

Technisches Handbuch

Für Absolut-Drehgeber
HTB36E und FHB58
mit CANopen-Schnittstelle



Impressum

MEGATRON Elektronik GmbH & Co. KG

Hermann-Oberth-Str. 7

D-85640 Putzbrunn

Tel: +49 (0) 89 / 46094 - 0

Fax: +49 (0) 89 / 46094 - 201

E-Mail: info@megatron.de

Homepage: <https://www.megatron.de>

Garantieverzicht, Änderungsvorbehalt, Urheberrechtsschutz:

Die Firma MEGATRON übernimmt keine Haftung oder Garantie für die Richtigkeit dieses Handbuches, sowie indirekte oder direkte Schäden, die daraus entstehen können. Im Sinne der stetigen Innovation und Zusammenarbeit mit Kunden behalten wir uns vor, technische Daten oder Inhalte jederzeit zu ändern.

Dieses Handbuch darf ohne vorherige schriftliche Genehmigung durch die Firma MEGATRON weder abgeändert, erweitert, vervielfältigt, noch an Dritte weitergegeben werden.

1	Einleitung	1
1.1	Drehgebertypen	1
1.2	Zu diesem Handbuch	2
1.2.1	Symbolerklärung	2
1.3	Leistungsbeschreibung	3
2	Sicherheitshinweise	4
2.1	Allgemeines	4
2.2	Bestimmungsgemäße Verwendung	4
2.3	Sicheres Arbeiten	5
2.4	Entsorgung	5
3	Gerätebeschreibung	6
3.1	Allgemeines Drehgeber-Design	6
3.2	Predefined Connection Settings	6
3.3	Leuchtdiode und Status-Signalisierung	6
4	Schnellstart	9
4.1	CAN-Netzwerk-Integration	9
4.2	SDO-Befehl zur Einstellung Node-ID	9
4.3	Drehgeber-Inbetriebnahme	10
5	Allgemeine CAN-Informationen	12
5.1	CAN-Physik	12
5.2	CANopen	14
5.3	Spezifikationen und Profile	15
5.3.1	Überblick	15
5.3.2	Kommunikationsmechanismen	15
5.3.3	Objektverzeichnis	16
5.4	Netzwerkmanagement (NMT)	17
5.5	Heartbeat und Node-Guarding	18
5.6	Emergency-Nachrichten	19
6	Objektverzeichnis HTB36E oder FHB58	20
6.1	Kommunikationsspezifische Objekte	20
6.2	Gerätespezifische Objekte	23
6.3	Herstellerspezifische Objekte	30
7	Objektbeschreibung	32
7.1	Netzwerkmanagementbefehle	32

7.2	Heartbeat-Protokoll.....	33
7.3	Emergency-Nachrichten	34
7.4	Fehler Objekte	35
7.4.1	Manufacturer status register	35
7.4.2	Alarms.....	36
7.4.3	Warnings.....	36
7.5	Elektronisches Nockenschaltwerk (CAM).....	36
7.5.1	CAM-state-register.....	36
7.5.2	CAM-enable-register.....	37
7.5.3	CAM-polarity-register	37
7.5.4	CAM-Low-Limit	38
7.5.5	CAM-High-Limit.....	38
7.5.6	CAM-Hysteresis	38
7.6	Geräte-Profil	38
7.7	SYNC	38
7.8	Drehgeber-Bezeichnung.....	39
7.9	Verhalten bei Auftreten von Fehlern	39
7.10	NMT-Startup-Verhalten	39
7.11	Bus-Off Auto-Reset	40
7.12	Customer Data	40
7.13	Temperatur.....	40
7.14	Konfigurationsabgleich	40
8	Inbetriebnahme.....	41
8.1	Mechanischer und elektrischer Anschluss.....	41
8.2	Einstellungen per LSS vornehmen	43
8.2.1	Allgemeine Einstellungen.....	43
8.2.2	LSS-Konfigurationsmodus mit „Switch Mode Global“	43
8.2.3	LSS-Konfigurationsmodus mit „Switch Mode Selective“	44
8.2.4	Beenden des LSS-Konfigurationsmodus	45
8.2.5	Baudrate einstellen	45
8.2.6	Node-ID des Drehgebers einstellen	46
8.3	Einstellungen per SDO vornehmen	47
8.3.1	Objekte einrichten und lesen.....	47
8.3.2	Große Objekte (>4 Byte) einrichten und lesen.....	49
8.3.3	Baudrate einstellen	56

8.3.4	Node-ID des Drehgebers einstellen	57
8.3.5	NMT-Master Basisbefehle.....	58
8.4	Heartbeateinstellungen.....	59
8.5	PDO-Konfiguration	59
8.5.1	PDO-Parametrierung	59
8.5.2	PDO in Synchronmodus setzen	61
8.5.3	PDO in Asynchronmodus setzen	61
8.5.4	Variables PDO-Mapping	62
8.6	Auflösung und Drehsinn ändern	65
8.7	Wellen-Position setzen	66
8.8	Positionswert filtern	67
8.9	Speed-Integration und Skalierung ändern	67
8.10	Frequency-Limit	68
8.11	CAM-Konfiguration.....	68
8.12	Einstellungen in das EEPROM speichern	70
8.12.1	Netzausfallsicheres Speichern von Parametern	70
8.12.2	Parameter zurücksetzen auf Werkseinstellungen.....	71
9	Fehler-Diagnose	72
9.1	Troubleshooting Gebereinstellungen.....	72
10	Technische Beratung	73

Abbildungsverzeichnis

Abbildung 1.1: Illustration eines Drehgeberetiketts.....	1
Abbildung 3.1: Drehgeberbauformen Vollwelle und Hohlwelle (Beispiel HTB36E).....	6
Abbildung 3.2: LED Signale 1	7
Abbildung 3.3: LED Signale 2.....	8
Abbildung 5.1: Beispiel einer Arbitrierung.....	13
Abbildung 5.2: Bitstuffing	13
Abbildung 5.3: ISO-OSI-Schichtmodell.....	14
Abbildung 8.1: Objekt Lesen.....	47
Abbildung 8.2: Objekt schreiben.....	48
Abbildung 8.3: Allgemeiner segmented SDO-Transfer Lesezugriff	49
Abbildung 8.4: Initiate SDO read	50
Abbildung 8.5: SDO-Segment auslesen	51
Abbildung 8.6: Segmented-SDO Transfer Schreibzugriff	52
Abbildung 8.7: Initiate SDO write.....	53
Abbildung 8.8: write SDO segment.....	55

Tabellenverzeichnis

Tabelle 3.1: CAN-Identifizier	6
Tabelle 4.1: SYNC-Nachricht.....	9
Tabelle 4.2: SDO-Schreibbefehl zum Setzen der Node-ID.....	10
Tabelle 4.3: Node-ID in dezimaler und hexadezimaler Form.....	10
Tabelle 5.1: CAN Baudraten und Kabellängen.....	14
Tabelle 5.2: Liste CiA Spezifikationen	15
Tabelle 5.3: Struktur des Objektverzeichnisses.....	16
Tabelle 5.4: Mögliche Kommunikation – Pre-Operational.....	17
Tabelle 5.5: Mögliche Kommunikation – Operational	18
Tabelle 5.6: Mögliche Kommunikation – Stopped.....	18
Tabelle 6.1: Objektverzeichnis 1000h – 100Dh	21
Tabelle 6.2: Objektverzeichnis 1010h – 1020h.....	22
Tabelle 6.3: Objektverzeichnis 1029h – 1A01h	23
Tabelle 6.4: Objektverzeichnis 1A03h – 1F80h	23
Tabelle 6.5: Gerätespezifische Objekte 6000h –6008h	24
Tabelle 6.6: Gerätespezifische Objekte 6009h –6310h	25
Tabelle 6.7: Gerätespezifische Objekte 6311h –6321h	26
Tabelle 6.8: Gerätespezifische Objekte 6322h –6332h	27
Tabelle 6.9: Gerätespezifische Objekte 6333h –6500h	28
Tabelle 6.10: Gerätespezifische Objekte 6501h –6510h	29
Tabelle 6.11: Herstellerspezifische Objekte 2100h –2500h.....	30
Tabelle 6.12: Herstellerspezifische Objekte 2502h –2504h.....	31
Tabelle 7.1: Aufbau NMT-Befehle	32
Tabelle 7.2: Commands für NMT-Befehle	32
Tabelle 7.3: Node-ID Werte für NMT-Befehle.....	32
Tabelle 7.4: Fremden Heartbeat überwachen	33
Tabelle 7.5: Beispielkonfiguration eines Consumer Heartbeats	33
Tabelle 7.6: Allgemeiner Aufbau einer Emergency-Nachricht	34
Tabelle 7.7: Emergency Error Code Liste.....	34
Tabelle 7.8: Error-register.....	34
Tabelle 7.9: Infofeld Liste.....	35
Tabelle 7.10: Manufacturer status register	35
Tabelle 7.11: Alarms - Objekt 6503h	36
Tabelle 7.12: Warnings – Objekt 6505h.....	36
Tabelle 7.13: CAM-state-register – Wert 89h	36
Tabelle 7.14: CAM-state-register – Wert 81h	37
Tabelle 7.15: CAM-enable-register – Wert 4Ah	37
Tabelle 7.16: Beispiel CAM-polarity-register.....	37
Tabelle 7.17: Werte zur Gebersteuerung bei Fehler.....	39
Tabelle 7.18: Werte zur Geber-Startup-Steuerung	39
Tabelle 8.1: Stecker-Kabel Belegung	42
Tabelle 8.2: LSS-Nachricht.....	43
Tabelle 8.3: Befehl um Drehgeber in „Stopped“-Modus zu setzen	43
Tabelle 8.4: LSS-Selective-Identification-Commands.....	44
Tabelle 8.5: Antwort des Drehgebers auf LSS-Selective-Identification-Commands .	44

Tabelle 8.6: LSS-Konfigurationsmodus beenden – Schritt 1: speichern.....	45
Tabelle 8.7: Konfigurations-Modus verlassen.....	45
Tabelle 8.8: Baudrate einstellen	45
Tabelle 8.9: Baudraten-Codierung.....	45
Tabelle 8.10: Antwort des LSS-Slaves	46
Tabelle 8.11: Einstellen der Node-ID.....	46
Tabelle 8.12: Beispiel SDO Masteranfrage – Objekt lesen.....	47
Tabelle 8.13: Beispiel SDO Antwort – Objekt lesen.....	47
Tabelle 8.14: Kommando-Definitionen	48
Tabelle 8.15: Beispiel SDO Masteranfrage – Objekt schreiben.....	48
Tabelle 8.16: Beispiel SDO Antwort – Objekt schreiben.....	48
Tabelle 8.17: SDO Lesezugriff auf Objekt 6008h	49
Tabelle 8.18: Erläuterung der in Abbildung 8.4 verwendeten Abkürzungen	50
Tabelle 8.19: Bestätigung des SDO Lesezugriffs auf Objekt 6008h.....	50
Tabelle 8.20: Erläuterung der in Abbildung 8.5 verwendeten Abkürzungen	51
Tabelle 8.21: Lesen des ersten Segmentes	51
Tabelle 8.22: Antwort mit erstem Datensegment.....	51
Tabelle 8.23: Lesen des ersten Segmentes	52
Tabelle 8.24: Antwort mit erstem Datensegment.....	52
Tabelle 8.25: SDO Schreibzugriff auf Objekt 6009h	53
Tabelle 8.26: Bestätigung des SDO Schreibzugriffs auf Objekt 6009h.....	53
Tabelle 8.27: Erläuterung der in Abbildung 8.7 verwendeten Abkürzungen	54
Tabelle 8.28: Erstes Segment senden.....	54
Tabelle 8.29: Bestätigung des Drehgebers.....	54
Tabelle 8.30: Erläuterung der in Abbildung 8.8 verwendeten Abkürzungen	55
Tabelle 8.31: Nächstes Segment senden.....	55
Tabelle 8.32: Bestätigung des Drehgebers.....	56
Tabelle 8.33: Nächstes Segment senden.....	56
Tabelle 8.34: Baudraten-Codierung.....	56
Tabelle 8.35: Node-ID einstellen	57
Tabelle 8.36: Beispielwerte der Node-ID	57
Tabelle 8.37: NMT Befehl - Start Remote Node	58
Tabelle 8.38: NMT Befehl - Stop Remote Node	58
Tabelle 8.39: NMT Befehl - Enter Pre-Operational-Status.....	58
Tabelle 8.40: NMT Befehl - Reset Node Communication	58
Tabelle 8.41: NMT Befehl - Reset Remote Node	58
Tabelle 8.42: Beispiel Heartbeateinstellung.....	59
Tabelle 8.43: Beispiel Heartbeateinstellung.....	59
Tabelle 8.44: Heartbeat NMT-Status-Codierung	59
Tabelle 8.45: PDO-Vorkonfiguration.....	59
Tabelle 8.46: Mögliche PDO-Zustände.....	60
Tabelle 8.47: PDO-Deaktivierung	60
Tabelle 8.48: Beispiel PDO1 Deaktivierung.....	60
Tabelle 8.49: Parametrierung von PDO1 Sub-Index 2	61
Tabelle 8.50: Parametrierung von PDO1 Sub-Index 2	61
Tabelle 8.51: Parametrierung von PDO1 Sub-Index 5	61
Tabelle 8.52: Parametrierung von PDO1 Sub-Index 2	62

Tabelle 8.53: Parametrierung von PDO1 Sub-Index 5	62
Tabelle 8.54: Beispiel Mapping-Tabelle.....	62
Tabelle 8.55: Aufbau PDO1 (Inhalt Tabelle 8.54)	63
Tabelle 8.56: Mapping Parameter – Konfiguration freischalten	63
Tabelle 8.57: Positionswert mappen.....	64
Tabelle 8.58: Geschwindigkeitswert mappen	64
Tabelle 8.59: Beschleunigungswert mappen	64
Tabelle 8.60: Mapping Parameter - Konfiguration abschließen	64
Tabelle 8.61: Parameter Drehsinn und Skalierung	65
Tabelle 8.62: Drehen gegen den Uhrzeigersinn	65
Tabelle 8.63: Ändern der Singleturnauflösung per SDO.....	66
Tabelle 8.64: Ändern der Gesamtauflösung per SDO	66
Tabelle 8.65: Wellen-Position setzen	66
Tabelle 8.66: aktuelle Position prüfen.....	67
Tabelle 8.67: Beispiel CAM-Konfiguration	68
Tabelle 8.68: Nockenschaltfunktion aktivieren.....	69
Tabelle 8.69: CAM-High-Limit 1.....	69
Tabelle 8.70: CAM-High-Limit 2.....	69
Tabelle 8.71: CAM-High-Limit 3.....	69
Tabelle 8.72: CAM-Low-Limit 1	69
Tabelle 8.73: CAM-Low-Limit 2	69
Tabelle 8.74: CAM-Low-Limit 3	70
Tabelle 8.75: Parameter-Speicheroptionen	70
Tabelle 8.76: Beispiel – Save all parameters.....	70
Tabelle 8.77: Parameter-Speicheroptionen	71
Tabelle 9.1: Fehler-Diagnose Gebereinstellungen	72

Abkürzungsverzeichnis

autom.	automatisch
bzw.	beziehungsweise
ca.	circa
CAN	Controller Area Network
CAN-ID	Hauptteil des Arbitrierungsfeldes eines CAN-Datenframes
CAM	Nocken „Simulation“- Vorsprung an einer Welle oder Scheibe, der bei Drehung einen meist kraftschlüssig mit dem Nocken verbundenen Maschinenteil eine Auf- und Abwärtsbewegung erteilt, hier über Software nachgebildet.
co	Constant: Parameter ist nur lesbar, ändert sich nicht
COB-ID	Communication Object Identifier, objektspezifische Bedeutung enthält die CAN-ID
CRC	Zyklische Redundanzprüfung
d.h.	das heißt
DLC	Data Length Code
DS	Draft Standard; Normenentwurf
DSP	Draft Standard Proposal; Normentwurf-Vorschlag
dyn	dynamisch; veränderliche Information in Abhängigkeit von Drehgebereigenschaft
EDS file	Standardisierte Datei, die die Funktionalität eines CANopen Geräts beschreibt
Drehgeber	Kurzform; steht hier als Synonym für Absolutwertdrehgeber
etc.	et cetera, und so weiter
evtl.	eventuell
Fa.	Firma
GND	Ground, Masse
i*	Platzhalter für Informationen, die vom individuellen Drehgeber abhängen
Idx	Sub-Index

LED	Light Emitting Diode
LSB	Least Significant Bit/Byte; niederwertigstes Bit/Byte
LSS	Layer Setting Services
MSB	Most Significant Bit/Byte; höchstwertigstes Bit/Byte
MT	Mutliturn
n.n.	nicht nötig
NMT	Network-Management
Node-ID	Teil der CAN-ID; ID des Drehgebers im CAN-Netzwerk
OSI	Open Systems Interconnection (Schichtenmodell)
PDO	Process Data Object. Kommunikations-Objekt zum Austausch von Prozessdaten
res.	reserviert
ro	Read Only: Parameter ist nur lesbar, kann sich aber verändern
RTR	Remote Transmission Request; Datenanforderungsnachricht
rw	Read/Write: Parameter kann gelesen und geschrieben werden
S.	Seitenverweis oder auch „siehe Seite“
SDO	Service Data Object; Kommunikations-Objekt zum Zugriff auf das Objektverzeichnis
ST	Singleton
SYNC	Synchronisations-Nachricht
uvm.	und vieles mehr
vgl.	vergleiche
wo	Write Only: Parameter kann nur geschrieben werden
xxb	Zeichen dafür, dass Voranstehende Zahl (xx) eine Binärzahl ist
xxd	Zeichen dafür, dass Voranstehende Zahl (xx) eine Dezimalzahl ist
xxh	Zeichen dafür, dass Voranstehende Zahl (xx) eine Hexadezimalzahl ist
z.B.	zum Beispiel

1 Einleitung

1.1 Drehgebertypen

Dieses Originalhandbuch ist folgenden Drehgebertypen der Firma MEGATRON zuzuordnen:

HTB36E, FHB58 mit CANopen Interface

Es gilt für alle HTB und FHB CANopen Drehgeber mit der Revision Number (=Revisions Nummer, Softwareversion) 2.08 und kleiner.

Die CANopen Vendor-ID von MEGATRON Elektronik GmbH & Co. KG ist:
0x000004F9

Der Product Code für MEGATRON Drehgeber
Singleturn:

HTB/FHB-ST-CA: 0x46485442

Multiturn:

HTB/FHB-PM-CA: 0x46485442

Die Revision Number und die Serial Number sind für jeden einzelnen Drehgeber unterschiedlich. Sie sind vom dem Drehgeber-Etikett identifizierbar:

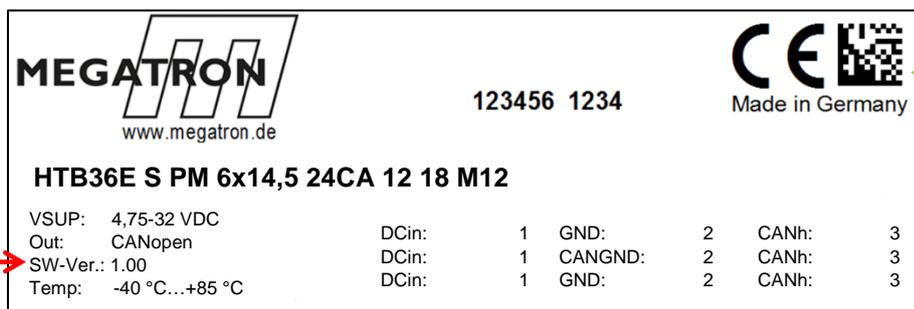


Abbildung 1.1: Illustration eines Drehgeberetiketts

In der Abbildung 1.1 die Software Revision Number ist mit dem roten Pfeil markiert (hier: 1.00). Sie ist mit einem führenden 0306 in der Software festgeschrieben. (z. B. 1.00 = 0306 0100h; 2.08 = 0306 0208h).

Die Serial Number ist grün markiert und liegt in Form eines QR codes vor (beispielsweise: 12345656). Mit einem QR-Code Reader kann diese Seriennummer als Dezimalwert ausgelesen werden.

Rechnet man diesen Dezimalwert in hex um, erhält man den Wert, der in der Software festgeschrieben ist (z. B. 12345656 = 00BC 6138h).

1.2 Zu diesem Handbuch

Dieses technische Handbuch beschreibt die Konfigurations- und Montagemöglichkeiten der MEGATRON Drehgeber HTB36E und HTB58 mit CANopen Profil. Es ist eine Ergänzung zu den anderen öffentlich verfügbaren MEGATRON Dokumenten, wie z. B. den Datenblättern, Montageanleitungen, Katalogen und Flyern. Wir empfehlen das Handbuch vor der Inbetriebnahme zu lesen. Vor der Verwendung sollte geprüft werden, ob eine aktuellere Version dieses Handbuchs verfügbar ist. Achten Sie beim Lesen besonders auf die Sicherheitshinweise am Anfang des Dokuments und den mit Warnsymbolen gekennzeichneten Textblöcken innerhalb der Handbuchabschnitte.

