

Kode 85199C Ausgabe 06-2020

INHALTSVERZEICHNIS

- 1 Einführung
- 2 Elektrische Anschlüsse
- 3 Network Management (NMT)
- 4 Baudrate
- 5 Knoten-ID und Auflösung
- 6 Einstellung der Parameter
- 7 Wiederherstellen der Parameter
- 8 Heartbeat
- 9 Fehlerbehandlung
- 10 SDO-Kommunikation
- 11 PDO-Kommunikation und Winkelberechnung
- 12 Überblick über die Eigenschaften von CANopen
- 13 Kommunikationsbeispiele

1. EINFÜHRUNG

Die berührungslos arbeitenden Winkelsensoren GRA/GRN (HALL-Technik) implementieren die Funktionen eines Slave-Geräts in einem CAN-Bus-Netzwerk, das mit dem standardisierten CANopen-Protokoll nach CiA (Can in Automation) arbeitet, das in dem Dokument „CANopen Application Layer and Communication Profile DS 301 v. 4.2“ und in den anderen nachstehend genannten Dokumenten beschrieben ist.

Weitere zugrundegelegte Dokumente sind „CiA DS-406 Device Profile for Encoders V3“ (nicht vollständig implementiert) und „CiA DSP-305 Layer Setting Services and Protocol V1.1.1“.

Das vorliegende Dokument beschreibt die Spezifikationen des implementierten CANopen-Standards. Es richtet sich an Fachleute, die CANopen-Systeme einrichten oder CANopen-Geräte entwerfen und schon mit den oben genannten Standards des CiA vertraut sind.

Die Details der vom Kommunikationsprotokoll CANopen definierten Aspekte sind nicht Gegenstand des vorliegenden Dokuments.

Um weitere Einzelheiten zum Protokoll zu erfahren, können Sie sich über unsere Homepage www.gefran.com mit uns in Verbindung setzen oder an die nächste Geschäftsstelle von Gefran wenden.

Definitionen und Kurzbezeichnungen

CAN: Controller Area Network.

Ein Bussystem für die serielle Datenübertragung, das die physikalische Schicht 1 und die Sicherungsschicht 2 des ISO/OSI-Referenzmodells implementiert.

CAL: CAN Application Layer.

Sie beschreibt die Umsetzung des CAN auf Ebene der Anwendungsschicht 7 des ISO/OSI-Referenzmodells, auf dem CANopen basiert.

CMS: CAN Message Specification.

Dienstelement von CAL, das CAL für die verschiedenen industriellen Anwendungen definiert.

COB: Communication Object.

Datentransporteinheit in einem CAN-Netzwerk (eine CAN-Nachricht). In einem CAN-Netzwerk kann es maximal 2048 COBs geben, von denen jedes von 0 bis maximal 8 Bytes transportieren kann.

COB-ID: COB Identifier.

Er dient zum Kennzeichnen einer CAN-Nachricht. Der Identifier bestimmt die Priorität eines COB im Falle von mehreren Nachrichten im Netzwerk.

D1 – D8: Data from 1 to 8.

Anzahl der Bytes im Datenfeld einer CAN-Nachricht.

DLC: Data Length code.

Anzahl der Datenbytes, die in einer einzelnen Nachricht übertragen werden.

ISO: International Standard Organization.

Internationale Vereinigung von Normungsorganisationen, die Normen für die verschiedenen Warendektoren erarbeitet.

NMT: Network Management.

Dienstelement von CAL. Es beschreibt die Konfiguration, Initialisierung und Fehlerbehandlung in einem CAN-Netzwerk.

PDO: Process Data Object.

Objekt für den Austausch von Prozessdaten (mit hoher Priorität).

RXSDO: Receive SDO.

Vom Remote-Gerät empfangene SDOs.

SDO: Service Data Object.

Kommunikationsobjekt der Servicedaten (mit niedriger Priorität). Der Wert dieser Daten ist im Objektverzeichnis von jedem Gerät im CAN-Netzwerk enthalten.

TXPDO: Transmit PDO.

Vom Remote-Gerät gesendete PDOs.

TXSDO: Transmit SDO.

Vom Remote-Gerät gesendete SDOs.

HINWEIS: Der Suffix „h“ kennzeichnet hexadezimale Zahlen, der Suffix „b“ kennzeichnet Binärzahlen und der Suffix „d“ Dezimalzahlen. Falls nicht anders angegeben, erfolgen die Angaben im Dezimalformat.

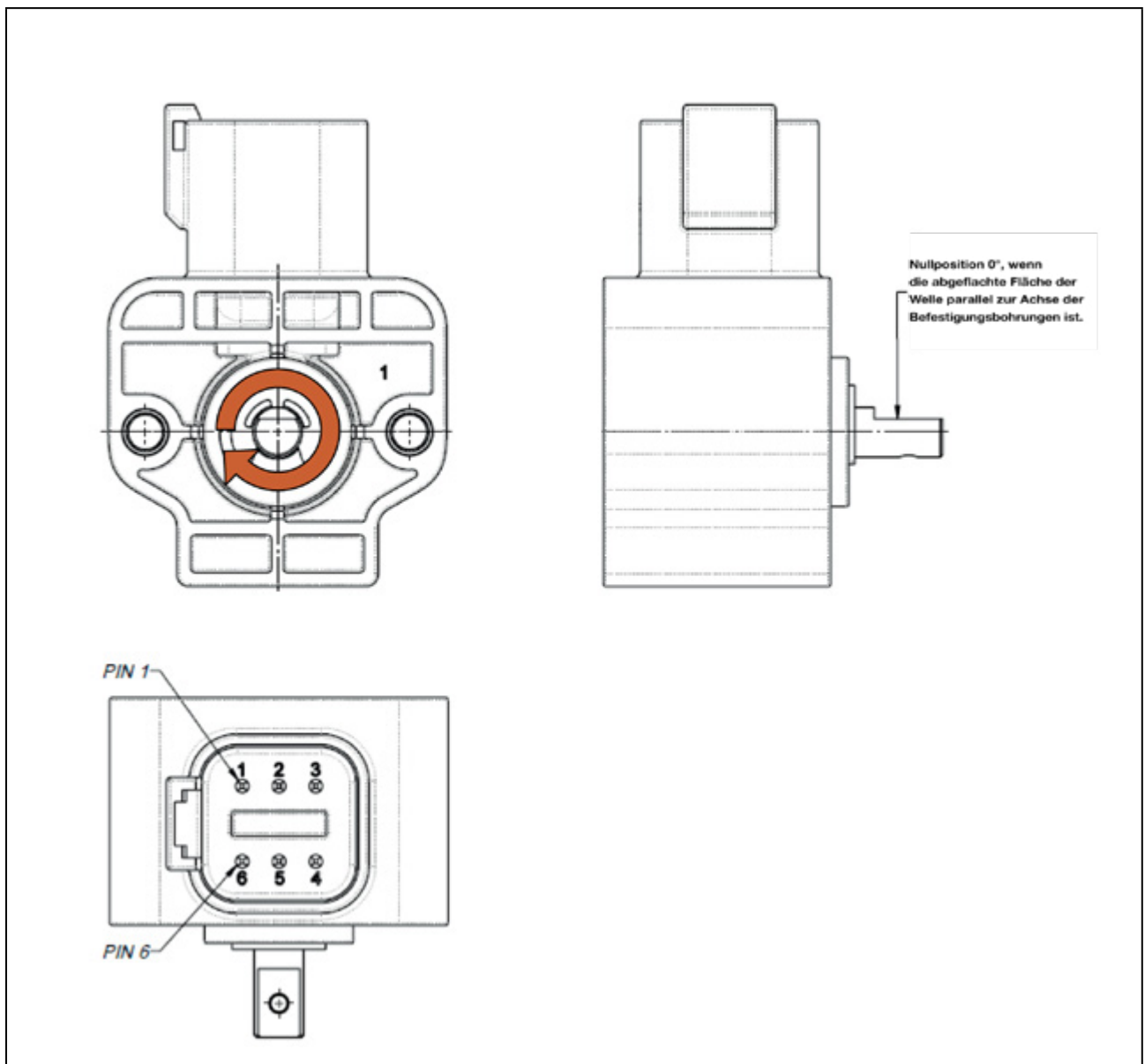
2. ELEKTRISCHE ANSCHLÜSSE

Bei den Anschlüssen ist die nachstehende Tabelle zugrunde zu legen:

| DEUTSCH DT04-6P | Bedeutung |
|-----------------|----------------------|
| 1 | OV (GND) |
| 2 | +Vs (+9 ... +36 Vdc) |
| 3 | NC |
| 4 | NC |
| 5 | CAN-L |
| 6 | CAN-H |

Hinweis: Der CAN-Bus muss terminiert sein.

Die zwischen CAN-H und CAN-L gemessene Impedanz muss 60 Ohm betragen. Daher muss das Kabel an jedem Ende der Busleitung an einen Widerstand von 120 Ohm angeschlossen werden. Der Messaufnehmer ist intern nicht mit dem Widerstand von 120 Ohm terminiert. Darauf achten, die Signalleitungen des CAN-Bus nicht zu verwechseln, da andernfalls die Kommunikation mit dem Messaufnehmer nicht möglich ist.



Bei den Anschlüssen ist die nachstehende Tabelle zugrundezulegen:

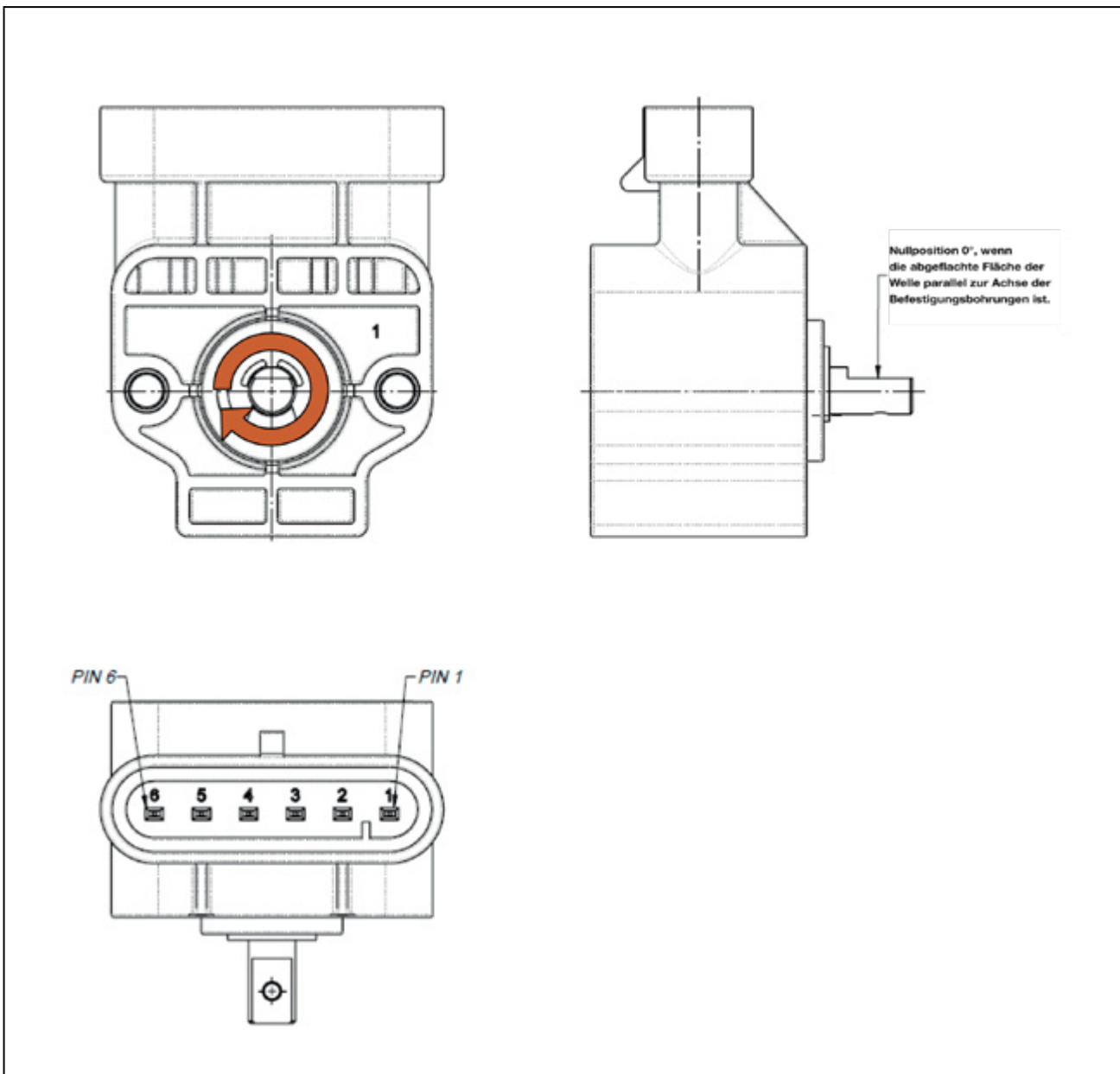
| AMP Superseal 6 P 282108-1 | Bedeutung |
|----------------------------|----------------------|
| 1 | OV (GND) |
| 2 | +Vs (+9 ... +36 Vdc) |
| 3 | n. a. |
| 4 | n. a. |
| 5 | CAN-L |
| 6 | CAN-H |

Hinweis: Der CAN-Bus muss terminiert sein.

