

# Drehgeber

## Baureihe:

- 362

- 582

- 802

- 1102

- \_Zusätzliche Sicherheitshinweise
- \_Installation
- \_Inbetriebnahme
- \_Parametrierung
- \_Fehlerursachen und Abhilfen

---

## **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. Drittenanwendungen 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:	03/06/2024
Dokument-/Rev.-Nr.:	TR-ECE-BA-DGB-0147 v07
Dateiname:	TR-ECE-BA-DGB-0147-07.docx
Verfasser:	STB

---

### **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® und CiA® sind eingetragene Gemeinschaftsmarken der CAN in Automation e.V.

---

# Inhaltsverzeichnis

<b>Änderungs-Index .....</b>	<b>7</b>
<b>1 Allgemeines .....</b>	<b>8</b>
1.1 Geltungsbereich.....	8
1.2 Referenzen .....	9
1.3 Verwendete Abkürzungen / Begriffe .....	10
<b>2 Zusätzliche Sicherheitshinweise .....</b>	<b>11</b>
2.1 Symbol- und Hinweis-Definition.....	11
2.2 Ergänzende Hinweise zur bestimmungsgemäßen Verwendung.....	11
<b>3 Optionale Schnittstellenvarianten .....</b>	<b>12</b>
<b>4 CANopen Informationen .....</b>	<b>13</b>
4.1 CANopen – Kommunikationsprofil.....	14
4.2 Prozess- und Service-Daten-Objekte .....	15
4.3 Objektverzeichnis (Object Dictionary) .....	16
4.4 CANopen Default Identifier, COB-ID .....	16
4.5 Übertragung von SDO Nachrichten .....	17
4.5.1 SDO-Nachrichtenformat.....	17
4.5.2 Lese SDO .....	19
4.5.3 Schreibe SDO .....	20
4.6 Netzwerkmanagement, NMT .....	21
4.6.1 Netzwerkmanagement-Dienste .....	22
4.6.1.1 NMT-Dienste zur Gerätekontrolle .....	22
4.6.1.2 NMT-Dienste zur Verbindungsüberwachung .....	23
4.7 Layer setting services (LSS) und Protokolle.....	24
4.7.1 Finite state automaton, FSA .....	25
4.7.2 Übertragung von LSS-Diensten.....	26
4.7.2.1 LSS-Nachrichtenformat.....	26
4.7.3 Switch mode Protokolle .....	27
4.7.3.1 Switch state global Protokoll .....	27
4.7.3.2 Switch state selective Protokoll.....	27
4.7.4 Configuration Protokolle.....	28
4.7.4.1 Configure Node-ID Protokoll .....	28
4.7.4.2 Configure bit timing parameters Protokoll .....	29
4.7.4.3 Activate bit timing parameters Protokoll.....	30
4.7.4.4 Store configuration Protokoll.....	30
4.7.5 Inquire LSS-Address Protokolle.....	31
4.7.5.1 Inquire identity Vendor-ID Protokoll .....	31
4.7.5.2 Inquire identity Product-Code Protokoll.....	31
4.7.5.3 Inquire identity Revision-Number Protokoll .....	32
4.7.5.4 Inquire identity Serial-Number Protokoll.....	32
4.7.6 Inquire Node-ID Protokoll.....	33
4.7.7 Identification Protokolle.....	34
4.7.7.1 LSS identify remote slave Protokoll .....	34
4.7.7.2 LSS identify slave Protokoll.....	34
4.7.7.3 LSS identify non-configured remote slave Protokoll .....	35
4.7.7.4 LSS identify non-configured slave Protokoll.....	35
4.8 Geräteprofil .....	36

<b>5 Installation / Inbetriebnahmevorbereitung.....</b>	<b>37</b>
5.1 Anschluss – Hinweise.....	38
5.2 Bus-Terminierung .....	38
5.3 Einschalten der Versorgungsspannung.....	38
5.4 Einstellen der Node-ID und Baudrate .....	39
5.4.1 Einstellung mittels DIP-Schalter .....	39
5.4.2 Einstellung mittels LSS-Dienste.....	39
5.4.2.1 Konfiguration der Node-ID .....	39
5.4.2.2 Konfiguration der Baudrate .....	40
5.4.3 Einstellung mittels SDOs .....	40
<b>6 Inbetriebnahme.....</b>	<b>41</b>
6.1 CAN – Schnittstelle .....	41
6.1.1 EDS-Datei .....	41
6.1.2 Bus-Statusanzeige.....	41
<b>7 Kommunikations-Profil.....</b>	<b>42</b>
7.1 Aufbau der Kommunikationsparameter, 1800h-1801h.....	42
7.2 Aufbau der Mappingparameter, 1A00h-1A01h.....	43
7.2.1 Ändern der Mappingeinstellung .....	44
7.3 Übertragung des Mess-System-Positionswertes.....	44
7.3.1 Erstes Sende-Prozessdaten-Objekt .....	45
7.3.2 Zweites Sende-Prozessdaten-Objekt .....	45
<b>8 Kommunikationsspezifische Standard-Objekte (CiA DS-301).....</b>	<b>46</b>
8.1 Objekt 1000h: Gerätetyp.....	47
8.2 Objekt 1001h: Fehlerregister .....	47
8.3 Objekt 1002h: Hersteller-Status-Register.....	48
8.4 Objekt 1003h: Vordefiniertes Fehlerfeld .....	48
8.5 Objekt 1005h: COB-ID SYNC Nachricht .....	49
8.6 Objekt 1008h: Hersteller Gerätenamen .....	49
8.7 Objekt 1009h: Hersteller Hardwareversion.....	49
8.8 Objekt 100Ah: Hersteller Softwareversion.....	49
8.9 Objekt 100Ch: Guard-Time (Überwachungszeit) .....	50
8.10 Objekt 100Dh: Life-Time-Faktor (Zeitdauer-Faktor) .....	50
8.11 Objekt 1010h: Parameter abspeichern.....	51
8.12 Objekt 1011h: Wiederherstellung der Parameter-Standardwerte .....	52
8.13 Objekt 1014h: COB-ID EMERGENCY (EMCY).....	53
8.14 Objekt 1016h: Consumer Heartbeat Time.....	53
8.15 Objekt 1017h: Producer Heartbeat Time.....	54
8.16 Objekt 1018h: Identity Objekt .....	54
8.17 Objekt 1021h: EDS abspeichern .....	55
8.18 Objekt 1022h: EDS Speicherformat .....	55
8.19 Objekt 1029h: Verhalten im Fehlerfall .....	55

8.20 Firmware-Update .....	56
8.20.1 Objekt 1F50h: Programmdaten .....	56
8.20.2 Objekt 1F51h: Programmsteuerung .....	56
8.20.3 Objekt 1F56h: Programm Software Identifikation .....	57
8.20.4 Objekt 1F57h: Programm Status .....	57
8.21 Objekt 1F80h: NMT Autostart .....	58
<b>9 Parametrierung .....</b>	<b>59</b>
9.1 Objekt 2000h: Mode-Umschaltung TR / CiA DS-406 .....	61
9.2 TR - Mode .....	62
9.2.1 Objekt 2001h: TR-Betriebsparameter, Zählrichtung .....	62
9.2.2 Skalierungsparameter .....	62
9.2.2.1 Objekt 2002h: TR-Gesamtmesslänge in Schritten .....	63
9.2.2.2 Objekt 2003h - 2004h: TR-Umdrehungen Zähler / Nenner .....	64
9.2.3 Objekt 2005h: Geschwindigkeitsauflösung .....	67
9.2.4 Objekt 2006h: Zusätzliche Parameter/Kommandos (gerätespezifisch) .....	67
9.2.5 Objekt 2007h: Geschwindigkeitsfaktor .....	68
9.2.6 Objekt 2008h: Geschwindigkeitsintegrationszeit .....	68
9.2.7 Objekt 200Ah: Geschwindigkeit INT32 .....	69
9.2.8 Objekt 2010h: Zustandsänderung - Faktor .....	69
9.2.9 Objekt 2011h: Zustandsänderung - Hysterese .....	70
9.2.10 Objekt 2100h: COB-ID für Boot-Up Nachricht .....	70
9.2.11 Objekt 2101h: Senden von PDO bei Node-Start .....	71
9.2.12 Objekt 2110h: LSS Node-ID .....	71
9.2.13 Objekt 2111h: LSS-BitTiming .....	72
9.2.14 Objekt 2200h: DMF abspeichern .....	72
9.2.15 SSI-Ausgabe .....	72
9.2.15.1 Objekt 2500h: SSI-Ausgang verfügbar .....	72
9.2.15.2 Objekt 2501h: SSI-Ausgangsdaten .....	73
9.2.15.3 Objekt 2502h: SSI-Ausgabecode .....	73
9.2.15.4 Objekt 2503h: SSI-Anzahl Datenbits .....	73
9.2.15.5 Objekt 2504h: SSI-Monozeit .....	74
9.2.16 Presetfunktion .....	74
9.2.16.1 Objekt 2510h: Preset-Eingänge verfügbar .....	74
9.2.16.2 Objekt 2511h: Presetwert 1 .....	74
9.2.16.3 Objekt 2512h: Presetwert 2 .....	75
9.2.17 Inkremental-Ausgabe .....	75
9.2.17.1 Objekt 2520h: Inkremental-Ausgang verfügbar .....	75
9.2.17.2 Objekt 2521h: Inkremental-Anzahl Impulse .....	75
9.2.17.3 Objekt 2522h: Inkremental-Phasenlage .....	76
9.2.17.4 Objekt 2523h: Inkremental-K0 Verhalten .....	76
9.2.17.5 Objekt 2524h: Inkremental-Pegel .....	77
9.2.17.6 Objekt 2525h: Inkremental-K0 Länge .....	77
9.2.17.7 Objekt 2526h: Inkremental-K0 setzen .....	77
9.2.18 Externe-Anzeige .....	78
9.2.18.1 Objekt 2530h: Anzeige verfügbar .....	78
9.2.18.2 Objekte 2531h ... 2534h: Anzeige-Schalterfunktion 1, 2, 3 und 4 .....	78
9.2.18.3 Objekt 2535h: Anzeige-Steuerung .....	78
9.2.18.4 Objekt 2536h: Anzeige-Daten numerisch .....	79
9.2.18.5 Objekt 2537h: Anzeige-Daten ASCII .....	79
9.2.18.6 Objekt 2538h: Anzeige-Status .....	79
9.2.19 TR-Produktion .....	79
9.3 CiA DS-406 - Mode .....	80
9.3.1 Objekt 6000h: Betriebsparameter .....	80
9.3.2 Skalierungsparameter .....	80
9.3.2.1 Objekt 6001h: Mess-Schritte pro Umdrehung .....	81
9.3.2.2 Objekt 6002h: Gesamt Messlänge in Schritten .....	81

---

9.3.3 Objekt 6003h: Presetwert .....	83
9.3.4 Objekt 6004h: Positionswert .....	83
9.3.5 Objekt 6030h: Geschwindigkeit .....	84
9.3.6 Objekt 6200h: Cyclic-Timer.....	84
9.3.7 Objekt 6400h: Bereichs-Statusregister .....	85
9.3.8 Objekt 6401h: Arbeitsbereich unterer Grenzwert .....	85
9.3.9 Objekt 6402h: Arbeitsbereich oberer Grenzwert .....	85
9.4 Mess-System Diagnose .....	86
9.4.1 Objekt 6500h: Betriebsstatus.....	86
9.4.2 Objekt 6501h: Single-Turn Auflösung.....	86
9.4.3 Objekt 6502h: Anzahl der Umdrehungen .....	87
9.4.4 Objekt 6503h: Alarmer .....	88
9.4.5 Objekt 6504h: Unterstützte Alarmer.....	89
9.4.6 Objekt 6505h: Warnungen .....	90
9.4.7 Objekt 6506h: Unterstützte Warnungen .....	91
9.4.8 Objekt 6507h: Profil- und Softwareversion .....	92
9.4.9 Objekt 6508h: Betriebsdauer .....	92
9.4.10 Objekt 6509h: Offsetwert .....	93
9.4.11 Objekt 650Ah: Hersteller-Offsetwert .....	93
9.4.12 Objekt 650Bh: Serien-Nummer.....	93
<b>10 Emergency-Meldung .....</b>	<b>94</b>
<b>11 Fehlerursachen und Abhilfen.....</b>	<b>95</b>
11.1 Optische Anzeigen.....	95
11.1.1 Mess-System mit zwei Status-LEDs.....	95
11.1.2 Mess-System mit einer Status-LED .....	97
11.2 SDO-Fehlercodes .....	98
11.3 Emergency-Fehlercodes.....	99
11.3.1 Objekt 1001h: Fehlerregister .....	99
11.3.2 Objekt 1003h: Vordefiniertes Fehlerfeld, Bits 0 – 15 .....	100
11.4 Alarm-Meldungen .....	100
11.5 Sonstige Störungen .....	101