Punkt 4 Schnellstart zeigt eine Variante der Konfiguration des Drehgebers mit grundlegenden Einstellungen für minimale Funktionalität. Für eine optimale Nutzung des Drehgebers werden aber alle Informationen der nachfolgenden Kapitel benötigt und sollten daher gelesen werden. Zu Beginn des Handbuches werden spezifische Abkürzungen und Fachwörter erklärt.

Dieses Handbuch richtet sich an Personen mit technischen Kenntnissen im Umgang mit Sensoren, CANopen Schnittstellen und Automatisierungselementen.

Bewahren Sie die mit unserem Produkt gelieferten Informationen gut auf, so dass Sie sich, wenn nötig, weiter oder zu einem späteren Zeitpunkt erneut informieren können.

1.2.1 Symbolerklärung

	<ul style="list-style-type: none">• Das INFO-Symbol steht neben einem Abschnitt, der besonders informativ oder wichtig für das weitere Verfahren mit dem Gerät ist.
	<ul style="list-style-type: none">• Das WICHTIG-Symbol steht neben einer Textstelle, in der ein Verfahren zum Lösen eines bestimmten Problems beschrieben wird.
	<ul style="list-style-type: none">• Das WARN-Symbol steht neben einer Textstelle, die besonders zu beachten ist, um den ordnungsgemäßen Einsatz zu gewährleisten und vor Gefahren zu schützen.

1.3 Leistungsbeschreibung

Der Drehgeber ist ein Sensor zur Erfassung von Winkelpositionen und Umdrehungen. Die Messdaten und daraus abgeleitete Größen werden vom Drehgeber aufbereitet und als elektrische Ausgangssignale für die nachfolgende Peripherie bereitgestellt.

Die Schnittstelle, über die der Drehgeber kommuniziert, folgt der CAN- bzw. CANopen-Spezifikation. Der Drehgeber ist CAN 2.0A und CAN 2.0B fähig. Er erfüllt das Drehgeberprofil CiA 406 und ist für die industrielle Anwendung bestimmt.

Zur einfachen Konfiguration des Drehgebers können Sie die EDS (Electronic Data Sheet) Dateie nutzen, welche auf der Produktseite des Drehgebers auf <https://www.megatron.de> unter Downloads zu finden ist.

2 Sicherheitshinweise

2.1 Allgemeines

	<ul style="list-style-type: none">• Zur Inbetriebnahme des Drehgebers sind die Montageanleitungen, das Handbuch und das Datenblatt zu beachten.• Eine Nichtbeachtung der Sicherheitshinweise kann zu Fehlfunktionen, Sach- und Personenschaden führen!• Die Betriebsanleitung des Maschinenherstellers (applikationsseitig) ist zu beachten.
---	--

2.2 Bestimmungsgemäße Verwendung

Drehgeber sind Komponenten zum Einbau in Anlagen, Maschinen und Geräten. Vor der Inbetriebnahme (Betrieb in bestimmungsgemäßer Weise) muss festgestellt sein, dass die Anlage, die Maschine oder das Gerät als Ganzes der EMV- und ggf. der Maschinenrichtlinie entspricht.

Der Drehgeber ist ein Sensor zur Erfassung von Winkelpositionen und Umdrehungen und ist nur in diesem Sinne zu verwenden! Drehgeber der Firma MEGATRON werden für den industriellen Einsatz im nicht sicherheitsrelevanten Bereich gefertigt und vertrieben.

	<ul style="list-style-type: none">• Der Drehgeber darf nicht außerhalb der spezifizierten Grenzparameter betrieben werden (siehe zugehöriges Datenblatt).
---	---

2.3 Sicheres Arbeiten

Der Einbau und die Montage des Drehgebers darf ausschließlich durch eine Elektrofachkraft vorgenommen werden.

Zur Errichtung von elektrotechnischen Anlagen sind die nationalen und internationalen Vorschriften unbedingt zu befolgen.

Bei einer nicht fachgerechten Inbetriebnahme des Drehgebers, kann es zu Fehlfunktionen oder zum Ausfall kommen.

	<ul style="list-style-type: none">• Vor der Inbetriebnahme sind alle elektrischen Verbindungen zu prüfen.• Durch geeignete Sicherheitsmaßnahmen muss sichergestellt werden, dass bei Ausfall oder Fehlfunktion keine Personen zu Schaden kommen und es zu keiner Beschädigung der Anlage oder von Betriebseinrichtungen führt.
---	---

2.4 Entsorgung

Geräte, die nicht mehr benötigt werden oder defekt sind, müssen vom Nutzer unter Beachtung der länderspezifischen Gesetze fachgerecht entsorgt werden. Dabei ist zu berücksichtigen, dass es sich um Elektronik-Sonderabfall handelt und eine Entsorgung über den normalen Hausmüll nicht zulässig ist.

Es besteht keine Rücknahmeverpflichtung seitens des Herstellers. Bei Fragen zur ordnungsgemäßen Entsorgung wenden sie sich an einen Entsorgungs-Fachbetrieb in Ihrer Nähe.

3 Gerätebeschreibung

3.1 Allgemeines Drehgeber-Design

In Verbindung mit CANopen gibt die MEGATRON Drehgeberserien HTB36E im Ø36 mm Gehäuse und den FHB im Ø58 mm Gehäuse. HTB36E Drehgeber gibt es mit Vollwelle und Hohlwelle diese zeigt

Abbildung 3.1:



Abbildung 3.1: Drehgeberbauformen Vollwelle und Hohlwelle (Beispiel HTB36E)

Die Welle bzw. Hohlwelle des Drehgebers wird mit dem sich drehenden Teil in der Applikation verbunden, von welchem die Winkelposition und/oder Drehzahl gemessen werden soll. Kabel- oder Steckerabgänge bilden die Schnittstelle zum Anschluss an das CAN-Netzwerk. Die im Gehäusedeckel integrierte Status-LED signalisiert verschiedene Zustände des Drehgebers während des Einsatzes und unterstützt die Konfiguration des Drehgebers oder die Fehlersuche im Feldbus. Die Flanschbohrungen bzw. das mitgelieferte Federblech dient der Befestigung an der Maschine bzw. an der Anwendung.

3.2 Predefined Connection Settings

Services	COB-ID
NMT	000h
SYNC	080h
EMCY	080h + Node-ID
PDO1(tx)	180h + Node-ID
PDO2(tx)	280h + Node-ID
PDO3(tx)	380h + Node-ID
SDO(rx)	600h + Node-ID
SDO(rx)	580h + Node-ID

Tabelle 3.1: CAN-Identifizier

Die Drehgeber HTB und FHB werden standardmäßig mit der Node-ID 127 und Baudrate Auto-Detection ausgeliefert.

3.3 Leuchtdiode und Status-Signalisierung

Definition der LED Signalfarben:

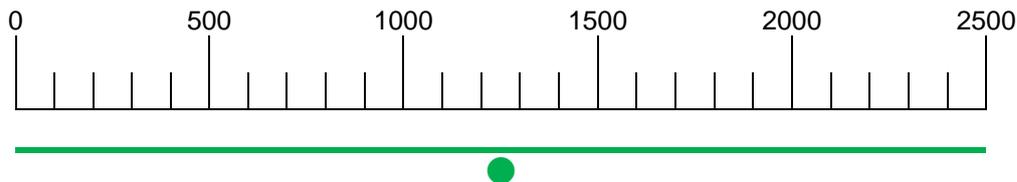
● = rot leuchtende LED = Information über „Physical Layer“

● = grün leuchtende LED = Information über den „NMT-Status“

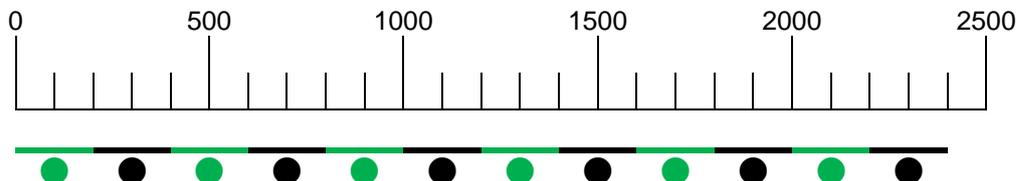
● = LED aus

→ = wird wie erster Zyklus fortgesetzt

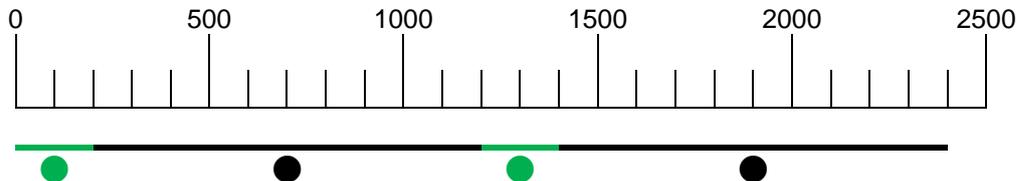
LED-Signalisierung [ms]:



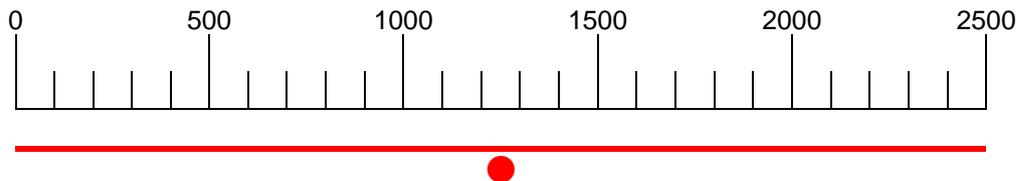
grünes Leuchten
Drehgeber ist im Operational-Status



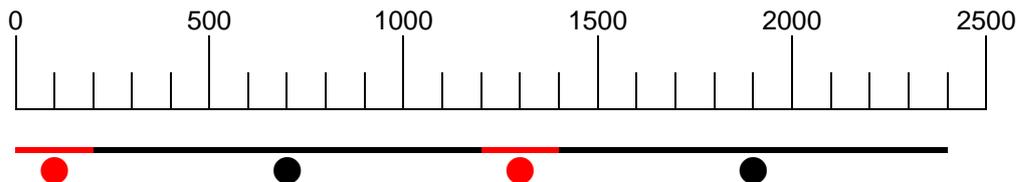
grünes Blinken
Drehgeber ist im Pre-Operational-Status



grünes Aufblitzen
Drehgeber ist im Stopped-Status



rotes Leuchten
NICHT Betriebsbereit / Bus-off



rotes Aufblitzen
Warnung, im Bus sind Error-frames versendet worden

Abbildung 3.2: LED Signale 1



rotes Doppelblitzen

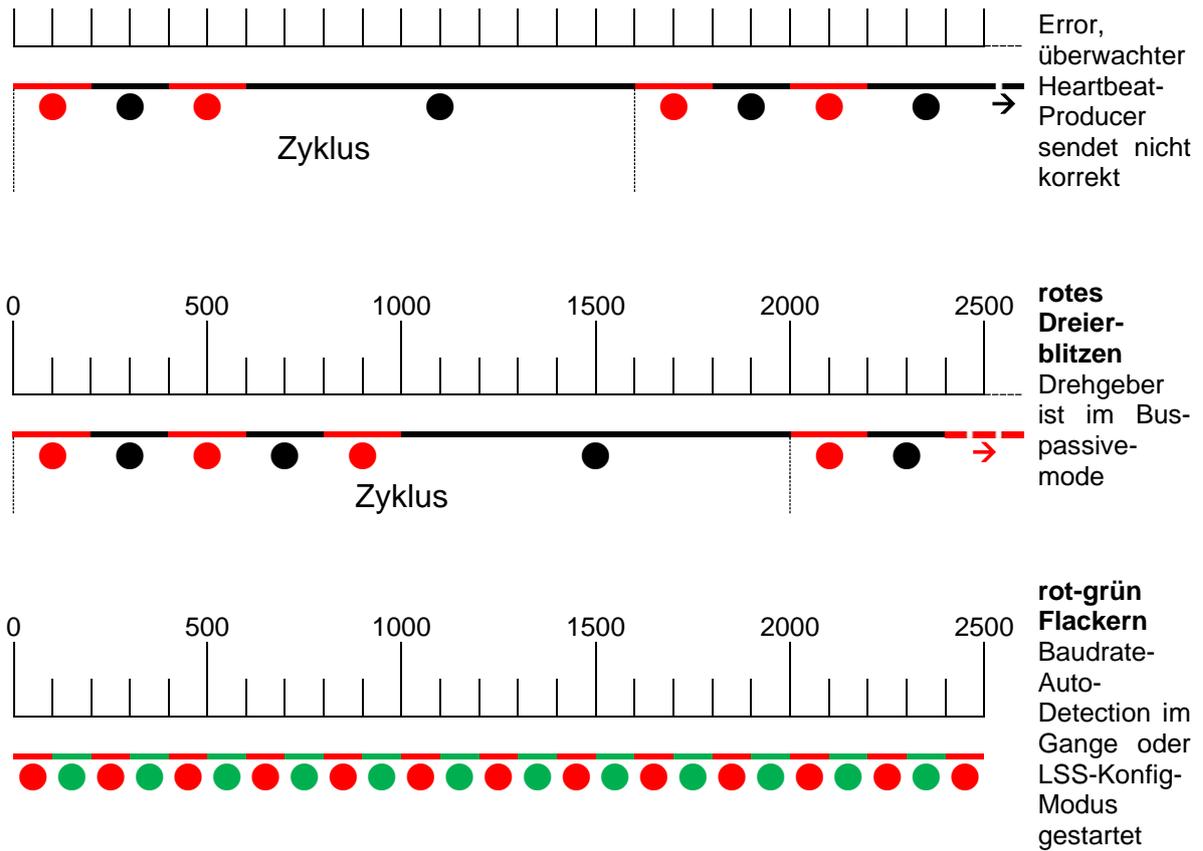


Abbildung 3.3: LED Signale 2

4 Schnellstart



- Der Drehgeber signalisiert jede Statusänderung mit seinem LED-Blinkverhalten. Siehe Kapitel 3.3 Leuchtdiode und Status-Signalisierung.

4.1 CAN-Netzwerk-Integration

MEGATRON Drehgeber HTB und FHB werden mit der Node-ID (Objekt 2101h Sub-Index: 00h) Wert: 127d ausgeliefert.

Damit der Drehgeber im CAN-Netzwerk arbeiten kann, muss die Baudrate eingestellt werden. Normalerweise geschieht dies mittels LSS (CiA DSP-305) oder über SDO-Befehle.

Bei den Drehgebern HTB und FHB wird die verwendete Baudrate im Bus jedoch automatisch erkannt (Objekt 2100h Sub-Index: 00h Wert: 09h - Baudrate-Auto-Detection), so dass keine Baudratenkonfiguration nötig ist. Um die im Bus verwendete Baudrate zu erkennen, bleibt der Drehgeber passiv und prüft Nachrichten auf dem Bus von anderen Teilnehmern, bis er deren Baudrate erkannt und selbst angenommen hat.

Da die Default Node-ID des Drehgebers evtl. in ihrem Netzwerk bereits vergeben sein könnte, raten wir Ihnen im Zweifelsfall, den Drehgeber nicht direkt in Ihre Anwendung zu implementieren, sondern zuerst 1:1 mit einem Master für die Konfiguration (z. B. Laptop mit passender Hard- und Software, voreingestellt auf die Sollbaudrate des CAN-Bus) zu verbinden und den Drehgeber über SDO oder LSS anzusprechen.

4.2 SDO-Befehl zur Einstellung Node-ID

Nachdem der Drehgeber HTB oder FHB mit dem CAN-Bus bzw. mit dem Master (z. B. Laptop s.o.) verbunden wurde, beginnt die LED mit „rot-grün Flackern“ (siehe Abbildung 3.3 LED Signale 2).

Schicken Sie als erstes eine SYNC-Nachricht:

080h	8	00h	00h	00h	00h	00h	00h	00h	00h
CAN-ID	DLC	Command	Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6

Abbildung 4.1: SYNC-Nachricht

Durch die SYNC-Nachricht erkennt der Drehgeber die benutzte Baudrate und nimmt diese an. Die LED beginnt nun mit grünem Blinken (siehe Abbildung 3.2).

Zum Setzen der Node-ID des Drehgebers muss das Objekt 2101h im Sub-Index 00h angepasst werden. (Dies ist nur im Pre-Operational-Status möglich!) Dazu ist ein einfacher SDO-Schreibbefehl mit einem Datenwert der gewünschten Node-ID (in hex) nötig.

600h+ID	8	2Fh	01h	21h	00h	Node-ID	00h	00h	00h
CAN-ID	DLC	Command	Object L	Object H	Sub-Index	Byte0	Byte1	Byte2	Byte3

Tabelle 4.2: SDO-Schreibbefehl zum Setzen der Node-ID

Beispiele für eine Node-ID des Drehgebers können sein:

Node-ID (d)	Node-ID (h)
1	01h
2	02h
...	...
4	04h
...	...
127	7Fh

Tabelle 4.3: Node-ID in dezimaler und hexadezimaler Form

Das Senden des SDO-Schreibbefehls stellt die Node-ID des Drehgebers ein. Eine Änderung der Node-ID des Drehgebers über SDO wird erst nach dem manuellen Reset des Drehgebers (Spannungs-Reset oder NMT-Reset) wirksam. Das Schreiben in Objekt 2101h bewirkt ein automatisches Speichern im EEPROM. Somit ist kein manuelles Speichern nötig. Das Einstellen der Node-ID des Drehgebers mit LSS wird in Kapitel 8 Inbetriebnahme beschrieben.

	<ul style="list-style-type: none"> Das Ändern der Node-ID bewirkt ein automatisches Anpassen der PDO und EMCY COB-IDs. Nach dem ersten manuellen Speichern werden diese auf ihrem derzeitigen Wert eingefroren und nicht mehr automatisch angepasst. Durch das Ausführen des „Restore Defaults“ Befehls wird die automatische Anpassung wieder aktiviert.
---	--

4.3 Drehgeber-Inbetriebnahme

Implementieren Sie den Drehgeber in Ihrer Anwendung. Bitte beachten Sie dabei die dem Drehgeber beiliegenden Montage- und Sicherheitshinweise. Weitere Informationen dazu finden Sie auch im Kapitel 8 Inbetriebnahme.

Wenn der Drehgeber in der Anwendung vollständig integriert und am Feldbus angeschlossen ist, können Sie ihn mit dem „Start-All-Nodes-Befehl“ (vgl. 7.1 Netzwerkmanagementbefehle) starten.

Der Drehgeber ist nun betriebsbereit (LED leuchtet grün) und wird Daten mittels verschiedener Prozess-Daten-Objekte (PDO) senden. Die Default-Einstellungen des Drehgebers sehen vor, dass das PDO1 gesendet wird, sobald sich der Messwert des Drehgebers ändert. Übertragen wird der Positionswert (Objekt 6004h) im Format Unsigned32. PDO2 wird standardmäßig mit dem gleichen Wert auf eine SYNC-Nachricht antworten. Das Heartbeat-Protokoll ist standardmäßig ausgeschaltet, so dass kein Heartbeat ausgesendet wird. Damit ist eine anwendungstaugliche Grundkonfiguration des Drehgebers erreicht. Er ist nun betriebsbereit.

5 Allgemeine CAN-Informationen

5.1 CAN-Physik

CAN ist ein Feldbus. Der CAN-Bus funktioniert nach dem CSMA/CA (Carrier Sense Multiple Access / Collision Avoidance) Verfahren. Das bedeutet, dass Nachrichten-Kollisionen auf dem Bus beim Buszugriff durch die sog. Bit-Arbitrierung vermieden werden. Die Daten bzw. Bits werden in NRZ-L (Non Return To Zero - Low) codiert.

Zur Datensicherung werden eine zyklische Redundanzprüfung und andere Sicherheitsvorkehrungen verwendet. Die Synchronisierung der Busteilnehmer wird über ein sog. „Bit-stuffing“ gewährleistet. CAN ist ein Multimaster-System. Das bedeutet, dass alle Busteilnehmer gleichberechtigt über den Bus verbunden sind und die Kommunikation nicht von einem einzelnen Master gesteuert werden muss.

Der CAN-Bus mit Kupferleitung-Ausführung arbeitet auf Basis von Differenzsignalen. Die Differenzsignale werden normalerweise über zwei Leitungen übertragen: CAN_{HIGH}, CAN_{LOW}. Der Pegel CAN_{HIGH} ist komplementär zu CAN_{LOW}. Gleichtaktstörungen werden so optimal unterdrückt.

Die Übertragung der Daten erfolgt so, dass ein Bit, je nach Zustand, entweder dominant oder rezessiv auf den Bus wirkt. Ein dominantes (0) überschreibt dabei ein rezessives Bit (1).

