<b>Transmit PDO 3 Comm. Parameter</b>		Index: 1802, Kurzname: TPD03ComParm
<b>SPP Windows</b>	TPD03 COB-ID	<u>80000380</u> + Node-ID
	TPD03 Transmission Type	<u>255</u>
	TPD03 Inhibit Time	<u>0,0</u> ms
	TPD03 Event Timer	<u>0</u> ms
<b>Typ</b>	Record, 5 Elemente (Subindex 0 .. 5), Typ PDO CommPar	
<b>Record-Element 1</b>	Default: not valid (1), RTR (0), 380 + Node-ID	
<b>Record-Element 2</b>	Default: transmission type = asynchron (255)	
<b>Record-Element 3</b>	Default: inhibit time = 0 ms	
<b>Record-Element 4</b>	nicht implementiert	
<b>Record-Element 5</b>	Default: event timer = 0 ms	
<b>Zugriff</b>	Lesen und Schreiben	
<b>Art</b>	Maschinendaten CANopen	

<b>Transmit PDO 4 Comm. Parameter</b>		Index: 1803, Kurzname: TPD04ComParm
<b>SPP Windows</b>	TPD04 COB-ID	<u>80000480</u> + Node-ID
	TPD04 Transmission Type	<u>255</u>
	TPD04 Inhibit Time	<u>0,0</u> ms
	TPD04 Event Timer	<u>0</u> ms
<b>Typ</b>	Record, 5 Elemente (Subindex 0 .. 5), Typ PDO CommPar	
<b>Record-Element 1</b>	Default: not valid (1), RTR (0), 480 + Node-ID	
<b>Record-Element 2</b>	Default: transmission type = asynchron (255)	
<b>Record-Element 3</b>	Default: inhibit time = 0 ms	
<b>Record-Element 4</b>	nicht implementiert	
<b>Record-Element 5</b>	Default: event timer = 0 ms	
<b>Zugriff</b>	Lesen und Schreiben	
<b>Art</b>	Maschinendaten CANopen	

## 14.4 Variablen-Beschreibung Synchronisation

### 14.4.1 Maschinendaten Synchronisation

Diese Variable finden Sie in SPP Windows im Fenster „Parametrierung“ unter „CANopen/Synchronisation“.

<b>COB-ID SYNC Message</b>		Index: 1005, Kurzname: IdSync
<b>SPP Windows</b>	COB-ID SYNC Message	<u>00000080</u>
<b>Typ</b>	Simple-Variable, Unsigned32	
<b>Zugriff</b>	Lesen und Schreiben	
<b>Art</b>	Maschinendaten CANopen	
<b>Standardwert</b>	80	

Über diese Variable wird die COB-ID für die SYNC-Message abgefragt und eingestellt.

COB-IDs werden in dieser Betriebsanleitung immer als Hexadezimalwerte angegeben.

## 14.5 Variablen-Beschreibung Interpolated Position Mode

### 14.5.1 Prozessdaten-Abbildung Interpolated Position Mode

<b>Interpolation Data Record</b>		Index: 60c1, Kurzname: Ipo1DataRecord
<b>SPP Windows</b>	–	
<b>Typ</b>	Record, 1 Element (Subindex 0 .. 1), Integer32	
<b>Record-Element 1</b>	Lageziel, Interpolated Position Mode (Index 60c1, Subindex 1, Länge 32 Bit)	
<b>Zugriff</b>	Lesen und Schreiben	
<b>Wertebereich</b>	$-2^{24}..2^{24}-1$ LgS	

Über diese Variable werden die Lageziele (in Lagegeber-Schritten) übermittelt.

<b>Alternativer Interpolation Data Record</b>		Index: 5fc1, Kurzname: AltIpo1DataRecord
<b>SPP Windows</b>	–	
<b>Typ</b>	Record, 2 Elemente (Subindex 0 .. 2)	
<b>Record-Element 1</b>	Interpolation Data Record Bit 0 .. 15, Integer16	
<b>Record-Element 2</b>	Interpolation Data Record Bit 16 .. 23, Integer8	
<b>Zugriff</b>	Lesen und Schreiben	
<b>Wertebereich</b>	$-2^{24}..2^{24}-1$ LgS	

Diese Variable kann als Alternative für den 32 Bit *Interpolation Data Record* verwendet werden, um diesen Record mit nur 24 Bit zu mappen.

Zusammen mit der *Alternative für Velocity Offset* können so die Werte für *Interpolation Data Record*, *Velocity Offset* und *Steuerwort* in ein 8-Byte-PDO gemappt werden.

<b>Velocity Offset</b>		Index: 60b1, Kurzname: VelocityOffset
<b>SPP Windows</b>	–	
<b>Typ</b>	Simple Variable, Integer32	
<b>Zugriff</b>	Lesen und Schreiben	
<b>Einheit</b>	0,25 1/min	
<b>Standardwert</b>	...	

Wird diese Variable im Interpolated Position Mode gemappt, wird die Geschwindigkeitsvorsteuerung diesem Wert entnommen, der Master muss zyklisch mit dem *IpolDataRecord* gültige Werte liefern.

In anderen Betriebsarten ist diese Variable ohne Wirkung.

Ist dieser Wert in Interpolated Position Mode nicht gemappt, wird die Geschwindigkeitsvorsteuerung intern berechnet.

<b>Alternative für Velocity Offset</b>		Index: 6fc2, Kurzname: AltVelocityOffset
<b>SPP Windows</b>	–	
<b>Typ</b>	Record, 2 Elemente (Subindex 0 .. 2)	
<b>Record-Element 1</b>	Velocity Offset Bit 0 .. 15, Integer16	
<b>Record-Element 2</b>	Velocity Offset Bit 16 .. 23, Integer8	
<b>Zugriff</b>	Lesen und Schreiben	
<b>Einheit</b>	0,25 1/min	
<b>Standardwert</b>	...	

Diese Variable kann als Alternative für den 32 Bit *Velocity Offset* verwendet werden, um diesen Record mit nur 24 Bit zu mappen.

Zusammen mit dem *Alternativen Interpolation Data Record* können so die Werte für *Interpolation Data Record*, *Velocity Offset* und *Steuerwort* in ein 8-Byte-PDO gemappt werden.

## 14.5.2 Maschinendaten Interpolated Position Mode

Die nachfolgend beschriebenen Parameter finden Sie in SPP Windows im Fenster „Parametrierung“ unter „Achsen/Interpolated Position Mode“.