## Änderungs-Index

Änderung	Datum	Index
Erstausgabe	22.05.2019	00
Zweitschnittstelle SSI erweitert	31.10.2019	01
Zweitschnittstelle Inkremental erweitert	06.03.2020	02
Objekt 6003h: Presetwert angepasst	06.11.2020	03
Kapitel „Sonstige Störungen“ keine paarig verdrehten Adern für Versorgung	27.01.2022	04
Objekte 200Ah, 20210h und 2011h ergänzt	31.08.2022	05
- Geltungsbereich um Baureihe 362 ergänzt - Fehlercode 5000h ergänzt - Kapitel „Mess-System mit einer Status-LED“ ergänzt	05.07.2023	06
Kap.: 10 „Emergency-Meldung“ Byte 3 bis 6 angepasst	06.03.2024	07

# 1 Allgemeines

Das vorliegende Benutzerhandbuch beinhaltet folgende Themen:

- Ergänzende Sicherheitshinweise zu den bereits in der Montageanleitung definierten grundlegenden Sicherheitshinweisen
- Installation
- Inbetriebnahme
- Parametrierung
- Fehlerursachen und Abhilfen

Da die Dokumentation modular aufgebaut ist, stellt dieses Benutzerhandbuch eine Ergänzung zu anderen Dokumentationen wie z.B. Produktdatenblätter, Maßzeichnungen, Prospekte und der Montageanleitung etc. dar.

Das Benutzerhandbuch kann kundenspezifisch im Lieferumfang enthalten sein, oder kann auch separat angefordert werden.

## 1.1 Geltungsbereich

Dieses Benutzerhandbuch gilt ausschließlich für Mess-System-Baureihen gemäß nachfolgendem Typenschlüssel mit **CANopen** und optionaler Zweitschnittstelle:

- 362
- 582
- 802
- 1102

Die Produkte sind durch aufgeklebte Typenschilder gekennzeichnet und sind Bestandteil einer Anlage.

Es gelten somit zusammen folgende Dokumentationen:

- siehe Kapitel „Mitgeltende Dokumente“ in der Montageanleitung
  - Baureihe 362: [www.tr-electronic.de/f/TR-ECE-BA-DGB-0108](http://www.tr-electronic.de/f/TR-ECE-BA-DGB-0108)
  - Baureihe 582: [www.tr-electronic.de/f/TR-ECE-BA-DGB-0035](http://www.tr-electronic.de/f/TR-ECE-BA-DGB-0035)
  - Baureihe 802: [www.tr-electronic.de/f/TR-ECE-BA-DGB-0075](http://www.tr-electronic.de/f/TR-ECE-BA-DGB-0075)
  - Baureihe 1102: [www.tr-electronic.de/f/TR-ECE-BA-DGB-0081](http://www.tr-electronic.de/f/TR-ECE-BA-DGB-0081)
- Produktdatenblätter
  - Baureihe 362: [www.tr-electronic.de/s/S025376](http://www.tr-electronic.de/s/S025376)
  - Baureihe 582: [www.tr-electronic.de/s/S020796](http://www.tr-electronic.de/s/S020796)
  - Baureihe 802: [www.tr-electronic.de/s/S020797](http://www.tr-electronic.de/s/S020797)
  - Baureihe 1102: [www.tr-electronic.de/s/S020798](http://www.tr-electronic.de/s/S020798)



## 1.2 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-206 V1.1, Empfohlene Namenskonventionen für die Schichten, Februar 1996
12.	CiA DS-207 V1.1, Namenskonventionen der Verarbeitungsschichten, Februar 1996
13.	CiA DS-301 V4.2, CANopen Kommunikationsprofil auf CAL basierend, Dezember 2007
14.	CiA DS-302 V4.1, Zusätzliche Application Layer Funktionen, Februar 2009
15.	CiA DS-303-3 V1.3, Indicator Spezifikation, August 2006
16.	CiA DS-305 V2.2.5, Layer Setting Services (LSS) und Protokolle, November 2010
17.	CiA DS-406 V3.2, CANopen Profil für Encoder, Dezember 2006

### 1.3 Verwendete Abkürzungen / Begriffe

EMV	<b>E</b> lektro- <b>M</b> agnetische- <b>V</b> erträglichkeit
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.
EDS	<b>E</b> lectronic- <b>D</b> ata- <b>S</b> heet (elektronisches Datenblatt)
FSA	Finite state automata. Statusmaschine zur Steuerung von LSS-Diensten
LSS	Layer Setting Services. Dienste und Protokolle für die Konfiguration der Node-ID und Baudrate über das CAN Netzwerk.
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.
SSI	<b>S</b> ynchron- <b>S</b> eriell- <b>I</b> nterface

## 2 Zusätzliche Sicherheitshinweise

### 2.1 Symbol- und Hinweis-Definition



bedeutet, dass Tod oder schwere Körperverletzung eintreten kann, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.

---



bedeutet, dass eine leichte Körperverletzung eintreten kann, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.

---

---

**ACHTUNG**

bedeutet, dass ein Sachschaden eintreten kann, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.

---



bezeichnet wichtige Informationen bzw. Merkmale und Anwendungstipps des verwendeten Produkts.

---

### 2.2 Ergänzende Hinweise zur bestimmungsgemäßen Verwendung

Das Mess-System ist ausgelegt für den Betrieb an CANopen Netzwerken nach dem internationalen Standard ISO/DIS 11898 und 11519-1 bis max. 1 MBit/s. Das Profil entspricht dem **"CANopen Device Profile für Encoder CiA DS-406 V2.0A"**.

Die technischen Richtlinien zum Aufbau des CANopen Netzwerks der CAN-Nutzerorganisation CiA sind für einen sicheren Betrieb zwingend einzuhalten.

### 3 Optionale Schnittstellenvarianten

Bei Schnittstellenvarianten variiert der Funktionsumfang und die Anschlusstechnik. Es dürfen nur die gerätespezifischen Datenblätter, Steckerbelegungen und technischen Zeichnungen verwendet werden.

Es gelten nur die Funktionen, Parameter und Optionen aus diesem Benutzerhandbuch, die auch vom Mess-System unterstützt werden. Die optionalen Funktionalitäten sind an entsprechender Stelle als „optional“ gekennzeichnet.

Welche Optionen durch das Mess-System unterstützt werden, kann durch folgende Punkte abgeleitet werden:

- Ausführung der Steckerbelegung
- Entsprechende Angaben auf dem Typenschild
- Funktionsumfang der dazugehörigen EDS-Datei
- Firmware-Nr.
- Vereinbarung zwischen TR-Electronic und dem Kunden

## 4 CANopen Informationen

CANopen wurde von der CiA entwickelt und ist seit Ende 2002 als europäische Norm EN 50325-4 standardisiert.

CANopen verwendet als Übertragungstechnik die Schichten 1 und 2 des ursprünglich für den Einsatz im Automobil entwickelten CAN-Standards (ISO 11898-2). Diese werden in der Automatisierungstechnik durch die Empfehlungen des CiA Industrieverbandes hinsichtlich der Steckerbelegung, Übertragungsraten erweitert. Im Bereich der Anwendungsschicht hat CiA den Standard CAL (CAN Application Layer) hervorgebracht.

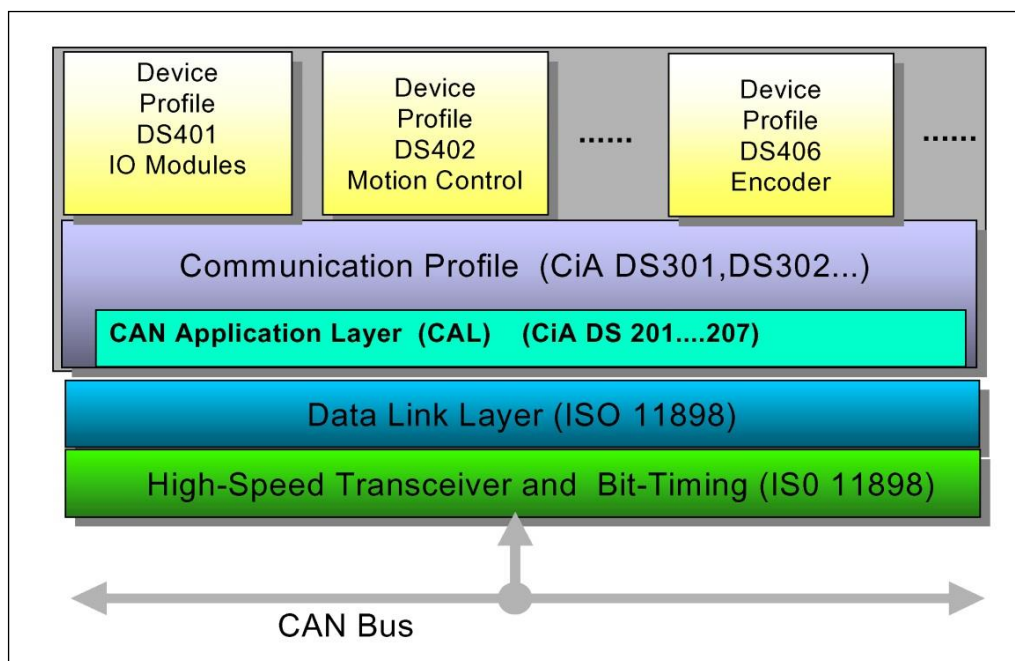


Abbildung 1: CANopen eingeordnet im ISO/OSI-Schichtenmodell

Bei CANopen wurde zunächst das Kommunikationsprofil sowie eine "Bauanleitung" für Geräteprofile entwickelt, in der mit der Struktur des Objektverzeichnisses und den allgemeinen Kodierungsregeln der gemeinsame Nenner aller Geräteprofile definiert ist.

## 4.1 CANopen – Kommunikationsprofil

Das CANopen Kommunikationsprofil (dokumentiert in CiA DS-301) regelt wie die Geräte Daten miteinander austauschen. Hierbei werden Echtzeitdaten (z.B. Positionswert) und Parameterdaten (z.B. Zählrichtung) unterschieden. CANopen ordnet diesen, vom Charakter her völlig unterschiedlichen Datenarten, jeweils passende Kommunikationselemente zu.

