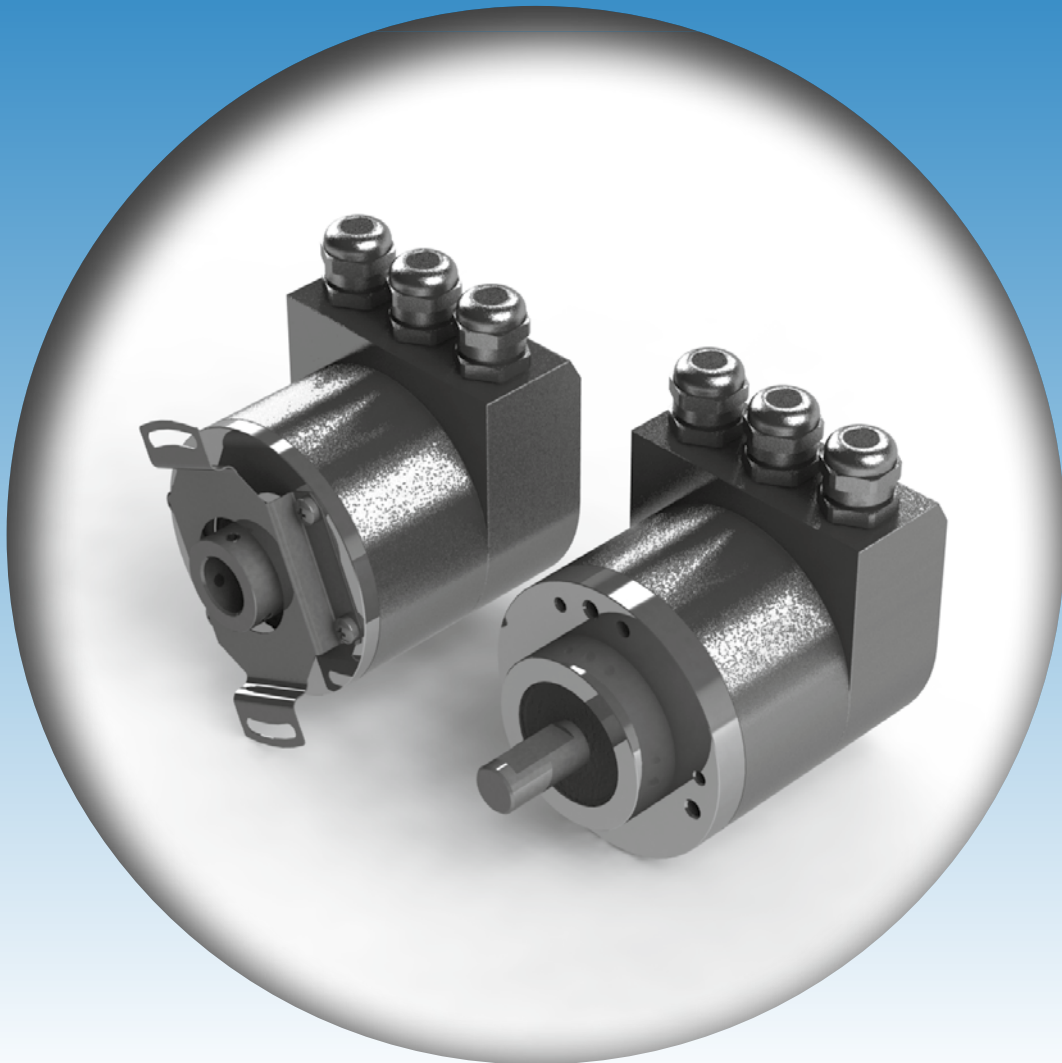


# **hohner**

Elektrotechnik Werke

## **Technisches Handbuch**

Absolut Drehgeber  
ACURO® industry mit **CANopen**



**CANopen**®

Ihr Partner für Standard- und Sonderausführungen  
– präzise, zuverlässig und schnell –

Hohner Elektrotechnik GmbH  
Gewerbehof 1  
59368 Werne  
Phone +49 - 2389 - 9878-0  
Fax +49 - 2389 - 9878-27  
E-Mail [info@hohner-elektrotechnik.de](mailto:info@hohner-elektrotechnik.de)  
Web [www.hohner-elektrotechnik.de](http://www.hohner-elektrotechnik.de)

### Inhalt

1	Definitionen	6
2	Sicherheits- und Betriebshinweise	8
3	Allgemeine Angaben	9
3.1	Einführung	9
3.2	Anwendungsbereich	9
3.3	CANopen-Kommunikationsmodell und Profile	10
3.4	Das CANopen-Profil	10
3.5	Das Geber Geräteprofil (CiA DSP 406)	11
4	Datenübertragung	12
4.1	COB-Identifizier	12
4.1.1	Knotennummer	13
4.1.2	Nachrichtenobjekte und Funktionscodes	13
4.2	Übertragung der Prozess Daten	14
4.3	Übertragung von Service Daten (Parametrierung)	15
4.3.1	SDO-COB-ID	16
4.3.2	Kommando	16
4.3.3	Beispiel für Übertragung von Service-Daten	17
5	Objektverzeichnis	18
5.1	AC58-Einträge im Objektverzeichnis	20
5.1.1	Überblick	20
5.1.2	Einzelbeschreibung der Kommunikations-Parameter	21
5.1.2.1	Objekt 1000h: Gerätetyp	21
5.1.2.2	Objekt 1001h: Fehlerregister	22
5.1.2.3	Objekt 1003h: Vordefiniertes Fehlerfeld	22
5.1.2.4	Objekt 1004h: Anzahl der unterstützten PDO's	22
5.1.2.5	Objekt 1005h: COB-ID für SYNC-Nachricht	23
5.1.2.6	Objekt 1008h: Hersteller-Gerätename	23
5.1.2.7	Objekt 1009h: Hardware-Version	23
5.1.2.8	Objekt 100Ah: Software-Version	23
5.1.2.9	Objekt 100Bh: Knotennummer	24
5.1.2.10	Objekt 100Ch und 100Dh: Überwachungszeit und Lebensdauer-Faktor	24
5.1.2.11	Objekt 100Eh: COB-ID für Knotenüberwachung	24
5.1.2.12	Objekt 100Fh: Anzahl der unterstützten SDO's	25
5.1.2.13	Objekt 1010h: Parameter abspeichern	25
5.1.2.14	Objekt 1011h: Standard-Werte laden	25
5.1.2.15	Objekt 1014h: COB-ID für Notfallobjekte	26
5.1.2.16	Objekt 1018h: Identity Object	26
5.1.2.17	Objekt 1800h: PDO1- Parameter (asynchron)	27
5.1.2.18	Objekt 1802h: PDO2- Parameter (synchron, zyklisch)	29
5.1.3	Einzelbeschreibung der herstellereigenen Geber- Parameter	30
5.1.3.1	Objekt 2000h: Warnpositionen	30
5.1.3.2	Objekt 2001h: Offset-Wert	30
5.1.3.3	Objekt 2002h: Geschwindigkeit	30
5.1.3.4	Objekt 2003h: Beschleunigung	31
5.1.3.5	Objekt 2004h: Meldeflags der Warnpositionen	31

5.1.3.6	Objekt 2005h: PDO-Typ	32
5.1.4	Einzelbeschreibung der allgemeinen Geber- Parameter	33
5.1.4.1	Objekt 6000h: Betriebsparameter	33
5.1.4.2	Objekt 6001h: Messschritte pro Umdrehung (Auflösung)	33
5.1.4.3	Objekt 6002h: Gesamtanzahl der Messschritte	34
5.1.4.4	Objekt 6003h: Preset-Wert	35
5.1.4.5	Objekt 6004h: Positionswert	35
5.1.4.6	Objekt 6200h: Zyklus-Timer	35
5.1.4.7	Objekt 6500h: Betriebs-Status anzeigen	36
5.1.4.8	Objekt 6501h: phys. Auflösung Singleturn	36
5.1.4.9	Objekt 6502h: Anzahl der Umdrehungen	37
5.1.4.10	Objekt 6503h: Alarmmeldung	37
5.1.4.11	Objekt 6504h: unterstützte Alarmmeldungen	37
5.1.4.12	Objekt 6505h: Warnmeldung	38
5.1.4.13	Objekt 6506h: unterstützte Warnmeldungen	38
5.1.4.14	Objekt 6507h: Profil- und Software-Version	39
5.1.4.15	Objekt 6508h: Betriebszeitähler	39
5.1.4.16	Objekt 6509h: Offset-Wert	39
5.1.4.17	Objekt 650Ah: Hersteller-Offset	39
5.1.4.18	Objekt 650Bh: Seriennummer	39
6	Netzwerkmanagement	40
6.1	Beschreibung der NMT-Kommandos	41
6.2	Knotenüberwachung	42
6.2.1	Allgemeines	42
6.2.2	Beispiel: Überwachung der Knoten mit den Knotennummern 1, 2 und 3	43
6.2.3	Störungserkennung des Knotenüberwachungsprotokoll:	44
6.3	Bus-Synchronisation	44
7	Busanschluss	45
7.1	Allgemeines zum Busanschluss des Gebers	45
7.2	Anschlussarten	45
7.2.1	Kabel, axial	46
7.2.2	Conin, axial	48
7.2.3	Bushaube mit zwei Coninstecker	50
7.2.4	Bushaube mit 3 Kabelverschraubungen	51
7.3	Anschlussbild	53
7.4	Kabelanschluss	53
8	Bedien- und Anzeigeelemente	54
8.1	DIP-Schalter (S1)	54
8.1.1	Einstellung der Knoten-Nummer	55
8.1.2	Einstellung der Baudrate	56
8.1.3	Aktivieren des Busabschlusswiderstandes	57
8.2	LED-Anzeige bei Bushaube	57
8.3	LED-Anzeige bei Flanschdose	58
9	Inbetriebnahme	59
9.1	Versorgungsspannung einschalten	59
9.2	Programmierung des Gebers	59
9.2.1	Übersicht Standard-Werte	60

---

9.2.2	Beispiel für eine Geber-Programmierung:	62
9.3	Betriebszustand herstellen	63
10	Technische Daten	64
10.1	Mechanisch	64
10.2	Elektrisch	65
11	Maßzeichnungen	66
11.1	Synchroflansch	66
11.1.1	Anschluss mit Bushaube	66
11.1.2	Anschluss mit Kappe	67
11.2	Klemmflansch	68
11.2.1	Anschluss mit Bushaube	68
11.2.2	Anschluss mit Kappe	69
11.3	Hohlwelle mit Federblech	70
11.3.1	Anschluss mit Bushaube	70
11.3.2	Anschluss mit Kappe	71
12	Bestellschlüssel	72

## 1 Definitionen

Dieses Technische Handbuch beschreibt die Software, Parametrierung und Inbetriebnahme des Drehgebers.

### 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 zeigt Handlungs-Anweisungen an.

### Verwendete Abkürzungen

<b>CAL</b>	CAN Application Layer. Anwendungsschicht (Schicht 7) im CAN Kommunikations-Modell
<b>CAN</b>	Controller Area Network
<b>CiA</b>	CAN in Automation. Internationaler Verein der Anwender und Hersteller von CAN-Produkten
<b>CMS</b>	CAN Message Specification. Service-Element von CAL
<b>COB</b>	Communication Object. Transporteinheit im CAN Netzwerk (CAN Nachricht). Daten werden innerhalb eines COB's über das Netzwerk gesendet.
<b>COB-ID</b>	COB-Identifizier. Eindeutige Kennung einer CAN-Nachricht. Der Identifizier bestimmt die Priorität des COB's im Netzwerk.
<b>DBT</b>	Distributor. Service-Element von CAL, verantwortlich für die dynamische Vergabe von Identifiern.
<b>DS</b>	Draft Standard; Normentwurf
<b>DSP</b>	Draft Standard Proposal; Normentwurfs-Vorschlag
<b>ID</b>	Identifizier, siehe COB-ID
<b>LMT</b>	Layer Management. Service-Element von CAL, verantwortlich für die Konfiguration der Parameter in den einzelnen Schichten des Kommunikationsmodells.
<b>LSB</b>	Least Significant Bit/Byte; niederwertigstes Bit/Byte
<b>MSB</b>	Most Significant Bit/Byte; höchstwertigstes Bit/Byte <b>NMT</b> Network Management. Service-Element von CAL, verantwortlich für die Initialisierung, Konfiguration und Fehlerbehandlung im Netzwerk.
<b>OSI</b>	Open Systems Interconnection. Schichtenmodell zur Beschreibung der Funktionsbereiche in einem Datenkommunikationssystem.
<b>PDO</b>	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

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

## 2 Sicherheits- und Betriebshinweise

Die Absolut-Drehgeber der Modellreihe ACURO® industry sind nach den anerkannten Regeln der Elektrotechnik hergestellte Qualitätsprodukte. Die Geräte haben das Herstellerwerk in sicherheitstechnisch einwandfreiem Zustand verlassen.

### Deshalb:

- Berücksichtigen sie die technischen Spezifikationen in dieser Dokumentation. Dadurch erhalten sie einen sicherheitstechnisch einwandfreien Zustand und einen störungsfreien Betrieb.
- Nur eine Elektrofachkraft darf elektrische Geräte einbauen und montieren!
- Die Geräte dürfen nur innerhalb der Grenzwerte betrieben werden. (siehe technische Daten)
- Die maximalen Betriebsspannungen dürfen nicht überschritten werden!  
Die Geräte sind nach DIN EN 61010 Teil1, Schutzklasse III gebaut.  
Sie müssen zur Verhinderung von gefährlichen Körperströmen mit Sicherheitskleinspannung (SELV) betrieben werden. Außerdem müssen sie sich in einem Bereich mit Potentialausgleich befinden.
- Verwenden Sie zum Schutz eine externe Sicherung (siehe Elektrische Daten).
- Anwendungsbereich: industrielle Prozesse und Steuerungen.  
Begrenzen sie Überspannungen an den Anschlussklemmen auf Werte der Überspannungskategorie II.
- Sie müssen vermeiden, dass Schocks auf das Gehäuse und vor allem auf die Geberwelle einwirken. Ebenso müssen sie vermeiden, dass die Geberwelle axial und radial überbelastet wird.
- Verwenden Sie nur eine geeignete Kupplung um die maximale Genauigkeit und Lebensdauer der Geber zu garantieren.
- Die guten EMV-Werte gelten nur in Verbindung mit den serienmäßig gelieferten Kabeln und Steckern. Bei geschirmten Kabeln müssen sie den Schirm beidseitig und großflächig mit Erde verbinden. Auch die Leitungen zur Spannungsversorgung müssen vollständig geschirmt sein. Ist dies nicht möglich, müssen sie entsprechende Filtermaßnahmen ergreifen.
- Die Einbauumgebung und Verkabelung hat maßgeblichen Einfluss auf die EMV des Gebers. Der Installateur muss deshalb die EMV der gesamten Anlage (Gerät) sicherstellen.
- Bei der Installation in elektrostatisch gefährdeten Bereichen müssen sie darauf achten Stecker und anzuschließendes Kabel vor ESD zu schützen.
- Als Bus- bzw. Anschlusskabel darf man nur paarig verseilte Signalkabel (CAN+ mit CAN-, +UB mit 0 V) mit Abschirmung verwenden.



## 3 Allgemeine Angaben

### 3.1 Einführung

Der ACURO industry ist ein absoluter Drehgeber (Encoder, Winkelcodierer), der in der hier beschriebenen Version über das Übertragungsmedium "CAN-Bus" (physikalisch: verdrehte und abgeschirmte 2-Draht-Leitung) seine aktuelle Position an einen anderen Busteilnehmer sendet. Das ursprünglich von Bosch/Intel für Anwendungen im Automobil entwickelte serielle Bussystem CAN (Controller Area Network) setzt sich auch in der industriellen Automatisierungstechnik immer mehr durch. Es ist multimasterfähig, d. h. mehrere CAN-Teilnehmer können gleichzeitig den Bus anfordern. Dabei setzt sich die Nachricht mit der höheren Priorität (festgelegt durch den Identifier) ohne Zeitverlust durch.

Der Datentransfer wird durch die Priorität der Nachricht geregelt. Es gibt bei CAN keine Teilnehmeradressen sondern Nachrichten-Identifier. Die Nachricht, die gesendet wird, kann von allen Teilnehmern gleichzeitig empfangen werden (Broadcast). Durch eine Akzeptanzfilterung übernimmt der Teilnehmer nur die für ihn bestimmten Nachrichten. Kriterium für diese Entscheidung ist der Identifier, der mit jeder Nachricht übertragen wird.

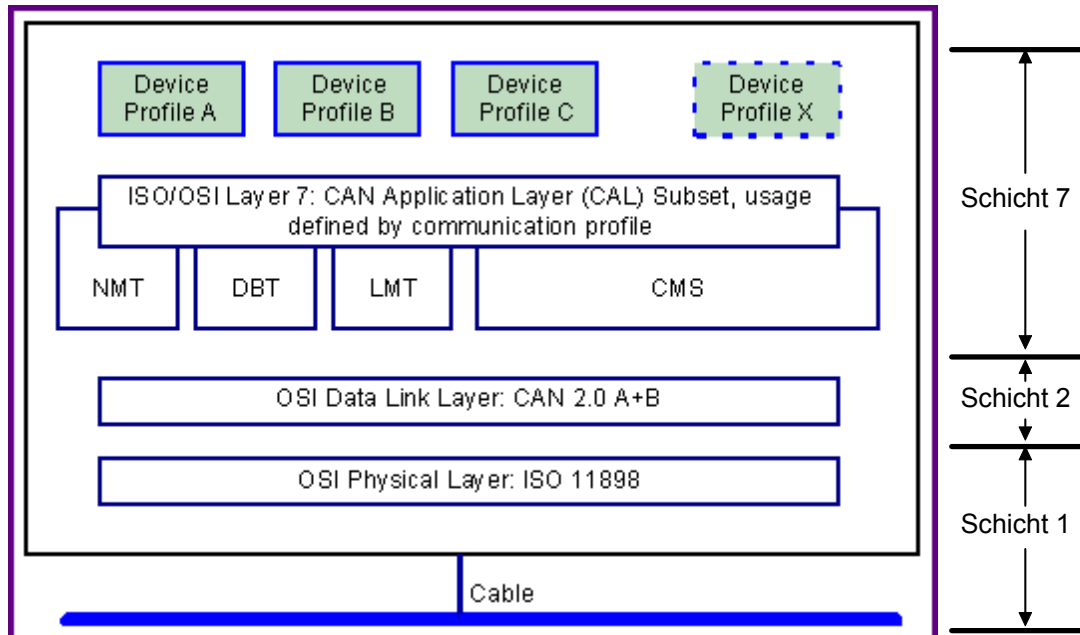
Die Busanbindung ist international genormt nach ISO-DIS 11898 (CAN High Speed) und erlaubt Datenübertragungsraten bis zu 1 Mbit/s. Das herausragendste Merkmal des CAN-Protokolls ist dessen hohe Übertragungssicherheit (Hamming Distanz = 6).

Der verwendete CAN-Controller Intel 82527 ist sowohl **Basic**- als auch **Full-CAN** tauglich und unterstützt die **CAN-Spezifikation 2.0 Part B (Standard-Protokoll mit 11-Bit-Identifier)** sowie **Extended-Protokoll mit 29-Bit-Identifier**. Bisher werden bei CANopen jedoch immer nur 11-Bit-Identifier verwendet.

### 3.2 Anwendungsbereich

In Systemen, in denen die Position eines Antriebes bzw. eines anderen Maschinenteiles erfasst und an eine Steuerung gemeldet werden muss übernimmt der ACURO industry diese Aufgabe. Der ACURO industry löst damit z.B. Positionieraufgaben, indem er die aktuelle Antriebsposition über das CANopen an die Positioniereinheit zurückmeldet.

