











Sendix[®] absolut Absoluter Singleturn Drehgeber

Serie M3658, M3678

-  Safety-Lock™
-  High rotational speed
-  Temperature
-40° + 85°
-  High IP
-  High shaft load capacity
-  Shock/vibration resistant
-  Short-circuit proof
-  Reverse polarity protection

CANopen
certified
CiA200704-301V402/20-0071



© Fritz Kübler GmbH

Urheberrechtsschutz

Für diese Dokumentation besteht Urheberrechtsschutz durch die Firma Fritz Kübler GmbH. Diese Dokumentation darf ohne vorherige schriftliche Zustimmung der Firma Fritz Kübler GmbH weder abgeändert, erweitert oder vervielfältigt noch an Dritte weitergegeben werden.

Änderungsvorbehalt

Technische Änderungen der in dem vorliegenden Dokument enthaltenen technischen Informationen, die aus dem stetigen Bestreben zur Verbesserung unserer Produkte resultieren, behalten wir uns jederzeit vor.

Verzicht auf Garantie

Die Fritz Kübler GmbH übernimmt in Bezug auf das gesamte Handbuch keine Garantie, weder stillschweigend noch ausdrücklich und haftet weder für direkte noch indirekte Schäden.

Dokumentationinformation

Ausgabestand 1-2009

verwendete Bildschirmausdrucke

CANalyzer[®] Fa. Vector –Informatik

Fritz Kübler GmbH Schubertstr.47

78054 VS-Schwenningen / Germany

Tel. +49 (0) 7720-3903-0

Fax +49 (0) 7720-21564

E-Mail: info@kuebler.com

Internet: www.kuebler.com

Inhaltsangabe

1	ALLGEMEINES - EIGENSCHAFTEN	1-5
	CANOPEN SINGLETURN DREHGEBER SERIE M36X8.....	1-5
	DAS CANOPEN COMMUNICATION PROFIL DS 301 V4.02.....	1-5
	DAS DREHGEBER GERÄTEPROFIL DS 406 V3.1	1-6
	LSS DIENSTE DS 305 V1.0.....	1-6
	DATENÜBERTRAGUNG	1-6
	OBJEKTE UND FUNKTIONSCODE IM PREDEFINED CONNECTION SET	1-7
	BROADCAST (NETZWERKWEITE) OBJEKTE.....	1-7
	PEER-TO PEER (GERÄT-ZU GERÄT)OBJEKTE	1-7
	EINGESCHRÄNKTE, RESERVIERTE OBJEKTE.....	1-7
	ÜBERTRAGUNG DER PROZESS DATEN	1-8
	TRANSMIT PDO 1 (1800H) POSITION ASYNCHRON	1-8
	TRANSMIT PDO2 (1801H) GESCHWINDIGKEIT ASYNCHRON	1-8
	TRANSMIT PDO3 (1802H) POSITION SYNCHRON	1-8
	ÜBERTRAGUNG DER SERVICE DATEN	1-9
	BEISPIEL EINER ÜBERTRAGUNG VON SERVICE-DATEN ZU UND VOM DREHGEBER	1-10
	LSS DIENSTE.....	1-10
2	VERSORGUNGSSPANNUNG UND CAN-BUS ANSCHLUSS ..	2-11
	CAN-KABELBELEGUNG.....	2-11
	CANOPEN M12-STECKERBELEGUNG.....	2-11
	MECHANISCHE KENNWERTE.....	2-11
3	ERST-INBETRIEBNAHME - GENERELLE EINSTELLUNGEN AM GERÄT	3-12
	BAUDRATE.....	3-12
	KNOTENNUMMER.....	3-12
	TERMINIERUNG.....	3-12
	SAVE ALL BUS PARAMETERS	3-12
4	LAYER SETTING SERVICES (LSS)	4-13
5	DEFAULTEINSTELLUNGEN BEI AUSLIEFERUNG	5-15
	KOMMUNIKATIONSPARAMETER.....	5-15
	ENCODER PROFIL.....	5-16
6	KOMMUNIKATIONSPARAMETER	6-17
	DEFINITION DES ÜBERTRAGUNGSTYP (TRANSMISSION TYPE) DES PDO's.....	6-19
	VARIABLES PDO MAPPING.....	6-19
7	BEISPIEL EINES VARIABLEN MAPPINGEINTRAGES	7-20
8	STRUKTUR EINES MAPPING EINTRAGES	8-21
9	BEISPIEL EINER APPLIKATIONS-PROGRAMMIERUNG:	9-22
	OBJEKTE EINRICHTEN	9-22
	<i>Measuring Units per Revolution auf 3600 begrenzen</i>	9-22
	<i>Preset value auf 0</i>	9-23
	<i>Transmit Parameter TPDO1 und TPDO2 Werte setzen</i>	9-23
	<i>Producer Heartbeat auf 500 ms setzen</i>	9-24
	<i>Work area Low- und High-Limit Werte setzen</i>	9-24
	<i>alle geänderten Parameter im EEprom speichern Store Parameters 1010h</i>	9-25
	<i>Objekt 1010h Parameter abspeichern</i>	9-25
	<i>Objekt 1011h: Standard-Werte laden</i>	9-25
	KOMMUNIKATIONSPROFIL – WEITERE OBJEKTE	9-26
	OBJEKT 1018H: IDENTITY OBJECT	9-26
10	EMERGENCY OBJEKTE	10-27
	UNTERSTÜTZTE ERROR CODES	10-27
11	EMERGENCY NACHRICHT	11-28
	BEISPIEL EINER NACHRICHT BEI HARDWAREFEHLER:	11-28

EMERGENCY PROTOKOLL	11-28
12 HEARTBEAT PROTOKOLL	12-28
13 CANOPEN OBJEKT VERZEICHNIS	13-29
GLIEDERUNG DES GESAMTEN OBJEKTVERZEICHNISSES:	13-29
14 CANOPEN KOMMUNIKATIONSPROFIL DS 301 V4.02	14-30
KOMMUNIKATIONSOBJEKTE	14-30
HERSTELLERSPEZIFISCHE OBJEKTE	14-30
15 CANOPEN ENCODER GERÄTEPROFIL DS 406 V3.1	15-31
GERÄTESPEZIFISCHE OBJEKTE	15-31
16 OBJEKTE IM DETAILS - ENCODER PROFIL DS 306 V3.1.16-32	
OBJEKT 6000H OPERATING PARAMETERS	16-32
OBJEKT 6001H: MESSSCHRITTE PRO UMDREHUNG (AUFLÖSUNG)	16-32
NACH EINER ÄNDERUNG DER SKALIERUNG MUSS UNBEDINGT DER PRESETWERT EBENFALLS	16-32
AUF NULL/ODER WERT GESETZT WERDEN. OBJEKT 6002H: GESAMTANZAHL DER MESSSCHRITTE	16-32
OBJEKT 6002H: GESAMTANZAHL DER MESSSCHRITTE	16-33
NACH EINER ÄNDERUNG DER GESAMTANZAHL MUSS UNBEDINGT DER PRESETWERT EBENFALLS	16-33
OBJEKT 6003H: PRESET-WERT	16-33
OBJEKT 6004H: POSITIONSWERT	16-33
OBJEKT KANN GEMAPPT WERDEN OBJEKT 6030H: SPEED VALUE	16-33
OBJEKT 6030H: SPEED VALUE	16-34
OBJEKT KANN GEMAPPT WERDEN	16-34
OBJEKT 6200H: CYCLE-TIMER	16-34
OBJEKT 6500H: OPERATING STATUS ANZEIGEN	16-34
OBJEKT 6503H: ALARMS	16-35
OBJEKT 6504H: SUPPORTED ALARMS	16-35
OBJEKT 6505H: WARNINGS	16-36
OBJEKT 6506H: SUPPORTED WARNINGS	16-36
OBJEKT 6400H: WORKING AREA STATE REGISTER	16-37
OBJEKT 6401H: WORKING AREA LOW LIMIT	16-37
OBJEKT 6402H: WORKING AREA HIGH LIMIT	16-37
OBJEKT 2100H: BAUDRATE	16-37
OBJEKT 2101H: KNOTENADRESSE	16-38
OBJEKT 2102H: CAN-BUSTERMINIERUNG AUS/EIN	16-38
OBJEKT 2103H: FIRMWARE FLASHVERSION	16-38
OBJEKT 2105H: SAVE ALL BUS PARAMETERS	16-39
OBJEKT 2140H: CUSTOMER MEMORY (16 BYTES)	16-39
OBJEKT 1029H ERROR BEHAVIOR	16-39
17 NETZWERKMANAGEMENT	17-40
18 NMT-KOMMANDOS	18-41
19 LED-ANZEIGEN WÄHREND DES BETRIEBES	19-42
GRÜNE LED = BUS STATUS	19-42
ROTE LED = ERR ANZEIGE	19-42
20 DEFINITIONEN	20-43
21 VERWENDETE ABKÜRZUNGEN	21-43
22 DEZIMAL-HEXADEZIMAL UMRECHNUNGSTABELLE	22-44
23 GLOSSAR	23-45

1 Allgemeines - Eigenschaften

CANopen Singleturn Drehgeber Serie M36X8 M=Magnetisches Abtastprinzip

Die CANopen-Drehgeber der Baureihe M3658/78 unterstützen das neueste CANopen Kommunikationsprofil nach **DS 301 V4.02**. Zusätzlich stehen gerätespezifische Profile wie das Encoderprofil **DS 406 V3.1** zur Verfügung.

Als Betriebsarten können Polled Mode, Cyclic Mode, Sync Mode gewählt werden. Weiterhin lassen sich Skalierungen, Presetwerte, Endschalterwerte und viele weitere, zusätzliche Parameter über den CAN-Bus programmieren. Beim Einschalten werden sämtliche Parameter aus einem Flash-Speicher geladen, die zuvor nullspannungssicher abgespeichert wurden. Als Ausgabewerte können **Position**, **Geschwindigkeit** sowie der Status der zwei **Endschalter** sehr variabel als **PDO** kombiniert werden (PDO Mapping).

Die Geräte sind mit **einem M12-Stecker** oder mit **Kabelabgang** verfügbar, die Geräteadresse und Baudrate werden softwaregesteuert verändert.

Eine 2-farbige LED auf der Rückseite signalisiert Betriebs- und Fehlerstatus des CAN-Busses sowie den Zustand einer internen Diagnose.

CANopen-Drehgeber sind als Sackloch- und Vollwellenversionen lieferbar und dank Schutzart IP 65 auch für raue Industrieumgebungen geeignet.

Das CANopen Communication Profil DS 301 V4.02

CANopen stellt eine einheitliche Anwenderschnittstelle dar und ermöglicht dadurch einen vereinfachten Systemaufbau mit unterschiedlichsten Geräten. CANopen ist optimiert für den schnellen Datenaustausch in Echtzeitsystemen und verfügt über verschiedene Geräteprofile, die standardisiert wurden. Der CAN in Automation (CiA) Hersteller- und Anwenderverein ist zuständig für die Erstellung und Normung der entsprechenden Profile.

CANopen bietet

- komfortablen Zugriff auf alle Geräteparameter.
- Auto-Konfiguration des Netzwerkes und der Geräte
- Gerätesynchronisation innerhalb des Netzwerkes
- zyklischen und ereignisgesteuerten Prozessdatenverkehr
- gleichzeitiges Einlesen oder Ausgeben von Daten

CANopen nutzt vier Kommunikationsobjekte (COB) mit unterschiedlichen Eigenschaften

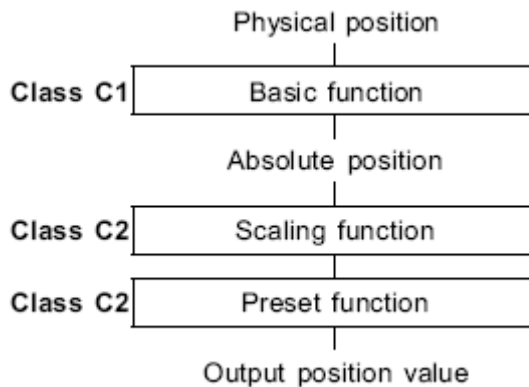
- Prozess-Daten-Objekte (PDO) für Echtzeitdaten,
- Service-Daten-Objekte (SDO) für Parameter- und Programmübertragung,
- Netzwerk Management (NMT, Life-Guarding, Heartbeat)
- Vordefinierte Objekte (für Synchronisation, Zeitstempel, Emergency)

Alle Geräteparameter sind in einem **Objektverzeichnis** abgelegt. Dieses Objektverzeichnis enthält die Beschreibung, Datentyp und Struktur der Parameter sowie die Adresse (Index).

Das Verzeichnis ist gegliedert in einen Kommunikations-Profil-Teil, einen Geräte-Profil bezogenen Teil sowie einen herstellerspezifischen Teil.

Das Drehgeber Geräteprofil DS 406 V3.1

Dieses Profil beschreibt eine **herstellerunabhängige** und verbindliche Festlegung der Schnittstelle für Drehgeber. Im Profil ist definiert, welche CANopen Funktionen verwendet werden und ebenso wie sie zu verwenden sind. Dieser Standard ermöglicht ein offenes und herstellerunabhängiges Bussystem. Das Geräteprofil ist gegliedert in zwei Objekt-Klassen:



- die **Klasse C1** beschreibt alle Grundfunktionen, die der Geber enthalten muss

- die **Klasse C2** enthält eine Vielzahl von erweiterten Funktionen, die von Gebern dieser Klasse entweder unterstützt werden müssen (Mandatory) oder aber optional sind. Geräte der Klasse C2 enthalten somit alle C1- und C2-mandatory-Funktionen, sowie herstellerabhängig weitere optionale Funktionen. Weiterhin ist zusätzlich im Profil ein Adressbereich definiert, der mit herstellereigenen Sonderfunktionen belegt werden kann.

LSS Dienste DS 305 V1.0

CiA DSP 305 CANopen Layer Setting Service und Protokoll (LSS) sind entstanden, um folgende Parameter über das Netzwerk zu lesen und zu ändern:

- Knotenadresse
- Baudrate
- LSS-Adresse

Diese Fähigkeiten erhöhen die „Plug-and-Play“ Kompatibilität des Gerätes und die Konfigurationsmöglichkeit wurde wesentlich vereinfacht. Der LSS-Master ist verantwortlich für die Konfiguration dieser Parameter von einem oder mehreren Slaves im Netzwerk.

Datenübertragung

Daten werden bei CANopen über zwei verschiedene Kommunikationsarten (COB=Communication Object) mit unterschiedlichen Eigenschaften übertragen:

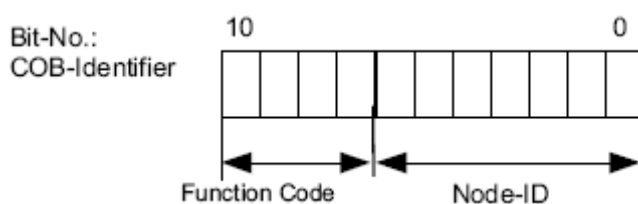
- **Prozess-Daten-Objekte (PDO - echtzeitfähig)**
- **Service-Daten-Objekte (SDO)**

Die Prozess-Daten-Objekte (**PDO**) dienen dem hochdynamischen Austausch von Echtzeitdaten (z.B. Geberposition, Geschwindigkeit, Status der Vergleichspositionen) mit maximal 8 Byte Länge. Diese Daten werden mit hoher Priorität (niedriger COB Identifier) übertragen. PDO's sind Broadcast-Nachrichten und stellen ihre Echtzeitdaten allen gewünschten Empfängern gleichzeitig zur Verfügung. PDO's können gemappt werden, d.h. in einem 8 Byte Datenwort können 4 Byte Position und 2 Byte Geschwindigkeit zusammengefasst werden.

Die Service-Daten-Objekte (**SDO**) bilden den Kommunikationskanal für die Übertragung von Geräteparametern (z.B. Programmierung der Geberauflösung). Da diese Parameter azyklisch (z.B. nur einmal beim Hochfahren des Netzes) übertragen werden, haben die SDO-Objekte eine niedrige Priorität (hoher COB-Identifizier).

Objekte und Funktionscode im Predefined Connection Set

Zur einfacheren Verwaltung der Identifier verwendet CANopen das "Predefined Master/Slave Connection Set". Dabei sind alle Identifier mit Standard-Werten im Objektverzeichnis definiert. Diese Identifier können jedoch über SDO-Zugriff kundenspezifisch geändert werden.



Der 11-Bit Identifier setzt sich aus einem **4 Bit Funktionscode** und einer **7 Bit Knotennummer** zusammen.



Je höher der Wert des COB-Identifiers ,desto niedriger ist dessen Priorität!

Broadcast (netzwerkweite) Objekte

object	function code (binary)	resulting COB-ID	Communication Parameters at Index
NMT	0000	0	-
SYNC	0001	128 (80h)	1005h, 1006h, 1007h
TIME STAMP	0010	256 (100h)	1012h, 1013h

Peer-To Peer (Gerät-zu Gerät)Objekte

object	function code (binary)	Resulting COB-IDs	Communication Parameters at Index
EMERGENCY	0001	129 (81h) – 255 (FFh)	1014h, 1015h
PDO1 (tx)	0011	385 (181h) – 511 (1FFh)	1800h
PDO1 (rx)	0100	513 (201h) – 639 (27Fh)	1400h
PDO2 (tx)	0101	641 (281h) – 767 (2FFh)	1801h
PDO2 (rx)	0110	769 (301h) – 895 (37Fh)	1401h
PDO3 (tx)	0111	897 (381h) – 1023 (3FFh)	1802h
PDO3 (rx)	1000	1025 (401h) – 1151 (47Fh)	1402h
PDO4 (tx)	1001	1153 (481h) – 1279 (4FFh)	1803h
PDO4 (rx)	1010	1281 (501h) – 1407 (57Fh)	1403h
SDO (tx)	1011	1409 (581h) – 1535 (5FFh)	1200h
SDO (rx)	1100	1537 (601h) – 1663 (67Fh)	1200h
NMT Error Control	1110	1793 (701h) – 1919 (77Fh)	1016h, 1017h

Eingeschränkte, reservierte Objekte

COB-ID	used by object
0 (000h)	NMT
1 (001h)	reserved
257 (101h) – 384 (180h)	reserved
1409 (581h) – 1535 (5FFh)	default SDO (tx)
1537 (601h) – 1663 (67Fh)	default SDO (rx)
1760 (6E0h)	reserved
1793 (701h) – 1919 (77Fh)	NMT Error Control
2020 (780h) – 2047 (7FFh)	reserved

Übertragung der Prozess Daten

Bei dem CANopen Drehgeber stehen die **drei PDO-Dienste** PDO1 (tx) ,PDO2 (tx) und PDO3(tx) zur Verfügung. Eine PDO-Übertragung kann durch verschiedene Ereignisse initiiert werden (siehe Objektverzeichnis Index 1800h):

- **asynchron** (Ereignisgesteuert) durch einen internen zyklischen Gerätetimer oder durch eine Prozesswertänderung der Sensordaten
- **synchron** als Antwort auf ein SYNC-Telegramm; (per SYNC-Befehl werden alle CANopen-Knoten zum synchronen Abspeichern ihrer Werte veranlasst, um sie dann nacheinander gemäß der eingestellten Priorität auf den Bus zu legen)
- **als Antwort** auf ein RTR-Telegramm (per Remote Frame=rezessives RTR-Bit wird genau die Nachricht mit dem übermittelten Identifier angefordert)

Die **PDO-Nachrichten** haben nach der **Standardeinstellung** folgenden Aufbau:

Defaulteinstellung für das **Mapping der Transmit PDO1-3:**

Mapping	TPDO1 1800h	TPDO2 1801h	TPDO3 1802h
Mappingobjekt	1A00h	1A01h	1A02h
Eintrag	0x60040020	0x60300110	0x60040020
Objekt	6004h	6030h	6004h
Subindex	00	01	00
Datenlänge	20h(32 Bit)	10h(16 Bit)	20h(32 Bit)
	Asynchron	Asynchron	Synchron

Transmit PDO 1 (1800h) Position asynchron

Default COB-ID ist 180 + Knotennummer: Beispiel 180h + 3Fh = 1BFh

Nachricht	Byte 0	Byte 1	Byte2	Byte 3
1BF	Position LSB	Position MSB	00	00

Die Positionswerte können einen maximalen Wert von 0 – 3FFFh oder 0 – 16383 annehmen.

Transmit PDO2 (1801h) Geschwindigkeit asynchron

Default COB-ID ist 280 + Knotennummer: Beispiel 280h + 3Fh = 2BFh

Nachricht	Byte 0	Byte 1
2BF	Geschwindigkeit LSB	Geschwindigkeit MSB

Der Wert für die Geschwindigkeit ist vorzeichenbehaftet und bewegt sich im Bereich 0 – 1A00h oder 0 – E600h annehmen.

Transmit PDO3 (1802h) Position synchron

Default COB-ID ist 380 + Knotennummer: Beispiel 380h + 3Fh = 3BFh

Nachricht	Byte 0	Byte 1	Byte2	Byte 3
3BF	Position LSB	Position MSB	00	00

Die Positionswerte können einen maximalen Wert von 0 – 3FFFh oder 0 – 16383 annehmen.

Übertragung der Service Daten

SDO-COB-ID

Folgende Identifier stehen standardmäßig für die SDO- Service Dienste zur Verfügung:

SDO (tx) (Geber→Master): 580h (1408) + Knotennummer

SDO (rx) (Master→Geber): 600h (1536) + Knotennummer

Die SDO-Identifier können nicht verändert werden!

Das Kommando-Byte beschreibt die Art der SDO-Nachricht:

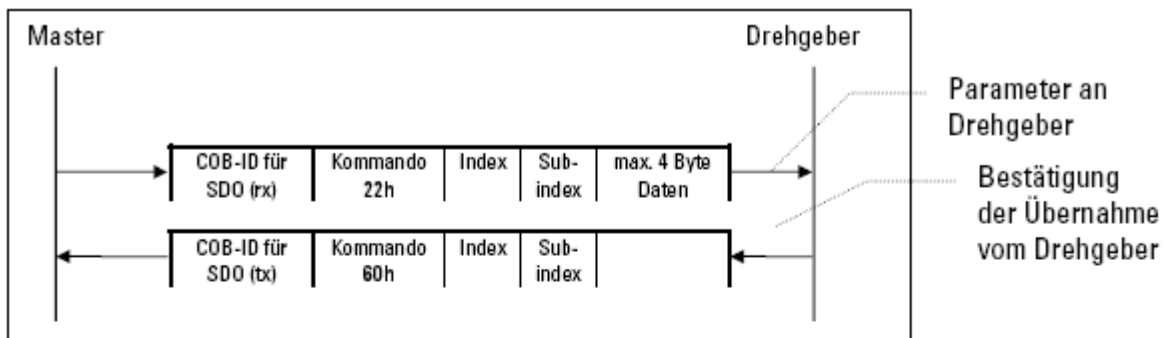
Kommando (Expedited Protocol)	Art	Funktion
22h	SDO(rx), Initiate Download Request	Parameter an Drehgeber senden (Datenlänge max. 4 Byte)
23h	SDO(rx), Initiate Download Request	Parameter an Drehgeber senden (Datenlänge = 4 Byte)
2Bh	SDO(rx), Initiate Download Request	Parameter an Drehgeber senden (Datenlänge = 2 Byte)
2Fh	SDO(rx), Initiate Download Request	Parameter an Drehgeber senden (Datenlänge = 1 Byte)
60h	SDO(tx), Initiate Download Response	Bestätigung der Übernahme an Master
40h	SDO(rx), Initiate Upload Request	Parameter vom Drehgeber anfordern
43h	SDO(tx), Initiate Upload Response	Parameter an Master mit Datenlänge=4 Byte (Unsigned 32)
4Bh	SDO(tx), Initiate Upload Response	Parameter an Master mit Datenlänge=2 Byte (Unsigned 16)
4Fh	SDO(tx), Initiate Upload Response	Parameter an Master mit Datenlänge=1 Byte (Unsigned 8)
80h	SDO(tx), Abort Domain Transfer	Drehgeber meldet Fehlercode an Master



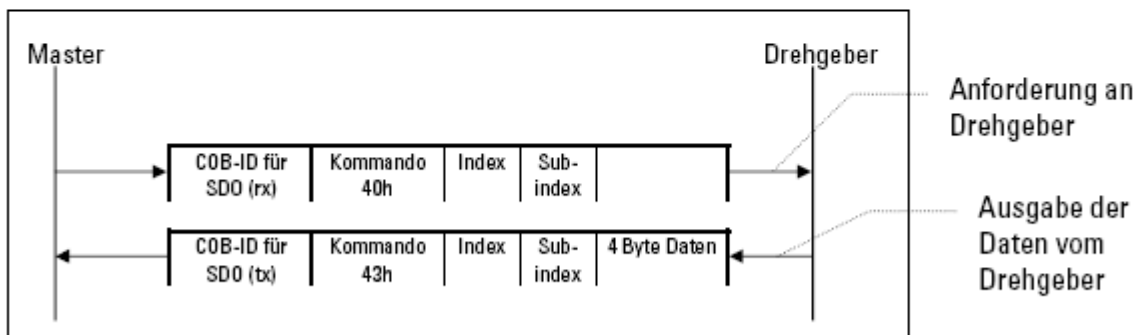
Eine Fehlermeldung (Kommando 80h) ersetzt im Fehlerfall die normale Bestätigung (Response). Die Fehlermeldung umfasst sowohl Kommunikations-Protokoll-Fehler (z.B. falsches Kommando-Byte) als auch Objektverzeichnis-Zugriffsfehler (z.B. falscher Index, Schreibversuch auf Read-Only-Objekt, falsche Datenlänge etc).

Die Fehlercodes sind im CANopen-Profil (DS 301) bzw. im Geräteprofil (DSP 406) beschrieben.

Beispiel einer Übertragung von Service-Daten zu und vom Drehgeber



Master überträgt Parameter an Drehgeber



Master fordert Parameter vom Drehgeber an

LSS Dienste

LSS Hardware Anforderungen (LSS Address)

Alle LSS-Slaves müssen einen gültigen Objekteintrag im Objektverzeichnis für das Identity-Object [1018h] vorweisen, um eine selektive Konfiguration des Knotens vornehmen zu können. Dieses Objekt besteht aus folgenden Sub-Indices:

- **Vendor-ID (numerical number)**
- **Product-Code (numerical number)**
- **Revision-Number (major and minor revision as numerical number)**
- **Serial-Number (numerical number)**
- **LSS Master CAN-ID 2021**
- **LSS Slave CAN-ID 2020**

Ein Produkt-Code, eine Revision-Number und eine Serial-Number werden vom Hersteller eingestellt. Die LSS - address muss im Netzwerk eindeutig sein.

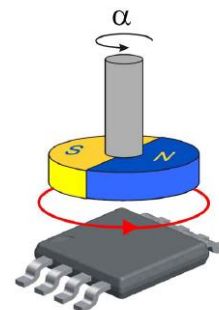
LSS Operative Einschränkungen

Um eine reibungslose LSS Funktionalität zu gewährleisten, müssen alle Geräte im Netz die LSS-Dienste unterstützen. Es kann nur einen LSS- Master geben. Alle Knoten müssen mit derselben Baudrate starten. Eine LSS Kommunikation kann nur im "Stopp-Mode" oder im "Pre-Operational" Mode stattfinden.