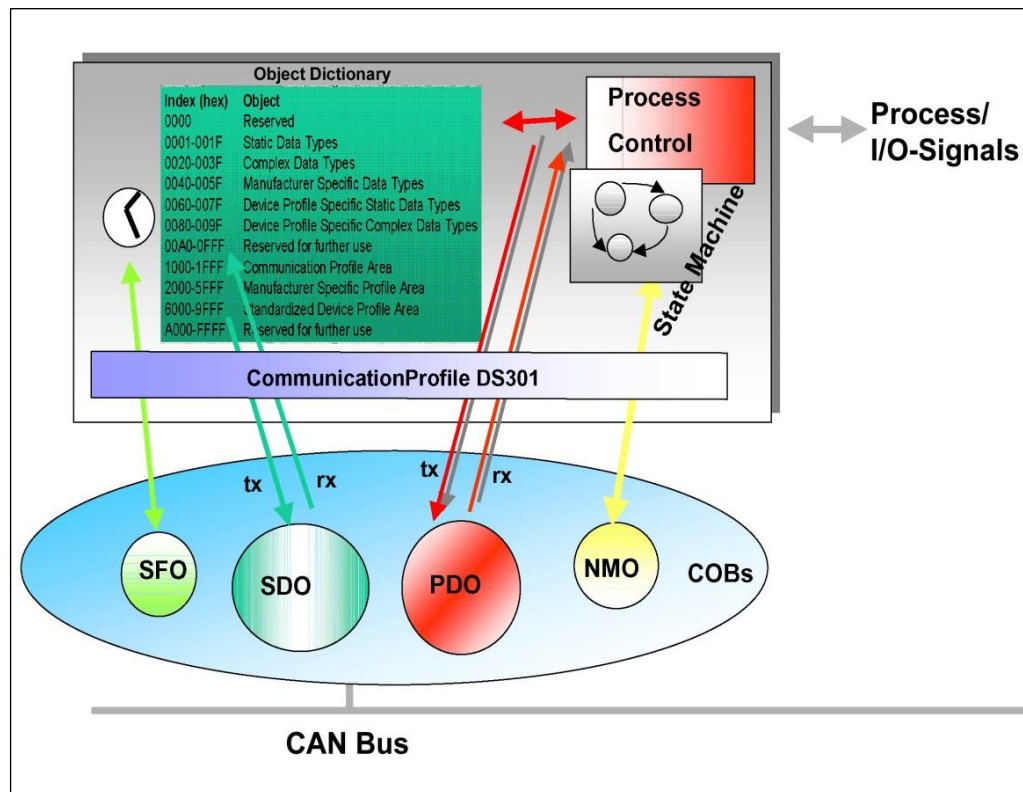


Abbildung 2: Kommunikationsprofil

### Special Function Object (SFO)

- Synchronization (SYNC)
- Emergency (EMCY) Protokoll

### Network Management Object (NMO)

z.B.

- Life / Node-Guarding
- Boot-Up,...
- Error Control Protokoll

## 4.2 Prozess- und Service-Daten-Objekte

### Prozess-Daten-Objekt (PDO)

Prozess-Daten-Objekte managen den Prozessdatenaustausch, z.B. die zyklische Übertragung des Positionswertes.

Der Prozessdatenaustausch mit den CANopen PDOs ist "CAN pur", also ohne Protokoll-Overhead. Die Broadcast-Eigenschaften von CAN bleiben voll erhalten. Eine Nachricht kann von allen Teilnehmern gleichzeitig empfangen und ausgewertet werden.

Vom Mess-System werden die beiden Sende-Prozess-Daten-Objekte 1800h für asynchrone (ereignisgesteuert) Positionsübertragung und 1801h für die synchrone (auf Anforderung) Positionsübertragung verwendet.

### Service-Daten-Objekt (SDO)

Service-Daten-Objekte managen den Parameterdatenaustausch, z.B. das azyklische Ausführen der Presetfunktion.

Für Parameterdaten beliebiger Größe steht mit dem SDO ein leistungsfähiger Kommunikationsmechanismus zur Verfügung. Hierfür wird zwischen dem Konfigurationsmaster und den angeschlossenen Geräten ein Servicedatenkanal für Parameterkommunikation ausgebildet. Die Geräteparameter können mit einem einzigen Telegramm-Handshake ins Objektverzeichnis der Geräte geschrieben werden bzw. aus diesem ausgelesen werden.

### Wichtige Merkmale von SDO und PDO

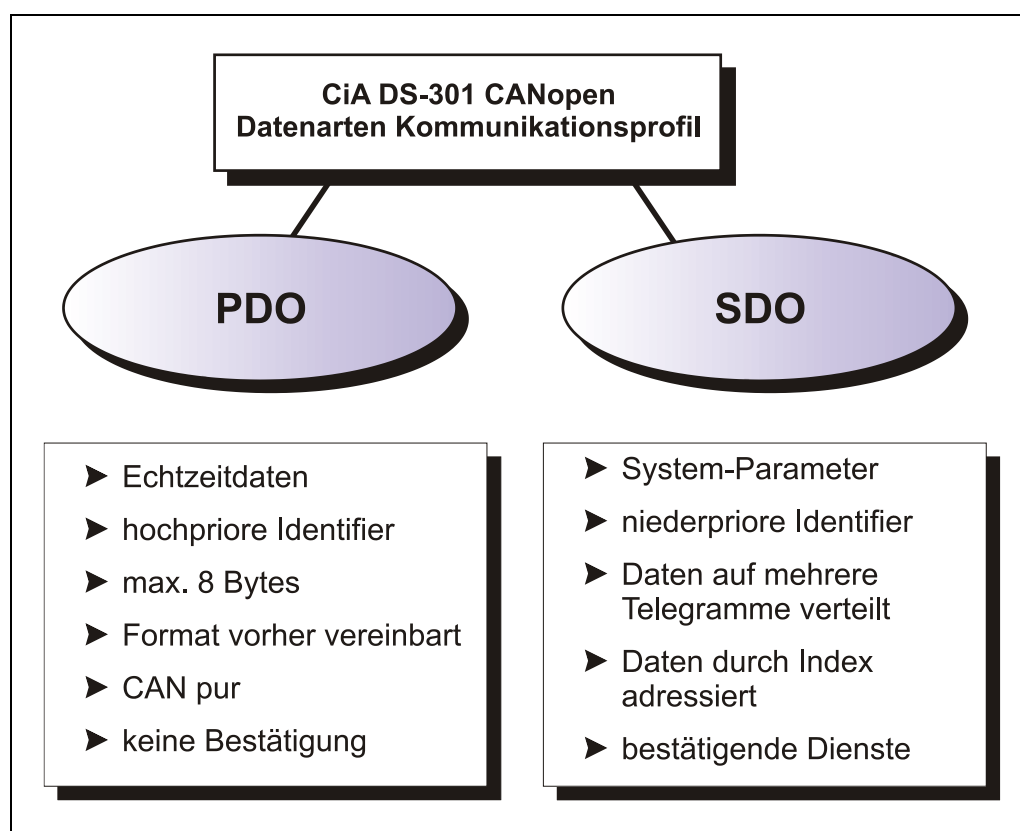


Abbildung 3: Gegenüberstellung von PDO/SDO-Eigenschaften

### 4.3 Objektverzeichnis (Object Dictionary)

Das Objektverzeichnis strukturiert die Daten eines CANopen- Gerätes in einer übersichtlichen tabellarischen Anordnung. Es enthält sowohl sämtliche Geräteparameter als auch alle aktuellen Prozessdaten, die damit auch über das SDO zugänglich sind.

Index	Object	
0000 <sub>h</sub>	unbenutzt	Standard für alle Geräte
0001 <sub>h</sub> - 025F <sub>h</sub>	Datentyp Definitionen	
0260 <sub>h</sub> - 0FFF <sub>h</sub>	Reserviert	
1000 <sub>h</sub> - 1FFF <sub>h</sub>	Kommunikations-Profilbereich	
2000 <sub>h</sub> - 5FFF <sub>h</sub>	Herstellerspezifischer-Profilbereich	Geräte-spezifisch
6000 <sub>h</sub> - 9FFF <sub>h</sub>	Standardisierter-Geräte-Profilbereich	
A000 <sub>h</sub> - BFFF <sub>h</sub>	Standardisierter-Schnittstellen-Profilbereich	
C000 <sub>h</sub> - FFFF <sub>h</sub>	Reserviert	

Abbildung 4: Aufbau des Objektverzeichnisses

### 4.4 CANopen Default Identifier, COB-ID

CANopen-Geräte können ohne Konfiguration in ein CANopen-Netzwerk eingesetzt werden. Lediglich die Einstellung einer Busadresse und der Baudrate ist erforderlich. Aus dieser Knotenadresse leitet sich die Identifizierungsordnung für die Kommunikationskanäle ab.

**COB-Identifier = Funktions-Code + Node-ID**

10											0
1	2	3	4	1	2	3	4	5	6	7	
Funktions-Code				Node-ID = Adressschalter-Einstellung + 1 oder LSS-Einstellung							

#### Beispiele

Objekt	Funktions-Code	COB-ID	Index Kommunikations-Parameter
NMT	0000bin	0	–
SYNC	0001bin	80h	1005
PDO1 (tx)	0011bin	181h – 1FFh	1800h



## 4.5 Übertragung von SDO Nachrichten

Die Übertragung von SDO Nachrichten geschieht über das CMS "Multiplexed-Domain" Protokoll (CIA DS-202-2).

Mit SDOs können Objekte aus dem Objektverzeichnis gelesen oder geschrieben werden. Es handelt sich um einen bestätigten Dienst. Der so genannte **SDO Client** spezifiziert in seiner Anforderung „Request“ den Parameter, die Zugriffsart (Lesen/Schreiben) und gegebenenfalls den Wert. Der so genannte **SDO Server** führt den Schreib- oder Lesezugriff aus und beantwortet die Anforderung mit einer Antwort „Response“. Im Fehlerfall gibt ein Fehlercode Auskunft über die Fehlerursache. Sende-SDO und Empfangs-SDO werden durch ihre Funktionscodes unterschieden.

Das Mess-System (Slave) entspricht dem SDO Server und verwendet folgende Funktionscodes:

Funktionscode	COB-ID	Bedeutung
11 (1011 bin)	0x580 + Node ID	Slave → SDO Client
12 (1100 bin)	0x600 + Node ID	SDO Client → Slave

Tabelle 1: COB-IDs für Service Data Object (SDO)

### 4.5.1 SDO-Nachrichtenformat

Der maximal 8 Byte lange Datenbereich einer CAN-Nachricht wird von einem SDO wie folgt belegt:

CCD	Index		Subindex	Daten			
Byte 0	Byte 1, Low	Byte 2, High	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7

Tabelle 2: SDO-Nachricht

Der **Kommando-Code (CCD)** identifiziert bei der SDO Request, ob gelesen oder geschrieben werden soll. Bei einem Schreibauftrag wird zusätzlich die Anzahl der zu schreibenden Bytes im CCD kodiert.

Bei der SDO Response zeigt der CCD an, ob die Request erfolgreich war. Im Falle eines Leseauftrags gibt der CCD zusätzlich Auskunft über die Anzahl der gelesenen Bytes:

CCD	Bedeutung	Gültig für
0x22	n Byte schreiben	SDO Request
0x23	4 Byte schreiben	SDO Request
0x2B	2 Byte schreiben	SDO Request
0x2F	1 Byte schreiben	SDO Request
0x60	Schreiben erfolgreich	SDO Response
0x80	Fehler	SDO Response
0x40	Leseanforderung	SDO Request
0x43	4 Byte Daten gelesen	SDO Response auf Leseanforderung
0x4B	2 Byte Daten gelesen	SDO Response auf Leseanforderung
0x4F	1 Byte Daten gelesen	SDO Response auf Leseanforderung

Tabelle 3: Kommando-Codes für SDO

Im Fall eines Fehlers (SDO Response CCD = 0x80) enthält der Datenbereich einen 4-Byte-Fehlercode, der über die Fehlerursache Auskunft gibt. Die Bedeutung der Fehlercodes ist aus der Tabelle 10, Seite 98 zu entnehmen.

### Segment Protokoll, Datensegmentierung

---

Manche Objekte beinhalten Daten, die größer als 4 Byte sind. Um diese Daten lesen zu können, muss das „Segment Protokoll“ benutzt werden.

Zunächst wird der Lesevorgang wie ein gewöhnlicher SDO-Dienst mit dem Kommando-Code = 0x40 eingeleitet. Über die Response wird angezeigt, um wie viele Datensegmente es sich handelt und wie viele Bytes gelesen werden können. Mit nachfolgenden Leseanforderungen können dann die einzelnen Datensegmente gelesen werden. Ein Datensegment besteht jeweils aus 7 Bytes.

