

Handbuch

Sendix 7058 / 7078 / 7158 / 7178
Absolute Singleturn Drehgeber

Sendix 7068 / 7088 / 7168 / 7188
Absolute Multiturn Drehgeber



CANopen®

Herausgeber	Kübler Group, Fritz Kübler GmbH Schubertstr. 47 78054 Villingen-Schwenningen Deutschland www.kuebler.com
Applikationssupport	Tel. +49 7720 3903 849 Fax +49 7720 21564 support@kuebler.com
Dokumenten-Nr.	R60712.0001
Dokumenten-Titel	Handbuch Sendix 7058 / 7078 / 7158 / 7178 Sendix 7068 / 7088 / 7168 / 7188 CANopen
Sprachversion	Deutsch (DE) – Englisch ist die Originalversion
Dokumentversion	28.06.2018, Dok-Nr. R60712.0001 - Index 3
Copyright	©2018, Kübler Group, Fritz Kübler GmbH
Rechtliche Hinweise	Sämtliche Inhalte dieser Gerätebeschreibung unterliegen den Nutzungs- und Urheberrechten der Fritz Kübler GmbH. Jegliche Vervielfältigung, Veränderung, Weiterverwendung und Publikation in anderen elektronischen oder gedruckten Medien, sowie deren Veröffentlichung im Internet, bedarf einer vorherigen schriftlichen Genehmigung durch die Fritz Kübler GmbH.

Inhaltsverzeichnis

Inhaltsverzeichnis

Abkürzungsverzeichnis

1. Technische Details und Eigenschaften.....	6
1.1 CANopen Multiturn- /Singleturn-Drehgeberserien 58X8.....	6
1.2 Das CANopen Kommunikationsprofil DS 301 V4.2.0.....	6
1.3 Drehgeber-Geräteprofil DS 406 V3.1.....	7
1.4 Ziele von LSS.....	7
1.5 Datenübertragung.....	7
1.6 Objekte und Funktionscode im Predefined Connection Set.....	8
1.7 Übertragung von Prozessdaten.....	9
2. Elektrische Installation und CAN-Bus.....	11
2.1 Versorgungsspannung (Stromversorgung).....	11
2.2 CANbus-Leitungen.....	11
2.3 Busabschluss.....	12
3. Erdung und Potentialausgleich.....	13
4. Quick Start Guide - Werksseitige Defaulteinstellungen.....	14
4.1 Drehgeber mit Kabelabgang.....	14
4.2 Kommunikationsparameter.....	14
4.3 Drehgeberprofil.....	15
5. Emergency Objekte.....	17
6. Heartbeat Protokoll.....	18
7. Objekte des Drehgeberprofils DS 306.....	19
7.1 Objekt 6000h Operating Parameters.....	19
7.2 Objekt 6001h: measuring steps per revolution (Auflösung).....	20
7.3 Objekt 6002h: Total number of measuring steps.....	20
7.4 Objekt 6003h: Preset Value.....	20
7.5 Objekt 6004h: Position Value.....	21
7.6 Objekt 6030h: Speed Value.....	21
7.7 Objekt 604h: Acceleration Value.....	21
7.8 Objekt 6200h: Cyclic Timer.....	22
7.9 Objekt 6500h: Display Operating Status.....	22
7.10 Objekt 6502h: Number of Multiturn revolutions.....	22

7.11	Objekt 6503h: Alarms	23
7.12	Objekt 6504h: Supported Alarms	23
7.13	Objekt 6505h: Warnings	23
7.14	Objekt 6506h: Supported Warnings	24
7.15	Objekt 6400h: Working Area State Register 2 Werte	24
7.16	Objekt 2100h: Baud rate	25
7.17	Objekt 2101h: Node address	25
7.18	Objekt 2102h: CAN bus termination OFF/ON	26
7.19	Objekt 2103h: Firmware flash version	26
7.20	Objekt 2105h: Save All Bus Parameters	27
7.21	Objekt 2110h: Sensor Configuration Data	27
7.22	Objekt 2120,4h: Actual temperature Position-Sensor *	27
7.23	Objekt 2120,2h: Actual temperature lower limit Position-Sensor	28
7.24	Objekt 2120,3h: Actual temperature upper limit Position-Sensor	28
7.25	Objekt 2130h: Encoder Measuring Step	29
7.26	Objekt 2140h: Customer Memory (16 Bytes)	29
7.27	Objekt 2150h: Temperature History	29
8.	Netzwerkmanagement	31
9.	NMT-Kommandos	32
10.	Glossar	33

Abkürzungsverzeichnis

Abkürzung	Bedeutung
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 CAN-Produkten.
CMS	CAN Message Specification. Service-Element von CAL.
COB	Communication Object. Transporteinheit im CAN Netzwerk (CAN Nachricht). Daten werden innerhalb eines COB über das Netzwerk gesendet.
COB-ID	C OB-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.
MT	Multiturn-Drehgeber
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.

