

**CANopen**<sup>®</sup>

- + 2 Magnete
- + Position
- + Geschwindigkeit

# Linear-Encoder CANopen Profil

**Technische  
Information**

---

## **TR-Electronic GmbH**

D-78647 Trossingen

Eglishalde 6

Tel.: (0049) 07425/228-0

Fax: (0049) 07425/228-33

E-mail: [info@tr-electronic.de](mailto:info@tr-electronic.de)

[www.tr-electronic.de](http://www.tr-electronic.de)

---

### **Urheberrechtsschutz**

Dieses Handbuch, einschließlich den darin enthaltenen Abbildungen, ist urheberrechtlich geschützt. Drittanwendungen dieses Handbuchs, welche von den urheberrechtlichen Bestimmungen abweichen, sind verboten. Die Reproduktion, Übersetzung sowie die elektronische und fotografische Archivierung und Veränderung bedarf der schriftlichen Genehmigung durch den Hersteller. Zuwiderhandlungen verpflichten zu Schadenersatz.

---

### **Änderungsvorbehalt**

Jegliche Änderungen, die dem technischen Fortschritt dienen, vorbehalten.

---

### **Dokumenteninformation**

Ausgabe-/Rev.-Datum: 30.03.2016  
Dokument-/Rev.-Nr.: TR - ELA - TI - D - 0039 - 01  
Dateiname: TR-ELA-TI-D-0039-01.docx  
Verfasser: MÜJ / BER

---

### **Schreibweisen**

*Kursive* oder **fette** Schreibweise steht für den Titel eines Dokuments oder wird zur Hervorhebung benutzt.

*Courier*-Schrift zeigt Text an, der auf dem Display bzw. Bildschirm sichtbar ist und Menüauswahlen von Software.

" < > " weist auf Tasten der Tastatur Ihres Computers hin (wie etwa <RETURN>).

---

### **Marken**

CANopen<sup>®</sup> und CiA<sup>®</sup> sind eingetragene Gemeinschaftsmarken der CAN in Automation e.V.

---

# Inhaltsverzeichnis

<b>Inhaltsverzeichnis .....</b>	<b>3</b>
<b>Änderungs-Index .....</b>	<b>5</b>
<b>1 Referenzen .....</b>	<b>6</b>
<b>2 Definitionen und Abkürzungen .....</b>	<b>7</b>
<b>3 Einführung .....</b>	<b>8</b>
<b>4 Kommunikations-Profil .....</b>	<b>9</b>
Positionsausgabewert .....	9
Geschwindigkeit / CAN State .....	9
4.1 Erstes Sende-Prozessdaten-Objekt .....	9
4.2 Zweites Sende-Prozessdaten-Objekt .....	9
<b>5 Standard Objekte .....</b>	<b>10</b>
5.1 Detaillierte Beschreibung der kommunikationsspezifischen Objekten .....	11
5.1.1 Objekt 1000h: Gerätetyp .....	11
5.1.2 Objekt 1001h: Fehlerregister .....	11
5.1.3 Objekt 1002h: Hersteller-Status-Register .....	12
5.1.4 Objekt 1003h: Vordefiniertes Fehlerfeld .....	12
5.1.5 Objekt 1004h: Anzahl unterstützter PDO's .....	12
5.1.6 Objekt 1005h: COB-ID SYNC Nachricht .....	13
5.1.7 Objekt 1008h: Hersteller Gerätenamen .....	13
5.1.8 Objekt 1009h: Hersteller Hardwareversion .....	13
5.1.9 Objekt 100Ah: Hersteller Softwareversion .....	13
5.1.10 Objekt 100Bh: Node-ID .....	14
5.1.11 Objekt 100Ch: Guard-Time (Überwachungszeit) .....	14
5.1.12 Objekt 100Dh: Life-Time-Faktor (Zeitdauer-Faktor) .....	14
5.1.13 Objekt 100Eh: Node-Guarding-Identifizier .....	15
5.1.14 Objekt 1010h: Parameter abspeichern .....	15
<b>6 Standardisierter Encoder-Profilbereich .....</b>	<b>17</b>
6.1 Encoder Parameter .....	18
6.1.1 Objekt 6000h - Betriebsparameter .....	18
6.1.2 Objekt 6001h - Meßschritte pro Umdrehung .....	19
6.1.3 Objekt 6002h - Gesamtmeßlänge in Schritten .....	19
6.1.4 Objekt 6003h - Presetwert .....	20
6.1.5 Objekt 6004h - Positionswert .....	20
6.1.6 Objekt 6005h - Meßschritt-Einstellungen bei Linear-Encodern .....	20
6.1.7 Objekt 6010h - Presetwert Multisensor .....	21
6.1.8 Objekt 6020h - Positionswert Multisensor .....	21
6.1.9 Objekt 6200h - Cyclic-Timer .....	22

6.2 Encoder Diagnose .....	22
6.2.1 Objekt 6500h - Betriebsstatus .....	22
6.2.2 Objekt 6501h - Single-Turn Auflösung (Drehgeber), Meßschritt (Linear-Encoder).....	23
6.2.2.1 Drehgeber .....	23
6.2.2.2 Linear-Encoder.....	23
6.2.3 Objekt 6502h - Anzahl der Umdrehungen .....	24
6.2.4 Objekt 6503h - Alarmer .....	25
6.2.5 Objekt 6504h - Unterstützte Alarmer .....	26
6.2.6 Objekt 6505h - Warnungen.....	26
6.2.7 Objekt 6506h - Unterstützte Warnungen .....	26
6.2.8 Objekt 6507h - Profil- und Softwareversion .....	27
6.2.9 Objekt 6508h - Betriebszeit.....	27
6.2.10 Objekt 6509h - Offsetwert .....	27
6.2.11 Objekt 650Ah - Hersteller-Offsetwert.....	27
6.2.12 Objekt 650Bh - Serien-Nummer .....	27
<b>7 Emergency-Meldung .....</b>	<b>28</b>
<b>8 Inbetriebnahme des Encoders am CAN-Bus .....</b>	<b>29</b>
8.1 Übertragung des Encoder-Positionswertes .....	29
8.2 Lese-/Schreib- Service-Daten-Objekt .....	30
8.2.1 Lese SDO: .....	30
8.2.2 Schreibe SDO: .....	31