Die zwischen CAN-H und CAN-L gemessene Impedanz muss 60 Ohm betragen. Daher muss das Kabel an jedem Ende der Busleitung an einen Widerstand von 120 Ohm angeschlossen werden.

Der Messaufnehmer ist intern nicht mit dem Widerstand von 120 Ohm terminiert.

Darauf achten, die Signalleitungen des CAN-Bus nicht zu verwechseln, da andernfalls die Kommunikation mit dem Messaufnehmer nicht möglich ist.



Bei den Anschlüssen ist die nachstehende Tabelle zugrunde zu legen:

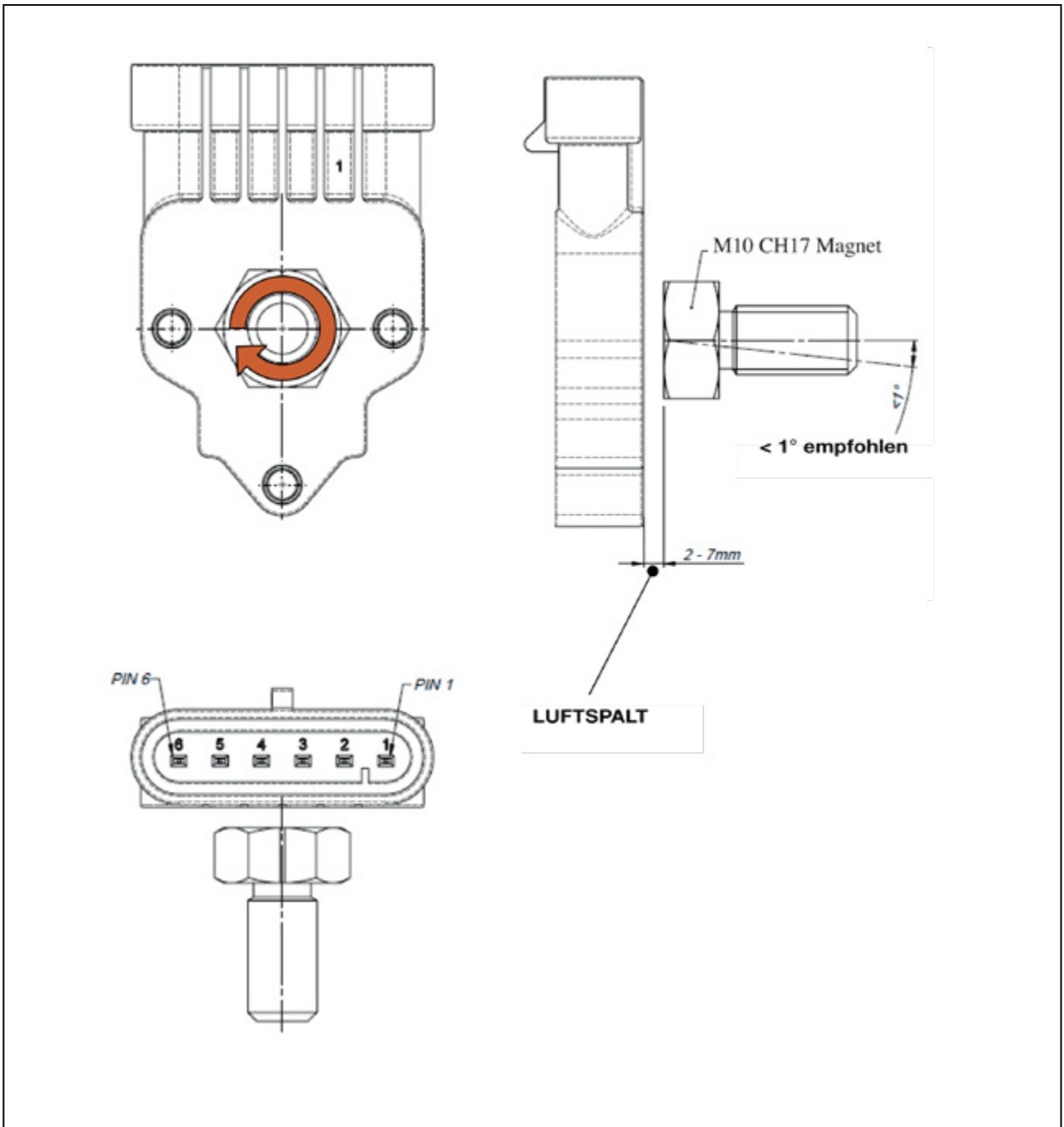
| AMP Superseal 6 P 282108-1 | Bedeutung |
|----------------------------|----------------------|
| 1 | OV (GND) |
| 2 | +Vs (+9 ... +36 Vdc) |
| 3 | n. a. |
| 4 | n. a. |
| 5 | CAN-L |
| 6 | CAN-H |

Hinweis: Der CAN-Bus muss terminiert sein.

Die zwischen CAN-H und CAN-L gemessene Impedanz muss 60 Ohm betragen. Daher muss das Kabel an jedem Ende der Busleitung an einen Widerstand von 120 Ohm angeschlossen werden.

Der Messaufnehmer ist intern nicht mit dem Widerstand von 120 Ohm terminiert.

Darauf achten, die Signalleitungen des CAN-Bus nicht zu verwechseln, da andernfalls die Kommunikation mit dem Messaufnehmer nicht möglich ist.



Bei den Anschlüssen ist die nachstehende Tabelle zugrunde zu legen:

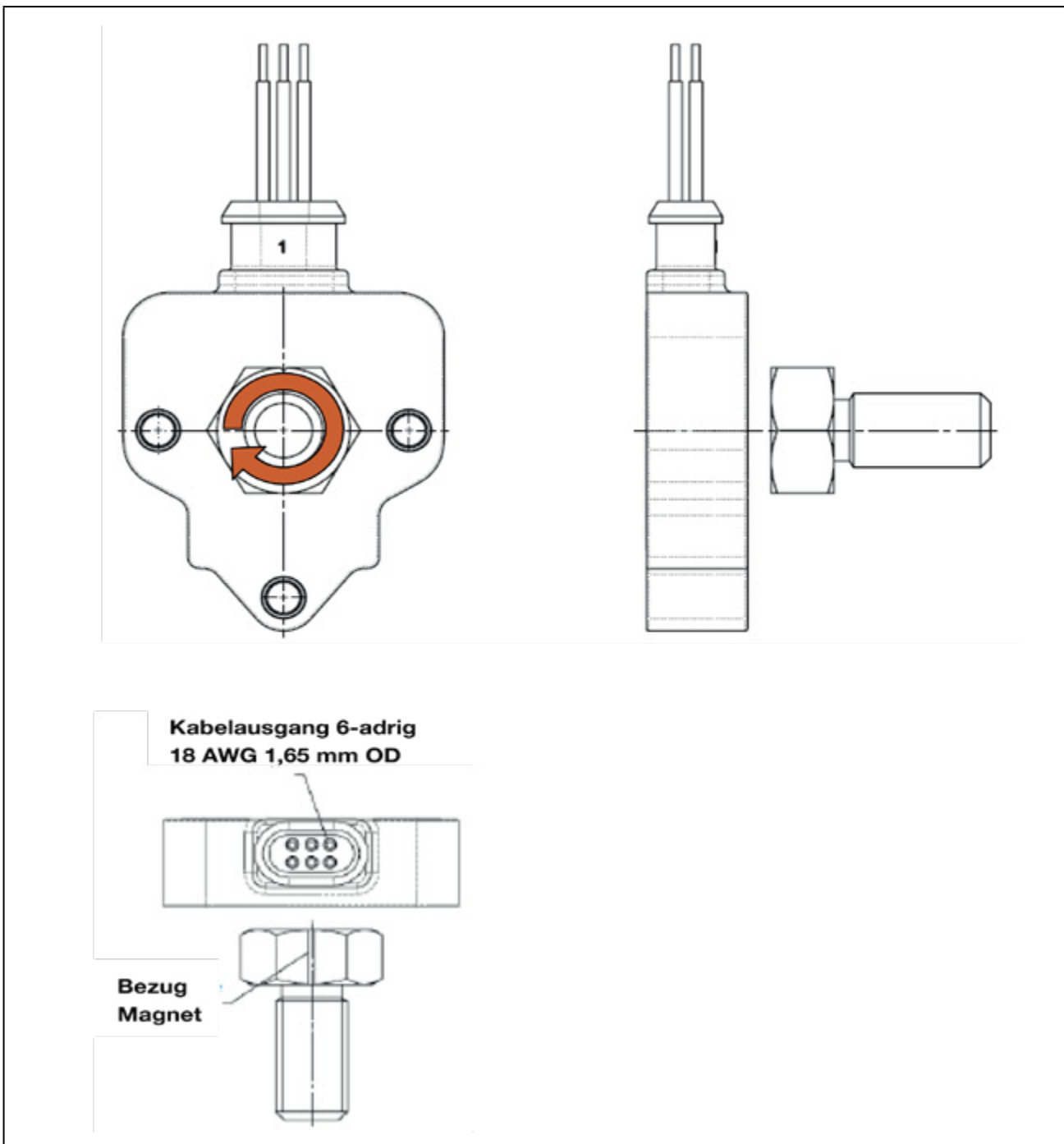
| Kabelausgang 6-adrig 18AWG 1,65mm OD | Meaning |
|--------------------------------------|----------------|
| SCHWARZ | ERDE |
| ROT | + VERSORGUNG 1 |
| GELB | n. a. |
| GRÜN | n. a. |
| BLAU | CAN-L |
| WEISS | CAN-H |

Hinweis: Der CAN-Bus muss terminiert sein.

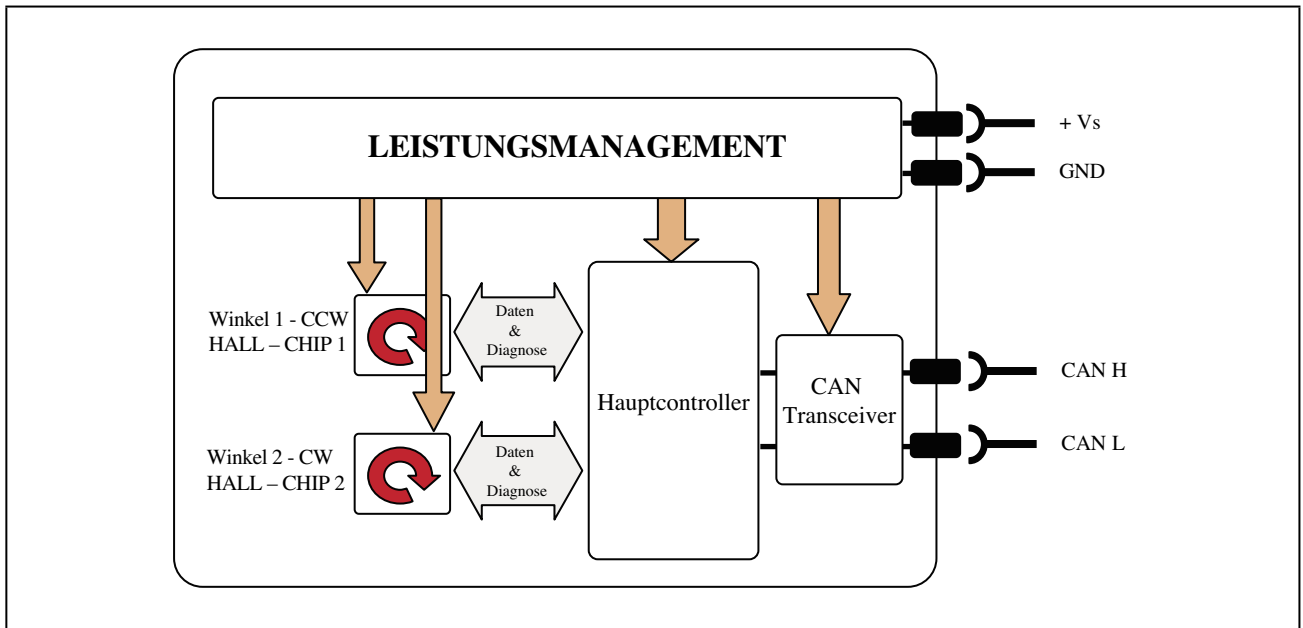
Die zwischen CAN-H und CAN-L gemessene Impedanz muss 60 Ohm betragen. Daher muss das Kabel an jedem Ende der Busleitung an einen Widerstand von 120 Ohm angeschlossen werden.