Beispiel für das Lesen eines Datensegmentes:

Telegramm 1

CCD	Bedeutung	Gültig für
0x40	Leseanforderung, Einleitung	SDO Request
0x41	1 Datensegment vorhanden Die Anzahl der zu lesenden Bytes steht in den Bytes 4 bis 7.	SDO Response

Telegramm 2

CCD	Bedeutung	Gültig für
0x60	Leseanforderung	SDO Request
0x01	Kein weiteres Datensegment vorhanden. Die Bytes 1 bis 7 beinhalten die angeforderten Daten.	SDO Response

## 4.5.2 Lese SDO

„Domain Upload“ einleiten

### Anforderungs-Protokoll-Format:

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

Lese SDOs								
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 SDOs								
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.

### 4.5.3 Schreibe SDO

„Domain Download“ einleiten

#### Anforderungs-Protokoll-Format:

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

Schreibe SDOs								
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 SDOs								
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.

## 4.6 Netzwerkmanagement, NMT

Das Netzwerkmanagement unterstützt einen vereinfachten Hochlauf (Boot-Up) des Netzes. Mit einem einzigen Telegramm lassen sich z.B. alle Geräte in den Betriebszustand (Operational) versetzen.

Das Mess-System befindet sich nach dem Einschalten zunächst im "Vor-Betriebszustand", (2).

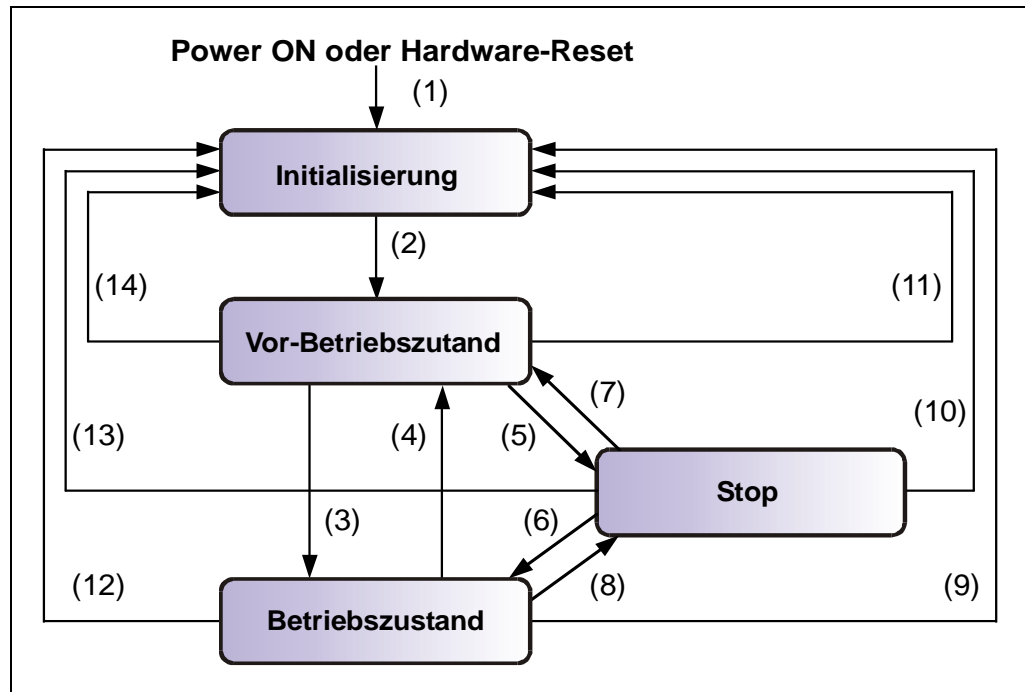


Abbildung 5: Boot-Up-Mechanismus des Netzwerkmanagements

Zustand	Beschreibung
(1)	Automatische Initialisierung nach dem Einschalten
(2)	Beendigung der Initialisierung --> Vor-Betriebszustand
(3),(6)	Start_Remote_Node --> Betriebszustand
(4),(7)	Enter_PRE-OPERATIONAL_State --> Vor-Betriebszustand
(5),(8)	Stop_Remote_Node --> Stop
(9),(10),(11)	Reset_Node --> Reset Knoten
(12),(13),(14)	Reset_Communication --> Reset Kommunikation

#### 4.6.1 Netzwerkmanagement-Dienste

Das **Network Management (NMT)** hat die Aufgabe, Teilnehmer eines CANopen-Netzwerks zu initialisieren, die Teilnehmer in das Netz aufzunehmen, zu stoppen und zu überwachen.

NMT-Dienste werden von einem **NMT-Master** initiiert, der einzelne Teilnehmer (**NMT-Slave**) über deren Node ID anspricht. Eine NMT-Nachricht mit der Node ID 0 richtet sich an **alle** NMT-Slaves.

**Das Mess-System entspricht einem NMT-Slave.**

##### 4.6.1.1 NMT-Dienste zur Gerätekontrolle

Die NMT-Dienste zur Gerätekontrolle verwenden die **COB-ID 0** und erhalten so die höchste Priorität.

Vom Datenfeld der CAN-Nachricht werden nur die ersten beiden Byte verwendet:

CCD	Node ID
Byte 0	Byte 1

Folgende Kommandos sind definiert:

CCD	Bedeutung	Zustand
-	Automatische Initialisierung nach dem Einschalten	(1)
-	Beendigung der Initialisierung --> PRE-OPERATIONAL	(2)
0x01	<b>Start Remote Node</b> Teilnehmer soll in den Zustand OPERATIONAL wechseln und damit den normalen Netzbetrieb starten	(3),(6)
0x02	<b>Stop Remote Node</b> Teilnehmer soll in den Zustand STOPPED übergehen und damit seine Kommunikation stoppen. Eine aktive Verbindungsüberwachung bleibt aktiv.	(5),(8)
0x80	<b>Enter PRE-OPERATIONAL</b> Teilnehmer soll in den Zustand PRE-OPERATIONAL gehen. Alle Nachrichten außer PDOs können verwendet werden.	(4),(7)
0x81	<b>Reset Node</b> Werte der Profilparameter des Objekts auf Default-Werte setzen. Danach Übergang in den Zustand RESET COMMUNICATION.	(9),(10), (11)
0x82	<b>Reset Communication</b> Teilnehmer soll in den Zustand RESET COMMUNICATION gehen. Danach Übergang in den Zustand INITIALIZATION, erster Zustand nach dem Einschalten.	(12),(13), (14)

**Tabelle 4: NMT-Dienste zur Gerätekontrolle**

#### 4.6.1.2 NMT-Dienste zur Verbindungsüberwachung

Mit der Verbindungsüberwachung kann ein NMT-Master den Ausfall eines NMT-Slave und/oder ein NMT-Slave den Ausfall des NMT-Master erkennen:

- **Node Guarding und Life Guarding:**  
Mit diesen Diensten überwacht ein NMT-Master einen NMT-Slave

Das **Node Guarding** wird dadurch realisiert, dass der NMT-Master in regelmäßigen Abständen den Zustand eines NMT-Slave anfordert. Das Toggle-Bit 2<sup>7</sup> im „Node Guarding Protocol“ toggelt nach jeder Abfrage:

Beispiel:

0x85, 0x05, 0x85 ... --> kein Fehler

0x85, 0x05, 0x05 ... --> Fehler

Ist zusätzlich das **Life Guarding** aktiv, erwartet der NMT-Slave innerhalb eines bestimmten Zeitintervalls eine derartige Zustandsabfrage durch den NMT-Master. Ist dies nicht der Fall, wechselt der Slave in den **PRE-OPERATIONAL** Zustand.

Die NMT-Dienste zur Verbindungsüberwachung verwenden den Funktionscode **1110 bin**, also die **COB-ID 0x700+Node ID**.

Index	Beschreibung	
0x100C	<b>Guard Time [ms]</b>	Spätestens nach Ablauf des Zeitintervalls
0x100D	<b>Life Time Factor</b>	<p><b>Life Time = Guard Time x Life Time Factor [ms]</b></p> <p>erwartet der NMT-Slave eine Zustandsabfrage durch den Master.</p> <p>Ist die Guard Time = 0, wird der entsprechende NMT-Slave nicht vom Master überwacht.</p> <p>Ist die Life Time = 0, ist das Life Guarding abgeschaltet.</p>

**Tabelle 5: Parameter für NMT-Dienste**

### 4.7 Layer setting services (LSS) und Protokolle

Die LSS-Dienste und Protokolle, dokumentiert in CiA DS-305 V2.2.5, unterstützen das Abfragen und Konfigurieren verschiedener Parameter des Data Link Layers und des Application Layers eines LSS-Slaves durch ein LSS-Master über das CAN Netzwerk.

Unterstützt werden folgende Parameter:

- Node-ID
- Baudrate
- LSS-Adresse, gemäß dem Identity Objekt 1018h

Somit ist es nicht mehr notwendig, die Node-ID bzw. Baudrate über die DIP-Schalter einzustellen. Der Zugriff auf den LSS-Slave erfolgt dabei über seine LSS-Adresse, bestehend aus:

- Vendor-ID
- Produkt-Code
- Revisions-Nummer und
- Serien-Nummer

Das Mess-System unterstützt folgende Dienste:

#### Switch state services

- Switch state selective
  - einen bestimmten LSS-Slave ansprechen
- Switch state global
  - alle LSS-Slaves ansprechen

#### Configuration services

- Configure Node-ID
  - Node-ID konfigurieren
- Configure bist timing parameters
  - Baudrate konfigurieren
- Activate bist timing parameters
  - Baudrate aktivieren
- Store configured parameters
  - konfigurierte Parameter speichern

#### Inquiry services

- Inquire LSS address
  - LSS-Adresse anfragen
- Inquire Node-ID
  - Node-ID anfragen

#### Identification services

- LSS identify remote slave
  - Identifizierung von LSS-Slaves innerhalb eines bestimmten Bereichs
- LSS identify slave
  - Rückmeldung der LSS-Slaves auf das vorherige Kommando
- LSS identify non-configured remote slave
  - Identifizierung von nicht-konfigurierten LSS-Slaves, Node-ID = FFh
- LSS identify non-configured slave
  - Rückmeldung der LSS-Slaves auf das vorherige Kommando



#### 4.7.1 Finite state automaton, FSA

Der FSA entspricht einer Zustandsmaschine und definiert das Verhalten eines LSS-Slaves. Gesteuert wird die Zustandsmaschine durch LSS COBs erzeugt durch einen LSS-Master, oder NMT COBs erzeugt durch einen NMT-Master, oder lokale NMT-Zustandsübergänge.