2 Versorgungsspannung und CAN-Bus Anschluss

CAN-Kabelbelegung

Sensor:	Magnetischer Hall -Sensor
	14 Bit Auflösung /9 Bit Genauigkeit
Versorgungsspannung:	10 ... 30 VDC
Stromaufnahme:	typ. 22mA bei 24 VDC
	max. 49 mA bei 10 VDC
Verpolschutz der Versorgung:	ja
CAN Transceiver	82C251 / kurzschlussfest
Galvanische Trennung	Nein
CANopen Kommunikation	DS 301 V4.02
CANopen Encoder Geräte Profil	DS406 V 3.1



Kurzzeichen	Bezeichnung	Farbe
CG	CAN Ground	grau
CL	CAN_Low (-)	gelb
CH	CAN_High (+)	grün
0V	0Volt Versorgung	weiß
+V	+UB Versorgung	braun

Anschlußbild Kabelabgang

CANopen M12-Steckerbelegung



Kurzzeichen	Bezeichnung	PIN	Farbe
CG	CAN Ground	1	Gy
CL	CAN_Low (-)	5	Ye
CH	CAN_High (+)	4	Gn
0V	0 Volt Versorgung	3	Wh
+V	+UB Versorgung	2	Bn

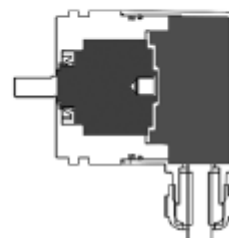
Anschlußbild M12-Stecker

Mechanische Kennwerte

Mechanische Kennwerte:

Max. Drehzahl:	6000 min ⁻¹
Anlaufdrehmoment:	< 0,06 Nm
Wellenbelastbarkeit radial:	40 N
Wellenbelastbarkeit axial:	20 N
Gewicht:	ca. 0,2 kg
Schutzart EN 60 529:	IP 67 (IP 69k auf Anfrage)
Arbeitstemperaturbereich:	-40 °C ... +85 °C
Werkstoffe:	Welle: nicht rostender Stahl, Flansch: Aluminium, Gehäuse: Zink-Druckgussgehäuse, Kabel: PUR
Schockfestigkeit nach DIN-IEC 68-2-27:	5000 m/s ² , 6 ms
Vibrationsfestigkeit nach DIN-IEC 68-2-6:	300 m/s ² , 10 ... 2000 Hz
Dauerschocken nach DIN-IEC 68-2-29	1000 m/s ² , 2 ms
Vibration (Breitbandrauschen) nach DIN-IEC 68-2-64	5 ... 2500 Hz, 100 m/s ² - rms

Rundum geschützt dank *Safety-Lockplus™* und *Sensor-Protect™* Technologie



Safety-Lockplus™:
Flanschseitig IP69k, robuste Lagerbaugruppen mit verblockten Lagern, mechanisch geschützte Wellendichtung

Sensor-Protect™:
Vollvergossene Elektronik, getrennte mechanische Baugruppe

3 Erst-Inbetriebnahme - Generelle Einstellungen am Gerät


Baudrate

Die Baudrate kann **mit einer CANopen-Software auf Objekt 2100h** oder über den entsprechenden **LSS-Dienst** geändert werden.

Defaulteinstellung: 250 kBit/s (Eintrag 5)

Wert	Baudrate in KBit/s
0	10
1	20
2	50
3	100
4	125
5	250
6	500
8	1000

Bitte beachten bei entsprechender Baudrate

 Die gewählte Zykluszeit (siehe Objekt 1800h, Subindex 5 Event Timer) muss größer als die Busübertragungsdauer sein, damit die PDO's fehlerfrei abgesetzt werden können!

Bei Baudrate 10 KBaud: Zykluszeit mindestens 14 ms
Bei Baudrate 20 KBaud: Zykluszeit mindestens 10 ms
Bei Baudrate 50 KBaud: Zykluszeit mindestens 4 ms

Bei Zykluszeit=0 im Event-Mode (d. h. PDO bei Wertänderung) muss die Baudrate mindestens 125 KBaud betragen.

Knotennummer

Die Knotennummer kann ebenfalls **per Software auf Objekt 2101h** oder den entsprechenden **LSS-Dienst** geändert werden..

Defaulteinstellung: 0x3F (63 dezimal).

Die **Knotennummer 0** ist reserviert und darf von keinem Knoten verwendet werden.

Die resultierenden Knotennummern liegen im Bereich **1...7Fh** hexadezimal (1...127 dezimal).



Die Übernahme einer neuen Knotennummer erfolgt erst beim nächsten Hochlaufen (Reset/Power-on) des Gebers oder über einen **NMT-Reset Node** Befehl.

Alle Einstellungen innerhalb der Objekttafel werden auf den Standard/Default wert gesetzt.

Terminierung

Die **CAN-Bus Terminierung** kann ebenfalls **per Software auf Objekt 2102h** geändert werden..

Defaulteinstellung: 0x01 (Busterminierung aktiv)

Der durchgeschleifte CAN-Bus muss an beiden Enden mit einem Busabschlusswiderstand von 120 Ohm zwischen CAN+ und CAN- abgeschlossen werden.



Beachten Sie, dass nur das **letzte Gerät** in der CAN-Buslinie terminiert wird.

Save All Bus Parameters

Dieser Parameter (**Objekt 2105h**) speichert die gewünschten Busparameter (Objekt 2100h ,2101h,2102h) permanent im Flash-Speicher. Dieses Objekt dient als zusätzliche Absicherung vor ungewolltem Ändern der Baudrate und Knotenadresse.



Erst durch gezieltes Abspeichern mit dem Parameter „**save**“ (**hexadezimal 0x65766173**) werden die Busparameter **Baudrate, Knotenadresse und Terminierung** permanent abgespeichert.

4 Layer Setting Services (LSS)

Exakt zwei Bedingungen müssen bei Geräten, die an ein CANopen Netzwerk angeschlossen werden, erfüllt sein - alle Geräte müssen dieselbe **Baudrate** haben und die **Knotenadresse** muss einzig innerhalb des Netzwerkes sein. Die Bedingungen für einen Einsatz unter LSS sind, dass zum Gerät eine 1:1 CAN-Verbindung besteht. Über einen speziellen Dialogmodus können danach die **Baudrate** und die **Knotenadresse** verändert werden. Die **COB-ID 0x7E5** wird vom Master zum Slave verwendet, der Slave antwortet mit der **COB-ID 0x7E4**. LSS-Nachrichten sind immer 8 Bytes lang. Nicht verwendete Bytes sind reserviert und sollten mit 0 aufgefüllt werden.

Um ein Gerät in den LSS-Konfigurationsmode zu schalten, wird ein "**Switch Mode Global**" Kommando gesendet:

0x04	0x01	reserved
------	------	----------

Unglücklicherweise wird dieses Kommando nicht bestätigt und es kann nur über eine visuelle Kontrolle der LED's festgestellt werden, ob das Gerät in diesen Modus umgeschaltet hat.

Als nächster Befehl wird der „**Inquire Node-ID**“ Service aufgerufen:

0x5E	reserved
------	----------

Wenn es erfolgreich war, antwortet der Slave mit:

0x5E	Node ID	reserved
------	---------	----------

Wenn keine Rückantwort vom Gerät eingelesen wurde, so wird vermutlich der **LSS-Service** nicht unterstützt oder die Baudrate ist nicht korrekt.

Das Kommando "**Configure Node-ID**" wird verwendet, um die **Knotenadresse** neu zu konfigurieren:

0x11	Node ID	reserved
------	---------	----------

Der Fehlercode ist in der Antwort des Slave-Gerätes enthalten:

0x11	Error code	Error extension	reserved
------	------------	-----------------	----------

Error code 0 bedeutet eine Akzeptanz des Befehles - Error code 1 bedeutet unzulässige Node-ID. Die restlichen Error codes sind reserviert. Die Error extension beinhaltet herstellerspezifische Informationen und kann nur bei dem Error code 0xFF angewendet werden.

Die Baudrate wird über das Kommando "**Configure Bit Timing Parameters**" aktiviert:

0x13	Bit timing	Table entry	reserved
------	------------	-------------	----------

Standardisierte Baudraten nach CiA sind folgende:

Baudrate table 0x00	
Table index	Baudrate
0	1000 kBit/s
1	800 kBit/s
2	500 kBit/s
3	250 kBit/s
4	125 kBit/s
5	reserved
6	50 kBit/s
7	20 kBit/s
8	10 kBit/s

Wieder antwortet das Gerät mit einem Fehlercode:

0x13	Error code	Error extension	reserved
-------------	-------------------	------------------------	-----------------

Error code 0 bedeutet eine Akzeptanz des Befehles - Error code 1 bedeutet unzulässige Node-ID. Die restlichen Error codes sind reserviert. Die Error extension beinhaltet herstellerspezifische Informationen und kann nur bei dem Error code 0xFF angewendet werden.

Nun sind beide netzwerk-spezifischen Parameter geändert worden und mit dem "**Store Configuration**" sollten die neuen Parameter gespeichert werden:

0x17	reserved
-------------	-----------------

Wieder antwortet das Gerät mit einem Fehlercode:

0x17	Error code	Error extension	reserved
-------------	-------------------	------------------------	-----------------

Error code 0 bedeutet eine Akzeptanz des Befehles - Error code 1 bedeutet unzulässige Node-ID. Die restlichen Error codes sind reserviert. Die Error extension beinhaltet herstellerspezifische Informationen und kann nur bei dem Error code 0xFF angewendet werden.

Zum Abschluss des LSS-Service wird das Gerät vom LSS-Konfigurationsmodus zurückgeschaltet in den **Preoperational Modus** mit dem Kommando "**Switch Mode Global**":

0x04	0x00	reserved
-------------	-------------	-----------------

Das Gerät führt **selbstständig einen neuen Boot-up (Reset node)** aus und alle neuen Einstellungen sind danach gültig.

5 Defaulteinstellungen bei Auslieferung



Folgende Softwareparameter sind bei der Auslieferung ab Werk eingestellt.

Bezeichnung	Einstellung	Software
Baudrate	250 kBit/s	Object 2100h = 05h
Knotenadresse	63	Object 2101h = 3Fh
Terminierung	Ein	Object 2102h = 01h

Index (hex)	Name	Standardwert
	Kommunikationsparameter	
1005h	COB-ID Sync	80h
100Ch	Guard Time	0
100Dh	Life Time Factor	0
1012h	COB-ID Time stamp	100h
1013h	High Resolution time stamp	0
1017h	Producer heartbeat time	0
1029h	Error Behaviour	0 = Comm Error 1 = Device specific 1 = Manufacturer Err.
1800h	TPDO1 Communication Parameter	
01h	COB-ID	180h + Knotennummer
02h	Transmission Type	255 (asynch)
03h	Inhibit Time	0 [step 100 µs]
05h	Event timer	0 [step 1 ms]
1801h	TPDO2 Communication Parameter	
01h	COB-ID	280h + Knotennummer
02h	Transmission Type	255 (asynch)
03h	Inhibit Time	0 [step 100 µs]
05h	Event timer	0 [step 1 ms]
1802h	TPDO3 Communication Parameter	
01h	COB-ID	380h + Knotennummer
02h	Transmission Type	1 (synchron)
03h	Inhibit Time	0 [step 100 µs]
05h	Event timer	0 [step 1 ms]
1A00h	TPDO1 Mapping	
01h	1.Mapped Object	0x60040020
1A01h	TPDO2 Mapping	
01h	1.Mapped Object	0x60300110
1A02h	TPDO3 Mapping	
01h	1.Mapped Object	0x60040020

Index (hex)	Name	Standardwert
	Encoder Profil	
6000h	Operating Parameter	0h Scaling off
6001h	Measuring Units per Revolution	16384 (14 Bit)
6002h	Total Measuring Range	16384 (14 Bit)
6003h	Preset value	0
6200h	Cyclic Timer (see TPDO1 Comm.Par)	0
6401h	Work area low limit	0
6402h	Work area high limit	16384
2100h	Baudrate	05h
2101h	Node number	63
2102h	CANbus Terminierung	1 (aktiv)
2105h	Save All Bus Parameters	0x65766173



Die ursprünglichen Standard-Werte (Default-Werte bei Auslieferung) können durch das Objekt **1011h (Restaurieren der Parameter)** wieder zurück geladen werden. mit der Angabe „load“ (**0x6C6F6164**)

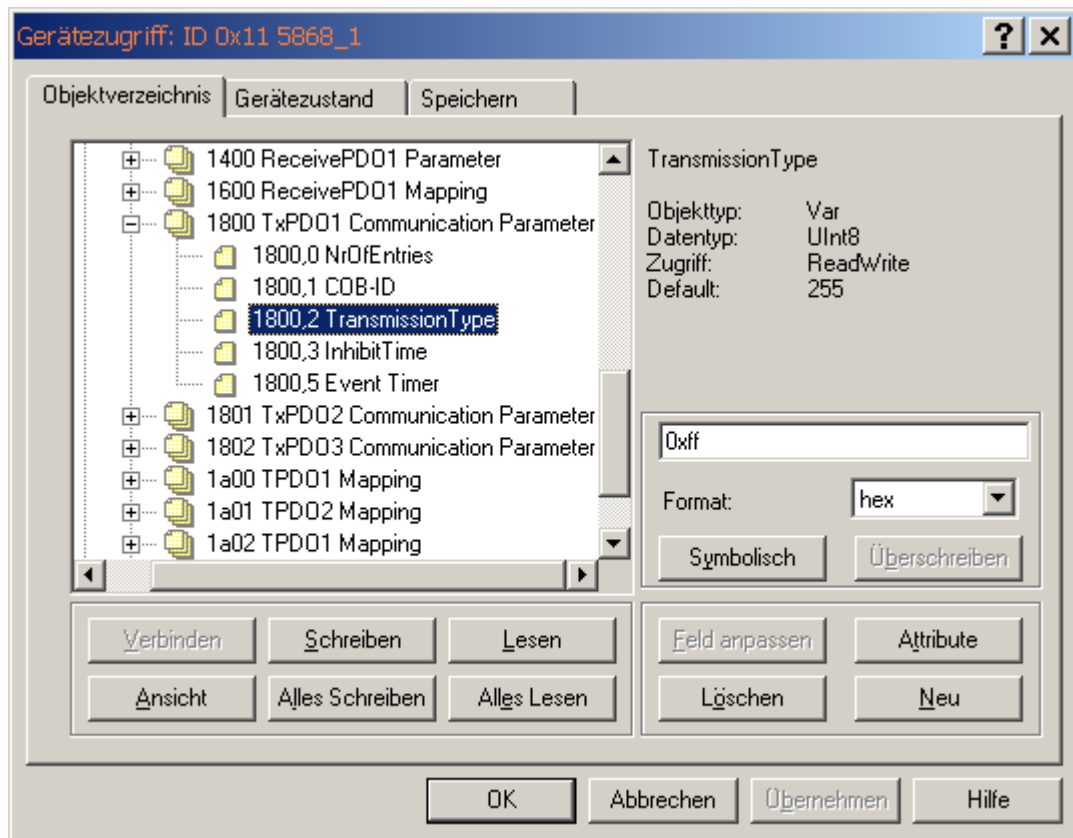
Um die geänderten Parameter auch spannungsausfallsicher abzuspeichern, müssen diese unbedingt über das Objekt **1010h** (Parameter speichern) in das EEPROM übertragen werden. Es werden dabei alle vorher im EEPROM vorhandenen Daten überschrieben!



Sind beim Programmieren der Objekte Fehler aufgetaucht und sind diese Parameter im EEPROM abgespeichert worden, dann lässt sich der Geber nach dem nächsten Einschalten nicht mehr ansprechen (Geber sendet nur noch **Emergency** Nachrichten). Dieser Fehler kann nur durch einen generellen **Reset** des Drehgebers behoben werden.

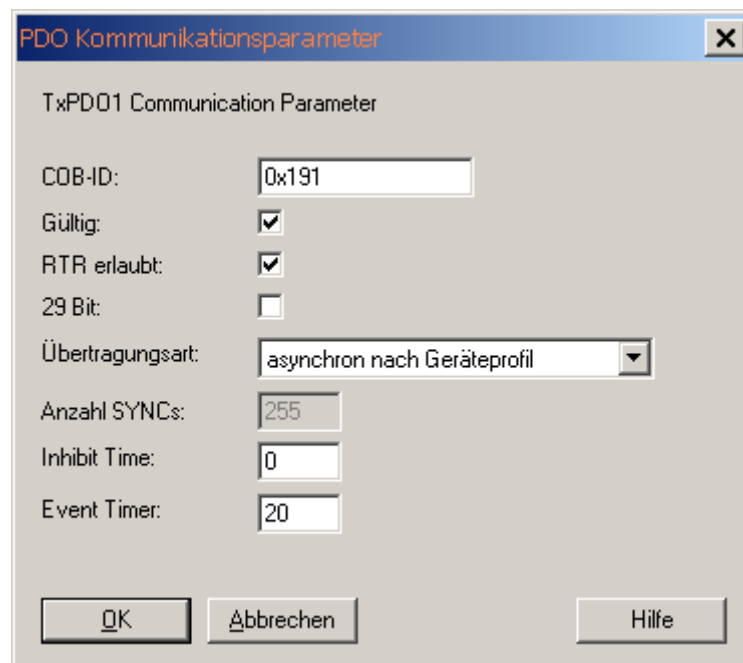
6 Kommunikationsparameter

Der COB-ID und die Übertragungsart für **PDO1** wird im Objektverzeichnis Index **1800h** festgelegt

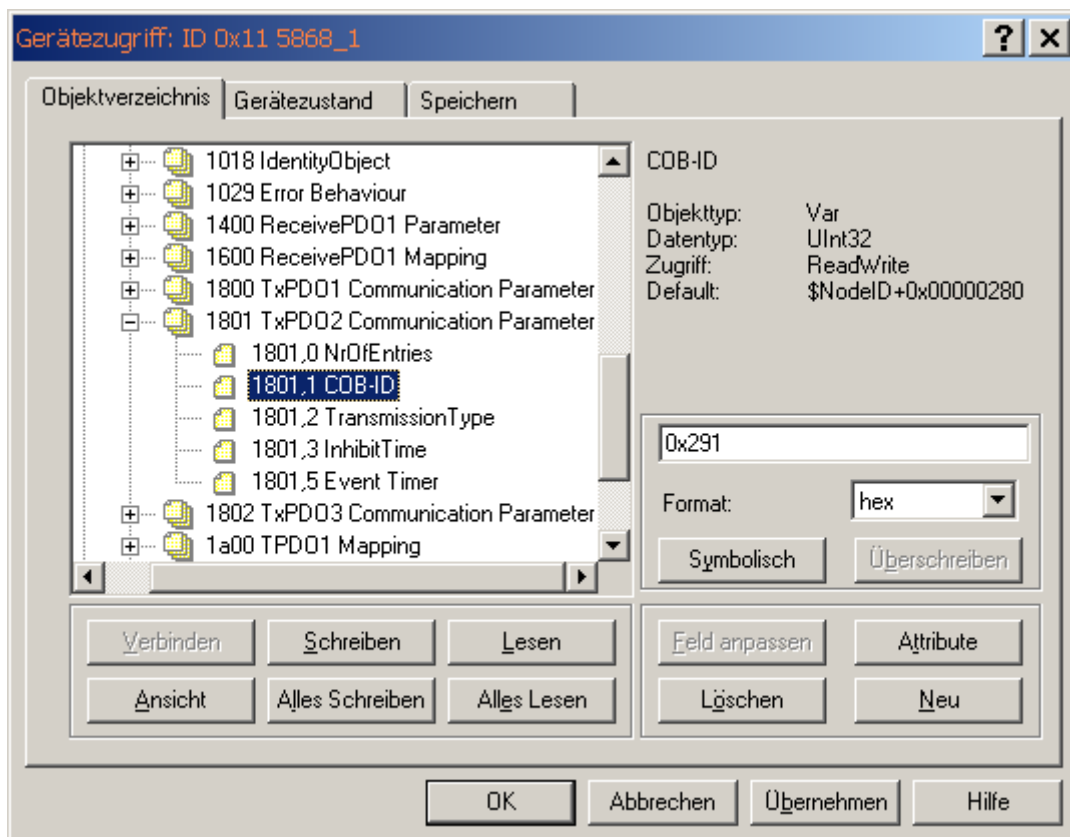


Defaulteinstellungen:

Freigabe:	PDO gültig (enabled)	RTR erlaubt
COB-ID:	180h + eingestellte Knotennummer (hier 11h)	
Übertragungsart:	255 = asynchron nach Geräteprofil	
Event Timer:	20 ms	

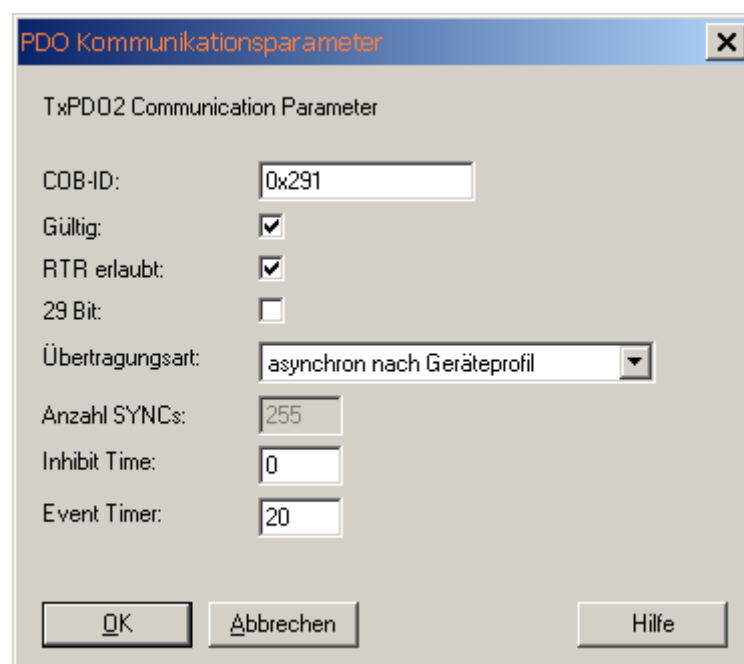


Der COB-ID und die Übertragungsart für **PDO2** wird im Objektverzeichnis Index 1801h festgelegt



Defaulteinstellungen:

Freigabe:	PDO gültig (enabled)	RTR erlaubt
COB-ID:	280h + eingestellte Knotennummer (hier 11h)	
Übertragungsart:	255 = asynchron nach Geräteprofil	
Event Timer:	20 ms	



Definition des Übertragungstyps (Transmission type) des PDO's

transmission type	PDO transmission				
	cyclic	acyclic	synchronous	asynchronous	RTR only
0		X	X		
1-240	X		X		
241-251	- reserved -				
252			X		X
253				X	X
254				X	
255				X	

Ein Wert zwischen 1 ...240 bedeutet ,dass das PDO **synchron und zyklisch** gesendet wird. Die Nummer des Transmission Typ bedeutet die **Anzahl der SYNC** Impulse ,die notwendig sind, um die PDO's zu versenden. Der Transmission Typ 252 und 253 sagt aus, dass das PDO nur auf Anfrage über RTR gesendet wird.



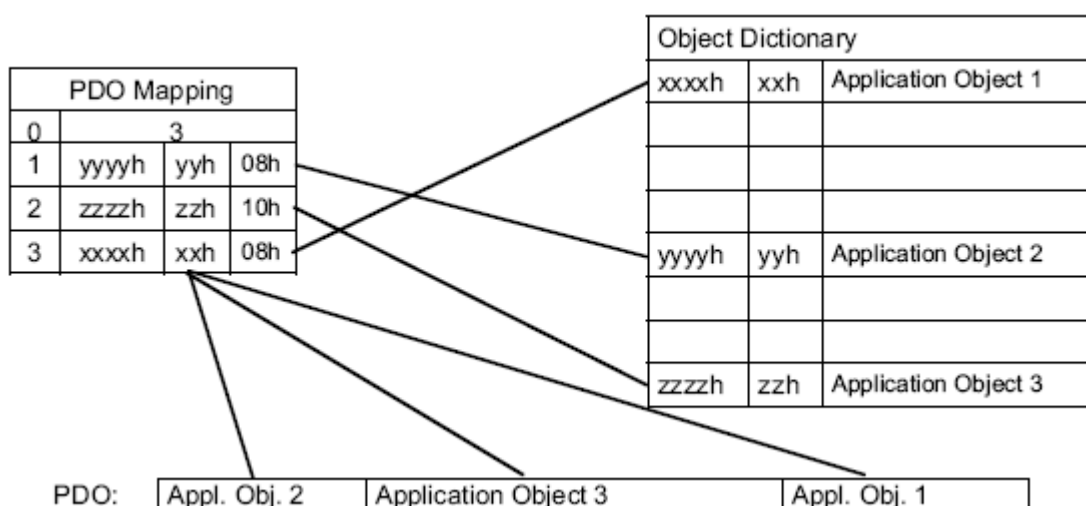
Der Typ 254 bedeutet, dass das Ereignis applikationsabhängig getriggert wird, während die Nummer 255 geräteprofilabhängig ist. Zusätzlich kann für die Nummer 254/255 eine zeitgesteuerter **EventTimer** eingesetzt werden. Der Wertebereich für den Timer erstreckt sich von **1ms ... 65535 ms**.

Variables PDO Mapping

Variables Mapping der verschiedenen Objekte bedeutet, dass der Anwender in der Lage ist, den Inhalt der Transmit PDO's applikationsabhängig zu konfigurieren.

Beispiel eines Eintrags in die Mappingtabelle:

Das gemappte PDO besteht aus 3 Applikationsobjekteinträgen mit unterschiedlicher Länge:

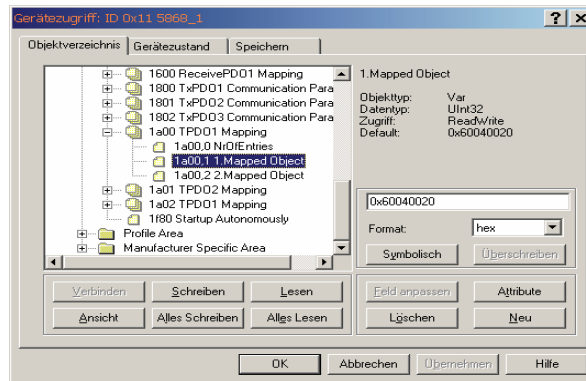


Das Applikationsobjekt 2 belegt im Sende PDO ein Byte (08h). Danach folgt das Applikationsobjekt 3 mit 16 Bit Länge(10h = 2 Bytes) und zum Schluß mit 1 Byte Länge das Applikationsobjekt 1. Insgesamt werden 32 Bit in diesem PDO belegt.

7 Beispiel eines variablen Mappingeintrages

Mapping Objekt 1A00h

Das Mapping Objekt **1A00h** beschreibt das **1.Transmit PDO**. Es können so viele Objekte gemappt werden, bis die maximale Datenlänge von **8 Bytes** erreicht ist. Analog dazu beschreibt Objekt **1A01h** das **Transmit PDO2** und **1A02h** das **Transmit PDO3**.