<b>Interpolation Sub Mode Select</b>		Index: 60c0, Kurzname: Ipo1SubmodeAuswahl
<b>SPP Windows</b>	Interpolation Submode	<u>0</u>
<b>Typ</b>	Simple-Variable, Typ Integer16	
<b>Zugriff</b>	nur Lesen, Schreiben möglich aber ohne Auswirkung	
<b>Art</b>	Konstante (Variablenwert ändert sich nicht)	
<b>Standardwert</b>	0 (lineare Interpolation)	

Diese Variable bestimmt den Interpolationsmodus. Es wird nur eine lineare Interpolation unterstützt.

<b>Interpolation Time Period</b>		Index: 60c2, Kurzname: Ipo1TimePeriod
<b>SPP Windows</b>	Interpolation Time Units	<u>4</u> $10^{\text{idx}}$ s
	Interpolation Time Index	<u>-3</u>
<b>Typ</b>	Record, 2 Elemente (Subindex 0 .. 2)	
<b>Record-Element 1</b>	Interpolation time units (1 .. 10), Unsigned8	
<b>Record-Element 2</b>	Interpolation time index (-3), Integer8	
<b>Zugriff</b>	Record-Element 1: Lesen und Schreiben Record-Element 2: nur Lesen, Schreiben möglich aber ohne Auswirkung	
<b>Art</b>	Maschinendaten CANopen, Interpolated Position Mode	

Über die *Interpolation Time Unit* wird das Zeitintervall bestimmt, in dem die Lageziele im Interpolated Position Mode übermittelt werden. Der *Interpolation Time Index* bestimmt dabei die Einheit für das Record-Element 1 (-3 für  $10^{-3}$  s = 1 ms).

<b>Interpolation Sync Definition</b>		Index: 60c3, Kurzname: Ipo1SyncDefinition
<b>SPP Windows</b>	Synchronize on group	<u>0</u>
	ip_sync every n events	<u>1</u>
<b>Typ</b>	Array, 2 Elemente (Subindex 0 .. 2), Unsigned8	
<b>Array-Element 1</b>	Synchronize on group	
<b>Array-Element 2</b>	ip_sync every n events	
<b>Zugriff</b>	nur Lesen, Schreiben möglich aber ohne Auswirkung	
<b>Art</b>	Konstante (Variablenwert ändert sich nicht)	



<b>Interpolation Data Configuration</b>		Index: 60c4, Kurzname: Ipo1DataConfig
<b>SPP Windows</b>	–	
<b>Typ</b>	Record, 6 Elemente (Subindex 0 .. 6)	
<b>Record-Element 1</b>	Max. buffer size, unsigned32; Konstante 1	
<b>Record-Element 2</b>	Actual size, unsigned32; Konstante 1	
<b>Record-Element 3</b>	Buffer organization, unsigned8; Konstante 0 (FIFO buffer)	
<b>Record-Element 4</b>	Buffer position, unsigned16; Konstante 0	
<b>Record-Element 5</b>	Size of data record, unsigned8; Konstante 1	
<b>Record-Element 6</b>	Buffer clear, unsigned8; Konstante 1	
<b>Zugriff</b>	nur Lesen, Schreiben möglich aber ohne Auswirkung	
<b>Art</b>	Konstante (Variablenwert ändert sich nicht)	

## 14.6 Variablen-Beschreibung Überwachung

### 14.6.1 Status Störungsbehandlung (Error, Emergency)

Die nachfolgend beschriebenen Variablen werden verwendet, um Störungen zu beschreiben und zu übermitteln.

<b>Error Register</b>		Index: 1001, Kurzname: ErrorReg
<b>SPP Windows</b>	–	
<b>Typ</b>	Simple-Variable, Typ Unsigned8	
<b>Zugriff</b>	nur Lesen	
<b>Art</b>	Status-Information	
<b>Standardwert</b>	0 oder eine nach dem Einschalten aufgetretene Störung	

Es werden die Bits 0 und 5 benutzt (generic error bzw. profile specific error). Im Störfall werden diese Bits gesetzt und signalisieren, dass ein Störungscode, der die Störungsursache näher beschreibt, aus der Variablen *Pre-Defined Error Field*, Subindex 1, gelesen werden kann.

Alle anderen Bits der Variablen *Error Register* sind immer 0.

<b>Pre-Defined Error Field</b>		Index: 1003 Kurzname: ErrorField
<b>SPP Windows</b>	–	
<b>Typ</b>	Array, 1 Element (Subindex 0 .. 1), Typ Unsigned32	
<b>Array-Element 1</b>	Standard Error-Field	
<b>Zugriff</b>	nur Lesen	
<b>Art</b>	Status-Information	

In dieser Variablen wird unter Subindex 1 der Störungscode der zuletzt aufgetretenen Störung angezeigt und gespeichert. Bei einer Änderung wird der neue Wert innerhalb der Emergency-Message übertragen. Weitere Informationen zu den Störungs-codes finden Sie in Anhang D (Seite 64).

## 14.6.2 Maschinendaten Störungsbehandlung (Emergency)

---

<b>COB-ID Emergency Message</b>		Index: 1014, Kurzname: IdEmcy
<b>SPP Windows</b>	COB-ID Emergency Message	<u>00000080</u> + Node-ID
<b>Typ</b>	Simple-Variable, Typ Unsigned32	
<b>Zugriff</b>	nur Lesen	
<b>Art</b>	Maschinendaten CANopen	
<b>Standardwert</b>	80 + Node-ID	

---

Diese Variable ermöglicht die Abfrage der COB-ID für die Emergency Message. COB-IDs werden in dieser Betriebsanleitung immer als Hexadezimalwerte angegeben.

---

<b>Inhibit Time Emergency Message</b>		Index: 1015, Kurzname: InhibitTimeEmcy
<b>SPP Windows</b>	Inhibit Time	<u>0,0</u> ms
<b>Typ</b>	Simple-Variable, Typ Integer16	
<b>Zugriff</b>	Lesen und Schreiben	
<b>Art</b>	Maschinendaten CANopen	
<b>Standardwert</b>	0	

---

Mit dieser Variablen kann die Häufigkeit der Übertragung der Emergency Message begrenzt werden.