Der Messaufnehmer ist intern nicht mit dem Widerstand von 120 Ohm terminiert.

Darauf achten, die Signalleitungen des CAN-Bus nicht zu verwechseln, da andernfalls die Kommunikation mit dem Messaufnehmer nicht möglich ist.

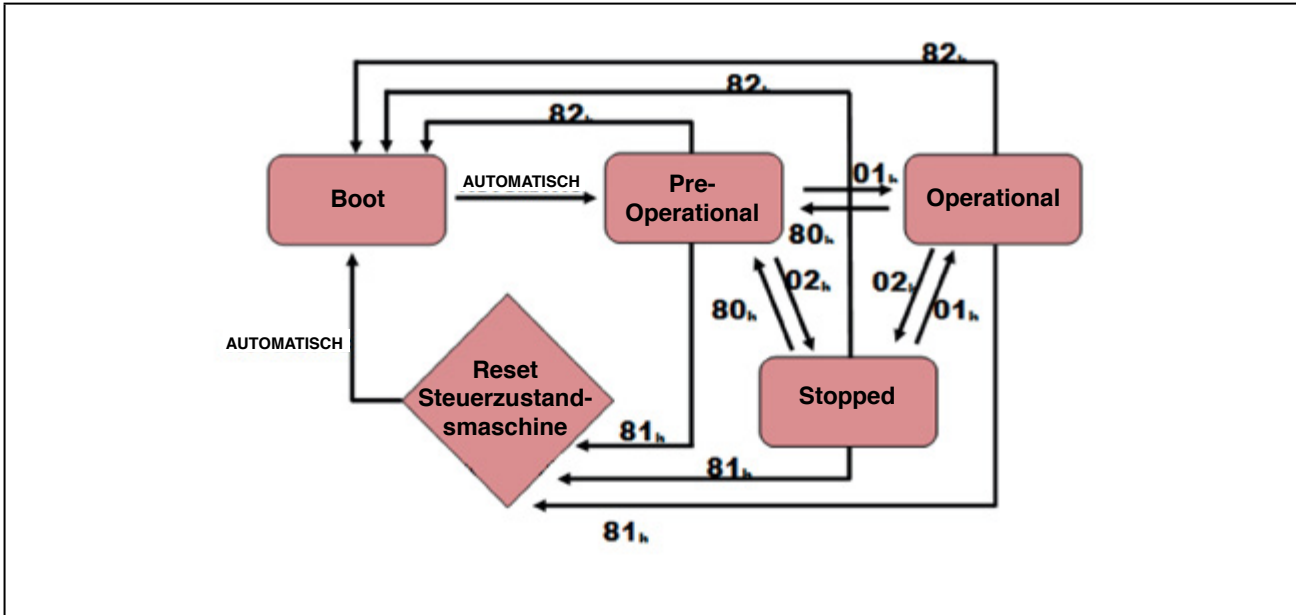


Funktionsblock:



3. NETWORK MANAGEMENT (NMT)

Das Gerät unterstützt die CANopen-Funktion für das Netzwerk-Management NMT Slave (Minimum Boot Up).



Jedes CANopen-Gerät enthält einen internen Server für das Netzwerkmanagement, der mit einem externen NMT-Master kommuniziert.

Ein Gerät im Netzwerk, im Allgemeinen der Host, kann als NMT-Master fungieren.

Jeder Netzwerkmanagement-Server der CANopen-Geräte steuert die Zustandsänderungen in seiner **Kommunikationszustandsmaschine** mit Hilfe von NMT-Nachrichten.

Diese ist abhängig von der Betriebszustandsmaschine jedes Knotens, die geräteabhängig ist und in der **Steuerzustandsmaschine** beschrieben ist.

Es ist wichtig, den Unterschied zwischen der Betriebszustandsmaschine und der Kommunikationszustandsmaschine eines CANopen-Geräts zu beachten.

CANopen-Sensoren und -E/A-Module haben beispielsweise völlig andere Betriebszustandsmaschinen als Servoantriebe.

Die **Kommunikationszustandsmaschine** entspricht bei allen CANopen-Geräten den Spezifikationen in DS301.

Die NMT-Nachrichten haben die höchste Priorität. Die fünf NMT-Nachrichten zum Steuern der Kommunikationszustandsmaschine enthalten jeweils 2 Datenbytes, welche die Knotennummer und einen Befehl für die Zustandsmaschine dieses Knotens enthalten.

Tabelle 1 zeigt die fünf unterstützten NMT-Nachrichten und **Tabelle 2** zeigt den richtigen Aufbau der Nachricht zum Versenden dieser Nachrichten.

Tabelle 1

| NMT-Nachricht | COB-ID | Datenbyte 1 | Datenbyte 2 |
|-----------------------|--------|-------------|-------------|
| Start Remote Node | 0 | 01h | Knoten-ID* |
| Stop Remote Node | 0 | 02h | Knoten-ID* |
| Pre-operational State | 0 | 80h | Knoten-ID* |
| Reset Node | 0 | 81h | Knoten-ID* |
| Reset Communication | 0 | 82h | Knoten-ID* |

* Knoten-ID = Adresse des Antriebs (von 1 bis 7Fh)

Table 2

| Arbitrationsfeld | Datenfeld | | | | | | | | |
|------------------|-----------|-----------------|-----------------|-----------------------|--------|--------|--------|--------|--------|
| | COB-ID | RTR | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 |
| 000h | 0 | Siehe Tabelle 1 | Siehe Tabelle 2 | nicht gesendete Bytes | | | | | |

4. BAUD RATE

Die Baudrate kann mittels Layer Setting Services and Protocol (LSS) und mittels SDO-Kommunikation (Index 0x5999) konfiguriert werden. Diese Parameter werden als Parameter LSS bezeichnet (Kennzeichnung LSS - PARA).

Die Baudrate ist werkseitig auf 250 kbit/s eingestellt.

Wichtiger Hinweis:

Wird dieser Parameterwert geändert, kann es zu Störungen im Netzwerk kommen! Diesen Dienst nur verwenden, wenn ein Gerät an das Netzwerk angeschlossen ist!

5. Knoten-ID und Auflösung

Die Knoten-ID kann mittels Layer Setting Services and Protocol (LSS) und mittels SDO-Kommunikation (Index 0x5999) konfiguriert werden. Diese Parameter werden als Parameter LSS bezeichnet (Kennzeichnung LSS - PARA).

Die Auflösung kann mit dem herstellerspezifischen Objekt 0x2100 konfiguriert werden.

Die Knoten-ID ist werkseitig auf 7F eingestellt.

Die Default-Auflösung beträgt 0,1°.

Wichtiger Hinweis:

Wird dieser Parameterwert geändert, kann es zu Störungen im Netzwerk kommen! Diesen Dienst nur verwenden, wenn ein Gerät an das Netzwerk angeschlossen ist!

6. EINSTELLUNG DER PARAMETER

Alle Parameter des Verzeichnisses (Objekte mit der Kennzeichnung PARA) können in einem speziellen Bereich des internen EEPROM mit Prüfsummenschutz gespeichert werden.

Die Spezialparameter LSS (Objekte mit der Kennzeichnung LSS-PARA), die Teil des Objektverzeichnisses sind, werden ebenfalls in einem speziellen Bereich des internen EEPROM mit Prüfsummenschutz gespeichert.

Aufgrund der internen Architektur des Mikrocontrollers sind die Parameter-Schreibzyklen auf 100.000 Zyklen begrenzt.

7. WIEDERHERSTELLEN DER WERKSEITIGEN PARAMETEREINSTELLUNGEN

Alle Objektverzeichnis-Parameter (Objekte mit der Kennzeichnung PARA) können mittels SDO-KOMMUNIKATION (INDEX 0x1011) wieder auf die werkseitigen Default-Werte zurückgesetzt werden.

8. HEARTBEAT

Der Heartbeat-Mechanismus wird bei diesem Gerät durch die zyklische Übertragung der Heartbeat-Nachricht durch den Heartbeat-Sender (Producer) festgelegt.

Eines oder mehrere Geräte im Netzwerk kennen diese Heartbeat-Nachricht.

Wenn der Heartbeat-Zyklus vom Heartbeat des Senders abweicht, wird die lokale Anwendung hierüber informiert.

Es ist zwingend ein Überwachungsmechanismus oder Heartbeat zu implementieren.

Das Gerät unterstützt die Funktion Heartbeat Producer. Die „Producer Heartbeat Time“ kann im Objekt 0x1017 hinterlegt werden.

Heartbeat-Nachricht

| COB-ID | Byte | 0 |
|---------------|--------|-------------|
| 700+Knoten-ID | Inhalt | NMT-Zustand |

9. FEHLERBEHANDLUNG

Prinzip

Die Fehler-Nachrichten (EMCY) werden von internen Fehlern des Geräts getriggert und erhalten die höchstmögliche Priorität, um sicherzustellen, dass sie ohne Verzögerung auf den Bus zugreifen können (EMCY Producer). Standardmäßig enthält die Fehler-Nachricht das Fehlerfeld mit vorgegebenen Fehlernummern und weiteren Informationen.

Fehlerverhalten (Objekt 0x4000)

Wenn ein schwerwiegender Defekt des Geräts festgestellt wird, gibt das Objekt 0x4000 an, auf welchen Zustand das Modul festgelegt werden muss:

- 0: Preoperational
- 1: kein Zustandswechsel (Default-Einstellung)
- 2: stopped

EMCY-Nachricht

Die COB-ID EMCY wird vom Objekt 0x1014 definiert. Die EMCY-Nachricht besteht aus 8 Bytes.

Sie enthält einen Fehlercode, den Inhalt des Objekts 0x1001 und 5 Bytes des herstellerspezifischen Fehlercodes.

Dieses Gerät verwendet nur das 1. Byte als herstellerspezifischen Fehlercode.

| Byte | Byte1 Byte2 | Byte3 | Byte4 | Byte5 | Byte6 Byte7 Byte8 |
|--|--------------------------|---|---|--|--|
| Beschreibung | Fehlercode ¹⁾ | Fehlerregister (Objekt 0x1001 ²⁾) | Herstellerspezifischer Fehlercode (Objekt 0x4001) | Herstellerspezifischer Fehlercode (immer 0x00) | Herstellerspezifischer Fehlercode NICHT IMPLEMENTIERT (immer 0xFF) |
| ¹⁾ 0x1000 als Generischer Fehler ²⁾ Immer 0 | | | | | |

Unterstützte herstellerspezifische Fehlercodes (Objekt 0x4001)

| Herstellerspezifischer Fehlercode (Bit-Feld) | Beschreibung |
|--|---|
| 0x01 | Winkel 1 Chip1 interner Fehler |
| 0x02 | Winkel 2 Chip2 interner Fehler |
| 0x04 | Winkelabweichung (zwischen Winkel 1 und Winkel 2), Objekt 0x2103 NICHT IMPLEMENTIERT |
| 0x10 | Programm-Prüfsummenfehler |
| 0x40 | Prüfsummenfehler Parameter LSS |
| 0x83 | Magnetfeld zu breit oder Magnetfeld zu schmal |

10. SDO KOMMUNIKATION

Das Gerät unterstützt die SDO-Server-Funktionalität.

Mittels Service Data Object (SDO) kann auf die Einträge des Objektverzeichnisses eines Geräts zugegriffen werden. Da diese Einträge Daten beliebiger Längen und beliebigen Datentyps enthalten können, werden SDOs dazu verwendet, mehrere Datensätze von einem Client an einen Server und umgekehrt zu übertragen.

Struktur der SDO-Anfrage vom Master

| COB-ID | DLC | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 | Byte8 |
|------------------|-----|-------|-------|-------|----------|-------|-------|-------|-------|
| 600h + Knoten-ID | 8 | CMD | Index | | Subindex | Daten | Daten | Daten | Daten |

Struktur der SDO-Antwort vom Slave

| COB-ID | DLC | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 | Byte8 |
|------------------|-----|-------|-------|-------|----------|-------|-------|-------|-------|
| 580h + Knoten-ID | 8 | RES | Index | | Subindex | Daten | Daten | Daten | Daten |

Schreibzugriff, Datenübertragung vom Host zum Slave

Jeder Zugriff auf das Objektverzeichnis wird vom Slave auf Gültigkeit geprüft. Jeder Schreibzugriff auf nicht existierende Objekte, auf Nur-Lese-Objekte oder mit einem nicht entsprechenden Datenformat wird verwehrt und mit einer entsprechenden Fehlernachricht beantwortet.

Das **CMD** bestimmt die Richtung der Datenübertragung und die Größe des Datenobjekts:

23 hex Senden von 4-Byte-Daten (die Bytes 5...8 enthalten einen 32-Bit-Wert)

2B hex Senden von 2-Byte-Daten (die Bytes 5, 6 enthalten einen 16-Bit-Wert)

2F hex Senden von 1-Byte-Daten (Byte 5 enthält einen 8-Bit-Wert)

Antworten des Slave:

RES Antwort des Slave:

60 hex Daten erfolgreich übertragen

80 hex Fehler

Lesezugriff, Datenübertragung vom Slave zum Host

Bei jedem Lesezugriff auf nicht existierende Objekte wird eine Fehlernachricht zurückübertragen.