Mapping	TPDO1 Mapping	TPDO1 Mapping	TPDO1 Mapping
Subindex	00	01	02
Eintrag	Nr.of Entries	1.Mapped Objekt	2.Mapped Objekt
Objekt	2	6004h	6030h
Subindex		00	01
Datenlänge	Byte	20h(32 Bit)	10h(16 Bit)
		Asynchron	Asynchron

Folgende Objekte können gemappt werden : (blau hervorgehoben)



Gerätespezifische Objekte

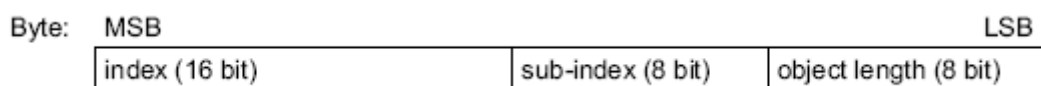
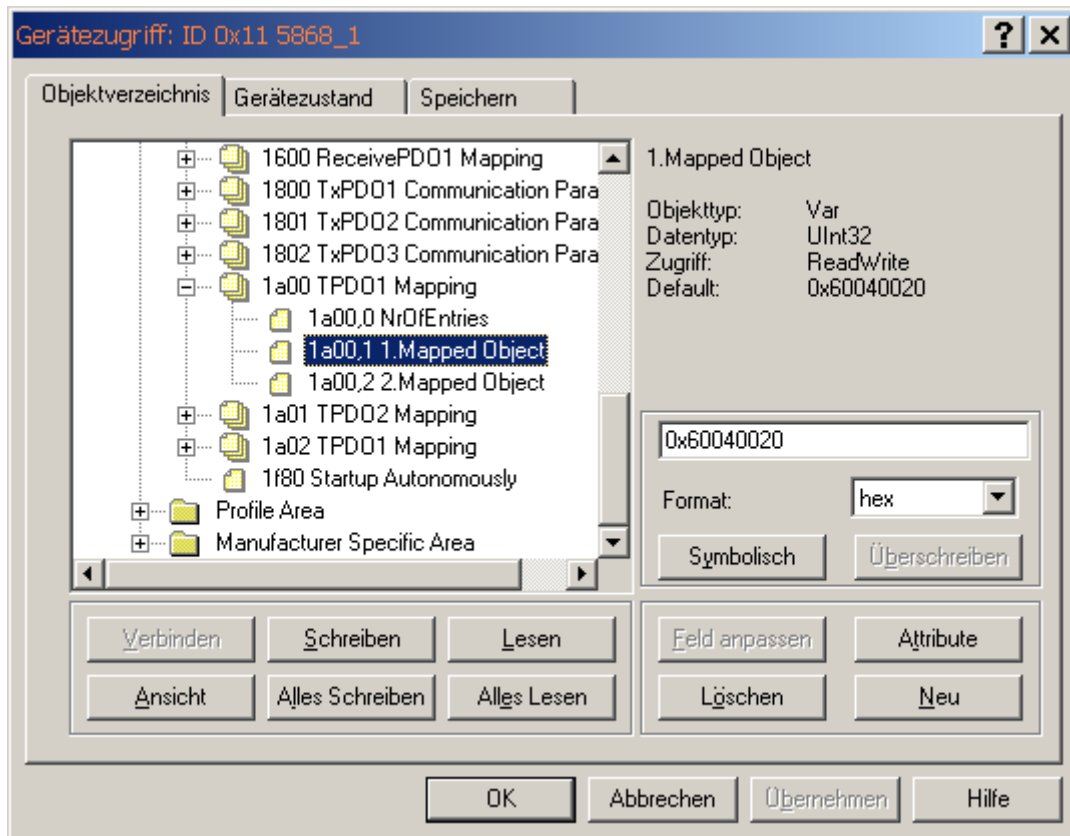
INDEX (hex)	Object Symb.	ATTRIB	Name	M/W C2	TYPE
6000	VAR	RW	Operating parameters	M	unsigned16
6001	VAR	RW	Measuring Units p.Revolution (MUR)	M	unsigned32
6002	VAR	RW	Total Measuring Range (TMR)	M	unsigned32
6003	VAR	RW	Preset value	M	unsigned32
6004	VAR	RO	Position value	M MAP	unsigned32
6030	ARRAY	RO	Speed Value	O MAP	signed16
6040	ARRAY	RO	Acceleration Value	O	Signed16
6200	VAR	RW	Cyclic Timer	M	unsigned16
6400	ARRAY	RO	Working Area state	O MAP	unsigned 8
6401	ARRAY	RW	Working Area Low Limit	O	Unsigned32
6402	ARRAY	RW	Working Area High Limit	O	Unsigned32
6500	VAR	RO	Operating Status	M	unsigned16
6501	VAR	RO	Measuring Step (Singleturn)	M	unsigned32
6502	VAR	RO	Number of revolutions	M	unsigned16
6503	VAR	RO	Alarms	M MAP	unsigned16
6504	VAR	RO	Supported alarms	M	unsigned16
6505	VAR	RO	Warnings	M MAP	unsigned16



In diesem Beispiel sind 2 Prozessdatenobjekte, Objekt 6004h und 6030h gemappt.

8 Struktur eines Mapping Eintrages

Das Mapping Object für **PDO 1** ist im Objektverzeichnis Index 1A00h festgelegt. Es besteht aus 1 Eintrag und kann durch den Anwender verändert werden (variable mapping).



Die Defaulteinstellung für das **Mapping der Transmit PDO:**

Mapping	TPDO1	TPDO2	TPDO3
1.Mapping	0x60040020	0x60300110	0x60040020
Objekt	6004h	6030h	6004h
Subindex	00	01	00
Datenlänge	20h(32 Bit)	10h(16 Bit)	20h(32 Bit)
	Asynchron	Asynchron	Synchron

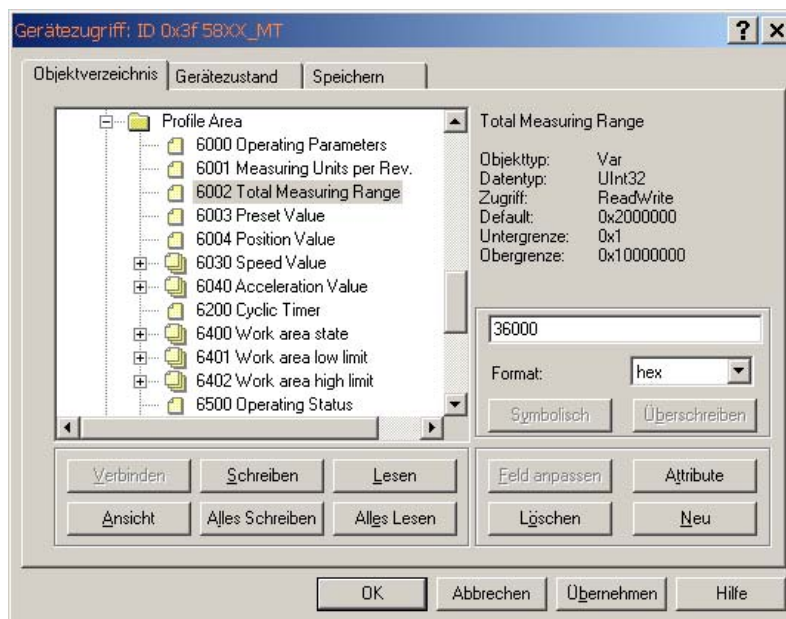
Der CANopen Drehgeber 36XX unterstützt ein **variables Mapping** auf allen 3 Transmit PDO's.

9 Beispiel einer Applikations-Programmierung:

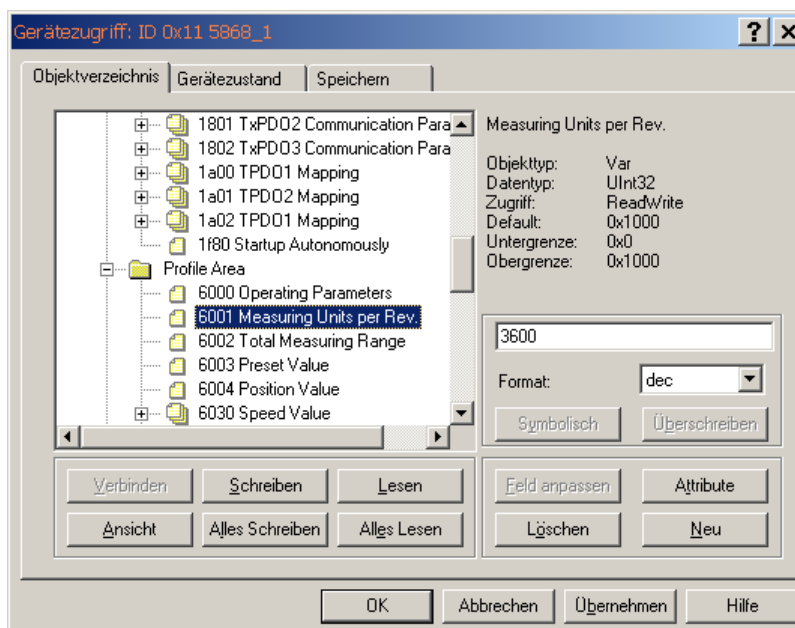
Objekte einrichten

- **Total Measuring Range** auf 3600 begrenzen
- **Measuring Units per Revolution** soll auf 3600 Schritte pro Umdrehung gesetzt werden
- **Positionswert** muss auf 0 gesetzt werden
- TPDO1 (Position) soll mit 10 ms Event senden
- TPDO2 (Speed) soll mit 20 ms Event senden
- **Producer Heartbeat** soll auf 500 ms gesenkt werden
- Work area Limit beträgt 1000 und 3500
- Die neuen Parameter sollen im **EEPROM** gespeichert werden

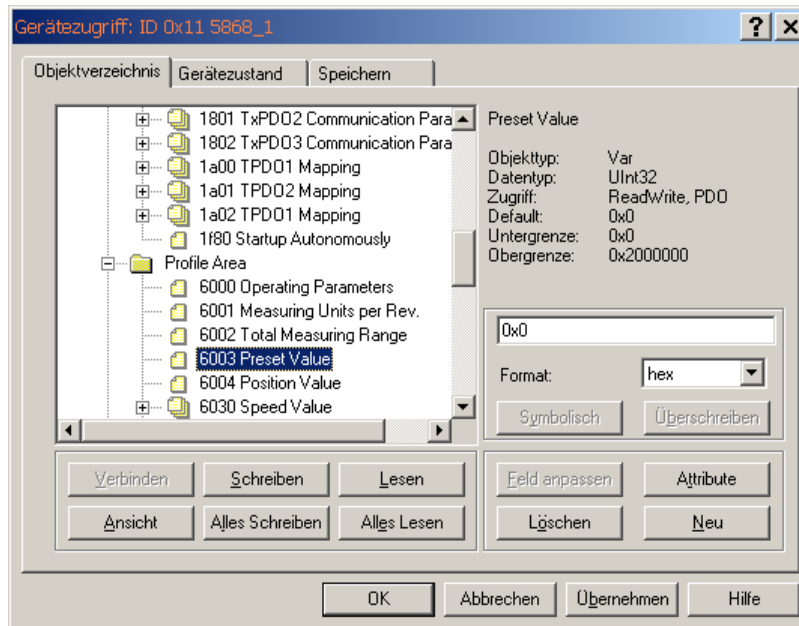
Total Measuring Range auf 3600 begrenzen



Measuring Units per Revolution auf 3600 begrenzen

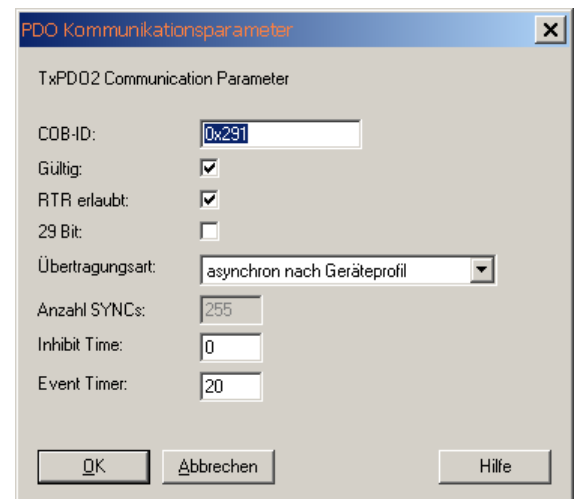
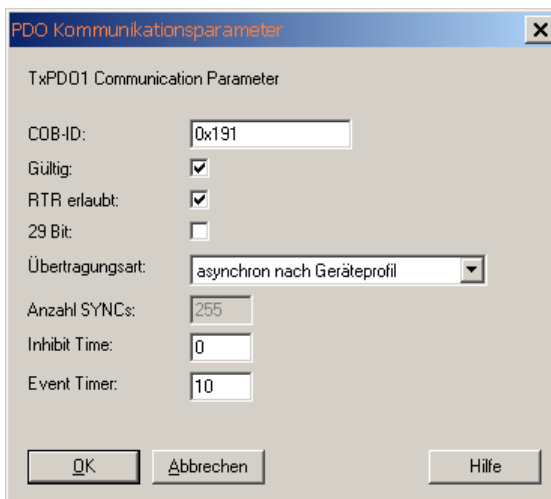


Preset value auf 0

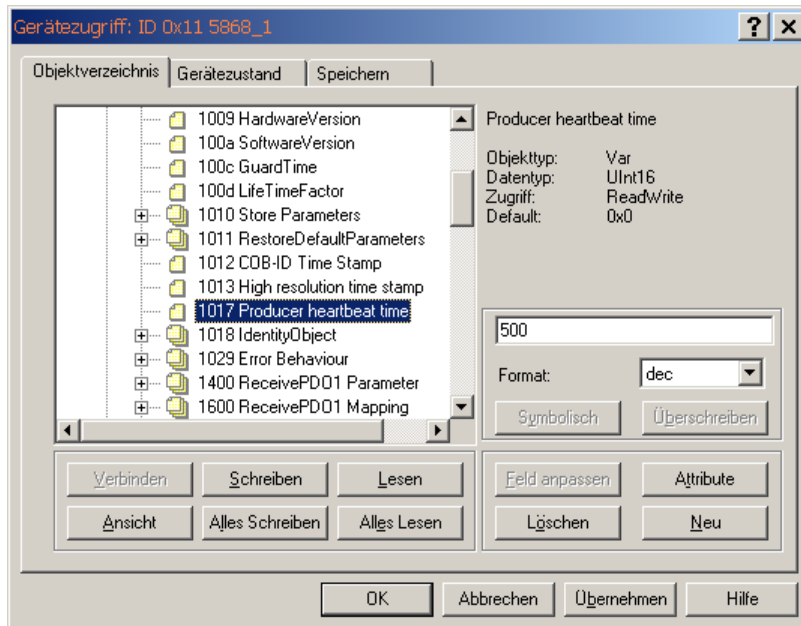


Transmit Parameter TPDO1 und TPDO2 Werte setzen

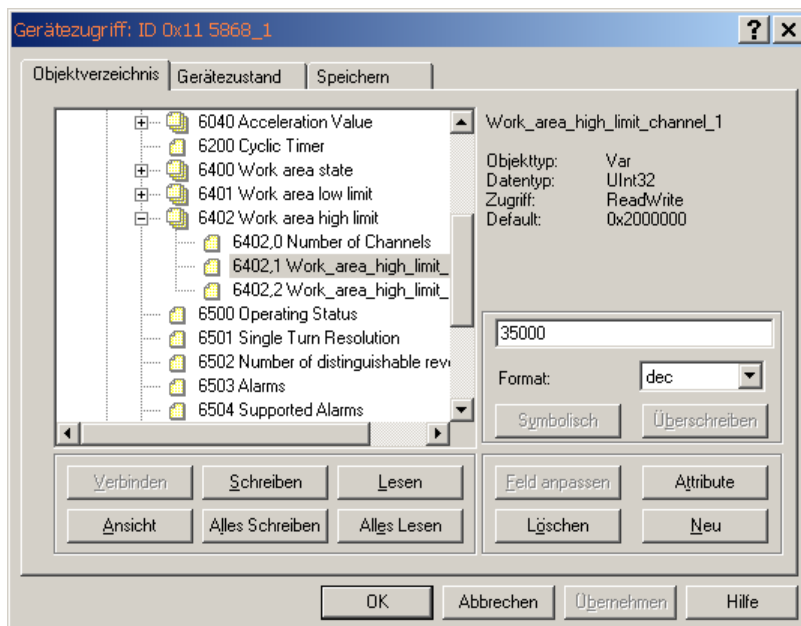
Der Typ 254 bedeutet, dass das Ereignis applikationsabhängig getriggert wird, während die Nummer 255 **Geräteprofil abhängig** ist. Zusätzlich kann für die Nummer 254/255 eine zeitgesteuerter **EventTimer** eingesetzt werden. Der Wertebereich für den Timer erstreckt sich von **1ms ... 65535 ms**.



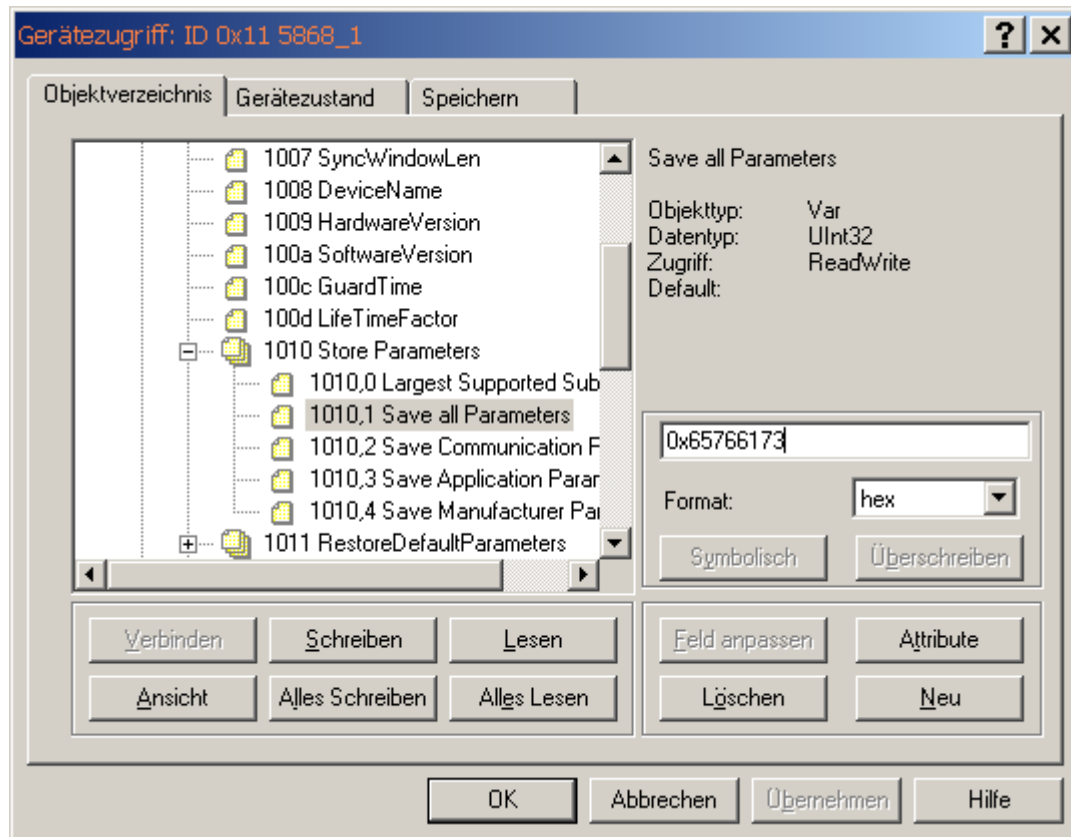
Producer Heartbeat auf 500 ms setzen



Work area Low- und High-Limit Werte setzen



alle geänderten Parameter im EEPROM speichern Store Parameters 1010h



Objekt 1010h Parameter abspeichern

Mit Hilfe des Kommandos "save" unter Sub-Index 1h (save all Parameters) wird das Abspeichern der Parameter in den nichtflüchtigen Speicher (EEPROM) veranlasst.

Unter diesem Unterpunkt werden alle Kommunikationsobjekte, Applikationsobjekte und herstellerspezifische Objekte abgespeichert. Dieser Vorgang benötigt ca. 14 ms.

Um ein versehentliches Abspeichern zu verhindern, wird der Befehl nur ausgeführt, wenn als Codewort der String "save" in diesen Sub-Index eingetragen wird.

Ein Lesezugriff auf den Sub-Index 1h liefert Informationen über die Speicherfunktionalität.

Bezeichnung	Inhalt	Kommentar
Byte 3	73h	(ASCII Code für "s")
Byte 2	61h	(ASCII Code für "a")
Byte 1	76h	(ASCII Code für "v")
Byte 0	65h	(ASCII Code für "e")

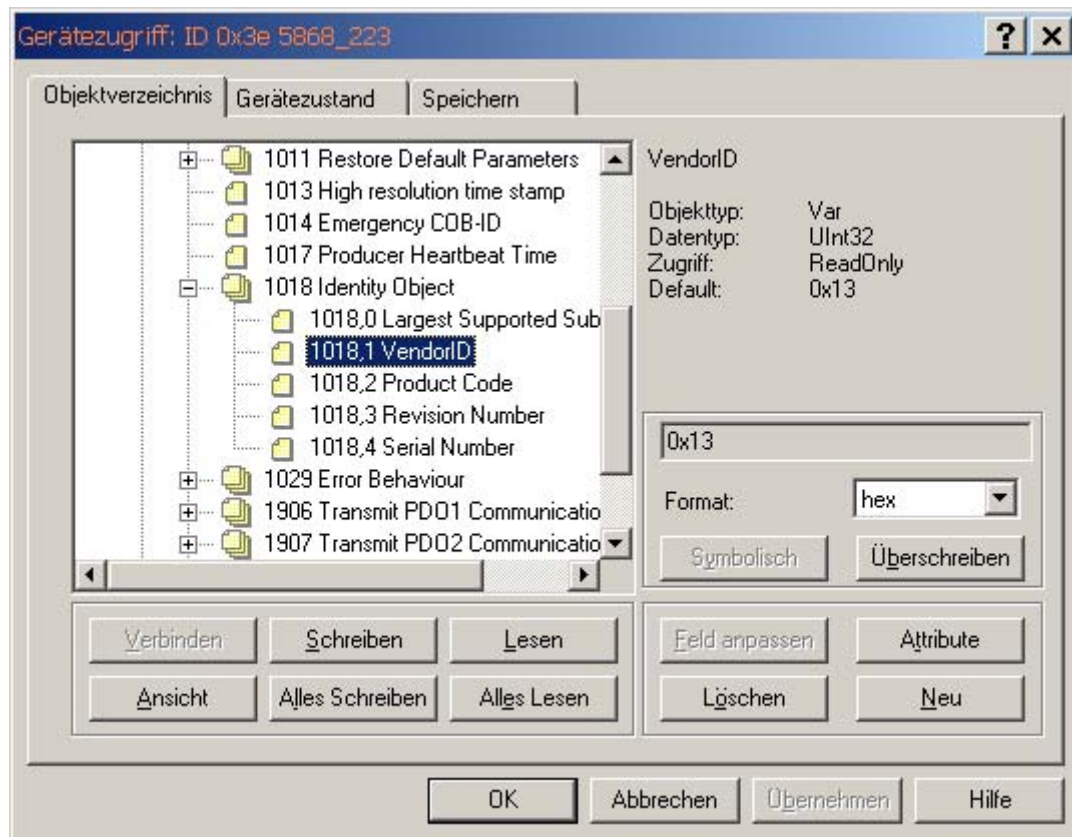
Objekt 1011h: Standard-Werte laden

Mit Hilfe des Kommandos "load" unter Sub-Index 1h werden alle Parameter auf ihre Standard-Werte zurückgesetzt. Um ein versehentliches Laden der Standard-Werte zu verhindern, wird der Befehl nur ausgeführt, wenn als Codewort der String "load" in diesen Sub-Index eingetragen wird.

Bezeichnung	Inhalt	Kommentar
Byte 3	6Ch	(ASCII Code für "l")
Byte 2	6Fh	(ASCII Code für "o")
Byte 1	61h	(ASCII Code für "a")
Byte 0	64h	(ASCII Code für "d")

Kommunikationsprofil – weitere Objekte Objekt 1018h: Identity Object

Informationen über den Hersteller und das Gerät:



1018 RECORD Device – Identifikation read only

Sub-Index 0h : Anzahl Subindices“
liefert den Wert 4

Sub-Index 1h: nur „read“

liefert die Vendor-ID (000000013h) Fritz Kübler GmbH

Sub-Index 2h: liefert den **Product-Code**
(z.B. 0x365821FF CANopen Geber)

Sub-Index 3h: nur „read“

liefert die Software -Revisionsnummer
(z.B. 102)

Sub-Index 4h: nur „read“

liefert die 8-stellige **Seriennummer** des Drehgebers

10 Emergency Objekte

Emergency Objekte treten bei fehlerhaften Situationen innerhalb eines CAN-Netzwerkes auf und werden je nach Ereignis ausgelöst und über den Bus mit einer **hohen Priorität** gesendet..

Wichtig: Ein Emergency Objekt wird nur einmal pro "Event" ausgelöst. Solange der Fehler besteht, wird kein neues Objekt generiert. Ist der Fehler behoben, wird ein erneutes Emergency-Objekt mit dem Inhalt 0 (Error Reset oder No Error) generiert und auf den Bus gesendet.

Unterstützte Error Codes

Die Error Codes sind **rot** hervorgehoben

Error Code (hex)	Meaning
00xx	Error Reset or No Error
10xx	Generic Error
20xx	Current
21xx	Current, device input side
22xx	Current inside the device
23xx	Current, device output side
30xx	Voltage
31xx	Mains Voltage
32xx	Voltage inside the device
33xx	Output Voltage
40xx	Temperature
41xx	Ambient Temperature
42xx	Device Temperature
50xx	Device Hardware
60xx	Device Software
61xx	Internal Software
62xx	User Software
63xx	Data Set
70xx	Additional Modules
80xx	Monitoring
81xx	Communication
8110	CAN Overrun (Objects lost)
8120	CAN in Error Passive Mode
8130	Life Guard Error or Heartbeat Error
8140	recovered from bus off
8150	Transmit COB-ID collision
82xx	Protocol Error
8210	PDO not processed due to length error
8220	PDO length exceeded
90xx	External Error
F0xx	Additional Functions
FFxx	Device specific

11 Emergency Nachricht

Byte	0	1	2	3	4	5	6	7
Content	Emergency Error Code (see Table 21)		Error register (Object 1001H)	Manufacturer specific Error Field				

Figure 34: Emergency Object Data

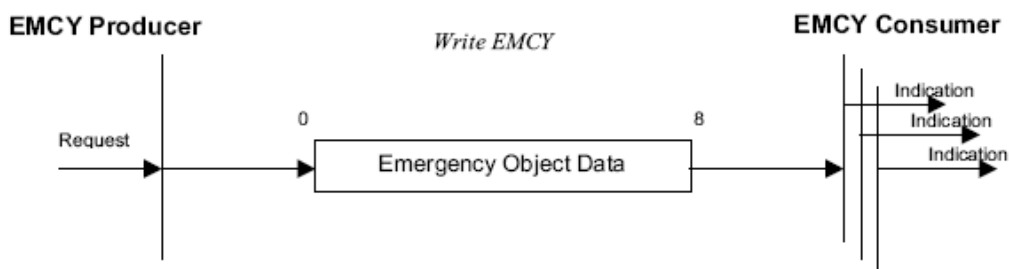
Beispiel einer Nachricht bei Hardwarefehler:

Transfer Data	00	52	09	80	56	20	50	2E
---------------	----	----	----	----	----	----	----	----

[Errcode]	5200	Hardwarefehler vom Sensor
[Error Register]	09	Fehler Register
[ManufacturerSpecific1]	80	Diagnosebytes
[ManufacturerSpecific2]	56	Diagnosebytes
[ManufacturerSpecific3]	20	Diagnosebytes
[ManufacturerSpecific4]	50	Diagnosebytes
[ManufacturerSpecific5]	2E	Diagnosebytes

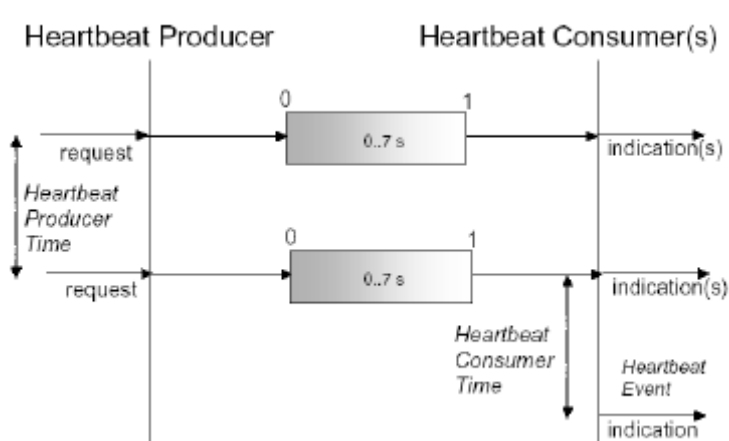
Emergency Protokoll

Eine „unbestätigte“ Service Nachricht ist definiert



Das Verhalten im Fehlerfall wird im **Objekt 1029h Error Behavior** beschrieben

12 Heartbeat Protokoll



Alternativ zum **Node-Guarding** sollte heute das moderne **Heartbeat-Protokoll** verwendet werden. Das Protokoll wird aktiviert, wenn im **Objekt 1017h** Producer Heartbeat Time ein Wert > 0 eingeschrieben wird.