Das CAN-Netzwerk an sich basiert auf der Linienstruktur. Diese kann durch Stichleitungen erweitert werden. Stichleitungen sind jedoch nur in eingeschränktem Umfang zulässig (bis zu einer maximalen Länge von 0,5 m). Es müssen immer zwei Abschlusswiderstände von je 120 Ohm (zwischen CAN_{HIGH} und CAN_{LOW}) an dem jeweiligen Ende verwendet werden. Andere Positionen oder Widerstandsgrößen sind nicht zulässig.

Die oben angesprochene Arbitrierung(*) wird zur Ordnung des Buszugriffes auf Basis der CAN-Identifizier der zu sendenden Nachrichten genutzt. Jeder Teilnehmer überwacht ständig den Bus. Senden zwei Teilnehmer gleichzeitig, so setzt sich der Teilnehmer durch, der den „stärkeren“ Identifizier hat. Der andere Teilnehmer erkennt, wenn sich ein anderer „stärkerer“ Identifizier durchgesetzt hat, und stoppt das Senden eigener Informationen, bis es auf dem Bus wieder „still“ ist (siehe Abbildung 5.1). Technisch gesehen überlagert das erste dominante Bit (=stärkere Bit) das entsprechend rezessive des anderen. Verwenden beide Teilnehmer einen gleichen CAN-Identifizier, wird ein Error ausgegeben, sobald eine Kollision innerhalb des restlichen Nachrichtenteils entsteht. Grundsätzlich darf ein CAN-Identifizier nur von maximal einem Teilnehmer verwendet werden!

(*)Arbitrierung

Die Arbitrierung, Arbitration, ist ein Zugangsverfahren für Netzwerke, bei dem sich die Teilnehmer nach einer gegenseitigen Vereinbarung das Zugangsrecht zuteilen. Bei der Arbitrierung hat jedes an ein Netzwerk angeschlossene Gerät generell die gleichen Rechte. Erst die Verhandlung eines Gerätes mit allen anderen sichert diesem den temporären Zugang. Die Rechteverteilung kann durch die Vergabe von Prioritäten bestimmte Nutzer bevorzugen. Dabei wird jedem Gerät eine Priorität zugeordnet, die diesem eine Priorität im Zugriff auf das Netz oder eine Netzressource einräumt

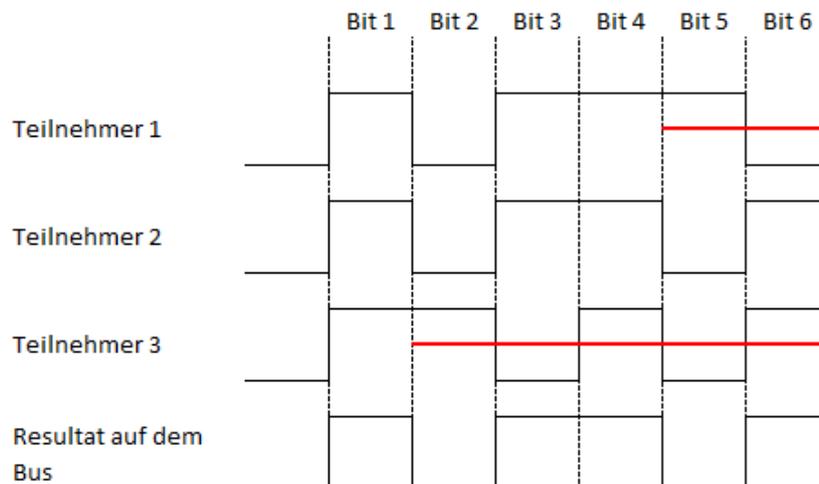


Abbildung 5.1: Beispiel einer Arbitrierung

Durch Nutzen der Arbitrierung stellt sich eine Hierarchie der Nachrichten untereinander ein. Die Nachricht mit dem niedrigsten Identifier hat höchste Priorität und wird immer Buszugriff erhalten. Für die Übertragung von zeitkritischen Nachrichten muss also ein CAN-Identifier hoher Priorität (mit niedrigem Wert) vergeben werden. Der Sendezeitpunkt kann aber selbst bei Nachrichten hoher Priorität nicht genau vorher bestimmt werden, da gerade in Übertragung befindliche Nachrichten nicht unterbrochen werden (nichtdeterministisches Verhalten).

Um eine Arbitrierung zu ermöglichen, müssen alle Teilnehmer zeitsynchronisiert senden. Bei Nachrichten mit vielen gleichen Bits hintereinander kann diese Synchronisation verloren gehen. Um dies zu verhindern, wird das sog. „Bitstuffing“ eingesetzt. Hier wird nach fünf gleichen Bits ein komplementäres Bit hinzugefügt (ohne, dass es in der Nachricht vorkommt). Durch die entstehenden Flanken können sich so alle Teilnehmer am Bus resynchronisieren (siehe Abbildung 5.2).

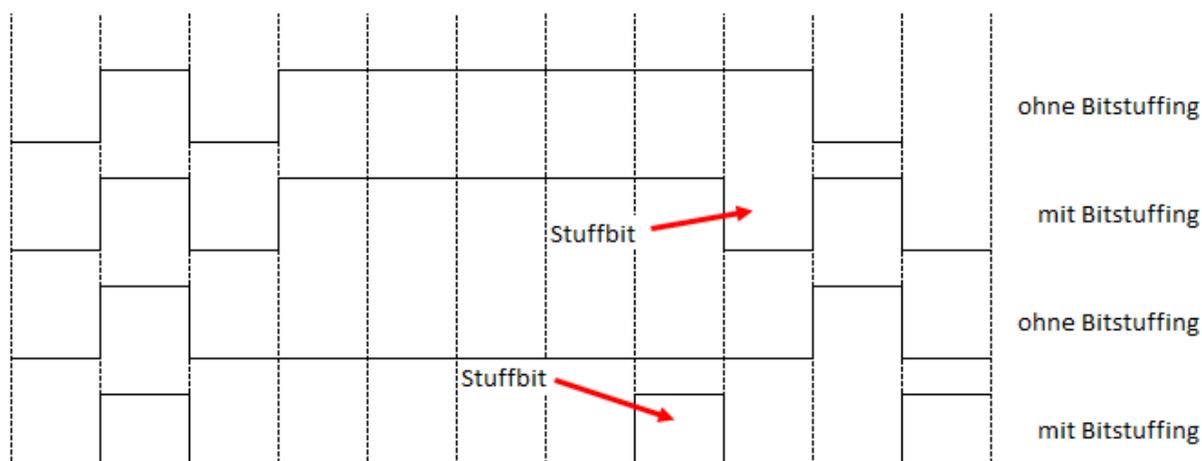


Abbildung 5.2: Bitstuffing

Ein CAN-Netzwerk kann Baudraten bis zu 1 MBit/s übertragen. Alle Teilnehmer müssen die Bits einer Nachricht gleichzeitig verarbeiten können. Dadurch ist die maximale Kabellänge abhängig von der Baudrate. Es gibt hierzu eine allgemeine Zuordnung der empfohlenen Baudraten und der entsprechenden maximalen Kabellänge.

Baudraten	Kabellänge
10 kBit/s	6,7 km
20 kBit/s	3,3 km
50 kBit/s	1,3 km
125 kBit/s	530 m
250 kBit/s	270 m
500 kBit/s	130 m
1 MBit/s	<40 m

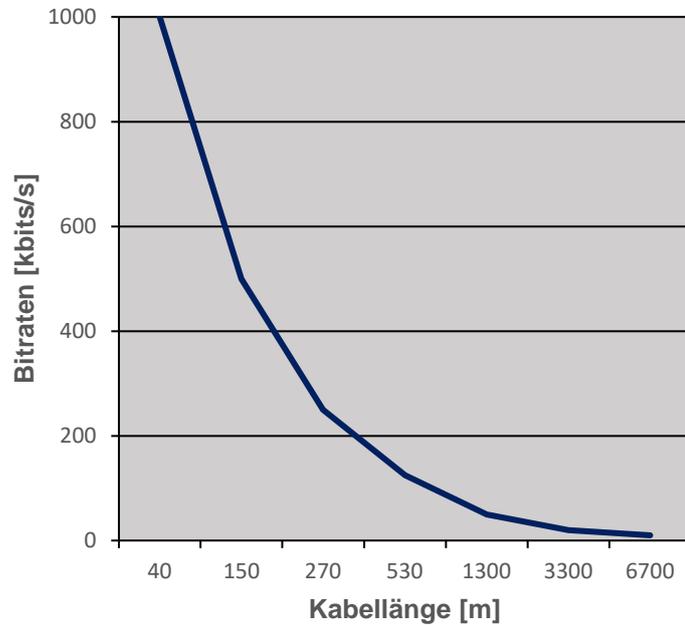


Tabelle 5.1: CAN Baudraten und Kabellängen

5.2 CANopen

CANopen ist die genormte Anwendungsschicht des standardisierten Schicht 7 Protokolls (Abbildung 5.3).

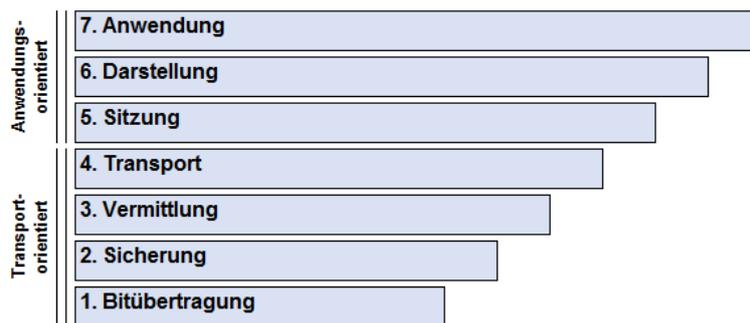


Abbildung 5.3: ISO-OSI-Schichtmodell

Mit CANopen ist es möglich große Datenmengen, Alarm-Nachrichten und Prozessdaten zu übermitteln. CANopen legt die Art der Kommunikation fest. Das bedeutet, dass Parameter zur Einstellung des Gerätes über eine definierte Schnittstelle übertragen werden (Profil).

Profile in CANopen bestehen aus mehreren Objekten, die in Tabellenform als sog. Objektverzeichnis organisiert sind.

Es gibt das Kommunikationsprofil, über das die grundlegenden Daten des Drehgebers abfragbar bzw. einstellbar sind. Beispiele hierfür sind Gerätebezeichnung, Versionsstände, verwendete CAN-Identifizier o.ä.

Das Geräteprofil beschreibt die besonderen Fähigkeiten des Geräts und „klassifiziert“ es. Das für den HTB oder FHB relevante Profil ist das Geräteprofil CiA 406.

5.3 Spezifikationen und Profile

5.3.1 Überblick

Die CANopen-Spezifikation wurde von der CiA im Draft Standard definiert. Besonders interessant in Bezug auf die Drehgeber HTB und FHB sind folgende Spezifikationen:

Spezifikation	Beschreibung
CiA 301	CANopen Kommunikationsprofil
CiA 303	Kabel, Darstellung physikalischer Einheit + Gerätefunktion
CiA 305	Einstellung Baudrate und Adresse über den Bus
CiA 306	Electronic Data Sheet
CiA 406	Geräte-/Applikationsprofil

Tabelle 5.2: Liste CiA Spezifikationen

5.3.2 Kommunikationsmechanismen

In CANopen gibt es verschiedene Kommunikationsservices:

SDO Service Data Object

Nutzung: Zur Statusabfrage und Änderungen im Objektverzeichnis. Es gibt einen SDO-Kanal, dem zwei Identifizier zugeordnet sind.

Ein SDO wird immer bestätigt, d. h., der Empfänger bestätigt den Empfang jeder erhaltenen SDO-Nachricht. Im Fehlerfall kann auch eine „Abort“-Nachricht gesendet werden. Die Delay-Time bis die Bestätigungsnachricht ausgesendet wird, beträgt bei den Drehgebern HTB und FHB maximal 1 Millisekunde.

PDO Prozess Data Object

Nutzung: Zur Prozessdatenübertragung. Ein PDO unterstützt die Nutzung einer vollen Länge einer CAN-Nachricht (8 Datenbytes), da ein PDO ohne Protokoll-Overhead auskommt. PDOs werden nicht quittiert und sind für zeitkritische Applikationen einsetzbar.

Durch Ausschöpfen des vollen 8 Datenbyte Umfanges für Nutzdaten, fällt die Protokollinformation weg. Dies hat zur Folge, dass das Format zwischen dem PDO-Producer und dem Consumer bei der Konfiguration vereinbart werden muss. Dies geschieht über ein PDO-Mapping.

PDOs können auf verschiedene Weise gesendet werden:

- **Auf Anforderung:** Ein anderer Busteilnehmer fordert über einen RTR Daten an. (Von der Nutzung des RTR rät die CiA ab, daher wird das RTR von MEGATRON nicht unterstützt!)
- **Im Synchronmodus:** Bei Empfang einer Synchronisationsnachricht (SYNC) eines anderen Busteilnehmers werden eigenständig PDOs ausgesendet.
- **Im Asynchronmodus:** Durch ein internes Ereignis (z. B. Messwertänderung, interner Event-Timer o. ä.) wird eine PDO-Nachricht ausgelöst.

5.3.3 Objektverzeichnis

Das Objektverzeichnis (= Daten-Matrix für Parameter) listet alle Datentypen und Objekte des Kommunikations- und Geräteprofils auf. Ebenso sind hier die herstellerspezifischen Objekte gelistet. Die Adressierung erfolgt über 16-Bit-Indizes (Zeilen) und 8-Bit-Sub-Indizes (Spalten).

Die Struktur des Objektverzeichnisses zeigt Tabelle 5.3:

Index(hex)	Objekt Beschreibung
0000	reserviert
0001 001F	statische Datentypen
0020 003F	komplexe Datentypen
0040 005F	herstellerspezifische Datentypen
0060 007F	profilspezifische statische Datentypen
0080 009F	profilspezifische komplexe Datentypen
00A0 0FFF	reserviert
1000 1FFF	Kommunikationsprofil
2000 5FFF	herstellerspezifische Parameter
6000 9FFF	Parameter aus den "Standard Profilen"
A000 AFFF	Netzwerk-Variable
B000 FFFF	reserviert / Systemvariablen

Tabelle 5.3: Struktur des Objektverzeichnisses

5.4 Netzwerkmanagement (NMT)

Im CANopen-System gibt es immer einen Netzwerkmanagement-Master. Der NMT-Master steuert alle anderen dem Feldbus zugehörigen Geräte in ihren NMT-Zuständen.

Ein solches Gerät kann in drei verschiedene Zustände gesetzt werden. Diese Zustände sind:

- Pre-Operational
- Operational
- Stopped

	<ul style="list-style-type: none"> • Nach dem Start eines CANopen-Geräts, in dem es selbstständig die interne Kommunikation und Applikationen initialisiert, wechselt es in den Pre-Operational-Status. Von hier aus kann das Gerät vom Master über NMT-Befehle in die anderen Zustände gesetzt werden. Um anzuzeigen, dass das Gerät nach dem Start betriebsbereit und im Pre-Operational-Status ist, wird eine „Boot-up-message“ ausgesendet, in der der CAN-Identifizier des Error-Control-Protokolls benutzt wird. Diese Message ist fest an die eingestellte Geräteadresse gebunden.
--	--

Die Zustände eines Gerätes werden im Folgenden beschrieben:

Pre-Operational	
Objekt	Kommunikation
SDO	JA
PDO	NEIN
NMT	JA
SYNC	NEIN
EMCY	JA
Heartbeat	JA
Kommunikation mit dem Gerät per SDO ist möglich. PDO Kommunikation ist nicht möglich.	

Tabelle 5.4: Mögliche Kommunikation – Pre-Operational

Operational	
Objekt	Kommunikation
SDO	JA
PDO	JA
NMT	JA
SYNC	JA
EMCY	JA
Heartbeat	JA
Gerät ist vollkommen Betriebsbereit und kann PDOs senden und empfangen.	

Tabelle 5.5: Mögliche Kommunikation – Operational

Stopped	
Objekt	Kommunikation
SDO	NEIN
PDO	NEIN
NMT	JA
SYNC	NEIN
EMCY	NEIN
Heartbeat	JA
Das Gerät ist vollständig von der Kommunikation getrennt. Gerät kann über ein NMT-Kommando nur in einen anderen Zustand gebracht werden (z. B. Start Node).	

Tabelle 5.6: Mögliche Kommunikation – Stopped

5.5 Heartbeat und Node-Guarding

Es gibt zwei Möglichkeiten die Verfügbarkeit und Betriebsbereitschaft eines CAN-Geräts während des Betriebs zu prüfen.

- Heartbeat
- Node-Guarding

Das Heartbeat-Protokoll ist unabhängig vom Master. Es gilt als bessere Lösung. Hierbei sendet das Gerät zyklisch eine „Lebens“-Nachricht aus.



- MEGATRON rät zur Nutzung des Heartbeats.

Das Node-Guarding-Protokoll sieht vor, dass der NMT-Master Remoteframes an die vorhandenen Slaves sendet und innerhalb einer bestimmten Zeit die Antwort eines jeden erwartet. Fehlt eine Antwort, so erkennt dies der Master. Node-Guarding führt zu einer hohen Abhängigkeit von der Betriebsbereitschaft des Masters, da dessen Ausfall zu einem Ausfall des Gesamtnetzwerkes führt.



- Eine Variante des Heartbeats ist die Bootup-Nachricht. Diese wird einmalig zum Start des Geräts ausgesendet und enthält keine Informationen (Datenbereich ist 00h). Einzig über die COB-ID der Nachricht ist die Node-ID des Aussenders zu erkennen (COB-ID = 700h + Node-ID).

5.6 Emergency-Nachrichten

Fehler im CAN-Gerät werden durch „verschiedene“ Emergency-Nachrichten gemeldet. Die Nachricht beinhaltet einen Code, anhand dessen der Fehler mittels der Funktionsbeschreibung des Gerätes eindeutig identifiziert werden kann. Ein CAN-Gerät kann auch so eingestellt werden, dass keine Emergency-Nachrichten ausgesendet werden.

6 Objektverzeichnis HTB36E oder FHB58

6.1 Kommunikationsspezifische Objekte

Die kommunikationsspezifischen Objekte folgen der Spezifikation CiA 301 V4.02 und können maximal die Objektadressen 1000h bis 1FFFh beinhalten.