---

<b>Abort Connection Option Code</b>		Index: 6007, Kurzname: AbortOptionCode
<b>SPP Windows</b>	–	
<b>Typ</b>	Simple-Variable, Integer16	
<b>Zugriff</b>	Lesen und Schreiben	
<b>Art</b>	Maschinendaten CANopen	
<b>Standardwert</b>	0 = keine Reaktion, 1 = Gerätestörung bei Node-Guarding-/Life-Guarding-Fehlern (Default)	

---

Mit dieser Variablen wird die Fehlerreaktion auf einen Kommunikations-Abbruch festgelegt

### 14.6.3 Maschinendaten Node Guarding, Life Guarding

<b>Guard Time</b>		Index: 100c, Kurzname: GuardTime
<b>SPP Windows</b>	Guard Time <u>0</u> ms	
<b>Typ</b>	Simple-Variable, Typ Unsigned16	
<b>Zugriff</b>	Lesen und Schreiben	
<b>Art</b>	Maschinendaten CANopen	
<b>Standardwert</b>	0	

Diese Variable ermöglicht die Einstellung der Überwachungszeit in Millisekunden.

<b>Life Time Factor</b>		Index: 100d, Kurzname: LifeTimeFac
<b>SPP Windows</b>	Life Time Factor <u>0</u>	
<b>Typ</b>	Simple-Variable, Typ Unsigned8	
<b>Zugriff</b>	Lesen und Schreiben	
<b>Art</b>	Maschinendaten CANopen	
<b>Standardwert</b>	0	

Diese Variable dient der indirekten Einstellung der so genannten Life Time, der Zeitspanne, innerhalb welcher der Master eine Message gesendet haben muss.

Der Wert der Variablen *Guard Time* multipliziert mit dem Wert dieses *Life Time Factor* ergibt die Life Time.

Weitere Informationen hierzu finden Sie in Abschnitt 11.2 (Seite 29).

<b>Producer Heartbeat Time</b>		Index: 1017, Kurzname:ProducerHeartbeatTime
<b>SPP Windows</b>	Producer heartbeat time <u>0 ms</u>	
<b>Typ</b>	Simple-Variable, Typ Unsigned16	
<b>Zugriff</b>	Lesen und Schreiben	
<b>Art</b>	Maschinendaten CANopen	
<b>Standardwert</b>	0 ms	

Mit dieser Variablen wird festgelegt, in welchen Abständen der Servoregler Heartbeat-Telegramme sendet (0 = aus).

Weitere Informationen hierzu finden Sie in Abschnitt 11.3 (Seite 30).

## 14.7 Variablen-Beschreibungen Speicherfunktionen

### 14.7.1 Steuerung Speicherfunktionen

Über die nachfolgend beschriebenen Variablen werden Speicherfunktionen gesteuert.

<b>Store Parameters</b>		Index: 1010, Kurzname: StoreParm
<b>SPP Windows</b>	–	
<b>Typ</b>	Array, 4 Elemente (Subindex 0 .. 4), Typ Unsigned32	
<b>Array-Element 1</b>	save all	
<b>Array-Element 2</b>	save communication (nicht unterstützt)	
<b>Array-Element 3</b>	save application (nicht unterstützt)	
<b>Array-Element 4</b>	save manufacturer (nicht unterstützt)	
<b>Zugriff</b>	Lesen und Schreiben	
<b>Art</b>	Steuer-Information	

Die Funktion „Save all“ (Subindex 1) wird unterstützt. Durch Eingabe eines bestimmten Codes in Array-Element 2 (ASCII: e v a s) werden alle Maschinendaten, das Teileprogramm und die Anwender-Variablen des Servoreglers im nichtflüchtigen Speicher des Servoreglers (Servo-EEPROM) abgespeichert. Teil der Maschinendaten sind die Maschinendaten CANopen (Art = MC) aus dieser Betriebsanleitung.



Sobald die Maschinendaten CANopen durch den entsprechenden Eintrag in diese Variable im Servo-EEPROM gespeichert sind, werden bei

- Einschalten der Steuerspannung des Servoreglers
- NMT-Service Reset\_Communication (nur CANopen-Parameter)
- NMT-Service Reset\_Node (alle Parameter)

die im Servo-EEPROM gespeicherten Werte übernommen. Die im Standard-Lieferzustand aktiven Werte des Predefined Connection Set (Standardeinstellungen oder Default genannt) werden dann nicht mehr verwendet.



Durch einen bestimmten Code, der im Folgenden bei der Variablen *Restore Default Parameters* beschrieben ist, können die Werte des Predefined Connection Set wieder aktiviert werden.

<b>Restore Default Parameters</b>		Index: 1011, Kurzname: RestoreParm
<b>SPP Windows</b>	–	
<b>Typ</b>	Array, 4 Elemente (Subindex 0 .. 4), Typ Unsigned32	
<b>Array-Element 1</b>	restore all (nicht unterstützt)	
<b>Array-Element 2</b>	restore communication	
<b>Array-Element 3</b>	restore application (nicht unterstützt)	
<b>Array-Element 4</b>	restore manufacturer (nicht unterstützt)	
<b>Zugriff</b>	Lesen und Schreiben	
<b>Art</b>	Steuer-Information	

Die Funktion „Restore Communication Default“ (Subindex 2) wird unterstützt. Durch Eingabe eines bestimmten Codes in Array-Element 2 (ASCII: d a o l) können die Standardeinstellungen (Standardwerte) aller CANopen-Variablen wieder hergestellt werden.

- Die Standardeinstellungen (Default-Werte) für die COB-IDs finden Sie in der Tabelle „Predefined Connection Set“ in Abschnitt 6.2 (Seite 20). Sie können auch den Variablen-Beschreibungen entnommen werden.
- Die Standardeinstellungen (Default-Werte) der übrigen CANopen-Maschinendaten finden Sie in den jeweiligen Variablen-Beschreibungen.



Damit die Standardeinstellungen tatsächlich wirksam werden, muss nach dem Eintrag in diese Variable noch der NMT-Service „Reset\_Node“ ausgeführt werden.



Falls die wieder hergestellten Standardeinstellungen dauerhaft gespeichert werden sollen, muss abschließend noch ein Speichern durch Zugriff auf die Variable *Store Parameters* durchgeführt werden.