Das **CMD** bestimmt die Richtung der Datenübertragung:

40 hex Lesezugriff (in jedem Fall)

Antworten des Slave:

RES Antwort des Slave:

42 hex Vom Knoten als Antwort auf einen Lesebefehl mit 4 oder weniger Datenbytes verwendete Bytes

43 hex Bytes 5...8 enthalten einen 32-Bit-Wert

4B hex Bytes 5, 6 enthalten einen 16-Bit-Wert

4F hex Byte 5 enthält einen 8-Bit-Wert

80 hex Fehler

11. PDO-KOMMUNIKATION und Winkelberechnung

Übertragung PDO #0

Dieses PDO überträgt asynchron den Wert der Winkelstellung des Sensors.

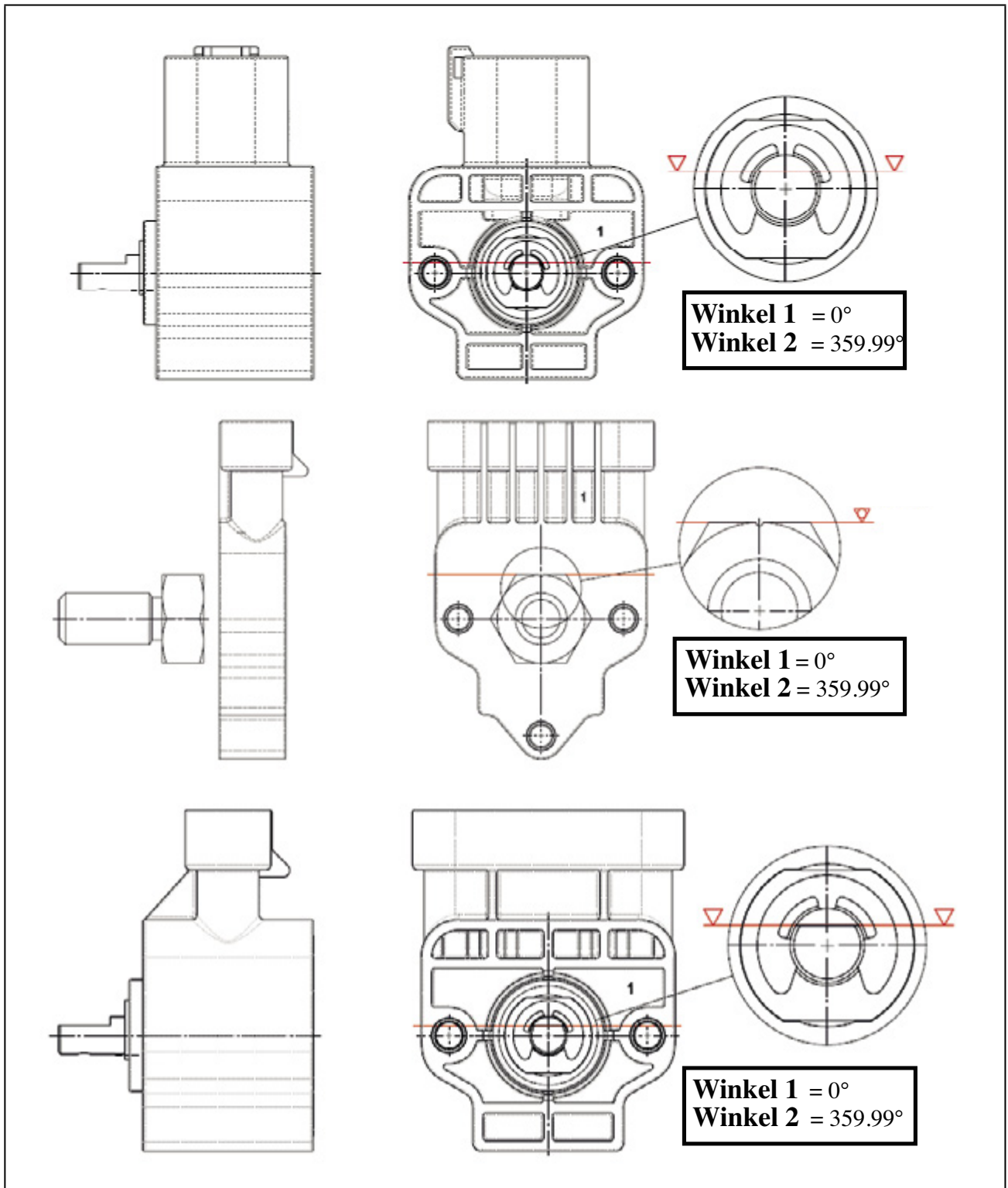
Das Tx PDO # 0 wird zyklisch gesendet, wenn der zyklische Timer (Objekt 0x1800.5) auf einen Wert > 0 programmiert ist. Die Werte zwischen 1 ms und 65535 ms müssen durch die Parametereinstellungen gewählt werden.

Das Tx PDO # 0 wird beim Wechsel in den Zustand „Operational“ gesendet.

| Byte | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 Byte6 Byte7 | Byte8 |
|---|---|--|---|--|-------------------------|-------------------------------|
| Beschreibung | WINKEL 1 Objekt (0x2110.1) High-Byte | WINKEL 1 Objekt (0x2110.2) Low-Byte | WINKEL 2 Objekt (0x2110.3) High-Byte | WINKEL 2 Objekt (0x2110.4) Low-Byte | (0xFF) | Fehlercode (Objekt 0x4001) |
| Tx PDO #0 mit vorgegebenem Mapping mit Objekt 0x5001 = 0 (Big-Endian) | | | | | | |

| Byte | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 Byte6 Byte7 | Byte8 |
|--|--|---|--|---|-------------------------|-------------------------------|
| Beschreibung | WINKEL 1 Objekt (0x2110.1) Low-Byte | WINKEL 1 Objekt (0x2110.2) High-Byte | WINKEL 2 Objekt (0x2110.3) Low-Byte | WINKEL 2 Objekt (0x2110.4) High-Byte | (0xFF) | Fehlercode (Objekt 0x4001) |
| Tx PDO #0 mit vorgegebenem Mapping mit Objekt 0x5001 = 1 (Little-Endian) | | | | | | |

Auf der nächsten Seite wird ein Beispiel eines PDO-Mappings für Winkel 1= +0,0° und Winkel 2= 359,9 ° dargestellt.



MESSAUFLÖSUNG $\pm 0.1^\circ$ (siehe das herstellerspezifische Objekt 0x2100 und Beispiel (7) am Ende dieser Betriebsanleitung)

Beispiel eines PDO-Mappings für:

Winkel 1 = 0.0° und Winkel 2 = 359.9° (Knoten-ID = 02h, Auflösung 0.1° , Null-Grad-Punkt = 0.0° , GUZ und Big-Endian)

| ID | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 | Byte8 |
|------|-------|-------|-------|-------|-------|-------|-------|-------|
| 182h | 00h | 00h | 0Eh | 0Fh | FFh | FFh | FFh | 00h |

Winkel 1:

Byte 1 MSB (00h) = 00h Byte 2 LSB (00h) = 00h

Winkel 1 = 0000h dezimal 0d (Auflösung 0.1°) = 0.0°

Winkel 2:

Byte 3 MSB (00h) = 0Eh Byte 4 LSB (00h) = 0Fh

Winkel 2 = 0E0Fh dezimal 3599d (Auflösung 0.1°) = 359.9°

MESSAUFLÖSUNG $\pm 0.01^\circ$ (siehe das herstellerspezifische Objekt 0x2100 und Beispiel (8) am Ende dieser Betriebsanleitung)

Beispiel eines PDO-Mappings für:

Winkel 1 = 0.0° und Winkel2 = 359.9° (Knoten-ID = 02h, resolution 0.1° , zero degree point = 0.0° , CCW and big endian)

| ID | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 | Byte8 |
|------|-------|-------|-------|-------|-------|-------|-------|-------|
| 182h | 00h | 00h | 8Ch | 9Fh | FFh | FFh | FFh | 00h |

Winkel 1:

Byte 1 MSB (00h) = 00h Byte 2 LSB (00h) = 00h

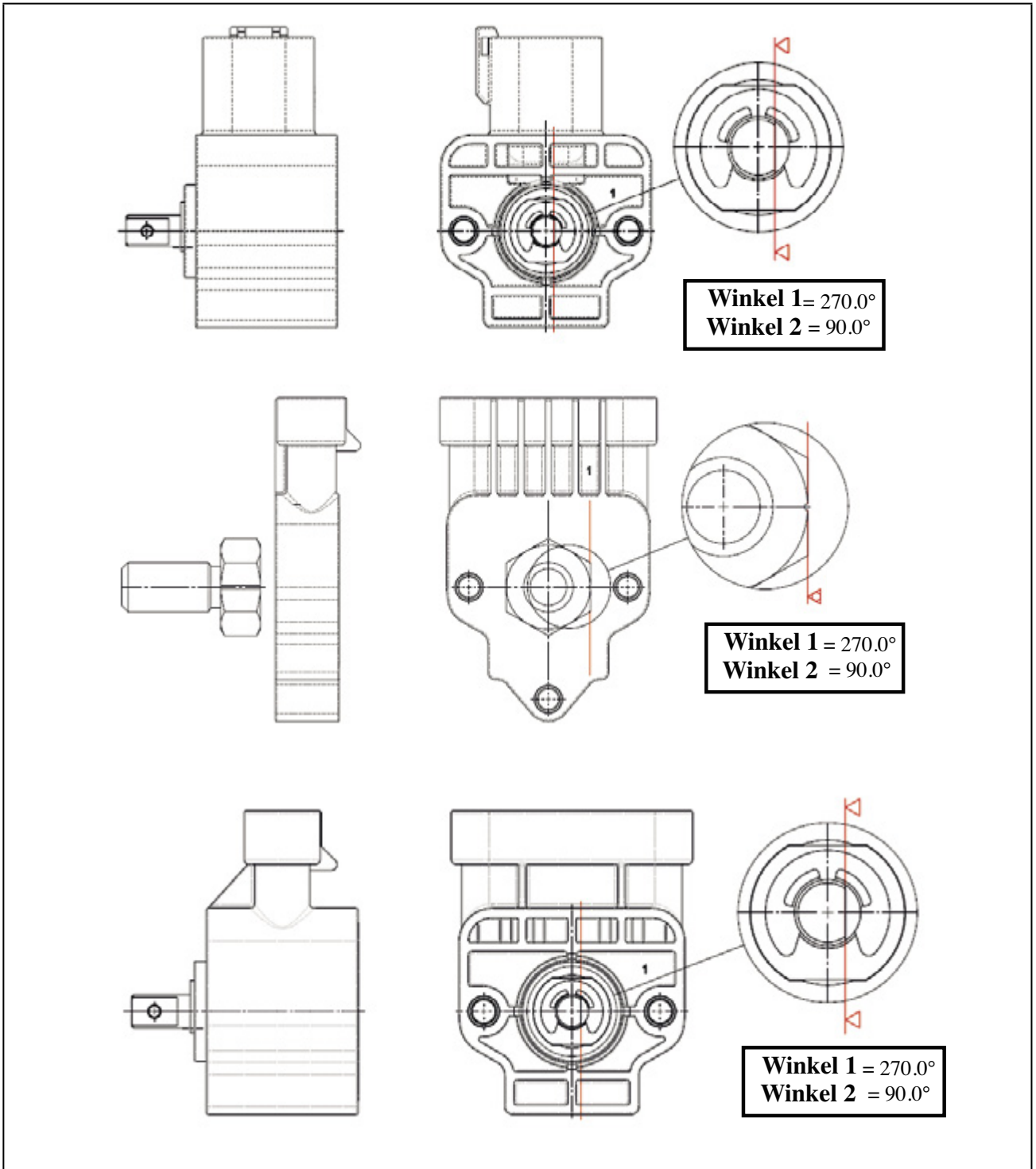
Winkel 1 = 0000h dezimal 0d (Auflösung 0.01°) = 0.00°

Winkel 2:

Byte 3 MSB (00h) = 8Ch Byte 4 LSB (00h) = 9Fh

Winkel 2 = 8C9F dezimal 35999d (Auflösung 0.01°) = 359.99°

Die Seite zeigt ein Beispiel eines PDO-Mappings. Beispiel eines PDO-Mappings für Winkel 1 = 270,0° und Winkel 2 = 90,0°.