Ein "Heartbeat-Producer" sendet zyklisch diese Heartbeat-Meldung. Ein oder mehrere "Heartbeat-Consumer" können diese Heartbeat-Meldung empfangen.

Falls das zyklische Senden dieser Heartbeat-Meldung ausbleibt, wird ein "Heartbeat Event" ausgelöst. Das Verhalten im Fehlerfall wird im Objekt 1029h Subindex 1 "Communication Error" definiert.

13 CANopen Objekt Verzeichnis

Die Beschreibung der Objektverzeichnis-Einträge ist folgendermaßen aufgebaut:

Index (hex)	Sub-Index (hex)	Objekt	Name	Typ	Attr.	M/O
-------------	-----------------	--------	------	-----	-------	-----

Index: 16 Bit-Adresse des Eintrages

Sub-Index: 8 Bit-Zeiger auf Untereintrag;

wird nur bei komplexen Datenstrukturen (z.B. Record, Array) verwendet;

wenn kein Untereintrag vorhanden: Sub-Index=0

Objekt:

NULL Eintrag ohne Daten

DOMAIN größere variable Datenmenge, z.B. Programmcode

DEFTYPE Definition der Datentypen, z.B. boolean, float, unsigned16 usw.

DEFSTRUCT Definition eines Record-Eintrages, z.B. PDO Mapping Struktur

VAR einzelner Datenwert, z.B. boolean, float, unsigned16, string usw.

ARRAY Feld mit gleichartigen Daten, z.B. unsigned16 Daten

RECORD Feld mit beliebig gemischten Datentypen

Name: kurze Beschreibung der Funktion

Typ: Datentyp, z.B. boolean, float, unsigned16, integer usw.

Attr.: Attribut gibt Zugriffsrechte auf das Objekt an:

rw Schreib- und Lesezugriff

ro nur Lesezugriff

const nur Lesezugriff, Wert ist eine Konstante

M/O

M Mandatory: Objekt muss im Gerät implementiert sein

O Optional: Objekt muss nicht im Gerät implementiert sein

Gliederung des gesamten Objektverzeichnisses:

Index (hex)	Objekt
0000	unbenutzt
0001 - 001F	statische Datentypen
0020 - 003F	komplexe Datentypen
0040 - 005F	herstellerspezifische Datentypen
0060 - 0FFF	reserviert
1000 - 1FFF	Kommunikations-Profil
2000 - 5FFF	herstellerspezifisches Profil
6000 - 9FFF	standardisiertes Geräteprofil
A000 - FFFF	reserviert

14 CANopen Kommunikationsprofil DS 301 V4.02

Kommunikationsobjekte

INDEX (hex)	OBJECT SYMBOL	ATTRIB	Name	M/O	TYPE
1000	VAR	CONST	Device Type	M	Unsigned32
1001	VAR	RO	Error Register	M	Unsigned8
1002	VAR	RO	Manufacturer Status	O	Unsigned32
1003	RECORD	RO	Predefined Error Field	O	Unsigned32
1004	ARRAY	RO	Number of PDO supported	O	Unsigned32
1005	VAR	RW	COB-ID Sync message	O	Unsigned32
1006	VAR	RW	Communication cycle period	O	Unsigned32
1007	VAR	RW	synchr.window length	O	Unsigned32
1008	VAR	CONST	Manufacturer Device Name	O	visible string
1009	VAR	CONST	Manufacturer Hardware Version	O	visible string
100A	VAR	CONST	Manufacturer Software Version	O	visible string
100B	VAR	RO	Node-ID	O	Unsigned32
100C	VAR	RW	Guard Time	O	Unsigned32
100D	VAR	RW	LifeTime Factor	O	Unsigned32
1010	VAR	RW	Store parameters (Device Profile)	O	Unsigned32
1011	VAR	RW	Restore parameters (Device Profile)	O	Unsigned32
1014	VAR	RO	COB_ID Emcy	O	Unsigned32
1015	VAR	RW	Inhibit Time Emcy	O	Unsigned32
1017	VAR	RW	Producer Heartbeat time	O	Unsigned16
1018	RECORD	RO	Identity Object	M	PDOComPar
1029	ARRAY	RW	Error Behaviour	O	Unsigned8
1800	RECORD		1 st transmit PDO Comm. Par.	O	PDOComPar
1801	RECORD		2 nd transmit PDO Comm. Par.	O	PDOComPar
1802	RECORD		3 rd transmit PDO Comm. Par.	O	PDOComPar
1A00	ARRAY		1 st transmit PDO Mapping Par.	O	PDOMapping
1A01	ARRAY		2 nd transmit PDO Mapping Par.	O	PDOMapping
1A02	ARRAY		3 rd transmit PDO Mapping Par.	O	PDOMapping

Herstellerspezifische Objekte

2100	VAR	RW	Baud Rate	O	Unsigned 8
2101	VAR	RW	Node number	O	Unsigned 8
2102	VAR	RW	CAN Bus Termination	O	Unsigned 8
2103	VAR	RO	Firmware Flash Version	O	Unsigned16
2105	VAR	RW	Save All Bus Parameters	O	Unsigned32
2140	Array	RW	Customer Memory	O	Unsigned32

15 CANopen Encoder Geräteprofil DS 406 V3.1

Gerätespezifische Objekte

INDEX (hex)	Object Symb.	ATTRIB	Name	M/O C2	TYPE
6000	VAR	RW	Operating parameters	M	unsigned16
6001	VAR	RW	Measuring Units p.Revolution (MUR)	M	unsigned32
6002	VAR	RW	Total Measuring Range (TMR)	M	unsigned32
6003	VAR	RW	Preset value	M	unsigned32
6004	VAR	RO	Position value	M MAP	unsigned32
6030	ARRAY	RO	Speed Value	O MAP	signed16
6040	ARRAY	RO	Acceleration Value	O	Signed16
6200	VAR	RW	Cyclic Timer	M	unsigned16
6400	ARRAY	RO	Working Area state	O MAP	unsigned 8
6401	ARRAY	RW	Working Area Low Limit	O	Unsigned32
6402	ARRAY	RW	Working Area High Limit	O	Unsigned32
6500	VAR	RO	Operating Status	M	unsigned16
6501	VAR	RO	Measuring Step (Singleturn)	M	unsigned32
6502	VAR	RO	Number of revolutions	M	unsigned16
6503	VAR	RO	Alarms	M MAP	unsigned16
6504	VAR	RO	Supported alarms	M	unsigned16
6505	VAR	RO	Warnings	M MAP	unsigned16
6506	VAR	RO	Supported warnings	M	unsigned16
6507	VAR	RO	Profile and SW version	M	unsigned32
6508	VAR	RO	Operating time	M	unsigned32
6509	VAR	RO	Offset value (calculated)	M	signed32
650A	VAR	RO	Module Identification	M	signed32
650B	VAR	RO	Serial Number	M	unsigned32

VAR = Variable

ARRAY = Array von Variablen

RW = Schreiben/Lesen

RO = Nur Lesen

const = Konstante

Name = Name des Objekts

M/O = Zwingend oder Optional.

MAP = Objekt kann gemappt werden

16 Objekte im Details - Encoder Profil DS 306 V3.1

Objekt 6000h Operating Parameters

- Bit 0: Codefolge: 0 = aufsteigend bei Drehung im Uhrzeigersinn (cw)
1 = aufsteigend bei Drehung gegen den Uhrzeigersinn (ccw)
Default: Bit = 0
- Bit 2: Skalierungsfunktion: 0 = disable, 1 = enable; Standard: Bit = 0 (s. Objekt 6001,6002)
Default: Bit = 1
- Bit14: Startup Mode: 0 = Bootup nach Pre-Operational, 1 = Bootup nach Operational
Default Bit = 0
- Bit15: Event Mode: 0 = Positionsausgabe lt. TPDO 1800h, 1 = jede Positionsänderung wird ausgegeben
Default Bit = 0



Bit	Function	Bit = 0	Bit =1	C1	C2
0	Codefolge	CW	CCW	m*	m*
1	Commissioning Diagnostic Control	Disabled	Enabled	o	o
2	Skalierung einschalten	Disabled	Enabled	o	m
3	Measuring direction	Forward	Reward	o**	o**
4..11	Reserved for further use				
12	Manufacturer specific parameter	N.A.	N.A.	o	o
13	Manufacturer specific parameter	N.A.	N.A.	o	o
14	Startup automatisch in OP-Mode	Disabled	Enabled	o	o
15	Event Mode Position	Disabled	Enabled	o	o

*m = Funktion muss unterstützt werden
o = optional

Objekt 6001h: Messschritte pro Umdrehung (Auflösung)

Dieser Parameter stellt die gewünschte Auflösung pro Umdrehung ein. Der Geber berechnet sich intern den entsprechenden Skalierungsfaktor. Der errechnete Skalierungsfaktor MURF (mit dem der physikalische Positionswert multipliziert wird) berechnet sich nach folgender Formel:

$$MUR = \text{Messschritte pro Umdrehung (6001h)} / \text{phys. Auflösung Singleturn (6501h)}$$

Dateninhalt:

Byte 0	Byte 1	Byte 2	Byte 3
$2^7 \dots 2^0$	$2^{15} \dots 2^8$	$2^{23} \dots 2^{16}$	$2^{31} \dots 2^{24}$



Wertebereich: 1....maximaler physikalischer Auflösung (16384) 14 Bit
Defaulteinstellung: 16384 (14 Bit)



Nach einer Änderung der Skalierung muss unbedingt der Presetwert ebenfalls auf Null/oder Wert gesetzt werden.

Objekt 6002h: Gesamtanzahl der Messschritte

Dieser Parameter stellt die Gesamtanzahl der Messschritte ein. Die maximale physikalische Auflösung wird mit einem Faktor beaufschlagt. Der Faktor ist immer < 1 . Nach der angegebenen Gesamtanzahl der Messschritte stellt sich der Drehgeber wieder auf Null.

Dateninhalt:

Byte 0	Byte 1	Byte 2	Byte 3
$2^7 \dots 2^0$	$2^{15} \dots 2^8$	$2^{23} \dots 2^{16}$	$2^{31} \dots 2^{24}$



Wertebereich: 1....maximaler physikalischer Auflösung (16384) 14 Bit

Defaulteinstellung: 16384 (14 Bit)



Nach einer Änderung der Gesamtanzahl muss unbedingt der Presetwert ebenfalls auf Null/oder Wert gesetzt werden.

Objekt 6003h: Preset-Wert

Der Positionswert des Gebers wird auf diesen Preset-Wert eingestellt.

Dadurch kann z.B. die Nullposition des Gebers mit dem Maschinen-Nullpunkt abgeglichen werden.

Dateninhalt:

Byte 0	Byte 1	Byte 2	Byte 3
$2^7 \dots 2^0$	$2^{15} \dots 2^8$	$2^{23} \dots 2^{16}$	$2^{31} \dots 2^{24}$



Wertebereich: 0....maximaler physikalischer Auflösung (16384) 14 Bit

Defaulteinstellung: 0

Objekt 6004h: Positionswert

Der Geber gibt den aktuellen (eventuell mit Skalierungsfaktor verrechneten) Positionswert aus

Dateninhalt:

Byte 0	Byte 1	Byte 2	Byte 3
$2^7 \dots 2^0$	$2^{15} \dots 2^8$	$2^{23} \dots 2^{16}$	$2^{31} \dots 2^{24}$

Wertebereich: 0....maximaler physikalischer Auflösung (16384) 14 Bit

0 .. 3FFFh



Objekt kann gemappt werden

Objekt 6030h: Speed Value

Der Geber gibt die aktuelle errechnete Geschwindigkeit (eventuell mit Skalierungsfaktor) als **vorzeichenbehafteter 16 Bit-Wert** aus. Die Geschwindigkeit wird in **U/min (rpm)** ausgegeben und wird mit einer fixen Synchronzeit von **60ms** berechnet.

Das Vorzeichen ist abhängig von der Drehrichtung der Achse. Negative Werte bedeuten eine linksdrehende Bewegung (**CCW**), während positive Werte rechtsdrehend (**CW**) ausgegeben werden.

Dateninhalt:

Byte 0	Byte 1
$2^7 \dots 2^0$	$2^{15} \dots 2^8$

Wertebereich: 0....maximale Geschwindigkeit 6500 U/min



Bei Werten größer 6500 U/min wird eine Warnmeldung ausgegeben und das Warning-Bit „Drehzahlüberschreitung Bit 0“ im Objekt Warnings 6505h gesetzt.



Objekt kann gemappt werden

Objekt 6200h: Cycle-Timer

Definiert die Zykluszeit, mit der die aktuelle Position mittels PDO 1 (siehe Objekt 1800h) ausgegeben wird. Die Timer gesteuerte Ausgabe wird aktiv, sobald eine Zykluszeit >0 eingetragen wird.



Dieses Objekt ist nur noch aus Kompatibilitätsgründen zu älteren Profilversionen vorhanden. Anstelle dieses Objekts sollte der Event Timer Subindex (05h) im aktuellen Transmit PDO verwendet werden.

Dateninhalt:

Byte 0	Byte 1
$2^7 \dots 2^0$	$2^{15} \dots 2^8$

Wertebereich: 0 ... FFFFh (65535) ergibt Zykluszeit in Millisekunden
Standard-Wert = 0h

Objekt 6500h: Operating Status anzeigen

Dieses Objekt zeigt den Status der programmierten Einstellungen von Objekt 6000h an.

Dateninhalt:

Byte 0	Byte 1
$2^7 \dots 2^0$	$2^{15} \dots 2^8$

Dateninhalt: siehe Objekt 6000h

Objekt 6503h: Alarms

Zusätzlich zu den Fehlern die über Notfall-Nachrichten (emergency messages) gemeldet werden, bietet das Objekt 6503h weitere Fehlermeldungen. Das zugehörige Fehlerbit wird auf 1 gesetzt, solange der Fehler anliegt.

Dateninhalt:

Byte 0	Byte 1
$2^7 \dots 2^0$	$2^{15} \dots 2^8$

Bit Nr.	Bezeichnung	Wert = 0	Wert = 1
Bit 0	Positionsfehler	Positionswert gültig	Positionsfehler
Bit 1	Hardwareprüfung	kein Fehler	Fehler
Bit 2..15	Unbenutzt		

In den beiden Fällen wird beim Auftreten eines Alarmes gleichzeitig eine Notfall-Nachricht (**ID=80h+Knotennummer**) mit dem Fehlercode **1000h (Generic error)** gesendet.



Objekt kann gemappt werden

Objekt 6504h: Supported Alarms

Über dieses Objekt wird angezeigt, welche Alarmmeldungen vom Geber unterstützt werden (siehe Objekt 6503h).

Dateninhalt:

Byte 0	Byte 1
$2^7 \dots 2^0$	$2^{15} \dots 2^8$

Wertebereich s. Objekt 6503h

Bit gesetzt auf 1 bedeutet, dass die Alarmmeldung unterstützt wird

Beispiel:

Bit 0 = 1 Positionsfehleranzeige wird unterstützt

Objekt 6505h: Warnings

Warnmeldungen zeigen an, dass Toleranzen interner Geberparameter überschritten sind. Bei einer Warnmeldung kann der Messwert, anders als bei Alarmmeldung oder Notfallnachricht, trotzdem gültig sein. Das zugehörige Warnbit wird auf 1 gesetzt, solange die Toleranzüberschreitung oder Warnung anliegt.

Dateninhalt:

Byte 0	Byte 1
$2^7 \dots 2^0$	$2^{15} \dots 2^8$

Bit Nr.	Bezeichnung	Wert = 0	Wert = 1
Bit 0	Drehzahlüberschreitung	keine	überschritten
Bit 1	unbenutzt		
Bit 2	Watchdog Status	System in Ordnung	Reset ausgeführt
Bit 3..15	Unbenutzt		

Bei aktivem Bit 0 wird gleichzeitig eine Emergency-Nachricht (ID=80h+Knotennummer) mit dem **Fehlercode 4200h** (Device specific) gesendet.

Bei aktivem Bit 2 oder 3 wird gleichzeitig eine Notfall-Nachricht (ID=80h+Knotennummer) mit dem **Fehlercode 5200h** (Device Hardware) gesendet.



Objekt kann gemappt werden

Objekt 6506h: Supported Warnings

Über dieses Objekt wird angezeigt, welche Warnmeldungen vom Geber unterstützt werden (siehe Objekt 6505h).

Dateninhalt:

Byte 0	Byte 1
$2^7 \dots 2^0$	$2^{15} \dots 2^8$

Wertebereich s. Objekt 6505h

Bit gesetzt auf 1 bedeutet, dass die Warnung unterstützt wird

Objekt 6400h: Working Area State Register

Diese Objekt beinhaltet den aktuellen Status der Encoder-Position in Abhängigkeit zu den programmierten Limits. Je nach Position der beiden Endwerte werden die Flags gesetzt oder rückgesetzt. Der Vergleich mit beiden Endwerten findet in „Echtzeit“ statt und kann zur Echtzeit-Positionierung oder zur Endabschaltung verwendet werden.

Work_area_state							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
					kleiner als LowLimit1	größer als HighLimit1	außerhalb Bereich1

Wertebereich 8-Bit Dateninhalt s. Bit 0...7



Damit die Ausgangssignale richtig aktiviert werden, sind die beiden Endwerte Objekt 6401h und 6402h zu überprüfen !



Objekt kann gemappt werden

Objekt 6401h: Working Area Low Limit Objekt 6402h: Working Area High Limit

Diese beiden Parameter stellen den Arbeitsbereich ein. Innerhalb und außerhalb dieses Bereiches kann der Status über Flag-bytes (**Objekt 6400h Working Area State**) gemeldet werden. Diese Bereichsmarker können auch als Software-Endschalter verwendet werden.

Dateninhalt:

Byte 0	Byte 1	Byte 2	Byte 3
$2^7 \dots 2^0$	$2^{15} \dots 2^8$	$2^{23} \dots 2^{16}$	$2^{31} \dots 2^{24}$



Wertebereich: 1....maximaler physikalischer Auflösung (16384) 14 Bit
Defaulteinstellung: 16384 (14 Bit) Working Area High Limit
0 Working Area Low Limit

Objekt 2100h: Baudrate

Über dieses Objekt kann die Baudrate softwaremäßig verändert werden. Standardmäßig ist der Wert auf FFh eingestellt, d.h. die hardwaremäßige Einstellung der Baudrate hat Priorität. Wird der Wert zwischen 1..9 eingestellt und der Parameter über das **Object 2105h Save All Bus Parameters** gespeichert, so bootet das Gerät beim erneuten Einschalten oder Reset Node mit der geänderten Baudrate.

Byte 0
$2^7 \dots 2^0$

Dateninhalt:



Wertebereich 1 ...8
Defaulteinstellung: 05h

Die Übernahme einer neuen Baudrate erfolgt erst beim nächsten Hochlaufen (Reset/Power-on) des Gebers oder über einen **NMT-Reset Node** Befehl.

Objekt 2101h: Knotenadresse

Über dieses Objekt kann die Knotenadresse softwaremäßig verändert werden. Standardmäßig ist der Wert auf 0x3Fh eingestellt. Wird der Wert zwischen 1..127 eingestellt und der Parameter über das **Objekt 2105h Save All Bus Parameters** gespeichert, so bootet das Gerät beim erneuten Einschalten oder Reset Node mit der geänderten Knotenadresse.

Dateninhalt:

Byte 0
$2^7 \dots 2^0$



Wertebereich 1 ...127 oder 1..7Fh

Defaulteinstellung: 0x3Fh

Die **Knotennummer 0** ist reserviert und darf von keinem Knoten verwendet werden. Die resultierenden Knotennummern liegen im Bereich **1...7Fh** hexadezimal oder (1...127)



Die Übernahme einer neuen Knotennummer erfolgt erst beim nächsten Hochlaufen (Reset/Power-on) des Gebers oder über einen **NMT-Reset Node** Befehl.

Objekt 2102h: CAN-Busterminierung aus/ein

Über dieses Objekt kann die Busterminierung softwaremäßig zugeschaltet werden. Standardmäßig ist der Wert auf 1 eingestellt.

Dateninhalt:

Byte 0
$2^7 \dots 2^0$



Wertebereich 0..1

Defaulteinstellung: 1

Objekt 2103h: Firmware Flashversion

Über dieses Objekt wird die aktuelle Firmwareversion als 16-Bit Hexadezimalwert angezeigt. Dieser Wert dient zur Verifizierung auf den aktuellen Stand des Gerätes.

Dateninhalt:

Byte 0	Byte 1
$2^7 \dots 2^0$	$2^{15} \dots 2^8$

Wertebereich bis FFFFh

Beispiel: **4FA6h** aktuelle Firmware

Objekt 2105h: Save All Bus Parameters

Dieser Parameter speichert die gewünschten Busparameter (Objekt 2100h ,2101h,2102h) permanent im Flash-Speicher. Dieses Objekt dient als zusätzliche Absicherung vor ungewolltem Ändern der Baudrate und Knotenadresse.

Erst durch gezieltes Abspeichern mit dem Parameter „**save**“ (hexadezimal **0x65766173**) werden die Busparameter **Baudrate, Knotenadresse und Terminierung** permanent abgespeichert.



Byte 0	Byte 1	Byte 2	Byte 3
$2^7 \dots 2^0$	$2^{15} \dots 2^8$	$2^{23} \dots 2^{16}$	$2^{31} \dots 2^{24}$

Wertebereich: „**save**“ in hexadezimal **0x65766173**



Objekt 2140h: Customer Memory (16 Bytes)

Diese 4 Parameter stellen einen Speicherbereich für den Anwender dar. Es sind **4 Datenwörter mit maximal 4 Bytes speicherbar**. Dieses Bereich wird nicht auf Inhalt geprüft, d.h. jegliches Format kann abgelegt werden.



Byte 0	Byte 1	Byte 2	Byte 3
$2^7 \dots 2^0$	$2^{15} \dots 2^8$	$2^{23} \dots 2^{16}$	$2^{31} \dots 2^{24}$

Wertebereich: Ziffern, Alphanumerisch

Defaulteinstellung: 0

Objekt 1029h Error Behavior

Wird ein ernsthafter Fehler erkannt, sollte das Gerät automatisch in den **Pre-Operational** Modus wechseln. Innerhalb dieses Objektes kann eingestellt werden, wie sich das Gerät beim Auftreten eines Fehlerfalles verhalten soll. Folgende Fehlerklassen werden abgedeckt:

1029h,Subindex 1 Kommunikationsfehler Default = 0

- Bus-off Zustand des CAN Interfaces
- Life guarding Ereignis ist aufgetreten
- Heartbeat Überwachung ist fehlgeschlagen

1029h,Subindex 2 Device Profile Specific Default = 1

- Sensorfehler und Controllerfehler
- Temperaturfehler

1029h,Subindex 3 Manufacturer Specific Default = 1

- interner Fehler

Der Wert der Objektklassen setzt sich folgendermaßen zusammen:

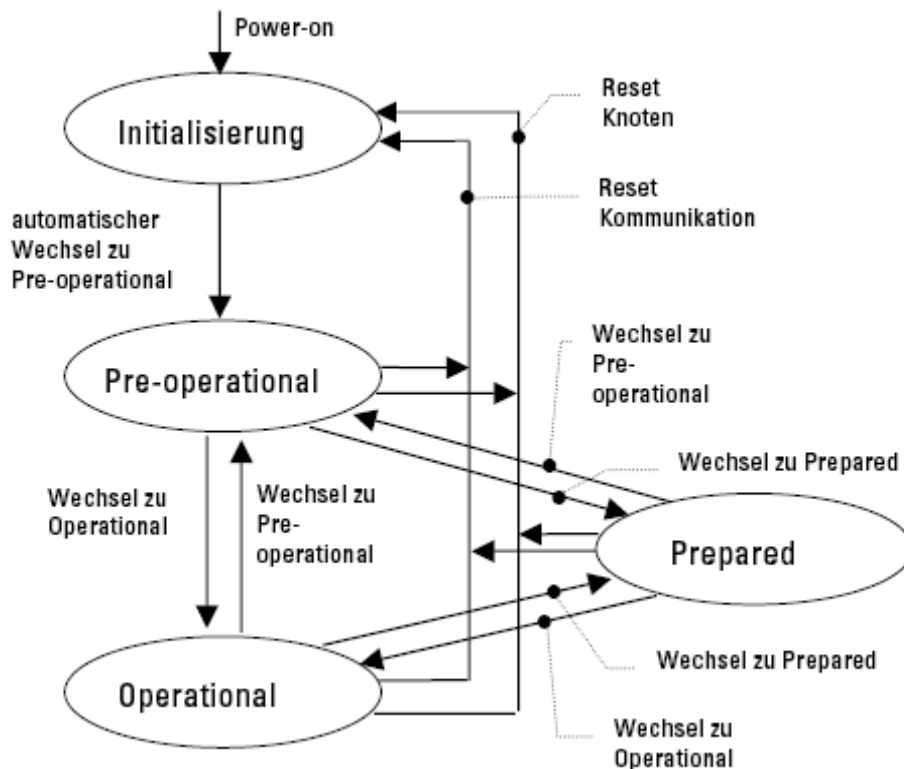
Byte 0
$2^7 \dots 2^0$

Wertebereich 8-Bit

- **0** Pre-Operational Modus (nur wenn zuvor Operational-Modus aktiv war)
- **1** keine Änderung des Modus
- **2** Stopped- Modus
- **3 .. 127** reserviert

17 Netzwerkmanagement

Der Geber unterstützt das im Profil für "minimum capability devices" definierte, vereinfachte Netzwerkmanagement (minimum boot up). Folgendes Zustandsdiagramm nach DS 301 zeigt die unterschiedlichen Knoten-Zustände und die entsprechenden Netzwerk-Kommandos (gesteuert vom Netzwerk-Master über NMT-Dienste):



Initialisierung: Nach einem Reset des Gerätes oder nach dem Einschalten ist dies der Ausgangszustand nach Anlegen der Versorgungsspannung. Der Knoten wechselt nach Durchlauf der Reset-/Initialisierungsroutinen automatisch in den Zustand Pre-operational. Die LED's zeigen den momentanen Status an.

Pre-operational: Der CAN-Knoten kann nun über SDO-Nachrichten oder mit NMT-Befehle unter dem Standard-Identifizier angesprochen werden. Es erfolgt die Programmierung der Geber- oder Kommunikations-Parameter.

Operational: Der Knoten ist aktiv. Prozesswerte werden über die PDO's ausgegeben. Alle NMT-Kommandos können ausgewertet werden.

Prepared oder Stopped: In diesem Zustand ist der Knoten nicht mehr aktiv, d.h. sowohl eine SDO- als auch eine PDO-Kommunikation ist nicht möglich. Der Knoten kann über NMT-Kommandos entweder in den Zustand Operational oder Pre-operational gesetzt werden.