## Änderungs-Index

Änderung	Datum	Index
Erstausgabe	17.07.02	00
Generelle Überarbeitung	30.03.16	01

### 1 Referenzen

- | 1 | : ISO 11898: Straßenfahrzeuge, Austausch von Digitalinformation - Controller Area Network (CAN) für Hochgeschwindigkeits-Kommunikation, November 1993
- | 2 | : Robert Bosch GmbH, CAN-Spezifikation 2.0 Teil A und B, September 1991
- | 3 | : CiA DS-201 V1.1, CAN im OSI Referenz-Model, Februar 1996
- | 4 | : CiA DS-202-1 V1.1, CMS Service Spezifikation, Februar 1996
- | 5 | : CiA DS-202-2 V1.1, CMS Protokoll Spezifikation, Februar 1996
- | 6 | : CiA DS-202-3 V1.1, CMS Verschlüsselungsregeln, Februar 1996
- | 7 | : CiA DS-203-1 V1.1, NMT Service Spezifikation, Februar 1996
- | 8 | : CiA DS-203-2 V1.1, NMT Protokoll Spezifikation, Februar 1996
- | 9 | : CiA DS-204-1 V1.1, DBT Service Spezifikation, Februar 1996
- | 10 | : CiA DS-204-2 V1.1, DBT Protokoll Spezifikation, Februar 1996
- | 11 | : CiA DS-205-1 V1.1, LMT Service Spezifikation, Februar 1996
- | 12 | : CiA DS-205-2 V1.1, LMT Protokoll Spezifikation, Februar 1996
- | 13 | : CiA DS-206 V1.1, Empfohlene Namenskonventionen für die Schichten, Februar 1996
- | 14 | : CiA DS-207 V1.1, Namenskonventionen der Verarbeitungsschichten, Februar 1996
- | 15 | : CiA DS-301 V3.0, CANopen Kommunikationsprofil auf CAL basierend, Oktober 1996
- | 16 | : CiA DS-406 V2.0, CANopen Profil für Encoder, Mai 1998

---

## 2 Definitionen und Abkürzungen

### CAL

CAN Application Layer. Die Anwendungsschicht für CAN-basierende Netzwerke ist im CiA-Draft-Standard 201 ... 207 beschrieben.

### CAN

Controller Area Network. Datenstrecken-Schicht-Protokoll für serielle Kommunikation, beschrieben in der ISO 11898.

### CiA

CAN in Automation. Internationale Anwender- und Herstellervereinigung e.V.: gemeinnützige Vereinigung für das Controller Area Network (CAN).

### CMS

CAN-based Message Specification. Eines der Serviceelemente in der Anwendungsschicht im CAN Referenz-Model.

### COB

Communication Object (CAN Message). Übertragungseinheit im CAN Netzwerk. Daten müssen in einem COB durch das CAN Netzwerk gesendet werden.

### COB-ID

COB-Identifizier. Eindeutige Zuordnung des COB. Der Identifizier bestimmt die Priorität des COB's im Busverkehr.

### DBT

Distributor. Eines der Serviceelemente in der Anwendungsschicht im CAN Referenz-Model. Es liegt in der Verantwortung des DBT's, COB-ID's an die COB's zu verteilen, die von der CMS benutzt werden.

### LMT

Layer Management. Eines der Serviceelemente in der Anwendungsschicht im CAN Referenz-Model. Wird benötigt, um Parameter in den einzelnen Schichten zu konfigurieren.

### NMT

Network Management. Eines der Serviceelemente in der Anwendungsschicht im CAN Referenz-Model. Führt die Initialisierung, Konfiguration und Fehlerbehandlung im Busverkehr aus.

### PDO

Process Data Object. Objekt für den Datenaustausch zwischen mehreren Geräten.

### SDO

Service Data Object. Punkt zu Punkt Kommunikation mit Zugriff auf die Objekt-Datenliste eines Gerätes.

### 3 Einführung

Die CAN-Bus-Schnittstelle ist durch die internationale Norm ISO/DIS 11898 definiert und spezifiziert die zwei untersten Schichten des CAN Referenz-Modells.

Die CAN-Bus-Schnittstelle mit dem Bustreiber PCA83C250T ist galvanisch von der Encoderelektronik getrennt und wird über einen internen DC/DC-Konverter gespeist. Eine externe Spannungsversorgung für den Bustreiber ist nicht notwendig.

Die Konvertierung der Encoderinformation in das CAN-Protokoll (CAN 2.0A) geschieht über den CAN-Kontroller PCA82C200. Die Funktion des CAN-Kontrollers wird durch einen Watchdog überwacht.

Das CANopen Kommunikationsprofil (CiA Standard DS 301) basiert auf dem CAN Application Layer (CAL) und beschreibt, wie die Dienste von Geräten benutzt werden. Das CANopen Profil erlaubt die Definition von Geräteprofilen für eine dezentralisierte E/A.

Die Encoder mit CANopen Protokoll unterstützen das Geräteprofil für Encoder (CiA Draft Standard 406, Version 2.0). Die Encoder unterstützen auch den erweiterten Funktionsumfang in Klasse C2.

Die Kommunikations-Funktionalität und Objekte, welche im Encoderprofil benutzt werden, werden in einer EDS-Datei (Electronic Data Sheet) beschrieben. Wird ein CANopen Konfigurations-Hilfsprogramm benutzt (z.B. CANSETTER), kann der Benutzer die Objekte (SDO's) des Encoders auslesen und die Funktionalität programmieren.

Die Auswahl der Übertragungsrate und Node-ID (Geräteadresse) erfolgt über Schalter.

## 4 Kommunikations-Profil

Im Gerät sind zwei Prozessdaten-Objekte (PDO) implementiert und die Übertragungsarten folgendermaßen festgelegt:

Übertragungsart = 1: Übertragung erfolgt synchron auf Anforderung über Remote-Frame oder SYNC-Telegramm.

Übertragungsart = 254: Übertragung erfolgt asynchron. Der Timerwert ist im Index 6200h gespeichert.