### 3.3 CANopen-Kommunikationsmodell und Profile



Schicht 1 (Physical Layer):	ISO-DIS 11898 (CAN High Speed)
Schicht 2 (Data Link Layer):	ISO-DIS 11898 (CAN High Speed)
Schicht 7 (Application Layer):	CiA DS 301 (CANopen CAL-based Communication Profile) +Geräteprofile CiA DS 4xx (CANopen Device Profile for xx)

Für folgende Geräte existieren bereits Profile:

- CiA Draft Standard Proposal 401 for Input/Output Modules
- CiA Draft Standard Proposal 402 for Drives and Motion Control
- CiA Work Item 403 for Human Machine Interfaces
- CiA Work Draft 404 for Closed-Loop Controllers and Transformers
- CiA Work Item 405 for IEC-1131 Interfaces
- **CiA Draft Standard Proposal 406 for Encoders**
- CiA Work Item 407 for Public Transport
- CiA Work Item 408 for Fork-Lifts

### 3.4 Das CANopen-Profil

Anwender und Hersteller von CAN-Produkten, wurde mit CANopen und den zugehörigen Geräteprofilen der Etwa 2 ½ Jahre nach Verabschiedung des CAN-Application Layer (CAL) durch CiA, die Vereinigung der Weg zum Aufbau offener Systeme geebnet.

CANopen wurde unter technischer Leitung des Steinbeis Transferzentrums für Automatisierung (STA Reutlingen) auf der Basis der Schicht 7 Spezifikation CAL entwickelt.

Im Vergleich zu CAL sind in CANopen nur die für diesen Einsatz geeigneten Funktionen enthalten.

CANopen stellt somit eine für die Anwendung optimierte Teilmenge von CAN dar und ermöglicht dadurch vereinfachten Systemaufbau und vereinfachte Geräte.

CANopen ist optimiert für den schnellen Datenaustausch in Echtzeitsystemen und über verschiedene Geräteprofile standardisiert.

Der CAN in Automation (CiA) Hersteller- und Anwenderverein ist zuständig für die Erstellung und Normung der entsprechenden Profile.

Der AC58 mit CANopen erfüllt die im Kommunikationsprofil (CiA DS 301) und im Geräteprofil für Geber (CiA DSP 406) festgelegten Forderungen.

CANopen ermöglicht

- Auto-Konfiguration des Netzwerkes,
- komfortablen Zugriff auf alle Geräteparameter.
- Gerätesynchronisation,
- 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)
- Vordefinierte Objekte (für Synchronisation, Zeitstempel, Notfallnachricht)

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 und einen Geräteprofil bezogenen Teil sowie einen herstellereigenen Teil (siehe Kapitel 5).

### 3.5 Das Geber Geräteprofil (CiA DSP 406)

Dieses Profil beschreibt eine herstellereigene 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 herstellereigenes Bussystem.

Das Geräteprofil ist gegliedert in zwei Objekt-Klassen:

- die Standard-Klasse C1 beschreibt alle Grundfunktionen, die der Geber enthalten muss
- die erweiterte Klasse C2 enthält eine Vielzahl von weiteren Funktionen, die von Gebern dieser Klasse entweder unterstützt werden müssen (Mandatory) oder optional sind. Geräte der Klasse C2 enthalten somit alle C1- und C2-mandatory-Funktionen, sowie herstellereigenen weitere optionale Funktionen.

Im Profil ist außerdem ein Adressbereich definiert, der mit herstellereigenen Sonderfunktionen belegt werden kann.

## 4 Datenübertragung

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

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

Die **Prozess-Daten-Objekte (PDO)** dienen dem hochdynamischen Austausch von Echtzeitdaten (z.B. Geberposition) mit maximal 8 Byte Länge. Diese Daten werden mit hoher Priorität (niedriger COB-Identifizier) übertragen. PDO's sind Broadcast-Nachrichten und stellen ihre Echtzeitdaten allen gewünschten Empfängern gleichzeitig zur Verfügung.

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).

**i** Die Priorität der Nachrichten-Objekte wird über den COB-Identifizier festgelegt.

### 4.1 COB-Identifizier

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

Der 11 Bit Identifizier setzt sich aus einem 4 Bit Funktionscode und einer 7 Bit Knotennummer zusammen.

Bit-Nr.	10	9	8	7	6	5	4	3	2	1	0
Art	Funktionscode				Knotennummer						
Belegung <sup>1</sup>	x	x	x	x	0	0	x	x	x	x	x

<sup>1</sup> x = binärer Wert frei wählbar (0 oder 1); 0 = Wert 0 festgelegt

**i** Je höher der Wert des COB-Identifiziers ist, umso niedriger ist dessen Priorität!

### 4.1.1 Knotennummer

Die 7 Bit Knotennummer wird hardwaremäßig über 5 DIP-Schalter auf der Geberrückseite eingestellt (siehe Kap. 8.1.1 Einstellung der Knoten-Nummer). Die 5 DIP-Schalter legen die Bits 0 bis 4 fest, die verbleibenden Bits 5 und 6 sind fest auf 0 eingestellt.

- i** Die Knotennummer 0 ist reserviert und darf von keinem Knoten verwendet werden. Die resultierenden Knotennummern liegen im Bereich 1...31. Die Übernahme einer neuen Knotennummer erfolgt erst beim nächsten Hochlaufen (Reset/Power-on) des Gebers. Alle Identifier werden dann auf ihre Standard-Werte zurückgesetzt.

### 4.1.2 Nachrichtenobjekte und Funktionscodes

Folgende Funktionscodes sind im Predefined Master/Slave Connection Set definiert:

Objekt <sup>1</sup>	Funktionscode binär	Resultierender COB-ID <sup>2</sup>		CMS Prioritäts- gruppe
		hexadezimal	dezimal	
NMT	0000	0	0	0
SYNC <sup>3</sup>	0001	80	128	0
TIME STAMP	0010	100	256	1
EMERGENCY <sup>3</sup>	0001	81 - FF	129 - 255	0, 1
PDO 1 (tx) <sup>3</sup>	0011	181 - 1FF	385 - 511	1, 2
PDO 2 (tx) <sup>3</sup>	0101	281 - 2FF	641 - 767	2, 3
SDO (tx)	1011	581 - 5FF	1409 - 1535	6
SDO (rx)	1100	601 - 67F	1537 - 1663	6, 7
Nodeguard	1110	701 - 77F	1793 - 1919	-

<sup>1</sup> NMT, SYNC, TIME STAMP sind Broadcast Objekte, die an alle Knoten gerichtet sind; alle anderen sind Peer-to-Peer Objekte, die nur vom adressierten Knoten angenommen werden

<sup>2</sup> COB-ID = Funktionscode • 128 + Knotennummer (DIP-Schalter)  
Die Standard-Identifizier können jedoch über SDO-Zugriff auf die entsprechenden Objektverzeichnis-Einträge geändert werden.

<sup>3</sup> Parameter können nach Änderung über den Bus durch das Objekt 1010h im nichtflüchtigen Drehgeber-Speicher (EEPROM) abgespeichert werden

## 4.2 Übertragung der Prozess Daten

Es stehen die zwei PDO-Dienste PDO1 (tx) und PDO2 (tx) zur Verfügung.

Eine PDO-Übertragung kann durch verschiedene Ereignisse initiiert werden (siehe Objektverzeichnis Index 1800h auf Seite 26):

- asynchron (Ereignisgesteuert) durch internen Gerätetimer oder durch Prozesswert-Änderung
- 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 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 folgenden Aufbau:

COB-ID	Prozess Daten im Binärcode							
11 Bit	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
Default:	Positionswert							
Typ1:	Positionswert				Flags <sup>1)</sup>			
Typ2:	Positionswert				Geschwindigkeit <sup>2)</sup>		Beschleunigung <sup>3)</sup>	

<sup>1)</sup> Meldeflags der Warnpositionen, siehe auch Objektverzeichnis Index 2004h

<sup>2)</sup> siehe Objektverzeichnis Index 2002h

<sup>3)</sup> siehe Objektverzeichnis Index 2003h



Der PDO-Typ (Default, Typ1, Typ2) wird im Objektverzeichnis unter **Index 2005h** eingestellt.



Der COB-ID und die Übertragungsart für **PDO1** wird im Objektverzeichnis **Index 1800h** festgelegt (siehe Seite 26). Die Einstellung kann hier über SDO-Zugriff verändert werden.

### Standardeinstellungen:

Freigabe: PDO **aktiv** (enabled)  
 COB-ID: **180h+Knotennummer**  
 Übertragungsart: **254** = Asynchron, d.h. die Prozesswert-Ausgabe wird vom Drehgeber initiiert (Ereignisgesteuert durch internen Geräte-Timer oder durch Prozesswert-Änderung). Timer hat Priorität falls unter Index 6200 eine Zykluszeit >0 definiert wurde. Zusätzlich ist die Anforderung über RTR möglich.

**i** Der COB-ID und die Übertragungsart für **PDO2** wird im Objektverzeichnis **Index 1802h** festgelegt (siehe Seite 29). Die Einstellung kann hier über SDO-Zugriff verändert werden.

**Standardeinstellungen:**

Freigabe: PDO **inaktiv** (disabled)  
 COB-ID: **280h+Knotennummer**  
 Übertragungsart: **1** = Synchron, d.h. die Prozesswert-Ausgabe wird durch SYNC- Telegramme Initiiert. Zusätzlich ist die Anforderung über RTR möglich.

### 4.3 Übertragung von Service Daten (Parametrierung)

Über eine SDO-Nachricht kann auf das Objektverzeichnis des Drehgebers zugegriffen werden. Alle Geräteparameter sind in diesem Objektverzeichnis (siehe Kapitel 5) unter genormten Adressen (Index) abgelegt, und können über SDO's beschrieben oder gelesen werden. SDO's werden im Handshake-Verfahren (Anforderung und Bestätigung) zwischen zwei Teilnehmern (Konfigurationsmaster und Drehgeber) ausgetauscht.

Es stehen zwei SDO-Dienste zur Verfügung:

- ein Sende-SDO (tx) für Nachrichten (Parameter oder Bestätigung) vom Drehgeber zum Master,
  - ein Empfangs-SDO (rx) für Nachrichten (Anforderung oder Parameter) vom Master zum Drehgeber.
- SDO-Nachrichten haben folgenden Aufbau (hier: Expedited Protocol für bis zu 4 Byte Daten):

COB-ID	Kommando	Index		Subindex	Service Daten (Parameter)			
11 Bit	Byte 0	Byte 1 (LSB)	Byte 2 (MSB)	Byte 3	Byte 4 (LSB)	Byte 5	Byte 6	Byte 7 (MSB)

Service Daten die länger als 4 Byte sind (Objekte 1008h, 1009h, 100Ah) werden über das Segmented Protocol (Datensegmentierung) übertragen.

Die Bedeutung von Index, Sub-Index und Daten finden Sie im Kapitel 5 "Objektverzeichnis".

### 4.3.1 SDO-COB-ID

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

- i** SDO (tx) (Geber→Master): **580h** (1408) + Knotennummer
  - SDO (rx) (Master→Geber): **600h** (1536) + Knotennummer
- Die SDO-Identifier können nicht verändert werden!**

### 4.3.2 Kommando

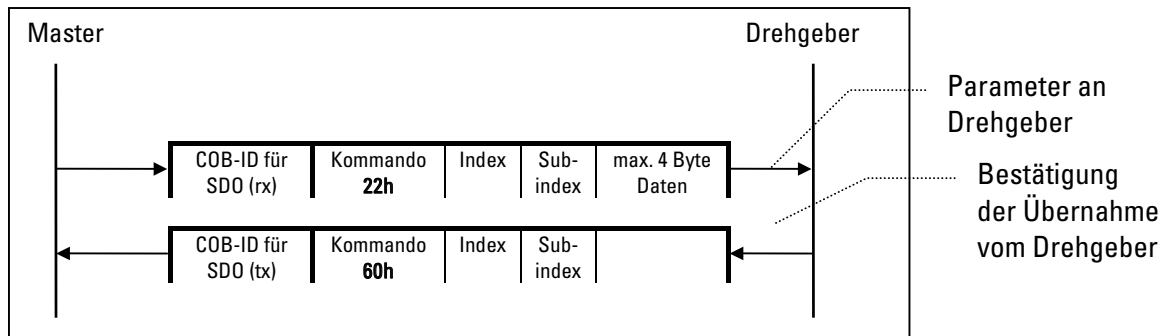
Das Kommando-Byte beschreibt die Art der SDO-Nachricht:

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

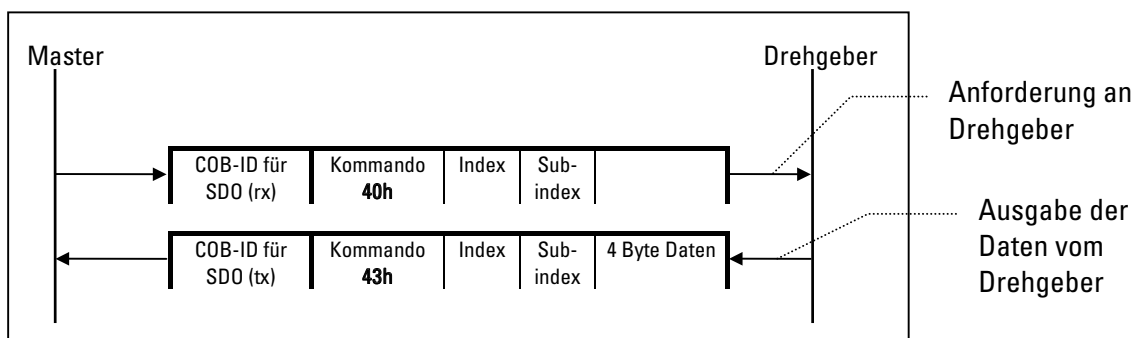
- i** • Eine Fehlermeldung (Kommando 80h) ersetzt im Fehlerfall die normale Bestätigung (Response).
  - Die Fehlermeldung umfasst sowohl Kommunikations-Protokoll-Fehler (z.B. falsches Kommando-Byte) als auch Objektverzeichnis-Zugriffsfehler (z.B. falscher Index, Schreibversuch auf Read-Only-Objekt, falsche Datenlänge u.s.w.).
- Die Fehlercodes sind im CANopen-Profil (DS 301) bzw. im Geräteprofil (DSP 406) beschrieben.



### 4.3.3 Beispiel für Übertragung von Service-Daten



Master überträgt Parameter an Drehgeber



Master fordert Parameter vom Drehgeber an

## 5 Objektverzeichnis

Im Objektverzeichnis eines CANopen-Gerätes sind alle Eigenschaften und Parameter dieses Gerätes abgelegt.

**i** Die Daten des Objektverzeichnisses befinden sich spannungsausfallsicher im EEPROM des Gebers und werden bei Power-on oder Reset in den Arbeitsspeicher (RAM) kopiert. Werden nun Daten im Objektverzeichnis geändert, so wird die Änderung nur im Arbeitsspeicher durchgeführt. Sollen die Daten jedoch dauerhaft gesichert werden, so müssen diese unbedingt über das Objekt 1010h (Parameter speichern) in das EEPROM übernommen werden.



**Die vorher im EEPROM vorhandenen Daten werden dadurch überschrieben!**

Der Zugriff auf das Objektverzeichnis (Schreiben oder Lesen von Parametern) erfolgt über die in Kapitel 4.3 "Übertragung von Service Daten (Parametrierung)" beschriebenen SDO-Dienste.

Das Objektverzeichnis ist gegliedert in verschiedene Bereiche:

- Eigenschaften, die für alle CANopen-Geräte gültig sind (DS 301)
- Eigenschaften, die für alle Geber gültig sind (Geräteprofil DSP 406)
- Eigenschaften, die herstellerspezifisch sind

Die Adresse (Index), die auf jeden Eintrag im Objektverzeichnis zeigt, ist ebenfalls mit Ausnahme des herstellerspezifischen Bereichs in den Profilen genormt. So ist sichergestellt, dass alle Geräte die im Profil beschriebenen Funktionen (Standard- und optionale Funktionen) immer unter dem gleichen Index liefern. Dies ist Voraussetzung für ein offenes System und für die Austauschbarkeit der Geräte.

Die Einträge im Objektverzeichnis werden durch einen 16 Bit Index adressiert. Jeder Index-Eintrag kann durch einen Sub-Index weiter untergliedert werden.

**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  
 (Die M/O-Angaben in den folgenden Kapiteln beziehen sich auf einen Geber der Klasse C2 entsprechend dem Geräteprofil DSP 406)

**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

## 5.1 AC58-Einträge im Objektverzeichnis

### 5.1.1 Überblick

Im Objektverzeichnis des AC58 sind folgende Einträge gemäß CANopen-Profil DS 301 bzw. Geräteprofil DSP 406 realisiert:

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

#### Kommunikations-Parameter:

1000	VAR	Gerätetyp	Unsigned32	ro	M	21
1001	VAR	Fehlerregister	Unsigned8	ro	M	22
1003	ARRAY	vordefiniertes Fehlerfeld	Unsigned32	ro	0	22
1004	ARRAY	Anzahl der unterstützten PDO's	Unsigned32	ro	0	22
1005 <sup>1</sup>	VAR	COB-ID für SYNC-Nachricht	Unsigned32	rw	0	23
1008	VAR	Hersteller-Gerätename	Vis-String	ro	0	23
1009	VAR	Hardware-Version	Vis-String	ro	0	23
100A	VAR	Software-Version	Vis-String	ro	0	23
100B	VAR	Knotennummer	Unsigned32	ro	0	24
100C	VAR	Überwachungszeit	Unsigned16	rw	0	24
100D	VAR	Lebensdauer-Faktor	Unsigned8	rw	0	24
100E	VAR	COB-ID für Knotenüberwachung	Unsigned32	rw	0	24
100F	VAR	Anzahl der unterstützten SDO's	Unsigned32	ro	0	25
1010	ARRAY	Parameter abspeichern	Unsigned32	rw	0	25
1011	ARRAY	Standardwerte laden	Unsigned32	rw	0	25
1014 <sup>1</sup>	VAR	COB-ID für Notfallobjekte	Unsigned32	rw	0	25
1800 <sup>2</sup>	RECORD	PDO1- Parameter		rw	M	26
1802 <sup>2</sup>	RECORD	PDO2- Parameter		rw	M	29

#### Geber-Parameter (herstellerspezifisch):

2000 <sup>1</sup>	ARRAY	Warnpositionen	Unsigned32	rw	0	30
2001 <sup>1</sup>	VAR	Offset-Wert	Signed32	rw	0	30
2002	VAR	Geschwindigkeit	Signed16	rw	0	30
2003	VAR	Beschleunigung	Signed16	rw	0	31
2004	VAR	Meldeflags der Warnpositionen	Unsigned8	ro	0	31
2005 <sup>1</sup>	VAR	PDO-Typ	Unsigned8	rw	0	32

### Geber-Parameter (allg.):

6000 <sup>1</sup>	VAR	Betriebsparameter	Unsigned16	rw	M/O	33
6001 <sup>1</sup>	VAR	Messschritte pro Umdrehung	Unsigned32	rw	M	33
6002 <sup>1</sup>	VAR	Gesamtanzahl der Messschritte	Unsigned32	rw	M	34
6003 <sup>1</sup>	VAR	Preset-Wert	Unsigned32	rw	M	35
6004	VAR	Positionswert	Unsigned32	ro	M	35
6200 <sup>1</sup>	VAR	Zyklus-Timer	Unsigned16	rw	M	35

### Geber-Diagnoseparameter:

6500	VAR	Betriebs-Status	Unsigned16	ro	M	36
6501	VAR	phys. Auflösung Singleturn	Unsigned32	ro	M	36
6502	VAR	phys. Anzahl der Umdrehungen	Unsigned16	ro	M	37
6503	VAR	Alarmmeldung	Unsigned16	ro	M	37
6504	VAR	unterstützte Alarmmeldungen	Unsigned16	ro	M	37
6505	VAR	Warnmeldung	Unsigned16	ro	M	37
6506	VAR	unterstützte Warnmeldungen	Unsigned16	ro	M	38
6507	VAR	Profil- und Software-Version	Unsigned32	ro	M	38
6508	VAR	Betriebszeitähler	Unsigned32	ro	M	39
6509	VAR	Offset-Wert	Signed32	ro	M	39
650A	VAR	Hersteller-Offset	Signed32	ro	M	39
650B	VAR	Seriennummer	Unsigned32	ro	M	39

<sup>1</sup> Parameter können über das Objekt 1010h im nichtflüchtigen Geber-Speicher (EEPROM) abgespeichert werden

<sup>2</sup> Die Sub-Index-Parameter COB-ID und Verbotzeit können über das Objekt 1010h im nichtflüchtigen Geber-Speicher (EEPROM) abgespeichert werden

## 5.1.2 Einzelbeschreibung der Kommunikations-Parameter

### 5.1.2.1 Objekt 1000h: Gerätetyp

Liefert Informationen über das verwendete Geräteprofil und den Gerätetyp.