1. Technische Details und Eigenschaften

1.1 CANopen Multiturn- /Singleturn-Drehgeberserien 58X8

Die CANopen-Drehgeber-Serie 7068/88 unterstützt das neueste CANopen-Kommunikationsprofil nach **DS 301 V4.02**. Zusätzlich wurden gerätespezifische Profile wie das Drehgeber-Profil **DS 406 V3.1** angepasst. Folgende Betriebsarten können gewählt werden: Polled Mode, Cyclic Mode, Sync Mode und ein High Resolution Sync Protokoll.

Weiterhin lassen sich Skalierungen, Presetwerte und viele weitere, zusätzliche Parameter über den CAN-Bus programmieren. Beim Einschalten werden sämtliche Parameter aus einer EEPROM geladen, die zuvor in dem nichtflüchtigen Speicher abgespeichert wurden, um sie gegen Stromausfall zu schützen. Folgende Ausgabewerte können sehr variabel als **PDO** (PDO Mapping) kombiniert werden: **Position, Geschwindigkeit, Beschleunigung**, sowie der Zustand der vier **Endschalter**.

Die Drehgeber sind mit **Kabelverbindung** erhältlich, die Änderung der Geräte-Adresse und der Baudrate erfolgen mittels Software. Diese Modelle haben einen integrierten T-Koppler für eine einfache Installation.

1.2 Das CANopen Kommunikationsprofil DS 301 V4.2.0

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. Die CAN in Automation (CiA) Hersteller- und Benutzergruppe ist für die Erarbeitung und Standardisierung der relevanten Profile zuständig.

CANopen bietet

- benutzerfreundlichen Zugang zu allen Geräteparametern
- Auto-Konfiguration des Netzwerkes und der Geräte
- Gerätesynchronisation innerhalb des Netzwerkes
- zyklischen und ereignisgesteuerten Prozessdatenverkehr
- gleichzeitiges Lesen und schreiben 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.

1.3 Drehgeber-Geräteprofil DS 406 V3.1

Dieses Profil beschreibt eine **Herstellerunabhängige** und verbindliche Festlegung der Schnittstelle für Drehgeber. In dem Profil wird festgelegt, welche CANopen-Funktionen verwendet werden, und wie sie verwendet werden. Dieser Standard ermöglicht ein offenes und herstellerunabhängiges Bussystem.

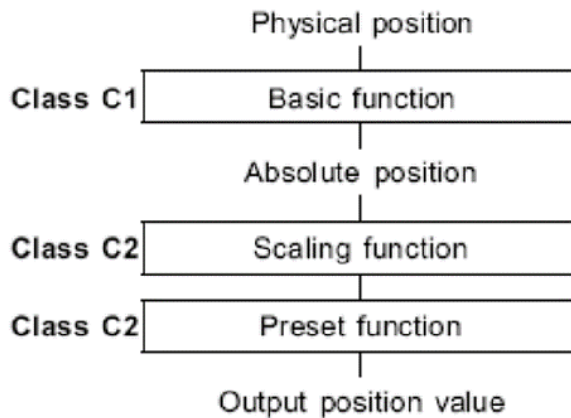


Abb. 1

Das Geräteprofil ist gegliedert in zwei Objekt-Klassen:

- Die **Klasse C1** beschreibt alle Grundfunktionen, die der Drehgeber aufweisen soll.
- 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 im Profil außerdem ein Adressbereich definiert, der mit herstellereigenen Sonderfunktionen belegt werden kann.

1.4 Ziele von LSS

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

1. CANopen-Knoten-ID
2. CANopen-Baudrate
3. LSS-Adresse

Dadurch werden die „plug-and-play“-Fähigkeiten der Geräte auf CANopen-Netzwerken erhöht, da die Vorkonfiguration des Netzwerks weniger restriktiv ist. Der LSS-Master ist verantwortlich für die Konfiguration dieser Parameter von einem oder mehreren LSS-Slaves in einem CANopen-Netzwerk.

1.5 Datenübertragung

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

- **Prozessdatenobjekte (PDO - echtzeitfähig),**
- **Servicedatenobjekte (SDO)**

Die Prozessdatenobjekte (**PDO**) bieten einen Hochgeschwindigkeit-Austausch von Echtzeitdaten (z.B. Drehgeber-Position, Geschwindigkeit, Positions-Status-Vergleich) mit einer Maximallänge von 8 Byte. 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 Drehzahl 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-Identifier).

1.6 Objekte und Funktionscode im Predefined Connection Set

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

Dieser 11 Bit-Identifier besteht aus einem **4 Bit-Funktionscode** und einer **7 Bit-Knoten-ID-Nummer**. Je höher der Wert des COB-Identifiers, desto niedriger die Priorität!