Der LSS FSA unterstützt folgende Zustände:

- (0) Initial: Pseudo-Zustand, zeigt die Aktivierung des FSAs an
- (1) LSS waiting: Unterstützung aller Dienste wie unten angegeben
- (2) LSS configuration: Unterstützung aller Dienste wie unten angegeben
- (3) Final: Pseudo-Zustand, zeigt die Deaktivierung des FSAs an

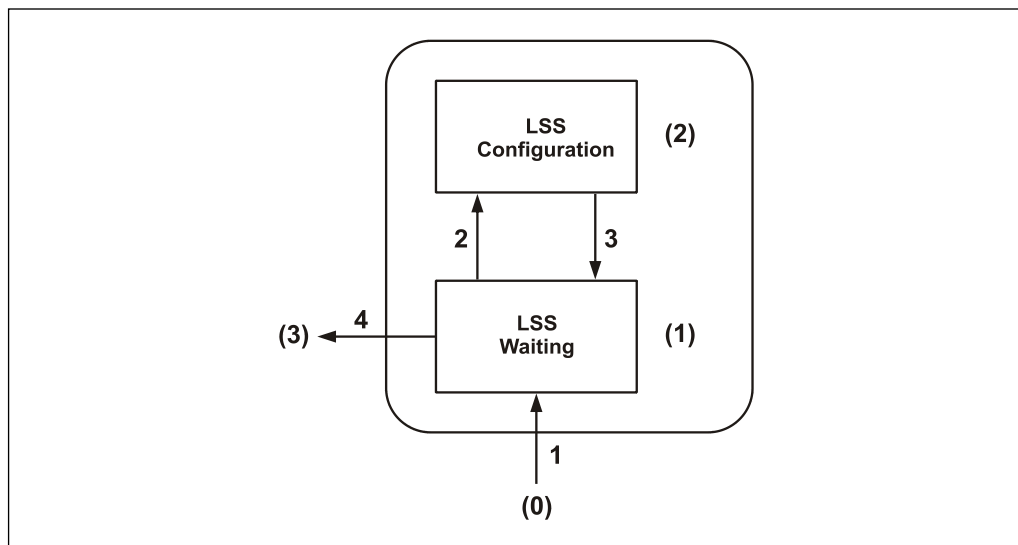


Abbildung 6: LSS FSA Zustandsmaschine

Zustandsverhalten der unterstützten Dienste

Dienste	LSS Waiting	LSS Configuration
Switch state global	Ja	Ja
Switch state selective	Ja	Nein
Activate bist timing parameters	Nein	Ja
Configure bist timing parameters	Nein	Ja
Configure Node-ID	Nein	Ja
Store configured parameters	Nein	Ja
Inquire LSS address	Nein	Ja
Inquire Node-ID	Nein	Ja
LSS identify remote slave	Ja	Ja
LSS identify slave	Ja	Ja
LSS identify non-configured remote slave	Ja	Ja
LSS identify non-configured slave	Ja	Ja

## LSS FSA Zustandsübergänge

Übergang	Ereignisse	Aktionen
1	Automatischer Übergang nach der Initialisierung beim Eintritt entweder in den NMT <code>PRE-OPERATIONAL</code> Zustand oder NMT <code>STOPPED</code> Zustand, oder NMT <code>RESET COMMUNICATION</code> Zustand mit Node-ID = FFh.	keine
2	LSS 'switch state global' Kommando mit Parameter 'configuration_switch' oder 'switch state selective' Kommando	keine
3	LSS 'switch state global' Kommando mit Parameter 'waiting_switch'	keine
4	Automatischer Übergang, wenn eine ungültige Node-ID geändert wurde und die neue Node-ID erfolgreich im nichtflüchtigen Speicher abgelegt werden konnte UND der Zustand LSS waiting angefordert wurde.	keine

Sobald das LSS FSA weitere Zustandsübergänge im NMT FSA von NMT `PRE-OPERATIONAL` auf NMT `STOPPED` und umgekehrt erfährt, führt dies nicht zum Wiedereintritt in den LSS FSA.

## 4.7.2 Übertragung von LSS-Diensten

Über die LSS-Dienste fordert der LSS-Master die einzelnen Dienste an, welche dann durch den LSS-Slave ausgeführt werden. Die Kommunikation zwischen LSS-Master und LSS-Slave wird über die implementierten LSS-Protokolle vorgenommen. Ähnlich wie bei der SDO-Übertragung, werden auch hier zwei COB-IDs für das Senden und Empfangen benutzt:

COB-ID	Bedeutung
0x7E4	LSS-Slave → LSS-Master
0x7E5	LSS-Master → LSS-Slave

Tabelle 6: COB-IDs für Layer Setting Services (LSS)

## 4.7.2.1 LSS-Nachrichtenformat

Der maximal 8 Byte lange Datenbereich einer CAN-Nachricht wird von einem LSS-Dienst wie folgt belegt:

CS	Daten						
Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7

Tabelle 7: LSS-Nachricht

Byte 0 enthält die **Command-Specifier** (CS), danach folgen 7 Byte für die Daten.

## 4.7.3 Switch mode Protokolle

### 4.7.3.1 Switch state global Protokoll

Das angegebene Protokoll hat den *Switch state global service* implementiert und steuert die LSS-Zustandsmaschine des LSS-Slaves. Über den LSS-Master können alle LSS-Slaves im Netzwerk in den *LSS waiting* oder *LSS configuration* Zustand versetzt werden.

LSS-Master --> LSS-Slave

	0	1	2	3	4	5	6	7
COB-ID	CS	Mode	Reserved by CiA					
0x7E5	0x04	0 = Waiting Mode 1 = Configuration Mode						

### 4.7.3.2 Switch state selective Protokoll

Das angegebene Protokoll hat den *Switch state selective service* implementiert und steuert die LSS-Zustandsmaschine des LSS-Slaves. Über den LSS-Master kann nur der LSS-Slave im Netzwerk in den *LSS configuration* Zustand versetzt werden, dessen LSS- Adressattribute der LSS-Adresse entsprechen.

LSS-Master --> LSS-Slave

	0	1	2	3	4	5	6	7
COB-ID	CS	Vendor-ID ( $\triangleq$ Index 1018h:01)				Reserved by CiA		
0x7E5	0x40	LSB		MSB				

	0	1	2	3	4	5	6	7
COB-ID	CS	Product-Code ( $\triangleq$ Index 1018h:02)				Reserved by CiA		
0x7E5	0x41	LSB		MSB				

	0	1	2	3	4	5	6	7
COB-ID	CS	Revision-No. ( $\triangleq$ Index 1018h:03)				Reserved by CiA		
0x7E5	0x42	LSB		MSB				

	0	1	2	3	4	5	6	7
COB-ID	CS	Serial-No. ( $\triangleq$ Index 1018h:04)				Reserved by CiA		
0x7E5	0x43	LSB		MSB				

LSS-Slave --> LSS-Master

	0	1	2	3	4	5	6	7
COB-ID	CS	Reserved by CiA						
0x7E4	0x44							

## 4.7.4 Configuration Protokolle

### 4.7.4.1 Configure Node-ID Protokoll

Das angegebene Protokoll hat den *Configure Node-ID service* implementiert. Über den LSS-Master kann die Node-ID eines einzelnen LSS-Slaves im Netzwerk konfiguriert werden. Hierbei darf sich nur ein LSS-Slave im Zustand *LSS configuration* befinden. Zur Speicherung der neuen Node-ID muss das *Store configuration protocol* an den LSS-Slave übertragen werden. Um die neue Node-ID zu aktivieren, muss der NMT-Dienst *Reset Communication* (0x82) aufgerufen werden.

LSS-Master --> LSS-Slave

	0	1	2	3	4	5	6	7
COB-ID	CS	Node-ID	Reserved by CiA					
0x7E5	0x11	0x01...0x7F, 0xFF						

LSS-Slave --> LSS-Master

	0	1	2	3	4	5	6	7
COB-ID	CS	Error Code	Spec. Error		Reserved by CiA			
0x7E4	0x11							

Node-ID

1...127: gültige Adressen

Error Code

0: Ausführung erfolgreich  
 1: Node-ID außerhalb Bereich, 1...127  
 2...254: Reserved  
 255: applikationsspezifischer Fehler aufgetreten

Specific Error

Wenn Error Code = 255 --> applikationsspezifischer Fehler aufgetreten,  
 sonst reserviert durch die CiA



Die hier festgelegte Node-ID, kann auch über „Objekt 2110h: LSS Node-ID“ mittels SDO geändert werden. Siehe hierzu auch Kap. 5.4 „Einstellen der Node-ID und Baudrate“ auf Seite 39 und Kap. 9.2.12 „Objekt 2110h: LSS Node-ID“ auf Seite 71.

#### 4.7.4.2 Configure bit timing parameters Protokoll

Das angegebene Protokoll hat den *Configure bit timing parameters service* implementiert. Über den LSS-Master kann die Baudrate eines einzelnen LSS-Slaves im Netzwerk konfiguriert werden. Hierbei darf sich nur ein LSS-Slave im Zustand *LSS configuration* befinden. Zur Speicherung der neuen Baudrate muss das *Store configuration protocol* an den LSS-Slave übertragen werden.

LSS-Master --> LSS-Slave

	0	1	2	3	4	5	6	7
COB-ID	CS	Table Selector	Table Index	Reserved by CiA				
0x7E5	0x13	0	0x00...0x08					

LSS-Slave --> LSS-Master

	0	1	2	3	4	5	6	7
COB-ID	CS	Error Code	Spec. Error	Reserved by CiA				
0x7E4	0x13							

Table Selector

0: Standard CiA Baudraten-Tabelle

Table Index

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

Error Code

0: Ausführung erfolgreich  
 1: selektierte Baudrate nicht unterstützt  
 2...254: Reserved  
 255: applikationsspezifischer Fehler aufgetreten

Specific Error

Wenn Error Code = 255 --> applikationsspezifischer Fehler aufgetreten,  
 sonst reserviert durch die CiA



Die hier festgelegte Baudrate, kann auch über „Objekt 2111h: LSS-BitTiming“ mittels SDO geändert werden. Siehe hierzu auch Kap. 5.4 „Einstellen der Node-ID und Baudrate“ auf Seite 39 und Kap. 9.2.13 „Objekt 2111h: LSS-BitTiming“ auf Seite 72.

#### 4.7.4.3 Activate bit timing parameters Protokoll

Das angegebene Protokoll hat den *Activate bit timing parameters service* implementiert und aktiviert die über *Configure bit timing parameters protocol* festgelegte Baudrate bei allen LSS-Slaves im Netzwerk, die sich im Zustand *LSS configuration* befinden.

LSS-Master --> LSS-Slave

	0	1	2	3	4	5	6	7
COB-ID	CS	Switch Delay [ms]		Reserved by CiA				
0x7E5	0x15	LSB	MSB					

##### Switch Delay

Der Parameter *Switch Delay* definiert die Länge zweier Verzögerungsperioden (D1, D2) mit gleicher Länge. Damit wird das Betreiben des Busses mit unterschiedlichen Baudratenparametern verhindert.

Nach Ablauf der Zeit D1 und einer individuellen Verarbeitungsdauer wird die Umschaltung intern im LSS-Slave vorgenommen. Nach Ablauf der Zeit D2 meldet sich der LSS-Slave wieder mit CAN-Nachrichten und der neu eingestellten Baudrate.

Es gilt:

Switch Delay > längste vorkommende Verarbeitungsdauer eines LSS-Slaves

#### 4.7.4.4 Store configuration Protokoll

Das angegebene Protokoll hat den *Store configuration service* implementiert. Über den LSS-Master können die konfigurierten Parameter eines einzelnen LSS-Slaves im Netzwerk in den nichtflüchtigen Speicher abgelegt werden. Hierbei darf sich nur ein LSS-Slave im Zustand *LSS configuration* befinden.

LSS-Master --> LSS-Slave

	0	1	2	3	4	5	6	7
COB-ID	CS	Reserved by CiA						
0x7E5	0x17							

LSS-Slave --> LSS-Master

	0	1	2	3	4	5	6	7
COB-ID	CS	Error Code	Spec. Error	Reserved by CiA				
0x7E4	0x17							

##### Error Code

- 0: Ausführung erfolgreich
- 1: *Store configuration* nicht unterstützt
- 2: Zugriff auf Speichermedium fehlerhaft
- 3...254: Reserved
- 255: applikationsspezifischer Fehler aufgetreten

##### Specific Error

Wenn Error Code = 255 --> applikationsspezifischer Fehler aufgetreten, sonst reserviert durch die CiA

## 4.7.5 Inquire LSS-Address Protokolle

### 4.7.5.1 Inquire identity Vendor-ID Protokoll

Das angegebene Protokoll hat den *Inquire LSS address service* implementiert. Über den LSS-Master kann die Vendor-ID eines einzelnen LSS-Slaves im Netzwerk ausgelesen werden. Hierbei darf sich nur ein LSS-Slave im Zustand *LSS configuration* befinden.

LSS-Master --> LSS-Slave

	0	1	2	3	4	5	6	7
<b>COB-ID</b>	<b>CS</b>	Reserved by CiA						
0x7E5	0x5A							

LSS-Slave --> LSS-Master

	0	1	2	3	4	5	6	7
<b>COB-ID</b>	<b>CS</b>	<b>Vendor-ID</b> ( $\triangleq$ Index 1018h:01)				Reserved by CiA		
0x7E4	0x5A	LSB		MSB				

### 4.7.5.2 Inquire identity Product-Code Protokoll

Das angegebene Protokoll hat den *Inquire LSS address service* implementiert. Über den LSS-Master kann der Produkt-Code eines einzelnen LSS-Slaves im Netzwerk ausgelesen werden. Hierbei darf sich nur ein LSS-Slave im Zustand *LSS configuration* befinden.