18 NMT-Kommandos

Sämtliche NMT-Kommandos werden als unbestätigtes NMT-Objekt übertragen. Durch das Broadcast (netzwerkweite) Kommunikationsmodell werden die NMT-Kommandos von jedem Teilnehmer erkannt. Ein NMT-Objekt ist folgendermaßen aufgebaut:

COB-ID = 0

Byte 0	Byte 1
$2^7 \dots 2^0$	$2^{15} \dots 2^8$

Byte 0 = Kommandobyte
Byte 1 = Knoten-Nummer



Der COB-ID des NMT-Objektes ist immer 0

Über die Knoten-Nummer wird der Knoten adressiert. Bei Knoten-Nummer 0 werden alle Knoten angesprochen.

Kommandobyte (hex)	Beschreibung
01h	Start_Remote_Node: Wechsel zu Operational
02h	Stop_Remote_Node: Wechsel zu Prepared
80h	Enter_Pre-Operational_State: Wechsel zu Pre-operational
81h	Reset_Node: Reset Knoten ¹
82h	Reset_Communication: Reset Kommunikation ²

¹ Alle Parameter des gesamten Objektverzeichnisses werden auf Power-On Werte gesetzt.


² Nur die Parameter im Abschnitt Kommunikationsprofil des Objektverzeichnisses werden auf Power-On Werte gesetzt.

19 LED-Anzeigen während des Betriebes

grüne LED = BUS Status

rote LED = ERR Anzeige



Anzeige	LED	Bedeutung	Fehlerursache	Zusatz
Bus aus		Keine Verbindung zum Master ²	Datenleitungsunterbrechung Falsche Baudrate Vertauschte Datenleitung	Kombination mit ERR LED beachten Wenn ERR-LED auch aus ist, bitte Spannungsversorgung prüfen ³
Bus Blinkend ca. 250ms		Verbindung zum Master Pre-operational Status		SDO Kommunikation
Bus Blinkend ca. 1Sec		Verbindung zum Master Stopped Status		SDO Kommunikation nicht möglich Nur NMT Befehle
Bus Ein		Verbindung zum Master Operational Status		PDO Transfer ist aktiv
ERR aus		Gerät arbeitet fehlerfrei		Kombination mit BUS LED beachten
ERR blinkend		Verbindung zum Master unterbrochen	Kombination mit BUS-Status	BUS-LED grün blinkend oder an- ist abhängig von Objekt 1029h Error Behaviour
ERR Ein		BUS-OFF Status	Kurzschluß am Bus oder Falsche Baudrate	
Bus+ERR blinkend		LSS Global Modus eingestellt	Gerät wartet auf LSS Kommando	LSS Mode kann über globales Kommando verlassen werden

Die einzelnen LED-Anzeigen können auch in Kombinationen auftreten.

² Master kann SPS oder 2-ter Kommunikationspartner sein

³ Betriebsspannung

20 Definitionen

Symbolerklärung:



Dieses Symbol steht bei Textstellen, die besonders zu beachten sind, damit der ordnungsgemäße Einsatz gewährleistet ist und Gefahren ausgeschlossen werden. Dieses Symbol gibt wichtige Hinweise für den sachgerechten Umgang mit dem Drehgeber. Das Nichtbeachten dieser Hinweise kann zu Störungen an dem Drehgeber oder in der Umgebung führen.



Dieses Symbol weist auf eine Besonderheit hin



Defaulteinstellung der Parameter ab Werk

21 Verwendete Abkürzungen

CAL	CAN Application Layer. Anwendungsschicht (Schicht 7) im CAN Kommunikations-Modell
CAN	Controller Area Network
CiA	CAN in Automation. Internationaler Verein der Anwender und Hersteller von CANProdukten
CMS	CAN Message Specification. Service-Element von CAL
COB	Communication Object. Transporteinheit im CAN Netzwerk (CAN Nachricht). Daten werden innerhalb eines COB's über das Netzwerk gesendet.
COB-ID	COB-Identifizier. Eindeutige Kennung einer CAN-Nachricht. Der Identifizier bestimmt die Priorität des COB's im Netzwerk.
DBT	Distributor. Service-Element von CAL, verantwortlich für die dynamische Vergabe von Identifiern.
DS	Draft Standard; Normentwurf
DSP	Draft Standard Proposal; Normentwurfs-Vorschlag
ID	Identifizier, siehe COB-ID
LMT	Layer Management. Service-Element von CAL, verantwortlich für die Konfiguration der Parameter in den einzelnen Schichten des Kommunikationsmodells.
LSB	Least Significant Bit/Byte; niederwertigstes Bit/Byte
MSB	Most Significant Bit/Byte; höchstwertigstes Bit/Byte
	NMT Network Management. Service-Element von CAL, verantwortlich für die Initialisierung, Konfiguration und Fehlerbehandlung im Netzwerk.
OSI	Open Systems Interconnection. Schichtenmodell zur Beschreibung der Funktionsbereiche in einem Datenkommunikationssystem.
PDO	Process Data Object. Objekt für den Austausch von Prozessdaten.
RTR	Remote Transmission Request; Datenanforderungstelegramm
SDO	Service Data Object; Kommunikationsobjekt, über das der Master auf das Objektverzeichnis eines Knotens zugreifen kann.
SYNC	Synchronisations-Telegramm. Busteilnehmer antworten mit ihrem Prozesswert auf das SYNC-Kommando

22 Dezimal-Hexadezimal Umrechnungstabelle

Bei Zahlenangaben werden dezimale Werte als Ziffern ohne Zusatz angegeben (z.B. 1408), binäre Werte werden mit b (z.B. 1101b) und hexadezimale Werte mit h (z.B. 680h) hinter den Ziffern gekennzeichnet.

Dez	Hex	Dez	Hex	Dez	Hex	Dez	Hex
0	00	32	20	64	40	96	60
1	01	33	21	65	41	97	61
2	02	34	22	66	42	98	62
3	03	35	23	67	43	99	63
4	04	36	24	68	44	100	64
5	05	37	25	69	45	101	65
6	06	38	26	70	46	102	66
7	07	39	27	71	47	103	67
8	08	40	28	72	48	104	68
9	09	41	29	73	49	105	69
10	0A	42	2A	74	4A	106	6A
11	0B	43	2B	75	4B	107	6B
12	0C	44	2C	76	4C	108	6C
13	0D	45	2D	77	4D	109	6D
14	0E	46	2E	78	4E	110	6E
15	0F	47	2F	79	4F	111	6F
16	10	48	30	80	50	112	70
17	11	49	31	81	51	113	71
18	12	50	32	82	52	114	72
19	13	51	33	83	53	115	73
20	14	52	34	84	54	116	74
21	15	53	35	85	55	117	75
22	16	54	36	86	56	118	76
23	17	55	37	87	57	119	77
24	18	56	38	88	58	120	78
25	19	57	39	89	59	121	79
26	1A	58	3A	90	5A	122	7A
27	1B	59	3B	91	5B	123	7B
28	1C	60	3C	92	5C	124	7C
29	1D	61	3D	93	5D	125	7D
30	1E	62	3E	94	5E	126	7E
31	1F	63	3F	95	5F	127	7F

23 Glossar

Baudrate

Die Baudrate ist die Übertragungsgeschwindigkeit. Sie steht in Zusammenhang mit dem nominellen Bit-Timing. Die maximal mögliche Baudrate ist von vielen Faktoren, welche die Laufzeit der Signale auf dem Bus beeinflussen, abhängig. Ein wesentlicher Zusammenhang besteht zwischen der maximalen Baudrate und der Buslänge und dem Kabeltyp. In CANopen sind verschiedene Baudraten zwischen 10 kbit/s und 1 Mbit/s definiert.

CANopen

CANopen ist ein auf CAN basierendes Protokoll, welches ursprünglich für industrielle Steuerungssysteme entwickelt wurde. Die Spezifikationen beinhalten sowohl verschiedene Geräteprofile, als auch den Rahmen für spezifische Anwendungen. CANopen Netzwerke werden auch in Off-Road Fahrzeugen, Schiffselektronik, medizinischen Geräten und Zügen verwendet. Der sehr flexible Application Layer und die vielen optionalen Features sind ideal für zugeschnittene Lösungen. Weiterhin gibt es eine Vielzahl von Konfigurationstools. Auf dieser Basis kann der Anwender anwendungsspezifische Geräteprofile definieren. Weitere Informationen zu CANopen finden Sie im Internet unter www.can-cia.org.

EDS-Datei

Die EDS-Datei (Electronic Data Sheet) wird vom Hersteller eines CANopen-Gerätes bereitgestellt. Sie hat ein standardisiertes Format für die Beschreibung von Geräten. Die EDS-Datei beinhaltet Informationen über:

- Beschreibung der Datei (Name, Version, Erstellungsdatum, u.a.)
- Allgemeine Geräteinformationen (Herstellername und -code)
- Gerätename und -typ, Version, LMT-Adresse
- unterstützte Baudraten sowie Boot-Up-Fähigkeit
- Beschreibung der unterstützten Objekte über deren Attribute.

Knotennummer

Innerhalb eines CANopen-Netzwerkes wird jedes Gerät über seine Knotennummer (Node-ID) identifiziert. Die erlaubten Knotennummern liegen im Bereich von 1-127 und dürfen nur einmal innerhalb eines Netzwerkes vorkommen.

Netzwerkmanagement

In einem verteilten System fallen verschiedene Aufgaben im Zusammenhang mit der Konfiguration, Initialisierung und Überwachung der Netzwerkteilnehmer an. Das in CANopen definierte Dienstelement »Netzwerkmanagement (NMT)« stellt diese Funktionalität zur Verfügung.

PDO

Die Prozessdatenobjekte (PDO) stellen die eigentlichen Transportmittel für die Übertragung von Prozessdaten (Anwendungsobjekten) dar. Ein PDO wird von einem Producer gesendet und kann von einem oder mehreren Consumern empfangen werden.

PDO-Mapping

Die Größe eines PDOs kann bis zu 8 Byte betragen. Es kann benutzt werden, um mehrere Anwendungsobjekte zu transportieren. Das PDO-Mapping beschreibt die Festlegung über die Anordnung der Anwendungsobjekte innerhalb des Datenfeldes des PDOs.

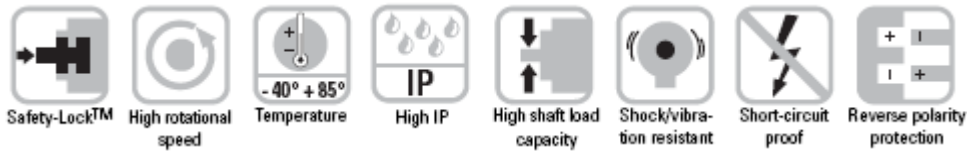
SDO

Über Dienstdatenobjekte (Service Data Objects, SDO) erfolgt der bestätigte Transfer von Daten beliebiger Länge zwischen zwei Netzteilnehmern. Der Datentransfer findet im Client-Server-Mode statt.



Sendix[®] absolut Absolute Singleturn Encoder

Series M3658, M3678



CANopen
certified
CiA200704-301V402/20-0071



© Fritz Kübler GmbH

Copyright Protection

The contents of this documentation are protected by copyright © Fritz Kübler GmbH. The contents of this documentation may not be altered, expanded, reproduced nor circulated to third parties, without the prior written agreement of Fritz Kübler GmbH.

Liability to modification without notice

As a result of ongoing efforts to improve our products, we reserve the right to make changes at any time to technical information contained in the document to hand.

Warranty Disclaimer

Fritz Kübler GmbH provides no guarantee, neither tacit nor express, in respect of the whole manual (whether this applies to the original German text or to the English translation) and assumes no liability for any damage, neither direct nor indirect, however caused.

Document information

Revised 4-2008

Screen printouts used

CANalyzer® Fa. Vector –Informatik

Fritz Kübler GmbH Schubertstr.47

78054 VS-Schwenningen / Germany

Tel. +49 (0) 7720-3903-0

Fax +49 (0) 7720-21564

E-Mail: info@kuebler.com

Internet: www.kuebler.com

Table of Contents

1	GENERAL	1-5
	CANOPEN SINGLETURN ENCODER SERIES M36X8.....	1-5
	THE CANOPEN COMMUNICATION PROFILE DS 301 V4.02	1-5
	ENCODER DEVICE PROFILE DS 406 V3.1	1-6
	OBJECTIVES OF LSS	1-6
	DATA TRANSMISSION	1-6
	OBJECTS AND FUNCTION CODE IN THE PREDEFINED CONNECTION SET	1-7
	BROADCAST (NETWORK-WIDE) OBJECTS.....	1-7
	PEER-TO PEER (DEVICE-TO-DEVICE) OBJECTS	1-7
	RESTRICTED, RESERVED OBJECTS	1-7
2	TRANSMISSION OF PROCESS DATA	2-8
	TRANSMIT PDO 1 (1800H) POSITION ASYNCHRONOUS	2-8
	TRANSMIT PDO2 (1801H) VELOCITY ASYNCHRONOUS	2-8
	TRANSMIT PDO3 (1802H) POSITION SYNCHRONOUS	2-8
3	TRANSMISSION OF SERVICE DATA	3-9
	EXAMPLE: TRANSMISSION OF SERVICE DATA TO AND FROM THE ENCODER	3-10
	LSS HARDWARE RESTRICTIONS (LSS ADDRESS)	3-10
	LSS OPERATING RESTRICTIONS	3-10
	LSS CONFIGURATION AND THE OPERATION MODES.....	3-10
4	POWER SUPPLY AND CAN-BUS CONNECTION	4-11
	POWER SUPPLY	4-11
	CAN - M12 CONNECTOR	4-11
	MECHANICAL CHARACTERISTICS	4-11
5	INITIAL STARTUP - GENERAL DEVICE SETTINGS	5-11
	BAUDRATE	5-11
	NODE NUMBER	5-12
	CAN-BUS TERMINATION	5-12
	SAVE ALL BUS PARAMETERS	5-12
6	LAYER SETTING SERVICES (LSS)	6-13
7	DEFAULT SETTINGS ON DELIVERY	7-15
	COMMUNICATION PARAMETER.....	7-15
	ENCODER PROFILE.....	7-16
8	COMMUNICATION PARAMETERS	8-17
	DEFINITION OF THE TRANSMISSION TYPE OF THE PDO	8-19
	VARIABLE PDO MAPPING.....	8-19
	STRUCTURE OF A MAPPING ENTRY	8-20
9	EXAMPLE OF A VARIABLE MAPPING TABLE	9-21
	MAPPING OBJECT 1A00H.....	9-21
	MAPPING TABLE OF OBJECT 1A00H:	9-21
10	APPLICATION PROGRAMMING EXAMPLE:	10-22
	SETTING UP OBJECTS.....	10-22
	<i>Total Measuring Range set to 3600</i>	10-22
	<i>Measuring Units per Revolution – limit to 3600</i>	10-22
	<i>Set Preset Value to 0</i>	10-23
	<i>Set the values of Transmit Parameters TPDO1 and TPDO2</i>	10-23
	<i>Producer Heartbeat - set to 500 ms</i>	10-24
	<i>Set Work Area low and high limit values</i>	10-24
	<i>Save all modified parameters in the EEPROM Store Parameters 1010h</i>	10-25
	<i>Object 1010h Store Parameters</i>	10-25
	<i>Object 1011h: Load Standard Values</i>	10-25
	COMMUNICATION PROFILE – FURTHER OBJECTS	10-26

Technical Manual

Absolute Singleturn Encoder **CANopen** series M36X8



OBJECT 1018H: IDENTITY OBJECT	10-26
11 EMERGENCY OBJECTS	11-27
ERROR CODES SUPPORTED	11-27
12 EMERGENCY MESSAGE	12-28
EXAMPLE OF AN OVER-TEMPERATURE MESSAGE:.....	12-28
EMERGENCY PROTOCOL.....	12-28
13 HEARTBEAT PROTOCOL.....	13-28
14 CANOPEN OBJECT DICTIONARY	14-29
STRUCTURE OF THE ENTIRE OBJECT DICTIONARY:.....	14-29
15 CANOPEN COMMUNICATION PROFILE DS 301 V4.02	15-30
COMMUNICATION OBJECTS.....	15-30
MANUFACTURER SPECIFIC OBJECTS	15-30
16 CANOPEN ENCODER DEVICE PROFILE DS 406 V3.1	16-31
DEVICE-SPECIFIC OBJECTS.....	16-31
17 OBJECTS IN DETAIL - ENCODER PROFILE DS 306 V3.1 .17-32	
OBJECT 6000H OPERATING PARAMETERS	17-32
OBJECT 6001H: MEASURING STEPS PER REVOLUTION (RESOLUTION)	17-32
AFTER CHANGING THE MEASURING STEP IT IS NECESSARY TO SET THE PRESET VALUE ALSO TO	ZERO /OR A
VALUE.OBJECT 6002H: TOTAL NUMBER OF MEASURING STEPS	17-32
OBJECT 6002H: TOTAL NUMBER OF MEASURING STEPS	17-33
OBJECT 6003H: PRESET VALUE	17-33
OBJECT 6004H: POSITION VALUE	17-33
OBJECT IS MAPPABLE OBJECT 6030H: SPEED VALUE	17-33
OBJECT 6030H: SPEED VALUE	17-34
OBJECT 6200H: CYCLIC TIMER.....	17-34
OBJECT 6500H: DISPLAY OPERATING STATUS.....	17-34
OBJECT 6503H: ALARMS	17-35
OBJECT 6504H: SUPPORTED ALARMS	17-35
OBJECT 6505H: WARNINGS.....	17-36
OBJECT 6506H: SUPPORTED WARNINGS	17-36
OBJECT 6400H: WORKING AREA STATE REGISTER.....	17-37
OBJECT 6401H: WORKING AREA LOW LIMIT	17-37
OBJECT 6402H: WORKING AREA HIGH LIMIT	17-37
OBJECT 2100H: BAUD RATE.....	17-37
OBJECT 2101H: NODE ADDRESS.....	17-38
OBJECT 2102H: CAN BUS TERMINATION OFF/ON	17-38
OBJECT 2103H: FIRMWARE FLASH VERSION.....	17-38
OBJECT 2105H: SAVE ALL BUS PARAMETERS	17-39
OBJECT 2140H: CUSTOMER MEMORY (16 BYTES).....	17-39
OBJECT 1029H ERROR BEHAVIOUR	17-39
18 NETWORK MANAGEMENT	18-40
19 NMT COMMANDS.....	19-41
20 LED STATES.....	20-42
GREENLED = BUS STATE	20-42
RED LED = ERR DISPLAY	20-42
21 DEFINITIONS.....	21-43
22 ABBREVIATIONS USED	22-43
23 DECIMAL-HEXADECIMAL CONVERSION TABLE.....	23-44
24 GLOSSARY	24-45

1 General

CANopen Singleturn Encoder Series M36X8

The CANopen encoder of Series 36X8 support the latest CANopen communication profile according **DS 301 V4.02** . In addition a device-specific encoder profile **DS 406 V3.1** is implemented.

The following operating modes can be selected: Polled Mode, Cyclic Mode, Sync Model. Moreover, scale factors, preset values, limit switch values and many other additional parameters can be programmed via the CAN-Bus. At Power ON all parameters are loaded from an EEPROM, which had previously been saved in the non-volatile memory to protect them in case of power failure. The following output values may be freely combined as **PDO** (PDO Mapping): **position, speed** as well as the status of the two **limit switches, alarms** and **warnings**.

The encoders are available with a **connector** or a **cable connection** - all changes to the device address and baud rate are software controlled.

One LED located on the back indicate the operating or fault status of the CAN bus, as well as the status of an internal diagnostic. CANopen encoders are available in blind hollow shaft and solid shaft versions, and are ideal for use in harsh industrial environments thanks to their IP 65 protection rating.

The CANopen Communication Profile DS 301 V4.02

CANopen represents a unified user interface and thus allows for a simplified system structure with a wide variety of devices. CANopen is optimized for the fast exchange of data in real-time systems and possesses a number of different device profile that have been standardized. The CAN in Automation (CiA) manufacturers and users group is responsible for creating and standardization of the relevant profiles.

CANopen offers

- user-friendly access to all device parameters.
- auto-configuration of the network and of the devices
- device synchronization within the network
- cyclic and event-driven process data exchange
- simultaneous read and write of data

CANopen uses four communication objects (COB) with different properties

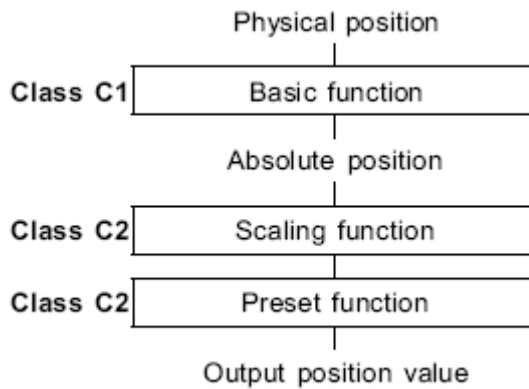
- Process Data Objects (PDO) for real-time data,
- Service Data Objects (SDO) for transmitting parameters and programs,
- Network Management (NMT, Life-Guarding, Heartbeat)
- Predefined Objects (for Synchronisation, Time-Stamp, Emergency)

All device parameters are filed in an **Object Dictionary**. This Object Dictionary contains the description, data type and structure of the parameters, as well as the address (Index).

The dictionary is divided into a communications profile section, a section covering the device profile as well as a section specific to the manufacturer.

Encoder Device Profile DS 406 V3.1

This profile describes a **vendor-independent** mandatory definition of the interface with regard to encoders. It is laid down in the profile, which CANopen functions are to be used as well as how they are to be used. This standard thus makes possible an open vendor-independent bus system. The device profile is broken down into two Object classes:



- **Class C1** describes all the basic functions that the encoder must contain

- **Class C2** contains numerous extended functions, which must either be supported by encoders of this class (Mandatory) or which are optional. Class 2 devices thus contain all C1 and C2 mandatory functions, as well as additional optional functions dependent on the manufacturer. An address range is also defined in the profile to which the manufacturer's own special functions can be assigned.

Objectives of LSS

CiA DSP 305 CANopen *Layer Setting Service and Protocol (LSS)* services and protocols were created to enable the following parameters to be read and changed through the network:

- The CANopen Node ID
- The CAN baud rate
- The LSS address

This increases the “plug-and-play” capabilities of devices on CANopen networks as preconfiguration of the network is less restrictive. The LSS Master is responsible for configuring these parameters on one or more LSS Slaves on a CANopen network.

Data transmission

With CANopen data are transferred via two different communication types (COB=Communication Object) with different properties:

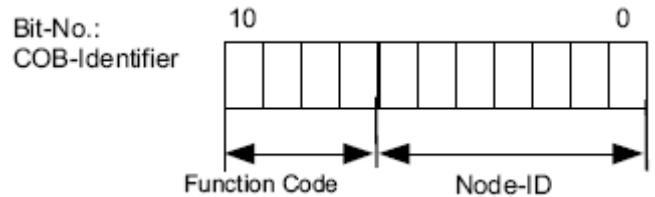
- **Process Data Objects (PDO – real-time capable)**
- **Service Data Objects (SDO)**

The Process Data Objects (**PDO**) provide high-speed exchange of real-time data (e.g. encoder position, speed, comparative position status) with a maximum length of 8 byte. These data are transmitted with a high priority (low COB-Identifier). PDOs are broadcast messages and provide their real-time data simultaneously to all desired receivers. PDOs can be mapped, i.e. 4 byte of position and 2 byte of speed can be combined in one 8 byte data word.

The Service Data Objects (**SDO**) form the communication channel for the transfer of device parameters (e.g. encoder resolution programming). As these parameters are transmitted acyclically (e.g. only once during boot-up of the network), the SDO objects have a low priority (high COB-Identifier).

Objects and Function Code in the Predefined Connection Set

For easier management of the Identifiers CANopen uses the "Predefined Master/Slave Connection Set", where all identifiers are defined with standard values in the object dictionary. These identifiers can however be changed and customized via SDO access.



The 11-bit Identifier is made up of a **4-bit function code** and a **7-bit node-ID** number.



The higher the value of the COB-Identifier, the lower is its priority!

Broadcast (network-wide) Objects

object	function code (binary)	resulting COB-ID	Communication Parameters at Index
NMT	0000	0	-
SYNC	0001	128 (80h)	1005h, 1006h, 1007h
TIME STAMP	0010	256 (100h)	1012h, 1013h

Peer-To Peer (device-to-device) Objects

object	function code (binary)	Resulting COB-IDs	Communication Parameters at Index
EMERGENCY	0001	129 (81h) – 255 (FFh)	1014h, 1015h
PDO1 (tx)	0011	385 (181h) – 511 (1FFh)	1800h
PDO1 (rx)	0100	513 (201h) – 639 (27Fh)	1400h
PDO2 (tx)	0101	641 (281h) – 767 (2FFh)	1801h
PDO2 (rx)	0110	769 (301h) – 895 (37Fh)	1401h
PDO3 (tx)	0111	897 (381h) – 1023 (3FFh)	1802h
PDO3 (rx)	1000	1025 (401h) – 1151 (47Fh)	1402h
PDO4 (tx)	1001	1153 (481h) – 1279 (4FFh)	1803h
PDO4 (rx)	1010	1281 (501h) – 1407 (57Fh)	1403h
SDO (tx)	1011	1409 (581h) – 1535 (5FFh)	1200h
SDO (rx)	1100	1537 (601h) – 1663 (67Fh)	1200h
NMT Error Control	1110	1793 (701h) – 1919 (77Fh)	1016h, 1017h

Restricted, reserved Objects

COB-ID	used by object
0 (000h)	NMT
1 (001h)	reserved
257 (101h) – 384 (180h)	reserved
1409 (581h) – 1535 (5FFh)	default SDO (tx)
1537 (601h) – 1663 (67Fh)	default SDO (rx)
1760 (6E0h)	reserved
1793 (701h) – 1919 (77Fh)	NMT Error Control
2020 (780h) – 2047 (7FFh)	reserved

2 Transmission of Process Data

Within the CANopen encoder **three PDO services PDO1 (tx) , PDO2 (tx) and PDO3(tx)** are available. A PDO transmission can be triggered by a variety of events (see Object Dictionary Index 1800h):

- **asynchronously** (event driven) by an internal cyclic device timer or by a change in the process value of the sensor data
- **synchronously** as a response to a SYNC telegram; (a SYNC command will cause all CANopen nodes to store their values synchronously, after which they are transferred in succession to the bus according to their set priority)
- **as a response** to an RTR-Telegram (per Remote Frame=recessive RTR-bit, exactly that message with the communicated ID will be requested)

Example : Default Mapping of the **PDO messages** have the following structure

Mapping	TPDO1 1800h	TPDO2 1801h	TPDO3 1802h
Mapping object	1A00h	1A01h	1A02h
Content	0x60040020	0x60300110	0x60040020
Object	6004h	6030h	6004h
Subindex	00	01	00
Length	20h(32 Bit)	10h(16 Bit)	20h(32 Bit)
	Asynchronous	Asynchronous	Synchronous

Transmit PDO 1 (1800h) Position asynchronous

Default COB-ID is 180 + Node number: e.g. 180h + 3Fh = 1BFh

Message	Byte 0	Byte 1	Byte2	Byte 3
1BF	Position LSB	Position MSB	00	00

Position values are in a range of 0 – 3FFFh oder 0 – 16383 (decimal).

Transmit PDO2 (1801h) Velocity asynchronous

Default COB-ID ist 280 + Node number: e.g. 280h + 3Fh = 2BFh

Message	Byte 0	Byte 1
2BF	Velocity LSB	Velocity MSB

The value of the velocity is signed and is shifted in a range of (plus) 0 – 1A00h resp (minus) 0 – E600h

Transmit PDO3 (1802h) Position synchronous

Default COB-ID ist 380 + Node number: e.g. 380h + 3Fh = 3BFh

Message	Byte 0	Byte 1	Byte2	Byte 3
3BF	Position LSB	Position MSB	00	00

Position values are in a range of 0 – 3FFFh oder 0 – 16383 (decimal).