Broadcast (netzwerkweite) Objekte

object	function code (binary)	resulting COB-ID	Communication Parameters at Index
NMT	000	0	-
SYNC	0001	128 (50h)	1005h, 1006h, 1007h
TIME STAMP	0010	256 (100h)	1012h, 1013h

Abb. 2

Peer-To-Peer (Gerät-zu Gerät) Objekte

object	function cond (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

Abb. 3

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

Abb. 4

1.7 Übertragung von Prozessdaten

In einem CANopen-Drehgeber stehen **drei PDO-Dienste**: PDO1 (tx), PDO2 (tx) und PDO3(tx), sowie ein **Receive-PDO**, 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, genau diese Nachricht mit dem übertragenen ID wird abgefragt)

Die **PDO-Nachrichten** könnten folgende Struktur aufweisen

Process Data in Binary Code							
Byte 0 $2^7 \dots 2^0$	Byte 1 $2^{15} \dots 2^8$	Byte 2 $2^{23} \dots 2^{16}$	Byte 3 $2^{31} \dots 2^{24}$	Byte 4	Byte 5	Byte 6	Byte 7
PDO 3				Position value			
PDO 1				Position value			
PDO 2				Speed ²⁾		Acceleration ³⁾	

Abb. 5

1Flags Status Byte des Arbeitsbereichs (Working area) Objekt 6400h

2Geschwindigkeit 16 Bit-Wort vorzeichenbehaftet

3Beschleunigung 16 Bit-Wort vorzeichenbehaftet

Transmit PDO 1 besteht aus (ist gemappt mit) dem 32 Bit-Positionswert-Status der **Working area-Register** (6400h).

Transmit PDO 2 besteht aus den 32 Bit-Positionswerten, 16 Bit-Geschwindigkeit und 16 Bit-Beschleunigung.

Transmit PDO 3 besteht aus der Position als **SYNC PDO**.

Alle anderen **PDO-Kombinationen** mit anderen Objekten sind möglich, solange die maximale 8 Byte-Datenlänge nicht überschritten wird.

2. Elektrische Installation und CAN-Bus

2.1 Versorgungsspannung (Stromversorgung)

Die Versorgungsspannung an **Adern 1 und 2 (0V) und (+ VDC)** anschließen.

Type:										mA	
Series -Nr.					VDC						
SIG.	0V	+V	CAN_H	CAN_L	CAN GND	CAN_H	CAN_L	CAN GND			☐
Lead nr.	1	2	4	5	6	7	8	9			PH
Ex II 2G Ex d II C T6 PTB 09 ATEX 1106 X										Working temperature	
II 2D EX tD A21 IP6X T85°C CE 0102										-40°...+60°C	

Abb. 6

2.2 CANbus-Leitungen

Die CANbus-Eingangsleitungen an **Adern 4 und 5 (CAN_H) und (CAN_L)**, und die Ausgangsleitungen an **Adern 7 und 8** anschließen.

Type:										mA	
Series -Nr.					VDC						
SIG.	0V	+V	CAN_H	CAN_L	CAN GND	CAN_H	CAN_L	CAN GND			☐
Lead nr.	1	2	4	5	6	7	8	9			PH
Ex II 2G Ex d II C T6 PTB 09 ATEX 1106 X										Working temperature	
II 2D EX tD A21 IP6X T85°C CE 0102										-40°...+60°C	

Abb. 7

3. Erdung und Potentialausgleich

Gute Erdung und Potentialausgleich sind für die Störfestigkeit der CANbus-Netze sehr wichtig. Erdung und Potentialausgleich dienen daher vorwiegend dem guten Betrieb des CANbus und nicht der Sicherheit. Eine sachgemäße Erdung des Kabelschirms stellt die Reduzierung der elektrostatischen Störungen sicher und minimiert somit die Einstreuungen. Der Potentialausgleich stellt sicher, dass das Erdpotential in dem ganzen Netzwerk gleich ist. Dies vermeidet das fließen von Erdströmen über den Schirm des CANbus-Kabels. Die folgenden Informationen geben allgemeine Anleitungen für die Installation der Erdung und des Potentialausgleichs.

An dem CANbus-Teilnehmer

Den **CANbus-Kabelschirm** an jedem CANbus-Teilnehmer mit dem Potentialausgleich verbinden. Wenn er verwendet wird sichert der CANbus-Stecker die Verbindung des Kabelschirms. Dazu muss jedoch der Kabelschirm sachgemäß mit dem Stecker verbunden werden.