Der Istwert wird im Binärcode übertragen:

COB-ID	Positionsausgabewert				Geschwindigkeit / CAN State		
	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6
11 Bit	$2^7$ bis $2^0$	$2^{15}$ bis $2^8$	$2^{23}$ bis $2^{16}$	$2^{31}$ bis $2^{24}$	$2^7$ bis $2^0$	$2^{15}$ bis $2^8$	0

### 4.1 Erstes Sende-Prozessdaten-Objekt

Dieses PDO überträgt den Encoder-Istwert des ersten Magneten.

Index	Subindex	Kommentar	Standardwert
1800h	0	Anz. unterstützter Einträge	3
	1	COB-ID benützt durch PDO 1	180h + Node-ID
	2	Übertragungsart	1
	3	Sperrzeit	0
1A00h	0	Anz. abgebildeter Objekte	3
	1	Positionswert, 1. Magnet	60200120h
	2	Geschwindigkeit, 1. Magnet	60300110h
	3	CAN State, 1. Magnet	63000108h

### 4.2 Zweites Sende-Prozessdaten-Objekt

Dieses PDO überträgt den Encoder-Istwert des zweiten Magneten.

Index	Subindex	Kommentar	Standardwert
1801h	0	Anz. unterstützter Einträge	3
	1	COB-ID benützt durch PDO 2	280 + Node-ID
	2	Übertragungsart	1
	3	Sperrzeit	0
1A01h	0	Anz. abgebildeter Objekte	3
	1	Positionswert, 2. Magnet	60200220h
	2	Geschwindigkeit, 2. Magnet	60300210h
	3	CAN State, 2. Magnet	63000208h

## 5 Standard Objekte

Folgende Tabelle zeigt eine Übersicht der unterstützten Indexe im Kommunikationsprofilbereich:

Index (h)	Objekt	Name	Typ	Attr.
1000	VAR	Gerätetyp	Unsigned32	const
1001	VAR	Fehlerregister	Unsigned8	ro
1002	VAR	Hersteller-Status-Register	Unsigned32	ro
1003	ARRAY	Vordefiniertes Fehlerfeld	Unsigned32	ro
1004	ARRAY	Anzahl unterstützter PDO's	Unsigned32	ro
1005	VAR	COB-ID SYNC-Nachricht	Unsigned32	rw
1008	VAR	Hersteller Gerätenamen	Vis-String	const
1009	VAR	Hardwareversion	Vis-String	const
100A	VAR	Softwareversion	Vis-String	const
100B	VAR	Node-ID (Geräteadresse)	Unsigned32	ro
100C	VAR	Guard-Time (Überwachungszeit)	Unsigned32	rw
100D	VAR	Life-Time-Faktor (Zeitdauer-Faktor)	Unsigned32	rw
100E	VAR	COB-ID Guarding-Protokoll (Wächterprotokoll)	Unsigned32	ro
1010	VAR	Parameter abspeichern	Unsigned32	rw

## 5.1 Detaillierte Beschreibung der kommunikationsspezifischen Objekten

### 5.1.1 Objekt 1000h: Gerätetyp

Beinhaltet Information über den Gerätetyp. Das Objekt mit Index 1000h beschreibt den Gerätetyp und seine Funktionalität. Es besteht aus einem 16 Bit Feld, welches das benutzte Geräteprofil beschreibt (Geräteprofil-Nr. 406 = 196h) und ein zweites 16 Bit Feld, welches Informationen über den Gerätetyp liefert.

#### Parameterstruktur

Unsigned32

Gerätetyp			
Geräte-Profil-Nummer		Encoder-Typ	
Byte 0	Byte 1	Byte 2	Byte 3
196h		$2^7$ bis $2^0$	$2^{15}$ bis $2^8$

#### Encoder-Typ

Code	Definition
01	Absoluter Single-Turn Drehgeber
02	Absoluter Multi-Turn Drehgeber
08	Absoluter Lineargeber
09	Absoluter Lineargeber mit zyklischer Kodierung

### 5.1.2 Objekt 1001h: Fehlerregister

Dieses Objekt beinhaltet das Fehlerregister für das Gerät. Falls ein Alarm-Bit (Objekt 6503) gesetzt wird, wird im Fehlerregister das Bit 5 gesetzt.

Unsigned8

Bit	Bedeutung
0	generischer Fehler
1	0
2	0
3	0
4	0
5	geräteprofilsspezifisch
6	0
7	0

### 5.1.3 Objekt 1002h: Hersteller-Status-Register

Dieses Objekt wird durch den Encoder nicht verwendet, bei Lesezugriff ist der Wert immer "0".

### 5.1.4 Objekt 1003h: Vordefiniertes Fehlerfeld

Dieses Objekt beinhaltet einen aufgetretenen Encoderfehler und zeigt den Fehler über das Emergency-Objekt an.

Index	Subindex	Kommentar	Typ
1003h	0	Anzahl der Fehler	Unsigned8
	1	Standard Fehlerfeld	Unsigned32

Subindex 0: Der Eintrag in Subindex 0 beinhaltet die Anzahl der aufgetretenen Fehler und registriert sie in Subindex 1.

Subindex 1: Das Fehlerfeld setzt sich aus einem 16 Bit Fehlercode und einer 16 Bit Zusatz-Fehlerinformation zusammen.

Unsigned32

Standard Fehlerfeld			
Byte 0	Byte 1	Byte 2	Byte 3
Fehlercode		Zusatz-Fehlerinformation	

### 5.1.5 Objekt 1004h: Anzahl unterstützter PDO's

Dieses Objekt beinhaltet die Information über die max. Anzahl der PDO's, die durch den Encoder unterstützt werden.

Index	Subindex	Kommentar	Typ
1004h	0	Anzahl der unterstützten PDO's	Unsigned32
	1	Anzahl der synchronen PDO's	Unsigned32
	2	Anzahl der asynchronen PDO's	Unsigned32

Subindex 0 beschreibt die Gesamtanzahl der unterstützten PDO's (synchron und asynchron).

Subindex 1 beschreibt die Anzahl der synchronen PDO's die durch den Encoder unterstützt werden.

Subindex 2 beschreibt die Anzahl der asynchronen PDO's die durch den Encoder unterstützt werden.