Objekt	Name	Idx	Beschreibung	Daten Größe (Bit)	ro rw co	Map	Default
1000h S. 38	Device type	0h	MSB = Drehgeber Typ; LSB = Geräte Profil Nr.	Unsigned32	co	no	Multiturn: 0002 0196h Singleturn: 0001 0196h
1001h S. 34	Error Register	0h	Signalisierung von internen Fehlern	Unsigned8	ro	yes	00h
1002h S. 35	Manufacturer status register	0h	Allgemeines Statusregister (herstellerspezifisch)	Unsigned32	ro	yes	dyn.
1003h	PreDefined Error Field	00h	Speichert die ersten, via EMCY signalisierten, Errors im EEPROM	Unsigned8	rw	no	dyn.
		01h	Standard error field 1	Unsigned32	ro		
		02h	Standard error field 2	Unsigned32	ro		
		03h	Standard error field 3	Unsigned32	ro		
		04h	Standard error field 4	Unsigned32	ro		
		05h	Standard error field 5	Unsigned32	ro		
1005h S. 38	COB-ID SYNC-Nachricht	00h	CAN-Identifizier der SYNC-Nachricht	Unsigned32	rw	no	0000 0080h
1008h S. 39	Manufacturer device name	00h	Hersteller Gerätebezeichnung	string256	co	no	HTB/FHB-PM-CA HTB/FHB-ST-CA
1009h	Manufacturer Hardware-Version	00h	Enthält die Geräte Hardware-Version	string16	co	co	i*
100Ah	Manufacturer Software-Version	00h	Enthält die Geräte Software-Version	string72	co	no	i*
100Ch	Guard time	00h	Zeitbasis (in ms), die in Verbindung mit 100Dh die Zeit ergibt, in der die Antwort des Node-Guards erwartet wird. Wert 0000h = deaktiviert	Unsigned16	rw	no	0000h
100Dh	Life time factor	00h	Faktor, der in Verbindung mit 100Ch die Zeit ergibt, in der die Antwort des Node-Guards erwartet wird. Wert 00h = deaktiviert	Unsigned8	rw	no	00h

Tabelle 6.1: Objektverzeichnis 1000h – 100Dh

Objekt	Name	Idx	Beschreibung	Daten Größe (Bit)	ro rw co	Map	Default
1010h S. 70	Store Parameters	00h	Speichert Objektverzeichnis-Einstellungen	Unsigned8	co	no	04h
		01h	Alle Objektverzeichnis-Inhalte	Unsigned32	rw		0000 0001h
		02h	Kommunikationsspezifische Inhalte	Unsigned32	rw		0000 0001h
		03h	Applikationsspezifische Inhalte	Unsigned32	rw		0000 0001h
		04h	Herstellerspezifische Inhalte	Unsigned32	rw		0000 0001h
1011h S. 71	Restore default Parameters	00h	Stellt Werkseinstellungen wieder her	Unsigned8	co	no	04h
		01h	Alle Objektverzeichnis-Inhalte	Unsigned32	rw		0000 0001h
		02h	Kommunikationsspezifische Inhalte	Unsigned32	rw		0000 0001h
		03h	Applikationsspezifische Inhalte	Unsigned32	rw		0000 0001h
		04h	Herstellerspezifische Inhalte	Unsigned32	rw		0000 0001h
1014h S. 34	COB-ID Emergency object	00h	Definiert die COB-ID des Emergency Objekts (EMCY)	Unsigned32	rw	no	0000 0080h+ Node-ID
1015h S. 35	Inhibit time EMCY	00h	Definiert die Pausenzeit (100 µs Schritte) zwischen dem Aussenden zweier EMCYs	Unsigned16	rw	no	0000h
1016h S. 33	Consumer heartbeat time	00h	Definiert die vom Heartbeat-Consumer erwartete Heartbeat-Zyklus-Zeit	Unsigned8	co	no	01h
		01h	Definition der Heartbeat-Zyklus-Zeit in ms	Unsigned32	rw		0000 0000h
1017h S. 33	Producer heartbeat time	00h	Definiert die vom Heartbeat-Producer genutzte Heartbeat-Zyklus-Zeit in ms. Wert 0000h = deaktiviert	Unsigned16	rw	no	0000h
1018h S. 1	Identity Object	00h	Daten zur Produktidentifikation	Unsigned8	co	no	04h
		01h	Vendor-ID	Unsigned32	co		0x000004F9
		02h	Product Code (HTB, FHB)	Unsigned32	co		0x485442
		03h	Revision Number	Unsigned32			i*
		04h	Serial Number	Unsigned32			i*
1020h S. 40	Verify Configuration	00h	Zeitpunkt der letzten Konfiguration auslesen bzw. speichern	Unsigned8	co	no	02h
		01h	Configuration date	Unsigned32	rw		0000 0000h

		02h	Configuration time	Unsigned32	rw		0000 0000h
--	--	-----	--------------------	------------	----	--	------------

Tabelle 6.2: Objektverzeichnis 1010h – 1020h

Objekt	Name	Idx	Beschreibung	Daten Größe (Bit)	ro rw co	Map	Default
1029h S. 39	Error behavior	00h	Fehlerverhalten ändern bei Node-Guarding, Heartbeat, etc.	Unsigned8	co	no	02h
		01h	Kommunikation Error	Unsigned8	rw		00h
		02h	Drehgeber Error	Unsigned8	rw		00h
1800h S. 59	Transmit PDO Communication Parameter	00h	Definiert die Kommunikations-Parameter für das erste TPDO	Unsigned8	co	no	05h
		01h	COB-ID des PDO	Unsigned32	rw		180h + Node-ID
		02h	Transmission Type	Unsigned8	rw		Feh
		05h	Event-Timer	Unsigned16	rw		0000h
1801h S. 59	Transmit PDO Communication Parameter	00h	Definiert die Kommunikations-Parameter für das 2. TPDO	Unsigned8	co	no	05h
		01h	COB-ID des PDO	Unsigned32	rw		280h + Node-ID
		02h	Transmission Type	Unsigned8	rw		01h
		05h	Event-Timer	Unsigned16	rw		0000h
1802h S. 59	Transmit PDO Communication Parameter	00h	Definiert die Kommunikations-Parameter für das 3. TPDO	Unsigned8	co	no	05h
		01h	COB-ID des PDO	Unsigned32	rw		380h + Node-ID
		02h	Transmission Type	Unsigned8	rw		01h
		05h	Event-Timer	Unsigned16	rw		0000h
1803h S. 59	Transmit PDO Communication Parameter	00h	Definiert die Kommunikations-Parameter für das 4. TPDO	Unsigned8	co	no	05h
		01h	COB-ID des PDO	Unsigned32	rw		480h + Node-ID
		02h	Transmission Type	Unsigned8	rw		01h
		05h	Event-Timer	Unsigned16	rw		0000h
1A00h S. 63	TPDO mapping Parameter	00h	Definiert das PDO-mapping für das erste TPDO	Unsigned8	rw	no	01h
		01h	Mapping von Objekt 1 in der Applikation	Unsigned32	rw		6004 0020h
	<i>Inaktiv durch Sub-index 00h</i>	02h bis 08h	Mapping von Objekt 2-8 in der Applikation	Unsigned32	rw		
1A01h S. 63	TPDO mapping Parameter	00h	Definiert das PDO-mapping für das 2. TPDO	Unsigned8	rw	no	01h
		01h	Mapping von Objekt 1 in der Applikation	Unsigned32	rw		6004 0020h

	<i>Inaktiv durch Sub-index 00h</i>	02h bis 08h	Mapping von Objekt 2-8 in der Applikation	Unsigned32	rw		0000 00h
--	------------------------------------	-------------	---	------------	----	--	----------

Tabelle 6.3: Objektverzeichnis 1029h – 1A01h

Objekt	Name	Idx	Beschreibung	Daten Größe (Bit)	ro rw co	Map	Default
1A02h S. 63	TPDO mapping Parameter	00h	Definiert das PDO-mapping für das 3. TPDO	Unsigned8	rw	no	01h
		01h	Mapping von Objekt 1 in der Applikation	Unsigned32	rw		6008 0020h
	<i>Inaktiv durch Sub-index 00h</i>	02h bis 08h	Mapping von Objekt 2-8 in der Applikation	Unsigned32	rw		
1A03h S. 63	TPDO mapping Parameter	00h	Definiert das PDO-mapping für das 4. TPDO	Unsigned8	rw	no	00h
	<i>Inaktiv durch Sub-index 00h</i>	01h bis 08h	Mapping von Objekt 1-8 in der Applikation	Unsigned32	rw		
1F80h S. 39	NMT-Startupverhalten	00h	NMT-Startupverhalten des Gerätes definieren	Unsigned32	rw	no	0000 0000h

Tabelle 6.4: Objektverzeichnis 1A03h – 1F80h

(S. = Seitenverweis; Idx = Sub-Index; ro / rw / co = Zugriffsart; Map = PDO-Mapping; i* = individuell ; dyn = dynamisch; ST = Singleturn; MT = Multiturn)

6.2 Gerätespezifische Objekte

Die gerätespezifischen Objekte folgen dem CiA-Drehgeber-Profil 406 und können maximal die Objektadressen 6000h bis 9FFFh beinhalten.

Objekt	Name	Idx	Beschreibung	Daten Größe (Bit)	ro rw co	Map	Default
6000h S. 65	Operating Parameters	00h	Änderung/Anzeige der Betriebsparameter	Unsigned16	rw	no	0004h
6001h S. 66	Measuring units per revolution	00h	Änderung der Singleturn-Auflösung	Unsigned32	rw	no	0000 4000h
6002h S. 66	Total measuring range	00h	Änderung der Gesamt-Auflösung	Unsigned32	rw	no	i*
6003h S. 66	Preset value	00h	Änderung / Anzeige eines Preset-Wertes zur Nullpunkt-Anpassung	Unsigned32	rw	no	0000 0000h
6004h	Position value	00h	Ausgabewert der Position (ST + MT)	Unsigned32	ro	yes	dyn

6008h	High precision position value	00h	Ausgabewert der Position, wenn Messbereich > 32 Bit	Unsigned64	ro	yes	dyn
-------	-------------------------------	-----	---	------------	----	-----	-----

Tabelle 6.5: Gerätespezifische Objekte 6000h –6008h

Objekt	Name	Idx	Beschreibung	Daten Größe (Bit)	ro rw co	Map	Default
6009h	High precision Preset Value	00h	Änderung / Anzeige des High-precision-Preset-Wertes zur Nullpunkt-Anpassung	Unsigned64	rw	no	0000 0000 0000 0000h
6030h	Speed Value	00h	Geschwindigkeit in Vielfachen von Einheiten/sek	Unsigned8	ro	yes	01h
		01h	Aktueller Geschwindigkeitswert	Signed16	ro		dyn
6040h	Acceleration Vakue	00h	Beschleunigung in Vielfachen von Maßeinheiten/s ²	Unsigned8	ro	yes	01h
		01h	Aktueller Beschleunigungswert	Signed16	ro		dyn
6050h	Jerk Value	00h	Ruck in Vielfachen von Maßeinheiten/s ³	Unsigned8	ro	yes	01h
		01h	Aktueller Wert des Rucks	Signed16	ro		dyn
6200h	Cyclic-Timer	00h	Änderung / Anzeige der Transmissions Periode für TPDO1	Unsigned16	rw	no	0001h
6300h S. 36	CAM state register	00h	Anzeige des Status der Nocken	Unsigned8	ro	yes	01h
		01h	Nockenstatus (bitweise Codierung: 0b = inactiv, 1b = activ)	Unsigned8	ro		00h
6301h S. 37	CAM enable register	00h	Ein bzw. Ausschalten von einzelnen Nocken	Unsigned8	ro	no	01h
		01h	Nocken Ein- bzw. Ausschalter (bitweise Codierung: 0b = inactiv, 1b = activ)	Unsigned8	rw		00h
6302h S. 37	CAM polarity register	00h	Logikinvertierung einzelner Nocken im betreffendem CAM status (1b: inactiv <=> activ)	Unsigned8	ro	no	01h
		01h	CAM polarity 0b = CAM Status nicht invertiert, 1b = CAM Status invertiert	Unsigned8	rw		00h
6310h	CAM1 low limit	00h	Unterer Umschalt-punkt für den 1. CAM	Unsigned8	co	no	01h
		01h	Änderung des unteren Umschalt-	Signed32	rw		0000 0000h

			punktes für den 1. CAM				
--	--	--	---------------------------	--	--	--	--

Tabelle 6.6: Gerätespezifische Objekte 6009h –6310h

Objekt	Name	Idx	Beschreibung	Daten Größe (Bit)	ro rw co	Map	Default
6311h	CAM2 low limit	00h	Unterer Umschalt- punkt für den 2. CAM	Unsigned8	co	no	01h
		01h	Änderung des unteren Umschalt- punktes für den 2. CAM	Signed32	rw		0000 0000h
6312h	CAM3 low limit	00h	Unterer Umschalt- punkt für den 3. CAM	Unsigned8	co	no	01h
		01h	Änderung des unteren Umschalt- punktes für den 3. CAM	Signed32	rw		0000 0000h
6313h	CAM4 low limit	00h	Unterer Umschalt- punkt für den 4. CAM	Unsigned8	co	no	01h
		01h	Änderung des unteren Umschalt- punktes für den 4. CAM	Signed32	rw		0000 0000h
6314h	CAM5 low limit	00h	Unterer Umschalt- punkt für den 5. CAM	Unsigned8	co	no	01h
		01h	Änderung des unteren Umschalt- punktes für den 5. CAM	Signed32	rw		0000 0000h
6315h	CAM6 low limit	00h	Unterer Umschalt- punkt für den 6. CAM	Unsigned8	co	no	01h
		01h	Änderung des unteren Umschalt- punktes für den 6. CAM	Signed32	rw		0000 0000h
6316h	CAM7 low limit	00h	Unterer Umschalt- punkt für den 7. CAM	Unsigned8	co	no	01h
		01h	Änderung des unteren Umschalt- punktes für den 7. CAM	Signed32	rw		0000 0000h
6317h	CAM8 low limit	00h	Unterer Umschalt- punkt für den 8. CAM	Unsigned8	co	no	01h
		01h	Änderung des unteren Umschalt- punktes für den 8. CAM	Signed32	rw		0000 0000h
6320h	CAM1 high limit	00h	Obere Umschalt- punkt für den 1. CAM	Unsigned8	co	no	01h
		01h	Änderung des oberen Umschalt-punktes für den 1. CAM	Signed32	rw		0000 0000h

6321h	CAM2 high limit	00h	Obere Umschalt-punkt für den 2. CAM	Unsigned8	co	no	01h
		01h	Änderung des oberen Umschalt-punktes für den 2. CAM	Signed32	rw		0000 0000h

Tabelle 6.7: Gerätespezifische Objekte 6311h –6321h

Objekt	Name	Idx	Beschreibung	Daten Größe (Bit)	ro rw co	Map	Default
6322h	CAM3 high limit	00h	Obere Umschalt-punkt für den 3. CAM	Unsigned8	co	no	01h
		01h	Änderung des oberen Umschalt-punktes für den 3. CAM	Signed32	rw		0000 0000h
6323h	CAM4 high limit	00h	Obere Umschalt-punkt für den 4. CAM	Unsigned8	co	no	01h
		01h	Änderung des oberen Umschalt-punktes für den 4. CAM	Signed32	rw		0000 0000h
6324h	CAM5 high limit	00h	Obere Umschalt-punkt für den 5. CAM	Unsigned8	co	no	01h
		01h	Änderung des oberen Umschalt-punktes für den 5. CAM	Signed32	rw		0000 0000h
6325h	CAM6 high limit	00h	Obere Umschalt-punkt für den 6. CAM	Unsigned8	co	no	01h
		01h	Änderung des oberen Umschalt-punktes für den 6. CAM	Signed32	rw		0000 0000h
6326h	CAM7 high limit	00h	Obere Umschalt-punkt für den 7. CAM	Unsigned8	co	no	01h
		01h	Änderung des oberen Umschalt-punktes für den 7. CAM	Signed32	rw		0000 0000h
6327h	CAM8 high limit	00h	Obere Umschalt-punkt für den 8. CAM	Unsigned8	co	no	01h
		01h	Änderung des oberen Umschalt-punktes für den 8. CAM	Signed32	rw		0000 0000h
6330h	CAM1 hysteresis	00h	Hysterese für die Umschaltpunkte des 1. CAM	Unsigned8	co	no	01h
		01h	Hysteresegröße ist abhängig von der Höhe dieses Wertes	Unsigned32	rw		0000 0000h
6331h	CAM2 hysteresis	00h	Hysterese für die Umschaltpunkte des 2. CAM	Unsigned8	co	no	01h
		01h	Hysteresegröße ist abhängig von der Höhe dieses Wertes	Unsigned32	rw		0000 0000h
6332h	CAM3 hysteresis	00h	Hysterese für die Umschaltpunkte des 3. CAM	Unsigned8	co	no	01h

		01h	Hysteresegröße ist abhängig von der Höhe dieses Wertes	Unsigned32	rw		0000 0000h
--	--	-----	---	------------	----	--	------------

Tabelle 6.8: Gerätespezifische Objekte 6322h –6332h

Objekt	Name	Idx	Beschreibung	Daten Größe (Bit)	ro rw co	Map	Default
6333h	CAM4 hysteresis	00h	Hysterese für die Umschaltpunkte des 4. CAM	Unsigned8	co	no	01h
		01h	Hysteresegröße ist abhängig von der Höhe dieses Wertes	Unsigned32	rw		0000 0000h
6334h	CAM5 hysteresis	00h	Hysterese für die Umschaltpunkte des 5. CAM	Unsigned8	co	no	01h
		01h	Hysteresegröße ist abhängig von der Höhe dieses Wertes	Unsigned32	rw		0000 0000h
6335h	CAM6 hysteresis	00h	Hysterese für die Umschaltpunkte des 6. CAM	Unsigned8	co	no	01h
		01h	Hysteresegröße ist abhängig von der Höhe dieses Wertes	Unsigned32	rw		0000 0000h
6336h	CAM7 hysteresis	00h	Hysterese für die Umschaltpunkte des 7. CAM	Unsigned8	co	no	01h
		01h	Hysteresegröße ist abhängig von der Höhe dieses Wertes	Unsigned32	rw		0000 0000h
6337h	CAM8 hysteresis	00h	Hysterese für die Umschaltpunkte des 8. CAM	Unsigned8	co	no	01h
		01h	Hysteresegröße ist abhängig von der Höhe dieses Wertes	Unsigned32	rw		0000 0000h
6400h	Area state register	00h	Anzahl der Statusbits des Arbeitsbereiches	Unsigned8	co	yes	01h
		01h	Status des Area state registers, 00h = im Arbeitsbereich, 02h oberhalb, 04h unterhalb des Arbeitsbereiches	Unsigned8	ro		dyn
6401h	Work area low limit	00h	Untere Grenze des Work area	Unsigned8	co	no	01h
		01h	Änderung des Work area low limits	Signed32	rw		0000 0000h
6402h	Work area high limit	00h	Obere Grenze des Work area	Unsigned8	co	no	01h
		01h	Änderung des Work area high limits	Signed32	rw		0000 4000h
6500h	Operating-status	00h	Status des Betriebszustands des Gerätes	Unsigned16	ro	no	dyn

Tabelle 6.9: Gerätespezifische Objekte 6333h –6500h

Objekt	Name	Idx	Beschreibung	Daten Größe (Bit)	ro rw co	Map	Default
6501h	Measuring units per revolution	00h	Anzeige der Singleturn-Auflösung	Unsigned32	co	no	0000 4000h
6502h	Number of distinguishable revolutions	00h	Anzeige der Multiturn-Auflösung	Unsigned16	co	no	ST: 0001h MT: FFFFh
6503h S. 36	Alarms	00h	Alarm bei Fehlfunktion	Unsigned16	ro	yes	dyn
6504h	Supported alarms	00h	Anzeige der im Drehgeber implementierten Alarme	Unsigned16	co	no	0001h
6505h S. 36	Warnings	00h	Warnung bei Abweichung von Betriebsparametern	Unsigned16	ro	yes	dyn
6506h	Supported warnings	00h	Anzeige der im Drehgeber implementierten Warnings	Unsigned16	co	no	7001h
6507h	Profile and software version	00h	die ersten 4 Stellen = Softwareversion, die nächsten 4 Stellen = Profil	Unsigned32	co	no	0105 0302h
6508h	Operating time	00h	nicht unterstützt	Unsigned32	co	no	FFFF FFFFh
6509h	Offset value	00h	Enthält den Offset-Wert, errechnet aus der Preset-Funktion (6003h)	Signed32	ro	no	0000 0000h
650Ah	Module identification	00h	Herstellerspezifischer Offset	Unsigned8	co	no	03h
		01h	Manufacturer offset value	Signed32	co		00h
		02h	Manufacturer min.-position	Signed32	co		-
		03h	Manufacturer max.-position	Signed32	co		-
650Bh	Serial number	00h	Anzeige der Seriennummer des Drehgebers, festgeschrieben mit Objekt 1018h-04h	Unsigned8	co	no	01h
		01h	Serial number	Unsigned32	co		i*
6510h	Number of high-precision-revolutions	00h	Anzeige der max. möglichen high-precision Multiturn-Auflösung	Unsigned40	co	no	0080 0000 0000h

Tabelle 6.10: Gerätespezifische Objekte 6501h –6510h

(S. = Seitenverweis; Idx = Sub-Index; ro / rw / co = Zugriffsart; Map = PDO-Mapping; i* = individuell; dyn = dynamisch; ST = Singleturn; MT = Multiturn)

6.3 Herstellerspezifische Objekte

Die Objekte 2000h bis 5FFFh sind herstellerspezifisch und werden nicht von der CiA festgelegt.