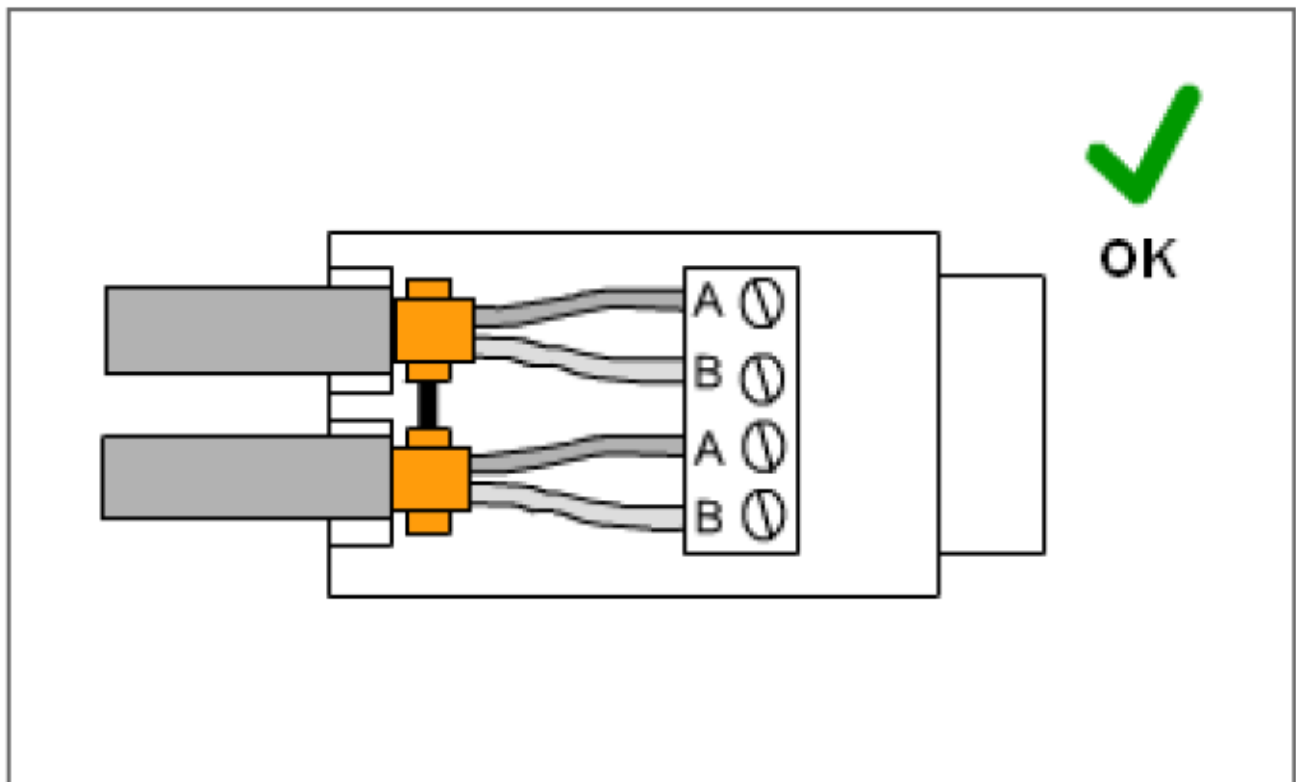


Abb. 9

4. Quick Start Guide - Werksseitige Defaulteinstellungen

4.1 Drehgeber mit Kabelabgang

Description	Setting	Switch	Software
Baud rate	250 Kbit/s	Switch setting 5	Object 2100h = 5h
Node address	63	Switch setting 3Fh	Object 2100h = 3Fh
Termination	Off	Switch setting off	Object 2100h = 01h

Abb. 10

4.2 Kommunikationsparameter

Index (hex)	Name	Standard value
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 Error
1800h	TPD01 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	TPD02 Communication Parameter	
01h	COB-ID	280h + Node number
02h	Transmission Type	01 (asynch)
03h	Inhibit Time	0 [steps in 100µs]
05h	Event Timer	0 [steps in ms]

1802h	TPD03 Communication Parameter	
01h	COB-ID	380h + Node number
02h	Transmission Type	255 / 0xFFh (synch)
03h	Inhibit Time	0 [steps in 100µs]
05h	Event Timer	0 [steps in ms]
1A00h	TPD01 Mapping	
01h	1. Mapped Object	0x60040020
1A01h	TPD02 Mapping	
01h	2. Mapped Object	0x60300110
1A02h	TPD03 Mapping	
01h	3. Mapped Object	0x60040020

Abb. 11

4.3 Drehgeberprofil

Index (hex)	Name	Standard value
6000h	Operating Parameter	0x04h Scaling on
6001h	Measuring Units per Revolution	8192 (13 Bit)
6002h	Total Measuring Range	33554432 (25 Bit)
6003h	Preset value	0
6200h	Cyclic Timer (see TPD01 Comm. Par)	0
6401h	Work area low limit	0
6402h	Work area high limit	65535
2105h	Save all Bus Parameters	0x65766173
2130h	Encoder Measuring Step	
	Speed Calculation Multiplier	10
	Speed Calculation Divisor	10
	Speed average value	10

Abb. 12

Die originalen Standardwerte (werksseitig eingestellte Defaultwerte) können über Objekt **1011h** (restore parameters) neu geladen werden. Um sicherzustellen, dass Parameteränderungen für den Fall eines Stromausfalls gespeichert werden müssen sie unbedingt über Objekt **1010h** (store parameters) in die EEPROM geladen werden. Dadurch werden alle bereits in der EEPROM gespeicherten Daten überschrieben!

Sind während der Programmierung der Objekte Fehler aufgetreten und sind diese Parameter in der EEPROM abgespeichert worden, so wird es nicht möglich sein, den Drehgeber beim nächsten Einschalten anzusprechen (der Drehgeber wird nur **Emergency**-Nachrichten aussenden) Dieser Fehler kann nur durch einen allgemeinen **Reset** des Drehgebers behoben werden.

Default-Einstellung für das Mapping des Transmit-PDO:

Mapping	TPD01	TPD02	TPD03
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

Abb. 13

Der CANopen-Drehgeber unterstützt das **variable Mapping** auf allen 3 Transmit-PDOs.

5. 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. Nach Behebung des Fehlers wird ein neues Emergency Object mit Inhalt 0 (Fehler zurückgesetzt oder Kein Fehler) erzeugt und über den Bus übertragen.