## 14.8 Variablen-Beschreibung Geräteinformationen

<b>Device Type</b>		Index: 1000, Kurzname: DeviceType
<b>SPP Windows</b>	–	
<b>Typ</b>	Simple-Variable, Unsigned32	
<b>Zugriff</b>	nur Lesen	
<b>Art</b>	Konstante (Variablen-Wert ändert sich nicht)	
<b>Standardwert</b>	Geräteprofil: CiA 402, Antriebstyp: Servoregler (00020192 <sub>hex</sub> )	

Über die Variable *Device Type* kann der Gerätetyp ausgelesen werden.

Der hexadezimale Wert 20192 dieser Variablen beschreibt gemäß CANopen-Kommunikationsprofil CiA 301 und CANopen-Antriebsprofil CiA 402 den Gerätetyp und seine Funktion:

Bit	Bedeutung
0..15	Geräteprofil: 192 <sub>hex</sub> = 402 <sub>dez</sub> = CiA 402
16..23	Antriebstyp: 2 = Servoregler
24..31	immer 0

---

**Manufacturer Device Name**

Index: 1008 Kurzname: DeviceName

---

<b>SPP Windows</b>	–
<b>Typ</b>	Simple-Variable, Visible-String, Länge 16
<b>Zugriff</b>	nur Lesen
<b>Art</b>	Konstante (Variablen-Wert ändert sich nicht)
<b>Standardwert</b>	–

---

Über die Variable *Manufacturer Device Name* kann der Gerätenamen ausgelesen werden.

---

**Manufacturer Hardware Version**

Index: 1009, Kurzname: HWVersion

---

<b>SPP Windows</b>	–
<b>Typ</b>	Simple-Variable, Visible-String, Länge 16
<b>Zugriff</b>	nur Lesen
<b>Art</b>	Konstante (Variablen-Wert ändert sich nicht)
<b>Standardwert</b>	–

---

Über die Variable *Manufacturer Hardware Version* kann die Hardware-Version des Servoreglers ausgelesen werden.

---

**Manufacturer Software Version**

Index: 100a, Kurzname: SWVersion

---

<b>SPP Windows</b>	–
<b>Typ</b>	Simple-Variable, Visible-String, Länge 16 Bit
<b>Zugriff</b>	nur Lesen
<b>Art</b>	Konstante (Variablen-Wert ändert sich nicht)
<b>Standardwert</b>	–

---

Über die Variable *Manufacturer Software Version* kann die Software-Version des Servoreglers ausgelesen werden.

---

<b>Identity</b>	Index: 1018, Kurzname: Identity
<b>SPP Windows</b>	–
<b>Typ</b>	Array, 4 Elemente (Subindex 0 .. 4), Typ unsigned32
<b>Array-Element 1</b>	Vendor-ID (15)
<b>Array-Element 2</b>	Produkt-Code (6755 bzw. 6745)
<b>Array-Element 3</b>	Revisionsnummer
<b>Array-Element 4</b>	Seriennummer
<b>Zugriff</b>	nur Lesen
<b>Art</b>	Konstante (Variablenwert ändert sich nicht)

---

Über die Variable *Identity* kann der Hersteller des Servoreglers ausgelesen werden.

# Anhang

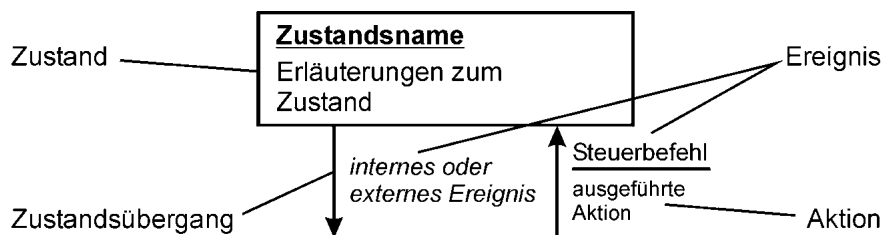
## Anhang A Zustandsmaschinen

Zustandsmaschinen (auch Zustandsautomaten oder State Machines genannt) beschreiben das Verhalten von Systemen. Die grafische Darstellung einer Zustandsmaschine wird Zustandsdiagramm genannt.

Die Elemente einer Zustandsmaschine sind

- Zustände
- Zustandsübergänge
- Ereignisse
- Aktionen

Diese Darstellung der Elemente einer Zustandsmaschine im Zustandsdiagramm zeigt das folgende Bild:



BN 2555.0000.00B.050

Bild 2: Elemente von Zustandsmaschinen

**Zustände** (als Rechtecke mit dem Namen des Zustandes und evtl. weiteren Erläuterungen dargestellt) können durch **Zustandsübergänge** (als Pfeile dargestellt) gewechselt werden. Ein Zustandsübergang erfolgt, wenn ein **Ereignis** auftritt. Beim Zustandsübergang wird eine **Aktion** ausgeführt (unterhalb des Ereignisses dargestellt und durch einen Strich davon getrennt). Es sind auch Zustandsübergänge erlaubt, bei denen keine Aktion ausgeführt wird.

Bei der Anwendung von Zustandsmaschinen werden in dieser Beschreibung folgende Arten von Ereignissen unterschieden:

- Steuerbefehle
- interne oder externe Ereignisse.

**Steuerbefehle** sind Ereignisse, die der Anwender auslösen kann, z. B. indem er vom NMT-Master her einen NMT-Service ausführen lässt.



**Interne oder externe Ereignisse** (kursiv dargestellt) werden vom Servoregler bzw. einer an seine Schnittstellen angeschlossenen Komponente ausgelöst. Es kann sich dabei z. B. um das Einschalten der Steuerspannung oder den Abschluss der Initialisierung handeln.

## Anhang B Variablenliste

Hier finden Sie einen Überblick über sämtliche CANopen-Variablen der Servoregler, sortiert nach Index.

Erläuterungen zu den einzelnen Spalten finden Sie in Abschnitt 14 (Seite 32).