Objekt	Name	Idx	Beschreibung	Daten Größe (Bit)	ro rw co	Map	Default
2100h S. 56	Baudrate	00h	Ändern / Anzeigen der Baudrate	Unsigned8	rw	no	09h
2101h S. 57	Node-ID	00h	Ändern / Anzeigen der Node-ID	Unsigned8	rw	no	7Fh
2103h S. 40	BUS-Off Auto-Reset	00h	Definiert die Zeit Busoff der Drehgeber Selbstständig einen Reset durchführt. 0h = gar nicht, 01h-FFh = Sekunden	Unsigned8	rw	no	00h
2105h S. 67	Integration value	00h	Anzahl Filterschritte für Speed, Accelleration und Jerk	Unsigned8	rw	no	02h
		01h	Integration-Positionsfilter	Unsigned8	rw		01h
		02h	Integration-Geschwindigkeitsfilter	Unsigned32	rw		03E8h
2106h S. 67	Speed scaling	00h	Geschwindigkeitswert skalierung	Unsigned8	co	no	02h
		01h	Multiplikator	Unsigned16	rw		0001h
		02h	Divisor	Unsigned16	rw		0001h
2107h S. 68	Frquency-Limit	00h	Limit für Geschwindigkeitswert	Unsigned16	rw	no	FFFFh
2120h S. 40	Customer Flash area	00h	Objekt zum Speichern beliebiger Daten	Unsigned8	co	no	08h
		01h	Customer data 1	Unsigned32	rw		FFFF FFFF
		02h	Customer data 2	Unsigned32	rw		FFFF FFFF
		03h	Customer data 3	Unsigned32	rw		FFFF FFFF
		04h	Customer data 4	Unsigned32	rw		FFFF FFFF
		05h	Customer data 5	Unsigned32	rw		FFFF FFFF
		06h	Customer data 6	Unsigned32	rw		FFFF FFFF
		07h	Customer data 7	Unsigned32	rw		FFFF FFFF
		08h	Customer data 8	Unsigned32	rw		FFFF FFFF
2500h S. 40	Temperature Object	00h	Überwachung der Betriebstemperatur	Unsigned8	co	yes	05h
		01h	Aktueller Temperaturwert	Signed16	ro		dyn
		02h	Upper Limit	Signed16	rw		100°
		03h	Lower Limit	Signed16	rw		-40°
		04h	Aufgetretener Maximalwert	Signed16	ro		dyn
		05h	Aufgetretener Minimalwert	Signed16	ro		dyn

Tabelle 6.11: Herstellerspezifische Objekte 2100h –2500h

Objekt	Name	Idx	Beschreibung	Daten Größe (Bit)	ro rw co	Map	Default
2502h	Error History	00h	Aufzeichnung aufgetretener Errors	Unsigned32	co	no	dyn
		01h	Errorfeld 1	Unsigned32	ro		dyn
		02h	Errorfeld 2	Unsigned32	ro		dyn
		03h	Errorfeld 3	Unsigned32	ro		dyn
		04h	Errorfeld 4	Unsigned32	ro		dyn
		05h	Errorfeld 5	Unsigned32	ro		dyn
2503h	Alarms-History	00h	Aufzeichnung aufgetretener Alarms	Unsigned8	co	no	dyn
		01h	Alarmwert 1	Unsigned16	ro		dyn
		02h	Alarmwert 2	Unsigned16	ro		dyn
		03h	Alarmwert 3	Unsigned16	ro		dyn
		04h	Alarmwert 4	Unsigned16	ro		dyn
		05h	Alarmwert 5	Unsigned16	ro		dyn
2504h	Warnings-History	00h	Aufzeichnung aufgetretener Warnings 6505h	Unsigned8	rw	no	dyn
		01h	Warningwert 1	Unsigned16	ro		dyn
		02h	Warningwert 2	Unsigned16	ro		dyn
		03h	Warningwert 3	Unsigned16	ro		dyn
		04h	Warningwert 4	Unsigned16	ro		dyn
		05h	Warningwert 5	Unsigned16	ro		dyn

Tabelle 6.12: Herstellerspezifische Objekte 2502h –2504h

(S. = Seitenverweis; Idx = Sub-Index; ro / rw / co = Zugriffsart; Map = PDO-Mapping; i* = individuell; dyn = dynamisch; ST = Singleturn; MT = Multiturn)

7 Objektbeschreibung

7.1 Netzwerkmanagementbefehle

Um den Drehgeber von einem Zustand (Stopped, Pre-Operational, Operational) in einen anderen zu bringen, können verschiedene Kommunikationsbefehle genutzt werden. Die Transmissionen hierfür sind 3 Byte groß und werden nicht bestätigt. Die Kennung (CAN-ID) ist für NMT-Befehle immer NULL.

0	02h	Command	Node-ID
CAN-ID	DLC	Byte 0	Byte 1

Tabelle 7.1: Aufbau NMT-Befehle

Command:

Der Wert des Commands bestimmt, welche Aktion der oder die ausgewählten Teilnehmer ausführen sollen.

Command	Wert
Start Node	01h
Stop Node	02h
Pre-Operational	80h
Reset Node	81h
Reset Communication	82h

Tabelle 7.2: Commands für NMT-Befehle

Node-ID:

Der Wert der Node-ID bestimmt, welcher Teilnehmer oder ob alle Teilnehmer mit dem NMT-Befehl angesprochen werden sollen.

Command	Wert
alle Nodes	00d
Nodes nach Node-ID	01..127d
ungültig	128..255d

Tabelle 7.3: Node-ID Werte für NMT-Befehle

7.2 Heartbeat-Protokoll

Das Heartbeat-Protokoll ist defaultmäßig ausgeschaltet. An/Aus bzw. die Zeit in Millisekunden wird gesteuert über den Wert des Objekts. Ein Heartbeat kann entweder ausgesendet oder überwacht werden:

Producer Heartbeat (Drehgeber sendet seinen Heartbeat)

Der Producer Heartbeat kann An/Aus bzw. die Producer-Heartbeat-Time in Millisekunden eingestellt werden. Dies wird über den Wert des Objekts 1017h im Sub-Index 0 (00h = Aus, Time = 0..9999h) gesteuert.

Consumer Heartbeat (Drehgeber überwacht einen fremden Heartbeat)

Über das Objekt 1016h Sub-Index = 01h kann die Consumer Heartbeat Time eingestellt werden. Damit kann durch den Drehgeber ein anderes Gerät (ein Heartbeat Producer) überwacht werden. Der Ausfall eines Heartbeat Producers innerhalb der eingestellten Zeit führt zum Aussenden einer Emergency- Nachricht mit dem Wert 8130h (Life guard error oder heartbeat error). Über den 32-Bit-Wert wird die Zeit und die Node-ID des zu überwachenden Geräts eingestellt.

Bit 31-24	Bit 23 -16	Bit 15 – 0
Reserviert (00h)	Node-ID	Heartbeat Producer time

Tabelle 7.4: Fremden Heartbeat überwachen

Der Wert für die Zeit wird in Millisekunden angegeben. Wird für die Zeit der Wert 0 oder für die Node der Wert 0 oder größer 127 eingetragen, so wird die Consumer Heartbeat Time nicht genutzt bzw. deaktiviert.

Im Folgenden wird ein Beispiel für die Konfiguration eines Consumer Heartbeats zur Überwachung des Heartbeat Producers mit der Node-ID = 127 (7Fh) über eine Zeit von 10000 Millisekunden (= 2710h) gezeigt. Dazu wird ein SDO Schreibbefehl an den überwachenden Drehgeber mit der Node-ID = 01h geschickt.

601h	8	23h	16h	10h	01h	10h	27h	7Fh
CAN-ID	DLC	Command	Object L	Object H	Sub-Index	Time L	Time H	Producer Node-ID

Tabelle 7.5: Beispielkonfiguration eines Consumer Heartbeats

7.3 Emergency-Nachrichten

Die Emergency-Nachricht wird bei Fehlern auf dem Bus oder bei Problemen im Gerät ausgesendet. Sie hat einen speziellen Aufbau und übermittle eine Errorcodierung.

Über den Index 1014h ist die COB-ID für Emergency-Nachrichten zu definieren. Der Standardwert für den Identifier der Emergency-Nachricht ist 80h + eingestellte Node-ID (1 - 127). Es können BasicCAN Frames und ExtendedCAN Frames verwendet werden (Bit 29 = 1).

80h+ID	8	Error Code L	Error Code H	Error Reg.	Info1	Info2
CAN-ID	DLC	Byte0	Byte1	Byte2	Byte3	Byte4

Tabelle 7.6: Allgemeiner Aufbau einer Emergency-Nachricht

Error (H,L)	Code	Beschreibung
0000h		Kein Error / Entwarnung
4200h		Temperatur außerhalb der Toleranz
5000h		Hardware defekt (EEPROM)
8110h		CAN-overrun
8120h		CAN Error-Passive-Status
8130h		Heartbeat-Fehler / Lifeguarding-Fehler
8140h		Busoff-Recover

Tabelle 7.7: Emergency Error Code Liste

Error-register:

Inhalt von Objekt 1001h (Zuteilung Bit - Bedeutung, Standard = 00000000):

Bit:	7	6	5	4	3	2	1	0
Info:	co	co	co	Communication	Temperature	co	co	Generic error

Tabelle 7.8: Error-register

Infofeld Liste:

Das Infofeld ist abhängig vom Wert des Error Codes:

ErrorCode	Feld	Bit	Hex-Wert	Fehler
4200h	Infofeld 1 (Byte 3)	6	40h	Temp. Read Error
		5	20h	low limit Verstoß
		4	10h	high limit Verstoß

ErrorCode	Feld	Bit	Hex-Wert	Fehler
5000h	Infofeld 2 (Byte 4)	0	01h	Init EEPROM-Fehler
		3	08h	EEPROM Timeout

Error Code	Feld	Bit	Hex-Wert	Fehler
8120h + 8100h	Infofeld 1 (Byte 3) Low Nibble	0	1h	active, no Error
		1+2	6h	Bus-Warning
		0+1+2	7h	Bus-Passive
8120h + 8100h	Infofeld 1 (Byte 3) High Nibble	0	1h	Bit
		1	2h	Stuffing-Error
		0+1	3h	Form
		2	4h	CRC
		0+2	5h	Ack

Tabelle 7.9: Infofeld Liste

Das Low Nibble beschreibt den CAN-Status, das High Nibble beschreibt den CAN-Error näher.

Das Senden der Emergency-Nachricht kann durch Setzen des Bit 31 (MSB) im Objekt 1014h Sub-Index 00h unterbunden werden. Über den Index 1015h kann die Verzögerungszeit in Vielfachen von 100µs zwischen dem Senden von zwei Emergency-Nachrichten definiert werden.

7.4 Fehler Objekte

7.4.1 Manufacturer status register

Inhalt von Objekt 1002h (Zuteilung Bit - Bedeutung, Standard = 00h):

Bit:	7	6	5	4	3	2	1	0
Info:	co	co	co	co	co	EEPROM*	MT*	ST*(1)

Bit:	15	14	13	12	11	10	9	8
Info:	ST*(8)	ST*(7)	ST*(6)	ST*(5)	ST*(4)	ST*(3)	ST*(2)	ST*(1)

Bit:	23	22	21	20	19	18	17	16
Info:	ST*(15)	ST*(14)	ST*(13)	ST*(12)	ST*(11)	ST*(10)	ST*(9)	ST*(8)

Bit:	31	30	29	28	27	26	25	24
Info:	MT*(9)	MT*(8)	MT*(7)	MT*(6)	MT*(5)	MT*(4)	MT*(3)	MT*(2)

Tabelle 7.10: Manufacturer status register

*= Errortyp(Zahl) | Definition über Support erhältlich

7.4.2 Alarms

Inhalt von Objekt 6503h (Zuteilung Bit - Bedeutung, Standard = 0000000000000000):

Bit:	15..1	0
Info:	co	Position Error

Tabelle 7.11: Alarms - Objekt 6503h

7.4.3 Warnings

Inhalt von Objekt 6505h (Zuteilung Bit - Bedeutung, Standard = 0000000000000000):

Bit:	15	14	13	12	11..1	0
Info:	co	Temp. read failed	Undertemp.	Overtemp.	co	Frequency limit

Tabelle 7.12: Warnings – Objekt 6505h

7.5 Elektronisches Nockenschaltwerk (CAM)

Die Drehgeber der Firma MEGATRON bieten die Möglichkeit der Konfiguration eines sog. elektronischen Nockenschaltwerks über CANopen. Es wird ein CAM-Kanal mit bis zu 8 Nocken-Schaltpositionen unterstützt. Jeder Positionsparameter wird durch seinen Minimum-Schaltpunkt, seinen Maximum-Schaltpunkt und seine Schalt-Hysterese bestimmt.

7.5.1 CAM-state-register

Das CAM-state-register (Objekt 6300h) dient zur Darstellung der Nocken-Schalt-Zustände in Abhängigkeit zur Position der Drehgeberwelle. Dazu ist der Wert des Registers in binärer Schreibweise aufzuschlüsseln (siehe unten). Jedes Bit der Unsigned 8 aus Objekt 6300h zeigt den Status einer bestimmten Schaltposition. Folgendes Beispiel zeigt ein CAM-state-register mit dem Wert 89h:

Position	7(MSB)	6	5	4	3	2	1	0(LSB)
Type	CAM 8	CAM 7	CAM 6	CAM 5	CAM 4	CAM 3	CAM 2	CAM 1
Value	1	0	0	0	1	0	0	1
Logic	High	Low	Low	Low	High	Low	Low	High

Tabelle 7.13: CAM-state-register – Wert 89h

Wie oben zu sehen ist, definiert der Wert 89h, dass die Nockenschaltpositionen CAM 1, CAM 4 und CAM 8 High und die restlichen Nocken Low sind. Bei einem Weiterdrehen der Welle könnte es z. B. passieren, dass schließlich CAM 4 ebenfalls Low wird. Dann wäre der Wert des CAM-state-registers = 81h:

Position	7(MSB)	6	5	4	3	2	1	0(LSB)
Type	CAM 8	CAM 7	CAM 6	CAM 5	CAM 4	CAM 3	CAM 2	CAM 1
Value	1	0	0	0	0	0	0	1
Logic	High	Low	Low	Low	Low	Low	Low	High

Tabelle 7.14: CAM-state-register – Wert 81h

Über das unabhängige Schalten jedes einzelnen CAMs können so innerhalb eines Objektes und Sub-Indices 256 unterschiedliche Zustände erzeugt werden, die zur Steuerung von Maschinen verwendet werden können.

7.5.2 CAM-enable-register

Jede Nockenschaltposition des CAM-Kanals im Drehgeber muss zur Verwendung einzeln „angeschaltet“ werden. Das „Anschalten“ der einzelnen CAM funktioniert, indem der passende Wert in Objekt 6301h Sub-Index 01h geschrieben wird. Der richtige Wert ist zu finden, indem das Bit für jede Nockenschaltposition, die aktiv sein soll, in binärer Schreibweise auf 1 gesetzt wird. Sollen z. B. nur CAM 2, CAM 4 und CAM 7 aktiv sein, so ergibt sich nach binärer Schreibweise:

Position	7(MSB)	6	5	4	3	2	1	0(LSB)
Type	CAM 8	CAM 7	CAM 6	CAM 5	CAM 4	CAM 3	CAM 2	CAM 1
Value	0	1	0	0	1	0	1	0

Tabelle 7.15: CAM-enable-register – Wert 4Ah

Dies entspricht dem Wert 4Ah. Wird dieser in Objekt 6301h Sub-Index 01h geschrieben, so sind nur die Nockenschaltpositionen CAM 2, CAM 4 und CAM 7 aktiv und können sich in Abhängigkeit von Ihrer Konfiguration verändern.

7.5.3 CAM-polarity-register

Über das CAM-polarity-register in Objekt 6302h Sub-Index 01h können die Polaritäten jeder Nockenschaltposition im CAM-Kanal verändert werden. Standardmäßig ist die Polarität so gesetzt, dass alle Nockenschaltpositionen bei Positionswerten in ihren Limits auf High (= 1b) „springen“ (Default = 00000000b = 00h). Durch Verändern der einzelnen Bits können die einzelnen Polaritäten der Nockenpositionen verändert werden. So sind bei einem Wert von 13h (= 00010011b) CAM 1, CAM 2 und CAM 6 invertiert (Bit = 0b (Low), wenn Positionswert innerhalb der Limits).

Position	7(MSB)	6	5	4	3	2	1	0(LSB)
Type	CAM 8	CAM 7	CAM 6	CAM 5	CAM 4	CAM 3	CAM 2	CAM 1
Value	0	0	0	1	0	0	1	1
Logic	Default	Default	Default	Inverted	Default	Default	Inverted	Inverted

Tabelle 7.16: Beispiel CAM-polarity-register

7.5.4 CAM-Low-Limit

Über das CAM-Low-Limit wird der untere Umschaltpunkt einer Nockenschaltposition definiert. Jede einzelne Nockenschaltposition (CAM 1 .. CAM 8) hat ein eigenes CAM-Low-Limit-Objekt (siehe Objektverzeichnis 6310h .. 6317h).

	<ul style="list-style-type: none">• Das CAM-Low-Limit kann erst konfiguriert, d. h. in seinem Wert verändert werden, wenn das CAM-High-Limit des gleichen CAMs bereits gesetzt wurde. Es gilt: Der Wert des CAM-Low-Limit muss kleiner sein als der Wert des CAM-High-Limits
---	--

7.5.5 CAM-High-Limit

Über das CAM-High-Limit wird der obere Umschaltpunkt einer Nockenschaltposition definiert. Jede einzelne Nockenschaltposition (CAM 1 .. CAM 8) hat ein eigenes CAM-High-Limit-Objekt (siehe Objektverzeichnis 6320h .. 6327h).

7.5.6 CAM-Hysteresis

Über die CAM-Hysteresis wird die Breite der Hysterese der Umschaltpunkte definiert. Für jede einzelne Nockenschaltposition (CAM 1 .. CAM 8) kann eine eigene CAM-Hysteresis eingestellt werden (siehe Objektverzeichnis 6320h .. 6327h).

7.6 Geräte-Profil

Über den Index 1000h kann das Geräte-Profil abgefragt werden. Es wird nur Sub-Index 0 unterstützt. Defaultwerte sind:

- 0001 0196h - für Singleturn-Drehgeber
- 0002 0196h - für Multiturn-Drehgeber

7.7 SYNC

Über den Index 1005h wird der Identifier für die Synchronisations-Message (SYNC-Nachricht) eingestellt. Über die SYNC-Message kann das Senden eines PDOs ausgelöst werden. Es können BasicCAN Frames und ExtendedCAN Frames (Bit 29 = 1) verwendet werden. Das Gerät kann SYNC nur empfangen, nicht senden!

7.8 Drehgeber-Bezeichnung

Über den Index 1008h kann die Geräte-Bezeichnung abgefragt werden. Es wird nur Sub-Index 0 unterstützt. Der Wert des Objektes ist abhängig von der Firmware Variante.

Device Name für Singleturn Drehgeber:

HTB/FHB-ST-CA

Device Name für Multiturn Drehgeber:

HTB/FHB-PM-CA

7.9 Verhalten bei Auftreten von Fehlern

Wenn eine Betriebsstörung im CAN-Bus oder im Drehgeber selbst erkannt wird und das Gerät befindet sich im Operational Status, so wird das Gerät automatisch in den Pre-Operational-Status geschaltet. Das Verhalten bei CAN-Bus Fehlern wird über das Objekt 1029h Sub-Index 01h, das Verhalten bei Drehgeber Fehler über Sub-Index 02h geändert. Es sind folgende Werte zur Steuerung des Fehlerverhaltens für den Sub-Index 01h und 02h erlaubt:

Wert	Beschreibung
00h	Standardverhalten, in Pre-Operational wechseln
01h	Der aktuelle NMT-Status wird nicht verändert
02h	In den NMT-Status „Stopped“ wechseln

Tabelle 7.17: Werte zur Gebersteuerung bei Fehler

7.10 NMT-Startup-Verhalten

Über den Index 1F80h wird das NMT-Startup-Verhalten des Gerätes definiert. Es wird nur Sub-Index 0 unterstützt. Es sind folgende Werte zur Änderung des Startup-Verhaltens erlaubt:

Wert	Beschreibung
00h	Standardverhalten, in Pre-Operational wechseln
02h	Sende NMT-Befehl „Start All Nodes“
08h	In den NMT-Status „Operational“ wechseln

Tabelle 7.18: Werte zur Geber-Startup-Steuerung

So kann definiert werden, dass der Drehgeber selbstständig nach Stromanschlüssen in den Operational-Status wechselt oder, falls nötig, als „CANopen-NMT-Master“ eine „Start-All-Nodes“ Nachricht aussendet. Die Konfiguration des Startup-Verhaltens muss gespeichert werden.

7.11 Bus-Off Auto-Reset

Über den Index 2103h kann das Verhalten bei Bus-Off geändert werden. Der Wert definiert die Zeit in Sekunden, die verstreicht, bevor das Gerät automatisch von CAN Bus-Off in CAN-Error-Active wechselt. Der Wert 0 ist die Default-Einstellung und schaltet dieses Verhalten ab, da eine andere Konfiguration evtl. kritisch sein kann.

7.12 Customer Data

Über den Index 2120h kann der Endkunde bis zu 8 Worte im EEPROM des Gerätes speichern. Es wird nur Sub-Index 0 bis 8 unterstützt. Ein Schreibzugriff auf die Sub-Indices 1 bis 8 bewirkt ein automatisches Speichern des Wertes im EEPROM. Ein Zugriff auf Objekt 1010h (siehe 8.12 „Einstellungen in das EEPROM speichern“) ist nicht erforderlich.

7.13 Temperatur

Über den Index 2500h kann die aktuelle Gerätetemperatur ausgelesen, sowie Temperaturgrenzen gesetzt werden. Es werden die Sub-Indices 0 bis 5 unterstützt. Die aktuelle Temperatur wird alle 60 Sekunden gemessen. Alle Temperaturwerte werden in °C angegeben. Ein Überschreiten der Grenzwerte löst einen einmaligen Temperaturalarm aus (EMCY-Nachricht, Warning-Objekt 6505h). Der Alarmstatus wird im Error-Register Objekt 1001h Sub-Index 0h mitgeführt. Bei einer Temperatur, die außerhalb der Limits liegt, wird das Error-Register den binären Wert 1000b (=08h) annehmen.

7.14 Konfigurationsabgleich

Über den Index 1020h kann der Zeitpunkt der letzten Konfiguration des Gerätes ausgelesen bzw. gespeichert werden. Bei Änderung der Konfiguration des Gerätes löschen sich der Inhalt des Sub-Indices 1 und 2 und es muss der neue Zeitpunkt der Konfiguration eingespeichert werden.



- Jede Veränderung der Parameter von Objekten muss, sofern nicht anders beschrieben, durch den Befehl „Store All Parameters“ (siehe 8.12 „Einstellungen in das EEPROM speichern“) gespeichert werden.