Anzahl der PDO's			
Byte 0	Byte 1	Byte 2	Byte 3
gesendete PDO's		empfangene PDO's	

Subindex 0: gesendete PDO's = 2, empfangene PDO's = 0

Subindex 1: gesendete PDO's = 1, empfangene PDO's = 0

Subindex 2: gesendete PDO's = 1, empfangene PDO's = 0

### 5.1.6 Objekt 1005h: COB-ID SYNC Nachricht

Dieses Objekt definiert die COB-ID des Synchronisierung-Objekts (SYNC). Es definiert weiterhin, ob das Gerät die SYNC-Nachricht verarbeitet, oder ob das Gerät die SYNC-Nachricht erzeugt.

Unsigned32				
MSB				LSB
31	30	29	28-11	10-0
1	0	0	0	00 1000 0000

Bit 31 = 1 , Gerät verarbeitet die SYNC-Nachricht  
 Bit 30 = 0 , Gerät erzeugt keine SYNC-Nachricht  
 Bit 29 = 0 , 11 Bit ID (CAN 2.0A)  
 Bit 28 -11 = 0  
 Bit 10 - 0 = 11 Bit SYNC-COB-IDENTIFIER, Standardwert = 080h

Wenn ein SYNC-Telegramm mit der Identifier, definiert in diesem Objekt (080h), und Datenlänge = 0 vom Gerät empfangen worden ist, wird der Positionswert des Encoders durch das zweite Sende-Prozessdaten-Objekt (Objekt 1802) übertragen.

### 5.1.7 Objekt 1008h: Hersteller Gerätenamen

Beinhaltet den Hersteller Gerätenamen (visible string).

### 5.1.8 Objekt 1009h: Hersteller Hardwareversion

Beinhaltet die Hersteller Hardwareversion (visible string).

### 5.1.9 Objekt 100Ah: Hersteller Softwareversion

Beinhaltet die Hersteller Softwareversion (visible string).  
 Siehe auch Objekt 6507.

### 5.1.10 Objekt 100Bh: Node-ID

Dieses Objekt beinhaltet die Node-ID (Geräteadresse).

Der Wert wird durch 6 Hardware-Schalter eingestellt und kann nicht durch die Benutzung von SDO-Diensten geändert werden.

Unsigned32

Node_ID			
Byte 0	Byte 1	Byte 2	Byte 3
Node-ID	reserviert	reserviert	reserviert

Wertebereich: 1 - 64.

Die Node-ID ist die eingestellte Hardwareadresse durch Schalter + 1. Dies bedeutet:

alle 6 Schalter auf OFF = 0, Node-ID = 1  
Schalter Bit 5 = ON = 32, Node-ID = 33

### 5.1.11 Objekt 100Ch: Guard-Time (Überwachungszeit)

Die Objekte der Indexe 100Ch und 100Dh beinhalten die Guard-Time in Milli-Sekunden und den Live-Time-Faktor (Zeitdauer-Faktor). Der Live-Time-Faktor multipliziert mit der Guard-Time ergibt die Zeitdauer für das Node-Guarding-Protokoll.

Unsigned16

Guard-Time	
Byte 0	Byte 1
$2^7$ bis $2^0$	$2^{15}$ bis $2^8$

### 5.1.12 Objekt 100Dh: Life-Time-Faktor (Zeitdauer-Faktor)

Der Live-Time-Faktor multipliziert mit der Guard-Time ergibt die Zeitdauer für das Node-Guarding-Protokoll.

Unsigned8

Life-Time-Faktor
Byte 0
$2^7$ bis $2^0$

### 5.1.13 Objekt 100Eh: Node-Guarding-Identifizier

Die Identifizier wird für die Node-Guarding- und die Life-Guarding-Prozedur benötigt.

Unsigned32

MSB				LSB	
31	30	29	28-11	10-0	
reserviert	0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0			11 Bit Identifizier

Bit 10 - 0 = 11 Bit Identifizier, Wert = 700h + Node-ID

### 5.1.14 Objekt 1010h: Parameter abspeichern

Dieses Objekt unterstützt das Abspeichern von Parametern in den nichtflüchtigen Speicher (EEPROM).

Index	Subindex	Kommentar	Typ
1010h	0	größter unterstützte Subindex	Unsigned8
	1	alle Parameter speichern	Unsigned32

**Subindex0:** Der Eintrag in Subindex 0 enthält den größten unterstützten Subindex. Wert = 1.

**Subindex1:** Bei Lesezugriff liefert das Gerät Informationen über seine Speichermöglichkeit.

Unsigned32		MSB		LSB	
Bits	31-2	1	0		
Wert	= 0	0	1		

Bei Lesezugriff liefert das Gerät Informationen über seine Speichermöglichkeit.

Bit 0 = 1, das Gerät speichert Parameter nur auf Kommando. Dies bedeutet, wenn Parameter durch den Benutzer geändert worden sind und das Kommando "Parameter abspeichern" nicht ausgeführt worden ist, nach dem nächsten Einschalten der Betriebsspannung, die Parameter wieder die alten Werte besitzen.

Bei Schreibzugriff speichert das Gerät die Parameter in den nichtflüchtigen Speicher. Um eine versehentliche Speicherung der Parameter zu vermeiden, wird die Speicherung nur ausgeführt, wenn eine spezielle Signatur in das Objekt geschrieben wird. Die Signatur heißt "save".

Unsigned32		MSB		LSB	
Signatur					
	e	v	a	s	
	65h	76h	61h	73h	

Beim Empfang der richtigen Signatur speichert das Gerät die Parameter ab. Schlug die Speicherung fehl, antwortet das Gerät mit Abbruch der Übertragung, Fehlerklasse 6, Fehlerkennung 6 (Hardwarefehler), siehe auch Objekt 6503h.

Wurde eine falsche Signatur geschrieben, verweigert das Gerät die Speicherung und antwortet mit Abbruch der Übertragung, Fehlerklasse 8, Fehlerkennung 0.

## 6 Standardisierter Encoder-Profilbereich

Die Einträge der Dateiliste von 6000h bis 65FFh werden von jedem Encoder genutzt.  
Die Einträge sind allgemein für Encoder.  
Beachten Sie, daß alle Einträge in der Spalte "Index" hexadezimal angegeben sind.