3 Transmission of Service Data

SDO-COB-ID

The following identifiers are available as standard for the SDO services:

SDO (tx) (Encoder→Master): 580h (1408) + node number

SDO (rx) (Master→Encoder): 600h (1536) + node number

The SDO identifiers cannot be changed!

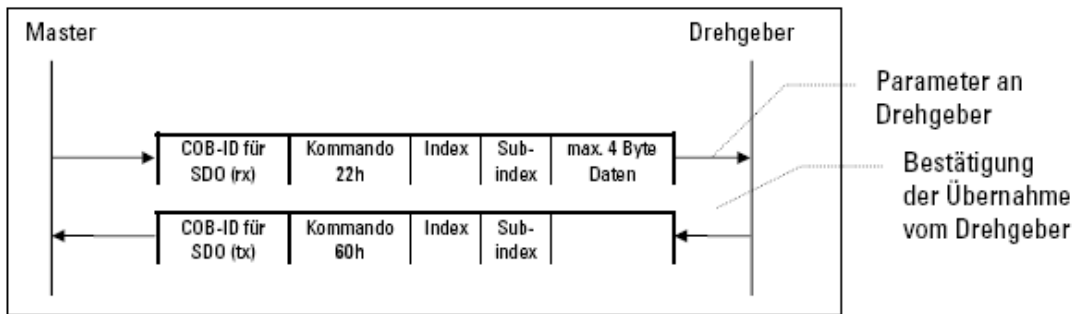
Kommando (Expedited Protocol)	Art	Funktion
22h	SDO(rx), Initiate Download Request	Parameter an Drehgeber senden (Datenlänge max. 4 Byte)
23h	SDO(rx), Initiate Download Request	Parameter an Drehgeber senden (Datenlänge = 4 Byte)
2Bh	SDO(rx), Initiate Download Request	Parameter an Drehgeber senden (Datenlänge = 2 Byte)
2Fh	SDO(rx), Initiate Download Request	Parameter an Drehgeber senden (Datenlänge = 1 Byte)
60h	SDO(tx), Initiate Download Response	Bestätigung der Übernahme an Master
40h	SDO(rx), Initiate Upload Request	Parameter vom Drehgeber anfordern
43h	SDO(tx), Initiate Upload Response	Parameter an Master mit Datenlänge=4 Byte (Unsigned 32)
4Bh	SDO(tx), Initiate Upload Response	Parameter an Master mit Datenlänge=2 Byte (Unsigned 16)
4Fh	SDO(tx), Initiate Upload Response	Parameter an Master mit Datenlänge=1 Byte (Unsigned 8)
80h	SDO(tx), Abort Domain Transfer	Drehgeber meldet Fehlercode an Master



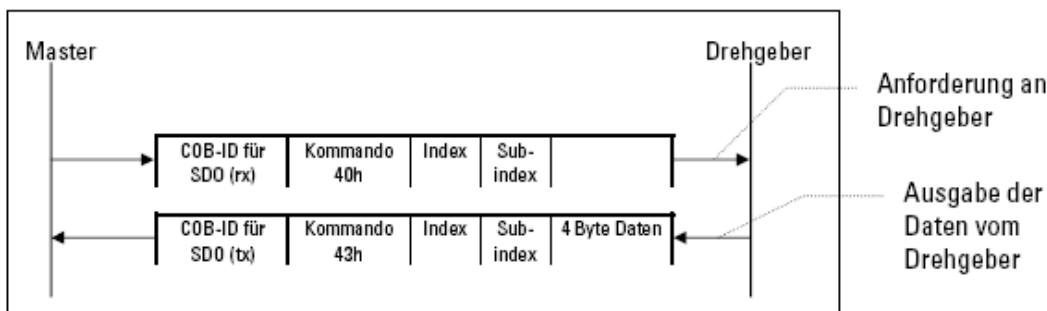
If an error occurs, then an error message (command 80h) will replace the normal confirmation (Response). The error message covers not only the communication protocol error but also the object dictionary access error (e.g. wrong index, attempted write to Read-Only Object, incorrect data length etc).

The error codes are described in the CANopen Profile (DS 301) or in the Device Profile (DSP 406).

Example: Transmission of Service Data to and from the encoder



Master überträgt Parameter an Drehgeber



Master fordert Parameter vom Drehgeber an

LSS Hardware Restrictions (LSS Address)

All LSS Slaves must support valid Object Dictionary entries for Identity object [1018h] which has 32 bits for each part of the LSS Address:

- Vendor-ID (numerical number)
- Product-Code (numerical number)
- Revision-Number (major and minor revision as numerical number)
- Serial-Number (numerical number)
- **LSS-Master CAN-ID 2021**
- **LSS-Slave CAN-ID 2020**

A Product-Code, Revision-Number and a Serial-Number are assigned by the device supplier. The LSS address which must be absolutely unique. No other LSS slave may have the same LSS address.

LSS Operating Restrictions

To function properly the following restrictions apply:

- All devices on a CANopen network must support LSS.
- There can be only one LSS Master.
- All nodes are required to start-up with the same initial baud rate.
- LSS communication can take place during any NMT state such as "stopped" or "pre-operational".

LSS Configuration and the Operation Modes

Configuration Mode

- When an LSS Slave is in this mode, it actively listens for and processes configuration commands from the LSS Master.
- Some configuration commands configure only one LSS Slave at the time (for example, to change of CANopen node ID)
- Some configuration commands configure multiple or all LSS Slave nodes (for example, to change the baud rate)

Operation Mode

- A LSS Slave in this mode ignores the configuration commands from the LSS

Technical Manual

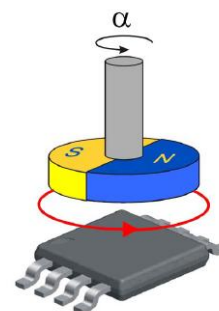
Absolute Singleturn Encoder **CANopen** series M36X8



4 Power supply and CAN-bus connection

Power supply

Sensor:	Magnetic Hall -Sensor
	14 Bit Resolution /9 Bit Accuracy
Power supply:	10 ... 30 VDC
Current consumption:	typ. 22mA at 24 VDC
	max. 49 mA at 10 VDC
Reverse polarity protection:	yes
CAN Transceiver:	82C251 / short circuit tested
Galvanic Isolation:	no
CANopen Kommunikation	DS 301 V4.02
CANopen Encoder Device Profile	DS406 V 3.1



Can-Cable terminal

Short name	Description	Cable color
CG	CAN Ground	gray
CL	CAN_Low (-)	yellow
CH	CAN_High (+)	green
0V	0Volt power	white
+V	+UB power	brown

CAN - M12 Connector



Short name	Description	PIN Nr.	Color
CG	CAN Ground	1	GY
CL	CAN_Low (-)	5	YE
CH	CAN_High (+)	4	GN
0V	0 Volt power	3	WH
+V	+UB power	2	BN

Terminal M12-Connector

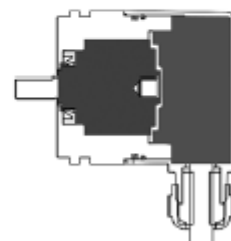
Mechanical characteristics

Mechanical characteristics:

Max. speed:	6000 min ⁻¹
Starting torque	< 0,06 Nm
Weight:	approx. 0,2 kg
Protection acc. to EN 60 529:	IP 67 (IP 69k on request)
Working temperature:	-40 °C ... +85 °C
Materials:	Shaft: stainless steel, Flange: aluminium, Housing: die cast zinc, Cable: PUR
Shock resistance acc. to DIN-IEC 68-2-27:	5000 m/s ² , 6 ms
Vibration resistance acc. to DIN-IEC 68-2-6:	300 m/s ² , 10 ... 2000 Hz
Permanent shock resistance acc. to DIN-IEC 68-2-29	1000 m/s ² , 2 ms
Vibration (broad-band random) to DIN-IEC 68-2-64	5 ... 2500 Hz, 100 m/s ² - rms

All-round protection thanks to Safety-Lockplus™ and Sensor-Protect™ technology

Safety-Lockplus™:
IP69k protection on the flange side, robust bearing assemblies with interlocking bearings, mechanically protected shaft seal



Sensor-Protect™:
Fully encapsulated electronics, separate mechanical bearing assembly

It is possible to change the baud rate using the **Software** at **Object 2100h** or one of the appropriate **LSS-Services**. The following baud rates are available to the user:

Factory default: 250 kBit/s (value 5)

Value	Baudrate in KBit/s
0	10
1	20
2	50
3	100
4	125
5	250
6	500
8	1000

Please note the following when selecting a baud rate

The chosen cycle time (see Object 1800h, Sub-index 5 Event Timer) must be longer than the bus transfer time, to ensure that the PDOs are communicated error-free!

With a baud rate of 10 KBaud: cycle time must be at least 14 ms
With a baud rate of 20 KBaud: cycle time must be at least 10 ms
With a baud rate of 50 KBaud: cycle time must be at least 4 ms



With a cycle time=0 in Event-Mode (i.e. PDO on value change) the baud rate must be at least **125 KBaud**.

Node number

It is possible to change the node number using the **Software** at **Object 2101h** or one of the appropriate **LSS-Services**.

Factory default : 0x3F (63 decimal).

Node number 0 is reserved and must not be used by any node. The resulting node numbers lie in the range **1...7Fh** hexadecimal (1...127 decimal).



The acceptance of a new node number only becomes effective when the encoder is rebooted (Reset/Power-on) or by means of an **NMT Reset Node** command.

All previous adjustments made to the object directory are lost and are setting to the factory default.

CAN-bus Termination

The **CAN-Bus Termination** can be changed in **Object 2102h**.

Factory default: 0x01 (Bus termination active)

Once the CAN bus has been looped through, it must be terminated between CAN+ and CAN- at both ends using 120 ohm bus termination resistors.

Save All Bus Parameters

The "save all bus parameters" can be changed in **Object 2105h**.

Factory default: 0x65766173 ("save")

This object stores all bus parameters (**Object 2100h ,2101h,2102h**) permanently in an EEPROM. Using the command "save" (save all Parameters) causes all the parameters to be stored. This process requires ca. 5ms. In order to prevent an inadvertent save, the instruction will only be executed if the string "save" is entered as a codeword into this Sub-Index.



The Parameter „**save**“ (**hexadezimal 0x65766173**) stores all bus adjustments as **Baudrate, Node number and Termination** permanently.

6 Layer Setting Services (LSS)

Exactly two conditions must be fulfilled for the interconnection of CANopen devices to a network: all devices must use the same Baudrate, and the CANopen Node-IDs must be unique. The condition for the use of the LSS is, in addition to support by the device itself, to establish a 1:1 wiring to the Node. Then the Baudrate and the Node-ID are set in dialog mode. The COB-ID **0x7E5** is used for CAN messages to the device, the device responds to COB-ID **0x7E4**. LSS messages are always a full 8 bytes long. Unused bytes are reserved and should be initialized with 0.

To make contact with a device to be configured, the "**Switch Mode Global**" command is transmitted:

0x04	0x01	reserved
-------------	-------------	-----------------

This command sets the device to LSS configuration mode. Unfortunately, this very service is the only unacknowledged LSS service, to which the device will therefore not respond, even if it has carried it out. The system integrator can therefore only find out with the following command whether the device has reacted.

Next the Node-ID is requested via the "**Inquire Node-ID**" service:

0x5E	reserved
-------------	-----------------

If successful, the device responds with:

0x5E	Node ID	reserved
-------------	----------------	-----------------

If there is no response, then either the device **does not support the LSS service** or the Baudrate is not correct. If, namely, the Baudrate when supplied is not known, the above-mentioned communication sequence must be tested with all permissible CANopen Baudrates until the device is found.

The "**Configure Node-ID**" service is used to configure the new Node-ID:

0x11	Node ID	reserved
-------------	----------------	-----------------

The error code is included in the device response:

0x11	Error code	Error extension	reserved
-------------	-------------------	------------------------	-----------------

Error code 0 means success; error code 1 means inadmissible Node-ID; the other error codes are reserved. The error extension contains vendor-specific information but is only valid for error code 0xFF.

The Baudrate is configured with the "**Configure Bit Timing Parameters**" service:

0x13	Bit timing	Table entry	reserved
-------------	-------------------	--------------------	-----------------

The standardized **CANopen Baudrates** are listed in the following table:

Baudrate table 0x00	
Table index	Baudrate
0	1000 kBit/s
1	800 kBit/s
2	500 kBit/s
3	250 kBit/s
4	125 kBit/s
5	reserved
6	50 kBit/s
7	20 kBit/s
8	10 kBit/s

Again the device response is:

0x13	Error code	Error extension	reserved
------	------------	-----------------	----------

Error code 0 means success; error code 1 means inadmissible baudrate; the other error codes are reserved. The error extension contains vendor-specific information, but is only valid for error code 0xFF.

Now that the node-ID and the baudrate are configured, these settings should be saved with the **"Store Configuration"** service:

0x17	reserved
------	----------

Whereupon the device acknowledges:

0x17	Error code	Error extension	reserved
------	------------	-----------------	----------

Error code 0 means success; error code 1 means that the device does not support saving; error code 2 means that there is a problem with access to the storage medium; the other error codes are reserved. The error extension contains vendor-specific information, but is only valid for error code 0xFF.

Finally, the device is switched back from configuration mode to normal mode via **"Switch Mode Global"**:

0x04	0x00	reserved
------	------	----------

After being switched **physically off and on again**, the device now works with the new settings.

7 Default settings on delivery



On delivery the following software parameters have been factory set.

Description	Setting	Software
Baud rate	250 Kbit/s	Object 2100h = 05h
Node address	63	Object 2101h = 3Fh
Termination	ON	Object 2102h = 01h

Index (hex)	Name	Standard value
	Communication parameter	
1005h	COB-ID Sync	80h
100Ch	Guard Time	0
100Dh	Life Time Factor	0
1012h	COB-ID Time stamp	100h
1013h	High Resolution time stamp	0
1017h	Producer heartbeat time	0
1029h	Error Behaviour	0 = Comm Error 1 = Device specific 1 = Manufacturer Err.
1800h	TPDO1 Communication Parameter	
01h	COB-ID	180h + Node number
02h	Transmission Type	255 (asynch)
03h	Inhibit Time	0 [steps in 100µs]
05h	Event timer	0 [steps in ms]
1801h	TPDO2 Communication Parameter	
01h	COB-ID	280h + Node number
02h	Transmission Type	255 (asynch)
03h	Inhibit Time	0 [steps in 100µs]
05h	Event timer	0 [steps in ms]
1802h	TPDO3 Communication Parameter	
01h	COB-ID	380h + Node number
02h	Transmission Type	1 (synch)
03h	Inhibit Time	0 [steps in 100µs]
05h	Event timer	0 [steps in ms]
1A00h	TPDO1 Mapping	
01h	1.Mapped Object	0x60040020
1A01h	TPDO2 Mapping	
01h	1.Mapped Object	0x60300110
1A02h	TPDO3 Mapping	
01h	1.Mapped Object	0x60040020

Index (hex)	Name	Standard value
	Encoder Profile	
6000h	Operating Parameter	Scaling off
6001h	Measuring Units per Revolution	16384 (14 Bit)
6002h	Total Measuring Range	16384 (14 Bit)
6003h	Preset value	0
6200h	Cyclic Timer (see TPDO1 Comm.Par)	0
6401h	Work area low limit	0
6402h	Work area high limit	16383
2100h	Baud rate	05h
2101h	Node number	3Fh
2102h	CANbus termination	1 (active)
2105h	Save All Bus Parameters	0x65766173



The original Standard Values (default values on delivery) can be reloaded again by means of Object **1011h** (restore parameters).

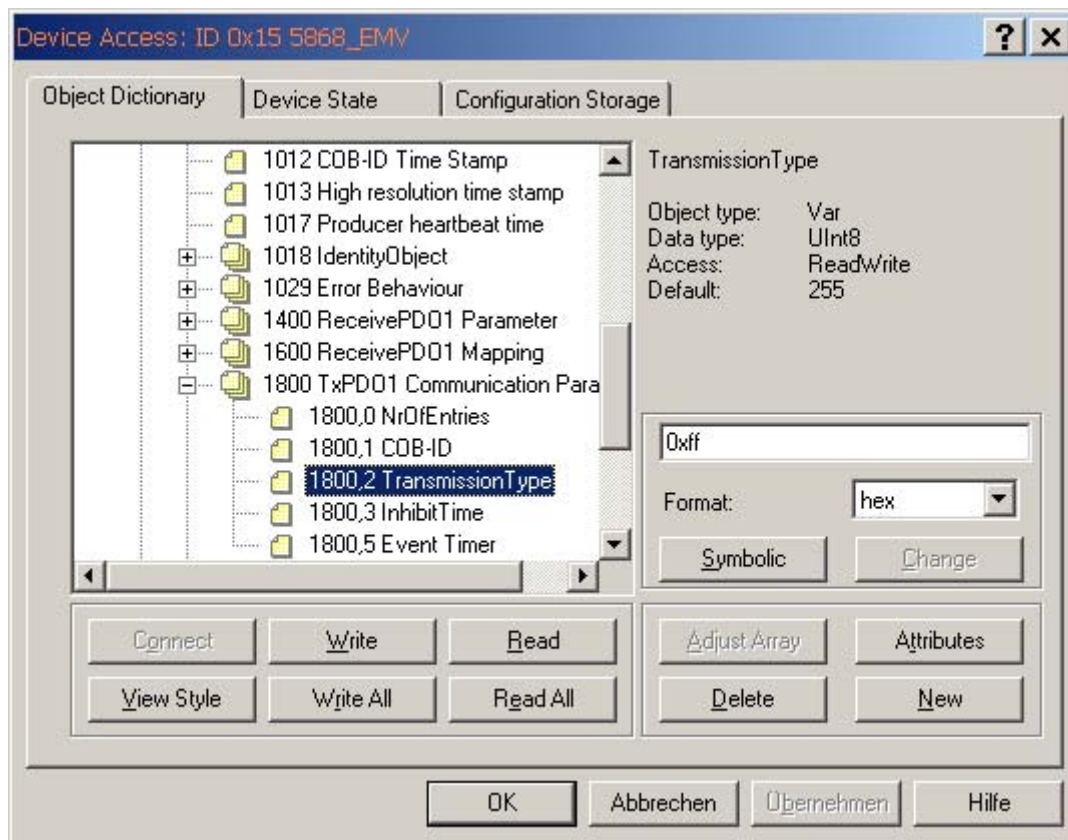
In order to ensure that parameter changes are saved in the event of power failure, then these must without fail be transferred to the EEPROM by means of Object **1010h** (store parameters). This will cause all data already present in the EPROM to be over-written!



If errors have occurred during programming of the objects and if these parameters are then saved in the EEPROM, it will not be possible to address the encoder next time it is switched on (the encoder will send only **Emergency** messages). This error can be cleared only by means of a general **Reset** of the encoder.

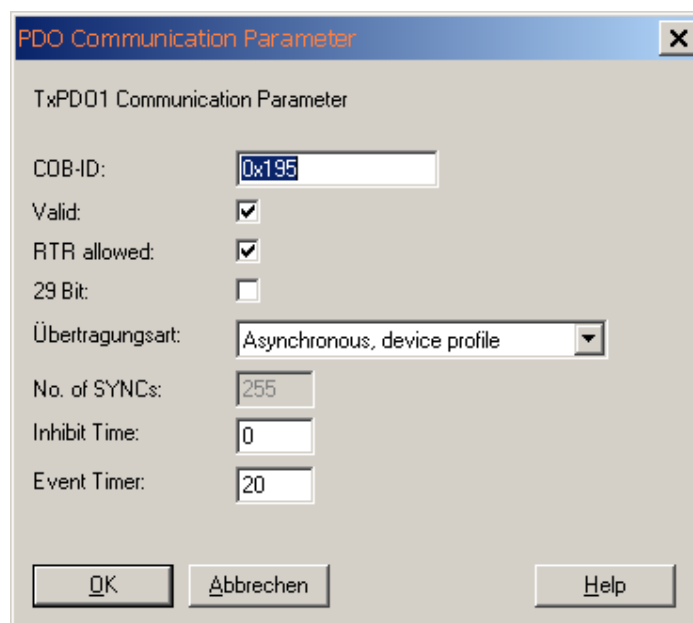
8 Communication Parameters

The COB-ID and the Transmission Type for **PDO1** are defined in the Object Dictionary Index **1800h** .

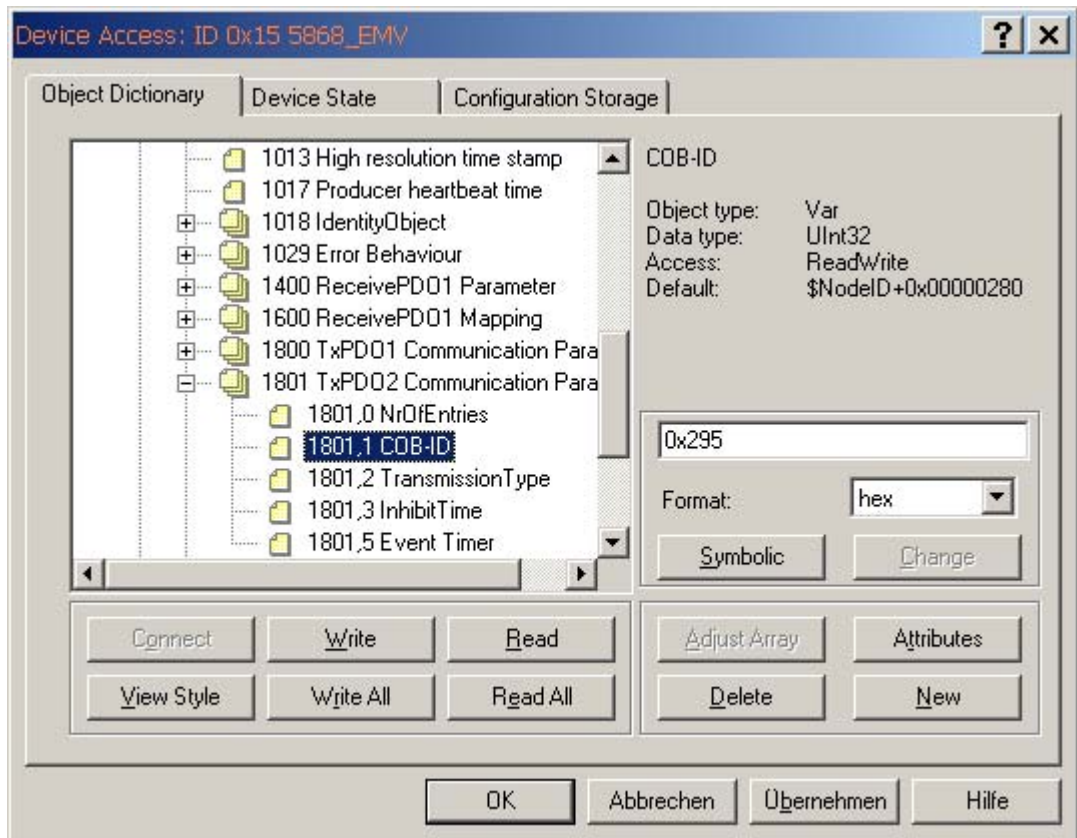


Default-settings:

Enabling: **PDO enabled** **RTR allowed**
 COB-ID: **180h + node number set (here 11h)**
 Transmission type: **255 = asynchronous acc. to device profile**
 Event Timer: **20 ms**

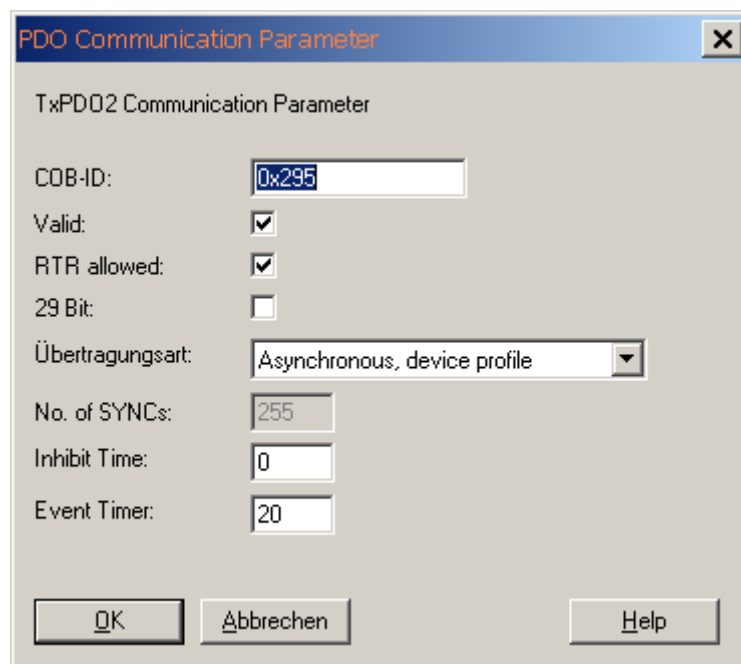


The COB-ID and the Transmission Type for **PDO2** are defined in the Object Dictionary Index **1800h** .



Defaults:

Enabling: **PDO enabled** **RTR allowed**
COB-ID: **280h + node number set (here 11h)**
Transmission type: **255 = asynchronous acc.to device profile**
Event Timer: **20 ms**



Definition of the Transmission type of the PDO

transmission type	PDO transmission				
	cyclic	acyclic	synchronous	asynchronous	RTR only
0		X	X		
1-240	X		X		
241-251	- reserved -				
252			X		X
253				X	X
254				X	
255				X	

A value between 1 ...240 means that the PDO will be sent **synchronously and cyclically**. The number of the Transmission Type signifies the **quantity of SYNC** pulses that are necessary to forward the PDOs. The Transmission Types 252 and 253 state that the PDO will only be sent when requested via an RTR.



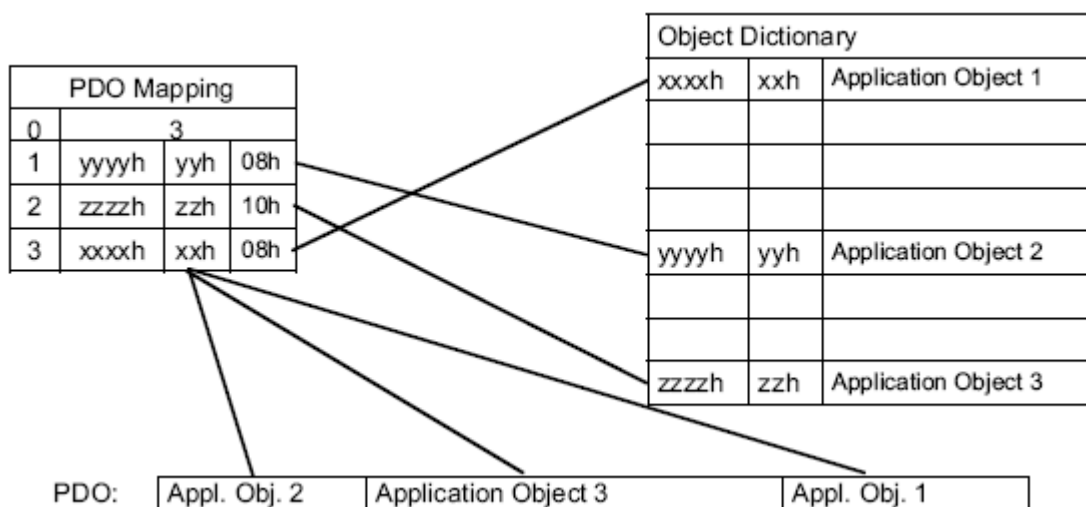
Type 254 means that the event will be triggered depending on the application (application-specific), whereas Type 255 is dependent on the device (device-specific). Additionally for Numbers 254/255 a time-controlled **EventTimer** can be used. The values for the timer can range from **1ms ... 65535 ms**.