Unterstützte Fehlercodes

Die Fehlercodes werden in **orange** 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

Abb. 14

6. Heartbeat Protokoll

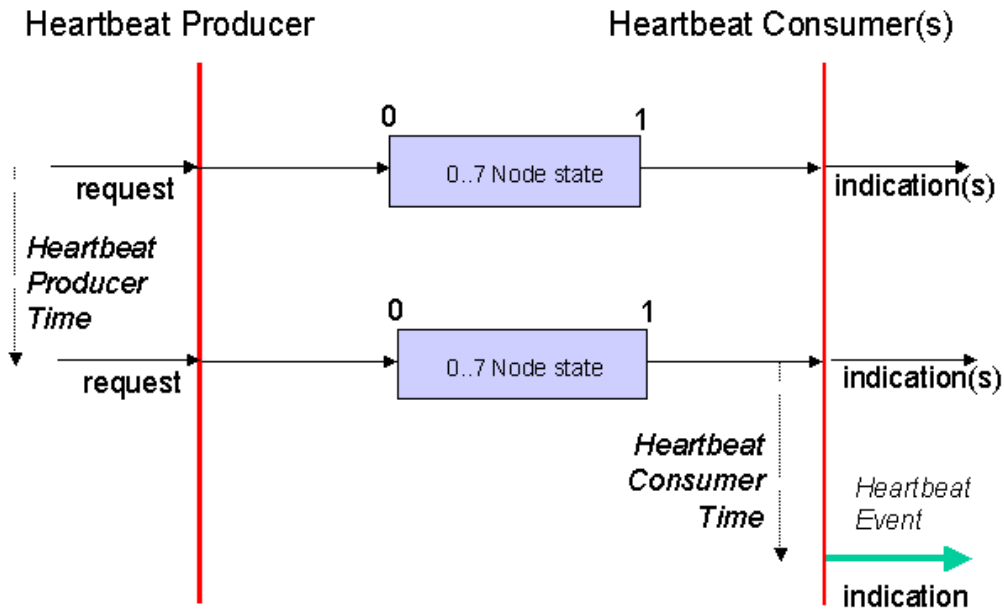


Abb. 15

Heute sollte das moderne **Heartbeat Protokoll** als Alternative zum **Node Guarding** verwendet werden. Dieses Protokoll wird aktiviert wenn ein Wert > 0 in **Objekt 1017h** Producer Heartbeat Time geschrieben wird.

Ein „Heartbeat-Producer“ überträgt die Heartbeat-Nachricht zyklisch. Ein oder mehrere „Heartbeat-Consumer“ können diese Heartbeat-Nachricht erhalten. Wenn die zyklische Übertragung dieser Heartbeat-Nachricht ausbleibt wird ein „Heartbeat Event“ generiert. Das Verhalten im Fehlerfall wird in Objekt 1029h Subindex 1 „Communication Error“ beschrieben.

7. Objekte des Drehgeberprofils DS 306

7.1 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 = 1 (s. Objekt 6001,6002)
Default: Bit = 1
- Bit13: Geschwindigkeitsformat: 0 = Umdrehungen/Min., 1 = Einheiten/ Sek.
Default Bit = 0
- Bit14: Startup Mode: 0 = nach Bootup Pre-Operational, 1 = nach Bootup Operational Modus
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	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 ... 11	Reserved for further use				
12	Manufacturer specific parameter	N.A.	N.A.	O	O
13	Speed Format	RPM	Unit/sec	O	O
14	Sartup automatic in OP-Mode	Disabled	Enabled	O	O
15	Event Mode Position*	Disabled	Enabled	O	O

Abb. 16

* diesen Modus in Transmission Type in TPDO auf 254 setzen

M = Funktion muss unterstützt sein

O = optional

7.2 Objekt 6001h: measuring steps per revolution (Auflösung)

Dieser Parameter konfiguriert die gewünschte Auflösung pro Umdrehung. Der Drehgeber errechnet dann intern den entsprechenden Maßstabfaktor. Der errechnete Maßstabfaktor MUR (durch den der physische Positionswert multipliziert wird) wird gemäß folgender Formel festgelegt:

MURF = Messschritte pro Umdrehung (6001h) / phys. Singleturn-Auflösung (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...maximale physische Auflösung (65536) 16-Bit

Default-Einstellung: 8192 (13 Bit)

7.3 Objekt 6002h: Total number of measuring steps

Dieser Parameter konfiguriert die Gesamtanzahl der **Singleturn- und Multiturn-Messschritte**. Ein Faktor wird für die maximale physische Auflösung angewendet. Dieser Faktor ist immer < 1 . Nach dem Erreichen der festgelegten Anzahl Messschritte setzt sich der Drehgeber auf Null zurück.

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...maximale physische Auflösung (268435456) 28-Bit

Default-Einstellung: 33554432 (25 Bit)

Beispiel: Eingabe 200000h

Der physische Positionswert wird durch einen Faktor 0 multipliziert. XXXXXX und Ausgabe als End-Position.

7.4 Objekt 6003h: Preset Value

Der Positionswert des Gebers wird auf den eingegebenen 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: 1...maximale physische Auflösung (268435456) 28-Bit

Default-Einstellung: 0

7.5 Objekt 6004h: Position Value

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: 1...maximale physische Auflösung (268435456) 28-Bit

7.6 Objekt 6030h: Speed Value

Der Geber gibt die aktuelle errechnete Geschwindigkeit (eventuell mit Skalierungsfaktor) als 16 Bit-Wert aus.