Die untenstehende Übersicht zeigt alle gemeinsamen Einträge:

Index	Objekt	Name	Datenlänge	Attr.
<b>Parameter</b>				
6000	VAR	Betriebsparameter	Unsigned16	rw
6001	VAR	Meßschritte pro Umdrehung	Unsigned32	rw
6002	VAR	Gesamtmeßlänge in Schritten	Unsigned32	rw
6003	VAR	Presetwert	Unsigned32	rw
6004	VAR	Positionswert	Unsigned32	ro
6005	REC	Linear-Encoder: Meßschritt-Einstellung	Unsigned32	ro
6010	VAR	Presetwert Multisensor	Unsigned32	rw
6020	VAR	Positionswert Multisensor	Unsigned32	r
6200	VAR	Cyclic-Timer	Unsigned16	rw
<b>Diagnose</b>				
6500	VAR	Betriebsstatus	Unsigned16	r
6501	VAR	Single-Turn Auflösung (Drehgeber), Meßschritt (Linear-Encoder)	Unsigned32	r
6502	VAR	Anzahl der Umdrehungen	Unsigned16	r
6503	VAR	Alarmer	Unsigned16	r
6504	VAR	Unterstützte Alarmer	Unsigned16	r
6505	VAR	Warnungen	Unsigned16	r
6506	VAR	Unterstützte Warnungen	Unsigned16	r
6507	VAR	Profil- und Softwareversion	Unsigned32	r
6508	VAR	Betriebszeit	Unsigned32	r
6509	VAR	Offsetwert	Signed32	r
650A	VAR	Hersteller-Offsetwert	Signed32	r
650B	VAR	Serien-Nummer	Unsigned32	r

Auf den folgenden Seiten wird jedes Objekt im Detail erklärt.

## 6.1 Encoder Parameter

### 6.1.1 Objekt 6000h - Betriebsparameter

Die Betriebsparameter beinhalten die Funktionen für die Zählrichtung, Inbetriebnahmediagnose- und Skalierungsfunktionssteuerung.

#### Parameterstruktur

Unsigned16

Bit	Funktion	Bit = 0	Bit = 1
0	Zählrichtung	CW	CCW
1	Inbetriebnahmediagnose	gesperrt	freigegeben
2	Skalierungsfunktion	gesperrt	freigegeben
3 - 11	reserviert		
12-15	herstellerspezifische Funktionen		

#### Zählrichtung:

Die Zählrichtung definiert, ob steigende oder fallende Positionswerte ausgegeben werden, wenn die Encoderwelle im Uhrzeigersinn oder Gegenuhrzeigersinn gedreht wird (Blickrichtung auf die Welle).

Für Linear-Encoder wird festgelegt, ob steigende oder fallende Positionswerte zum Ende des Stabes ausgegeben werden.

0 = steigend, 1 = fallend

#### Inbetriebnahmediagnosesteuerung:

Nicht unterstützt.

#### Skalierungsfunktionssteuerung:

Mit der Skalierungsfunktion wird der numerische Encoderwert softwaremäßig konvertiert, um die physikalische Auflösung des Encoders zu ändern.

Die Parameter "Meßschritte pro Umdrehung" und "Gesamtmeßlänge in Meßschritten" sind die Skalierungsparameter. Das Skalierungsfunktionsbit wird in den Betriebsparametern gesetzt. Wenn das Skalierungsfunktionsbit gelöscht wird, wird die Skalierungsfunktion gesperrt und die zwei Skalierungsparameter auf ihre Standardwerte gesetzt.

Bevor die Parameter "Meßschritte pro Umdrehung" und "Gesamtmeßlänge in Meßschritten" an den Encoder übertragen werden, muß das Skalierungsfunktionsbit gesetzt werden.

Um die neuen Parameterwerte zu speichern, muß nach dem Übertragen der Parameter das Kommando "Parameter abspeichern" ausgeführt werden.

### 6.1.2 Objekt 6001h - Meßschritte pro Umdrehung

Der Parameter "Meßschritte pro Umdrehung" legt die Anzahl der Schritte pro Umdrehung fest.

#### Parameterstruktur

Unsigned32

Meßschritte pro Umdrehung			
Byte 0	Byte 1	Byte 2	Byte 3
$2^7$ bis $2^0$	$2^{15}$ bis $2^8$	$2^{23}$ bis $2^{16}$	$2^{31}$ bis $2^{24}$

#### Drehgeber

Standardwert: 4096 = 1000h (abhängig von der Kapazität, siehe Typenschild).

#### Linear-Encoder

Für Linear-Encoder ist der Wert immer "1".

### 6.1.3 Objekt 6002h - Gesamtmeßlänge in Schritten

Der Parameter "Gesamtmeßlänge in Schritten" legt die Anzahl der Schritte über den gesamten Meßbereich fest.

#### Parameterstruktur

Unsigned32

Gesamtmeßlänge in Schritten			
Byte 0	Byte 1	Byte 2	Byte 3
$2^7$ bis $2^0$	$2^{15}$ bis $2^8$	$2^{23}$ bis $2^{16}$	$2^{31}$ bis $2^{24}$

#### Drehgeber

Standardwert: 16 777 216 = 1 00 00 00h  
(abhängig von der Kapazität, siehe Typenschild).

Meßlänge in Schritten = Meßschritte pro Umdrehung x Anzahl der Umdrehungen

#### Linear-Encoder

Standardwert: Gegebene Meßlänge, siehe Typenschild, multipliziert mit 100  
entsprechend der Auflösung von 0,01mm

$$\text{Meßlänge in Schritten} = \frac{\text{Meßlänge}}{\text{Auflösung in mm}}$$

### 6.1.4 Objekt 6003h - Presetwert

Die Presetfunktion wird verwendet, um den Encoderwert auf einen beliebigen Positionswert innerhalb des Bereiches von 0 bis Meßlänge in Schritten — 1 zu setzen. Der Positionswert des ersten Magneten wird auf den Presetwert gesetzt, wenn auf dieses Objekt geschrieben wird.

Die Position des zweiten Magneten wird dabei in Bezug auf den ersten Magneten gesetzt. Voneinander unabhängige Presets müssen über das Objekt 6010h vorgenommen werden.

#### Parameterstruktur

Unsigned32

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

### 6.1.5 Objekt 6004h - Positionswert

Das Objekt 6004h "Positionswert" definiert den Ausgabe-Positionswert für das Kommunikationsobjekt 1800h. Es wird der Positionswert des ersten Magneten ausgegeben.