Index	Name	Kurzname	Obj.-Code Elemente	Datentyp Länge	R/W	Art	Einheit
1000	Device Type	DeviceType	Var	u32	R	F	-
1001	Error Register	ErrorReg	Var	u8	R	S	-
1003	Pre-Defined Error Field	ErrorField	Array 1	u32	R	S	-
1005	COB-ID SYNC Message	IdSync	Var	u32	R/W	MC	-
1008	Manufacturer Device Name	Device Name	Var	VisStr 16	R	F	-
1009	Manufacturer Hardware Version	HWVersion	Var	VisStr 16	R	F	-
100a	Manufacturer Software Version	SWVersion	Var	VisStr 16	R	F	-
100b	NodeID	NodeID	Var	32 Bit	R	V	-
100c	Guard Time	GuardTime	Var	u16	R/W	MC	ms
100d	Life Time Factor	LifeTimeFac	Var	u8	R/W	MC	-
1010	Store Parameters	StoreParm	Array 4	u32	R/W	S	-
1011	Restore Default Parameters	RestoreParm	Array 4	u32	R/W	S	-
1014	COB-ID Emergency Message	IDEmcy	Var	u32	R/W	MC	-
1015	Inhibit Time Emergency Message	InhibitTimeEmcy	Var	i16	R/W	MC	ms
1017	Producer Heartbeat Time	ProducerHeartbeatTime	Var	u16	R/W	MC	ms
1018	Identity	Identity	Array 4	u32	R	F	-
1200	Server SDI 2 Parameter	SSD01	Record 2	SDO Par.	R	V	-
1201	Server SDO 2 Parameter	SSD01	Record 3	SDO Par.	R/W	V	-
1400	Receive PDO 1 Comm. Parameter	RPD01ComParm	Record 2	PDO CommPar	R/W	MC	-
1401	Receive PDO 2 Comm. Parameter	RPD02ComParm	Record 2	PDO CommPar	R/W	MC	-
1402	Receive PDO 3 Comm. Parameter	RPD03ComParm	Record 2	PDO CommPar	R/W	MC	-
1403	Receive PDO 4 Comm. Parameter	RPD04ComParm	Record 2	PDO CommPar	R/W	MC	-
1600	Receive PDO 1 Mapping	RPD01Mapping	Record 8	PDOMap	R/W	MC	-
1601	Receive PDO 2 Mapping	RPD02Mapping	Record 8	PDOMap	R/W	MC	-
1602	Receive PDO 3 Mapping	RPD03Mapping	Record 8	PDOMap	R/W	MC	-
1603	Receive PDO 4 Mapping	RPD04Mapping	Record 8	PDOMap	R/W	MC	-
1800	Transmit PDO 1 Comm. Parameter	TPD01ComParm	Record 5	PDO CommPar	R/W	MC	-
1801	Transmit PDO 2 Comm. Parameter	TPD02ComParm	Record 5	PDO CommPar	R/W	MC	-
1802	Transmit PDO 3 Comm. Parameter	TPD03ComParm	Record 5	PDO CommPar	R/W	MC	-
1803	Transmit PDO 4 Comm. Parameter	TPD04ComParm	Record 5	PDO CommPar	R/W	MC	-
1a00	Transmit PDO 1 Mapping	TPD01Mapping	Record 8	PDOMap	R/W	MC	-
1a01	Transmit PDO 2 Mapping	TPD02Mapping	Record 8	PDOMap	R/W	MC	-
1a02	Transmit PDO 3 Mapping	TPD03Mapping	Record 8	PDOMap	R/W	MC	-
1a03	Transmit PDO 4 Mapping	TPD04Mapping	Record 8	PDOMap	R/W	MC	-
1f80	NMT Startup	NMTStartup	Var	u32	R/W	V	-
5fb4	Node-ID	CANNodeId	Array 2	u8	R	V	-
5fb5	CAN Baudrate	CANBaudrate	Var	u8	R/W	V	Kbps
5fc1	Alternat. Interpol. Data Record	AltIpolDataRecord	Record 2	i16/i8	R/W	MC	LgS
5fc2	Alternative für Velocity Offset	AltVelocityOffset	Record 2	i16/i8	R/W	MC	-
6007	Abort Connection Option Code	AbortOptionCode	Var	i16	R/W	ME	-

Index Name	Kurzname	Obj.-Code Elemente	Datentyp Länge	R/W	Art	Einheit
60b1	Velocity Offset	VelocityOffset	Var	i32	R/W	ME min <sup>-1</sup>
60c0	Interpolation Sub Mode Select	IpolSubmodeAuswahl	Var	i16	R/W	MC -
60c1	Interpolation Data Record	IpolDataRecord	Record 1	i32	R/W	MC LgS
60c2	Interpolation Time Period	IpolTimePeriod	Record 2	IpolTime	R/W	MC -
60c3	Interpolation Sync Definition	IpolSyncDefinition	Array 2	u8	R/W	MC -
60c4	Interpolation Data Configuration	IpolDataConfig	Record 6	IpolConf	R/W	MC -

## Anhang C Projektierungshilfe

Diese Projektierungshilfe dient zur Einstellung der CANopen-Schnittstelle. Kopieren Sie sich diese Projektierungshilfe je Servoregler einmal. Füllen Sie die Blätter aus und nehmen Sie erst dann die Einstellungen am Gerät vor. Zuerst müssen die Hardware-Einstellungen vorgenommen werden, dann können die Maschinendaten der CANopen-Schnittstelle eingestellt werden.

Achten Sie darauf, dass Sie die Einstellungen der Maschinendaten der CANopen-Schnittstelle mit der CANopen-Konfigurationssoftware auf Diskette/Festplatte und im Servoregler speichern, siehe Abschnitt 12 (Seite 31) und/oder die Betriebsanleitung der verwendeten CANopen-Konfigurationssoftware.

### Daten der Anwendung

Bezeichnung der Anwendung: \_\_\_\_\_

(z. B. Palettiermaschine)

Bezeichnung der Achse: \_\_\_\_\_

(z. B. X-Achse)

Erstellt, geändert (Name, Datum): \_\_\_\_\_

### Hardware-Einstellungen

Node-ID hexadezimal: \_\_\_\_\_

Die Node-ID wird auch verwendet, um die COB-ID zu bestimmen. Folgende COB-ID dürfen nicht verändert werden, um Probleme innerhalb des CAN-Netzwerks zu vermeiden:

COB-ID NMT (= 0), COB-ID SDO Tx (= 580 + Node-ID), COB-ID SDO Rx (= 600 + Node-ID).

Übertragungsrate: \_\_\_\_\_

Zulässige Werte [kBit/s]: 10, 20, 50, 125, 250, 500, 800, 1000. Bitte beachten Sie auch die maximal zulässigen Buslängen, siehe Abschnitt 5.1.2 (Seite 16).