1000	VAR	Gerätetyp	Unsigned32	ro	M
------	-----	-----------	------------	----	---

#### Dateninhalt:

Geräteprofil Nummer		Gebertyp	
Byte 0 (LSB)	Byte 1	Byte 2	Byte 3 (MSB)
96h*	01h*	01h (absoluter Drehgeber Singleturn) 02h (absoluter Drehgeber Multiturn)	00h

\* 196h = 406 dezimal (Geberprofil)

### 5.1.2.2 Objekt 1001h: Fehlerregister

Auftretende Gerätefehler werden hier angezeigt.

1001	VAR	Fehlerregister	Unsigned8	ro	M
------	-----	----------------	-----------	----	---

#### Dateninhalt:

Bit 0 = 1: allgemeiner Fehler (Drehgeber-Alarmmeldung)

Bit 1...7: unbenutzt

### 5.1.2.3 Objekt 1003h: Vordefiniertes Fehlerfeld

Die auftretenden Gerätefehler werden hier eingetragen. Im Fehlerfeld werden immer die letzten 8 Fehler gespeichert.

1. Der Eintrag unter Sub-Index 0 enthält die Anzahl der gespeicherten Fehler.
2. Jeder neue Fehler wird unter Sub-Index 1 abgespeichert, die bisherigen Einträge rutschen in der Liste um eine Position nach unten.
3. Die gesamte Fehlerliste wird durch das Schreiben des Wertes 0 bei Sub-Index 0 gelöscht.

1003	ARRAY	vordefiniertes Fehlerfeld	Unsigned32	ro	0
------	-------	---------------------------	------------	----	---

### 5.1.2.4 Objekt 1004h: Anzahl der unterstützten PDO's

Das Objekt enthält Informationen über die maximale Anzahl der vom Gerät unterstützten PDO's. Subindex 0h enthält die Gesamtanzahl der Transmit (Byte 0 und 1) und Receive (Byte 2 und 3) PDO's. Subindex 1h enthält die Anzahl der synchronen Transmit (Byte 0 und 1) und Receive (Byte 2 und 3) PDO's.

Subindex 2h enthält die Anzahl der asynchronen Transmit (Byte 0 und 1) und Receive (Byte 2 und 3) PDO's.

1004	ARRAY	Anzahl der unterstützten PDO's	Unsigned32	ro	0
------	-------	--------------------------------	------------	----	---

#### Dateninhalt:

Sub-Index 0h: 2h (2 Transmit-PDO's)

Sub-Index 1h: 1h (1 synchroner Transmit-PDO)

Sub-Index 2h: 1h (1 asynchroner Transmit-PDO)

### 5.1.2.5 Objekt 1005h: COB-ID für SYNC-Nachricht

Das Objekt definiert den COB-ID für die SYNC-Nachricht. Außerdem wird definiert, ob das Gerät ein Erzeuger oder Empfänger von SYNC-Objekten ist.

1005	VAR	COB-ID für SYNC-Nachricht	Unsigned32	rw	0
------	-----	---------------------------	------------	----	---

#### Dateninhalt:

Bit 0...10: 11 Bit Identifier; **Standard-ID = 80h**  
 Bit 11...29: 0 (reserviert für 29 Bit Identifier-Geräte)  
 Bit 30: 0 (Gerät erzeugt keine SYNC-Nachrichten)  
 Bit 31: 1 (Gerät ist Empfänger für SYNC-Nachrichten)

### 5.1.2.6 Objekt 1008h: Hersteller-Gerätename

Enthält den Hersteller-Gerätenamen.

1008	VAR	Hersteller-Gerätename	Vis-String	ro	0
------	-----	-----------------------	------------	----	---

#### Dateninhalt:

"AC58CAN" im ASCII-Code

### 5.1.2.7 Objekt 1009h: Hardware-Version

Enthält die Hardware Versionsnummer.

1009	VAR	Hardware-Version	Vis-String	ro	0
------	-----	------------------	------------	----	---

#### Dateninhalt:

z.B. "HW-V2" im ASCII-Code

### 5.1.2.8 Objekt 100Ah: Software-Version

Enthält die Software Versionsnummer.

100A	VAR	Software-Version	Vis-String	ro	0
------	-----	------------------	------------	----	---

#### Dateninhalt:

z.B. "SV03.00" im ASCII-Code

### 5.1.2.9 Objekt 100Bh: Knotennummer

Das Objekt zeigt die eingestellte Knotennummer an. Der Wert errechnet sich aus dem über die DIP-Schalter eingestellten binären Wert (siehe Kapitel 8.1.1 Einstellung der Knoten-Nummer).

100B	VAR	Knotennummer	Unsigned32	ro	0
------	-----	--------------	------------	----	---

#### Dateninhalt:

Knotennummer im Bereich 1h... 1Fh (1...31 dezimal);

### 5.1.2.10 Objekt 100Ch und 100Dh: Überwachungszeit und Lebensdauer-Faktor

Die Objekte 100Ch, 100Dh und 100Eh definieren die Parameter für die Knotenüberwachung (siehe Kapitel 6.2).

Die Objekte unter Index 100Ch und 100Dh enthalten die Überwachungszeit in Millisekunden und den Lebensdauer-Faktor. Der Lebensdauer-Faktor multipliziert mit der Überwachungszeit ergibt die Lebensdauer für das Knotenüberwachungs-Protokoll.

100C	VAR	Überwachungszeit	Unsigned16	rw	0
100D	VAR	Lebensdauer-Faktor	Unsigned8	rw	0

#### Dateninhalt:

Überwachungszeit 0000...FFFFh [ms]; Standardwert: = 0h  
Lebensdauer-Faktor: 00...FFh; Standardwert = 0h

### 5.1.2.11 Objekt 100Eh: COB-ID für Knotenüberwachung

Die Objekte 100Ch, 100Dh und 100Eh definieren die Parameter für die Knotenüberwachung (siehe Kapitel 6.2).

Das Objekt 100Eh definiert den COB-ID für das Knotenüberwachungs-Protokoll.

100E	VAR	COB-ID für Knotenüberwachung	Unsigned32	rw	0
------	-----	------------------------------	------------	----	---

#### Dateninhalt:

Bit 0...10: 11 Bit Identifier; **Standard-ID = 700h + Knotennummer**  
Bit 11...29: 0 (reserviert für 29 Bit Identifier-Geräte)  
Bit 30, 31: reserviert



### 5.1.2.12 Objekt 100Fh: Anzahl der unterstützten SDO's

Das Objekt enthält Informationen über die Anzahl der vom Gerät unterstützten SDO's. Byte 0 und 1 enthält die Anzahl der Server SDO's, Byte 2 und 3 enthält die Anzahl der Client SDO's

100F	VAR	Anzahl der unterstützten SDO's	Unsigned32	ro	0
------	-----	--------------------------------	------------	----	---

**Dateninhalt:**

00000001h: 1 Server-SDO mit 2 Richtungen (SDO (rx) und SDO (tx))

### 5.1.2.13 Objekt 1010h: Parameter abspeichern

Durch das Schreiben des Kommandos "save" unter **Sub-Index 1h** wird das Abspeichern der Parameter in den nichtflüchtigen Speicher (EEPROM) veranlasst.

Folgende Objekte werden durch dieses Kommando abgespeichert: 1005h, 1014h, 1800h (Sub-Index 1 und 3), 1802h (Sub-Index 1), 2000h, 2001h, 2005h, 6000h, 6001h, 6002h, 6003h, 6200h.

**Objekte, die vom Speicherkommando nicht erfasst werden, müssen nach jedem Reset/Power-ON neu an den Geber geschickt werden!**

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

**i** Die im EEPROM abgespeicherten Werte (Power-ON-Werte) werden durch dieses Kommando unwiderruflich überschrieben!

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

1010	ARRAY	Parameter abspeichern	Unsigned32	rw	0
------	-------	-----------------------	------------	----	---

**Dateninhalt:**

**Schreibzugriff:**

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

**Lesezugriff:**

Bit 0 = 1: Gerät speichert Parameter auf Kommando  
 Bit 1 = 0: Gerät speichert Parameter nicht automatisch  
 Bit 2...31 = 0: reserviert

### 5.1.2.14 Objekt 1011h: Standard-Werte laden

Durch das Schreiben des Kommandos "load" unter **Sub-Index 1h** werden alle Parameter im Geber-RAM 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.

**i** Die Parameter im Arbeitsspeicher werden mit diesem Kommando gelöscht und durch Standardwerte (Herstellerwerte wie bei Auslieferung des Gebers) ersetzt.

Ein Lesezugriff auf den **Sub-Index 1h** liefert die Information, ob ein Laden der Standard-Werte überhaupt möglich ist.

1011	ARRAY	Standard-Werte laden	Unsigned32	rw	0
------	-------	----------------------	------------	----	---

**Dateninhalt:**

**Schreibzugriff:**

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

**Lesezugriff:**

Bit 0 = 1: Gerät unterstützt das Laden von Standard-Werten  
 Bit 1...31 = 0: reserviert

**i** Die Standardwerte werden erst nach einem "Reset Knoten" (siehe Kapitel 6.1) gültig. Wenn die Standardwerte auch in das EEPROM übernommen werden sollen, so muss nach dem "Reset Knoten" noch das Kommando "Parameter abspeichern" (siehe Objekt 1010h) durchgeführt werden.

**5.1.2.15 Objekt 1014h: COB-ID für Notfallobjekte**

Das Objekt definiert den COB-ID für Notfall-Nachrichten.

1014	VAR	COB-ID für Notfallobjekte	Unsigned32	rw	0
------	-----	---------------------------	------------	----	---

**Dateninhalt:**

Bit 0...10: 11 Bit Identifier; **Standard-ID = 80h + Knotennummer**  
 Bit 11...29: 0 (reserviert für 29 Bit Identifier-Geräte)  
 Bit 30, 31: reserviert

**5.1.2.16 Objekt 1018h: Identity Object**

Hiermit kann die Device – Identifikation gelesen werden.

1018	RECORD	Device – Identifikation		ro	M
------	--------	-------------------------	--	----	---

Sub-Index 0h : nur „read“  
 liefert konstant den Wert 4  
 Sub-Index 1h: nur „read“  
 liefert die Vendor-ID (00000008h)  
 Sub-Index 2h: liefert den Product-Code  
 (z.B. 2 für ACURO CANopen)  
 Sub-Index 3h: nur „read“  
 liefert die SW-Revisionsnummer  
 (z.B. 02 00 04 00 → 04. 02)  
 Sub-Index 4h: nur „read“  
 liefert die 8-stellige Seriennummer des Gebers

### 5.1.2.17 Objekt 1800h: PDO1- Parameter (asynchron)

Das Objekt enthält die Parameter für das Prozessdatenobjekt PDO1. Über diesen Dienst werden in der Standardeinstellung die Prozessdaten des Gebers asynchron, ausgelöst durch den internen Zyklus-Timer (Voraussetzung: **Zyklus-Timer** über **Objekt 6200h** eingestellt) durch Prozesswert-Änderung oder durch eine RTR-Anforderung ausgegeben.

Der **PDO-Typ** wird unter **Objekt 2005h** eingestellt.

SUB Index 0h: nur „read“; liefert konstant den Wert „3“

SUB-Index 1h: COB-ID und Freigabe

SUB-Index 2h: Übertragungsart

SUB-Index 3h: Verbotszeit; d.h. Mindest-Wartezeit bevor dieser PDO erneut gesendet werden darf (Einheit :0,1 ms)

Beispiel: Wert 400 entspricht 40 ms Wartezeit

1800	RECORD	PDO1- Parameter		rw	M
------	--------	-----------------	--	----	---

#### Dateninhalt:

Sub-Index 1h: Bit 0...10: 11 Bit Identifier; **Standard-ID = 180h + Knotennummer**  
 (unsigned32) Bit 11...29: 0 (reserviert für 29 Bit Identifier-Geräte)  
 Bit 30: 0 = RTR erlaubt (nicht veränderbar)  
 Bit 31: 0 (PDO enabled), 1 (PDO disabled); **Standard-Wert = 0**

Sub-Index 2h: **Standard-Wert = FEh (254)** (Übertragungsart = asynchron, gesteuert (unsigned8) durch herstellerspezifisches Ereignis)

Sub-Index 3h: **Standard-Wert = 0h** ,keine Verbotszeit (keine Inhibit-Zeit) (unsigned16)

**i Hinweis zu Sub-Index 1h:**  
 Über Bit 31 ist PDO1 standardmäßig **freigegeben** (0=enabled).  
 Diese Einstellung kann geändert werden, muss aber **nach jedem Hochlaufen** (Reset oder Power-on) des Gebers **neu übertragen** werden, da dieses Bit nicht im EEPROM abgespeichert werden kann.

**i Hinweis zu Sub-Index 2h:**  
 Der Standardwert FEh kann geändert werden, muss aber **nach jedem Hochlaufen** (Reset oder Power-on) des Gebers **neu übertragen** werden, da dieses Byte nicht im EEPROM abgespeichert werden kann.

**i Hinweis zu Sub-Index 3h:**  
 Der Standardwert kann geändert werden. Übernahme in den nichtflüchtigen Speicher [EEPROM] mittels Objekt 1010.

### Übersicht der Übertragungsarten:

Code (dez.)	Übertragungsart				
	zyklisch	azyklisch	synchron	asynchron	nur RTR
0		X	X		
1-240	X		X		
241-251	reserviert				
252			X		X
253				X	X
254				X	
255				X	

### Bedeutung der Übertragungsart-Codes:

- 0: nach SYNC, aber nur bei Wertänderung seit dem letzten SYNC
- 1 ... 240: Wert senden nach 1. ... 240. SYNC
- 252: SYNC führt zu interner Werteabspeicherung, Wert muss aber per RTR abgeholt werden
- 253: Wert wird nach RTR aktualisiert und gesendet
- 254: Wert wird aktualisiert und gesendet nach einer Wertänderung (wenn Geräte-Timer = 0) oder nach Ablauf der Zykluszeit (Geräte-Timer ≠ 0)

### **i** Bemerkung zu Übertragungsart 0 ... 253:

Bei Anforderung mit RTR bzw. SYNC wird das PDO nach Abschluss der laufenden Berechnung (ca. 800 µs) gesendet. Dadurch ergibt sich eine "Wartezeit" von 0,2...1 ms.

Das Auslesen der Codescheiben geschieht Timergesteuert jede Millisekunde, anschließend werden die Werte berechnet.

### **i** Falls für das PDO die Übertragungsart 254 verwendet wird (asynchron Ereignisgesteuert):

Die gewählte **Zykluszeit (siehe Objekt 6200h)** muss größer als die Busübertragungsdauer sein, damit die PDO's ungestört 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 (d. h. PDO bei Wertänderung) muss die Baudrate mindestens 125 KBaud betragen.

Bei asynchroner Zyklus-Timer gesteuerten Positionsübertragung kann eine Positionsänderung nicht nur nach Zyklus-Timer-Ablauf übertragen werden, sondern auch sofort:

Positions-Übertragung	Zyklus-Timer-Wert(Objekt 6200H)	Inhibit-Zeit (Objekt 1800, Sub-Index 2)	Funktion
Asynchron	0	0	Max. Ausgabefrequenz bei Wertänderung
Asynchron	0	>0	Ausgabe bei Wertänderung und anschließender Wartezeit (Inhibit-Zeit)
Timer-Zyklisch	>0	0	Zeitzyklische Ausgabe
Asynchron/ Timer-Zyklisch	>0	>0	Kombination Asynchron und Timer-Zyklisch

### 5.1.2.18 Objekt 1802h: PDO2- Parameter (synchron, zyklisch)

Das Objekt enthält die Parameter für das Prozessdatenobjekt PDO 2. Über diesen Dienst werden in der Standardeinstellung die Prozessdaten des Gebers synchron ausgegeben, initiiert durch SYNC- oder RTR-Objekte.

SUB-Index 1h: COB-ID und Freigabe

SUB-Index 2h: Übertragungsart (**Übersicht siehe Objekt 1800h**)

SUB-Index 3h: Verbotszeit, Mindest-Wartezeit bevor dieser PDO erneut gesendet werden darf (Einheit: 0,1 ms)

1802	RECORD	PDO2- Parameter		rw	M
------	--------	-----------------	--	----	---

#### Dateninhalt:

Sub-Index 1h: Bit 0...10: 11 Bit Identifier; **Standard-ID = 280h + Knotennummer**  
(unsigned32) Bit 11...29: 0 (reserviert für 29 Bit Identifier-Geräte)  
Bit 30: 0 = RTR erlaubt (nicht veränderbar)  
Bit 31: 0 (PDO enabled), 1 (PDO disabled); **Standard-Wert = 1**

Sub-Index 2h: **Standard-Wert = 1h** (Übertragungsart = synchron, zyklisch)

(unsigned8) (Übersicht siehe Objekt 1800h)

Sub-Index 3h: **Standard-Wert = 0h** (keine Verbotszeit)

(unsigned16)



#### Hinweis zu Sub-Index 1h:

Über Bit 31 ist PDO2 standardmäßig **gesperrt** (1=disabled). Diese Einstellung kann geändert werden, muss aber **nach jedem Hochlaufen** (Reset oder Power-on) des Gebers **neu übertragen** werden, da dieses Bit nicht im EEPROM abgespeichert werden kann.



#### Hinweis zu Sub-Index 2h:

Der Standardwert 01h kann geändert werden, muss aber **nach jedem Hochlaufen** (Reset oder Power-on) des Gebers **neu übertragen** werden, da dieses Byte nicht im EEPROM abgespeichert werden kann.



#### Hinweis zu Sub-Index 3h:

Der Standardwert 0h kann geändert werden, muss aber **nach jedem Neustart** (Reset oder Power-on) des Gebers **neu geladen** werden, da der Wert nicht im EEPROM abgespeichert werden kann.

### 5.1.3 Einzelbeschreibung der herstellerspezifischen Geber- Parameter

#### 5.1.3.1 Objekt 2000h: Warnpositionen

Es können bis zu 4 Warnpositionen programmiert werden. Je nach Programmierung kann bei Über-/Unterschreitung der Warnposition eine Nachricht über einen Typ1-PDO (Positionswert + Meldeflags) ausgelöst werden (siehe **Objekt 2005h, PDO-Typ**).

Der Zustand der **Meldeflags** kann auch über **Objekt 2004h** abgefragt werden.

Die Warnpositionen sind unter Sub-Index 1, 2, 3 und 4 abgelegt.

Sub-Index 0 nur „read“ (liefert immer den Wert 4).

2000	ARRAY	Warnposition 1...4	Unsigned32	rw	0
------	-------	--------------------	------------	----	---

#### Dateninhalt:

Subindex 1...4:

Bit 0...29: Warnposition im Wertebereich 0 ... programmierte Gesamtauflösung;  
**Standard-Wert = 0h**

Bit 30: Aktivierung bei Unterschreitung der Warnposition; 1=aktiv, 0=inaktiv;  
**Standard-Wert = 0**

Bit 31: Aktivierung bei Überschreitung der Warnposition; 1=aktiv, 0=inaktiv;  
**Standard-Wert = 0**

#### 5.1.3.2 Objekt 2001h: Offset-Wert

Der Offset ermöglicht es den skalierten Wertebereich zu verschieben.