Die Geschwindigkeit ist abhängig von den **Einstellungen von Objekt 2130h**. Diese Werte beeinflussen die Berechnung und das Ergebnis.

Dateninhalt:

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

Wertebereich: 0... maximale Geschwindigkeit 15000 U/min

HINWEIS

Bei Werten > 12000 U/min wird eine Warnmeldung ausgegeben und das Warning-Bit „Drehzahlüberschreitung Bit 0“ im Objekt Warnings 6505h gesetzt.

Parameter, die auch einen Einfluss auf dieses Objekt haben, sind in 2130h erwähnt.

7.7 Objekt 604h: Acceleration Value

Der Geber gibt die aktuelle errechnete Beschleunigung (vorzeichenrichtig) als vorzeichenbehafteter 16 Bit Wert aus. Die Beschleunigung wird aus den Geschwindigkeitsänderungen errechnet und ist deshalb auch indirekt von den **Einstellungen von Objekt 2130h** abhängig. Diese Werte beeinflussen die Berechnung und das Ergebnis.

Dateninhalt:

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

HINWEIS

Wertebereich: 0.... +/- maximale Beschleunigung

Negative Werte bedeuten eine negative Beschleunigung (Drehzahl sinkt)

Eine mittlere Beschleunigung **a** ist die zeitliche Änderung der Geschwindigkeit **v** und lässt sich somit formal aus der Ableitung Geschwindigkeit nach der Zeit **t** beschreiben, hier wird eine **mittlere** Beschleunigung aus der Differenz der Geschwindigkeiten Δv zu 2 verschiedenen Zeitpunkten Δt ($t_2 - t_1$) errechnet.

$$a = \Delta v / \Delta t \text{ oder } a = v_2 - v_1 / t_2 - t_1$$

7.8 Objekt 6200h: Cyclic 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.

HINWEIS

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

Standardwert = 0h

7.9 Objekt 6500h: Display Operating Status

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

7.10 Objekt 6502h: Number of Multiturn revolutions

Über dieses Objekt ist die Anzahl der Umdrehungen auslesbar, die der Multiturn Drehgeber zählen soll. Der Wert ist abhängig vom Gebertyp und kann von 4096 (12Bit) bis 65535 (16 Bit) jeden beliebigen Wert aufnehmen.

Dieser vordefinierte Wert beeinflusst nur die Anzahl Umdrehungen. Er hat keinen Einfluss auf die Auflösung.

Dateninhalt:

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

Wertebereich: 4096 bis 65535

Defaulteinstellung 1000h entspricht 4096

7.11 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 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		

Abb. 17

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

7.12 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: siehe Objekt 6503h

Bit gesetzt auf 1 bedeutet, dass die Alarmmeldung unterstützt wird

Beispiel: Bit 0 = 1 Positionsfehleranzeige wird unterstützt

7.13 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 No.	Description	Value = 0	Value = 1
Bit 0	Overspeed	none	exceeded
Bit 1	Not used		
Bit 2	Watchdog Status	System OK	Reset carried out
Bit 3	Operting time	Below < 100000h	> 100000h
Bit 4...15	Not used		

Abb. 18

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

7.14 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: siehe Objekt 6505h

Bit gesetzt auf 1 bedeutet, dass die Warnmeldung unterstützt wird.

7.15 Objekt 6400h: Working Area State Register 2 Werte

Dieses Objekt beinhaltet den aktuellen Status der Drehgeber-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
1 = CCW		smaller than LowLimit2	larger than HighLimit2	inside range2	smaller than LowLimit1	larger than HighLimit 1	inside range1

Abb. 19

Wertebereich 8 Bit

Dateninhalt siehe Bit 0...7

HINWEIS

Beide Endwerte Objekt 6401h und 6402h müssen überprüft werden, um sicherzustellen, dass die Ausgangssignale korrekt aktiviert sind:

Objekt 6401h: Working Area Low Limit 2 Werte

Objekt 6402h: Working Area High Limit 2 Werte

Diese beiden Parameter konfigurieren den Arbeitsbereich. Innerhalb und außerhalb dieses Bereiches kann der Status über Flagbytes (**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...maximale physische Auflösung (268435456) 28-Bit

Default-Einstellung: **33554432 (25 Bit) Working Area High Limit**

0 Working Area Low Limit

7.16 Objekt 2100h: Baud rate

Über dieses Objekt kann die Baudrate softwaremäßig verändert werden. Standardmäßig ist der Wert auf FFh eingestellt, d.h. die hardwaremäßige Einstellung der Baudrate hat Vorrang. Wird der Wert zwischen 1..9 eingestellt und der Parameter gespeichert, so bootet das Gerät beim erneuten Einschalten oder Reset Node mit der geänderten Baudrate. Nach der Änderung der Baudrate müssen die Parameter über **Objekt 2105h** permanent in der EEPROM gespeichert werden.