MESSAUFLÖSUNG $\pm 0,1^\circ$ (siehe das herstellerspezifische Objekt 0x2100 und Beispiel (7) am Ende dieser Betriebsanleitung)

Beispiel eines PDO-Mappings für Winkel 1 = 270,0° und Winkel 2 = 90° (Knoten-ID = 02h, Auflösung 0,1°, Null-Grad-Punkt = 0,0°, GUZ und Big-Endian)

| ID | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 | Byte8 |
|------|-------|-------|-------|-------|-------|-------|-------|-------|
| 182h | 0Ah | 8Ch | 03h | 84h | FFh | FFh | FFh | 00h |

Winkel 1:

Byte 1 MSB (00h) = 0Ah Byte 2 LSB (00h) = 8Ch Winkel 1 = 0A8Ch, dezimal 0d (Auflösung 0,1°) = 270,0°

Winkel 2:

Byte 3 MSB (00h) = 03h Byte 4 LSB (00h) = 84h Winkel 2 = 0384h, dezimal 900d (Auflösung 0,1°) = 90,0°

MESSAUFLÖSUNG $\pm 0.01^\circ$ (siehe das herstellerspezifische Objekt 0x2100 und Beispiel (8) am Ende dieser Betriebsanleitung)

Beispiel eines PDO-Mappings für Winkel 1 = 270,00° und Winkel 2 = 90,00° (Knoten-ID = 02h, Auflösung 0,1°, Null-Grad-Punkt = 0,0°, GUZ und Big-Endian)

| ID | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 | Byte8 |
|------|-------|-------|-------|-------|-------|-------|-------|-------|
| 182h | 69h | 78h | 23h | 28h | FFh | FFh | FFh | 00h |

Winkel 1:

Byte 1 MSB (00h) = 69h Byte 2 LSB (00h) = 78h Winkel 1 = 6978h, dezimal 27000d (Auflösung 0.01°) = 270.00°

Winkel 2:

Byte 3 MSB (00h) = 23h Byte 4 LSB (00h) = 28h Winkel 2 = 2328h, dezimal 9000d (Auflösung 0.01°) = 90.00°

Die Abbildung 1 veranschaulicht die Winkelberechnung für WINKEL 1 und WINKEL 2.

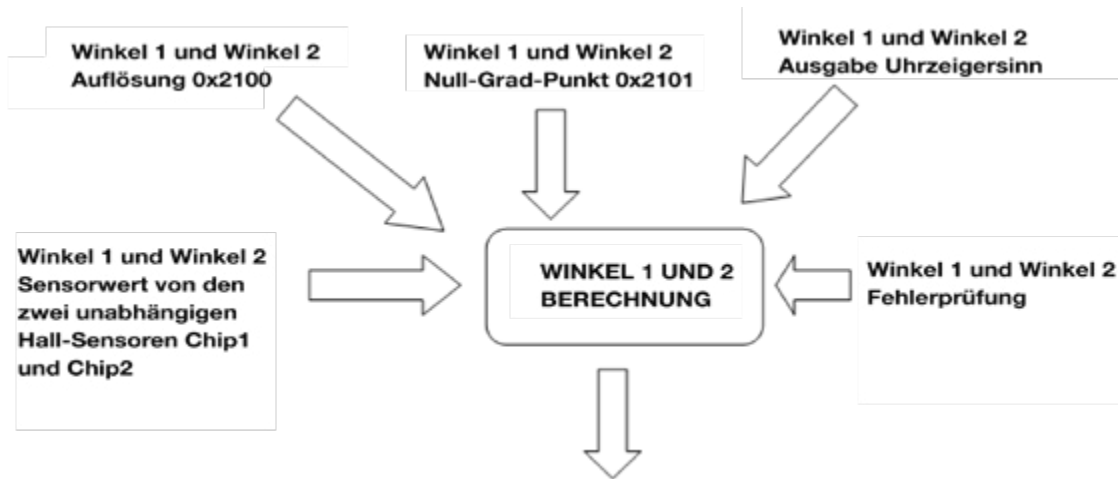


Abbildung 1 Winkelberechnung

| ID | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 | Byte8 |
|------|-------|-------|-------|-------|-------|-------|-------|-------|
| 182h | 00h | 00h | 0Eh | 0Fh | FFh | FFh | FFh | 00h |

Winkel 1 Winkel 2

Wichtiger Hinweis: Die zwei Winkel 1 und 2 werden aufgrund der redundanten Konfiguration unabhängig voneinander ermittelt (da bei den berührungslosen Drehsensoren der Serie GRA/GRN zwei HALL-Chips eingebaut sind).

12. ÜBERBLICK ÜBER DIE EIGENSCHAFTEN VON CANopen

Kommunikationsprofil

Die kritischen Parameter für die Kommunikation werden vom Kommunikationsprofil festgelegt. Dieser Bereich ist allen CANopen-Geräten gemein.

| Index | Sub Index | Name | Typ | Zugriff | Defaultwert | Bemerkungen |
|-------|----------------------|--|-------------|---------|------------------|---|
| 1000h | | Geräteprofil | Unsigned 32 | Ro | 0x00000000 | Kein standardisiertes Geräteprofil verwendet |
| 1001h | | Fehlerregister | Unsigned 8 | Ro | 0x00 | Immer NULL |
| 1008h | | Name des Geräteherstellers | String | Ro | “GRA” or “GRN” | Siehe den Produktkatalog von GEFRAN: GRA: Berührungsloser Winkelsensor mit Welle GRN: Berührungsloser Winkelsensor ohne Welle |
| 1009h | | Hardware-Version Hersteller | String | Ro | “1.00” | |
| 100Ah | | Software-Version Hersteller | String | Ro | “1.14” | |
| 1010h | 0 | Anzahl Einträge | Unsigned 8 | Ro | 1 | „Speichern“ (0x65766173) zum Speichern aller Parameter (Objekte mit der Kennzeichnung PARA) |
| | 1 | Speicherung aller Parameter | Unsigned 32 | Rw | | |
| 1011h | 0 | Wiederherstellen der werkseitigen Parametereinstellungen | Unsigned 8 | Ro | | „load“ (0x64616F6C) zum Wiederherstellen aller Parameter (Objekte mit Kennzeichnung PARA und LSS-PARA). |
| | 1 | Wiederherstellen aller Parameter | Unsigned 32 | Rw | | |
| 1014h | 0 | Emergency ID | Unsigned 32 | Rw | 0x80 + Knoten-ID | |
| 1017h | 0 | Producer Heartbeat Time | Unsigned 16 | Rw | 0 | Min= 0 und Max=65535 Mit Einheit = 1ms Wenn 0: NICHT VERWENDET |
| 1018h | 0 | Identity Object | Unsigned 8 | Ro | 4 | Siehe “Gefran Product Overview CANopen” Gefran Vendor ID:0x0000093 |
| | 1 | Vendor ID | Unsigned 32 | Ro | 0x0000093 | |
| | 2 | Produktcode | Unsigned 32 | Ro | 0x0000064 | |
| | 3 | Versionsnummer | Unsigned 32 | Ro | 0x0000001 | |
| | 4 | Seriennummer | Unsigned 32 | Ro | 0x0000000 | |
| 1200h | Server SDO-Parameter | | | | | |
| | 0 | Anzahl Einträge | Unsigned 8 | Ro | 2 | |
| | 1 | COB-ID Client to Server (Rx) | Unsigned 32 | Ro | 0x600+ Knoten-ID | |
| | 2 | COB-ID Server to Server (Tx) | Unsigned 32 | Ro | 0x580+ Knoten-ID | |
| 1800h | 0 | 1° Parameter Übertragung PDO | Unsigned 8 | Ro | Default valueRo | CommentsRo |
| | 1 | COB-ID | Unsigned 32 | Ro | 180h + Knoten-ID | Comments180h Nodoid |
| | 2 | Übertragungsart | Unsigned 8 | Rw | 254 | Asynchrone Übertragung. |
| | 3 | Sperrzeit | Unsigned 16 | Ro | 0 | Min= 0 & Max=65535 |
| | 4 | Reserviert | // | // | 14 | Min 14 Max14 with unit 1ms |
| | 5 | Timer | Unsigned 16 | Rw | 100 | Min= 4 & Max=65535 |

| Index | Sub Index | Name | Typ | Zugriff | Defaultwert | Bemerkungen |
|-------|-------------------------------|-------------------------------|-------------|------------|-------------|---|
| 1A00h | Mapping-Parameter Tx PDO | | | | | Objekte 0x2110.1 0x2110.2 0x2110.3 0x2110.4 0x2110.5 0x2110.6 0x2110.7 0x4001 |
| | 0 | Anzahl Einträge | Unsigned 8 | Ro | 8 | |
| | 1 | 1 st Mapped Object | Unsigned 32 | Ro | 0x21100108 | |
| | 2 | 2 nd Mapped Object | Unsigned 32 | Ro | 0x21100208 | |
| | 3 | 3 st Mapped Object | Unsigned 32 | Ro | 0x21100308 | |
| | 4 | 4 th Mapped Object | Unsigned 32 | Ro | 0x21100408 | |
| | 5 | 5 th Mapped Object | Unsigned 32 | Ro | 0x21100508 | |
| | 6 | 6 th Mapped Object | Unsigned 32 | Ro | 0x21100608 | |
| | 7 | 7 th Mapped Object | Unsigned 32 | Ro | 0x21100708 | |
| 8 | 8 th Mapped Object | Unsigned 32 | Ro | 0x40010008 | | |

Ro = nur Lese-Zugriff auf den Parameter

Rw = Lese- und Schreibzugriff auf den Parameter

Wo = nur Schreibzugriff auf den Parameter

Herstellerspezifisches Objektprofil

Dieser Abschnitt enthält die Indizes des herstellerspezifischen Profils für den Messaufnehmer.

| Index | Sub Index | Name | Typ | Zugriff | Defaultwert | Bemerkungen |
|-------|-----------------------------|-------------------------------|-------------|---------|-------------|---|
| 2000h | Winkelwert | | | | | Winkel 1 des Sensors und Winkel 2 des Sensors in einem einzigen Chip mit Skalierendwert 360° und Auflösung von ca. 0,022°/Bit Min.= 0 und Max.=16383 |
| | 0 | Anzahl Einträge | Unsigned 8 | Ro | 2 | |
| | 1 | Wert von Winkel 1 des Sensors | Unsigned 16 | Ro | 0 | |
| 2001h | Funktion FILTER des Winkels | | | | | Min.= 0 und Max.=255 0: nicht verwendet NICHT IMPLEMENTIERT |
| | 0 | Anzahl Einträge | Unsigned 8 | Ro | 2 | |
| | 1 | Winkel 1 FILTER | Unsigned 8 | Rw | 0 | |
| 2011h | Winkelsensor Prozess-Daten | | | | | Winkel 1 und 2 Prozess-Daten gleichzeitig gesendet |
| | 0 | Anzahl Einträge | Unsigned 8 | Ro | 3 | |
| | 1 | Winkel 1 Prozess-Daten | Unsigned 16 | Ro | 0 | |
| | 2 | Winkel 2 Prozess-Daten | Unsigned 16 | Ro | 0 | |
| 2100h | AUFLÖSUNG des Winkels | | | | | Zulässige Werte der AUFLÖSUNG für Winkel 1 und Winkel 2: 1000d: 1 Grad/Bit 100d: 0,1 Grad/Bit 22d: 0,01 Grad/Bit (14 Bit effektive Auflösung 0,02°) |
| | 0 | Anzahl Einträge | Unsigned 8 | Ro | 2 | |
| | 1 | Auflösung von Winkel 1 | Unsigned 16 | Rw | 100 | |
| 2101h | Nullpunkt des Winkels | | | | | Der Nullpunkt von Winkel 1 und der Nullpunkt von Winkel 2 sind auf den zulässigen maximalen Gradwert zu beziehen. Min.= 0 und Max.=16383 |
| | 0 | Anzahl Einträge | Unsigned 8 | Ro | 2 | |
| | 1 | Nullpunkt von Winkel 1 | Unsigned 16 | Ro | 0 | |
| | 2 | Nullpunkt von Winkel 2 | Unsigned 16 | Ro | 0 | |

| Index | Sub Index | Name | Typ | Zugriff | Defaultwert | Bemerkungen |
|-------|-----------------------------|------------------------|------------|---------|-------------|--|
| 2102h | Winkel Uhrzeigersinn | | | | | 0: CCW 1: CW |
| | 0 | Anzahl Einträge | Unsigned 8 | Ro | 2 | |
| | 1 | Winkel 1 Uhrzeigersinn | Unsigned 8 | Rw | 0 | |
| | 2 | Winkel 2 Uhrzeigersinn | Unsigned 8 | Rw | 1 | |
| 2103h | | Maxi Winkeldifferenz | | | 0 | NICHT IMPLEMENTIERT |
| 2110h | Werte Winkel 1 und Winkel 2 | | | | | Einheit: 0x2100 MSB bei Verwendung der Codierung PDO Big Endian (Index 0x5001) Min.=0 und Max.=255 Einheit: 0x2100 MSB bei Verwendung der Codierung PDO Little Endian (Index 0x5001) Min.=0 und Max.=255 Einheit: 0x2100 MSB bei Verwendung der Codierung PDO Big Endian (Index 0x5001) Min.=0 und Max.=255 Einheit: 0x2100 MSB bei Verwendung der Codierung PDO Little Endian (Index 0x5001) Min.=0 und Max.=255 |
| | 0 | Anzahl Einträge | Unsigned 8 | Ro | 4 | |
| | 1 | Ausgangswert | Unsigned 8 | Ro | 0 | |
| | 2 | Winkel 1 Byte 0 | Unsigned 8 | Ro | 0 | |
| | 3 | Ausgangswert | Unsigned 8 | Ro | 0 | |
| | 4 | Winkel 1 Byte 1 | Unsigned 8 | Ro | 0 | |