LSS-Master --> LSS-Slave

	0	1	2	3	4	5	6	7
<b>COB-ID</b>	<b>CS</b>	Reserved by CiA						
0x7E5	0x5B							

LSS-Slave --> LSS-Master

	0	1	2	3	4	5	6	7
<b>COB-ID</b>	<b>CS</b>	<b>Product-Code</b> ( $\triangleq$ Index 1018h:02)				Reserved by CiA		
0x7E4	0x5B	LSB		MSB				

## 4.7.5.3 Inquire identity Revision-Number Protokoll

Das angegebene Protokoll hat den *Inquire LSS address service* implementiert. Über den LSS-Master kann die Revisionsnummer eines einzelnen LSS-Slaves im Netzwerk ausgelesen werden. Hierbei darf sich nur ein LSS-Slave im Zustand *LSS configuration* befinden.

LSS-Master --> LSS-Slave

	0	1	2	3	4	5	6	7
<b>COB-ID</b>	<b>CS</b>	Reserved by CiA						
0x7E5	0x5C							

LSS-Slave --> LSS-Master

	0	1	2	3	4	5	6	7
<b>COB-ID</b>	<b>CS</b>	<b>Revision-No.</b> ( $\triangleq$ Index 1018h:03)				Reserved by CiA		
0x7E4	0x5C	LSB		MSB				

## 4.7.5.4 Inquire identity Serial-Number Protokoll

Das angegebene Protokoll hat den *Inquire LSS address service* implementiert. Über den LSS-Master kann die Seriennummer eines einzelnen LSS-Slaves im Netzwerk ausgelesen werden. Hierbei darf sich nur ein LSS-Slave im Zustand *LSS configuration* befinden.

LSS-Master --> LSS-Slave

	0	1	2	3	4	5	6	7
<b>COB-ID</b>	<b>CS</b>	Reserved by CiA						
0x7E5	0x5D							

LSS-Slave --> LSS-Master

	0	1	2	3	4	5	6	7
<b>COB-ID</b>	<b>CS</b>	<b>Serial-No.</b> ( $\triangleq$ Index 1018h:04)				Reserved by CiA		
0x7E5	0x5D	LSB		MSB				



#### 4.7.6 Inquire Node-ID Protokoll

Das angegebene Protokoll hat den *Inquire Node-ID service* implementiert. Über den LSS-Master kann die Node-ID eines einzelnen LSS-Slaves im Netzwerk ausgelesen werden. Hierbei darf sich nur ein LSS-Slave im Zustand *LSS configuration* befinden.

LSS-Master --> LSS-Slave

	0	1	2	3	4	5	6	7
COB-ID	CS	Reserved by CiA						
0x7E5	0x5E							

LSS-Slave --> LSS-Master

	0	1	2	3	4	5	6	7
COB-ID	CS	Node-ID	Reserved by CiA					
0x7E4	0x5E	0x01...0x7F, 0xFF						

#### Node-ID

Entspricht der Node-ID des selektierten Gerätes. Wenn die Node-ID eben gerade erst über den *Configure Node-ID service* geändert wurde, wird die ursprüngliche Node-ID zurückgemeldet. Erst nach Ausführung des NMT-Dienstes *Reset Communication* (0x82) wird die aktuelle Node-ID zurückgemeldet.

## 4.7.7 Identification Protokolle

### 4.7.7.1 LSS identify remote slave Protokoll

Das angegebene Protokoll hat den *LSS identify remote slave service* implementiert. Über den LSS-Master können LSS-Slaves im Netzwerk in einem bestimmten Bereich identifiziert werden. Alle LSS-Slaves, die der angegebenen Vendor-ID, Product-Code, Revision-No. – Bereich und Serial-No. – Bereich entsprechen, antworten mit dem *LSS identify slave protocol*.

LSS-Master --> LSS-Slave

	0	1	2	3	4	5	6	7
<b>COB-ID</b>	<b>CS</b>	<b>Vendor-ID</b> ( $\triangle$ Index 1018h:01)				Reserved by CiA		
0x7E5	0x46	LSB		MSB				

	0	1	2	3	4	5	6	7
<b>COB-ID</b>	<b>CS</b>	<b>Product-Code</b> ( $\triangle$ Index 1018h:02)				Reserved by CiA		
0x7E5	0x47	LSB		MSB				

	0	1	2	3	4	5	6	7
<b>COB-ID</b>	<b>CS</b>	<b>Revision-No. LOW</b>				Reserved by CiA		
0x7E5	0x48	LSB		MSB				

	0	1	2	3	4	5	6	7
<b>COB-ID</b>	<b>CS</b>	<b>Revision-No. HIGH</b>				Reserved by CiA		
0x7E5	0x49	LSB		MSB				

	0	1	2	3	4	5	6	7
<b>COB-ID</b>	<b>CS</b>	<b>Serial-No. LOW</b>				Reserved by CiA		
0x7E5	0x4A	LSB		MSB				

	0	1	2	3	4	5	6	7
<b>COB-ID</b>	<b>CS</b>	<b>Serial-No. HIGH</b>				Reserved by CiA		
0x7E5	0x4B	LSB		MSB				

### 4.7.7.2 LSS identify slave Protokoll

Das angegebene Protokoll hat den *LSS identify slave service* implementiert. Alle LSS-Slaves, die den im *LSS identify remote slave protocol* angegebenen LSS-Adress-Attributen entsprechen, antworten mit diesem Protokoll.

LSS-Slave --> LSS-Master

	0	1	2	3	4	5	6	7
<b>COB-ID</b>	<b>CS</b>	Reserved by CiA						
0x7E4	0x4F							

#### 4.7.7.3 LSS identify non-configured remote slave Protokoll

Das angegebene Protokoll hat den *LSS identify non-configured remote slave service* implementiert. Über den LSS-Master werden alle nicht-konfigurierten LSS-Slaves (Node-ID = FFh) im Netzwerk identifiziert. Die betreffenden LSS-Slaves antworten mit dem *LSS identify non-configured slave protocol*.

LSS-Master --> LSS-Slave

	0	1	2	3	4	5	6	7
COB-ID	CS	Reserved by CiA						
0x7E5	0x4C							

#### 4.7.7.4 LSS identify non-configured slave Protokoll

Das angegebene Protokoll hat den *LSS identify non-configured slave service* implementiert. Alle LSS-Slaves, die eine ungültige Node-ID (FFh) besitzen, antworten nach Ausführung des *LSS identify non-configured remote slave protocol* mit diesem Protokoll.

LSS-Slave --> LSS-Master

	0	1	2	3	4	5	6	7
COB-ID	CS	Reserved by CiA						
0x7E4	0x50							

### 4.8 Geräteprofil

Die CANopen Geräteprofile beschreiben das "was" der Kommunikation. In ihnen wird die Bedeutung der übertragenen Daten eindeutig und hersteller-unabhängig festgelegt. So lassen sich die Grundfunktionen einer jeden Gerätekategorie

z.B. für Encoder: **CiA DS-406**

einheitlich ansprechen. Auf der Grundlage dieser standardisierten Profile kann auf identische Art und Weise über den Bus auf CANopen Geräte zugegriffen werden. Damit sind Geräte, die dem gleichen Geräteprofil folgen, weitgehend untereinander austauschbar.

Weitere Informationen zum CANopen erhalten Sie auf Anfrage von der **CAN in Automation** Nutzer- und Herstellervereinigung (CiA) unter nachstehender Adresse:

---

#### **CAN in Automation**

Am Weichselgarten 26  
DE-91058 Erlangen

Tel. +49-9131-69086-0  
Fax +49-9131-69086-79

Website: [www.can-cia.org](http://www.can-cia.org)  
E-Mail: [headquarters@can-cia.org](mailto:headquarters@can-cia.org)

---

## 5 Installation / Inbetriebnahmevorbereitung

Das CANopen System wird in Bustopologie mit Abschlusswiderständen (120 Ohm) am Anfang und am Ende verkabelt. Stichleitungen sollten möglichst vermieden werden. Das Kabel ist als geschirmtes Twisted Pair Kabel auszuführen und sollte eine Impedanz von 120 Ohm und einen Widerstand von 70 mΩ/m haben. Die Datenübertragung erfolgt über die Signale CAN-H und CAN-L mit einem gemeinsamen GND als Datenbezugspotential. Optional kann auch eine 24 Volt Versorgungsspannung mitgeführt werden.

In einem CANopen Netzwerk können maximal **127** Teilnehmer angeschlossen werden. Das Mess-System unterstützt den Node-ID Bereich von 1–127. Die Übertragungsgeschwindigkeit lässt sich per Dip-Schalter, LSS Protokoll oder SDO einstellen und unterstützt die folgenden Baudraten:

- 10 kbit/s
- 20 kbit/s
- 50 kbit/s
- 100 kbit/s (*nur LSS oder SDO*)
- 125 kbit/s
- 250 kbit/s
- 500 kbit/s
- 800 kbit/s
- 1 Mbit/s

Die Länge eines CANopen Netzwerkes ist abhängig von der Übertragungsgeschwindigkeit und ist nachfolgend dargestellt:

Kabelquerschnitt	10 kbit/s	20 kbit/s	50 kbit/s	100 kbit/s	125 kbit/s	250 kbit/s	500 kbit/s	800 kbit/s	1 Mbit/s
0.25 mm <sup>2</sup> – 0.34 mm <sup>2</sup>	5000 m	2500 m	1000 m	ca. 600 m	500 m	250 m	100 m	50 m	25 m

*Um einen sicheren und störungsfreien Betrieb zu gewährleisten, sind die*



- *ISO 11898,*
- *die Empfehlungen der CiA DR 303-1 (CANopen cabling and connector pin assignment)*
- *und sonstige einschlägige Normen und Richtlinien zu beachten!*

*Insbesondere sind die EMV-Richtlinie sowie die Schirmungs- und Erdungsrichtlinien in den jeweils gültigen Fassungen zu beachten!*

### 5.1 Anschluss – Hinweise

Die elektrischen Ausstattungsmerkmale werden hauptsächlich durch die variable Anschluss-Technik vorgegeben.



*Der Anschluss kann nur in Verbindung mit der gerätespezifischen Steckerbelegung vorgenommen werden!*

*Bei der Auslieferung des Mess-Systems wird jeweils eine Steckerbelegung in gedruckter Form beigelegt und sie kann nachträglich auch von der Seite „[www.tr-electronic.de/service/downloads/steckerbelegungen.html](http://www.tr-electronic.de/service/downloads/steckerbelegungen.html)“ heruntergeladen werden. Die Steckerbelegungsnummer ist auf dem Typenschild des Mess-Systems vermerkt.*

### 5.2 Bus-Terminierung

Ist das Mess-System der letzte Teilnehmer im CAN-Segment, ist der Bus durch den Terminierungsschalter = ON abzuschließen. In diesem Zustand wird der weiterführende CAN-Bus abgekoppelt (siehe Steckerbelegung). Position und Lage des Terminierungsschalters sind der gerätespezifischen Steckerbelegung zu entnehmen.

Bei Mess-Systemen ohne Terminierungsschalter muss der weiterführende CAN-Bus mit einem 120  $\Omega$  Abschlusswiderstand versehen werden.

### 5.3 Einschalten der Versorgungsspannung

Nachdem der Anschluss und alle Einstellungen vorgenommen worden sind, kann die Versorgungsspannung eingeschaltet werden.

Nach dem Einschalten der Versorgungsspannung und Beendigung der Initialisierung geht das Mess-System in den Vor-Betriebszustand (PRE-OPERATIONAL). Dieser Zustand wird in der Standardeinstellung durch die Boot-Up-Meldung „**COB-ID 0x700+Node ID**“ bestätigt. Falls das Mess-System einen internen Fehler erkennt, wird eine Emergency-Meldung mit dem Fehlercode übertragen (siehe Kapitel „Emergency-Meldung“, Seite 94).