**Maschinendaten PDOs (Prozessdatenkanal): Receive PDOs**

Index	Name	Standardwert	Einzustellender Wert (falls nicht Standard)
1600	Receive PDO 1 Mapping	Anzahl Einträge = 1	
		60400010	
		00000000	
		00000000	
		00000000	
		00000000	
		00000000	
		00000000	
		00000000	
1601	Receive PDO 2 Mapping	Anzahl Einträge = 2	
		60400010	
		60600010	
		00000000	
		00000000	
		00000000	
		00000000	
		00000000	
		00000000	
1602	Receive PDO 3 Mapping	Anzahl Einträge = 2	
		60400010	
		607A0020	
		00000000	
		00000000	
		00000000	
		00000000	
		00000000	
		00000000	
1603	Receive PDO 4 Mapping	Anzahl Einträge = 2	
		60400010	
		60810020	
		00000000	
		00000000	
		00000000	
		00000000	
		00000000	
		00000000	

**Maschinendaten PDOs (Prozessdatenkanal): Transmit PDOs**

Index	Name	Standardwert	Einzustellender Wert (falls nicht Standard)
1a00	Transmit PDO 1 Mapping	Anzahl Einträge = 1	
		60410010	
		00000000	
		00000000	
		00000000	
		00000000	
		00000000	
		00000000	
		00000000	
1a01	Transmit PDO 2 Mapping	Anzahl Einträge = 2	
		60410010	
		60610010	
		00000000	
		00000000	
		00000000	
		00000000	
		00000000	
		00000000	
1a02	Transmit PDO 3 Mapping	Anzahl Einträge = 2	
		60410010	
		60640020	
		00000000	
		00000000	
		00000000	
		00000000	
		00000000	
		00000000	
1a03	Transmit PDO 4 Mapping	Anzahl Einträge = 2	
		60410010	
		606C020	
		00000000	
		00000000	
		00000000	
		00000000	
		00000000	
		00000000	

**Maschinendaten PDOs (Prozessdatenkanal): Comm. Parameter**

Index	Name	Standardwert	Einzustellender Wert (falls nicht Standard)
1400	Receive PDO 1 Comm. Parameter	valid (0)	
		200 + Node-ID = ____ (hexadezimal)	
		asynchron (255)	
1401	Receive PDO 2 Comm. Parameter	not valid (1)	
		300 + Node-ID = ____ (hexadezimal)	
		asynchron (255)	
1402	Receive PDO 3 Comm. Parameter	not valid (1)	
		400 + Node-ID = ____ (hexadezimal)	
		asynchron (255)	
1403	Receive PDO 4 Comm. Parameter	not valid (1)	
		500 + Node-ID = ____ (hexadezimal)	
		asynchron (255)	
1800	Transmit PDO 1 Comm. Parameter	valid (0)	
		180 + Node-ID = ____ (hexadezimal)	
		asynchron (255)	
		Inhibit time = 0 ms	
		Event timer = 0 ms	
1801	Transmit PDO 2 Comm. Parameter	not valid (1)	
		280 + Node-ID = ____ (hexadezimal)	
		asynchron (255)	
		Inhibit time = 0 ms	
		Event timer = 0 ms	
1802	Transmit PDO 3 Comm. Parameter	not valid (1)	
		380 + Node-ID = ____ (hexadezimal)	
		asynchron (255)	
		Inhibit time = 0 ms	
		Event timer = 0 ms	
1803	Transmit PDO 4 Comm. Parameter	not valid (1)	
		480 + Node-ID = ____ (hexadezimal)	
		asynchron (255)	
		Inhibit time = 0 ms	
		Event timer = 0 ms	

**Maschinendaten Synchronisation**

Index	Name	Standardwert	Einzustellender Wert (falls nicht Standard)
1005	COB-ID Sync Message	80 (hexadezimal)	

**Maschinendaten Node Guarding, Life Guarding**

Index	Name [Einheit]	Standardwert	Einzustellender Wert (falls nicht Standard)
100c	Guard Time [ms]	ausgeschaltet (0)	
100d	Life Time Factor [Vielfache der Guard Time]	ausgeschaltet (0)	
1017	Producer Heartbeat Time	0 ms	

**Maschinendaten Interpolated Position Mode**

Index	Name	Standardwert	Einzustellender Wert (falls nicht Standard)
60c2	Interpolation Time Period	Time Units: $4 \cdot 10^{-3}$ s	



## Anhang D Störungscode

Beim Auftreten eines Fehlers geht der Geräte-Zustand (angezeigt im Achsen-Statuswort) in den Zustand „Störung“ über und der entsprechende Achsen-Störungscode wird angezeigt. Nachdem die Fehlerursache beseitigt wurde, kann der Fehlerzustand durch einen Übergang des Bits „Reset Störung“ von 0 nach 1 im Achsen-Steuerwort wieder zurückgesetzt werden.

Es können grob zwei Arten von Störungen unterschieden werden:

- Störungen des Antriebs, die in der Betriebsanleitung 6710.101 „Funktionen und Parameter“ dokumentiert sind, und die auch in der Antriebsvariablen *Achsen-Störungscode* angezeigt werden.

Siehe Betriebsanleitung 6710.101 „Funktionen und Parameter“, Abschnitt „Achsen-Störungscode“.

- Störungen der CANopen-Schnittstelle oder des CAN-Controllers.

Im Folgenden werden die möglichen CANopen-Störungscode (hexadezimal) mit Fehlerbezeichnung aufgelistet.

### 8110 Message-Verlust im CAN-Controller

Diese Störung wird nur im CAN-System verarbeitet, sie wird nicht an die Firmware des Servoreglers gemeldet.

### 8120 Störung CAN-Bus, Warnungsschwelle erreicht

Diese Störung wird nur im CAN-System verarbeitet, sie wird nicht an die Firmware des Servoreglers gemeldet.

### 8130 Life Guarding / Heartbeat Error

Diese Störung wird auch als Störungscode an die Firmware des Servoreglers gemeldet und führt zum Abschalten des Antriebs.

### 8140 Bus OFF Zustand verlassen

Diese Störung wird nur im CAN-System verarbeitet, sie wird nicht an die Firmware des Servoreglers gemeldet.

### 8150 COB-ID Kollision

Diese Störung wird auch als Störungscode an die Firmware des Servoreglers gemeldet und führt zum Abschalten des Antriebs.

### 8180 Störung, Sende- und/oder Empfangsfehler zu stark (CAN-Begriff: Bus-Off); Gerät hat sich vom CAN-Bus abgekoppelt, Antrieb wurde über Schnellhalt-Befehl angehalten, Näheres siehe Abschnitt 11.4 (Seite 30).