Ro = nur Lese-Zugriff auf den Parameter
Rw = Lese- und Schreibzugriff auf den Parameter
Wo = nur Schreibzugriff auf den Parameter

Herstellerspezifisches Objektprofil

Dieser Abschnitt enthält die Indizes des herstellerspezifischen Profils für den Messaufnehmer.

| Index | Sub Index | Name | Typ | Zugriff | Defaultwert | Bemerkungen |
|-------|-----------|--|------------|---------|-------------|---|
| 4000h | | Fehlerverhalten - PARA | Unsigned 8 | Rw | 1 | 0: Preoperational 1: kein Zustandswechsel 2: stopped Min.= 0 und Max.=255 |
| 4001h | | Fehlercode | Unsigned 8 | Ro | 0 | 0: kein Fehler Min.=0 und Max.=255 |
| 5000h | | NMT automatischer Start nach Einschaltung - PARA | Unsigned 8 | Rw | 1 | 0: nicht aktiviert 1: aktiviert Min.=0 und Max.=1 |
| 5001h | | verwendete Standardcodierung PDO - PARA | Unsigned 8 | Rw | 0 | 0: Big Endian 1: Little Endian Min.=0 und Max.=1 |

| Index | Sub Index | Name | Typ | Zugriff | Defaultwert | Bemerkungen |
|-------|---------------|--|-------------|---------|-------------|--|
| 5999h | Parameter LSS | | | | | |
| | 0 | Anzahl Einträge | Unsigned 8 | Ro | 3 | |
| | 1 | Übertragungsgeschwindigkeit - LSS-PARA | Unsigned 16 | Rw | 250 | Mögliche Werte: 50 kbit/s 125 kbit/s 250 kbit/s 500 kbit/s 800 kbit/s 1000 kbit/s Min=50 und Max=1000 WICHTIGER HINWEIS: Diesen Dienst nur verwenden, wenn ein Gerät an das Netzwerk angeschlossen ist |
| | 2 | Node-ID - LSS-PARA | Unsigned 8 | Rw | 2 | Min=1 und Max=127 WICHTIGER HINWEIS: Diesen Dienst nur verwenden, wenn ein Gerät an das Netzwerk angeschlossen ist |
| | 3 | Store - LSS-PARA | Unsigned 32 | Wo | | „Speichern“ (0x65766173) zum Speichern aller Parameter LSS (Objekt mit Kennzeichnung LSS-PARA) WICHTIGER HINWEIS: Diesen Dienst nur verwenden, wenn ein Gerät an das Netzwerk angeschlossen ist. |

Ro = nur Lese-Zugriff auf den Parameter

Rw = Lese- und Schreibzugriff auf den Parameter

Wo = nur Schreibzugriff auf den Parameter

13. KOMMUNIKATIONSBEISPIELE

Beispiel 1) Ändern der Baudrate-Einstellung von 250 kBaud nach 500 kBaud

Mit dem Service Data Object (SDO) wird auf die Einträge des Objektverzeichnisses des Geräts zugegriffen. Da diese Einträge Daten beliebiger Länge und beliebigen Typs enthalten können, kann das SDO zum Übertragen von mehreren Datensätzen von einem Client an einen Server und umgekehrt verwendet werden.

SDO-Struktur - Anfrage vom Master

| COB-ID | DLC | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 | Byte8 |
|---------------|-----|-------|-------|-----------|-------|-------|-------|-------|-------|
| 600+Knoten-ID | 8 | CMD | Index | Sub-Index | Daten | | | | |

Das **CMD** bestimmt die Richtung der Datenübertragung und die Größe des Datenobjekts:

23 hex Senden von 4-Byte-Daten (die Bytes 5...8 enthalten einen 32-Bit-Wert)

2B hex Senden von 2-Byte-Daten (die Bytes 5, 6 enthalten einen 16-Bit-Wert)

2F hex Senden von 1-Byte-Daten (Byte 5 enthält einen 8-Bit-Wert)

SDO-Struktur - Antwort vom Slave

| COB-ID | DLC | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 | Byte8 |
|---------------|-----|-------|-------|-----------|-------|-------|-------|-------|-------|
| 580+Knoten-ID | 8 | CMD | Index | Sub-Index | Daten | | | | |

RES Antwort des Slave:

60 hex Daten erfolgreich übertragen

80 hex Fehler

Schreiben (im Beispiel Knoten-ID = 0x03)

| ID | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 | Byte8 |
|------|-------|-------|-------|-------|-------|-------|-------|-------|
| 603h | 2Bh | 99h | 59h | 01h | F4h | 01h | 00h | 00h |

Objekt:

| | | | | | | | | |
|-------|---|--|-------------|----|-----|--|--|--|
| 5999h | | Parameter LSS | | | | | | |
| | 0 | Anzahl Einträge | Unsigned 8 | Ro | 3 | | | |
| | 1 | Übertragungsgeschwindigkeit – LSS-PARA | Unsigned 16 | Rw | 500 | Mögliche Werte: 50 kbit/s 125 kbit/s 250 kbit/s 500 kbit/s 800 kbit/s 1000 kbit/s Min=50 und Max=1000 WICHTIGER HINWEIS: Diesen Dienst nur verwenden, wenn ein Gerät an das Netzwerk angeschlossen ist | | |
| | 2 | Knoten-ID – LSS-PARA | Unsigned 8 | Rw | 2 | Min=1 und Max=127 WICHTIGER HINWEIS: Diesen Dienst nur verwenden, wenn ein Gerät an das Netzwerk angeschlossen ist | | |
| | 3 | Store - LSS-PARA | Unsigned 32 | Wo | | "Speichern" (0x65766173) zum Speichern aller Parameter LSS (Objekt mit Kennzeichnung LSS- PARA) WICHTIGER HINWEIS: Diesen Dienst nur verwenden, wenn ein Gerät an das Netzwerk angeschlossen ist | | |

In der nachstehenden Tabelle sind die unterstützten Baudraten angegeben:

| Byte5 | Byte6 | BaudRate |
|-------|-------|----------|
| 32h | 00h | 50Kbaud |
| 7Dh | 00h | 125Kbaud |
| FAh | 00h | 250Kbaud |
| F4h | 01h | 500Kbaud |
| 20h | 03h | 800Kbaud |
| E8h | 03h | 1Mbaud |

Die Antwort nach dem ordnungsgemäßen Speichern ist:

| ID | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 | Byte8 |
|------|-------|-------|-------|-------|-------|-------|-------|-------|
| 583h | 60h | 99h | 59h | 01h | 00h | 00h | 00h | 00h |

Zum Speichern der neuen Baudrate den Befehl „Speichern“ wie nachstehend angegeben schreiben:

Schreiben (im Beispiel Knoten-ID = 0x03)

| ID | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 | Byte8 |
|------|-------|-------|-------|-------|-------|-------|-------|-------|
| 603h | 23h | 99h | 59h | 03h | 73h | 61h | 76h | 65h |

Hinweis: Der Befehl „Speichern“ wird durch Senden des nachstehenden Codes gegeben:

| | | | |
|-----|-----|-----|-----|
| 73h | 61h | 76h | 65h |
|-----|-----|-----|-----|

Wobei gilt:

73h = ASCII code "s"

61h = ASCII code "a"

76h = ASCII code "v"

65h = ASCII code "e"

Die Antwort nach dem ordnungsgemäßen Speichern ist:

| ID | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 | Byte8 |
|------|-------|-------|-------|-------|-------|-------|-------|-------|
| 583h | 60h | 99h | 59h | 03h | 00h | 00h | 00h | 00h |

WICHTIGER HINWEIS:

Damit die neuen Einträge gültig werden, muss ein Reset durchgeführt werden (das Modul kurz ausschalten).

Beispiel 2) Ändern der Knoten-ID von 0x03h (3d) nach 0x06h (6d)

Mit dem Service Data Object (SDO) wird auf die Einträge des Objektverzeichnisses des Geräts zugegriffen. Da diese Einträge Daten beliebiger Länge und beliebigen Typs enthalten können, kann das SDO zum Übertragen von mehreren Datensätzen von einem Client an einen Server und umgekehrt verwendet werden.

SDO-Struktur - Anfrage vom Master

| COB-ID | DLC | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 | Byte8 |
|-------------|-----|-------|-------|-------|-----------|-------|-------|-------|-------|
| 600+Nodo-ID | 8 | CMD | Index | | Sub-Index | Daten | | | |

Das **CMD** CMD bestimmt die Richtung der Datenübertragung und die Größe des Datenobjekts:

23 hex Senden von 4-Byte-Daten (die Bytes 5...8 enthalten einen 32-Bit-Wert)

2B hex Senden von 2-Byte-Daten (die Bytes 5, 6 enthalten einen 16-Bit-Wert)

2F hex Senden von 1-Byte-Daten (Byte 5 enthält einen 8-Bit-Wert)

SDO-Struktur - Antwort vom Slave

| COB-ID | DLC | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 | Byte8 |
|-------------|-----|-------|-------|-------|-----------|-------|-------|-------|-------|
| 580+Nodo-ID | 8 | CMD | Index | | Sub-Index | Daten | | | |

RES Antwort des Slave:

60 hex Daten erfolgreich übertragen

80 hex Fehler

Schreiben (im Beispiel Knoten-ID = 0x03)

| ID | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 | Byte8 |
|------|-------|-------|-------|-------|-------|-------|-------|-------|
| 603h | 2Fh | 99h | 59h | 02h | 06h | 00h | 00h | 00h |

Objekt:

| | | | | | | | | |
|-------|---|--|-------------|----|-----|--|--|--|
| 5999h | | Parameter LSS | | | | | | |
| | 0 | Anzahl Einträge | Unsigned 8 | Ro | 3 | | | |
| | 1 | Übertragungsgeschwindigkeit – LSS-PARA | Unsigned 16 | Rw | 250 | Mögliche Werte: 50 kbit/s 125 kbit/s 250 kbit/s 500 kbit/s 800 kbit/s 1000 kbit/s Min=50 und Max=1000 WICHTIGER HINWEIS: Diesen Dienst nur verwenden, wenn ein Gerät an das Netzwerk angeschlossen ist | | |
| | 2 | Knoten-ID – LSS-PARA | Unsigned 8 | Rw | 6 | Min=1 und Max=127 WICHTIGER HINWEIS: Diesen Dienst nur verwenden, wenn ein Gerät an das Netzwerk angeschlossen ist | | |

| | | | | | | |
|--|---|------------------|-------------|----|--|---|
| | 3 | Store - LSS-PARA | Unsigned 32 | Wo | | "Speichern" (0x65766173) zum Speichern aller Parameter LSS (Objekt mit Kennzeichnung LSS-PARA) WICHTIGER HINWEIS: Diesen Dienst nur verwenden, wenn ein Gerät an das Netzwerk angeschlossen ist |
|--|---|------------------|-------------|----|--|---|

Unterstützt werden die Knoten-IDs von 0x01 bis 0x7F :

Die Antwort nach dem ordnungsgemäßen Speichern ist:

| ID | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 | Byte8 |
|------|-------|-------|-------|-------|-------|-------|-------|-------|
| 583h | 60h | 99h | 59h | 02h | 00h | 00h | 00h | 00h |

Zum Speichern der neuen Baudrate den Befehl „Speichern“ wie nachstehend angegeben schreiben:

Schreiben (im Beispiel Knoten-ID = 0x03)

| ID | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 | Byte8 |
|------|-------|-------|-------|-------|-------|-------|-------|-------|
| 603h | 23h | 99h | 59h | 03h | 73h | 61h | 76h | 65h |

Hinweis: Der Befehl „Speichern“ wird durch Senden des nachstehenden Codes gegeben:

| | | | |
|-----|-----|-----|-----|
| 73h | 61h | 76h | 65h |
|-----|-----|-----|-----|

Wobei gilt:

73h = ASCII-Code "s"

61h = ASCII-Code "a"

76h = ASCII-Code "v"

65h = ASCII-Code "e"

Die Antwort nach dem ordnungsgemäßen Speichern ist:

| ID | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 | Byte8 |
|------|-------|-------|-------|-------|-------|-------|-------|-------|
| 583h | 60h | 99h | 59h | 03h | 00h | 00h | 00h | 00h |

WICHTIGER HINWEIS:

Damit die neuen Einträge gültig werden, muss ein Reset durchgeführt werden (das Modul kurz ausschalten).