Dateninhalt:

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

Wertebereich 1 ...8 (siehe Hardware-Schalter-Tabelle: CANopen Baudrate)

Default-Einstellung: 0x05

7.17 Objekt 2101h: Node address

Über dieses Objekt kann die Knotenadresse softwaremäßig verändert werden. Standardmäßig ist der Wert auf 0xFFh eingestellt, d.h. die hardwaremäßige Einstellung der Knotenadresse hat Vorrang. Nach der Änderung der Knotenadresse müssen die Parameter über **Objekt 2105h permanent in der EEPROM gespeichert werden.**

Dateninhalt:

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

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

Default-Einstellung: FFh

Knotennummer 0 ist reserviert und darf für keinen Knoten verwendet werden. Die resultierenden Knotennummern befinden sich 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. Alle anderen Einstellungen innerhalb der Objekttabelle bleiben aber erhalten.

7.18 Objekt 2102h: CAN bus termination OFF/ON

Über dieses Objekt kann die Buserminierung softwaremäßig verändert werden. Standardmäßig ist der Wert auf 0 eingestellt, d.h. die hardwaremäßige Einstellung der Buserminierung hat Vorrang. Nach der Änderung der Buserminierung müssen die Parameter über **Objekt 2105h** permanent in der EEPROM gespeichert werden.

Dateninhalt:

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

Wertebereich 0..1

Default-Einstellung: 0 *Terminierung Ein an Drehgebern mit Kabelabgang und einem M12-Stecker

Bitte beachten: wenn die Software-Terminierung gewählt wurde sind die Hardware-Einstellungen inaktiv und umgekehrt.

7.19 Objekt 2103h: Firmware flash version

Über dieses Objekt wird die aktuelle Firmwareversion als 16-Bit Hexadezimalwert angezeigt.

Mit diesem Wert wird überprüft, ob die neueste Version installiert ist.

Dateninhalt:

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

Wertebereich: bis FFFFf

Beispiel: 4FA6h aktuelle Firmware

7.20 Objekt 2105h: Save All Bus Parameters

Dieses Objekt speichert alle Busparameter (Objekt 2100h, 2101h, 2102h) permanent in einer EEPROM. Mit dem Befehl "save" (save all parameters) werden alle Parameter abgespeichert. Dieser Vorgang benötigt ca. 200ms. 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 dem Index ergibt 0xFFFFFFFF.

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

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")

Wertebereich: „save“ in hexadezimal 0x65766173

7.21 Objekt 2110h: Sensor Configuration Data

Dieses Objekt liefert Informationen über die Konfiguration des Sensors. Als Default werden Werks-Voreinstellungen geladen, d.h. dass normalerweise keine Änderungen erforderlich sind.

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

Wertebereich: 0... FF,FFh...

7.22 Objekt 2120,4h: Actual temperature Position-Sensor *

Über dieses Objekt kann die Ist-Temperatur ausgelesen werden. Die Ist-Temperatur im Gerät wird alle 6 Minuten unter **Objekt 2150,sub1 Last Stored Temperature** gespeichert. Die Maximal- und Minimaltemperatur werden unter

Objekt 2130,sub2 und sub 3 gespeichert. Der maximale Wertebereich ist 1...256.

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

Wertebereich 00...FFh

Beispiel: **0x59** entspricht ca. **25°C**

Folgende Temperaturwerte sind eingestellte Bezugswerte:

-20°C entspricht **0x2Ch**

0°C entspricht **0x40h**

100°C entspricht **0xA4h**

Beispiel: Gewählter Wert 0x71h von **Objekt 2120,4h**

0x71h – 0x40h = 0x31h entspricht 49°C dezimal

HINWEIS

Dieses Objekt kann zu den PDO-Informationen gemappt werden. Die Genauigkeit des Messwerts ist im Durchschnitt $\pm 6^\circ\text{C}$, gemessen durch die interne Sensor-Logik.

7.23 Objekt 2120,2h: Actual temperature lower limit Position-Sensor

7.24 Objekt 2120,3h: Actual temperature upper limit Position-Sensor

Diese beiden Parameter konfigurieren den Temperatur-Arbeitsbereich. Außerhalb dieses Bereiches kann der Status über eine **Notfall-Nachricht** gemeldet werden. Diese Bereichsmarker können auch als eine Art Temperatur-Endschalter verwendet werden.

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

Wertebereich 00...FFh

Beispiel: **0x20** entspricht ca. **-32°C**

Folgende Temperaturwerte sind eingestellte Bezugswerte:

-20°C entspricht **0x2Ch**

0°C entspricht **0x40h**

100°C entspricht **0xA4h**

Wertebereich: 0x20h .. 0xACh

Default-Einstellungen: 0xA2h Temperature High Limit
0x20h Temperature Low Limit

7.25 Objekt 2130h: Encoder Measuring Step

Mit Hilfe von Parameter **Objekt 2130,sub1 Speed Calculation Multiplier** ist es möglich z. B. den Umfang eines Messrades zu bestimmen, um die Position in mm auslesen zu können .

Mit diesem Objekt wird die Ausgabe der Geschwindigkeit festgelegt. Unter **Objekt 2130,sub2 Speed Calculation Divisor** steht ein Parameter als Divisor für einen Einheitenfaktor zur Verfügung. Unter **Objekt 2130,sub3 Speed Average Value** wird die Anzahl der für die Errechnung des gleitenden Durchschnitts benötigten Messwerte eingegeben. Der maximale Wertebereich ist 1...32.