Dieser Offsetwert wird im Geber zum skalierten Positionswert addiert. Damit verschiebt sich der Positionswert um den eingetragenen Offset nach oben oder unten.

2001	VAR	Offset-Wert	Signed32	rw	0
------	-----	-------------	----------	----	---

#### Dateninhalt:

**Standard-Wert = 0h**

#### 5.1.3.3 Objekt 2002h: Geschwindigkeit

Als Maß für die Drehgeschwindigkeit der Geberwelle wird die Wertedifferenz zweier physikalischer (unskalierter) Positionswerte mit Zeitabstand 5 ms ermittelt. Diese Wertedifferenz wird bei einem Lesezugriff mit Objekt 2002h als Vorzeichenbehafteter 16 Bit Wert ausgegeben (positiver Wert = Drehrichtung im Uhrzeigersinn).

Die Ausgabe der Geschwindigkeit mittels PDO wird durch Einstellen des gewünschten PDO's auf Typ 2 (siehe Objekt 2005h) ermöglicht.

2002	VAR	Geschwindigkeit	Signed16	rw	0
------	-----	-----------------	----------	----	---

### 5.1.3.4 Objekt 2003h: Beschleunigung

Als Maß für die Beschleunigung der Geberwelle wird die Wertedifferenz zweier Geschwindigkeitswerte (siehe Objekt 2002h) mit Zeitabstand 5 ms ermittelt. Diese Wertedifferenz wird bei einem Lesezugriff mit Objekt 2003h als Vorzeichenbehafteter 16 Bit Wert ausgegeben.

Die Ausgabe der Beschleunigung mittels PDO wird durch Einstellen des gewünschten PDO's auf Typ 2 (siehe Objekt 2005h) ermöglicht.

2003	VAR	Beschleunigung	Signed16	rw	0
------	-----	----------------	----------	----	---

### 5.1.3.5 Objekt 2004h: Meldeflags der Warnpositionen

Unter diesem Objekt kann der Zustand der Meldeflags für die Warnpositionen (siehe Objekt 2000h) abgefragt werden.

Die Ausgabe der Meldeflags mittels PDO wird durch Einstellen des gewünschten PDO's auf Typ 1 (siehe Objekt 2005h) ermöglicht.

2004	VAR	Meldeflags der Warnpositionen	unsigned8	ro	0
------	-----	-------------------------------	-----------	----	---

#### Dateninhalt:

Bit 0: Warnposition 1 überschritten  
Bit 1: Warnposition 1 unterschritten  
Bit 2: Warnposition 2 überschritten  
Bit 3: Warnposition 2 unterschritten  
Bit 4: Warnposition 3 überschritten  
Bit 5: Warnposition 3 unterschritten  
Bit 6: Warnposition 4 überschritten  
Bit 7: Warnposition 4 unterschritten

Die jeweiligen Bits werden auf 1 gesetzt, wenn die programmierte Warnposition über- bzw. unterschritten wird.

### 5.1.3.6 Objekt 2005h: PDO-Typ

Dieses Objekt dient zur Festlegung des Typs für PDO1 und PDO2.

Der COB-ID und die Übertragungsart für die PDO's wird in den Objekten **1800h** und **1802h** festgelegt.

**i** Über **Bit 31 im Objekt 1800h, Sub-Index 1h** ist PDO1 standardmäßig **freigegeben**.  
 Über **Bit 31 im Objekt 1802h, Sub-Index 1h** ist PDO2 standardmäßig **gesperrt**.  
 Diese Einstellung kann geändert werden, muss aber **nach jedem Hochlaufen** (Reset oder Power-on) des Gebers **neu übertragen** werden, da dieses Bit nicht im EEPROM abgespeichert werden kann.

2005	VAR	PDO-Typ	unsigned8	rw	0
------	-----	---------	-----------	----	---

#### Dateninhalt:

	Bit 7 ... 4 = PDO2	Bit 3 ... 0 = PDO1
00h	Default	Default
01h	Default	Typ1
02h	Default	Typ2
10h	Typ1	Default
11h	Typ1	Typ1
12h	Typ1	Typ2
20h	Typ2	Default
21h	Typ2	Typ1
22h	Typ2	Typ2

Standard: PDO1 und PDO2 auf Typ Default (00h) eingestellt

#### Bedeutung:

COB-ID	Prozess Daten im Binärcode							
	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
Default:	Positionswert							
Typ1:	Positionswert				Flags <sup>1)</sup>			
Typ2:	Positionswert				Geschwindigkeit		Beschleunigung	

<sup>1)</sup> Meldeflags der Warnpositionen, siehe auch Objektverzeichnis Index 2004h.

Bei Übertragungsart 254 (asynchron Ereignisgesteuert; siehe Übersicht bei Objekt 1800h) und Geräte-Timer=0 wird das PDO bei Änderung der Flags gesendet.



## 5.1.4 Einzelbeschreibung der allgemeinen Geber- Parameter

### 5.1.4.1 Objekt 6000h: Betriebsparameter

Aktivierung von: Codefolge umkehren, Diagnoseanforderung, Skalierungsfunktion.

**i** Bei einer Änderung der Codefolge oder bei Aktivierung der Skalierungsfunktion wird der Presetwert (siehe Objekt 6003h) gelöscht.

6000	VAR	Betriebsparameter	Unsigned16	rw	M/O
------	-----	-------------------	------------	----	-----

#### Dateninhalt:

Bit 0: Codefolge; 0 = aufsteigend bei Drehung im Uhrzeigersinn (cw)  
 1 = aufsteigend bei Drehung gegen den Uhrzeigersinn (ccw)

**Standard: Bit = 0**

Bit 1: Hardwareprüfung; 0 = disable, 1 = enable; **Standard: Bit = 0** (siehe Objekt 6503)

Bit 2: Skalierungsfunktion; 0 = disable, 1 = enable; **Standard: Bit = 0** (s. Objekt 6001,6002)

Bit 12: Restwertverrechnung; 0 = disable, 1 = enable; **Standard: Bit = 0** (s. Objekt 6002);

Eine Änderung des Bits wird erst nach einem Geber-Reset wirksam

Bit 3...11, 13..15: unbenutzt (0)

### 5.1.4.2 Objekt 6001h: Messschritte pro Umdrehung (Auflösung)

Dieser Parameter stellt die gewünschte Auflösung pro Umdrehung an. Der Geber berechnet sich intern den entsprechenden Skalierungsfaktor.

**i** Die Gesamtanzahl der Messschritte wird durch Objekt 6002h eingestellt.

Bei einer Änderung der Auflösung wird ein eventuell programmierter Presetwert (siehe Objekt 6003h) gelöscht.

Der resultierende Skalierungsfaktor SKF (mit dem der physikalische Positionswert multipliziert wird) berechnet sich nach folgender Formel:

$$SKF = \frac{\text{Meßschritte pro Umdrehung (6001h)}}{\text{phys. Auflösung Singleturn (6501h)}}$$

6001	VAR	Messschritte pro Umdrehung	Unsigned32	rw	M
------	-----	----------------------------	------------	----	---

#### 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 ... (max. physikalische Auflösung pro Umdrehung)

**Standard-Wert = physikalische Auflösung pro Umdrehung**

**Beispiel** mit Standard-Wert:

AC58/1213 (Auflösung = 13 Bit pro Umdrehung):

Dateninhalt = 20 00h

Byte 0	Byte 1	Byte 2	Byte 3
00	20h	00h	00h

### 5.1.4.3 Objekt 6002h: Gesamtanzahl der Messschritte

Dieser Parameter stellt die Gesamtanzahl der Messschritte ein. Nach der angegebenen Gesamtanzahl der Messschritte steht der Geber wieder auf Null.



**Die Messschritte pro Umdrehung (Auflösung) werden durch Objekt 6001h eingestellt.**

**Bei einer Änderung der Gesamtanzahl der Messschritte wird ein eventuell programmierter Presetwert (siehe Objekt 6003h) gelöscht.**

**Bei Betrieb ohne Restwertverrechnung (Bit 12 von Objekt 6000h = 0):**

Wird der Geber im Endlosbetrieb benutzt, darf die "Gesamtanzahl der Messschritte" nur  $2^x \cdot$  "Messschritte pro Umdrehung (6001h)" betragen (mit  $x=1 \dots 12$ ). Andernfalls tritt immer beim physikalischen Nulldurchgang der Codescheibe(n) ein Sprung im Ausgangscode auf (bei Singleturn nach jeder Umdrehung, bei Multiturn nach 4096 Umdrehungen).

**Bei Betrieb mit Restwertverrechnung (Bit 12 von Objekt 6000h = 1):**

Es können beliebige Werte eingegeben werden, da bei diesem Modus der Restwert bei jedem physikalischen Nulldurchgang (s. o.) automatisch in das Geber-EEPROM abgespeichert wird. **Es sind insgesamt 1.000.000 Schreibzyklen in das EEPROM möglich!**

6002	VAR	Gesamtanzahl der Messschritte	Unsigned32	rw	M
------	-----	-------------------------------	------------	----	---

**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 ... (max. physikalische Gesamtauflösung)

**Standard-Wert = physikalische Gesamtauflösung**

Beispiel mit Standard-Wert:

AC58/1213 (Gesamtauflösung = 13 Bit pro Umdrehung • 12 Bit Umdrehungen): Dateninhalt = 2 00 00 00h

Byte 0	Byte 1	Byte 2	Byte 3
00h	00h	00h	02h

### 5.1.4.4 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.

**i** Bei einer Änderung der Codefolge oder bei Aktivierung/Änderung der Skalierung wird der Presetwert gelöscht (siehe Objekte 6000h, 6001h und 6002h).

Im Geber wird der Preset-Wert in einen entsprechenden Offset-Wert umgerechnet und zum Positionswert addiert (Offset = Preset - Position). Dieser Offset-Wert kann über Objekt 6509h gelesen werden.

6003	VAR	Preset-Wert	Unsigned32	rw	M
------	-----	-------------	------------	----	---

#### 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 ... (programmierte Gesamtauflösung)

**Standard-Wert = 0**

**Durch Schreiben des Wertes FFFF FFFFh wird der Presetwert gelöscht!**

### 5.1.4.5 Objekt 6004h: Positionswert

Der Geber gibt den aktuellen (mit Skalierungsfaktor, Preset und Offset verrechneten) Positionswert aus.

6004	VAR	Positionswert	Unsigned32	ro	M
------	-----	---------------	------------	----	---

#### Dateninhalt:

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

### 5.1.4.6 Objekt 6200h: Zyklus-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.

6200	VAR	Zyklus-Timer	Unsigned16	rw	M
------	-----	--------------	------------	----	---

#### Dateninhalt:

Wertebereich: 0 ... FFFFh (65535) ergibt Zykluszeit in Millisekunden

**Standard-Wert = 0h**

**i** Falls für das PDO die **Übertragungsart 254** verwendet wird (asynchron Ereignisgesteuert, siehe Objekt 1800h):

Die gewählte Zykluszeit muss größer als die Busübertragungsdauer sein, damit die PDO's ungestört 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 (d. h. PDO bei Wertänderung) muss die Baudrate mindestens 125 KBaud betragen.

#### 5.1.4.7 Objekt 6500h: Betriebs-Status anzeigen

Das Objekt zeigt die über Objekt 6000h programmierten Einstellungen an.

6500	VAR	Betriebs-Status	Unsigned16	ro	M
------	-----	-----------------	------------	----	---

#### Dateninhalt:

siehe Objekt 6000h.

#### 5.1.4.8 Objekt 6501h: phys. Auflösung Singleturn

Das Objekt zeigt die physikalische Auflösung (Anzahl der Positionswerte) pro Umdrehung an. Der Wert ist abhängig vom Gebertyp.

6501	VAR	phys. Auflösung Singleturn	Unsigned32	ro	M
------	-----	----------------------------	------------	----	---

#### Dateninhalt:

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

Beispiel:

AC58/1213 (Auflösung = 13 Bit pro Umdrehung): Dateninhalt = 20 00h

Byte 0	Byte 1	Byte 2	Byte 3
00	20h	00h	00h

#### 5.1.4.9 Objekt 6502h: Anzahl der Umdrehungen

Das Objekt zeigt die Anzahl der Umdrehungen an, die der Multiturn-Geber aufnehmen kann. Der Wert ist abhängig vom Gebertyp.

6502	VAR	Anzahl der Umdrehungen	Unsigned16	ro	M
------	-----	------------------------	------------	----	---

#### Dateninhalt:

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

Beispiel:

AC58/1213 (12 Bit für Umdrehungen = 4096): Dateninhalt = 10 00h

Byte 0	Byte 1
00	10h

#### 5.1.4.10 Objekt 6503h: Alarmmeldung

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.

6503	VAR	Alarmmeldung	Unsigned16	ro	M
------	-----	--------------	------------	----	---

#### Dateninhalt:

Bit 0: Positionsfehler; 0 = Positionswert gültig, 1 = Positionsfehler

Bit 1: Hardwareprüfung; 0 = kein Fehler, 1 = Fehler

Bit 2...15: unbenutzt

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

#### 5.1.4.11 Objekt 6504h: unterstützte Alarmmeldungen

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

6504	VAR	unterstützte Alarmmeldungen	Unsigned16	ro	M
------	-----	-----------------------------	------------	----	---

#### Dateninhalt:

Bit 0: 1 = Positionsfehler wird unterstützt

Bit 1: 1 = Hardwareprüfung wird unterstützt

Bit 2...15: unbenutzt

#### 5.1.4.12 Objekt 6505h: Warnmeldung

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 anliegt.

6505	VAR	Warnmeldung	Unsigned16	ro	M
------	-----	-------------	------------	----	---

##### Dateninhalt:

Bit 0: Drehzahl-Überschreitung; 0 = keine, 1 = Überschreitung von 10.000 U/min  
 Bit 1: unbenutzt  
 Bit 2: CPU Watchdog-Status; 0 = in Ordnung, 1 = Reset durchgeführt  
 Bit 3: Betriebszeit; 0 = nicht überschritten, 1 = 100.000 Stunden überschritten  
 Bit 4...15: unbenutzt

Bei aktivem Bit 0 wird gleichzeitig eine Notfall-Nachricht (ID=80h+Knotennummer) mit dem Fehlercode FF00h (Device specific) gesendet.

Bei aktivem Bit 2 oder 3 wird gleichzeitig eine Notfall-Nachricht (ID=80h+Knotennummer) mit dem Fehlercode 5000h (Device hardware) gesendet.

#### 5.1.4.13 Objekt 6506h: unterstützte Warnmeldungen

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

6506	VAR	unterstützte Warnmeldungen	Unsigned16	ro	M
------	-----	----------------------------	------------	----	---

##### Dateninhalt:

Bit 0: 1 = Drehzahl-Überschreitung wird unterstützt  
 Bit 1: unbenutzt  
 Bit 2: 1 = CPU Watchdog-Status wird unterstützt  
 Bit 3: 1 = Betriebszeitwarnung wird unterstützt  
 Bit 4...15: unbenutzt

### 5.1.4.14 Objekt 6507h: Profil- und Software-Version

In den ersten 16 Bit ist die Versionsnummer des hier angewendeten Geberprofils abgelegt. Die zweiten 16 Bit enthalten die Nummer der im Geber implementierten Software-Version.

6507	VAR	Profil- und Software-Version	Unsigned32	ro	M
------	-----	------------------------------	------------	----	---

**Dateninhalt:**

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

Beispiel:

Profil-Version 1.0 und Software-Version 1.1:

Byte 0	Byte 1	Byte 2	Byte 3
00	01h	10h	01h

### 5.1.4.15 Objekt 6508h: Betriebszeitähler

Solange die Geber-Versorgungsspannung anliegt wird alle 6 Minuten (= 0,1 Stunden) der Betriebszeitähler um 1 inkrementiert.

Hinweis: Nach einer Betriebszeit von 100.000 Stunden wird im Objekt 6505h (Warnmeldung) das Bit Nr. 3 gesetzt.

6508	VAR	Betriebszeitähler	Unsigned32	ro	M
------	-----	-------------------	------------	----	---

**Dateninhalt:**

Betriebszeit in 1/10 Stunden als binärer 32 Bit Wert.

### 5.1.4.16 Objekt 6509h: Offset-Wert

Ein über Objekt 6003h eingegebener Presetwert wird intern in einen entsprechenden Offsetwert umgerechnet. Das Objekt 6509h zeigt diesen errechneten Offset-Wert an.

6509	VAR	Offset-Wert	Signed32	ro	M
------	-----	-------------	----------	----	---

### 5.1.4.17 Objekt 650Ah: Hersteller-Offset

Enthält den über Objekt 2001h festgelegten Offsetwert

650A	VAR	Hersteller-Offset	Signed32	ro	M
------	-----	-------------------	----------	----	---

### 5.1.4.18 Objekt 650Bh: Seriennummer

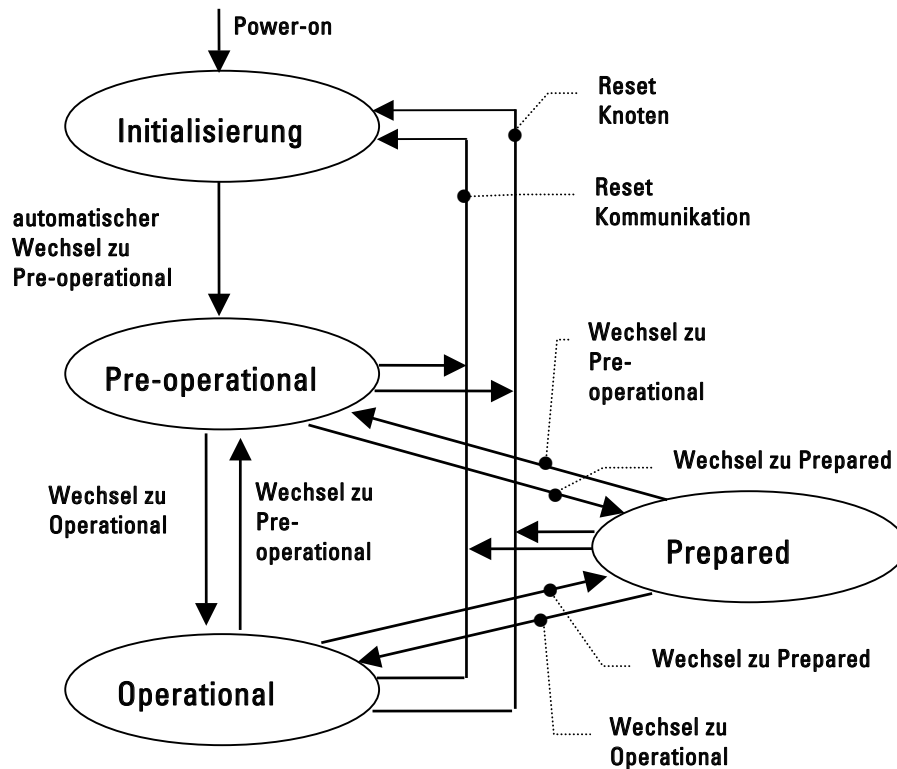
Dieses Objekt liefert die Seriennummer des Gebers.

650B	VAR	Seriennummer	Unsigned32	ro	M
------	-----	--------------	------------	----	---

## 6 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:** Ausgangszustand nach Anlegen der Versorgungsspannung. Der Knoten wechselt nach Durchlauf der Reset-/Initialisierungsroutinen automatisch in den Zustand Pre-operational.

**Pre-operational:** Der Knoten kann nun über SDO-Nachrichten unter dem Standard-Identifizier angesprochen werden. Der Knoten kann somit über sein Objektverzeichnis parametrisiert werden, z.B. Programmierung der Geber- oder Kommunikations-Parameter.

**Operational:** Der Knoten ist aktiv. Prozesswerte werden über die PDO's ausgegeben.

**Prepared:** 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.



## 6.1 Beschreibung der NMT-Kommandos

Die Kommandos werden als unbestätigtes NMT-Objekt übertragen.  
Das NMT-Objekt ist folgendermaßen aufgebaut:

	Byte 0	Byte 1
<b>COB-ID = 0</b>	<b>Kommandobyte</b>	<b>Knoten-Nummer</b>

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.