**Beispiel 3) Aktivieren eines automatischen NMT-Starts nach dem Einschalten
(das PDO wird nach dem Einschalten automatisch gesendet)**

Mit dem Service Data Object (SDO) wird auf die Einträge des Objektverzeichnisses des Geräts zugegriffen. Da diese Einträge Daten beliebiger Länge und beliebigen Typs enthalten können, kann das SDO zum Übertragen von mehreren Datensätzen von einem Client an einen Server und umgekehrt verwendet werden.

SDO-Struktur - Anfrage vom Master

| COB-ID | DLC | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 | Byte8 |
|-------------|-----|-------|-------|-----------|-------|-------|-------|-------|-------|
| 600+Nodo-ID | 8 | CMD | Index | Sub-Index | Daten | | | | |

Das **CMD** bestimmt die Richtung der Datenübertragung und die Größe des Datenobjekts:

23 hex Senden von 4-Byte-Daten (die Bytes 5...8 enthalten einen 32-Bit-Wert)

2B Senden von 2-Byte-Daten (die Bytes 5, 6 enthalten einen 16-Bit-Wert)

2F hex Senden von 1-Byte-Daten (Byte 5 enthält einen 8-Bit-Wert)

SDO-Struktur - Antwort vom Slave

| COB-ID | DLC | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 | Byte8 |
|-------------|-----|-------|-------|-----------|-------|-------|-------|-------|-------|
| 580+Nodo-ID | 8 | CMD | Index | Sub-Index | Daten | | | | |

RES Antwort des Slave:

60 hex Daten erfolgreich übertragen

80 hex Fehler

Schreiben (im Beispiel Knoten-ID = 0x03)

| ID | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 | Byte8 |
|------|-------|-------|-------|-------|-------|-------|-------|-------|
| 603h | 2Fh | 00h | 50h | 00h | 01h | 00h | 00h | 00h |

Objekt:

| | | | | | |
|-------|---|------------|----|---|--|
| 5000h | Automatischer NMT-Start nach dem Einschalten - PARA | Unsigned 8 | Rw | 1 | 0: nicht aktiviert 1: aktiviert Min.= 0 und Max.=1 |
|-------|---|------------|----|---|--|

Zum Speichern der Funktionalität den Befehl „Speichern“ wie nachstehend angegeben schreiben:

Schreiben (im Beispiel Knoten-ID = 0x03)

| ID | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 | Byte8 |
|------|-------|-------|-------|-------|-------|-------|-------|-------|
| 603h | 23h | 10h | 10h | 01h | 73h | 61h | 76h | 65h |

Hinweis: Der Befehl „Speichern“ wird durch Senden des nachstehenden Codes gegeben:

| | | | |
|-----|-----|-----|-----|
| 73h | 61h | 76h | 65h |
|-----|-----|-----|-----|

Wobei gilt:

73h = ASCII-Code "s"

61h = ASCII-Code "a"

76h = ASCII-Code "v"

65h = ASCII-Code "e"

Die Antwort nach dem ordnungsgemäßen Speichern ist:

| ID | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 | Byte8 |
|------|-------|-------|-------|-------|-------|-------|-------|-------|
| 583h | 60h | 10h | 10h | 01h | 00h | 00h | 00h | 00h |

WICHTIGER HINWEIS:

Damit die neuen Einträge gültig werden, muss ein Reset durchgeführt werden (das Modul kurz ausschalten).

Beispiel 4) Ändern der PDO-Rate (Zeitintervall) von 100 ms nach 20 ms

Mit dem Service Data Object (SDO) wird auf die Einträge des Objektverzeichnisses des Geräts zugegriffen. Da diese Einträge Daten beliebiger Länge und beliebigen Typs enthalten können, kann das SDO zum Übertragen von mehreren Datensätzen von einem Client an einen Server und umgekehrt verwendet werden.

SDO-Struktur - Anfrage vom Master

| COB-ID | DLC | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 | Byte8 |
|-------------|-----|-------|-------|-------|-----------|-------|-------|-------|-------|
| 600+Node-ID | 8 | CMD | Index | | Sub-Index | Daten | | | |

Das **CMD** bestimmt die Richtung der Datenübertragung und die Größe des Datenobjekts:

23 hex Senden von 4-Byte-Daten (die Bytes 5...8 enthalten einen 32-Bit-Wert)

2B hex Senden von 2-Byte-Daten (die Bytes 5, 6 enthalten einen 16-Bit-Wert)

2F hex Senden von 1-Byte-Daten (Byte 5 enthält einen 8-Bit-Wert)

SDO-Struktur - Antwort vom Slave

| COB-ID | DLC | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 | Byte8 |
|-------------|-----|-------|-------|-------|-----------|-------|-------|-------|-------|
| 580+Node-ID | 8 | CMD | Index | | Sub-Index | Daten | | | |

RES Antwort des Slave:

60 hex Daten erfolgreich übertragen

80 hex Fehler

Schreiben (im Beispiel Knoten-ID = 0x03)

| ID | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 | Byte8 |
|------|-------|-------|-------|-------|-------|-------|-------|-------|
| 603h | 2Bh | 00h | 18h | 05h | 14h | 00h | 00h | 00h |

Objekt:

| | | | | | | |
|-------|---|--|-------------|----|-----------------|--|
| 1800h | 0 | 1 st Transmit PDO Parameter | Unsigned 8 | Ro | | |
| | 1 | COB-ID | Unsigned 32 | Ro | 180h+ Knoten-ID | |
| | 2 | Übertragungsart | Unsigned 8 | Rw | 254 | Asynchrone Übertragung.. |
| | 3 | Sperrzeit | Unsigned 16 | Ro | 0 | Min.= 0 und Max.=65535 mit Einheit=1ms |
| | 4 | Reserviert | // | // | | |
| | 5 | Timer | Unsigned 16 | Rw | 20 | Min.= 0 und Max.=65535 mit Einheit=1ms |

Die Antwort nach dem ordnungsgemäßen Speichern ist:

| ID | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 | Byte8 |
|------|-------|-------|-------|-------|-------|-------|-------|-------|
| 583h | 60h | 00h | 18h | 05h | 00h | 00h | 00h | 00h |

Zum Speichern der Funktionalität den Befehl „Speichern“ wie nachstehend angegeben schreiben:

Schreiben (im Beispiel Knoten-ID = 0x03)

| ID | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 | Byte8 |
|------|-------|-------|-------|-------|-------|-------|-------|-------|
| 603h | 23h | 10h | 10h | 01h | 73h | 61h | 76h | 65h |

Hinweis: Der Befehl „Speichern“ wird durch Senden des nachstehenden Codes gegeben:

| | | | |
|-----|-----|-----|-----|
| 73h | 61h | 76h | 65h |
|-----|-----|-----|-----|

Wobei gilt:

73h = ASCII-Code "s"

61h = ASCII-Code "a"

76h = ASCII-Code "v"

65h = ASCII-Code "e"

Die Antwort nach dem ordnungsgemäßen Speichern ist:

| ID | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 | Byte8 |
|------|-------|-------|-------|-------|-------|-------|-------|-------|
| 583h | 60h | 10h | 10h | 01h | 00h | 00h | 00h | 00h |

WICHTIGER HINWEIS:

Damit die neuen Einträge gültig werden, muss ein Reset durchgeführt werden (das Modul kurz ausschalten).

Beispiel 5) HEinstellen des Null-Grad-Punkts von Winkel 1 (Beispiel mit Auflösung ± 0.1°)

Mit dem Service Data Object (SDO) wird auf die Einträge des Objektverzeichnisses des Geräts zugegriffen. Da diese Einträge Daten beliebiger Länge und beliebigen Typs enthalten können, kann das SDO zum Übertragen von mehreren Datensätzen von einem Client an einen Server und umgekehrt verwendet werden.

SDO-Struktur - Anfrage vom Master

| COB-ID | DLC | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 | Byte8 |
|-------------|-----|-------|-------|-------|-----------|-------|-------|-------|-------|
| 600+Node-ID | 8 | CMD | Index | | Sub-Index | Daten | | | |

Das **CMD** bestimmt die Richtung der Datenübertragung und die Größe des Datenobjekts:
 23 hex Senden von 4-Byte-Daten (die Bytes 5...8 enthalten einen 32-Bit-Wert)
 2B Senden von 2-Byte-Daten (die Bytes 5, 6 enthalten einen 16-Bit-Wert)
 2F hex Senden von 1-Byte-Daten (Byte 5 enthält einen 8-Bit-Wert)

SDO-Struktur - Antwort vom Slave

| COB-ID | DLC | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 | Byte8 |
|-------------|-----|-------|-------|-------|-----------|-------|-------|-------|-------|
| 580+Node-ID | 8 | CMD | Index | | Sub-Index | Daten | | | |

RES Antwort des Slave:
 60 hex Daten erfolgreich übertragen
 80 hex Fehler

Schreiben (im Beispiel Knoten-ID = 0x03)

Wenn der Wert von Winkel 1 02h 65 h = 0265 h = 613d = 61,3° ist, muss man zum Verschieben von Winkel 1 zum Nullpunkt zu Byte 5 und Byte 6 die folgenden Werte hinzufügen:

| ID | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 | Byte8 |
|------|-------|-------|-------|-------|-------|-------|-------|-------|
| 603h | 2Bh | 01h | 21h | 01h | 65h | 02h | 00h | 00h |

Objekt:

| | | | | | | | | |
|-------|---|------------------------------|-------------|----|-----|--|--|--|
| 2101h | | Null-Grad-Punkt des Winkels | | | | | Die Null-Grad-Punkte von Winkel 1 und Winkel 2 sind auf den zulässigen maximalen Gradwert zu beziehen. Min.= 0 und Max =16383 | |
| | 0 | Anzahl Einträge | Unsigned 8 | Ro | 2 | | | |
| | 1 | Null-Grad-Punkt von Winkel 1 | Unsigned 16 | Rw | 613 | | | |
| | 2 | Null-Grad-Punkt von Winkel 2 | Unsigned 16 | Rw | 0 | | | |

Die Antwort nach dem ordnungsgemäßen Speichern ist:

| ID | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 | Byte8 |
|------|-------|-------|-------|-------|-------|-------|-------|-------|
| 583h | 60h | 01h | 21h | 01h | 00h | 00h | 00h | 00h |

Zum Speichern der Funktionalität den Befehl „Speichern“ wie nachstehend angegeben schreiben:

Schreiben (im Beispiel Knoten-ID = 0x03)

| ID | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 | Byte8 |
|------|-------|-------|-------|-------|-------|-------|-------|-------|
| 603h | 23h | 10h | 10h | 01h | 73h | 61h | 76h | 65h |

Hinweis: Der Befehl „Speichern“ wird durch Senden des nachstehenden Codes gegeben:

| | | | |
|-----|-----|-----|-----|
| 73h | 61h | 76h | 65h |
|-----|-----|-----|-----|

Wobei gilt:

73h = ASCII-Code "s"

61h = ASCII-Code "a"

76h = ASCII-Code "v"

65h = ASCII-Code "e"

Die Antwort nach dem ordnungsgemäßen Speichern ist:

| ID | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 | Byte8 |
|------|-------|-------|-------|-------|-------|-------|-------|-------|
| 583h | 60h | 10h | 10h | 01h | 00h | 00h | 00h | 00h |

WICHTIGER HINWEIS:

Damit die neuen Einträge gültig werden, muss ein Reset durchgeführt werden (das Modul kurz ausschalten).

Beispiel 6) Einstellen des Null-Grad-Punkts von Winkel 2 (Beispiel mit Auflösung ± 0.1°)

Mit dem Service Data Object (SDO) wird auf die Einträge des Objektverzeichnisses des Geräts zugegriffen. Da diese Einträge Daten beliebiger Länge und beliebigen Typs enthalten können, kann das SDO zum Übertragen von mehreren Datensätzen von einem Client an einen Server und umgekehrt verwendet werden.