8210 Empfangenes PDO zu kurz

Diese Störung wird nur im CAN-System verarbeitet, sie wird nicht an die Firmware des Servoreglers gemeldet.

8220 Empfangenes PDO zu lang

Diese Störung wird nur im CAN-System verarbeitet, sie wird nicht an die Firmware des Servoreglers gemeldet.

## Anhang E Stichwortverzeichnis

Im Stichwortverzeichnis wurden Gruppen gebildet, um zusätzlich zum direkten Eintrag in alphabetischer Reihenfolge einen besseren Überblick über bestimmte Stichworte zu geben:

**Variablen** Über die Bezeichnung einer Variablen finden Sie die Variablen-Beschreibung der zugehörigen Variablen.

**Variablen-Kurznamen** Über den Kurznamen einer Variablen finden Sie die Variablen-Beschreibung der zugehörigen Variablen.

Daneben wurden weitere Gruppen gebildet, die Zusammenhänge veranschaulichen sollen.

Abbildung (von Variablen auf PDOs) .....	22
Abbildungs-PDOs .....	37
Abschlusswiderstand .....	16
Aktion (Zustandsmaschine) .....	55
Anschluss Bus .....	15
Anschluss und Inbetriebnahme .....	15
Anschlusstecker .....	16
Antriebs-Variablen .....	7
Array (Objekt-Code einer Variablen) .....	33
Array-Element (einer Variablen) .....	34
Art (einer Variablen) .....	34
asynchron (Übertragungsart eines PDO) .....	25
Bedien- und Inbetriebnahmeprogramm SPP Windows .....	7
Speicherfunktion .....	31
Begriffe und Abkürzungen .....	11
CAL .....	11
CAN .....	11
CAN-Controller .....	11
CANopen .....	11
CiA .....	11
CMS .....	12
COB .....	12
COB-ID .....	12
DBT .....	12
DRIVECOM .....	12
EDS .....	12
EMCY .....	12
NMT .....	13
Node-ID .....	13
PDO .....	13
Profil .....	13
RPDO .....	13
SDO .....	13
SYNC .....	13
TPDO .....	13
Variable .....	14
Boolean (Datentyp einer Variablen) .....	33
Bus-Off (CAN-Controller) .....	16, 30, 64
Busanschluss .....	15
Buslängen .....	16
Busleitungen .....	16

CAL (Abkürzung) .....	11
CAN (Abkürzung) .....	11
CAN-Controller .....	11, 30
CANopen	
Einführung .....	11
CANopen (Begriff) .....	11
CANopen-Antriebsprofil CiA 402 .....	6
CANopen-Kommunikationsprofil CiA 301 .....	6, 11
CANopen-Variablen .....	7
CiA (Abkürzung) .....	11
Client .....	21
CMS (Abkürzung) .....	12
COB (Abkürzung) .....	12
COB-ID (Abkürzung) .....	12
COB-ID Distribution (CANopen-Funktion) .....	10
COB-ID-Verteilung .....	20
DBT (Abkürzung) .....	12
Default (Eigenschaft einer Variablen) .....	52
Device Profile (CANopen-Funktion) .....	10
Drehcodierschalter .....	15
DRIVECOM (Begriff) .....	12
EDS (Electronic Data sheet) .....	12
Einführung in CANopen .....	11
Electronic Data sheet .....	12
Element (einer Variablen) .....	33
EMCY (Abkürzung) .....	12
Emergency Message (CANopen-Funktion) .....	10
Enter_Preoperational_State (NMT-Service) .....	19
Ereignis (Zustandsmaschine) .....	55
Error Active (CAN-Controller) .....	16
Error Passive (CAN-Controller) .....	16
Event Timer .....	25
Extended Boot-up (CANopen-Funktion) .....	10
Fehler, Sende- und Empfangs- .....	30
Funktionen und Parameter .....	6
Funktionsbausteine .....	8
Gefahr (Sicherheitshinweise) .....	9
Geräteinformationen .....	31
Heartbeat .....	30
Hinweise (siehe Sicherheitshinweise) .....	9
i16 (Datentyp einer Variablen) .....	33
i32 (Datentyp einer Variablen) .....	33
i8 (Datentyp einer Variablen) .....	33
Inbetriebnahme .....	17, 59
Inbetriebnahme des CANopen-Systems .....	17
Parametrierung der Antriebsfunktionen .....	17
Projektierung der CANopen-Schnittstelle .....	17
Inbetriebnahme des CANopen-Systems .....	17
Index (einer Variablen) .....	32
Inhibit Time (eines PDO) .....	23
Integer16 (Datentyp einer Variablen) .....	33
Integer32 (Datentyp einer Variablen) .....	33
Integer8 (Datentyp einer Variablen) .....	33
Interpolated Position Mode .....	26
aktivieren .....	27
deaktivieren .....	28
Kurzname (einer Variablen) .....	32
Längen, Bus- .....	16
Leitungen, Bus- .....	16

Leuchtdioden	16
Aux1	16
Aux2	16
Error	16
Run	16
Life Guarding	29
Life Guarding (CANopen-Funktion)	10
Mapping	
variables	40
Mapping (von Variablen auf PDOs)	22
Maschinendaten CANopen	34
Interpolated Position Mode	47
Node Guarding, Life Guarding	50
Störungsbehandlung	49
Synchronisation	45
Maschinendaten PDOs	40
Minimum Boot-up (CANopen-Funktion)	10
Name (einer Variablen)	32
Netzwerkmanagement	18
NMT (Abkürzung)	13
NMT Node State Diagram	18
Leuchtdiode	16
NMT-Master (CANopen-Funktion)	10
NMT-Services	18, 19
NMT-Slave (CANopen-Funktion)	10
Node Guarding	29
Node-ID (Abkürzung)	13
Node-ID Distribution (CANopen-Funktion)	10
Objekt-Code (einer Variablen)	33
OctStr (Datentyp einer Variablen)	33
Parameter	21
Parametrierung der Antriebsfunktionen	17
PDO	22
Abbildung	22
asynchron	25
COB-ID	20
Inhibit Time	23
Kommunikation	22
Mapping	22
synchron	25
Transmission Type	25
Übertragungsart	25
PDO (Abkürzung)	13
PDO CommPar (Datentyp eines Records)	33
PDO CoP (Datentyp eines Records)	33
PDO Mapping	37
PDO Mapping, variable (CANopen-Funktion)	10
PDO Modes (CANopen-Funktion)	10
PDO-Kommunikation	22
PDO-Kommunikation (Variablenbeschreibung)	37
PDOs, No. of (CANopen-Funktion)	10
Predefined Connection Set	20
Profil (Begriff)	13
Projektierung	59
Projektierung der CANopen-Schnittstelle	17
Projektierungshilfe	59
Prozessdaten	22
Prüfen (Hinweis)	9
R (Zugriff auf eine Variable)	34