#### Parameterstruktur

Unsigned32

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

### 6.1.6 Objekt 6005h - Meßschritt-Einstellungen bei Linear-Encodern

Dieser Parameter definiert die eingestellte Schrittweite des Positions- und Geschwindigkeitwertes bei Linear-Encodern.

Index	Subindex	Kommentar	Typ
6005h	0	Anzahl der Einträge	Unsigned8
	1	Meßschritt	Unsigned32
	2	Geschwindigkeit	Unsigned32

Unsigned32

Meßschritt / Geschwindigkeit			
Byte 0	Byte 1	Byte 2	Byte 3
$2^7$ bis $2^0$	$2^{15}$ bis $2^8$	$2^{23}$ bis $2^{16}$	$2^{31}$ bis $2^{24}$

Messschritt: Wert in 0,001  $\mu$ m

z.B.: 0,01 mm = 00 00 27 10 h = 10 000 dez.

Geschwindigkeit: Wert in 0,01 mm / s

z.B. 0,1 mm / s = 0A h = 10 dez.

### 6.1.7 Objekt 6010h - Presetwert Multisensor

Entsprechend zum Objekt 6003h definiert dieses Objekt den Presetwert für Multisensor-Geräte. Im Subindex 00h wird die Anzahl der unterstützten Magneten definiert.

Die Positionswerte in den Subindizes des Objekts 6020h werden dementsprechend auf die Subindizes des Parameters "Presetwert" im Objekt 6010h gesetzt.

Jeder Magnet kann auf eine individuelle Position gesetzt werden, jeweils unabhängig von dem anderen Magneten.

Index	Subindex	Kommentar	Typ
6010H	0	Anzahl Magnete	Unsigned8
	1	Presetwert des 1. Magneten	Unsigned32
	2	Presetwert des 2. Magneten	Unsigned32

#### Parameterstruktur Unsigned32

Presetwert Multisensor			
Byte 0	Byte 1	Byte 2	Byte 3
$2^7$ bis $2^0$	$2^{15}$ bis $2^8$	$2^{23}$ bis $2^{16}$	$2^{31}$ bis $2^{24}$

### 6.1.8 Objekt 6020h - Positionswert Multisensor

Entsprechend zum Objekt 6004h definiert dieses Objekt die Ausgabe-Positionswerte für die Kommunikationsobjekte 1800h und 1801h.

Index	Subindex	Kommentar	Typ
6020H	0	Anzahl Magnete	Unsigned8
	1	Positionswert des 1. Magneten	Unsigned32
	2	Positionswert des 2. Magneten	Unsigned32

#### Parameterstruktur Unsigned32

Positionswert Multisensor			
Byte 0	Byte 1	Byte 2	Byte 3
$2^7$ bis $2^0$	$2^{15}$ bis $2^8$	$2^{23}$ bis $2^{16}$	$2^{31}$ bis $2^{24}$

### 6.1.9 Objekt 6200h - Cyclic-Timer

Definiert den Parameter "Cyclic-Timer". Eine asynchrone Übertragung des Positionswertes wird eingestellt, wenn der Cyclic-Timer auf > 0 programmiert wird. Werte zwischen 1 ms und 65535 ms können ausgewählt werden.

z.B.: 1 ms = 1 h  
256 ms = 100 h

Wenn der Encoder mit dem Kommando NODE-START gestartet wird und der Wert des Cyclic-Timers > 0 ist, überträgt das erste Sende-Prozessdaten-Objekt (Objekt 1800h) die Encoderposition.

## 6.2 Encoder Diagnose

### 6.2.1 Objekt 6500h - Betriebsstatus

Dieses Objekt enthält den Betriebsstatus des Encoders und beinhaltet Informationen über die intern programmierten Parameter.

#### Parameterstruktur

Unsigned16

Bit	Funktion	Bit = 0	Bit = 1
0	Zählrichtung	CW	CCW
1	Inbetriebnahmediagnose	nicht unterstützt	unterstützt
2	Skalierungsfunktion	gesperrt	freigegeben
3 - 11	reserviert		
12- 15	herstellerspezifische Funktionen		

## 6.2.2 Objekt 6501h - Single-Turn Auflösung (Drehgeber), Meßschritt (Linear-Encoder)

Die Single-Turn Auflösung in Objekt 6501h enthält abhängig vom Encodertyp unterschiedliche Inhalte.

### 6.2.2.1 Drehgeber

Das Objekt 6501h für Drehgeber enthält die Anzahl der Meßschritte pro Umdrehung welche als Absolut-Single-Turn Positionswert ausgegeben werden. Die max. Single-Turn Auflösung beträgt  $2^{32}$ .

#### Parameterstruktur

Unsigned32

Single-Turn Auflösung			
Byte 0	Byte 1	Byte 2	Byte 3
$2^7$ bis $2^0$	$2^{15}$ bis $2^8$	$2^{23}$ bis $2^{16}$	$2^{31}$ bis $2^{24}$

Standardwert: 4096 = 1000h Schritte pro Umdrehung (abhängig von der Kapazität, siehe Typenschild), falls keine Skalierung vorgenommen wurde.

Mit Skalierungsfunktion (Objekt 6000h) wird der Wert auf die programmierte Anzahl der Schritte pro Umdrehung gesetzt.

### 6.2.2.2 Linear-Encoder

Für Linear-Encoder zeigt das Objekt 6501h den Meßschritt an, welcher durch den Encoder ausgegeben wird. Der Meßschritt wird in nm ( $0,001\mu\text{m}$ ) angegeben.  
z.B.:  $1\mu\text{m} = 00\ 00\ 03\ \text{E}8\ \text{h}$

#### Parameterstruktur

Unsigned32

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

Standardwert: 2710 h = 10 000 = 0,01mm

### 6.2.3 Objekt 6502h - Anzahl der Umdrehungen

Dieses Objekt beinhaltet die Anzahl der Umdrehungen, welche der Encoder ausgeben kann.

#### **Drehgeber**

Für einen Multi-Turn Encoder ergibt sich aus der Anzahl der Umdrehungen und der Single-Turn Auflösung der Meßbereich, welcher sich nach der unten stehen Formel berechnen läßt. Die max. Anzahl der Umdrehungen ist 65536 (16 Bit).