SDO-Struktur - Anfrage vom Master

| COB-ID | DLC | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 | Byte8 |
|-------------|-----|-------|-------|-------|-----------|-------|-------|-------|-------|
| 600+Node-ID | 8 | CMD | Index | | Sub-Index | Daten | | | |

Das **CMD** bestimmt die Richtung der Datenübertragung und die Größe des Datenobjekts:

23 hex Senden von 4-Byte-Daten (die Bytes 5...8 enthalten einen 32-Bit-Wert)

2B Senden von 2-Byte-Daten (die Bytes 5, 6 enthalten einen 16-Bit-Wert)

2F hex Senden von 1-Byte-Daten (Byte 5 enthält einen 8-Bit-Wert)

SDO-Struktur - Antwort vom Slave

| COB-ID | DLC | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 | Byte8 |
|-------------|-----|-------|-------|-------|-----------|-------|-------|-------|-------|
| 580+Node-ID | 8 | CMD | Index | | Sub-Index | Daten | | | |

RES Antwort des Slave:

60 hex Daten erfolgreich übertragen

80 hex Fehler

Schreiben (im Beispiel Knoten-ID = 0x03)

Wenn der Wert von Winkel 2 02h 65 h = 0265 h = 613d = 61,3° ist, muss man zum Verschieben von Winkel 2 zum Nullpunkt zu Byte 5 und Byte 6 die folgenden Werte hinzufügen:

| ID | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 | Byte8 |
|------|-------|-------|-------|-------|-------|-------|-------|-------|
| 603h | 2Bh | 01h | 21h | 02h | 65h | 02h | 00h | 00h |

Objekt:

| | | | | | | | | |
|-------|---|------------------------------|-------------|----|-----|--|--|--|
| 2101h | | Null-Grad-Punkt des Winkels | | | | | | Die Null-Grad-Punkte von Winkel 1 und Winkel 2 sind auf den zulässigen maximalen Gradwert zu beziehen. Min.= 0 und Max.=16383 |
| | 0 | Anzahl Einträge | Unsigned 8 | Ro | 2 | | | |
| | 1 | Null-Grad-Punkt von Winkel 1 | Unsigned 16 | Rw | 0 | | | |
| | 2 | Null-Grad-Punkt von Winkel 2 | Unsigned 16 | Rw | 613 | | | |

Die Antwort nach dem ordnungsgemäßen Speichern ist:

| ID | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 | Byte8 |
|------|-------|-------|-------|-------|-------|-------|-------|-------|
| 583h | 60h | 01h | 21h | 02h | 00h | 00h | 00h | 00h |

Zum Speichern der Funktionalität den Befehl „Speichern“ wie nachstehend angegeben schreiben:

Schreiben (im Beispiel Knoten-ID = 0x03)

| ID | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 | Byte8 |
|------|-------|-------|-------|-------|-------|-------|-------|-------|
| 603h | 23h | 10h | 10h | 01h | 73h | 61h | 76h | 65h |

Hinweis: Der Befehl „Speichern“ wird durch Senden des nachstehenden Codes gegeben:

| | | | |
|-----|-----|-----|-----|
| 73h | 61h | 76h | 65h |
|-----|-----|-----|-----|

Wobei gilt:

73h = ASCII-Code "s"

61h = ASCII-Code "a"

76h = ASCII-Code "v"

65h = ASCII-Code "e"

Die Antwort nach dem ordnungsgemäßen Speichern ist:

| ID | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 | Byte8 |
|------|-------|-------|-------|-------|-------|-------|-------|-------|
| 583h | 60h | 10h | 10h | 01h | 00h | 00h | 00h | 00h |

WICHTIGER HINWEIS:

Damit die neuen Einträge gültig werden, muss ein Reset durchgeführt werden (das Modul kurz ausschalten).

Objekt:

| | | | | | | |
|-------|---|--------------------|-------------|----|-----|--|
| 2100h | | AUFLÖSUNG Winkel | | | | Zulässige Werte der AUFLÖSUNG von Winkel 1 und Winkel 2 : 1000d: 1 Grad/Bit 100d: 0,1 Grad/Bit 22d: 0,01 Grad/Bit (14 Bit effektive Auflösung 0.02°) |
| | 0 | Anzahl Einträge | Unsigned 8 | Ro | 2 | |
| | 1 | Winkel 1 Auflösung | Unsigned 16 | Rw | 100 | |
| | 2 | Winkel 2 Auflösung | Unsigned 16 | Rw | 100 | |

Die Antwort nach dem ordnungsgemäßen Speichern ist:

| ID | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 | Byte8 |
|------|-------|-------|-------|-------|-------|-------|-------|-------|
| 583h | 60h | 00h | 21h | 01h | 00h | 00h | 00h | 00h |

| ID | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 | Byte8 |
|------|-------|-------|-------|-------|-------|-------|-------|-------|
| 583h | 60h | 00h | 21h | 02h | 00h | 00h | 00h | 00h |

Zum Speichern der Funktionalität den Befehl „Speichern“ wie nachstehend angegeben schreiben:

Schreiben (im Beispiel Knoten-ID = 0x03)

| ID | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 | Byte8 |
|------|-------|-------|-------|-------|-------|-------|-------|-------|
| 603h | 23h | 10h | 10h | 01h | 73h | 61h | 76h | 65h |

Hinweis: Der Befehl „Speichern“ wird durch Senden des nachstehenden Codes gegeben:

| | | | |
|-----|-----|-----|-----|
| 73h | 61h | 76h | 65h |
|-----|-----|-----|-----|

Wobei gilt:

73h = ASCII-Code "s"

61h = ASCII-Code "a"

76h = ASCII-Code "v"

65h = ASCII-Code "e"

Die Antwort nach dem ordnungsgemäßen Speichern ist:

| ID | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 | Byte8 |
|------|-------|-------|-------|-------|-------|-------|-------|-------|
| 583h | 60h | 10h | 10h | 01h | 00h | 00h | 00h | 00h |

WICHTIGER HINWEIS:

Damit die neuen Einträge gültig werden, muss ein Reset durchgeführt werden (das Modul kurz ausschalten).

Beispiel 8) Einstellen der Auflösung auf $\pm 0.01^\circ$ bei Winkel 1 und Winkel 2 (die aktuelle Einstellung der Auflösung ist $\pm 0.1^\circ$)

Mit dem Service Data Object (SDO) wird auf die Einträge des Objektverzeichnisses des Geräts zugegriffen. Da diese Einträge Daten beliebiger Länge und beliebigen Typs enthalten können, kann das SDO zum Übertragen von mehreren Datensätzen von einem Client an einen Server und umgekehrt verwendet werden.

SDO-Struktur - Anfrage vom Master

| COB-ID | DLC | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 | Byte8 |
|-------------|-----|-------|-------|-----------|-------|-------|-------|-------|-------|
| 600+Node-ID | 8 | CMD | Index | Sub-Index | Daten | | | | |

Das **CMD** bestimmt die Richtung der Datenübertragung und die Größe des Datenobjekts:
 23 hex Senden von 4-Byte-Daten (die Bytes 5...8 enthalten einen 32-Bit-Wert)
 2B Senden von 2-Byte-Daten (die Bytes 5, 6 enthalten einen 16-Bit-Wert)
 2F hex Senden von 1-Byte-Daten (Byte 5 enthält einen 8-Bit-Wert)

SDO-Struktur - Antwort vom Slave

| COB-ID | DLC | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 | Byte8 |
|-------------|-----|-------|-------|-----------|-------|-------|-------|-------|-------|
| 580+Node-ID | 8 | CMD | Index | Sub-Index | Daten | | | | |

RES Antwort des Slave:
 60 hex Daten erfolgreich übertragen
 80 hex Fehler

Zum Einstellen der Auflösung bei Winkel 1 auf $\pm 0,01^\circ$ das PDO senden/schreiben (im Beispiel Knoten-ID = 0x03):

| ID | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 | Byte8 |
|------|-------|-------|-------|-------|-------|-------|-------|-------|
| 603h | 2Bh | 00h | 21h | 01h | 16h | 00h | 00h | 00h |

Objekt:

| | | | | | | | |
|-------|---|--------------------|-------------|----|-----|--|---|
| 2100h | | AUFLÖSUNG Winkel | | | | | Zulässige Werte der AUFLÖSUNG von Winkel 1 und Winkel 2: 1000d: 1 Grad/Bit 100d: 0,1 Grad/Bit 22d: 0,01 Grad/Bit (14 Bit effektive Auflösung 0.02°) |
| | 0 | Anzahl Einträge | Unsigned 8 | Ro | 2 | | |
| | 1 | Winkel 1 Auflösung | Unsigned 16 | Rw | 22 | | |
| | 2 | Winkel 2 Auflösung | Unsigned 16 | Rw | 100 | | |

Zum Einstellen der Auflösung bei Winkel 2 auf $\pm 0,01^\circ$ das PDO senden/schreiben (im Beispiel Knoten-ID = 0x03):

| ID | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 | Byte8 |
|------|-------|-------|-------|-------|-------|-------|-------|-------|
| 603h | 2Bh | 00h | 21h | 02h | 16h | 00h | 00h | 00h |

Objekt:

| | | | | | | |
|-------|---|--------------------|-------------|----|----|---|
| 2100h | | AUFLÖSUNG Winkel | | | | Zulässige Werte der AUFLÖSUNG von Winkel 1 und Winkel 2: 1000d: 1 Grad/Bit 100d: 0,1 Grad/Bit 22d: 0,01 Grad/Bit (14 Bit effektive Auflösung 0.02°) |
| | 0 | Anzahl Einträge | Unsigned 8 | Ro | 2 | |
| | 1 | Winkel 1 Auflösung | Unsigned 16 | Rw | 22 | |
| | 2 | Winkel 2 Auflösung | Unsigned 16 | Rw | 22 | |

Die Antwort nach dem ordnungsgemäßen Speichern ist:

| ID | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 | Byte8 |
|------|-------|-------|-------|-------|-------|-------|-------|-------|
| 583h | 60h | 00h | 21h | 01h | 00h | 00h | 00h | 00h |

| ID | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 | Byte8 |
|------|-------|-------|-------|-------|-------|-------|-------|-------|
| 583h | 60h | 00h | 21h | 02h | 00h | 00h | 00h | 00h |

Zum Speichern der Funktionalität den Befehl „Speichern“ wie nachstehend angegeben schreiben:

Schreiben (im Beispiel Knoten-ID = 0x03)

| ID | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 | Byte8 |
|------|-------|-------|-------|-------|-------|-------|-------|-------|
| 603h | 23h | 10h | 10h | 01h | 73h | 61h | 76h | 65h |

Hinweis: Der Befehl „Speichern“ wird durch Senden des nachstehenden Codes gegeben:

| | | | |
|-----|-----|-----|-----|
| 73h | 61h | 76h | 65h |
|-----|-----|-----|-----|

Wobei gilt:

73h = ASCII-Code "s"

61h = ASCII-Code "a"

76h = ASCII-Code "v"

65h = ASCII-Code "e"

Die Antwort nach dem ordnungsgemäßen Speichern ist:

| ID | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 | Byte8 |
|------|-------|-------|-------|-------|-------|-------|-------|-------|
| 583h | 60h | 10h | 10h | 01h | 00h | 00h | 00h | 00h |

WICHTIGER HINWEIS:

Damit die neuen Einträge gültig werden, muss ein Reset durchgeführt werden (das Modul kurz ausschalten).

Beispiel 9) Befehl RESTORE senden

Mit dem Service Data Object (SDO) wird auf die Einträge des Objektverzeichnisses des Geräts zugegriffen. Da diese Einträge Daten beliebiger Länge und beliebigen Typs enthalten können, kann das SDO zum Übertragen von mehreren Datensätzen von einem Client an einen Server und umgekehrt verwendet werden.

Struktur der SDO-Anfrage vom Master

| COB-ID | DLC | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 | Byte8 |
|-------------|-----|-------|-------|-----------|-------|-------|-------|-------|-------|
| 600+Node-ID | 8 | CMD | Index | Sub-Index | Data | Data | Data | Data | Data |

Das **CMD** bestimmt die Richtung der Datenübertragung und die Größe des Datenobjekts:

23 hex Senden von 4-Byte-Daten (die Bytes 5...8 enthalten einen 32-Bit-Wert)

2B hex Senden von 2-Byte-Daten (die Bytes 5, 6 enthalten einen 16-Bit-Wert)

2F hex Senden von 1-Byte-Daten (Byte 5 enthält einen 8-Bit-Wert)

Struktur der SDO-Antwort vom Slave

| COB-ID | DLC | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 | Byte8 |
|-------------|-----|-------|-------|-----------|-------|-------|-------|-------|-------|
| 580+Node-ID | 8 | CMD | Index | Sub-Index | Data | Data | Data | Data | Data |

RES Antwort des Slave:

60 hex Daten erfolgreich übertragen

80 hex Fehler

Schreiben (im Beispiel die Knoten-ID = 0x7F)

| ID | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 | Byte8 |
|------|-------|-------|-------|-------|-------|-------|-------|-------|
| 67Fh | 23h | 11h | 10h | 01h | 6Ch | 6Fh | 61h | 64h |

Objekt:

| | | | | | | | | |
|-------|---|----------------------|------------|----|--|--|--|---|
| 1011h | 1 | Alle Parameter laden | Unsigned 8 | Wo | | | | „Laden“ (0x64616F6C) zum Wiederherstellen aller Parameter (Objekte mit den Kennzeichnungen PARA und LSSPARA). |
|-------|---|----------------------|------------|----|--|--|--|---|

Nach erfolgreichem Abschluss des Speichervorgangs erhält man folgende Antwort:

| ID | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 | Byte8 |
|------|-------|-------|-------|-------|-------|-------|-------|-------|
| 5FFh | 60h | 11h | 10h | 01h | 00h | 00h | 00h | 00h |

WICHTIGER HINWEIS:

Damit die neuen Einträge gültig werden, muss ein Reset durchgeführt werden (das Modul kurz ausschalten).