Im PRE-OPERATIONAL-Zustand ist zunächst nur eine Parametrierung über Service-Daten-Objekte möglich. Es ist aber möglich, PDOs unter Nutzung von SDOs zu konfigurieren. Ist das Mess-System in den Zustand OPERATIONAL überführt worden, ist auch eine Übertragung von PDOs möglich.

## 5.4 Einstellen der Node-ID und Baudrate

Die Node-ID und Baudrate können über DIP-Schalter, LSS-Dienste oder SDOs eingestellt werden.

Im Auslieferungszustand müssen die Node-ID und die Baudrate über DIP-Schalter festgelegt werden. Um die Einstellungen mittels LSS oder SDOs vornehmen zu können, müssen die DIP-Schalter 1...7 vor dem Einschalten der Versorgungsspannung auf „ON“ gesetzt werden um die LSS- bzw. SDO-Dienste zu aktivieren.

Besitzt das Mess-System keine DIP-Schalter, sind automatisch die über LSS- bzw. SDO-Dienste eingestellte Node-ID und Baudrate aktiv.



- Eine Node-ID darf nur einmal im CAN-Bus vergeben werden.
- Sind die LSS-Dienste aktiv, ist die über DIP-Schalter eingestellte Baudrate unwirksam.

### 5.4.1 Einstellung mittels DIP-Schalter

Verfügt das Mess-System über DIP-Schalter wird die Node-ID über die DIP-Schalter 1...7 und die Baudrate über die DIP-Schalter 8...10 gemäß Steckerbelegung eingestellt.



- Die Node-ID entspricht der Schalterstellung als Binärwert + 1.
- Die Schalterstellung wird nur im Einschaltmoment gelesen, nachträgliche Änderungen werden nicht erkannt.

### 5.4.2 Einstellung mittels LSS-Dienste

Nachfolgend ist der Ablauf zur Einstellung der Node-ID und Baudrate mittels LSS-Diensten beschrieben. Siehe auch Kap.: 4.7.4.1 „Configure Node-ID Protokoll“ und 4.7.4.2 „Configure bit timing parameters Protokoll“.

#### 5.4.2.1 Konfiguration der Node-ID

Annahme:

- LSS-Adresse unbekannt
- der LSS-Slave ist der einzige Teilnehmer in Netzwerk
- es soll die Node-ID 12 dez. eingestellt werden

Vorgehensweise:

- LSS-Slave mit dem Dienst 04 *Switch state global protocol*, Mode = 1 in den Zustand *Configuration state* bringen.
- Dienst 17 *Configure Node-ID protocol*, Node-ID = 12 ausführen.  
--> Rückmeldung abwarten und erfolgreiche Ausführung überprüfen,  
--> Error Code = 0.
- Dienst 23 *Store configuration protocol* ausführen.  
--> Rückmeldung abwarten und erfolgreiche Ausführung überprüfen,  
--> Error Code = 0.
- LSS-Slave mit dem Dienst 04 *Switch state global protocol*, Mode = 0 in den Zustand *Waiting state* bringen.
- NMT-Dienst *Reset Communication* (0x82) aufrufen, damit die neue Node-ID aktiv wird.

### 5.4.2.2 Konfiguration der Baudrate

Annahme:

- LSS-Adresse unbekannt
- der LSS-Slave ist der einzige Teilnehmer in Netzwerk
- es soll die Baudrate 125 kbit/s eingestellt werden

Vorgehensweise:

- NMT-Dienst *Stop Remote Node* (0x02) aufrufen, um den LSS-Slave in den *Stopped state* zu bringen. Der LSS-Slave sollte keine CAN-Nachrichten mehr senden --> Heartbeat abgeschaltet.
- LSS-Slave mit dem Dienst 04 *Switch state global protocol*, Mode = 1 in den Zustand *Configuration state* bringen.
- Dienst 19 *Configure bit timing parameters protocol* ausführen, Table Selector = 0, Table Index = 4  
--> Rückmeldung abwarten und erfolgreiche Ausführung überprüfen,  
--> Error Code = 0.
- Dienst 21 *Activate bit timing parameters protocol* aufrufen, damit die neue Baudrate aktiv wird.
- Dienst 23 *Store configuration protocol* ausführen.  
--> Rückmeldung abwarten und erfolgreiche Ausführung überprüfen,  
--> Error Code = 0.
- LSS-Slave mit dem Dienst 04 *Switch state global protocol*, Mode = 0 in den Zustand *Waiting state* bringen.

### 5.4.3 Einstellung mittels SDOs

Die Node-ID wird über „Objekt 2110h: LSS Node-ID“ und die Baudrate über „Objekt 2111h: LSS-BitTiming“ eingestellt. Siehe Kap.: 9.2.12 und 9.2.13.



Die über SDOs eingestellte Node-ID und Baudrate entsprechen den über LSS-Dienste eingestellten Werten und überschreiben sich gegenseitig.

---



## 6 Inbetriebnahme

### 6.1 CAN – Schnittstelle

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 ist galvanisch von der Mess-System-Elektronik getrennt und wird über einen internen DC/DC-Konverter gespeist. Eine externe Spannungsversorgung für den Bustreiber ist nicht notwendig.

Das CANopen Kommunikationsprofil CiA 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.

Das Mess-System mit CANopen Protokoll unterstützt das Geräteprofil für Encoder (CiA DS 406, Version 3.2.0). **Die Mess-Systeme 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 (SDOs) des Mess-Systems auslesen und die Funktionalität programmieren.

Die Auswahl der Übertragungsrate und Node-ID (Geräteadresse) erfolgt über DIP-Schalter oder LSS-Dienste.

#### 6.1.1 EDS-Datei

Die EDS-Datei (elektronisches Datenblatt) enthält alle Informationen über die Mess-System-spezifischen Parameter sowie Betriebsarten des Mess-Systems. Die EDS-Datei wird durch das CANopen-Netzwerkkonfigurationswerkzeug eingebunden, um das Mess-System ordnungsgemäß konfigurieren bzw. in Betrieb nehmen zu können.

**Download:** [www.tr-electronic.de/f/TR-ECE-ID-MUL-0064](http://www.tr-electronic.de/f/TR-ECE-ID-MUL-0064)

#### 6.1.2 Bus-Statusanzeige

Lage, Zuordnung und Blinkfrequenz der Statusanzeige (LEDs) ist der gerätespezifischen Steckerbelegung zu entnehmen.



*Bei der Auslieferung des Mess-Systems wird jeweils eine Steckerbelegung in gedruckter Form beigelegt und sie kann nachträglich auch von der Seite „[www.tr-electronic.de/service/downloads/steckerbelegungen.html](http://www.tr-electronic.de/service/downloads/steckerbelegungen.html)“ heruntergeladen werden. Die Steckerbelegungsnummer ist auf dem Typenschild des Mess-Systems vermerkt.*

---

Entsprechende Maßnahmen im Fehlerfall siehe Kapitel „Optische Anzeigen“, Seite 95.

## 7 Kommunikations-Profil

Generell existieren zwei Arten von Prozessdaten-Objekten (PDO):

1. Sende-PDOs (TPDO), um Daten zu übertragen
2. Empfangs-PDOs (RPDO), um Daten zu empfangen

Vom Mess-System werden nur Sende-PDOs unterstützt, um den Istwert bzw. Geschwindigkeitswert zu übertragen.

Die TPDOs werden festgelegt durch die TPDO Kommunikationsparameter 1800h-1801h und die TPDO Mappingparameter 1A00h-1A01h. Während die TPDO Kommunikationsparameter die Kommunikationsmöglichkeiten beschreiben, beinhalten die TPDO Mappingparameter Informationen über den Inhalt des TPDOs.

### 7.1 Aufbau der Kommunikationsparameter, 1800h-1801h

Subindex 0 beinhaltet die Anzahl der gültigen Objekteinträge.

Subindex 1 beinhaltet die COB-ID für das TPDO:

31	30	29	28	11	10	0
Valid	RTR	Frame	0 0000h	11-Bit CAN-ID		
MSB				LSB		

Bit(s)	Beschreibung
Valid	0: PDO existiert / ist gültig 1: PDO existiert nicht / ist nicht gültig
RTR	0: Remote Frame erlaubt für dieses PDO 1: kein Remote Frame erlaubt für dieses PDO
Frame	0: 11-Bit CAN-ID gültig, normaler CAN Frame 1: 29-Bit CAN-ID gültig, erweiterter CAN Frame (nicht unterstützt)
11-Bit CAN-ID	11-Bit CAN-ID des normalen CAN Frames

Subindex 2 definiert die Übertragungsart für das TPDO:

Wert	Beschreibung
01h	Istwert wird synchron über einen Remote-Frame oder SYNC-Telegramm übertragen
02h	Istwert wird synchron über einen Remote-Frame oder zyklisch nach jedem 2. SYNC-Telegramm übertragen
03h	Istwert wird synchron über einen Remote-Frame oder zyklisch nach jedem 3. SYNC-Telegramm übertragen
...	...
F0h	Istwert wird synchron über einen Remote-Frame oder zyklisch nach jedem 240. SYNC-Telegramm übertragen
FDh	Istwert kann nur über einen Remote-Frame übertragen werden
FEh	Istwert wird asynchron mit dem Timerwert aus den Objekten 1800h und 1801h übertragen (Subindex 5)

Subindex 3 beinhaltet die Sperrzeit für das TPDO. Die Zeit definiert die Mindestzeit zwischen zwei hintereinander folgenden PDO Übertragungen, wenn die Übertragungsart FEh eingestellt wurde. Der Wert wird definiert als Vielfaches von 100 µs. Der Wert 0 deaktiviert die Sperrzeit.

Der Wert darf nicht geändert werden während das PDO existiert (Bit 31 von Subindex 1 = 0).

Subindex 4 wird nicht unterstützt.

Subindex 5 beinhaltet den Event-Timer. Die Zeit definiert die Maximalzeit zwischen zwei hintereinander folgenden PDO Übertragungen, wenn die Übertragungsart FEh eingestellt wurde. Der Wert wird definiert als Vielfaches von 1 ms. Der Wert 0 deaktiviert den Event-Timer.

Der Event-Timer, Subindex 5 des Kommunikationsparameters 1800h, ist fest verknüpft mit dem Objekt 6200h: Cyclic-Timer. Dies bedeutet, dass eine Änderung des Event-Timers sich auch im Cyclic Timer auswirkt und umgekehrt. Der Kommunikationsparameter 1801h benutzt ausschließlich seinen eigenen Timer, Zugriff über Subindex 5.

## 7.2 Aufbau der Mappingparameter, 1A00h-1A01h

Subindex 0 beinhaltet die Anzahl der gültigen Objekteinträge. Der Wert 0 deaktiviert das Mapping.

Die nachfolgenden Subindizes beinhalten die Information der gemappten Applikationsobjekte. Das Objekt beschreibt den Inhalt des PDOS durch ihren Index, Subindex und der Länge in Bit:

31	16	15	8	7	0
Index			Subindex		Länge in Bit
MSB			LSB		

### 7.2.1 Ändern der Mappingeinstellung

Vorgehensweise:

- Löschen des TPDOs durch Setzen des Bits „Valid“ auf 1 im Subindex 1 des entsprechenden Kommunikationsparameters 1800h-1801h.
- Deaktivieren der Mappingfunktion durch Setzen des Subindexes 0 auf 0 in den entsprechenden Mappingparametern 1A00h-1A01h.
- Änderung des Mappings in den entsprechenden Mappingparametern 1A00h-1A01h vornehmen (ab Subindex 1).
- Aktivieren der Mappingfunktion durch Setzen des Subindexes 0 auf die Anzahl der gemappten Objekte in den entsprechenden Mappingparametern 1A00h-1A01h.
- Erzeugen des TPDOs durch Setzen des Bits „Valid“ auf 0 im Subindex 1 des entsprechenden Kommunikationsparameters 1800h-1801h. Die gewünschte COB-ID und das Bit „Valid“ müssen mit einem Schreibvorgang gesetzt werden!
- Mapping-Konfiguration über „Objekt 1010h: Parameter abspeichern“ speichern.