Meßbereich = Anzahl der Umdrehungen x Single-Turn Auflösung

Standardwert: 4096 = 1000h Umdrehungen (abhängig von der Kapazität, siehe Typenschild), falls keine Skalierung vorgenommen wurde.

Ist die Skalierungsfunktion (Objekt 6000h) freigegeben, ist der Wert die programmierte Anzahl der Umdrehungen.

#### **Linear-Encoder**

Für Linear-Encoder ist der Wert immer "1".

## 6.2.4 Objekt 6503h - Alarmer

Das Objekt 6503h liefert zusätzlich zur "Emergency-Meldung" weitere Alarm-Meldungen. Ein Alarm wird gesetzt, wenn eine Störung im Encoder zum falschen Positionswert führen könnte. Falls ein Alarm auftritt, wird das zugehörige Bit solange auf logisch "High" gesetzt, bis der Alarm gelöscht und der Encoder bereit ist, einen richtigen Positionswert zu liefern.

### Parameterstruktur

Unsigned16

Bit	Funktion	Bit = 0	Bit = 1
0	Positionsfehler	Nein	Ja
1	Inbetriebnahmediagnose	OK	Fehler
2	Reserviert für weitere Verwendung		
3	Reserviert für weitere Verwendung		
4	Reserviert für weitere Verwendung		
5	Reserviert für weitere Verwendung		
6	Reserviert für weitere Verwendung		
7	Reserviert für weitere Verwendung		
8	Reserviert für weitere Verwendung		
9	Reserviert für weitere Verwendung		
10	Reserviert für weitere Verwendung		
11	Reserviert für weitere Verwendung		
12	EE-PROM-Fehler	OK	Fehler
13	Parameterfehler	OK	Fehler
14	herstellerspezifische Funktionen		
15	herstellerspezifische Funktionen		

#### Positionsfehler:

Das Bit wird gesetzt, wenn ein Drehgeber eine Störung des Systems erkennt, oder falls ein Linear-Encoder kein Magnet erkannt hat.

#### Inbetriebnahmediagnose:

Nicht unterstützt

#### EE-PROM-Fehler:

Der Encoder erkannte eine falsche Checksumme im EEPROM-Bereich, oder ein Schreibvorgang in das EEPROM konnte nicht erfolgreich abgeschlossen werden.

#### Parameterfehler:

Der Wert eines übertragenen Parameters ist außerhalb des Bereiches. Die Überprüfung wird vorgenommen, wenn das Kommando "Parameter abspeichern" empfangen wurde.

### 6.2.5 Objekt 6504h - Unterstützte Alarmer

Das Objekt 6504h beinhaltet Informationen über die Alarmer, die durch den Encoder unterstützt werden.

#### Parameterstruktur

Unsigned16

Bit	Funktion	Bit = 0	Bit = 1
0	Positionsfehler	Nein	Ja
1	Inbetriebnahmediagnose	Nein	Ja
2	Reserviert für weitere Verwendung		
3	Reserviert für weitere Verwendung		
4	Reserviert für weitere Verwendung		
5	Reserviert für weitere Verwendung		
6	Reserviert für weitere Verwendung		
7	Reserviert für weitere Verwendung		
8	Reserviert für weitere Verwendung		
9	Reserviert für weitere Verwendung		
10	Reserviert für weitere Verwendung		
11	Reserviert für weitere Verwendung		
12	EE-PROM-Fehler	Nein	Ja
13	Parameterfehler	Nein	Ja
14	herstellerspezifische Funktionen		
15	herstellerspezifische Funktionen		

### 6.2.6 Objekt 6505h - Warnungen

Dieses Objekt wird nicht unterstützt.  
Bei Lesezugriff ist der Wert immer "0".

### 6.2.7 Objekt 6506h - Unterstützte Warnungen

Dieses Objekt wird nicht unterstützt.  
Bei Lesezugriff ist der Wert immer "0".

## 6.2.8 Objekt 6507h - Profil- und Softwareversion

Dieses Objekt enthält in den ersten 16 Bits die implementierte Profilversion des Encoders. Sie ist kombiniert mit einer Revisionsnummer und einem Index.

z.B.: Profilversion: 1.40  
 Binärcode: 0000 0001 0100 0000  
 Hexadezimal: 1 40

Die zweiten 16 Bits enthalten die implementierte Softwareversion des Encoders. Nur die letzten 4 Ziffern sind verfügbar.

z.B.: Softwareversion: 5022.01  
 Binärcode: 0020 0020 0000 0001  
 Hexadezimal: 22 01

### Parameterstruktur

Unsigned32

Profilversion		Softwareversion	
Byte 0	Byte 1	Byte 2	Byte 3
$2^7$ bis $2^0$	$2^{15}$ bis $2^8$	$2^7$ bis $2^0$	$2^{15}$ bis $2^8$

## 6.2.9 Objekt 6508h - Betriebszeit

Dieses Objekt wird nicht unterstützt.

Die Betriebszeit-Funktion wird nicht verwendet, der Betriebszeitwert wird auf den Maximalwert gesetzt ( FF FF FF FF h).

## 6.2.10 Objekt 6509h - Offsetwert

Dieses Objekt enthält den Offsetwert, der durch die Preset-Funktion berechnet wird. Der Offsetwert wird gespeichert und kann vom Encoder gelesen werden.

## 6.2.11 Objekt 650Ah - Hersteller-Offsetwert

Dieses Objekt wird nicht unterstützt.

Bei Lesezugriff ist der Offsetwert "0".

## 6.2.12 Objekt 650Bh - Serien-Nummer

Dieses Objekt wird nicht unterstützt.

Der Parameter Serien-Nummer wird nicht verwendet, der Wert wird auf den Maximalwert FF FF FF FF h gesetzt.

### 7 Emergency-Meldung

Emergency-Meldungen werden beim Auftreten einer geräteinternen Störung ausgelöst und werden von dem betreffenden Anwendungsgerät an die anderen Geräte mit höchster Priorität übertragen.