Receive PDOs Comm Par	40
Receive PDOs Mapping	37
Record (Objekt-Code einer Variablen)	33
Record-Element (einer Variablen)	34
Reset_Communication (NMT-Service)	19
Reset_Node (NMT-Service)	19
Restore Default Parameters (Variable)	52
RPDO (Abkürzung)	13
SDO (Abkürzung)	13
SDO Par (Datentyp eines Records)	33
SDO Parameter (Datentyp eines Records)	33
SDO-Kommunikation	19, 21
SDO-Kommunikation (Variablenbeschreibung)	36
SDOs, No. of (CANopen-Funktion)	10
Sende- und Empfangsüberwachung	30
Leuchtdiode	16
Server	21
Sicherheitshinweise	9
Achtung	9
Gefahr	9
Prüfen	9
Tipp	9
Simple Variable (Objekt-Code einer Variablen)	33
Speicherfunktionen	31
Bedien- und Inbetriebnahmeprogramm SPP Windows	31
SPP Windows	7, 31, 32
Standardeinstellung (einer Variablen)	52
Start_Remote_Node (NMT-Service)	19
Steuerbefehle (Zustandsmaschine)	55
Stop_Remote_Node (NMT-Service)	19
Store Parameters (Variable)	51
Störungsbehandlung	29
Subindex (einer Variablen)	33
SYNC (Abkürzung)	13
SYNC-Message	25
synchron (Übertragungsart eines PDO)	25
Synchronisation	25
Technische Daten	10
Terminierung des Busses	16
Time Stamp (Synchronisation)	25
Tipp (Hinweis)	9
TPDO	13
Transmission Type (eines PDO)	25
Transmit PDOs Comm Par	42
Transmit PDOs Mapping	38
Typ (einer Variablen)	33
u16 (Datentyp einer Variablen)	33
u32 (Datentyp einer Variablen)	33
u8 (Datentyp einer Variablen)	33
Übertragungsart (eines PDO)	25
Übertragungsraten	10
Überwachung, Sende- und Empfangs-	30
Leuchtdiode	16
Überwachungsmechanismen	29
Unsigned16 (Datentyp einer Variablen)	33
Unsigned32 (Datentyp einer Variablen)	33
Unsigned8 (Datentyp einer Variablen)	33
Variable (Begriff)	14
Variablen	
Abort Connection Option Code	49

Alternative für Velocity Offset	46
Alternativer Interpolation Data Record	45
CAN Baudrate	35
COB-ID Emergency Message	49
COB-ID SYNC Message	45
Device Type	52
Error Register	48
Guard Time	50
Identity	54
Inhibit Time Emergency Message	49
Interpolation Data Configuration	48
Interpolation Data Record	45
Interpolation Sub Mode Select	47
Interpolation Sync Definition	47
Interpolation Time Period	47
Life Time Factor	50
Manufacturer Device Name	53
Manufacturer Hardware Version	53
Manufacturer Software Version	53
NMT Startup	36
Node-ID	35
Pre-Defined Error Field	48
Producer Heartbeat Time	50
Receive PDO 1 Comm. Parameter	41
Receive PDO 1 Mapping	37
Receive PDO 2 Comm. Parameter	41
Receive PDO 2 Mapping	38
Receive PDO 3 Comm. Parameter	41
Receive PDO 3 Mapping	38
Receive PDO 4 Comm. Parameter	42
Receive PDO 4 Mapping	38
Restore Default Parameters	52
Server SDO 1 Parameter	36
Server SDO 2 Parameter	36
Store Parameters	51
Transmit PDO 1 Comm. Parameter	43
Transmit PDO 1 Mapping	39
Transmit PDO 2 Comm. Parameter	44
Transmit PDO 2 Mapping	39
Transmit PDO 3 Comm. Parameter	44
Transmit PDO 3 Mapping	39
Transmit PDO 4 Comm. Parameter	44
Transmit PDO 4 Mapping	40
Velocity Offset	46
Variablen-Arten	34
Variablen-Beschreibungen	32
Variablen-Gruppen	
Geräteinformationen	52
Interpolated Position Mode	45
Netzwerk-Management	35
PDO-Kommunikation	37
SDO-Kommunikation	36
Speicherfunktionen	51
Synchronisation	45
Überwachung	48
Variablen-Kurznamen	
AbortOptionCode	49
AltIpolDataRecord	45
AltVelocityOffset	46
CANBaudrate	35

CANNodeId	35
DeviceName	53
DeviceType	52
ErrorField	48
ErrorReg	48
GuardTime	50
HWVersion	53
IdEmcy	49
Identity	54
IdSync	45
InhibitTimeEmcy	49
IpolDataConfig	48
IpolDataRecord	45
IpolSubmodeAuswahl	47
IpolSyncDefinition	47
IpolTimePeriod	47
LifeTimeFac	50
NMTStartup	36
NodId	35
ProducerHeartbeatTime	50
RestoreParm	52
RPDO1ComParm	41
RPDO1Mapping	37
RPDO2ComParm	41
RPDO2Mapping	38
RPDO3ComParm	41
RPDO3Mapping	38
RPDO4ComParm	42
RPDO4Mapping	38
SSDO1	36
SSDO2	36
StoreParm	51
SWVersion	53
TPDO1ComParm	43
TPDO1Mapping	39
TPDO2ComParm	44
TPDO2Mapping	39
TPDO3ComParm	44
TPDO3Mapping	39
TPDO4ComParm	44
TPDO4Mapping	40
VelocityOffset	46
Variablenname	32
VisStr (Datentyp einer Variablen)	33
W (Zugriff auf eine Variable)	34
Zugriff (auf eine Variable)	34
Zustand (Zustandsmaschine)	55
Zustandsautomat	55
Zustandsdiagramm	55
Zustandsmaschine	
Aktion	55
Ereignis	55
Steuerbefehl	55
Zustand	55
Zustandsübergang	55
Zustandsübergang (Zustandsmaschine)	55