Variable PDO Mapping

Variable Mapping of the various objects means that the user is able to configure the content of the Transmit PDOs dependent on the application.

Example of an entry in the Mapping Table:

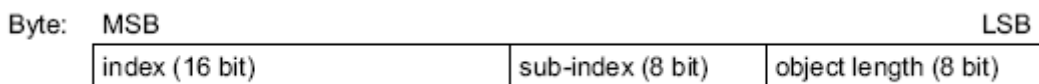
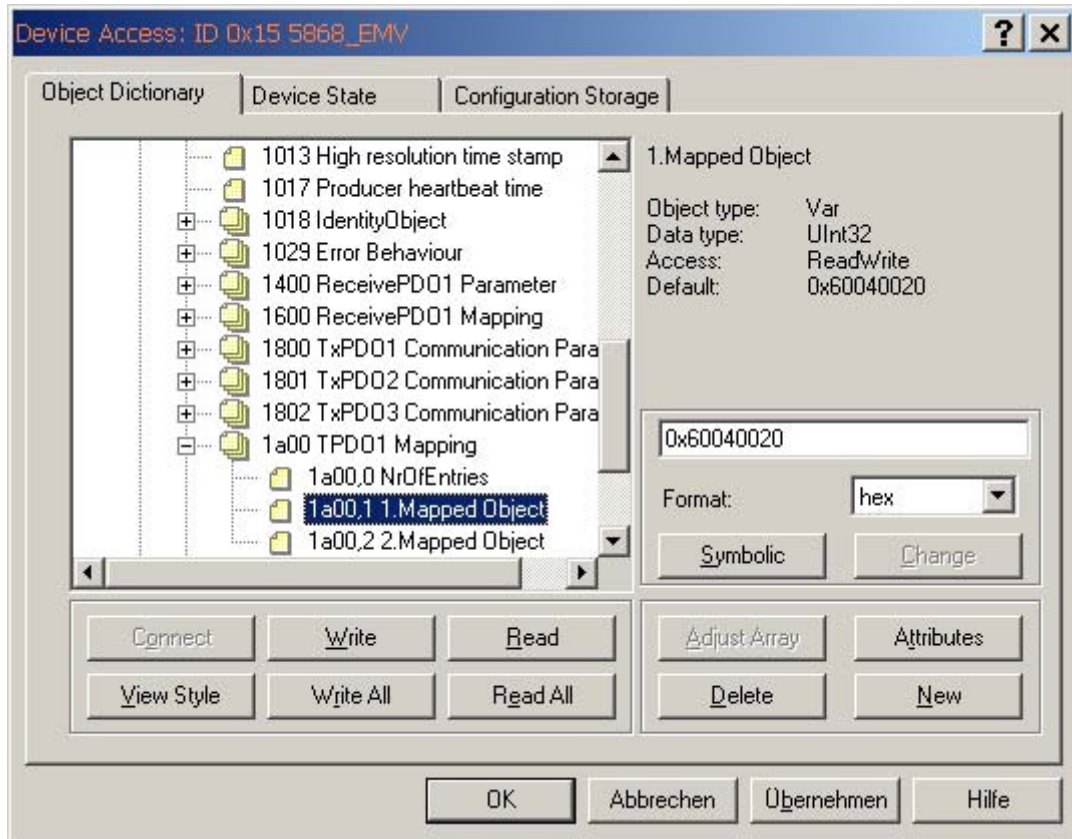
The mapped PDO consists of 3 Application Object entries of varying lengths:



Application Object 2 occupies Byte 1 (08h) in the Transmit PDO. Thereafter follows Application Object 3 with a length of 16 bit (10h = 2 bytes) and finally Application Object 1 with a length of 1 byte. In total, 32 bits are occupied in this PDO.

Structure of a Mapping entry

The Mapping Object for **PDO 1** is defined in the Object Dictionary Index 1A00h. It consists of 2 entries and can be modified by the user (variable mapping).



The default setting for the **Mapping of the Transmit PDO:**

Mapping	TPDO1	TPDO2	TPDO3
1.Mapping	0x60040020	0x60300110	0x60040020
Object	6004h	6030h	6004h
Subindex	00	01	00
Data length	20h(32 Bit)	10h(16 Bit)	20h(32 Bit)
	Asynchron	Asynchron	Synchron

The CANopen encoder supports **variable mapping** on all 3 Transmit PDOs.

9 Example of a variable Mapping table

Mapping Object 1A00h

The Mapping Object **1A00h** describes the **1.Transmit PDO**. It is possible to map as much as objects, until the maximum data length of **8 Bytes** is reached.

The Objects **1A01h** of **Transmit PDO2** and **1A02h** of **Transmit PDO3** following the input of **PDO1**.

Mapping table of Object 1A00h:

Mapping	TPDO1 Mapping	TPDO1 Mapping	TPDO1 Mapping
Subindex	00	01	02
Content	Nr.of Entries	1.Mapped Object	2.Mapped Object
Object	2	6004h	6030h
Subindex		00	01
Length	Byte	20h(32 Bit)	10h(16 Bit)
		Asynchronous	Asynchronous

Encoder Profile - Device specific Objects

Gerätespezifische Objekte					
INDEX (hex)	Object Symb.	ATTRIB	Name	M/O C2	TYPE
6000	VAR	RW	Operating parameters	M	unsigned16
6001	VAR	RW	Measuring Units p.Revolution (MUR)	M	unsigned32
6002	VAR	RW	Total Measuring Range (TMR)	M	unsigned32
6003	VAR	RW	Preset value	M	unsigned32
6004	VAR	RO	Position value	M MAP	unsigned32
6030	ARRAY	RO	Speed Value	O MAP	signed16
6040	ARRAY	RO	Acceleration Value	O	Signed16
6200	VAR	RW	Cyclic Timer	M	unsigned16
6400	ARRAY	RO	Working Area state	O MAP	unsigned 8
6401	ARRAY	RW	Working Area Low Limit	O	Unsigned32
6402	ARRAY	RW	Working Area High Limit	O	Unsigned32
6500	VAR	RO	Operating Status	M	unsigned16
6501	VAR	RO	Measuring Step (Singleturn)	M	unsigned32
6502	VAR	RO	Number of revolutions	M	unsigned16
6503	VAR	RO	Alarms	M MAP	unsigned16
6504	VAR	RO	Supported alarms	M	unsigned16
6505	VAR	RO	Warnings	M MAP	unsigned16

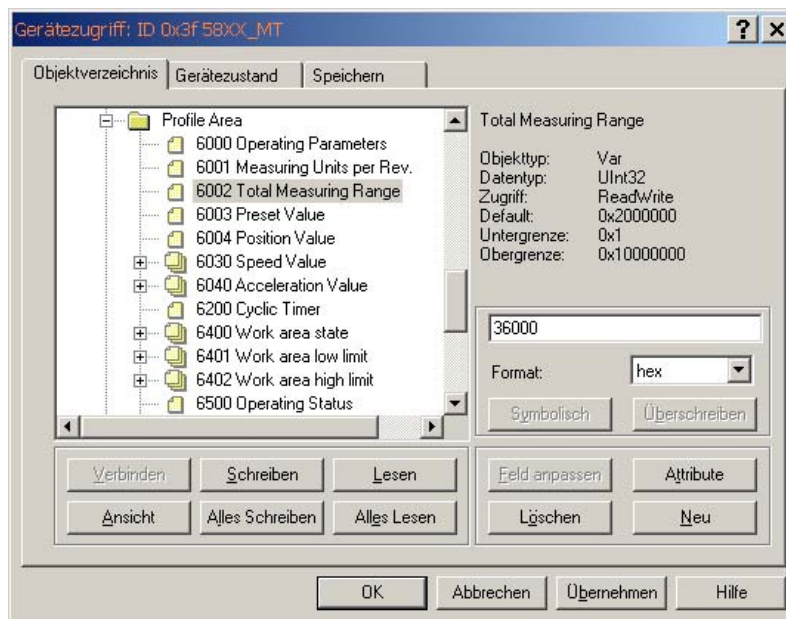
In this example two Objects are mapped to the 1.Transmit PDO at 1800h

10 Application Programming Example:

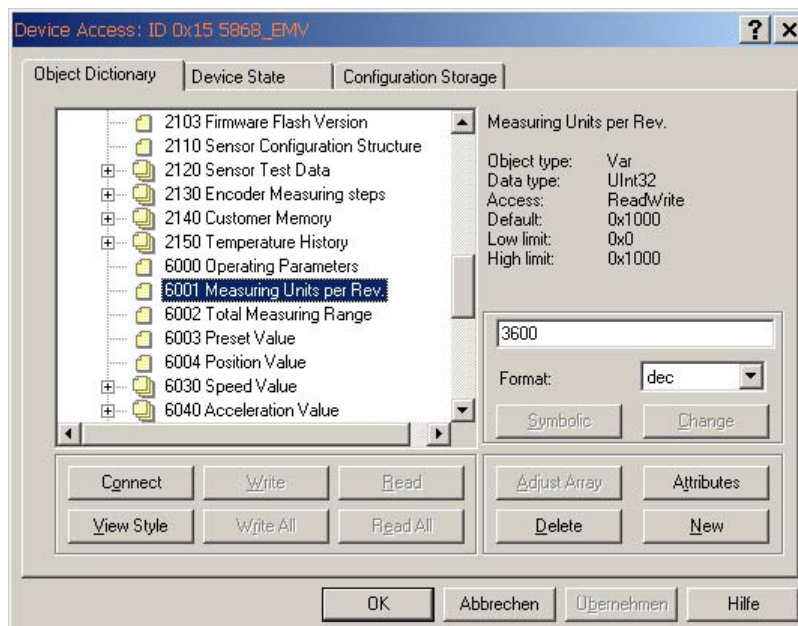
Setting up Objects

- **Total Measuring Range to 3600**
- **Measuring Units per Revolution** should be set to 3600 steps per revolution
- Position Value should be set to 0
- TPDO1 (Position) should transmit the event every 10 ms
- TPDO2 (Speed) should transmit the event every 20 ms
- **Producer Heartbeat** should be reduced to 500 ms
- Work area limits are 1000 and 3500
- The new parameters should be saved in the **EEPROM**

Total Measuring Range set to 3600



Measuring Units per Revolution – limit to 3600

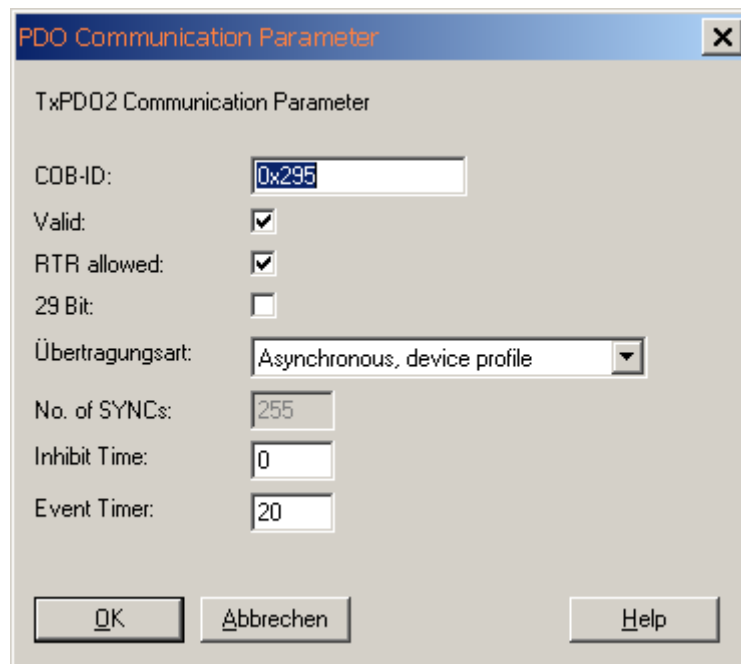


Set Preset Value to 0

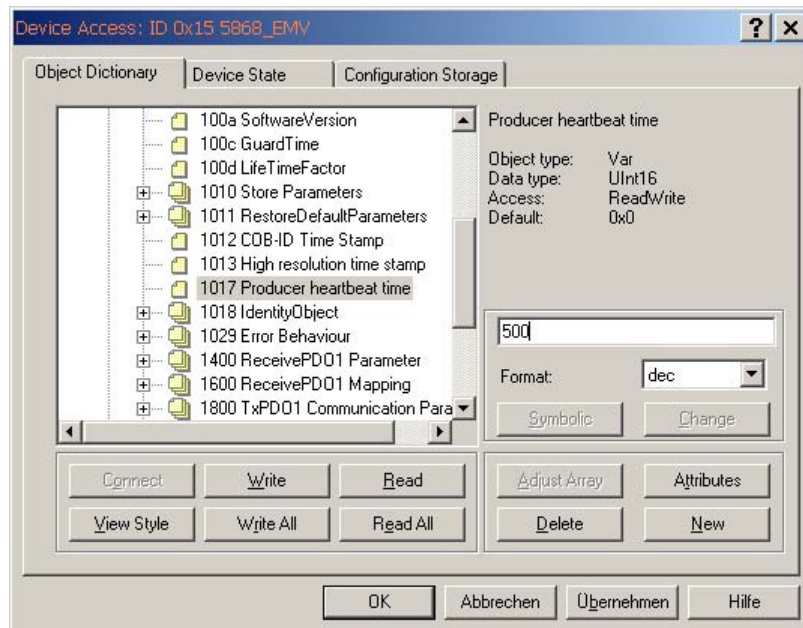


Set the values of Transmit Parameters TPDO1 and TPDO2

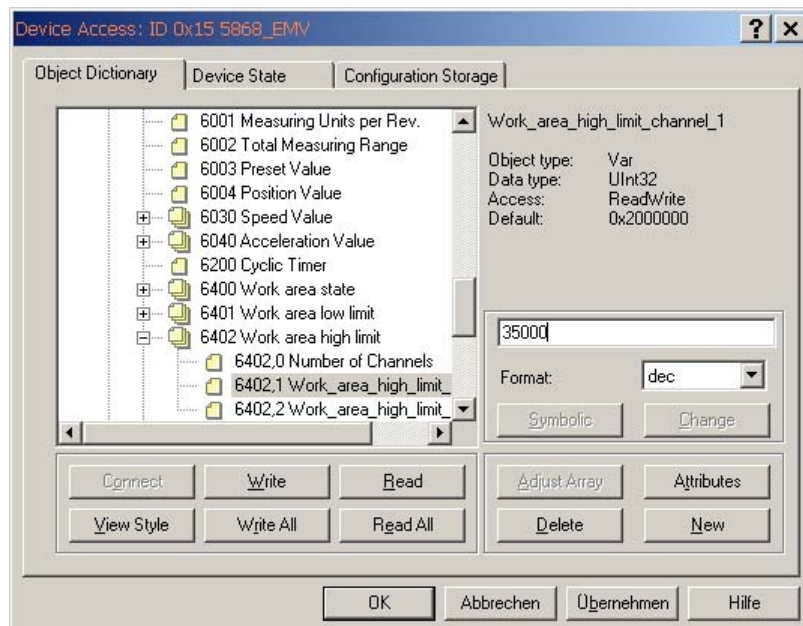
Type 254 means that the event will be triggered depending on the application, whereas Number 255 is **dependent on the device**. Additionally for Numbers 254/255 a time-controlled **EventTimer** can be used. The values for the timer can range from **1ms ... 65535 ms**.



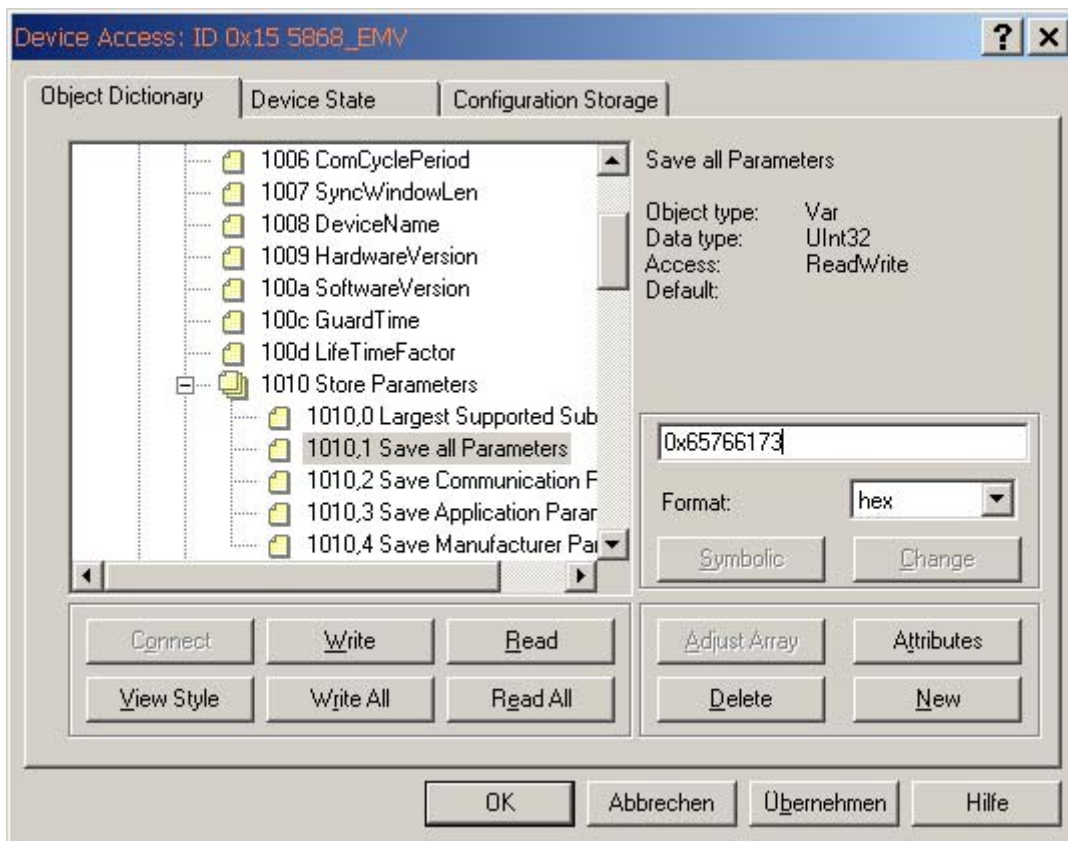
Producer Heartbeat - set to 500 ms



Set Work Area low and high limit values



Save all modified parameters in the EEPROM Store Parameters 1010h



Object 1010h Store Parameters

Using the command "save" under Sub-Index 1h (save all Parameters) causes all the parameters to be stored in the non-volatile memory (EEPROM). All Communication Objects, Application Objects and Manufacturer-specific Objects are saved under this Sub-Index. This process requires ca. 14 ms. In order to prevent an inadvertent save, the instruction will only be executed if the string "save" is entered as a codeword into this Sub-Index. A read access to the Sub-Index 1h provides information about the functionality of the memory.

Term	Content	Notes
Byte 3	73h	(ASCII Code für "s")
Byte 2	61h	(ASCII Code für "a")
Byte 1	76h	(ASCII Code für "v")
Byte 0	65h	(ASCII Code für "e")

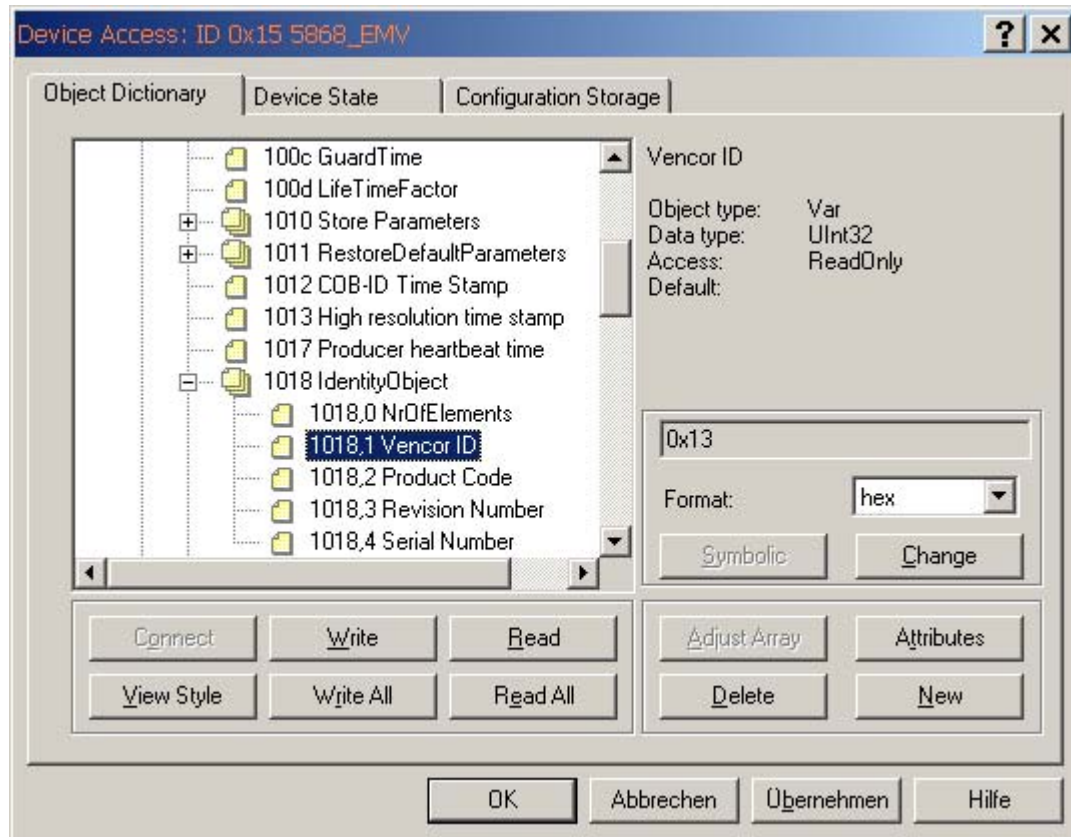
Object 1011h: Load Standard Values

Using the command "load" under Sub-Index 1h causes all parameters to be reset to their standard values. In order to prevent inadvertent loading of the standard values, the instruction will only be executed if the string "load" is entered as a codeword into this Sub-Index.

Term	Content	Notes
Byte 3	6Ch	(ASCII Code für "l")
Byte 2	6Fh	(ASCII Code für "o")
Byte 1	61h	(ASCII Code für "a")
Byte 0	64h	(ASCII Code für "d")

Communication Profile – further objects Object 1018h: Identity Object

Information concerning the vendor and the device:



1018 RECORD Device – Identification read only

Sub-Index 0h : "Number of Sub-indices"
supplies the value 4

Sub-Index 1h: "read" only

supplies the Vendor-ID (00000013h) Fritz Kübler GmbH

Sub-Index 2h: supplies the Product Code
(e.g. 0x36582001 CANopen encoder)

Sub-Index 3h: "read" only

supplies the Software revision Number
(e.g. 102)

Sub-Index 4h: "read" only

supplies the 8-digit Serial Number of the encoder

11 Emergency Objects

Emergency Objects arise with error situations within a CAN network and are triggered depending on the event and transmitted over the bus with a **high priority**.

Important: an Emergency Object is only triggered once per “Event”. No new object is generated while the error still exists. Once the error is eliminated, then a new Emergency Object with the content 0 (Error Reset or No Error) is generated and transmitted over the bus.

Error Codes supported

The Error Codes are highlighted in **red**

Error Code (hex)	Meaning
00xx	Error Reset or No Error
10xx	Generic Error
20xx	Current
21xx	Current, device input side
22xx	Current inside the device
23xx	Current, device output side
30xx	Voltage
31xx	Mains Voltage
32xx	Voltage inside the device
33xx	Output Voltage
40xx	Temperature
41xx	Ambient Temperature
42xx	Device Temperature
50xx	Device Hardware
60xx	Device Software
61xx	Internal Software
62xx	User Software
63xx	Data Set
70xx	Additional Modules
80xx	Monitoring
81xx	Communication
8110	CAN Overrun (Objects lost)
8120	CAN in Error Passive Mode
8130	Life Guard Error or Heartbeat Error
8140	recovered from bus off
8150	Transmit COB-ID collision
82xx	Protocol Error
8210	PDO not processed due to length error
8220	PDO length exceeded
90xx	External Error
F0xx	Additional Functions
FFxx	Device specific

12 Emergency Message

Byte	0	1	2	3	4	5	6	7
Content	Emergency Error Code (see Table 21)		Error register (Object 1001H)	Manufacturer specific Error Field				

Figure 34: Emergency Object Data

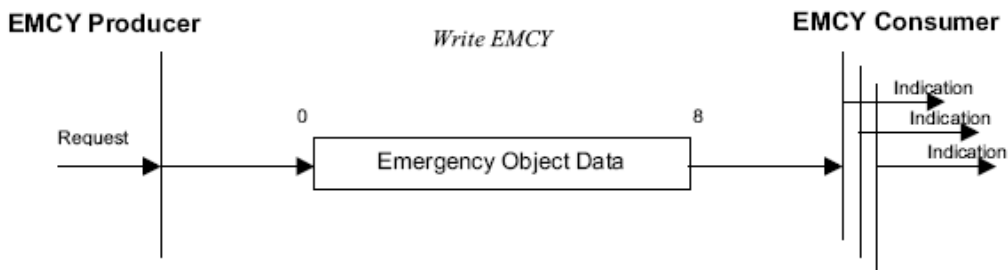
Example of an over-temperature message:

Transfer Data	00	42	09	80	56	20	50	2E
----------------------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------

[Errcode]	4200	Device Overtemperature
[Error Register]	09	Error Register
[ManufacturerSpecific1]	80	Diagnosebytes
[ManufacturerSpecific2]	56	Diagnosebytes
[ManufacturerSpecific3]	20	Diagnosebytes
[ManufacturerSpecific4]	50	Diagnosebytes
[ManufacturerSpecific5]	2E	Diagnosebytes

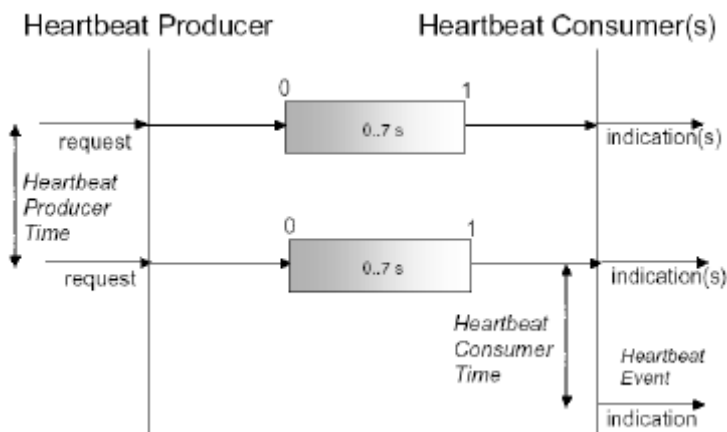
Emergency Protocol

An "unconfirmed" Service message is defined



The behaviour in the case of an error is described in **Object 1029h Error Behaviour**

13 Heartbeat Protocol



Nowadays as an alternative to **Node Guarding** the modern **Heartbeat Protocol** should be used. The protocol is activated if a value > 0 is written to **Object 1017h** Producer Heartbeat Time.

A "Heartbeat-Producer" cyclically transmits this Heartbeat message. One or more "Heartbeat-Consumer(s)" can receive this Heartbeat message.

If the cyclic transmission of this Heartbeat message is missing, then a "Heartbeat Event" is generated. The behaviour in the case of an error is defined in Object 1029h Subindex 1 "Communication Error".