Diese Parameter beeinflussen nur die **Einheiten pro Sekunde**.

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

Wertebereich: siehe Tabelle

2130h Sub 1 Speed Calculation Multiplier Default-Einstellung: 10

2130h Sub 2 Speed Calculation Divisor Default-Einstellung: 10

2130h Sub 3 Speed Average Calc Value Default-Einstellung: 10

7.26 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**. Dieser Bereich wird nicht auf Inhalt geprüft, d.h. jegliches Format kann abgelegt 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: Numerisch, alphanumerisch

Default-Einstellung: 0

7.27 Objekt 2150h: Temperature History

Über dieses Objekt kann die Temperatur ausgelesen werden. Die Ist-Temperatur im Gerät wird alle 6 Minuten unter **Objekt 2150,sub1 Last Stored Temperature** gespeichert. Die Maximal- und Minimaltemperaturen werden unter **Objekt 2130,sub2 und sub 3** gespeichert. Der maximale Wertebereich ist 1...256.

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

Dateninhalt:

Ein Wert 0x50 entspricht ca. 20°C

Ein Wert 0x40 entspricht ca. 0°C

Ein Wert 0x90 entspricht ca. 85°C

2150h Sub 1 Last Stored Temperature	Default-Einstellung: 0x50
2150h Sub 2 Temperature maximum Val	Default-Einstellung: 0x50
2150h Sub 3 Temperature minimum Val	Default-Einstellung: 0x50
2150h Sub 4 Flag Byte	Default-Einstellung: 0

Objekt 1029h: Error Behaviour

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 Communication Errors

- Bus-Off Zustand der CAN-Schnittstelle
- Life guarding Ereignis ist aufgetreten
- Heartbeat Überwachung ist fehlgeschlagen

1029h, Subindex 2 Device Profile Specific

- Sensorfehler und Controllerfehler
- Temperaturfehler

1029h, Subindex 3 Manufacturer Profile Specific

- Interner Controllerfehler

Der Wert der Objekt-Klassen setzt sich wie folgt zusammen:

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

Wertebereich: 8-Bit

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

Nicht genannte Objekte

Alle hier nicht genannten Objekte dienen der zusätzlichen Information und können dem **Drehgeberprofil DS 406 V3.1** entnommen werden.

8. Netzwerkmanagement

Der Geber unterstützt das im Profil für „minimum capability devices“ definierte, vereinfachte Netzwerkmanagement (minimum boot up).

Folgendes Funktionen-Zustandsdiagramm nach DS 301 zeigt die unterschiedlichen Knoten-Zustände und die entsprechenden Netzwerk-Kommandos (gesteuert vom Netzwerk-Master über NMT-Dienste):

- | | |
|------------------|---|
| (1) | Beim Einschalten wird die Initialisierung des NMT-Zustandes selbständig eingeleitet |
| (2) | Initialisierung des NMT-Zustandes beendet - automatisches Umschalten in den NMT-Zustand Pre-operational |
| (3) | NMT Dienst Start Remote Node Indication oder durch lokale Steuerung (Selbststart) |
| (4), (7) | NMT Dienst Enter Pre-Operational Indication |
| (5), (8) | NMT Dienst Stop Remote Node Indication |
| (6) | NMT Dienst Start Remote Node Indication |
| (9), (10), (11) | NMT Dienst Reset Node Indication |
| (12), (13), (14) | NMT Dienst Reset Communication Indication |

Initialisierung:

Nach einem Reset des Gerätes oder nach dem Einschalten ist dies der Ausgangszustand nach Anlegen der Versorgungsspannung.

Nach dem Durchlaufen der Reset- und Initialisierungs-Routinen schaltet der Knoten automatisch in den Status Pre-operational. Die LED's zeigen den momentanen Zustand an.

Pre-operational:

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

9. NMT-Kommandos

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:

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

COB-ID = 0

Byte 0 = Kommandobyte

Byte 1 = Knotennummer

HINWEIS

Der COB-ID des NMT-Objektes ist immer 0

Der Knoten wird über die Knoten-Nummer adressiert. Bei Knoten-Nummer 0 werden alle Knoten angesprochen.

Kommandobyte (hex)	Description
01h	Start_Remote_Node: Switch to Operational
02h	Stop_Remote_Node: Switch to Prepared
80h	Enter_Pre-Operational_State: Switch to Pre-operational
81h	Reset_Node: Reset node ¹
82h	Reset_Communication: Reset Communication ²

Abb. 20

¹ Beim Einschalten werden die Werte aller Parameter des gesamten Objektverzeichnisses gesetzt.

² Beim Einschalten werden nur die Werte der Parameter im Abschnitt Kommunikationsprofil des Objektverzeichnisses gesetzt.

10. 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.

Kübler Group
Fritz Kübler GmbH
Schubertstr. 47
78054 Villingen-Schwenningen
Deutschland
Tel.: +49 7720 3903-0
Fax: +49 7720 21564
info@kuebler.com
www.kuebler.com