Folgende Tabelle gibt einen Überblick über die verwendeten Kommandos:

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 <sup>1</sup>
82h	Reset_Communication: Reset Kommunikation <sup>2</sup>

<sup>1</sup> Alle Parameter des gesamten Objektverzeichnisses werden auf Power-On Werte gesetzt.

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

## 6.2 Knotenüberwachung

### 6.2.1 Allgemeines

Die Knotenüberwachung wird verwendet, um die Kommunikationsfähigkeit der Teilnehmer zu überwachen.

Die Knotenüberwachung sollte immer dann verwendet werden, wenn der Teilnehmer nur in unregelmäßigen Abständen (Ereignisgesteuert) Daten auf dem Bus sendet.

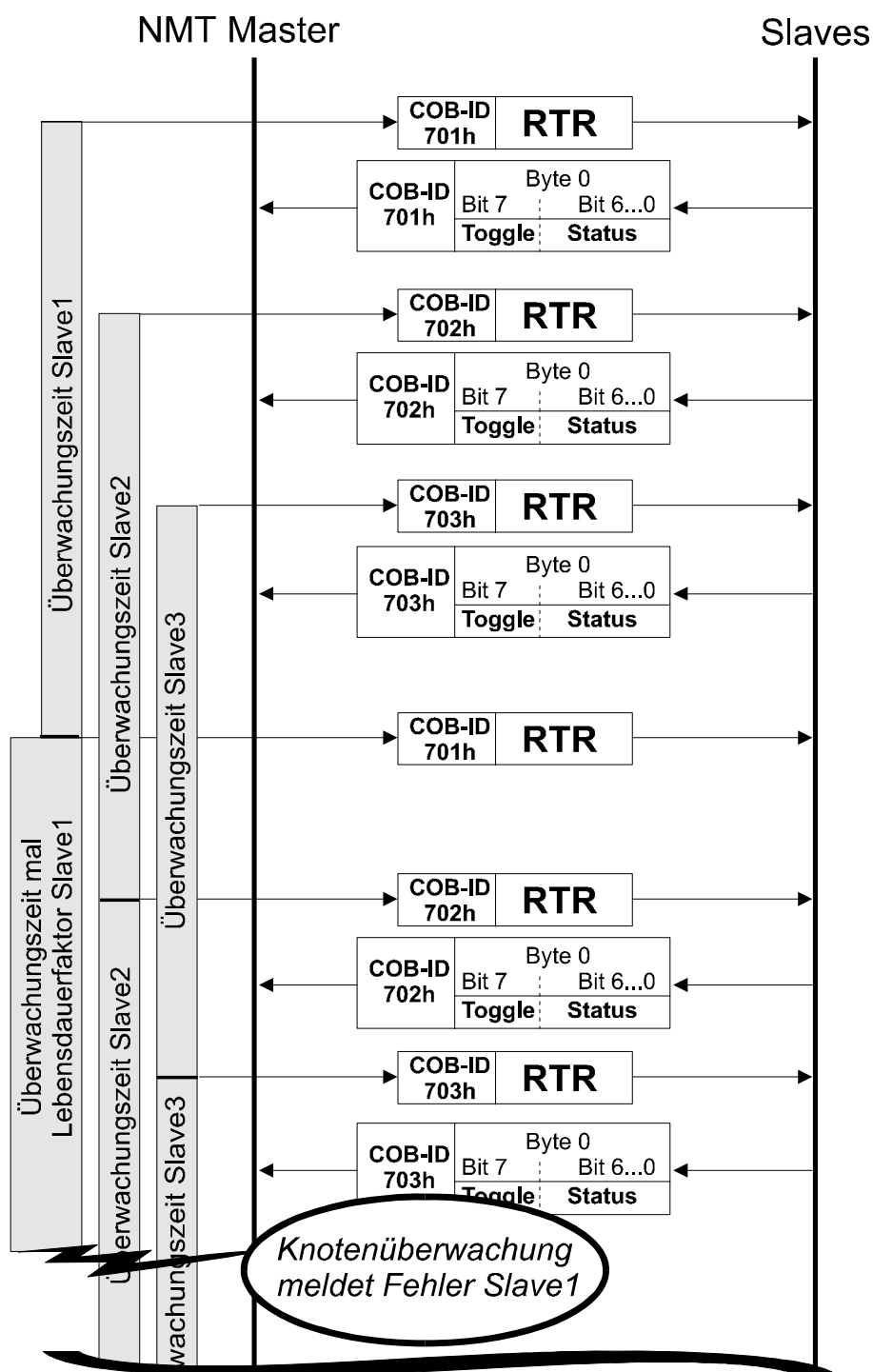
Werden die Daten des Teilnehmers regelmäßig über RTR's angefordert, so ist eine zusätzliche Knotenüberwachung nicht nötig.

Wird durch die Knotenüberwachung ein Fehler festgestellt, so sollte der fehlerhafte Knoten in einen sicheren Zustand gebracht werden. Abhängig von der Anwendung kann dies üblicherweise der Status Prepared sein.

Bei aktiver Knotenüberwachung sendet der Netzwerkmaster in regelmäßigen Abständen (Überwachungszeit) ein RTR-Telegramm mit dem COB-ID für Knotenüberwachung an den Knoten. Der Knoten muss innerhalb seiner mit dem Netzwerkmaster vereinbarten Lebensdauerzeit mit einer Statusmeldung antworten.

**i** Der COB-Identifizier für Knotenüberwachung wird durch Objekt 100Eh (**Standard-ID = 700h+Knotennummer**) festgelegt.  
Die **Überwachungszeit (100Ch)** multipliziert mit dem **Lebensdauer-Faktor (100Dh)** ergibt die **Lebensdauerzeit** für das Knotenüberwachungs-Protokoll.

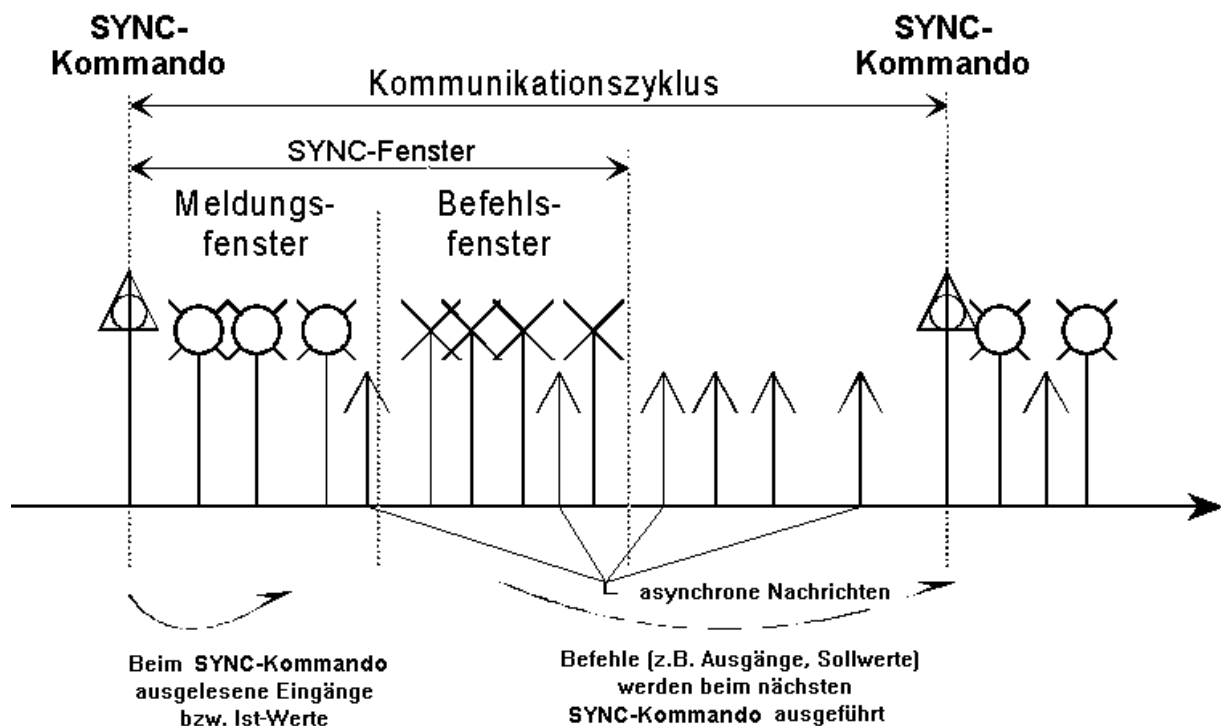
6.2.2 Beispiel: Überwachung der Knoten mit den Knotennummern 1, 2 und 3



### 6.2.3 Störungserkennung des Knotenüberwachungsprotokoll:

Störung	Ursache
Rückmeldetelegramm mit ungeändertem Toggle-Bit	Der Anwendungsprozess des Slaves ist nicht mehr aktiv oder im Moment zu beschäftigt
Der rückgemeldete Status des Knotens entspricht nicht mehr dem vom NMT-Master erwarteten Zustand	z.B. kurzer Spannungseinbruch
Der Slave antwortet nicht	<ul style="list-style-type: none"> <li>• Störung oder Abschaltung des Knotens</li> <li>• Ausfall des CAN-Busses</li> </ul>
Slave erhält keine Überwachungs-RTR's vom NMT-Master	Ausfall des NMT-Master

### 6.3 Bus-Synchronisation



Viele Echtzeitanwendungen verlangen neben der zyklischen Übertragung von Daten auch nach Synchronisation zwischen mehreren Busknoten. So müssen etwa Achsen einer Kinematik synchronisiert werden oder E/A-Baugruppen nach SPS-Art gleichzeitig Ausgänge setzen bzw. Eingänge lesen. Synchronisierte Antriebe erwarten Sollwerte und senden Istwerte in vorbestimmten Zeitfenstern. Das CANopen Kommunikationsprofil erfüllt diese Anforderungen durch (optionale) Synchronisationstelegramme, welche die Zeitachse in gleichgroße Kommunikationszyklen unterteilen. Der Identifier für die SYNC-Nachricht ist im Objekt **1005h** festgelegt. Standard-Wert = 80h

**i** Die SYNC-Nachrichten enthalten keine Daten und können von E/A-Modulen als Interrupt verwendet werden, um zu diesem Zeitpunkt Eingänge auszulesen bzw. Ausgänge zu setzen. Intelligente Geräte wie z.B. Antriebe können sich dementsprechend etwa per PLL-Verfahren synchronisieren.

Im **Meldungsfenster** direkt nach dem SYNC-Telegramm senden Knoten ihre Istwerte / Eingangswerte  
Im **Befehlsfenster** werden Soll- und andere Ausgangswerte übertragen, die dann beim nächsten SYNC-Telegramm gültig werden.

Die angesprochenen Fenster sind durch unterschiedliche Nachrichtenprioritäten gekennzeichnet. Da sich das Meldungsfenster an das SYNC-Kommando anschließt, wird es auch von einfachen Komponenten ohne Timer eingehalten.

Nicht genutzte Bandbreite in den Fenstern kann von Niedrigpriorären SDO-Nachrichten verwendet werden. Diese haben ansonsten den Zeitraum zwischen Befehlsfenster und SYNC-Telegramm zur Verfügung.

## 7 Busanschluss

### 7.1 Allgemeines zum Busanschluss des Gebers



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



Der Schirm ist beidseitig mit Erde zu verbinden. Auch die Leitungen zur Spannungsversorgung sollten vollständig abgeschirmt sein. Ist dies nicht möglich, so sind entsprechende Filtermaßnahmen zu ergreifen.



Verwenden Sie als Bus- bzw. Anschlusskabel nur paarig verseilte Signalkabel (CAN+ mit CAN-, +UB mit 0 V) mit Abschirmung.

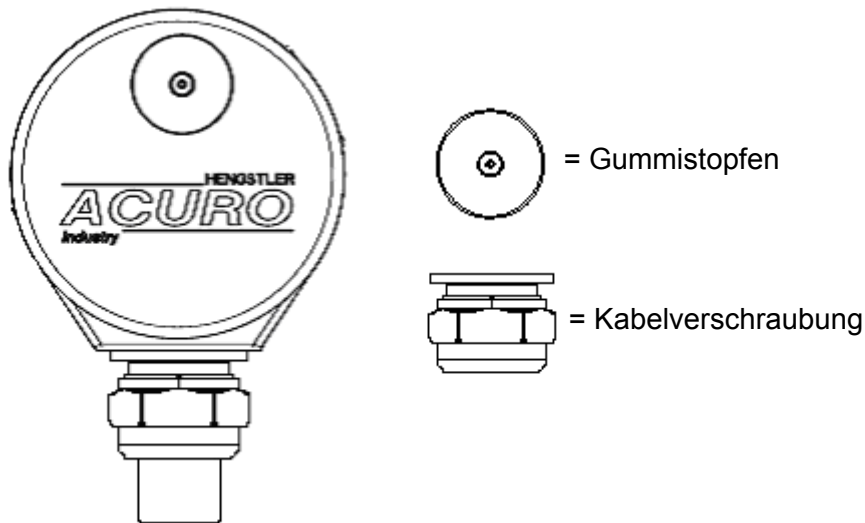
### 7.2 Anschlussarten

Für CANopen - Drehgeber sind vier Anschlussvarianten verfügbar!

Im Folgenden wird beschrieben, wie die jeweilige Anschlussvariante an den Bus anzuschließen ist.

Anschlussvariante	Seiten
Flanschdose 1-fach mit Kabel	46/47
Flanschdose 1-fach mit Conin	48/49
Bushaube mit 2 Coninstecker	50/51
Bushaube mit 3 Kabelverschraubungen	52/53

### 7.2.1 Kabel, axial

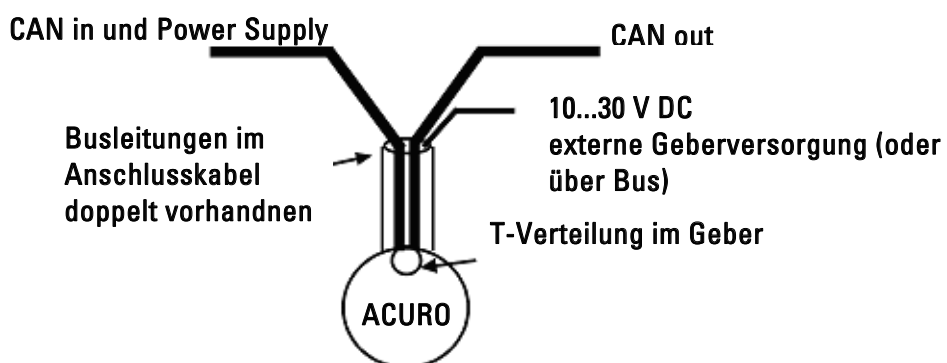


**⚠ Keine Stichleitung verwenden!**

- ⇒ Doppelt ausgeführte Busleitung verwenden
- ⇒ Ankommende Busleitung mit Leitungspaar 1 (CAN in + und CAN in -), blau (CAN GND in) sowie mit Leitungspaar 3 (UB in, 0 V in) des TPE-Kabels verbinden (siehe Anschlussbelegung nächste Seite).

Falls im gleichen Bus-Strang noch weitere Geräte folgen:

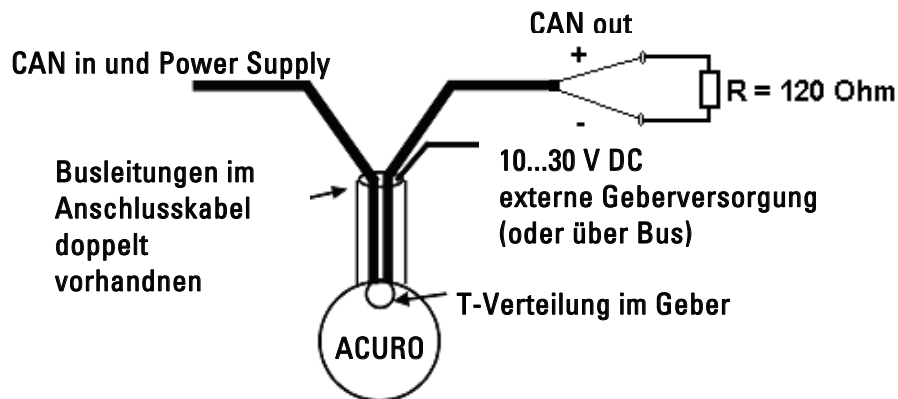
- ⇒ Abgehende Busleitung mit Leitungspaar 2 (CAN out +, CAN out -) sowie mit braun (CAN GND out) verbinden (siehe Anschlussbelegung nächste Seite).



Falls im gleichen Bus-Strang keine weiteren Geräte folgen:

**⚠ Externer Busabschluss erforderlich!**

⇒ Am Ende der abgehenden Busleitung zwischen CAN out + und CAN out - einen Abschlusswiderstand von **120 Ohm** anbringen!

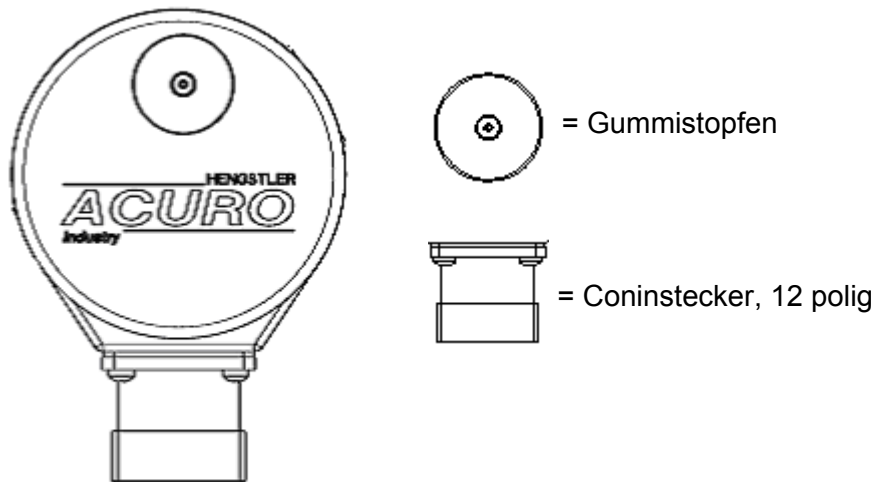


**Anschlussbelegung:**

Leitungspaare	TPE-Kabel	Signal	Beschreibung
Paar 1	gelb	CAN in+	CAN+ Busleitung (dominant H)
	grün	CAN in -	CAN - Busleitung (dominant L)
Paar 2	rosa	CAN out +	CAN+ Busleitung (dominant H)
	grau	CAN out -	CAN - Busleitung (dominant L)
	blau	CAN GND in	Ground (0V Potential für CAN Bus)
	braun	CAN GND out	Ground (0V Potential für CAN Bus)
Paar 3	weiß	UB in	Geberversorgung
	braun	0V in	Ground (0V für Geberversorgung)
	Schirm	Schirm	Schirm mit Gebergehäuse verbunden

**Hinweis:** Paarig verseilte Signalkabel mit Abschirmung verwenden, Schirm beidseitig auflegen!

## 7.2.2 Conin, axial

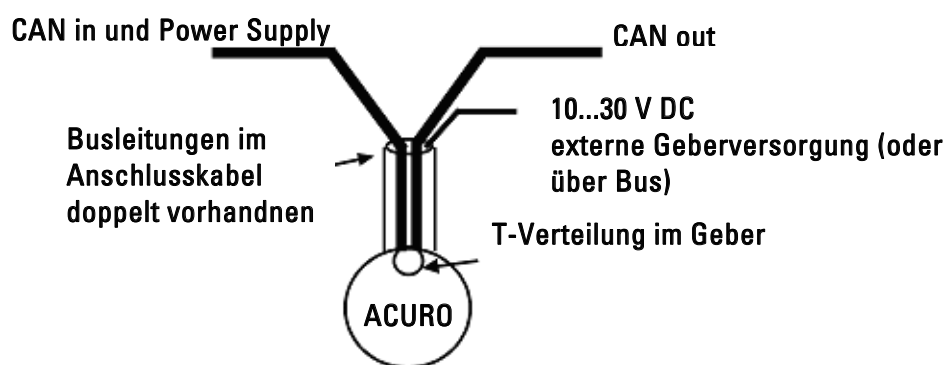


### Keine Stichleitung verwenden!

- ⇒ Doppelt ausgeführte Busleitung verwenden
- ⇒ Ankommende Busleitung mit PIN 7 (CAN in +), PIN 2 (CAN in -), PIN 3 (CAN GND in), PIN 12 (UB in) und PIN 10 (0 V in) verbinden (siehe Anschlussbelegung nächste Seite unten).