8 Inbetriebnahme

8.1 Mechanischer und elektrischer Anschluss

	<ul style="list-style-type: none">• Für den mechanischen und elektrischen Anschluss beachten Sie bitte unbedingt die dem Drehgeber beigelegten Montageanleitungen und Informationen.
---	--

Vollwellen-Drehgeber:

	<ul style="list-style-type: none">• Drehgeberwelle und Antriebswelle immer über eine dafür geeignete Kupplung verbinden. Die Kupplung sorgt für den Ausgleich des Spiels beider Wellen in radialer und axialer Richtung. Drehgeber- und Antriebswelle dürfen sich niemals berühren. Die maximalen Achslasten des Antriebs und des Drehgebers sind zu beachten. Passendes Zubehör finden Sie auf https://www.megatron.de• Der Drehgeber kann über die Bohrungen im Flansch auf der Wellenseite einfach an einer geeigneten Platte verschraubt werden.• Eine weitere Möglichkeit der Befestigung des Drehgebers bietet der Einsatz von MEGATRON Synchroklemmen SFN1.
--	--

Hohlwellen-Drehgeber:

	<ul style="list-style-type: none">• Drehgeber vollständig auf die Antriebswelle stecken. Mit den Madenschrauben in der Drehgeber-Hohlwelle durch Schrauben auf der Antriebswelle befestigen.• Der Drehgeber verfügt über ein Federblech, welches das im Flansch entstehende Drehmoment aufnimmt. Es wird mit zwei Schrauben an der Maschine befestigt. Das Federblech ist „federnd“ ausgeführt, um Vibrationen und Spiel auf der Antriebswelle ausgleichen zu können und die Lager des Drehgebers dabei nicht zu überlasten.
---	---

Der Drehgeber ist entweder über einen Sensorstecker oder ein Kabel mit dem CAN-Bus zu verbinden. Um den Drehgeber einfach in den Bus zu integrieren, legen wir Ihnen nahe, sog. Busweichen (T-Stückadapter) zu verwenden. Sollte der Drehgeber am Ende des Busses eingesetzt werden sind auch 120 Ohm Abschlusswiderstände erhältlich. Zubehörteile finden Sie unter <https://www.megatron.de>

Belegung (nach CiA 303):

(HTB36E-Belegung kann abweichen (z.B. FHB58))



Definition	Kabelfarbe (Kabelgeber)	Steckerpin (Steckergeber)
Versorgung U_B (10-30V)	braun	2
Ground (GND)	weiß	3
CAN _{High}	grün	4
CAN _{Low}	gelb	5
CAN _{GND}	grau	1

Tabelle 8.1: Stecker-Kabel Belegung

8.2 Einstellungen per LSS vornehmen

8.2.1 Allgemeine Einstellungen

Das Layer Setting Services Protokoll (LSS) ist in dem Draft Standard Proposal 305 beschrieben. Das LSS erlaubt die Konfiguration des Drehgebers, auch wenn dessen Node-ID nicht eindeutig im CAN-Bus vergeben ist. Dies kann passieren, wenn der Drehgeber zum ersten Mal integriert und noch nicht konfiguriert wurde. Die Drehgeber HTB und FHB unterstützen folgende LSS-Dienste:

- Switch Mode Global
- Switch Mode Selective
- Configure Baudrate Service
- Configure Node-ID Service
- Store Configuration Service
- Identification And Inquire Services (Node-ID, Vendor-ID, Product Code, Revision Number, Serial Number)

Eine LSS-Nachricht ist wie folgt aufgebaut:

CAN-ID	DLC	Command	Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6
--------	-----	---------	-------	-------	-------	-------	-------	-------	-------

Tabelle 8.2: LSS-Nachricht

Dabei gilt für die CAN-ID:

- LSS-Master -> LSS-Slave: 2021(7E5h)
- LSS-Slave) -> LSS-Master: 2020(7E4h)

Das LSS kann nur genutzt werden, wenn sich der Drehgeber im Stopped-Status bzw. Pre-Operational-Status befindet. Um den Drehgeber über LSS konfigurierbar zu machen, muss er in den LSS Konfig-Modus versetzt werden. Dafür gibt es verschiedene Möglichkeiten:

- Switch Mode Global
- Switch Mode Selective

8.2.2 LSS-Konfigurationsmodus mit „Switch Mode Global“

Verbinden Sie den LSS-Master mit dem Drehgeber. Schalten Sie zuerst den Drehgeber ein, dann den Master. Die eingestellte Baudrate des LSS-Masters wird vom Drehgeber erkannt. Setzen Sie den Drehgeber per NMT-Befehl in den „Stopped“ Modus. Schicken Sie diese Nachricht:

7E5h	04h	01h	00h	00h	00h	00h	00h	00h
------	-----	-----	-----	-----	-----	-----	-----	-----

Tabelle 8.3: Befehl um Drehgeber in „Stopped“-Modus zu setzen

Der Drehgeber ist nun im Konfigurationsmodus. Sie können jetzt die Baudrate und die Node-ID des Drehgebers über LSS einstellen (siehe Kapitel 8.2.5 und 8.2.6).

8.2.3 LSS-Konfigurationsmodus mit „Switch Mode Selective“

Verbinden Sie den LSS-Master mit dem Drehgeber. Schalten Sie zuerst den Drehgeber ein, dann den Master. Die eingestellte Baudrate des LSS-Masters wird vom Drehgeber erkannt. Setzen Sie den Drehgeber per NMT-Befehl in den „Stopped“ Modus. Im Switch Mode Selektiv kann ein bestimmtes Gerät über das Aussenden einer Folge von vier verschiedenen Identifikations-Nachrichten ausgewählt werden:

LSS-Command	Information	Beschreibung
40h	Vendor-ID	0100 021Fh
41h	ProductCode	5744 4741h
42h	RevisionNumber	Revision des Drehgebers
43h	SerialNumber	Seriennummer des Drehgebers

Tabelle 8.4: LSS-Selective-Identification-Commands

Weitere Informationen zur Revision Number und der Serial Number finden Sie unter Punkt 1 Einleitung.

Nachdem die letzte der vier Identifikations-Nachrichten gesendet wurde, antwortet der angesprochene Drehgeber mit dem Code:

LSS-Command	Information	Beschreibung
44h	Mode	Mode = 1 -> Konfig-Modus Mode = 0 -> Operation-Modus

Tabelle 8.5: Antwort des Drehgebers auf LSS-Selective-Identification-Commands

Der Drehgeber ist nun im Konfigurationsmodus. Sie können jetzt die Baudrate und die Node-ID des Drehgebers über LSS einstellen (siehe Kapitel 8.2.5 und 8.2.6).

	<ul style="list-style-type: none"> Sobald der Drehgeber durch LSS in den LSS-Konfig-Modus geschaltet wurde (selective oder global) sind Baudrate sowie Node-ID des Drehgebers über LSS einstellbar. Nach der Konfiguration müssen die Einstellungen gespeichert und der Konfigurationsmodus wieder deaktiviert werden (siehe nächste Seite „Beenden des LSS-Konfigurationsmodus:“).
---	--

8.2.4 Beenden des LSS-Konfigurationsmodus

Nach Beendigung der Konfiguration muss der Drehgeber die veränderten Parameter speichern und wieder in den Pre-Operational-Status versetzt werden. Dies wird mit folgenden Nachrichten durchgeführt:

Schritt 1 – speichern:

7E5h	17h	00h						
------	-----	-----	-----	-----	-----	-----	-----	-----

Tabelle 8.6: LSS-Konfigurationsmodus beenden – Schritt 1: speichern

Schritt 2 – Konfigurations-Modus verlassen:

7E5h	04h	00h						
------	-----	-----	-----	-----	-----	-----	-----	-----

Tabelle 8.7: Konfigurations-Modus verlassen

Danach muss ein neuer Boot-up ausgelöst werden (z. B. durch Spannungsreset).

8.2.5 Baudrate einstellen

Um die Baudrate des in der Anwendung verwendeten CAN-Bus einzustellen, schicken Sie den Befehl:

7E5h	13h	00h	Baudrate	00h	00h	00h	00h	00h
CAN-ID	Command	Sub-Index	Baudrate	Byte2	Byte3	Byte4	Byte5	Byte6

Tabelle 8.8: Baudrate einstellen

Für die Baudraten sind folgende Werte vorgesehen:

Wert	Baudrate
0	1 Mbit/s
1	800 kBit/s
2	500 kBit/s
3	250 kBit/s
4	125 kBit/s
5	100 kBit/s
6	50 kBit/s
7	20 kBit/s
8	10 kBit/s
9	Auto

Tabelle 8.9: Baudraten-Codierung

Prüfen Sie die Antwort des LSS-Slaves auf obigen Befehl:

7E4h	13h	00h	00h	00h	00h	00h	00h	00h
CAN-ID	Command	Error Code	Specific Error	Byte2	Byte3	Byte4	Byte5	Byte6

Tabelle 8.10: Antwort des LSS-Slaves

Error Code:

- 00h = OK
- 01h = Außerhalb des Bereichs

Specific Error:

- 00h = OK
- FFh = Applikations-spezifischer Fehler

Es ist möglich, dass nach Beendigung des LSS-Konfigurationsmodus Ihre Verbindung mit dem Drehgeber abbricht, da Konfigurationstool und Drehgeber auf unterschiedlichen Baudraten arbeiten. Für weitere Konfigurationen führen Sie die Baudrateneinstellung auf Ihrem Konfigurationstool nach!

	<ul style="list-style-type: none"> • Stellen Sie die Baudrate nur um, wenn Sie die Baudrate Ihrer Anwendung genau kennen und auch Ihr Konfigurationstool diese unterstützt. Notieren Sie sich die eingestellte Baudrate sicherheitshalber in Ihren Unterlagen oder auf dem Drehgeber-Etikett.
---	--

8.2.6 Node-ID des Drehgebers einstellen

Zum Einstellen der Node-ID des Drehgebers schicken Sie den Befehl:

7E5h	11h	Node-ID	00h	00h	00h	00h	00h	00h
CAN-ID	Command	Node-ID	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6

Tabelle 8.11: Einstellen der Node-ID

Der Wertebereich für die Node-ID des Drehgebers liegt zwischen 00h und 7Fh.

	<ul style="list-style-type: none"> • Vergessen Sie nach Beenden der Konfiguration nicht, den LSS-Konfigurationsmodus zu verlassen (siehe oben)!
---	--

8.3 Einstellungen per SDO vornehmen

	<ul style="list-style-type: none"> • Alle Konfigurationen, soweit hier nicht anders beschrieben, müssen mit einem Speicherbefehl (8.12 Einstellungen in das EEPROM speichern) vor einem Reset gesichert werden.
--	--

8.3.1 Objekte einrichten und lesen

Mittels eines SDOs können Daten eingerichtet, abgespeichert oder gelesen werden.

Lesen eines Objektes:

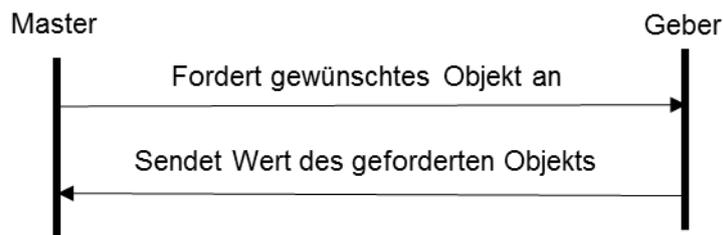


Abbildung 8.1: Objekt Lesen

Die Nachrichten strukturieren sich dabei wie folgt:

Masteranfrage:

600h+ID	8	40h	04h	60h	00h	00h	00h	00h	00h
CAN-ID	DLC	Command	Object L	Object H	Sub-Index	Byte0	Byte1	Byte2	Byte3

Tabelle 8.12: Beispiel SDO Masteranfrage – Objekt lesen

Drehgeber Antwort mit 4 Byte Daten (d1d2d3d4):

580h+ID	8	43h	04h	60h	00h	d4	d3	d2	d1
CAN-ID	DLC	Command	Object L	Object H	Sub-Index	Byte0	Byte1	Byte2	Byte3

Tabelle 8.13: Beispiel SDO Antwort – Objekt lesen

Die Kommandowerte folgen dabei einer grundlegenden Logik. Tabelle 8.14: Kommando-Definitionen zeigt einen allgemeinen Überblick.

Command	Typ	Beschreibung
22h	Schreibbefehl	Parameter an Drehgeber
23h	Schreibbefehl	4 Byte Parameter an Drehgeber
27h	Schreibbefehl	3 Byte Parameter an Drehgeber
2Bh	Schreibbefehl	2 Byte Parameter an Drehgeber
2Fh	Schreibbefehl	1 Byte Parameter an Drehgeber
60h	Bestätigung	Parameter empfangen
40h	Lesebefehl	Parameter von Drehgeber fordern
42h	Antwort	Parameter an Master
43h	Antwort	4 Byte Parameter an Master
47h	Antwort	3 Byte Parameter an Master
4Bh	Antwort	2 Byte Parameter an Master
4Fh	Antwort	1 Byte Parameter an Master
80h	Abbruchcode	Übertragungsfehler / Fehlercode
41h	Antwort	SDO segmented Transfer (s. CiA 301)

Tabelle 8.14: Kommando-Definitionen

Schreiben eines Objektes:

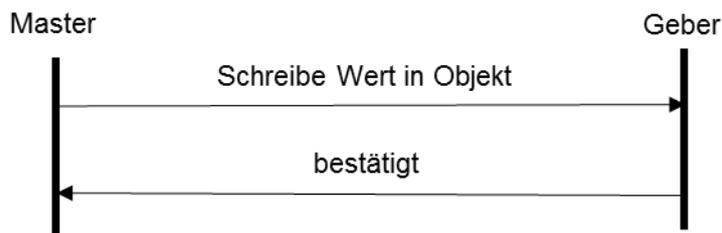


Abbildung 8.2: Objekt schreiben

Die Nachrichten strukturieren sich dabei wie in folgendem Beispiel:

Master sendet 1 Byte Daten (d1) zum Speichern an den Drehgeber:

600h+ID	8	2Fh	00h	21h	00h	d1	00h	00h	00h
CAN-ID	DLC	Command	Object L	Object H	Sub-Index	Byte0	Byte1	Byte2	Byte3

Tabelle 8.15: Beispiel SDO Masteranfrage – Objekt schreiben

Drehgeber Antwort mit ohne Datenbytes:

580h+ID	8	2Fh	00h	21h	00h	00h	00h	00h	00h
CAN-ID	DLC	Command	Object L	Object H	Sub-Index	Byte0	Byte1	Byte2	Byte3

Tabelle 8.16: Beispiel SDO Antwort – Objekt schreiben

Auch hier zeigt Tabelle 8.14 den Überblick über die verwendeten Kommandos.

8.3.2 Große Objekte (>4 Byte) einrichten und lesen

Ein SDO kann maximal 4 Byte Nutzdaten übertragen. Soll eine größere Datenmenge übertragen werden, so kann dies über den segmented SDO-Transfer oder einen Block-Transfer mit bis zu 127 Segmenten á 4 Bytes gemacht werden. Als Anwendungsbeispiel soll hier das Auslesen von dem Objekt 6008h (High Precision Position Value) und danach das Schreiben auf Objekt 6009h (High Precision Preset) über segmented SDO-Transfer gezeigt werden.

Einleiten eines segmented SDO-Transfer Lesezugriffs:

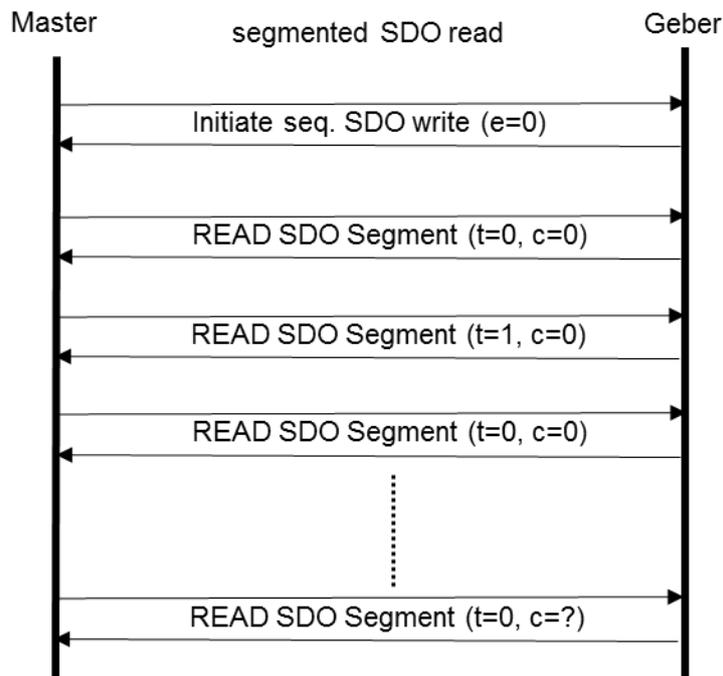


Abbildung 8.3: Allgemeiner segmented SDO-Transfer Lesezugriff

Beispiel: 8 Byte "High Precision Position Value" (Objekt 6008h) lesen:

600h+ID	8	40h 01000000b ccs=2,e=0, s=0	08h	06h	00h	00h	00h	00h	00h
CAN-ID	DLC	Command	Object L	Object H	Sub-Index	Byte0	Byte1	Byte2	Byte3

Tabelle 8.17: SDO Lesezugriff auf Objekt 6008h

Lesen von einem Objekt über segmented SDO-Transfer:

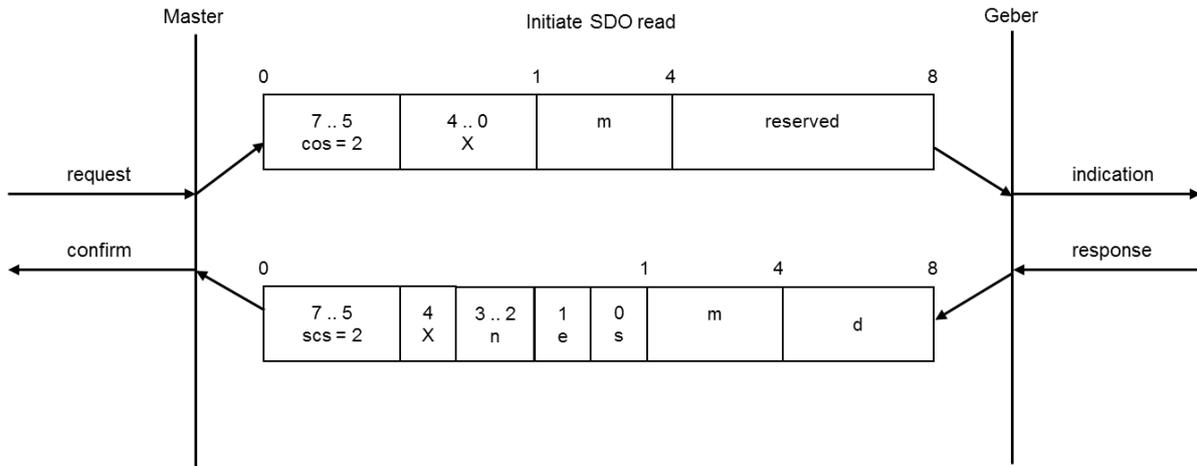


Abbildung 8.4: Initiate SDO read

ccs	client command specifier	2 = initiate read (upload) request
scs	server command specifier	2 = initiate read (upload) response
n	Indicates that bytes [8-n,7] don't contain segmented data	Only valid if e=1 and s=1, otherwise 0.
e	Transfer type	0 = segmented transfer 1 = expedited transfer
s	size indicator	0 = data set size not indicated 1 = data set size indicated
m	multiplexor	index/sub index of data to be transferred
d	data	e=0, s=0 -> d is reserved. e=0, s=1 -> d = number of bytes to be read. e=1, s=1 -> d = data of length 4-n to be read. e=1, s=0 -> d = unspecified number of bytes to be read.
X	not used	always 0
	reserved	reserved for further use, always 0

Tabelle 8.18: Erläuterung der in Abbildung 8.4 verwendeten Abkürzungen

Der Drehgeber bestätigt den segmented SDO-Transfer über 8 Datenbyte:

580h+ID	8	41h 01000001b scs=2,e=0, s=1	08h	06h	00h	08h	00h	00h	00h
CAN-ID	DLC	Command	Object L	Object H	Sub-Index	Byte0	Byte1	Byte2	Byte3

Tabelle 8.19: Bestätigung des SDO Lesezugriffs auf Objekt 6008h

SDO-Segment auslesen:



Abbildung 8.5: SDO-Segment auslesen

ccs	client command specifier	3 = read (upload) segment request
scs	server command specifier	0 = read (upload) segment response
t	toggle bit	Must alternate for each subsequent segment with t=0 for the first segment. Equal for each pair of request and response.
c	more segments indicator	0 = more segments to be read (uploaded). 1 = no more segments to be read (uploaded).
seg-data	Segment data	At most 7 byte of segment data.
n	Number of bytes that don't contain segment data	Bytes [8-n;7] don't contain segment data n = 0 if no segment size indicated
x	not used	always 0
	reserved	reserved for further use, always 0

Tabelle 8.20: Erläuterung der in Abbildung 8.5 verwendeten Abkürzungen

Das erste Segment lesen:

600h+ID	8	60h 01100000b ccs=3, t=0	00h	00h	00h	08h	00h	00h	00h
CAN-ID	DLC	Command	Object L	Object H	Sub-Index	Byte0	Byte1	Byte2	Byte3

Tabelle 8.21: Lesen des ersten Segmentes

Der Drehgeber antwortet mit dem ersten Datensegment:

580h+ID	8	00h 00000000b scs=0, t=0, n=0, c=0	data	data	data	data	data	data	data
CAN-ID	DLC	Command	Object L	Object H	Sub-Index	Byte0	Byte1	Byte2	Byte3

Tabelle 8.22: Antwort mit erstem Datensegment

Dann wird das nächste Segment angefragt:

600h+ID	8	70h 01110000b ccs=3, t=1	00h	00h	00h	08h	00h	00h	00h
CAN-ID	DLC	Command	Object L	Object H	Sub-Index	Byte0	Byte1	Byte2	Byte3

Tabelle 8.23: Lesen des ersten Segmentes

Der Drehgeber antwortet mit dem nächsten Datenssegment:

580h+ID	8	1Dh 00011101b scs=0, t=1, n=6, c=1	data	x	x	x	x	x	x
CAN-ID	DLC	Command	Object L	Object H	Sub-Index	Byte0	Byte1	Byte2	Byte3

Tabelle 8.24: Antwort mit erstem Datenssegment

In diesem Segment teilt der Drehgeber mit, dass es das letzte zu übertragende Segment ist und dass nur das erste Datenbyte noch Nutzdaten enthält. Die 7 Datenbytes aus dem ersten Segment und das eine Nutzdatenbyte aus Segment zwei beschreiben zusammengenommen den Wert des Objekts 6008h (High Precision Position Value).