14 CANopen Object Dictionary

Die Beschreibung der Objektverzeichnis-Einträge ist folgendermaßen aufgebaut:

Index (hex)	Sub-Index (hex)	Objekt	Name	Typ	Attr.	M/O
-------------	-----------------	--------	------	-----	-------	-----

Index:	16 Bit-Adresse des Eintrages					
Sub-Index:	8 Bit-Zeiger auf Untereintrag; wird nur bei komplexen Datenstrukturen (z.B. Record, Array) verwendet; wenn kein Untereintrag vorhanden: Sub-Index=0					
Objekt:	NULL	Eintrag ohne Daten				
	DOMAIN	größere variable Datenmenge, z.B. Programmcode				
	DEFTYPE	Definition der Datentypen, z.B. boolean, float, unsigned16 usw.				
	DEFSTRUCT	Definition eines Record-Eintrages, z.B. PDO Mapping Struktur				
	VAR	einzelner Datenwert, z.B. boolean, float, unsigned16, string usw.				
	ARRAY	Feld mit gleichartigen Daten, z.B. unsigned16 Daten				
	RECORD	Feld mit beliebig gemischten Datentypen				
Name:	kurze Beschreibung der Funktion					
Typ:	Datentyp, z.B. boolean, float, unsigned16, integer usw.					
Attr.:	Attribut gibt Zugriffsrechte auf das Objekt an:					
	rw	Schreib- und Lesezugriff				
	ro	nur Lesezugriff				
	const	nur Lesezugriff, Wert ist eine Konstante				
M/O	M	Mandatory: Objekt muss im Gerät implementiert sein				
	O	Optional: Objekt muss nicht im Gerät implementiert sein				

Structure of the entire Object Dictionary:

Index (hex)	Object
0000	unused
0001 - 001F	static data types
0020 - 003F	complex data types
0040 - 005F	manufacturer-specific data types
0060 - 0FFF	reserved
1000 - 1FFF	Communication Profile
2000 - 5FFF	Manufacturer-specific Profile
6000 - 9FFF	Standardized Device Profile
A000 - FFFF	reserved

15 CANopen Communication Profile DS 301 V4.02

Communication Objects

INDEX (hex)	OBJECT SYMBOL	ATTRIB	Name	M/O	TYPE
1000	VAR	CONST	Device Type	M	Unsigned32
1001	VAR	RO	Error Register	M	Unsigned8
1002	VAR	RO	Manufacturer Status	O	Unsigned32
1003	RECORD	RO	Predefined Error Field	O	Unsigned32
1004	ARRAY	RO	Number of PDO supported	O	Unsigned32
1005	VAR	RW	COB-ID Sync message	O	Unsigned32
1006	VAR	RW	Communication cycle period	O	Unsigned32
1007	VAR	RW	synchr.window length	O	Unsigned32
1008	VAR	CONST	Manufacturer Device Name	O	visible string
1009	VAR	CONST	Manufacturer Hardware Version	O	visible string
100A	VAR	CONST	Manufacturer Software Version	O	visible string
100B	VAR	RO	Node-ID	O	Unsigned32
100C	VAR	RW	Guard Time	O	Unsigned32
100D	VAR	RW	LifeTime Factor	O	Unsigned32
1010	VAR	RW	Store parameters (Device Profile)	O	Unsigned32
1011	VAR	RW	Restore parameters (Device Profile)	O	Unsigned32
1014	VAR	RO	COB_ID Emcy	O	Unsigned32
1015	VAR	RW	Inhibit Time Emcy	O	Unsigned32
1017	VAR	RW	Producer Heartbeat time	O	Unsigned16
1018	RECORD	RO	Identity Object	M	PDOComPar
1029	ARRAY	RW	Error Behaviour	O	Unsigned8
1800	RECORD		1 st transmit PDO Comm. Par.	O	PDOComPar
1801	RECORD		2 nd transmit PDO Comm. Par.	O	PDOComPar
1802	RECORD		3 rd transmit PDO Comm. Par.	O	PDOComPar
1A00	ARRAY		1 st transmit PDO Mapping Par.	O	PDOMapping
1A01	ARRAY		2 nd transmit PDO Mapping Par.	O	PDOMapping
1A02	ARRAY		3 rd transmit PDO Mapping Par.	O	PDOMapping

Manufacturer specific Objects

2100	VAR	RW	Baud Rate	O	Unsigned 8
2101	VAR	RW	Node number	O	Unsigned 8
2102	VAR	RW	CAN Bus Termination	O	Unsigned 8
2103	VAR	RO	Firmware Flash Version	O	Unsigned16
2105	VAR	RW	Save All Bus Parameters	O	Unsigned32
2140	Array	RW	Customer Memory	O	Unsigned32

16 CANopen Encoder Device Profile DS 406 V3.1

Device-specific Objects

INDEX (hex)	Object Symb.	ATTRIB	Name	M/O C2	TYPE
6000	VAR	RW	Operating parameters	M	unsigned16
6001	VAR	RW	Measuring Units p.Revolution (MUR)	M	unsigned32
6002	VAR	RW	Total Measuring Range (TMR)	M	unsigned32
6003	VAR	RW	Preset value	M	unsigned32
6004	VAR	RO	Position value	M MAP	unsigned32
6030	ARRAY	RO	Speed Value	O MAP	signed16
6040	ARRAY	RO	Acceleration Value	O	Signed16
6200	VAR	RW	Cyclic Timer	M	unsigned16
6400	ARRAY	RO	Working Area state	O MAP	unsigned 8
6401	ARRAY	RW	Working Area Low Limit	O	Unsigned32
6402	ARRAY	RW	Working Area High Limit	O	Unsigned32
6500	VAR	RO	Operating Status	M	unsigned16
6501	VAR	RO	Measuring Step (Singleturn)	M	unsigned32
6502	VAR	RO	Number of revolutions	M	unsigned16
6503	VAR	RO	Alarms	M MAP	unsigned16
6504	VAR	RO	Supported alarms	M	unsigned16
6505	VAR	RO	Warnings	M MAP	unsigned16
6506	VAR	RO	Supported warnings	M	unsigned16
6507	VAR	RO	Profile and SW version	M	unsigned32
6508	VAR	RO	Operating time	M	unsigned32
6509	VAR	RO	Offset value (calculated)	M	signed32
650A	VAR	RO	Module Identification	M	signed32
650B	VAR	RO	Serial Number	M	unsigned32

VAR = Variable

ARRAY = Variable Array

RW = Read/Write

RO = Read only

const = Constants

Name = Object Name

M/O = Mandatory or Optional

MAP = Object mappable

17 Objects in detail - Encoder Profile DS 306 V3.1

Object 6000h Operating Parameters

- Bit 0: Code sequence: 0 = increasing when turning clockwise (cw)
1 = increasing when turning counter-clockwise (ccw)
Default: Bit = 0
- Bit 2: Scaling Function: 0 = disable, 1 = enable; Standard: Bit = 1 (s. Object 6001,6002)
Default: Bit = 0
- Bit14: Startup Mode: 0 = after Bootup Pre-Operational, 1 = **after Bootup Operational mode**
Default Bit = 0
- Bit15: Event Mode: 0 = Position output acc. to TPDO 1800h, 1 = output on each change of position
Default Bit = 0



Bit	Function	Bit = 0	Bit =1	C1	C2
0	Code sequence	CW	CCW	m*	m*
1	Commissioning Diagnostic Control	Disabled	Enabled	o	o
2	Enable scaling	Disabled	Enabled	o	m
3	Measuring direction	Forward	Reverse	o**	o**
4..13	Reserved for further use				
14	Startup automatic in OP-Mode	Disabled	Enabled	o	o
15	Event Mode Position	Disabled	Enabled	o	o

*m = Function must be supported
o = optional
orange = defaults

Object 6001h: measuring steps per revolution (Resolution)

This parameter configures the desired resolution per revolution. The encoder itself then internally calculates the appropriate scale factor. The calculated scaling factor MUR (by which the physical position value will be multiplied) is worked out according to the following formula:

$$\text{MUR} = \text{Measuring steps per revolution (6001h)} / \text{phys. resolution Singleturn (6501h)}$$

Data content:

Byte 0	Byte 1	Byte 2	Byte 3
$2^7 \dots 2^0$	$2^{15} \dots 2^8$	$2^{23} \dots 2^{16}$	$2^{31} \dots 2^{24}$



Range of values: 1.....maximum physical resolution (16384) 14-bit
Default setting: 16384(14-bit)



After changing the measuring step it is necessary to set the preset value also to zero /or a value.

Object 6002h: Total number of measuring steps

This parameter configures the total number **Singleturn measuring steps**. A factor will be applied to the maximum physical resolution. The factor is always < 1 . After the stated number of measuring steps, the encoder will **reset itself to zero**.

Data content:

Byte 0	Byte 1	Byte 2	Byte 3
$2^7 \dots 2^0$	$2^{15} \dots 2^8$	$2^{23} \dots 2^{16}$	$2^{31} \dots 2^{24}$



Range of values: 1....maximum physical resolution (16384) 14-bit
Default setting: 16384 (14-bit)



After changing the measuring step it is necessary to set the preset value also to zero /or a value.

Object 6003h: Preset Value

The position value of the encoder will be set to this preset value. This allows, for example, for the encoder's zero position to be compared with the machine's zero position.

Data content:

Byte 0	Byte 1	Byte 2	Byte 3
$2^7 \dots 2^0$	$2^{15} \dots 2^8$	$2^{23} \dots 2^{16}$	$2^{31} \dots 2^{24}$



Range of values: 1.... maximum physical resolution (16384) 14-bit
Default setting: 0

Object 6004h: Position Value

The encoder transmits the current position value (adjusted possibly by the scaling factor)

Data content:

Byte 0	Byte 1	Byte 2	Byte 3
$2^7 \dots 2^0$	$2^{15} \dots 2^8$	$2^{23} \dots 2^{16}$	$2^{31} \dots 2^{24}$

Range of values: 1.... maximum physical resolution (16384) 14-bit



Object is mappable

Object 6030h: Speed Value

The encoder outputs the current calculated speed (possibly with scaling factor) as a signed 16-bit value. The speed value is in **rpm** and the measuring time varied between **10ms** and **100 ms**.

Data content:

Byte 0	Byte 1
$2^7 \dots 2^0$	$2^{15} \dots 2^8$

Range of values: **0....maximum speed 6500 RPM**



With values greater than 6500 RPM a warning message will be sent and the Warning Bit "Overspeed Bit 0" in the Object Warnings 6505h will be set.



Object is mappable

Object 6200h: Cyclic Timer

Defines the cycle time, with which the current position will be output by means of PDO 1 (see Object 1800h). The timer-controlled output becomes active, as soon as a cycle time >0 is entered.



This Object is only present for reasons of compatibility with earlier profile versions. Instead of this Object, please use the Event Timer Sub index (05h) in the current Transmit PDO.

Data content:

Byte 0	Byte 1
$2^7 \dots 2^0$	$2^{15} \dots 2^8$

Range of values: **0 ... FFFFh (65535) gives a cycle time in milliseconds**
Standard value = 0h

Object 6500h: Display Operating Status

This Object displays the status of the programmed settings of Object 6000h.

Data content:

Byte 0	Byte 1
$2^7 \dots 2^0$	$2^{15} \dots 2^8$

Data content: see Object 6000h

Object 6503h: Alarms

In addition to the errors that are signalled via emergency messages, Object 6503h provides for further error messages. The corresponding error bit is set to 1 for as long as the error condition applies.

Data content:

Byte 0	Byte 1
$2^7 \dots 2^0$	$2^{15} \dots 2^8$

Bit No.	Description	Value = 0	Value = 1
Bit 0	Position error	Position value valid	Position error
Bit 1	Hardware check	No error	Error
Bit 2..15	Not used		

If an error occurs, then in both cases an emergency message (**ID=80h+node number**) with the error code **1000h (Generic error)** is sent.



Object is mappable

Object 6504h: Supported Alarms

This Object is used to display which alarm messages are supported by the encoder (see Object 6503h).

Data content:

Byte 0	Byte 1
$2^7 \dots 2^0$	$2^{15} \dots 2^8$

Range of values: see Object 6503h

The alarm message is supported when the bit is set to 1

Example:

Bit 0 = 1 Position error display is supported

Object 6505h: Warnings

Warning messages show that tolerances of internal encoder parameters have been exceeded. With a warning message – unlike with an alarm message or emergency message – the measured value can still be valid. The corresponding warning bit will be set to 1 for as long as the tolerance is exceeded or the warning applies.

Data content:

Byte 0	Byte 1
$2^7 \dots 2^0$	$2^{15} \dots 2^8$

Bit No.	Description	Value = 0	Value = 1
Bit 0	Overspeed	none	exceeded
Bit 1..15	Not used		

When Bit 0 is active then simultaneously an emergency message (ID=80h+node number) with the **Error code 4200h** (Device specific) is sent.



Object is mappable

Object 6506h: Supported Warnings

This Object is used to display which warning messages are supported by the encoder (see Object 6505h).

Data content:

Byte 0	Byte 1
$2^7 \dots 2^0$	$2^{15} \dots 2^8$

Range of values: see Object 6505h
The warning is supported when the bit is set to 1

Object 6400h: Working Area State Register

This Object contains the current state of the encoder position with respect to the programmed limits. The flags are either set or reset depending on the position of both limit values. The comparison with both limit values takes place in "real time" and can be used for real-time positioning or for limit switching.

Work_area_state							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
					smaller than LowLimit1	larger than HighLimit1	outside range1

Range of values 8-bit

Data content see Bit 0...7



Both limit values Object 6401h and 6402h must be checked to ensure that the output signals are correctly activated !



Object is mappable

Object 6401h: Working Area Low Limit Object 6402h: Working Area High Limit

These two parameters configure the working area. The state inside and outside this area can be signalled by means of Flag bytes (**Object 6400h Working Area State**). These area markers can also be used as software limit switches.

Data content:

Byte 0	Byte 1	Byte 2	Byte 3
$2^7 \dots 2^0$	$2^{15} \dots 2^8$	$2^{23} \dots 2^{16}$	$2^{31} \dots 2^{24}$



Range of values: 1....maximum physical resolution (268435456) 28-bit
Default setting: 16384 (14-bit) Working Area High Limit
 0 Working Area Low Limit

Object 2100h: Baud rate

This Object is used to change the baud rate via software. If the value is set between 1..9 and the parameter saved, then on the next Power ON or with a reset node, the device will boot up with the modified baud rate. After changing the baudrate it is necessary to save the parameters with **object 2105h** permanently in the EEPROM.

Byte 0
$2^7 \dots 2^0$

Data content:



Range of values 1 ...8 (see Table CANopen Baudrate)
Default setting: 05h

Object 2101h: Node address

This Object is used to change the node address via software. After changing the node address it is necessary to save the parameters with **object 2105h** permanently in the EEPROM.

Data content:

Byte 0
$2^7 \dots 2^0$



Range of values 1 ...127 or 1..7Fh

Default setting: 3Fh



The **node number 0** is reserved and may not be used by any node. The resulting node numbers lie in the range **1...7Fh** hexadecimal or (1...127)

The acceptance of a new node number only becomes effective when the encoder is rebooted (Reset/Power-on) or by means of an **NMT Reset Node** command. All other settings within the object table are however retained.

Object 2102h: CAN bus termination OFF/ON

This Object can be used to set the bus termination via software. By default the value is set to 1. After changing the CAN bus Termination it is necessary to save the parameters with **object 2105h** permanently in the EEPROM.

Data content:

Byte 0
$2^7 \dots 2^0$



Range of values 0..1

Default setting: 1



Please note that when software termination is selected, then the hardware settings are non-operative and vice versa.

Object 2103h: Firmware flash version

This object is used to display the current firmware version as a 16-bit hexadecimal value. This value serves to verify that the device is to the latest revision.

Data content:

Byte 0	Byte 1
$2^7 \dots 2^0$	$2^{15} \dots 2^8$

Range of values: to FFFFh

Example: 4FA6h current firmware

Object 2105h: Save All Bus Parameters

This object stores all bus parameters (Objekt 2100h ,2101h,2102h) permanently in an EEprom. Using the command "save" (save all Parameters) causes all the parameters to be stored. This process requires ca. 200ms. In order to prevent an inadvertent save, the instruction will only be executed if the string "save" is entered as a codeword into this Sub-Index.



Byte 0	Byte 1	Byte 2	Byte 3
$2^7 \dots 2^0$	$2^{15} \dots 2^8$	$2^{23} \dots 2^{16}$	$2^{31} \dots 2^{24}$

Default value: „save“ in hexadezimal **0x65766173**



Object 2140h: Customer Memory (16 Bytes)

These 4 parameters constitute a memory area for the user. **4 data words with a maximum of 4 bytes can be stored.** This area is not checked for content, which means in effect that any format can be filed.

Data content:

Byte 0	Byte 1	Byte 2	Byte 3
$2^7 \dots 2^0$	$2^{15} \dots 2^8$	$2^{23} \dots 2^{16}$	$2^{31} \dots 2^{24}$



Range of values: Numeric, alphanumeric

Default setting: **0**

Object 1029h Error Behaviour

If a serious error is detected, then the device should automatically switch to **Pre-Operational** mode. The settings in this Object can be used to determine how the device is to behave when an error arises. The following error classes are covered.

1029h,Subindex 1 Communication Errors

- Bus Off state of the CAN interface
- Life guarding event has occurred
- Heartbeat monitoring has failed

1029h,Subindex 2 Device Profile Specific

- Sensor error and Controller error
- Temperature error

1029h,Subindex 3 Manufacturer Profile Specific

- internal Controller error

The value of the Object classes is put together as follows:

Byte 0
$2^7 \dots 2^0$

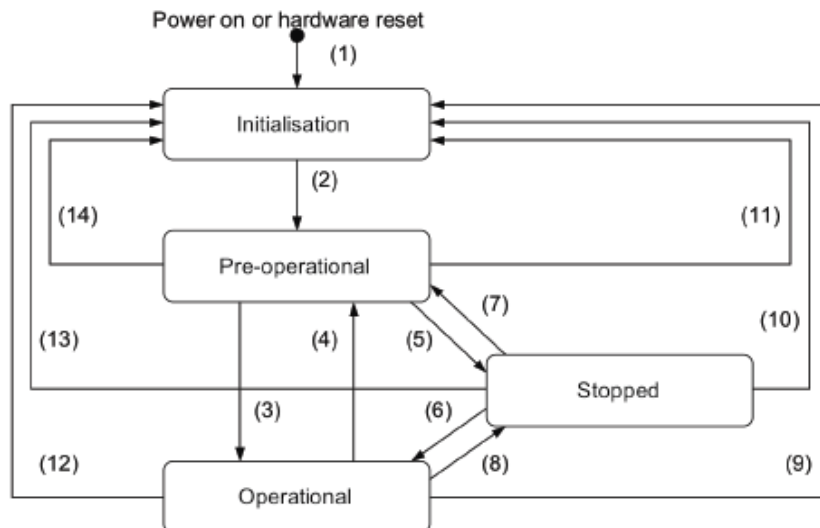
Range of values: 8-bit

- **0 Pre-Operational Mode (only if Operational Mode was active before)**
- **1 no change of mode**
- **2 Stopped Mode**
- **3 .. 127 reserved**

18 Network Management

The encoder supports the simplified Network Management as defined in the profile for "minimum capability devices" (minimum boot up).

The following function state diagram acc. to DS 301 shows the various node states and the corresponding network commands (controlled by the Network Master via NMT services):



(1)	At Power on the NMT state initialisation is entered autonomously
(2)	NMT state Initialisation finished - enter NMT state Pre-operational automatically
(3)	NMT service start remote node indication or by local control (self-starting)
(4),(7)	NMT service enter pre-operational indication
(5),(8)	NMT service stop remote node indication
(6)	NMT service start remote node indication
(9),(10),(11)	NMT service reset node indication
(12),(13),(14)	NMT service reset communication indication



Initialization: this is the initial state after the power supply is applied, following a device Reset or Power ON. The node automatically enters the Pre-operational state once it has run through the Reset and Initialization routines. The LEDs display the momentary status.

Pre-operational: The CAN node can now be addressed via SDO messages or with NMT commands under the standard identifier. Then follows the programming of the encoder or communication parameters.

Operational: The node is active. Process values are transmitted over the PDOs. All NMT commands can be evaluated.

Prepared or Stopped: In this state the node is no longer active, which means that neither SDO nor PDO communications are possible. The node can be set to either the Operational or Pre-operational state by means of NMT commands.

19 NMT Commands

All NMT commands are transferred as an unconfirmed NMT Object. Because of the broadcast (network-wide) communication model, the NMT commands are recognized by each station.

An NMT Object is structured as follows:

COB-ID = 0

Byte 0	Byte 1
$2^7 \dots 2^0$	$2^{15} \dots 2^8$

Byte 0 = Command byte

Byte 1 = Node number



The COB-ID of the NMT Object is always 0

The node is addressed via the node numbers. With node number 0 all nodes are addressed.

Kommandobyte (hex)	Beschreibung
01h	Start_Remote_Node: Wechsel zu Operational
02h	Stop_Remote_Node: Wechsel zu Prepared
80h	Enter_Pre-Operational_State: Wechsel zu Pre-operational
81h	Reset_Node: Reset Knoten ¹
82h	Reset_Communication: Reset Kommunikation ²

¹ On Power ON all the parameters in the whole Object Dictionary will have their values set.

² On Power ON only the parameters in the section Communication Profile of the Object Dictionary will have their values set.

Technical Manual

Absolute Singleturn Encoder **CANopen** series M36X8



20 LED states

greenLED = BUS State

red LED = ERR display



Annunciator	LED	Description	Cause of error	Addendum
Bus OFF		No connection to the Master ²	Data transmission line break Incorrect baud rate Inverted data line	Observe combination with ERR LED If ERR LED is also OFF, please check power supply ³
Bus flashing ca. 250ms		Connection to Master Pre-operational state		SDO communication
Bus flashing ca. 1sec		Connection to Master Stopped state		SDO communication not possible Only NMT commands
Bus ON		Connection to Master Operational state		PDO Transfer is active
ERR OFF		Device working normally		Observe combination with BUS LED
ERR flashing		Connection to Master interrupted	Combination with BUS status	BUS LED green, flashing or ON - is dependent on Object 1029h Error Behaviour
ERR ON		BUS OFF State	Short circuit on the Bus or Incorrect baud rate	
ERR +Bus flashing		LSS-Mode	LSS Mode Global selected	Device is waiting for a LSS-command

The individual LED annunciators can of course also occur in combinations.

² The Master can be a PLC or a second communication partner.

³ Operating voltage

21 Definitions

Explanation of Symbols:



This symbol highlights those parts of the text to which particular attention must be paid. This is to ensure correct usage and to eliminate danger. This symbol provides important advice concerning the proper handling of the encoder. Non-observance of this advice can lead to malfunctions of the encoder or in the vicinity.



This symbol refers to a special characteristic



Factory default setting of the parameter

22 Abbreviations used

CAL	CAN Application Layer. Application layer (layer 7) in the CAN Communication Model
CAN	Controller Area Network
CIa	CAN in Automation. International Association of Users and Manufacturers of CAN products
CMS	CAN Message Specification. Service element of CAL
COB	Communication Object. Transport unit in the CAN network (CAN message). Data will be sent over the network within a COB.
COB-ID	COB-Identifier. Unique identifier of a CAN message. The identifier defines the priority of the COB in the network.
DBT	Distributor. Service element of CAL, responsible for the dynamic allocation of identifiers.
DS	Draft Standard
DSP	Draft Standard Proposal;
ID	Identifier, see COB-ID
LMT	Layer Management. Service element of CAL, responsible for the configuration of the parameters in the individual layers of the communication model.
LSB	Least significant bit/byte
MSB	Most significant bit/byte
NMT	Network Management. Service element of CAL, responsible for the initialization, configuration and error handling in the network.
OSI	Open Systems Interconnection. Layer model for describing the function areas in a data communication system.
PDO	Process Data Object. Object for the exchange of process data.
RTR	Remote Transmission Request; Data request telegram.
SDO	Service Data Object. Communication Object, by means of which the Master can access the Object Dictionary of a node.
SYNC	Synchronization telegram. Stations on the Bus reply to the SYNC command by transmitting their process value.

23 Decimal-Hexadecimal Conversion Table

With numerical data, the decimal values are given as numerals with no affix (e.g. 1408), binary values are identified by the letter b (e.g. 1101b) and hexadecimal values with an h (e.g., 680h) after the numerals.

Dez	Hex	Dez	Hex	Dez	Hex	Dez	Hex
0	00	32	20	64	40	96	60
1	01	33	21	65	41	97	61
2	02	34	22	66	42	98	62
3	03	35	23	67	43	99	63
4	04	36	24	68	44	100	64
5	05	37	25	69	45	101	65
6	06	38	26	70	46	102	66
7	07	39	27	71	47	103	67
8	08	40	28	72	48	104	68
9	09	41	29	73	49	105	69
10	0A	42	2A	74	4A	106	6A
11	0B	43	2B	75	4B	107	6B
12	0C	44	2C	76	4C	108	6C
13	0D	45	2D	77	4D	109	6D
14	0E	46	2E	78	4E	110	6E
15	0F	47	2F	79	4F	111	6F
16	10	48	30	80	50	112	70
17	11	49	31	81	51	113	71
18	12	50	32	82	52	114	72
19	13	51	33	83	53	115	73
20	14	52	34	84	54	116	74
21	15	53	35	85	55	117	75
22	16	54	36	86	56	118	76
23	17	55	37	87	57	119	77
24	18	56	38	88	58	120	78
25	19	57	39	89	59	121	79
26	1A	58	3A	90	5A	122	7A
27	1B	59	3B	91	5B	123	7B
28	1C	60	3C	92	5C	124	7C
29	1D	61	3D	93	5D	125	7D
30	1E	62	3E	94	5E	126	7E
31	1F	63	3F	95	5F	127	7F

24 Glossary

Baudrate

The baud rate is the data transfer rate. It is linked to the nominal bit timing. The maximum possible baud rate is dependent on numerous factors that affect the transfer time on the bus. There is a significant connection between the maximum baud rate and the bus length and type of cable. In CANopen the various baud rates are defined between 10 Kbit/s and 1 Mbit/s.

CANopen

CANopen is a protocol based on CAN that was originally developed for industrial control systems. The specifications contain various device profiles as well as the framework for specific applications. CANopen networks are used in off-road vehicles, electronics on-board ships, medical equipment and the railways. The very flexible application layer together with the many optional features are ideal for tailor-made solutions. Furthermore, a wide variety of configuration tools are available. On this basis the user is able to define device profiles that are specific to his application. More information on CANopen can be found in the Internet at www.can-cia.org.

EDS file

The EDS (Electronic Data Sheet) is provided by the vendor/manufacturer of the CANopen device. It has a standardized format for describing the device. The EDS contains information concerning:

- Description of the file (name, version, date programme was generated etc.)
- General information about the device (manufacturer's name and code)
- Device name and type, Version, LMT address
- Supported baud rates, as well as boot-up capability
- Description of the attributes of supported Objects.

Node number

Every device within a CANopen network can be identified by its node number (Node-ID). The permitted range for node numbers is from 1 to 127 and each may only occur once within a network.

Network Management

In a distributed system, various tasks arise that have to do with the configuration, initialization and control of stations on the network. This functionality is provided in CANopen by the defined service element »Network Management (NMT)«.

PDO

The Process Data Objects (PDOs) provide the actual transport means for transferring the process data (Application Objects). A PDO is transmitted by a Producer and can be received by one or more Consumers.

PDO Mapping

The size of a PDO can be up to 8 byte. It can be used to transport several Application Objects. PDO Mapping describes the definition of the structure of the Application Objects within the data field of the PDO.

SDO

The confirmed transfer of data, of any length, between two stations on the network occurs via Service Data Objects (SDOs). Data transfer takes place in the Client-Server mode.