### Falls im gleichen Bus-Strang noch weitere Geräte folgen:

- ⇒ Abgehende Busleitung mit PIN 4 (CAN out +), PIN 5 (CAN out -) sowie mit PIN 11 (CAN GND out) verbinden (siehe Anschlussbelegung nächste Seite unten).



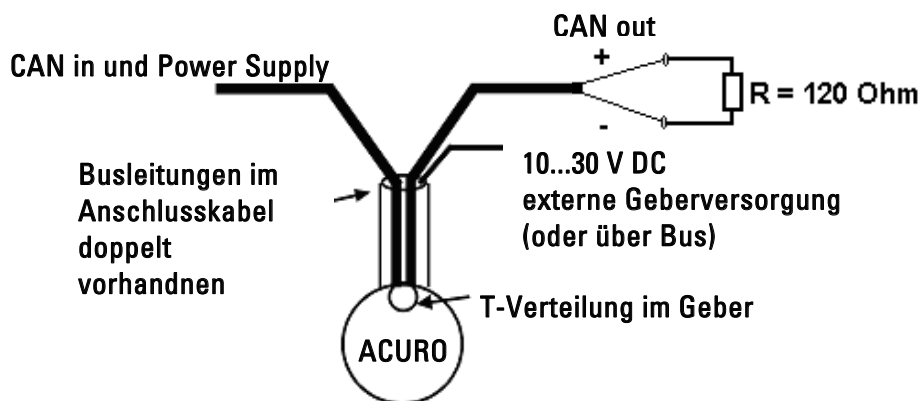


Falls im gleichen Bus-Strang keine weiteren Geräte folgen:

**Möglichkeit A: Busleitung wird aus dem Geber weitergeführt!**

**Externer Busabschluss erforderlich!**

⇒ Am Ende der abgehenden Busleitung zwischen CAN out + und CAN out - einen Abschlusswiderstand von **120 Ohm** anbringen!



**Möglichkeit B: Busleitung endet im Geber!**

⇒ Gummistopfen von der Geberkappe entfernen. Dadurch erhalten sie freie Sicht auf DIP-Schalter und LED-Anzeige (siehe 8.2).

⇒ Busabschlusswiderstand (DIP-Schalter 9 und 10 auf „ON“) aktivieren!

**Warnung !**

**Geber kann undicht werden, wenn der Gummistopfen falsch aufgesetzt oder beschädigt wird!**

- IP-Schutzart nicht garantiert!
- Ausfall des Gebers möglich!

⇒ Gummistopfen vorsichtig entfernen!

⇒ Drücken Sie nach dem Einsetzen mehrmals auf den Gummistopfen, um einen dichteten Sitz des Stopfens in der Bohrung zu erreichen

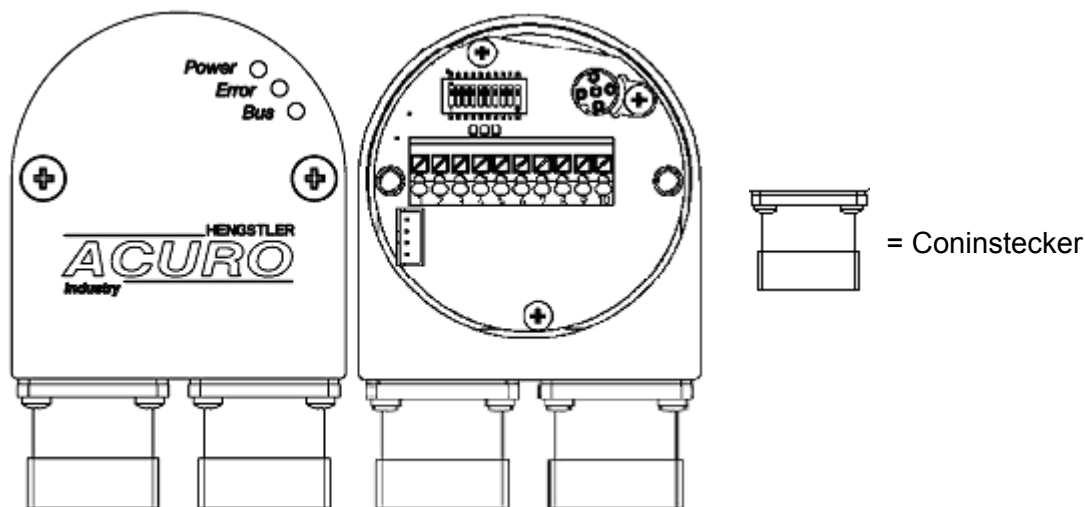
⇒ Beschädigte Stopfen müssen ersetzt werden (Art.-Nr. 2565007)

**Anschlussbelegung:**

Conin-PIN	Signal	Beschreibung
7	CAN in+	CAN+ Busleitung (dominant H)
2	CAN in -	CAN - Busleitung (dominant L)
4	CAN out +	CAN+ Busleitung (dominant H)
5	CAN out -	CAN - Busleitung (dominant L)
3	CAN GND in	Ground (0V Potential für CAN Bus)
11	CAN GND out	Ground (0V Potential für CAN Bus)
12	UB in	Geberversorgung
10	0V in	Ground (0V für Geberversorgung)
Schirm	Schirm	Schirm mit Gebergehäuse verbunden

**Hinweis:** Paarig verseilte Signalkabel mit Abschirmung verwenden, Schirm beidseitig auflegen!

### 7.2.3 Bushaube mit zwei Coninstecker



⇒ Schließen Sie den ankommenden Bus an den Gebereingang (Stifteinsatz) an.

Falls im gleichen Bus-Strang noch weitere Geräte folgen:

⇒ Weiterführenden Bus an den Geberausgang (Buchseinsatz) anschließen.



**Maximaler Strom von 2 A über Pin 7 und 8 zu beachten!**

**Empfohlene externe Sicherung für die gesamte Bus-Versorgungsspannung: T=2A**

Falls im gleichen Bus-Strang keine weiteren Geräte folgen:

⇒ Schrauben lösen und Bushaube vom Geber abziehen.

⇒ **Blickrichtung in die geöffnete Bushaube**

⇒ In der Bushaube den **Busabschlusswiderstand** (DIP-Schalter 9 und 10 auf „ON“) aktivieren!

⇒ Bushaube auf Geber stecken und Schrauben anziehen.

#### Anschlussbelegung:

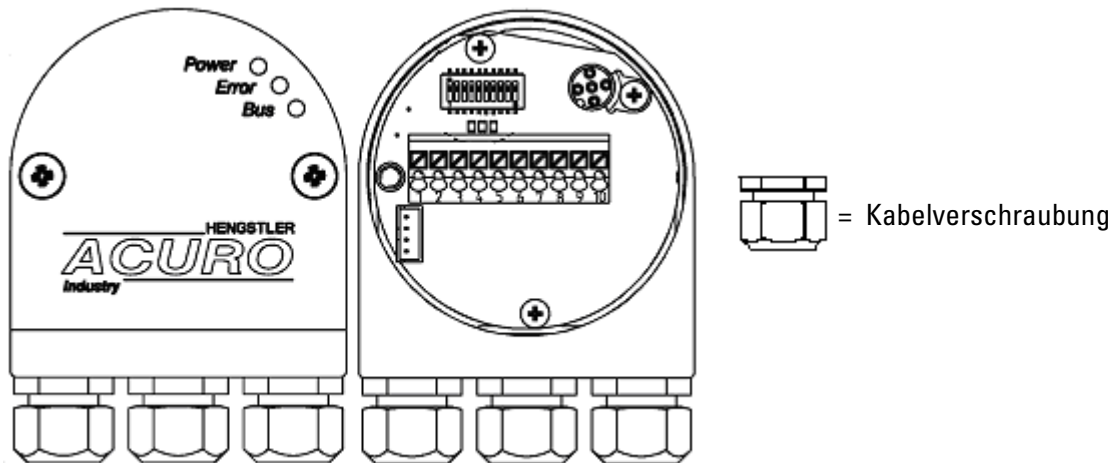
Conin-PIN	Bushaube mit		Beschreibung
	Stifteinsatz (IN)	Buchseinsatz (OUT)	
1	CAN in +	CAN out +	CAN+ Busleitung (dominant H)
2	CAN in –	CAN out -	CAN – Busleitung (dominant L)
3	CAN GND in	CAN GND out	Ground (0V Potential für CAN Bus)
4	N.C.	N.C.	nicht belegt
5	N.C.	N.C.	nicht belegt
6	N.C.	N.C.	nicht belegt
7	UB in	UB out	Geberversorgung
8	0V in	0V out	Ground (0V für Geberversorgung)
9	N.C.	N.C.	nicht belegt
Schirm	Schirm	Schirm	Schirm mit Gebergehäuse verbunden

**Hinweis:** Paarig verseilte Signalkabel mit Abschirmung verwenden, Schirm beidseitig auflegen!

## 7.2.4 Bushaube mit 3 Kabelverschraubungen

**i** Bei dieser Anschlussvariante gibt es zwei verschiedene Möglichkeiten den Geber anzuschließen.

- **Möglichkeit A: Anschluss mit Spannungsversorgung im Datenkabel**
- **Möglichkeit B: Anschluss mit separater Spannungsversorgung**



### Möglichkeit A: Anschluss mit Spannungsversorgung im Datenkabel

⇒ Schrauben lösen und Bushaube vom Geber abziehen.

⇒ **Blickfeld: in die geöffnete Bushaube**

⇒ **Die mittlere Kabelverschraubung durch die beiliegende Verschlusschraube ersetzen. Dadurch wird sichergestellt, dass der Geber dicht ist.**

⇒ Durch die linke Kabelverschraubung die Geber-Versorgungsspannung und Datenkabel führen und an der Klemme 1 (UB in), Klemme 2 (0V in), Klemme 3 (CAN in -), Klemme 4 (CAN in +) und Klemme 5 (CAN GND in) anschließen (siehe Anschlussbild, Seite 53). Kabelschirm an der Kabelverschraubung auflegen (siehe Grafik Kabelanschluss, Seite 53)

Falls im gleichen Bus-Strang keine weiteren Geräte folgen:

⇒ In der Bushaube den Busabschlusswiderstand (DIP-Schalter 9 und 10 von S1 auf „ON“) aktivieren!  
**Rechte Kabelverschraubung durch Verschlusschraube ersetzen.**

Falls im Bus-Strang noch weitere Geräte folgen:

⇒ Weiterführendes Kabel durch die rechte Kabelverschraubung führen und an Klemme 6 (CAN GND out), Klemme 7 (CAN out +), Klemme 8 (CAN out-), Klemme 9 (0V out) und Klemme 10 (UB out) anschließen (siehe Anschlussbild, Seite 53). Kabelschirm an der Kabelverschraubung auflegen (siehe Grafik Kabelanschluss, Seite 53).

⇒ Bushaube auf Geber stecken und Schrauben anziehen.

## **Möglichkeit B: Anschluss mit separater Spannungsversorgung**

- ⇒ Schrauben lösen und Bushaube vom Geber abziehen.
- ⇒ **Blickfeld: in die geöffnete Bushaube**
- ⇒ Durch die mittlere Kabelverschraubung die Geber-Versorgungsspannung führen und an die Klemme 1 (UB in) und Klemme 2 (0V in) anschließen (siehe Anschlussbild, Seite 53). Kabelschirm an der Kabelverschraubung auflegen (siehe Grafik Kabelanschluss, Seite 53).
- ⇒ Ankommendes Buskabel durch die linke Kabelverschraubung führen und an der Klemme 3 (CAN in -), Klemme 4 (CAN in +) und Klemme 5 (CAN GND in) anschließen (siehe Anschlussbild, Seite 53). Kabelschirm an der Kabelverschraubung auflegen (siehe Grafik Kabelanschluss, Seite 53).

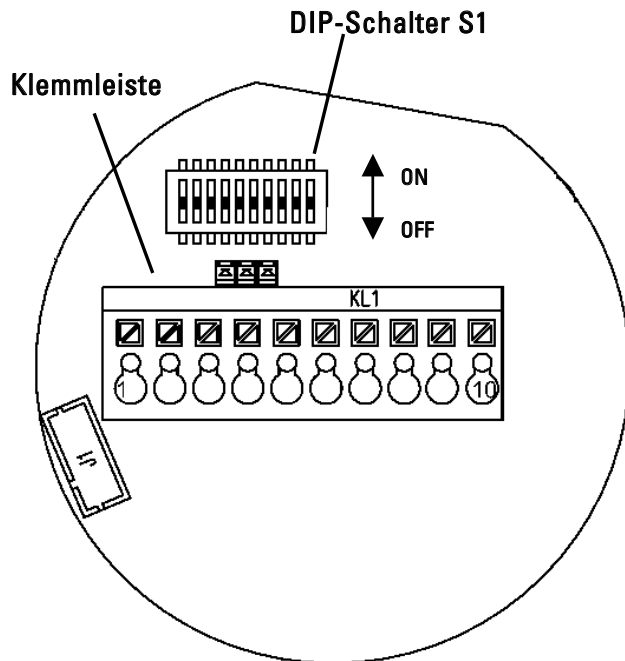
## **Falls im gleichen Bus-Strang keine weiteren Geräte folgen:**

- ⇒ In der Bushaube den Busabschlusswiderstand (DIP-Schalter 9 und 10 von S1 auf „ON“) aktivieren!  
***Rechte Kabel-Verschraubung durch Verschlusschraube ersetzen.***

## **Falls im Bus-Strang noch weitere Geräte folgen:**

- ⇒ Weiterführendes Kabel durch die rechte Kabelverschraubung führen und an der Klemme 6 (CAN GND out) , Klemme 7 (CAN out +) und Klemme 8 (CAN out-) anschließen (siehe Anschlussbild, Seite 53). Kabelschirm an der Kabelverschraubung auflegen (siehe Grafik Kabelanschluss, Seite 53).
- ⇒ Bushaube auf Geber stecken und Schrauben anziehen.

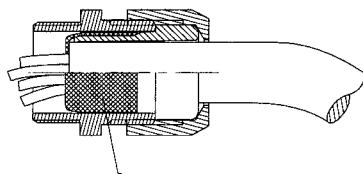
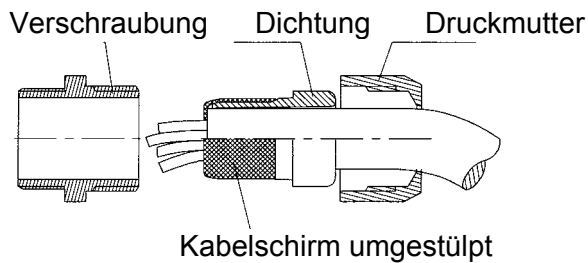
## 7.3 Anschlussbild



### Anschlussbelegung:

Klemmleiste KL 1 (10-polig)	
Nr.	Signalname
1	UB in (10-30V)
2	0V in
3	CAN in - (dominant L)
4	CAN in + (dominant H)
5	CAN GND in
6	CAN GND out
7	CAN out + (dominant H)
8	CAN out - (dominant L)
9	0V out
10	UB out (10-30V)

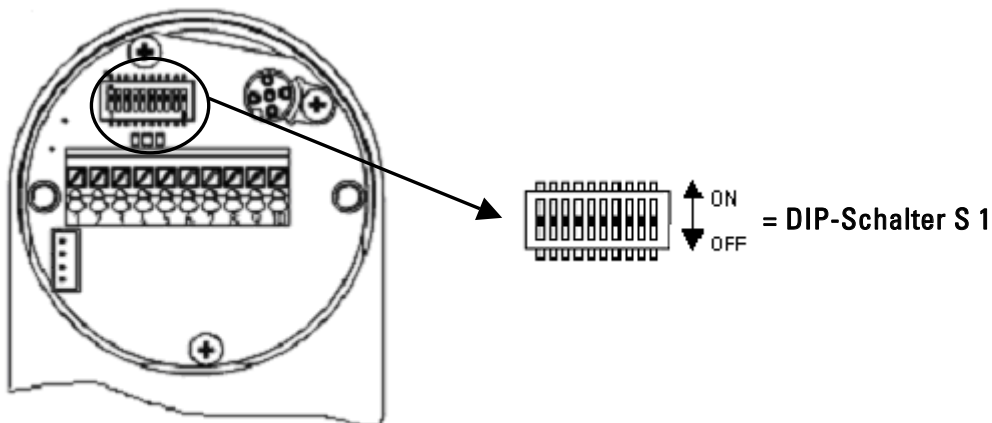
## 7.4 Kabelanschluss



Schirm umgestülpt stellt Kontakt zum Gehäuse her

## 8 Bedien- und Anzeigeelemente

### 8.1 DIP-Schalter (S1)



⇒ Bei der Ausführung mit Flanschdose 1-fach erreichen Sie die DIP-Schalter nach Entfernung eines Gummistopfens in der Geberkappe(siehe 8.4).



#### Warnung !

**Geber kann undicht werden, wenn der Gummistopfen falsch aufgesetzt oder beschädigt wird!**

- IP-Schutzart nicht garantiert!
  - Ausfall des Gebers möglich!
- ⇒ Gummistopfen vorsichtig entfernen!  
⇒ Drücken Sie nach dem Einsetzen mehrmals auf den Gummistopfen, um einen dichteten Sitz des Stopfens in der Bohrung zu erreichen  
⇒ Beschädigte Stopfen müssen ersetzt werden (Art.-Nr. 2565007),

⇒ **Bei den Ausführungen mit Bushaube sind die DIP-Schalter bei abgezogener Bushaube zugänglich.**

**i** Die DIP-Schalter werden nur beim Hochlaufen des Gebers (nach Reset oder Power-ON) ausgewertet. Eine Änderung der Schalterstellung hat damit bis zum nächsten Reset/Power-ON keine Wirkung.

Die Übernahme erfolgt nur dann, wenn sich seit dem vorherigen Reset die Einstellung der DIP-Schalter geändert hat. Ansonsten bleibt die im EEPROM abgelegte, evtl. Über den Bus programmierte Identifier erhalten.

Mit den DIP-Schaltern:

- 1 bis 5 von S1 wird die Knotennummer eingestellt.
- 6 bis 8 von S1 wird die Baudrate eingestellt.
- 9 und 10 von S1 wird der Abschlusswiderstand aktiviert.

## 8.1.1 Einstellung der Knoten-Nummer

Die 7 Bit Knotennummer wird hardwaremäßig über die **DIP-Schalter 1 bis 5** eingestellt. Die 5 DIP-Schalter legen die Bits 0 bis 4 fest, die verbleibenden Bits 5 und 6 sind fest auf 0 eingestellt.

DIP-Schalter (ON = 1, OFF = 0)							
DIP-Schalter	1	2	3	4	5	-	-
	LSB				MSB		
Bit-Nr.	0	1	2	3	4	5	6
Wert	1	2	4	8	16	0	0

- i** Die Knotennummer 0 ist reserviert und darf von keinem Knoten verwendet werden. Die Knotennummer 0 wird deshalb intern in 1 abgeändert.  
Die zulässigen Knotennummern liegen im Bereich 1...31.  
Jede Knotennummer darf nur einmal im Netzwerk vorhanden sein!

Standardeinstellung ab Werk: DIP-Schalter auf "OFF" (=0), d.h. die resultierende Knotennummer ist 1!