Emergency-Meldung								
Byte	0	1	2	3	4	5	6	7
Inhalt	Emergency-Fehlercode		Fehler-Register (Objekt 1001h)	0	0	0	0	0

#### **COB-Identifizier = 080h + Node-ID**

Wenn der Encoder einen internen Fehler erkennt, wird eine Emergency-Meldung mit dem Fehlercode des Objekts 1003h (Vordefiniertes Fehlerfeld) und dem Fehler-Register (Objekt 1001h) übertragen. Zusätzlich zum Emergency-Objekt wird im Alarm-Objekt 6503H das zugehörige Bit gesetzt.

Wenn der Fehler nicht mehr vorhanden ist, überträgt der Encoder eine Emergency-Meldung mit dem Fehlercode "0" (Reset Fehler / kein Fehler) und Fehler-Register "0".

## 8 Inbetriebnahme des Encoders am CAN-Bus

Bevor der Anschluß des Encoders am Bus erfolgt, muß die Baudrate und die Node-ID mit den Hardwareschaltern eingestellt werden. Die Schalter werden nur im Einschaltmoment der Betriebsspannung neu gelesen.

Nach dem Einschalten der Betriebsspannung und Beendigung der Initialisierung geht der Encoder in den Vor-Betriebszustand und wartet auf Kommandos. Falls der Encoder einen internen Fehler erkennt, wird eine Emergency-Meldung mit dem Fehlercode übertragen.

Während dieses Zustandes ist eine Parametrierung über das SDO (z.B. bei Benutzung eines Konfigurations-Tool's) möglich.

### 8.1 Übertragung des Encoder-Positionswertes

Bevor die Encoderposition übertragen werden kann, muß der Encoder mit dem "Node-Start"-Kommando gestartet werden.

#### Node-Start Protokoll

COB-Identifizier = 0

Byte 0	Byte 1
1	Node-ID

Das Node-Start Kommando mit der Node-ID des Encoders (Slave) startet nur dieses Gerät.

Das Node-Start Kommando mit der **Node-ID = 0** startet alle Slaves die am Bus angeschlossen sind.

Nach dem Node-Start Kommando überträgt der Encoder den Positionswert einmal mit der COB-ID des Objekts 1800h.

Jetzt kann der Positionswert auf verschiedene Arten übertragen werden:

#### Asynchron-Übertragung

Die Sende-Prozessdaten-Objekte bei denen die Übertragungsart auf 254 eingestellt ist, übertragen die Positionswerte. Der Timerwert wird definiert durch den Wert des Cyclic-Timers (Objekt 6200h). Diese Übertragung startet automatisch nach dem Kommando Node-Start und der Wert des Cyclic-Timers ist > 0.

### Synchron-Übertragung

Die Sende-Prozessdaten-Objekte bei denen die Übertragungsart auf 1 eingestellt ist, übertragen die Positionswerte nach einer Anforderung (Remote / Sync).

Der Encoder empfängt ein Remote-Frame mit der COB-ID des PDO's.

Der Encoder empfängt ein SYNC-Telegramm mit der COB-ID (Standardwert 080h), definiert in Objekt 1005h. Alle Slaves mit dieser SYNC-COB-ID und PDO-Übertragungsart = 1 übertragen den Positionswert.

Mit dem Kommando "Node-Stop" wird die Übertragung beendet.

### Node-Stop Protokoll

#### COB-Identifizier = 0

Byte 0	Byte 1
2	Node-ID

Das Node-Stop Kommando mit der Node-ID des Encoders (Slave) stoppt nur dieses Gerät.

Das Node-Stop Kommando mit der **Node-ID = 0** stoppt alle Slaves die am Bus angeschlossen sind.

## 8.2 Lese-/Schreib- Service-Daten-Objekt

Die Übertragung des Service-Daten-Objekt (SDO) geschieht über das CMS "Multiplexed-Domain" Protokoll (CIA DS202/2).

### 8.2.1 Lese SDO:

("Domain Upload" einleiten)

#### Anforderungs-Protokoll-Format:

#### COB-Identifizier = 600h + Node-ID

Lese SDO's								
Byte	0	1	2	3	4	5	6	7
Inhalt	Code	Index		Sub-index	Daten 0	Daten 1	Daten 2	Daten 3
	40h	Low	High	Byte	0	0	0	0

Das "Lese-SDO" Telegramm muß an den Slave gesendet werden.

Der Slave antwortet mit folgendem Telegramm:

**Antwort-Protokoll-Format:**

**COB-Identifizier = 580h + Node-ID**

Lese SDO's								
Byte	0	1	2	3	4	5	6	7
Inhalt	Code	Index		Sub-index	Daten 0	Daten 1	Daten 2	Daten 3
	4xh	Low	High	Byte	Daten	Daten	Daten	Daten

**Format-Byte 0:**

MSB				LSB			
7	6	5	4	3	2	1	0
0	1	0	0	n		1	1

n = Anzahl der Datenbytes ( Bytes 4-7), welche keine Daten beinhalten.

Wenn nur 1 Datenbyte (Daten 0) Daten enthält, ist der Wert von Byte 0 = "4Fh".

Ist Byte 0 = 80h, wird die Übertragung abgebrochen.

**8.2.2 Schreibe SDO:**

("Domain Download" einleiten)

**Anforderungs-Protokoll-Format:**

**COB-Identifizier = 600h + Node-ID**

Schreibe SDO's								
Byte	0	1	2	3	4	5	6	7
Inhalt	Code	Index		Sub-index	Daten 0	Daten 1	Daten 2	Daten 3
	2xh	Low	High	Byte	0	0	0	0

### Format-Byte 0:

MSB				LSB			
7	6	5	4	3	2	1	0
0	0	1	0	n		1	1

n = Anzahl der Datenbytes (Bytes 4-7), welche keine Daten beinhalten.

Wenn nur 1 Datenbyte (Daten 0) Daten enthält, ist der Wert von Byte 0 = "2Fh".

Das "Schreibe-SDO" Telegramm muß an den Slave gesendet werden.

Der Slave antwortet mit folgendem Telegramm:

### Antwort-Protokoll-Format:

**COB-Identifizier = 580h + Node-ID**

Lese SDO's								
Byte	0	1	2	3	4	5	6	7
Inhalt	Code	Index		Sub-index	Daten 0	Daten 1	Daten 2	Daten 3
	60h	Low	High	Byte	0	0	0	0

Ist Byte 0 = 80h, wird die Übertragung abgebrochen.