Segmented-SDO Transfer Schreibzugriff:



Abbildung 8.6: Segmented-SDO Transfer Schreibzugriff

Folgendes Beispiel zeigt, wie 8 Nutzdaten über einen Segmented SDO-Transfer auf Objekt 6009h (High Precision Preset Value) geschrieben werden können.

SDO Schreibzugriff Anfrage für 8 Nutzdatenbytes auf Objekt 6009h:

600h+ID	8	21h 00100001b ccs=1,e=0, s=1	09h	06h	00h	08h	00h	00h	00h
CAN-ID	DLC	Command	Object L	Object H	Sub-Index	Byte0	Byte1	Byte2	Byte3

Tabelle 8.25: SDO Schreibzugriff auf Objekt 6009h

Der Drehgeber bestätigt den segmented SDO-Transfer über 8 Datenbyte und erwartet das erste Segment:

580h+ID	8	60h 01100000b scs=3	09h	06h	00h	00h	00h	00h	00h
CAN-ID	DLC	Command	Object L	Object H	Sub-Index	Byte0	Byte1	Byte2	Byte3

Tabelle 8.26: Bestätigung des SDO Schreibzugriffs auf Objekt 6009h

Einleiten eines segmented SDO-Transfer Schreibzugriffs:

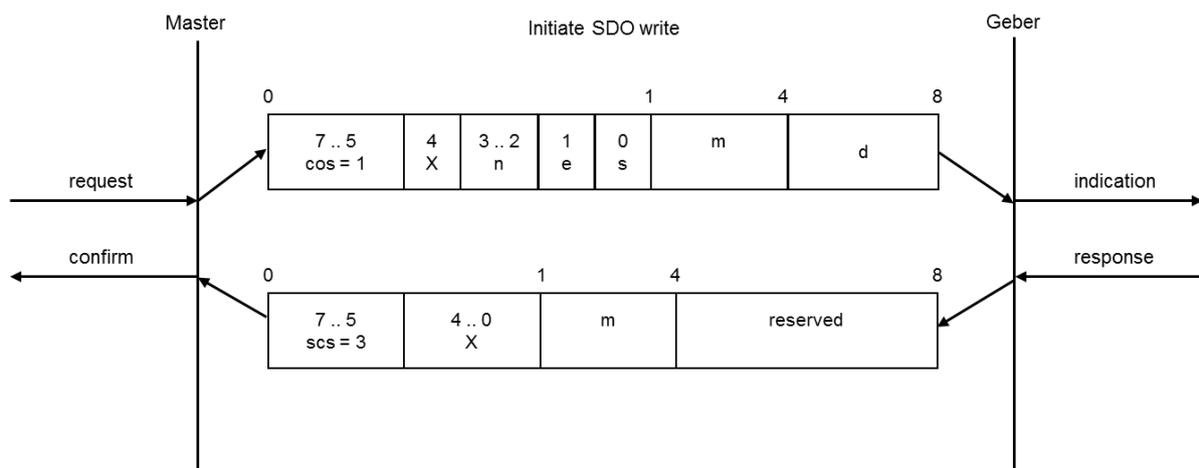


Abbildung 8.7: Initiate SDO write

ccs	client command specifier	1 = initiate write (download) request
scs	server command specifier	3 = initiate write (download) response
n	Indicates that bytes [8-n,7] don't contain segmented data	Only valid if e=1 and s=1, otherwise 0.
e	Transfer type	0 = segmented transfer 1 = expedited transfer
s	size indicator	0 = data set size not indicated 1 = data set size indicated
m	multiplexor	index/sub index of data to be transferred
d	data	e=0, s=0 -> d is reserved. e=0, s=1 -> d = number of bytes to be written. e=1, s=1 -> d = data of length 4-n to be written. e=1, s=0 -> d = unspecified number of bytes to be written.
X	not used	always 0
	reserved	reserved for further use, always 0

Tabelle 8.27: Erläuterung der in Abbildung 8.7 verwendeten Abkürzungen

Das erste Segment wird gesendet:

600h+ID	8	00h 00000000b ccs=0, t=0, n=0, c=0	data	data	data	data	data	data	data
CAN-ID	DLC	Command	Object L	Object H	Sub-Index	Byte0	Byte1	Byte2	Byte3

Tabelle 8.28: Erstes Segment senden

Der Drehgeber bestätigt und erwartet das nächste Segment:

580h+ID	8	20h 00100000b scs=1, t=0	data	data	data	data	data	data	data
CAN-ID	DLC	Command	Object L	Object H	Sub-Index	Byte0	Byte1	Byte2	Byte3

Tabelle 8.29: Bestätigung des Drehgebers

Segmented SDO-Transfer Schreibzugriff:



Abbildung 8.8: write SDO segment

ccs	client command specifier	0 = write (download) segment request
scs	server command specifier	0 = write (download) segment response
t	toggle bit	Must alternate for each subsequent segment with t=0 for the first segment. Equal for each pair of request and response.
c	more segments indicator	0 = more segments to be written (downloaded). 1 = no more segments to be written (downloaded).
seg-data	Segment data	At most 7 byte of segment data.
n	Number of bytes that don't contain segment data	Bytes [8-n;7] don't contain segment n = 0 if no segment size indicated
x	not used	always 0
	reserved	reserved for further use, always 0

Tabelle 8.30: Erläuterung der in Abbildung 8.8 verwendeten Abkürzungen

Das nächste Segment wird gesendet:

600h+ID	8	1Dh 00011101b ccs=0, t=1, n=6, c=1	data	x	x	x	x	x	x
CAN-ID	DLC	Command	Object L	Object H	Sub-Index	Byte0	Byte1	Byte2	Byte3

Tabelle 8.31: Nächstes Segment senden

In diesem Segment wird mitgeteilt, dass dies das letzte zu übertragende Segment war und dass nur das erste Datenbyte Nutzdaten enthält.

Der Drehgeber bestätigt dies mit:

580h+ID	8	20h 00100000b scs=1, t=1	data	x	x	x	x	x	x
CAN-ID	DLC	Command	Object L	Object H	Sub-Index	Byte0	Byte1	Byte2	Byte3

Tabelle 8.32: Bestätigung des Drehgebers

Die 7 Datenbytes aus dem ersten Segment und das eine Nutzdatenbyte aus Segment zwei beschreiben zusammengenommen den Wert des Objekts 6009h (High Precision Position Preset Value).

8.3.3 Baudrate einstellen

Die Baudrate der Drehgeber HTB und FHB der Firma Megatron sind mit einer automatischen Baudratenerkennung und Einstellung ausgestattet. Eine manuelle Einstellung der Baudrate über SDO ist ebenfalls möglich.

Die SDO Konfiguration ist nur im Pre-Operational-Status möglich. Zum Einstellen der Baudrate muss das Objekt 2100h im Sub-Index 00h angepasst werden. Dazu ist ein einfacher SDO-Schreibbefehl mit der Baudrate als Wert nötig.

600h+ID	8	2Fh	00h	21h	00h	Baud	00h	00h	00h
CAN-ID	DLC	Command	Object L	Object H	Sub-Index	Byte0	Byte1	Byte2	Byte3

Tabelle 8.33: Nächstes Segment senden

Für die Baudrate (Baud) sind folgende Werte vorgesehen:

Wert	Baudrate
0	1 Mbit/s
1	800 kBit/s
2	500 kBit/s
3	250 kBit/s
4	125 kBit/s
5	100 kBit/s
6	50 kBit/s
7	20 kBit/s
8	10 kBit/s
9	Auto

Tabelle 8.34: Baudraten-Codierung

	<ul style="list-style-type: none"> • Eine Änderung der Baudrate über SDO wird erst nach dem manuellen Reset des Drehgebers (Spannungs-Reset oder NMT-Reset, siehe 7.2.7 Drehgeber Basisbefehle (NMT)) wirksam. Das Schreiben in Objekt 2100h ist NICHT passwortgeschützt und bewirkt ein automatisches Speichern im EEPROM. Somit ist kein manuelles Speichern durch ein „Save Parameters“-SDO nötig.
--	--

8.3.4 Node-ID des Drehgebers einstellen

Das Setzen der Node-ID des Drehgebers ist mit SDO möglich. Zum Setzen der Node-ID muss das Objekt 2101h im Sub-Index 00h angepasst werden. (Nur im Pre-Operational-Status möglich!) Dazu ist ein einfacher SDO-Schreibbefehl mit der Soll-Node-ID als Datenwert nötig.

600h+ID	8	2Fh	01h	21h	00h	Node	00h	00h	00h
CAN-ID	DLC	Command	Object L	Object H	Sub-Index	Byte0	Byte1	Byte2	Byte3

Tabella 8.35: Node-ID einstellen

Ein Beispiel für eine Node-ID des Drehgebers kann sein:

Drehgeber Nummer (d)	Node-ID des Drehgebers (h)
1	01h
2	02h
127	7Fh

Tabella 8.36: Beispielwerte der Node-ID

	<ul style="list-style-type: none"> • Eine Änderung der Node-ID des Drehgebers über SDO wird erst nach dem manuellen Reset des Drehgebers (Spannungs-Reset oder NMT-Reset, siehe 8.3.5 NMT-Master Basisbefehle) wirksam. Das Schreiben in Objekt 2101h ist NICHT passwortgeschützt und bewirkt ein automatisches Speichern im EEPROM. Somit ist kein manuelles Speichern durch ein „Save Parameters“-SDO nötig.
--	---

	<ul style="list-style-type: none"> • Das Ändern der Node-ID bewirkt ein automatisches Anpassen der PDO und EMCY COB-IDs. Nach dem ersten manuellen Speichern werden diese auf ihrem derzeitigen Wert eingefroren und nicht mehr automatisch angepasst. Durch das Ausführen des „Restore Defaults“ Befehls wird die automatische Anpassung wieder aktiviert.
--	--

8.3.5 NMT-Master Basisbefehle

Im Folgenden werden verschiedene Grundbefehle, die dem NMT-Master zur Verfügung stehen, beschrieben. Grundlegende Informationen sind unter Abschnitt 5.4 zu finden.

Zum **Starten** des Drehgebers wird der „Start Remote Node“ Befehl genutzt:

0	02h	01h	0 – 127
CAN-ID	DLC	Command Byte	Node-ID

Tabelle 8.37: NMT Befehl - Start Remote Node

Zum **Stoppen** des Drehgebers wird der „Stop Remote Node“ Befehl genutzt:

0	02h	02h	0 – 127
CAN-ID	DLC	Command Byte	Node-ID

Tabelle 8.38: NMT Befehl - Stop Remote Node

Zum **Wechsel in den Pre-Operational-Status** wird der „Enter Pre-Operational-Status“ Befehl genutzt:

0	02h	80h	0 – 127
CAN-ID	DLC	Command Byte	Node-ID

Tabelle 8.39: NMT Befehl - Enter Pre-Operational-Status

Ein **Reset der Kommunikation** und damit der Wechsel in den Pre-Operational-Status nach einer Neuinitialisierung wird durch den Befehl „Reset Node Communication“ erreicht:

0	02h	82h	0 – 127
CAN-ID	DLC	Command Byte	Node-ID

Tabelle 8.40: NMT Befehl - Reset Node Communication

Der **Reset des gesamten Drehgebers** wird durch den Befehl „Reset Remote Node“ erreicht. Hier springt der Drehgeber nach einer Neu-Initialisierung in den Pre-Operational Status:

0	02h	81h	0 – 127
CAN-ID	DLC	Command Byte	Node-ID

Tabelle 8.41: NMT Befehl - Reset Remote Node

8.4 Heartbeateinstellungen

Der Producer-Heartbeat kann wie im folgenden Beispiel gestartet und konfiguriert werden (setzen des Heartbeats auf 5000 Millisekunden mit 5000d = 1388h über SDO auf Objekt 1017h):

600h+ID	8	2Bh	17h	10h	00h	88h	13h	00h	00h
CAN-ID	DLC	Command	Object L	Object H	Sub-Index	Byte0	Byte1	Byte2	Byte3

Tabelle 8.42: Beispiel Heartbeateinstellung

Ein ausgesendeter Heartbeat setzt sich wie folgt zusammen:

700h+ID	1	D	NMT-Status
CAN-ID	DLC	Data/Remote	Byte 0

Tabelle 8.43: Beispiel Heartbeateinstellung

NMT-Status:

NMT-Status	Code
Boot-up	00h
Stopped	04h
Pre-Operational	7Fh
Operational	05h

Tabelle 8.44: Heartbeat NMT-Status-Codierung

8.5 PDO-Konfiguration

8.5.1 PDO-Parametrierung

Es können vier PDOs parametrierung werden. Jedes der PDOs ist in einer bestimmten Weise vorkonfiguriert („gemappt“):

Objekt	PDO	Vor-Konfiguration (Scheduling)	Prozessdaten
1800h	PDO1	asynchron / auf Messwertänderung	Position-value
1801h	PDO2	synchron / jede SYNC-Nachricht beantworten	Position-value
1802h	PDO3	synchron / jede SYNC-Nachricht beantworten	HighPrecision-value
1803h	PDO4	nicht aktiviert	

Tabelle 8.45: PDO-Vorkonfiguration

Es ist möglich ein PDO durch die Parametrierung in fünf Zustände zu setzen:

Sub-Index 2	Sub-Index 5	Beschreibung
01h-F0h	n.n.	PDO synchron / auf SYNC-Nachricht
FFh	0000h	PDO deaktiviert
FEh	0001h-FFFFh	PDO asynchron / auf internem Timer UND Messwertänderung
FEh	0000h	PDO asynchron / auf Messwertänderung
FFh	0001h-FFFFh	PDO asynchron / auf internem Timer

Tabelle 8.46: Mögliche PDO-Zustände

- Die Zustandsparametrierung kann nur im Pre-Operational-Status durchgeführt werden und muss gespeichert werden!

Um ein PDO vollständig zu deaktivieren, muss der Wert der PDO-COB-ID geändert werden:

PDO	Objekt	COB-ID für aktives PDO	COB-ID für deaktiviertes PDO
1	1800h	4000 0181h	C000 0181h
2	1801h	4000 0281h	C000 0281h
3	1802h	4000 0381h	C000 0381h
4	1803h	4000 0481h	C000 0481h

Tabelle 8.47: PDO-Deaktivierung

Als Beispiel soll PDO 1 deaktiviert werden. Dazu wird folgender SDO Schreibbefehl gesendet:

600h+ID	8	23h	00h	18h	01h	81h	01h	00h	C0h
CAN-ID	DLC	Command	Object L	Object H	Sub-Index	Byte0	Byte1	Byte2	Byte3

Tabelle 8.48: Beispiel PDO1 Deaktivierung

- Eine weitergehende Parametrierung der PDO-COB-IDs (Objekte: 1800h-01h; 1801h-01h; 1802h-01h; 1803h-01h) ist auch möglich. Die PDO-COB-IDs werden automatisch, wie im Objektverzeichnis beschrieben, gebildet (z. B. 180h + Node-ID für Objekt 1800h).
- Eine Änderung der Node-ID wirkt sich dabei nur solange auf die PDO-COB-ID aus, solange noch KEIN „Save communication objects“ oder „Save all parameters“ durchgeführt wurde. Nach dem Speichern können die PDO-COB-IDs nur manuell geändert werden.

8.5.2 PDO in Synchronmodus setzen

Die PDO-Kommunikation kann für synchrone, d.h. auf Anforderung durchgeführte, Übertragung über SDO konfiguriert werden. Der in Sub-Index 2 geschriebene Wert definiert, nach welcher Anzahl von SYNC-Nachrichten geantwortet wird (Bsp.: Ist 05h eingetragen, wird nur auf jeden 5. SYNC geantwortet).

Beispiel für die Parametrierung von PDO1 mit Wert 01h des Sub-Index 2:

600h+ID	8	2Fh	00h	18h	02h	01h	00h	00h	00h
CAN-ID	DLC	Command	Object L	Object H	Sub-Index	Byte0	Byte1	Byte2	Byte3

Tabelle 8.49: Parametrierung von PDO1 Sub-Index 2

PDO1 ist nun im Synchronmodus. Nach dem Versetzen des Drehgebers in den Operational-Status antwortet der Drehgeber auf jede SYNC-Nachricht.

8.5.3 PDO in Asynchronmodus setzen

Zyklisch (auf internem Event-Timer):

Die PDO-Kommunikation kann auf asynchron-zyklisch durchgeführte Übertragung über SDO konfiguriert werden.

Der in Sub-Index 2 geschriebene Wert muss FFh betragen. Dann kann in Sub-Index 5 die Zykluszeit in Millisekunden parametrieren werden.

Beispiel für die Parametrierung von PDO1 für das PDO-Aussenden auf internem Event-Timer:

600h+ID	8	2Fh	00h	18h	02h	FFh	00h	00h	00h
CAN-ID	DLC	Command	Object L	Object H	Sub-Index	Byte0	Byte1	Byte2	Byte3

Tabelle 8.50: Parametrierung von PDO1 Sub-Index 2

Beispiel für die Parametrierung von PDO1 mit Zykluszeit 30 (= 1Eh) Millisekunden:

600h+ID	8	2Bh	00h	18h	05h	1Eh	00h	00h	00h
CAN-ID	DLC	Command	Object L	Object H	Sub-Index	Byte0	Byte1	Byte2	Byte3

Tabelle 8.51: Parametrierung von PDO1 Sub-Index 5

PDO1 ist nun im Asynchronmodus und sendet, nach dem Versetzen des Drehgebers in Operational Status, alle 30 Millisekunden ein PDO aus.

Auf Messwertänderung:

PDO-Kommunikation für asynchrone, auf Messwertänderung getriggerte Übertragung über SDO konfigurieren. Der in Sub-Index 2 geschriebene Wert muss FEh betragen. Zusätzlich muss in Sub-Index 5 der Wert 00h geschrieben werden.

Beispiel für die Parametrierung von PDO1 für PDO-Senden auf Messwertänderung:

600h+ID	8	2Fh	00h	18h	02h	FEh	00h	00h	00h
CAN-ID	DLC	Command	Object L	Object H	Sub-Index	Byte0	Byte1	Byte2	Byte3

Tabelle 8.52: Parametrierung von PDO1 Sub-Index 2

600h+ID	8	2Bh	00h	18h	05h	00h	00h	00h	00h
CAN-ID	DLC	Command	Object L	Object H	Sub-Index	Byte0	Byte1	Byte2	Byte3

Tabelle 8.53: Parametrierung von PDO1 Sub-Index 5

	<ul style="list-style-type: none"> Nach dem Reset des Drehgebers ist PDO1 im Asynchronmodus und sendet, wenn sich der Messwert ändert, ein PDO aus.
--	--

8.5.4 Variables PDO-Mapping

Durch variables Mapping kann der Inhalt des geberspezifischen TPDOs nach eigenen Wünschen konfiguriert werden. Dieses Mapping muss für den Drehgeber, wie auch für den Empfänger durchgeführt werden. Als Begrenzung gilt die maximale Größe eines PDOs von acht Datenbytes. Ein PDO kann z. B. so gemappt werden, dass damit die „aktuelle Winkelposition“, die „aktuelle Winkelgeschwindigkeit“ und die „aktuelle Winkelbeschleunigung“ in einem übertragen werden. Dies ist von Vorteil, da so, ohne mehr Buslast zu erzeugen, drei Informationen übertragen werden und kein weiterer Overhead benötigt wird.