Die Übernahme einer neuen Knotennummer erfolgt erst beim nächsten Hochlaufen (Reset/Power-on) des Gebers. Alle Identifier werden dann auf ihre Standard-Werte zurückgesetzt

**Beispiel:**

DIP-Schalter (ON = 1, OFF = 0)							
DIP-Schalter	1=ON	2=OFF	3=OFF	4=ON	5=OFF	-	-
	LSB				MSB		
Bit-Nr.	0	1	2	3	4	5	6
Wert	1	2	4	8	16	0	0

→ Die **Knotennummer = 9**, weil DIP-Schalter 1=ON und DIP-Schalter 4=ON.

## 8.1.2 Einstellung der Baudrate

Die Baudrate wird über die DIP-Schalter 6, 7 und 8 eingestellt:

DIP-Schalter (ON = 1, OFF = 0)			Baudrate in KBit/s
DIP6	DIP7	DIP8	
OFF	OFF	OFF	1000
<b>ON</b>	<b>OFF</b>	<b>OFF</b>	<b>800</b>
OFF	ON	OFF	500
ON	ON	OFF	250
OFF	OFF	ON	125
ON	OFF	ON	50
OFF	ON	ON	20
ON	ON	ON	10

**i** Die Standardeinstellung ab Werk für die Baudrate ist 800 KBit/s

**i** Falls für das PDO die **Übertragungsart 254 verwendet** wird (asynchron Ereignisgesteuert, siehe Objekt 1800h):  
Die gewählte Zykluszeit (siehe Objekt 6200h) muss größer als die Busübertragungsdauer sein, damit die PDO's ungestört 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 (d. h. PDO bei Wertänderung) muss die Baudrate mindestens 125 KBaud betragen.**



### 8.1.3 Aktivieren des Busabschlusswiderstandes

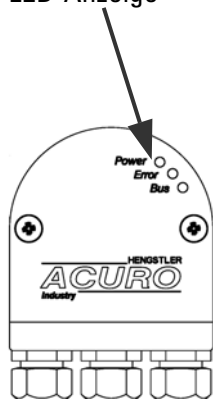
⇒ DIP-Schalter 9 und 10 auf „ON“ stellen, falls der Geber das letzte Gerät am Bus ist.

**i** Wenn sie extern einen Abschlusswiderstand anschließen müssen sie folgenden Wert beachten: 120 Ohm

**i** Bei Verwendung eines externen Abschluss muss der interne Abschluss deaktiviert sein (DIP 9 und DIP 10 = off)

### 8.2 LED-Anzeige bei Bushaube

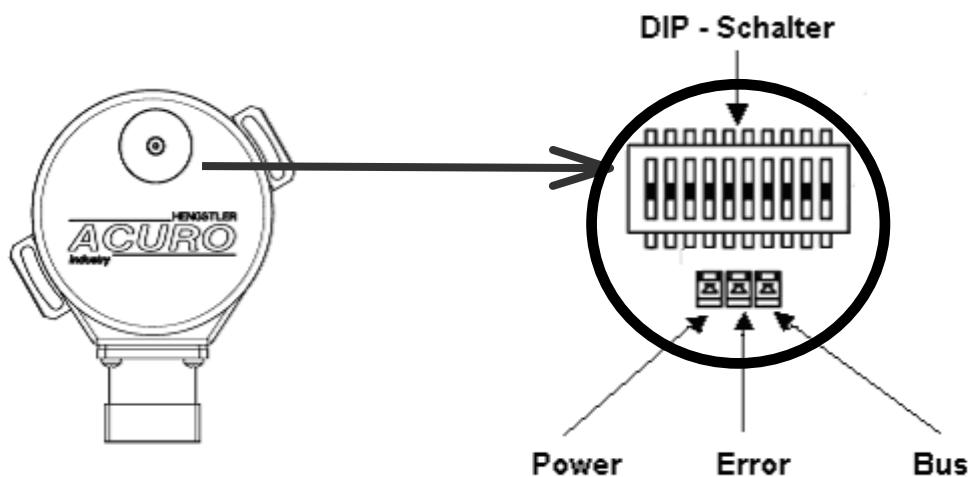
LED-Anzeige



	LED	Bedeutung	Mögliche Fehlerursache
<b>Power (grün)</b>	AUS	Spannungsversorgung fehlt	- Unterspannung - Polarität falsch
	EIN	Spannungsversorgung ist in Ordnung	
<b>Error (rot)</b>	AUS	Geberteil liefert fehlerfreie Positionsdaten	
	Blinkend	Geberteil liefert keine oder falsche Positionsdaten	- Unterbrechung BUS-Teil zu Geberteil - Gültiger LED-Sendestromregelbereich verlassen
<b>Bus (grün)</b>	AUS	Keine Verbindung zum Master	- Datenleitungsunterbrechung - Falsche Baudrate eingestellt - Vertauschte Datenleitung
	Blinkend	Verbindung zum Master, pre-operational-Zustand	
	EIN	Verbindung zum Master, operational-Zustand	

### 8.3 LED-Anzeige bei Flanschdose

⇒ Entfernen Sie den Gummistopfen. Dadurch erhalten sie freie Sicht auf DIP-Schalter und LED-Anzeige!



#### Warnung !

**Geber kann undicht werden, wenn der Gummistopfen falsch aufgesetzt oder beschädigt wird!**

- IP-Schutzart nicht garantiert!
  - Ausfall des Gebers möglich!
- ⇒ Gummistopfen vorsichtig entfernen!  
⇒ Drücken Sie nach dem Einsetzen mehrmals auf den Gummistopfen, um einen dichteten Sitz des Stopfens in der Bohrung zu erreichen  
⇒ Beschädigte Stopfen müssen ersetzt werden (Art.-Nr. 2565007),

## 9 Inbetriebnahme

Es wird vorausgesetzt, dass der Geber richtig angeschlossen, die korrekte Baudrate und eine Knotennummer eingestellt wurde.

### 9.1 Versorgungsspannung einschalten

Nach dem Einschalten der Versorgungsspannung wird der Geber initialisiert und befindet sich dann im Status Pre-operational.

Durch die Initialisierungsroutine werden die Parameter des Objektverzeichnisses aus dem EEPROM in den Arbeitsspeicher (RAM) des Gebers geladen.

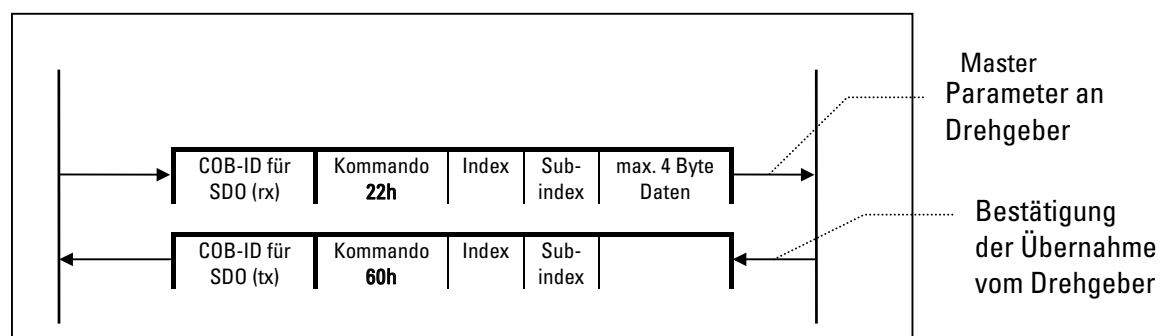
Wurde der Geber noch nicht programmiert, sind alle Parameter auf Standard-Werte gesetzt.

Wurde der Geber bereits programmiert, so arbeitet der Geber mit den zuletzt im EEPROM gespeicherten Werten.

### 9.2 Programmierung des Gebers

Soll der Geber neu programmiert werden, so wird empfohlen, dies jetzt im Zustand Pre-operational zu tun. In diesem Zustand ist die Busaktivität niedrig und der Geber ist noch nicht mit der Ausgabe der Prozessdaten belastet.

Die Kommunikations- und Betriebs-Parameter im Objektverzeichnis (siehe Kapitel 5) des Gebers können durch SDO-Zugriff geändert werden:



- i** Als Hilfsmittel für die Verwendung von Standard-CANopen-Tools dient die sogenannte EDS-Datei (Electronic Data Sheet Specification). Diese EDS-Datei beinhaltet das zur Verfügung stehende Objektverzeichnis des Gebers. Sie ist als Download-Datei auf unserer Internet-Homepage verfügbar.

### 9.2.1 Übersicht Standard-Werte

**i** Zur einfacheren Verwaltung der Identifier verwendet CANopen das "Predefined Master/Slave Connection Set". Dabei sind alle Identifier mit Standard-Werten im Objektverzeichnis definiert, so dass der Geber auch ohne weitere Programmierung in Betrieb genommen werden kann.

**Folgende Parameter können jedoch über SDO-Zugriff kundenspezifisch umprogrammiert werden (siehe auch Kapitel 5.1.2 "Einzelbeschreibung der Kommunikations-Parameter")**

Index (hex)	Name	Standard-Wert (hex)
<b>Kommunikations-Parameter:</b>		
1005	COB-ID für SYNC	80
100C	Überwachungszeit	0
100D	Lebensdauerfaktor	0
100E	COB-ID für Überwachungs-Protokoll	700 + Knotennummer
1014	COB-ID für Notfallobjekte	80 + Knotennummer
nicht veränderbar	SDO-Parameter Identifier für SDO (rx) (Master→ Geber) Identifier für SDO (tx) (Geber→ Master)	600 + Knotennummer 580 + Knotennummer
1800	PDO1 Parameter (asynchron)	
Sub-Index 1	Identifier	180 + Knotennummer
Sub-Index 2	Übertragungsart	FE
Sub-Index 3	Verbotszeit	0
1802	PDO2 Parameter (synchron, zyklisch)	
Sub-Index 1	Identifier	280 + Knotennummer
Sub-Index 2	Übertragungsart	1
Sub-Index 3	Verbotszeit	0
<b>Geber-Parameter:</b>		
2000	Warnpositionen	0
2001	Offset-Wert	0
2005	PDO Typ	0 (nur Position)
6000	Betriebsparameter	0 (cw, Skalierung aus)
6001	Messschritte pro Umdrehung	max. phys. Auflösung <sup>1</sup>
6002	Gesamtanzahl der Messschritte	max. physikalische Gesamtauflösung <sup>1</sup>
6003	Preset-Wert	0
6200	Zyklus-Timer	0

<sup>1</sup>Wert abhängig vom Gebertyp

**i** Um die geänderten Parameter auch spannungsausfallsicher abzuspeichern, müssen diese unbedingt über das Objekt 1010h (Parameter speichern) in das EEPROM übertragen werden. Die ursprünglichen Standard-Werte (Default-Werte bei Auslieferung) können durch das Objekt 1011h wieder zurück geladen werden.

**Achtung:** Die vorher im Geber-RAM vorhandenen Daten werden dadurch überschrieben!

**i** Sollte versehentlich für zwei CAN-Nachrichtenobjekte der gleiche Identifier über den Bus programmiert und ins EEPROM abgespeichert worden sein, dann lässt sich der Geber nach dem nächsten Hochfahren nicht mehr ansprechen (Geber sendet nur noch Emergency Nachrichten).

**Dieser Fehler wird durch folgende Maßnahmen behoben:**

- ⇒ Geber-DIP-Schalter ändern (alle Identifier werden auf ihre Standard-Werte gesetzt)
- ⇒ evtl. über den Bus neue Identifier setzen
- ⇒ Parameter über Objekt 1010h ins Geber-EEPROM abspeichern

## 9.2.2 Beispiel für eine Geber-Programmierung:

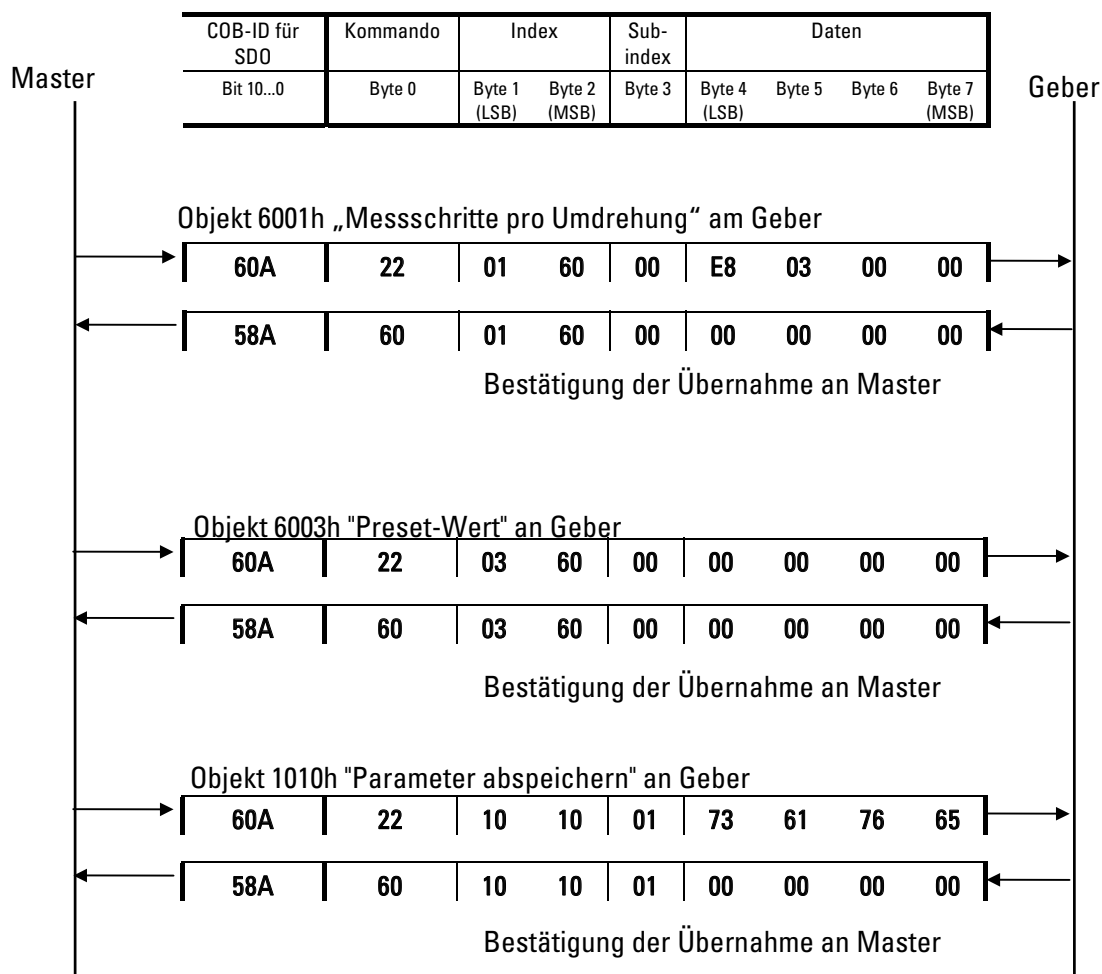
Vorgaben:

1. Knotennummer des Gebers = Ah (10 dezimal)
2. Auflösung soll auf 1000 Schritte pro Umdrehung gesetzt werden
3. Positionswert soll auf 0 gesetzt werden
4. Die neuen Parameter sollen im EEPROM gespeichert werden
5. Alle anderen Parameter bleiben auf Standard-Werten

Lösung:

1. COB-ID für SDO (rx) (Master an Drehgeber) = 600h + Knotennummer = 60Ah  
COB-ID für SDO (tx) (Drehgeber an Master) = 580h + Knotennummer = 58Ah
2. Objekt 6001h (Messschritte pro Umdrehung) mit Dateninhalt 3E8h (=1000) senden
3. Objekt 6003h (Preset-Wert) mit Dateninhalt 0 senden
4. Objekt 1010 (Parameter abspeichern) unter Sub-Index 1 mit Dateninhalt 65 76 61 73h als Codewort senden

**Ablauf der Datenübertragung:**



### 9.3 Betriebszustand herstellen

Über das NMT-Kommando 1: Wechsel zu Operational. Es kann entweder nur der Geber alleine über seine Knotennummer angesprochen werden, oder es werden unter Knotennummer 00h alle Knoten adressiert:

	Byte 0	Byte 1
COB-ID = 0	01h	Knoten-Nummer

## 10 Technische Daten

### 10.1 Mechanisch

Gehäusedurchmesser	58 mm
Schutzart Welleneingang	IP 64 oder IP 67
Schutzart Gehäuse	Bushaube IP 67 Kabel oder Conin IP 64 (IP 67 optional)
Flange	Synchroflansch, Klemmflansch, Federblech mit Hohlwelle, Quadratflansch
Wellendurchmesser	Vollwelle 6 mm, 10 mm; Hohlwelle 10 mm, 12mm
Max. Drehzahl	12000 min <sup>-1</sup> (kurzzeitig), 10000 min <sup>-1</sup> (durchgehend)
Drehmoment	≤ 0,5 Ncm
Trägheitsmoment	3,8 10 <sup>-6</sup> kgm <sup>2</sup>
Federblech (Hohlwelle)	
Toleranz axial	±1,5 mm
Toleranz radial	±0,2 mm
Max. Wellenbelastung	axial 40 N, radial 60 N
Schwingfestigkeit (IEC 68-2-6)	100 m/s <sup>2</sup> (10 - 500 Hz)
Schockfestigkeit (IEC 68-2-27)	1000 m/s <sup>2</sup> (6 ms)
Betriebstemperatur	-40...+85 °C
Lagertemperatur	-40...+85 °C
Material Welle	Edelstahl
Material Gehäuse	Aluminium
Masse ST/ MT	350g/ 400g



## 10.2 Elektrisch

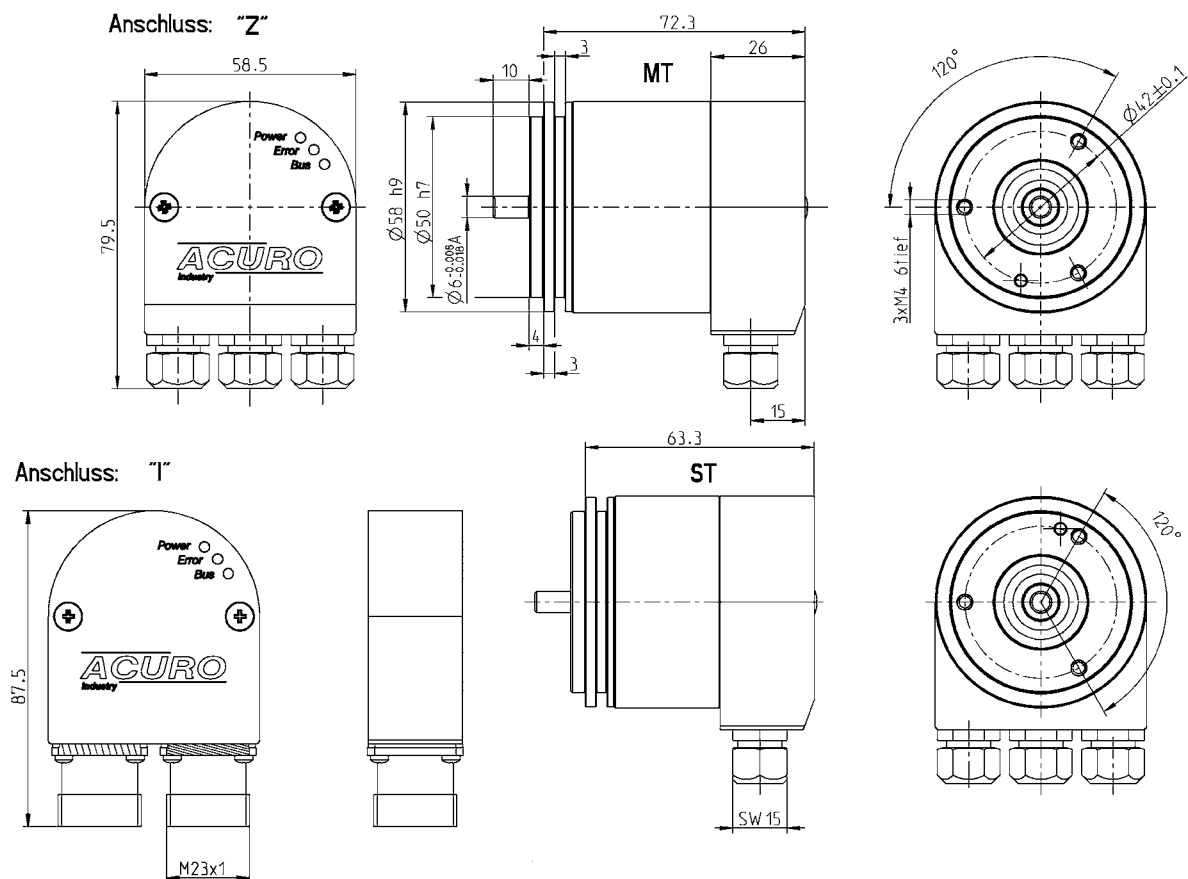
Versorgungsspannung	DC 10 - 30 V
Eigenstromaufnahme ST/ MT	220 mA/ 250 mA
EMV	Störaussendung nach EN 50081-2 Störfestigkeit nach EN 50082-2
Schnittstelle	CAN High-Speed nach ISO/ DIS 11898
Protokoll	CANopen nach DS 301 mit profile DSP406, Programmierbarer Geber nach Klasse C2
Allgemeine Auslegung	gemäß EN 601010-Teil 1, Schutzklasse III, Verschmutzungsgrad 2, Überspannungskategorie II
Programmierbar	Auflösung, Preset, Offset, Direction
Auflösung Singleturn	10 bis 14 Bit
Auflösung Multiturn	12 Bit
Linearität	$\pm \frac{1}{2}$ LSB ( $\pm 1$ LSB für Auflösung 13, 14, 25, 26 Bit)
Ausgabecode	binär
Integrierte Spezialfunktionen	Geschwindigkeit, Beschleunigung, Rundachse, Grenzwerte
Wertaktualisierung	jede Millisekunde (einstellbar), auf Anforderung
Knotennummer	über DIP-Schalter einstellbar
Baudrate	über DIP-Schalter einstellbar
Abschlusswiderstand	über DIP-Schalter einstellbar
Anschlussvarianten	Kabel radial oder axial Conin radial oder axial, cw oderr ccw Bushaube mit <ul style="list-style-type: none"><li>• 3 Kabelverschraubungen</li><li>• Doppelconin, 9 –pol., cw, radial</li></ul>

## 11 Maßzeichnungen

### 11.1 Synchroflansch

#### 11.1.1 Anschluss mit Bushaube

- I Bushaube Doppelconin, 9-pol., rechtsdrehend, radial
- Z Bushaube 3 x Kabelverschraubung



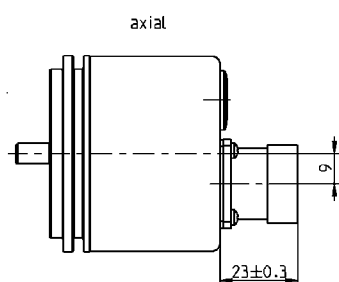
**i** Der bei Kabel-Verschraubungen geforderter Durchmesser der Anschlusskabel:  
7.0 ... 7.4 mm

**i** Buskabel mit Durchmesser 7.9 ... 8.7 mm können verwendet werden, wenn die Standard -  
Dichtungen gegen die beiliegenden Dichtungen mit größerem Durchmesser ausgetauscht  
werden.

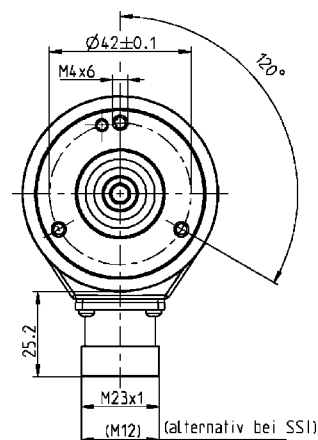
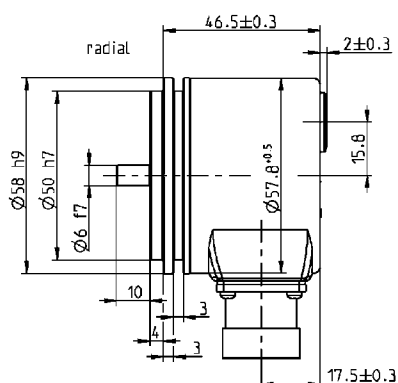
### 11.1.2 Anschluss mit Kappe

- A Kabel, axial
- B Kabel, radial
- C Coninstecker, 12-pol., axial, rechtsdrehend
- D Coninstecker, 12-pol., radial, rechtsdrehend
- G Coninstecker, 12-pol., axial, linksdrehend
- H Coninstecker, 12-pol., radial, linksdrehend

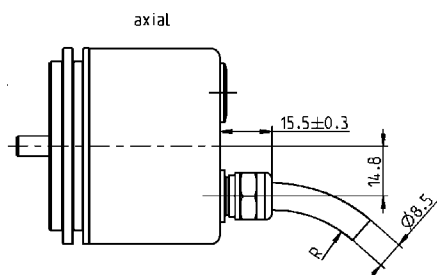
Anschluss: "C G"



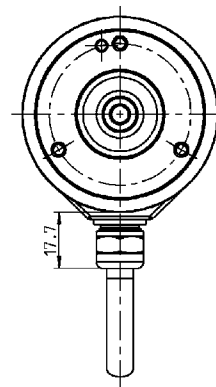
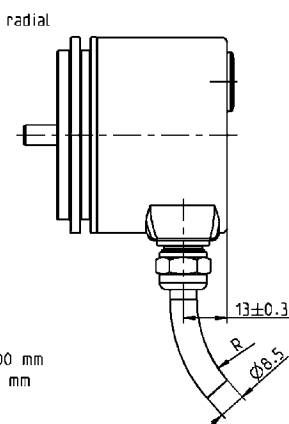
Anschluss: "D H"



Anschluss: "A"



Anschluss: "B"

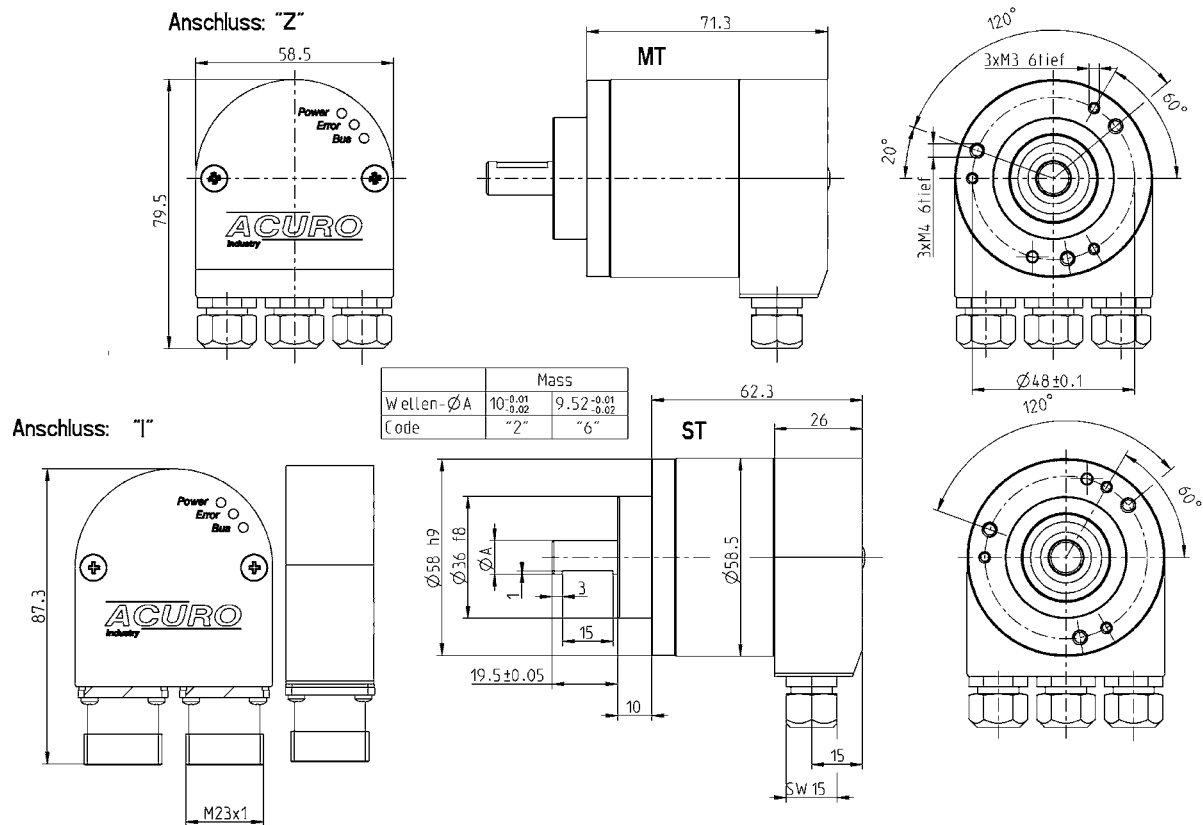


R bei bewegtem Einsatz => 100 mm  
R bei fester Verlegung => 40 mm

## 11.2 Klemmflansch

### 11.2.1 Anschluss mit Bushaube

- I Bushaube Doppelconin, 9-pol., rechtsdrehend, radial
- Z Bushaube 3 x Kabelverschraubung



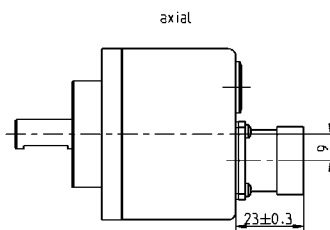
**i** Der bei Kabel-Verschraubungen geforderter Durchmesser der Anschlusskabel:  
7.0 ... 7.4 mm

**i** Buskabel mit Durchmesser 7.9 ... 8.7 mm können verwendet werden, wenn die Standard -  
Dichtungen gegen die beiliegenden Dichtungen mit größerem Durchmesser ausgetauscht  
werden.

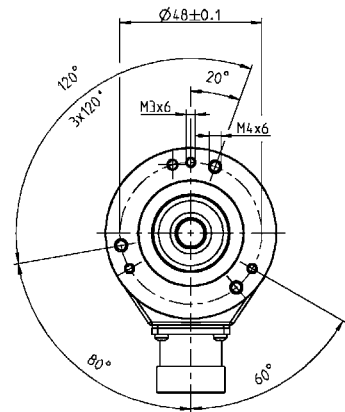
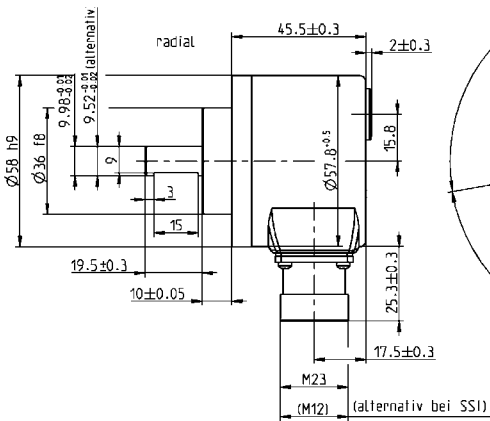
## 11.2.2 Anschluss mit Kappe

- A Kabel, axial
- B Kabel, radial
- C Coninstecker, 12-pol., axial, rechtsdrehend
- D Coninstecker, 12-pol., radial, rechtsdrehend
- G Coninstecker, 12-pol., axial, linksdrehend
- H Coninstecker, 12-pol., radial, linksdrehend

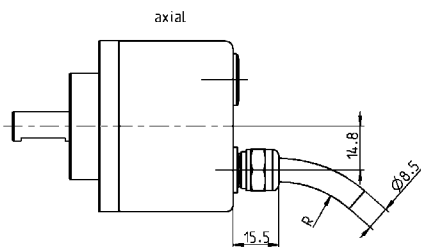
Anschluss: "C G"



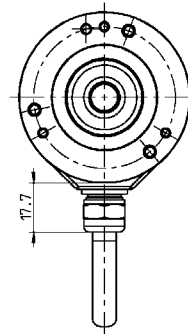
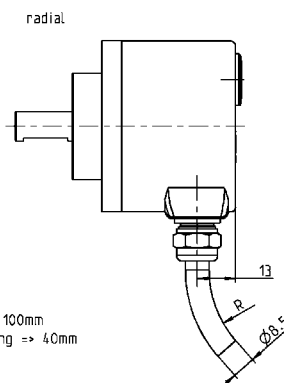
Anschluss: "D H"



Anschluss: "A"



Anschluss: "B"

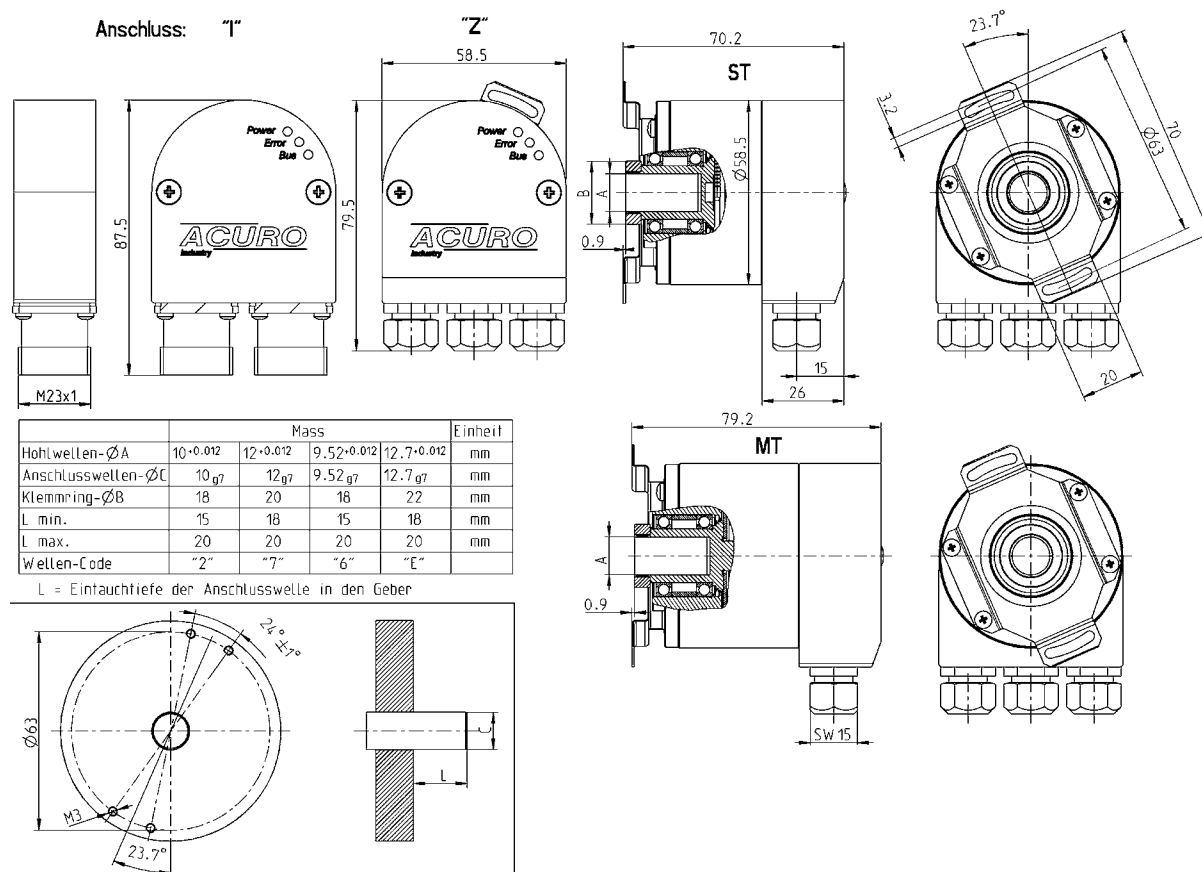


R bei Dauerbiegung => 100mm  
R bei einmaliger Biegung => 40mm

## 11.3 Hohlwelle mit Federblech

### 11.3.1 Anschluss mit Bushaube

- I Bushaube Doppelconin, 9-pol., rechtsdrehend, radial
- Z Bushaube 3 x Kabelverschraubung



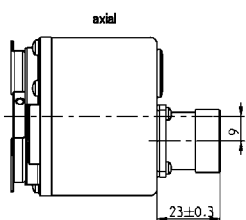
**i** Der bei Kabel-Verschraubungen geforderter Durchmesser der Anschlusskabel:  
7.0 ... 7.4 mm

**i** Buskabel mit Durchmesser 7.9 ... 8.7 mm können verwendet werden, wenn die Standard -  
Dichtungen gegen die beiliegenden Dichtungen mit größerem Durchmesser ausgetauscht  
werden.

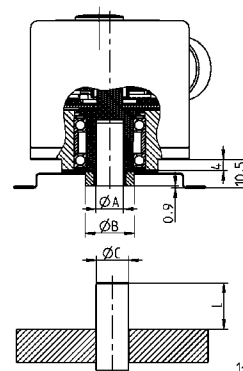
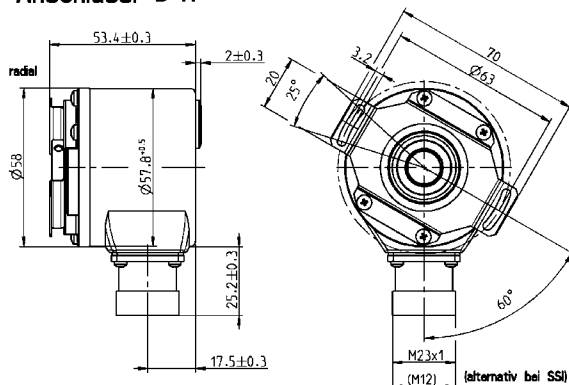
## 11.3.2 Anschluss mit Kappe

- A Kabel, axial
- B Kabel, radial
- C Coninstecker, 12-pol., axial, rechtsdrehend
- D Coninstecker, 12-pol., radial, rechtsdrehend
- G Coninstecker, 12-pol., axial, linksdrehend
- H Coninstecker, 12-pol., radial, linksdrehend

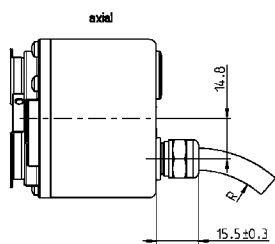
Anschluss: "C G"



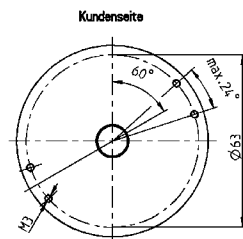
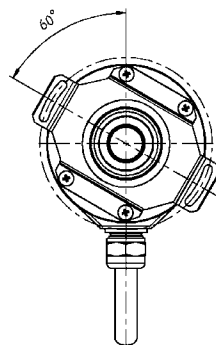
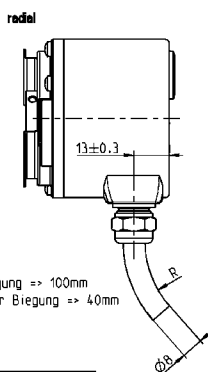
Anschluss: "D H"



Anschluss: "A"



Anschluss: "B"



R bei Dauerbiegung => 100mm  
R bei einmaliger Biegung => 40mm

L = Eintauchtiefe der Anschlusswelle in den Geber

	Metrische Maße		Zoll-Maße		Einheit
Hohlwellen-ØA	10+0,012	12+0,012	9.52+0,012	12.7+0,012	mm
Anschlusswellen-ØC	10 g7	12 g7	9.52 g7	12.7 g7	mm
Klemmring-ØB	18	20	18	22	mm
L min.	15	18	15	18	mm
L max.	20	20	20	20	mm
Wellen-Code	"2"	"7"	"6"	"E"	