Zur Verdeutlichung wird die Mapping-Tabelle aufgestellt.

Objekt Nr	Sub-Index	Wert	Größe	Übertragener Wert
6004h	00h	Unsigned32	4 Byte	akt. Position
6030h	01h	Integer16	2 Byte	akt. Geschwindigkeit
6040h	01h	Integer16	2 Byte	akt. Beschleunigung

Tabelle 8.54: Beispiel Mapping-Tabelle

Die Informationen 1, 2 und 3 (siehe Mapping Tabelle 8.54) werden auf die 8 Datenbytes des PDOs verteilt. Dadurch wird das PDO mit einem Datenvolumen von 4 Byte + 2 Byte + 2 Byte = 8 Byte gefüllt und erreicht die volle Auslastung der 8 möglichen Nutzdatenbytes.

Das durch das Mapping resultierende PDO hat damit folgenden Aufbau.

PDO1:

180h+ID	8	1d	1c	1b	1a	2b	2a	3b	3a
CAN-ID	DLC	Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7

Tabelle 8.55: Aufbau PDO1 (Inhalt Tabelle 8.54)

Mit 1a, 1b, 1c, 1d = 4 Bytes der Information 1; 2a, 2b = 2 Bytes der Information 2; 3a, 3b = 2 Bytes der Information 3.

	<ul style="list-style-type: none"> • Damit ein Mapping in den Drehgebern HTB und FHB möglich ist, müssen die Mapping Parameter für die Transmit PDOs konfiguriert werden (siehe Objektverzeichnis Tabelle 6.1 ff.).
---	--

- **Schritt 1** – Mapping Parameter – Konfiguration freischalten
- **Schritt 2** – Mapping des betreffenden PDOs vornehmen
- **Schritt 3** – Mapping Parameter – Konfiguration abschließen

Zur Konfiguration des PDO1 muss z. B. Objekt 1A00h verändert werden. Zuerst muss der Sub-Index 0 auf NULL zurückgesetzt werden (damit ist die Konfiguration für das PDO freigeschaltet). *PDO1: Objekt 1A00h, Sub-Index 0=00h*

Schritt 1 – Mapping Parameter – Konfiguration freischalten

Dazu werden folgende SDO-Befehle gesendet:

600h+ID	8	2Fh	00h	1Ah	00h	00h	00h	00h	00h
CAN-ID	DLC	Command	Object L	Object H	Sub-Index	Byte0	Byte1	Byte2	Byte3

Tabelle 8.56: Mapping Parameter – Konfiguration freischalten

Das PDO ist nun konfigurier-/mappbar.

Schritt 2 – Mapping des betreffenden PDOs vornehmen

Positionswert mappen (Nr.: 1 (Größe 32 Bit = 20h) in Objekt 1A00h Sub-Index 1 für PDO1):

600h+ID	8	23h	00h	1Ah	01h	20h	00h	04h	60h
CAN-ID	DLC	Command	Object L	Object H	Sub-Index	Byte0	Byte1	Byte2	Byte3

Tabelle 8.57: Positionswert mappen

Die Nutzdaten beinhalten die Informationen über das Objekt, dessen Wert per PDO geschickt werden soll, und die Größe des Wertes (Objekt 6004h Sub-Index 2 Größe 20h = 4 Byte).

Geschwindigkeitswert mappen (Nr.: 2 (Größe 16 Bit = 10h) auf Objekt 1A00h Sub-Index 2 für PDO1):

600h+ID	8	23h	00h	1Ah	02h	10h	01h	30h	60h
CAN-ID	DLC	Command	Object L	Object H	Sub-Index	Byte0	Byte1	Byte2	Byte3

Tabelle 8.58: Geschwindigkeitswert mappen

Die Nutzdaten beinhalten die Informationen über das Objekt, dessen Wert per PDO geschickt werden soll, und die Größe des Wertes (Objekt 6030h Sub-Index 1 Größe 10h = 2 Byte).

Beschleunigungswert mappen (Nr.: 3 (Größe 16 Bit = 10h) auf Objekt 1A00h Sub-Index 3 für PDO1):

600h+ID	8	23h	00h	1Ah	03h	10h	01h	40h	60h
CAN-ID	DLC	Command	Object L	Object H	Sub-Index	Byte0	Byte1	Byte2	Byte3

Tabelle 8.59: Beschleunigungswert mappen

Die Nutzdaten beinhalten die Informationen über das Objekt, dessen Wert per PDO geschickt werden soll, und die Größe des Wertes (Objekt 6040h Sub-Index 1 Größe 10h = 2 Byte).

Schritt 3 - Mapping Parameter - Konfiguration abschließen.

Um die Konfiguration zu beenden, muss das gesamte PDO-Mappingformat „gespeichert“ werden (Hier werden drei Objekte gemappt, also auch drei Sub-Indices verwendet = 03h):

600h+ID	8	2Fh	00h	1Ah	00h	03h	00h	00h	00h
CAN-ID	DLC	Command	Object L	Object H	Sub-Index	Byte0	Byte1	Byte2	Byte3

Tabelle 8.60: Mapping Parameter - Konfiguration abschließen

Das Mapping des PDO1 ist nun abgeschlossen. Zum Speichern der Konfiguration siehe 8.12, Einstellungen in das EEPROM speichern.

8.6 Auflösung und Drehsinn ändern

	<ul style="list-style-type: none"> Um die Auflösung des Drehgebers anpassen zu können, muss die Option „Skalieren“ eingeschaltet werden. Im gleichen Arbeitsgang kann die Dreh- bzw. Zählrichtung festgelegt werden, d. h. es kann definiert werden, ob ein Hochzählen der Positionen bei Drehung der Welle (Sicht auf Welle mit Flansch) im Uhrzeigersinn [CW] oder gegen den Uhrzeigersinn geschieht [CCW] (default = im Uhrzeigersinn).
---	---

Die Einstellungen werden im Objekt 6000h Sub-Index 00h vorgenommen. Eine Aufstellung der möglichen Konfigurationen wird im Folgenden gezeigt:

Code Byte 0	Skalierung	Drehsinn
00h	aus	im Uhrzeigersinn CW
01h	aus	gegen Uhrzeigersinn CCW
04h (default)	ein	im Uhrzeigersinn CW
05h	ein	gegen Uhrzeigersinn CCW

Tabelle 8.61: Parameter Drehsinn und Skalierung

Als Beispiel wird eine Nachricht zur Einstellung des Objekts 6000h auf „Skalieren ein“ und „Drehen gegen den Uhrzeigersinn“ gezeigt:

600h+ID	8	2Bh	00h	60h	00h	05h	00h	00h	00h
CAN-ID	DLC	Command	Object L	Object H	Sub-Index	Byte0	Byte1	Byte2	Byte3

Tabelle 8.62: Drehen gegen den Uhrzeigersinn

Der Drehgeber antwortet auf den Befehl mit einer Standard-SDO-Bestätigung.

Nun kann die Singleturn- und die Gesamtauflösung geändert werden.

- Die Singleturnauflösung gibt die Anzahl der Unterteilungen oder Schritte pro Umdrehung der Welle an.
- Die Gesamtauflösung besteht aus dem Produkt von Singleturnauflösung und Anzahl der maximalen Multiturnumdrehungen.

Rechenbeispiel: Singleturn: 4096 Schritte pro Umdrehung = 12 Bit = 10 00h

Gesamtauflösung: 536 870 912 Gesamt-Schritte = 29 Bit = 20 00 00 00h

=> Max. Multiturnumdrehungen: 29 Bit - 12 Bit = 17 Bit = 131072 Schritten (02 00 00h)

Die Singleturnaflösung wird mit einem SDO-Schreibbefehl auf Objekt 6001h geändert:

600h+ID	8	23h	01h	60h	00h	00h	10h	00h	00h
CAN-ID	DLC	Command	Object L	Object H	Sub-Index	Byte0	Byte1	Byte2	Byte3

Tabelle 8.63: Ändern der Singleturnaflösung per SDO

Hier wird als Datenpaket die Soll-Singleturnaflösung (00 00 10 00h von rechts nach links) übertragen. Der Drehgeber antwortet auf den Befehl mit einer Bestätigung.

Die Gesamtauflösung wird über SDO auf Objekt 6002h geändert. Hier wird eine Gesamtauflösung von 29 Bit eingestellt, um bei 12 Bit Singleturn eine Multiturnaflösung von 17 Bit zu erreichen:

600h+ID	8	23h	02h	60h	00h	00h	00h	00h	20h
CAN-ID	DLC	Command	Object L	Object H	Sub-Index	Byte0	Byte1	Byte2	Byte3

Tabelle 8.64: Ändern der Gesamtauflösung per SDO

Als Datenpaket wird die Soll-Gesamtauflösung (20 00 00 00h von rechts nach links) übertragen.

8.7 Wellen-Position setzen

Über die Einstellung des Index 6003h kann der tatsächliche Messwert angepasst werden. So kann z. B. der Nullpunkt des Drehgebers dem Nullpunkt Ihrer Anwendung angeglichen werden. Integrieren Sie dazu den Drehgeber in Ihre Anwendung, dann schreiben Sie in Objekt 6003h den gewünschten Positionswert, den der Drehgeber in der aktuellen Wellen-Position ausgeben soll.

Die Kommunikation läuft dabei wie folgt ab:

Es wird ein SDO mit den Datenbytes p1, p2, p3, p4 gesendet.

600h+ID	8	23h	03h	60h	00h	p1	p2	p3	p4
CAN-ID	DLC	Command	Object L	Object H	Sub-Index	Byte0	Byte1	Byte2	Byte3

Tabelle 8.65: Wellen-Position setzen



- Um den Drehgeber-Nullpunkt dem Nullpunkt Ihrer Anwendung anzugleichen ist p1, p2, p3, p4 = 00h, 00h, 00h, 00h zu wählen.

Um die aktuelle Position zu prüfen, schicken Sie einen SDO-Lesebefehl für die Position (Objekt 6004h):

600h+ID	8	40h	04h	60h	00h	00h	00h	00h	00h
CAN-ID	DLC	Command	Object L	Object H	Sub-Index	Byte0	Byte1	Byte2	Byte3

Tabelle 8.66: aktuelle Position prüfen

Der Drehgeber wird mit einer Nachricht, in dem die aktuelle Position in 4 Datenbytes verschlüsselt ist, antworten.

8.8 Positionswert filtern

Über den Sub-Index 1 des Objektes 2105h kann der Positionswert gefiltert werden. Der Wert des Sub-Index 1 beschreibt die Filterkonstante des internen „IR“-Filters (infinite impulse response filter). Ein Wert von 01h schaltet den Filter aus. Es sind nur Werte zwischen 01h und 04h erlaubt. Der Positionswert ist bei höheren Werten „stabiler“, die Zeit bis zur Aktualisierung eines alten Positionswertes aber steigt minimal.

8.9 Speed-Integration und Skalierung ändern

Die **Integrationszeit**, über welche der Drehgeber die Geschwindigkeit ermittelt, kann über Objekt 2105h Sub-Index 2 in ms angepasst werden. Der Defaultwert hierfür ist 1000 ms und gilt als gute Einstellung für die meisten Anwendungen. Ein Verändern des Werts lässt den ausgegeben Geschwindigkeitswert (Einheit = Inkremente pro Sekunde) träger (hohe Integrationszeit) oder schneller veränderlich (niedrige Integrationszeit) werden. Damit ist durch Sub-Index 2 quasi die Parametrierung eines Filters für die Geschwindigkeitswerte möglich.

Die **Skalierung** der anzeigbaren Geschwindigkeit kann über Objekt 2106h eingestellt werden. Die Sub-Indices 1 (= Zähler) und 2 (= Nenner) bilden dabei eine Variable (hier: „z“) mit welcher der Geschwindigkeitswert skaliert wird. Als Default Einstellung ist die Variable mit dem Wert 1 beschrieben. Die Einheit der Geschwindigkeitsausgabe ist immer Inkremente/s. Das Objekt ist ein signed16-Wert und hat daher einen Bereich von ± 32767 der ± 120 U/min entspricht.

Als Beispiel soll die Geschwindigkeit auf maximal ± 2500 U/min skaliert werden:

$$z = \text{Skalierungsfaktor} \Rightarrow z = \frac{k}{n} \quad (1)$$

$$n = \text{Max Drehzahl in U/min} \Rightarrow z = \frac{120}{2500} \quad (2)$$

$$k = \text{Berechnungsfaktor} = 120 \Rightarrow z = \frac{6}{125} \quad (3)$$

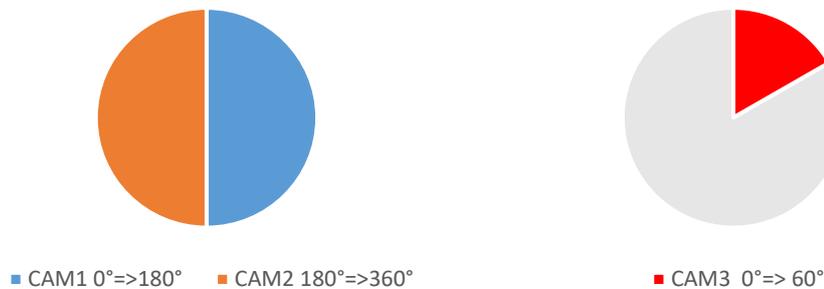
Damit wäre Sub-Index 1 des Objekt 2106h mit 6d = 06h und Sub-Index 2 mit 125d = 7Dh zu beschreiben. Das bedeutet: Die Grenzen ± 32767 entsprechen bei Skalierung mit obigen Werten somit ± 2500 U/min.

8.10 Frequency-Limit

Über den Index 2107h kann die Geschwindigkeitsgrenze eingestellt werden, bei deren Überschreitung ein Warning gesetzt wird. Der zulässige Wertebereich ist 1 bis 65535 und beschreibt die maximale „Drehzahl“ der Drehgeber-Welle in Umdrehungen pro Sekunde. z.B.: 2520 U/min = 42 U/s = 002Ah als Frequency-Limit-Wert.

8.11 CAM-Konfiguration

Es soll beispielhaft der CAM-Kanal wie folgt konfiguriert werden:



Dies bedeutet für die einzelnen CAMs:

CAM	Winkelbereich	Wert unteres CAM-Limit	Wert oberes CAM-Limit	Hysterese
1	0°..180°	0	2048	0
2	180°..360°	2049	4095	0
3	0°..60°	0	682	0

Tabelle 8.67: Beispiel CAM-Konfiguration

- Die Konfiguration des CAM-Kanals ist im Pre-Operational-Status durchzuführen

Um die Nockenschaltfunktion „anzuschalten“, muss zuerst das CAM-enable-register in Objekt 6301 Sub-Index 1 beschrieben werden. In folgendem Beispiel werden nun die ersten drei CAMs „angeschaltet“, indem der Wert 00000111b = 07h in das Register geschrieben wird.

600h+ID	8	23h	01h	63h	01h	07h	00h	00h	00h
CAN-ID	DLC	Command	Object L	Object H	Sub-Index	Byte0	Byte1	Byte2	Byte3

Tabelle 8.68: Nockenschaltfunktion aktivieren

Nun werden die CAM-high-limits 1, 2 und 3 nach obiger Aufstellung beschrieben:

CAM 1 = 2048 = 0800h

600h+ID	8	23h	20h	63h	01h	00h	08h	00h	00h
CAN-ID	DLC	Command	Object L	Object H	Sub-Index	Byte0	Byte1	Byte2	Byte3

Tabelle 8.69: CAM-High-Limit 1

CAM 2 = 4095 = 0FFFh

600h+ID	8	23h	21h	63h	01h	FFh	0Fh	00h	00h
CAN-ID	DLC	Command	Object L	Object H	Sub-Index	Byte0	Byte1	Byte2	Byte3

Tabelle 8.70: CAM-High-Limit 2

CAM 3 = 682 = 02AAh

600h+ID	8	23h	23h	63h	01h	AAh	02h	00h	00h
CAN-ID	DLC	Command	Object L	Object H	Sub-Index	Byte0	Byte1	Byte2	Byte3

Tabelle 8.71: CAM-High-Limit 3

Nun werden die CAM-low-limits 1, 2 und 3 nach obiger Aufstellung beschrieben:

CAM 1 = 0 = 00h

600h+ID	8	23h	10h	63h	01h	00h	00h	00h	00h
CAN-ID	DLC	Command	Object L	Object H	Sub-Index	Byte0	Byte1	Byte2	Byte3

Tabelle 8.72: CAM-Low-Limit 1

CAM 2 = 2049 = 0801h

600h+ID	8	23h	11h	63h	01h	01h	08h	00h	00h
CAN-ID	DLC	Command	Object L	Object H	Sub-Index	Byte0	Byte1	Byte2	Byte3

Tabelle 8.73: CAM-Low-Limit 2

CAM 3 = 0 = 00h

600h+ID	8	23h	12h	63h	01h	00h	00h	00h	00h
CAN-ID	DL C	Command	Object L	Object H	Sub- Index	Byte0	Byte1	Byte2	Byte3

Tabelle 8.74: CAM-Low-Limit 3

Die CAM-hysteresis bleibt 0 und muss nicht neu beschrieben werden.

Über Objekt 6300 Sub-Index 1 kann nun das CAM-state-register ausgelesen werden. Das CAM-state-register ist auch PDO mapbar! Zur Interpretation des Wertes, der sich im Sub-Index 1 einstellt, siehe auch 7.5.1 CAM-state-register. Zum Speichern der Konfiguration siehe 8.12, Einstellungen in das EEPROM speichern.

8.12 Einstellungen in das EEPROM speichern

8.12.1 Netzausfallsicheres Speichern von Parametern

Parameter sind über den Index 1010h speicherbar.

Sub-Index	Zugriff	Bedeutung
0	co	Number of objects
1	wo	Save all parameters
2	wo	Save communication objects
3	wo	Save application objects
4	wo	Save manufacturer objects

Tabelle 8.75: Parameter-Speicheroptionen



- Das Abspeichern wird ausgelöst, indem der entsprechende Sub-Index mit dem „ASCII“ Wert „save“ (in hex: 65766173h) beschrieben wird.

Als Beispiel wird eine „Save all parameters“ Nachricht gezeigt:

600h+ID	8	23h	10h	10h	01h	73h	61h	76h	65h
CAN-ID	DLC	Command	Object L	Object H	Sub- Index	Byte0	Byte1	Byte2	Byte3

Tabelle 8.76: Beispiel – Save all parameters

8.12.2 Parameter zurücksetzen auf Werkseinstellungen

Das Laden der Werkseinstellung ist über Index 1011h möglich.

Sub-Index	Zugriff	Bedeutung
0	co	Number of objects
1	wo	Restore all parameters
2	wo	Restore communication objects
3	wo	Restore application objects
4	wo	Restore manufacturer objects

Tabelle 8.77: Parameter-Speicheroptionen



- Das Laden der Ursprungsparameter wird ausgelöst, indem der entsprechende Sub-Index mit dem ASCII Wert „load“ (in hex: 64616F6Ch) beschrieben wird.
- Achtung: Die Baudraten- und die Node-ID-Einstellungen werden hierdurch nicht zurückgesetzt! Auch das Objekt „Customer-Data“ ist nicht betroffen.

9 Fehler-Diagnose

9.1 Troubleshooting Gebereinstellungen

Fehlerbeschreibung	Zu prüfen
Drehgeber funktioniert nicht, es leuchtet keine LED.	Prüfen Sie die Anschlüsse, Spannungsversorgung und Anschlussbelegung.
Drehgeber funktioniert nicht, ist aber korrekt angeschlossen.	Verbinden Sie ein CAN Monitoring-Tool. Prüfen Sie, ob der Geber beim Start eine Boot-up-Message aussendet.
Es kann keine Verbindung zum Geber hergestellt werden.	Prüfen Sie die Baudrate und die Node-ID des Gebers auf ihre Korrektheit.
LED leuchtet orange	Geber hängt in der Boot-up-Phase und kann keine Boot-up-Message aussenden. Bitte Buslast prüfen und ggf. reduzieren!
Die Buslast liegt über 85 Prozent.	Zu viele Error-Nachrichten auf dem Bus, Abschlusswiderstände und Stickleitungslänge überprüfen.
Der Drehgeber geht nach Anschluss an den Bus sofort Bus-passive bzw. Bus-off.	Prüfen Sie die Baudrate und die Node-ID aller Geräte im Bus auf ihre Korrektheit.
In unregelmäßigen Abständen treten bei der Übertragung Störungen auf.	Prüfen Sie, ob die Abschlusswiderstände korrekt angeschlossen sind und den richtigen Wert haben. Nur zwei Abschlusswiderstände an je einem Ende dürfen vorhanden sein!

Tabelle 9.1: Fehler-Diagnose Gebereinstellungen