Für das Mapping vorgesehene Objekte:

- Objekt 200Ah: Geschwindigkeit INT32, siehe Seite 69
- Objekt 6004h: Positionswert, siehe Seite 83
- Objekt 6030h: Geschwindigkeit, siehe Seite 84
- Objekt 6400h: Bereichs-Statusregister, siehe Seite 85
- Objekt 6503h: Alarme, siehe Seite 88

### 7.3 Übertragung des Mess-System-Positionswertes

Bevor die Mess-System-Position übertragen werden kann, muss das Mess-System 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 Mess-Systems (Slave) startet nur dieses Gerät.

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

In der Standardeinstellung wird nach dem Node-Start Kommando einmal der Mess-System Positionswert mit der COB-ID des Objekts 1800h übertragen. Dieser Dienst kann über das "Objekt 2101h: Senden von PDO bei Node-Start" verhindert werden, siehe Seite 71.

**In der Standardeinstellung kann jetzt der Positionswert auf verschiedene Arten übertragen werden:**

### 7.3.1 Erstes Sende-Prozessdaten-Objekt

Dieses TPDO überträgt in der Standardeinstellung den Mess-System-Istwert asynchron. Der Timerwert ist im Subindex 5 bzw. Index 6200h gespeichert. Die Standardeinstellung des Timers ist 0, d.h. der Timer ist abgeschaltet.

Index	Subindex	Kommentar	Standardwert	Attr.
1800h	0	größter unterstützter Subindex	5	ro
	1	COB-ID benützt durch TPDO 1	180h + Node-ID	rw
	2	Übertragungsart	254	rw
	3	Sperrzeit	0	rw
	4	-	-	-
	5	Event Timer	0	rw
1A00h	0	größter unterstützter Subindex	1 (max. 8)	rw
	1	1. gemapptes Objekt	6004 0020h	rw
	2	2. gemapptes Objekt	0000 0000h	rw
	3	3. gemapptes Objekt	0000 0000h	rw
	4	4. gemapptes Objekt	0000 0000h	rw
	5	5. gemapptes Objekt	0000 0000h	rw
	6	6. gemapptes Objekt	0000 0000h	rw
	7	7. gemapptes Objekt	0000 0000h	rw
	8	8. gemapptes Objekt	0000 0000h	rw

### 7.3.2 Zweites Sende-Prozessdaten-Objekt

Dieses TPDO überträgt in der Standardeinstellung den Mess-System-Istwert synchron (einmalig auf Anforderung). Anforderung über Remote-Frame (Standard COB-ID: 280h+Node-ID) oder SYNC-Telegramm (Standard COB-ID: 080h).

Index	Subindex	Kommentar	Standardwert	Attr.
1801h	0	größter unterstützter Subindex	5	ro
	1	COB-ID benützt durch TPDO 2	280h + Node-ID	rw
	2	Übertragungsart	1	rw
	3	Sperrzeit	0	rw
	4	-	-	-
	5	Event Timer	0	rw
1A01h	0	größter unterstützter Subindex	1 (max. 8)	rw
	1	1. gemapptes Objekt	6004 0020h	rw
	2	2. gemapptes Objekt	0000 0000h	rw
	3	3. gemapptes Objekt	0000 0000h	rw
	4	4. gemapptes Objekt	0000 0000h	rw
	5	5. gemapptes Objekt	0000 0000h	rw
	6	6. gemapptes Objekt	0000 0000h	rw
	7	7. gemapptes Objekt	0000 0000h	rw
	8	8. gemapptes Objekt	0000 0000h	rw

## 8 Kommunikationsspezifische Standard-Objekte (CiA DS-301)

Folgende Tabelle zeigt eine Übersicht der unterstützten Indexe im Kommunikationsprofilbereich:

M = Mandatory (zwingend)

O = Optional

Index (h)	Objekt	Name	Typ	Attr.	M/O	Seite
1000	VAR	Gerätetyp	Unsigned32	ro	M	47
1001	VAR	Fehlerregister	Unsigned8	ro	M	47
1002	VAR	Hersteller-Status-Register	Unsigned32	ro	O	48
1003	ARRAY	Vordefiniertes Fehlerfeld	Unsigned32	rw	O	48
1005	VAR	COB-ID SYNC-Nachricht	Unsigned32	rw	O	49
<sup>1)</sup> 1008	VAR	Hersteller Gerätenamen	VisibleString	const	O	49
<sup>1)</sup> 1009	VAR	Hardwareversion	VisibleString	const	O	49
<sup>1)</sup> 100A	VAR	Softwareversion	VisibleString	const	O	49
100C	VAR	Guard-Time (Überwachungszeit)	Unsigned16	rw	O	50
100D	VAR	Life-Time-Faktor (Zeitdauer-Faktor)	Unsigned8	rw	O	50
1010	ARRAY	Parameter abspeichern	Unsigned32	rw	O	51
1011	ARRAY	Parameter wieder herstellen	Unsigned32	rw	O	52
1014	VAR	COB-ID EMCY	Unsigned32	rw	O	53
1016	ARRAY	Consumer Heartbeat Time	Unsigned32	rw	O	53
1017	VAR	Producer Heartbeat Time	Unsigned16	rw	O	54
1018	RECORD	Identity Objekt	Identity (23h)	ro	M	54
<sup>1)</sup> 1021	VAR	EDS abspeichern	Domain	ro	O	55
1022	VAR	EDS Speicherformat	Unsigned8	ro	M	55
1029	ARRAY	Verhalten im Fehlerfall	Unsigned8	rw	O	55
<sup>1)</sup> 1F50	ARRAY	Programmdaten	Domain	rw	O	56
1F51	ARRAY	Programmsteuerung	Unsigned8	rw	M	56
1F56	ARRAY	Programm Software Identifikation	Unsigned32	ro	M	57
1F57	ARRAY	Programmstatus	Unsigned32	ro	M	57
1F80	VAR	NMT Autostart	Unsigned32	rw	O	58

**Tabelle 8: Kommunikationsspezifische Standard-Objekte**

<sup>1)</sup> segmentiertes Lesen

## 8.1 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.

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 Encoder
02	Absoluter Multi-Turn Encoder

## 8.2 Objekt 1001h: Fehlerregister

Das Fehlerregister zeigt bitkodiert den Fehlerzustand des Mess-Systems an. Es können auch mehrere Fehler gleichzeitig durch ein gesetztes Bit angezeigt werden. Die genauere Fehlerursache kann den Bits 0 – 15 aus dem Objekt 0x1003 entnommen werden. Im Moment des Auftretens wird ein Fehler durch eine EMCY-Nachricht signalisiert.

Unsigned8

Bit	Bedeutung
0	generischer Fehler
1	0
2	0
3	0
4	Kommunikationsfehler (Überlauf, Fehlerstatus)
5	0
6	0
7	0

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

Dieses Objekt wird durch das Mess-System nicht verwendet, bei Lesezugriff ist der Wert immer "0".

### 8.4 Objekt 1003h: Vordefiniertes Fehlerfeld

Dieses Objekt speichert den zuletzt aufgetretenen Mess-System-Fehler und zeigt den Fehler über das Emergency-Objekt an. Jeder neue Fehler überschreibt einen zuvor gespeicherten Fehler in Subindex 1. Subindex 0 enthält die Anzahl der aufgetretenen Fehler. Die Bedeutung der Fehlercodes kann aus der Tabelle 11, Seite 100 entnommen werden.

Index	Subindex	Kommentar	Typ	Attr.
1003h	0	Anzahl der Fehler	Unsigned8	ro
	1	Standard Fehlerfeld	Unsigned32	rw

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, wird nicht unterstützt	



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

Dieses Objekt definiert die COB-ID des Synchronisierungs-Objekts (SYNC). Es definiert weiterhin, ob das Gerät die SYNC-Nachricht verarbeitet, oder ob das Gerät die SYNC-Nachricht erzeugt. Das Mess-System unterstützt jedoch nur die Verarbeitung von SYNC-Nachrichten und verwendet den 11-Bit-Identifizier.

Unsigned32

MSB

LSB

31	30	29	28-11	10-0
X	0	0	0	00 1000 0000

Bit 31 keine Bedeutung

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 Mess-Systems in der Standardeinstellung einmalig durch das zweite Sendeprozessdaten-Objekt (Objekt 1801h) übertragen.

Objekt	Funktions-Code	COB-ID
SYNC	0001	80h

## 8.6 Objekt 1008h: Hersteller Gerätenamen

Enthält den Hersteller Gerätenamen (visible string),  
Übertragung per „Segment Protokoll“.

## 8.7 Objekt 1009h: Hersteller Hardwareversion

Enthält die Hersteller Hardwareversion (visible string),  
Übertragung per „Segment Protokoll“.

## 8.8 Objekt 100Ah: Hersteller Softwareversion

Enthält die Firmware-Nr. (visible string)  
Übertragung per „Segment Protokoll“.

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

Die Objekte der Indexe 100Ch und 100Dh beinhalten die Guard-Time in Millisekunden und den Live-Time-Faktor (Zeitdauer-Faktor). Der Live-Time-Faktor multipliziert mit der Guard-Time ergibt die Zeitdauer für das Node-Guarding-Protokoll. Standardwert = 0.

Unsigned16

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

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

Der Live-Time-Faktor multipliziert mit der Guard-Time ergibt die Zeitdauer für das Node-Guarding-Protokoll. Standardwert = 0.

Unsigned8

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

## 8.11 Objekt 1010h: Parameter abspeichern

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

Index	Subindex	Kommentar	Typ	Attr.
1010h	0	größter unterstützter Subindex	Unsigned8	ro
	1	alle Parameter speichern	Unsigned32	rw
	2	Kommunikations-Parameter speichern (Objekte: 1000h...1FFFh)	Unsigned32	rw
	3	Gerätespezifische-Parameter speichern (Objekte: 6000h...9FFFh)	Unsigned32	rw
	4	Herstellerspezifische-Parameter speichern (Objekte: 2000h...5FFFh)	Unsigned32	rw

**Subindex 0:** Der Eintrag in Subindex 0 enthält den größten unterstützten Subindex. Wert = 4.

**Subindex 1...4:** Beinhaltet den Speicherbefehl

Bei Lesezugriff Subindex 1 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, besitzen die Parameter nach dem nächsten Einschalten der Betriebsspannung wieder die alten Werte.

Unsigned32

MSB

LSB

Bits	31-2	1	0
Wert	= 0	0	1

Um eine versehentliche Speicherung der Parameter zu vermeiden, wird die Speicherung nur ausgeführt, wenn eine spezielle Signatur in den entsprechenden Subindex geschrieben wird. Die Signatur heißt "save".

Unsigned32

MSB

LSB

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: Fehlercode 0606 0000 h.

Wurde eine falsche Signatur geschrieben, verweigert das Gerät die Speicherung und antwortet mit Abbruch der Übertragung: Fehlercode 0800 0020 h.

## 8.12 Objekt 1011h: Wiederherstellung der Parameter-Standardwerte

Dieses Objekt unterstützt das Laden der Standardwerte der beschreibbaren Parameter.

Index	Subindex	Kommentar	Typ	Attr.
1011h	0	größter unterstützter Subindex	Unsigned8	ro
	1	alle Parameter wiederherstellen	Unsigned32	rw
	2	Kommunikations-Parameter wiederherstellen (Objekte: 1000h...1FFFh)	Unsigned32	rw
	3	Gerätespezifische-Parameter wiederherstellen (Objekte: 6000h...9FFFh)	Unsigned32	rw
	4	Herstellerspezifische-Parameter wiederherstellen (Objekte: 2000h...5FFFh)	Unsigned32	rw

**Subindex 0:** Der Eintrag in Subindex 0 enthält den größten unterstützten Subindex. Wert = 4.

**Subindex 1...4:** Beinhaltet den Wiederherstellungsbefehl

Bei Lesezugriff auf den Subindex 1 liefert das Gerät Informationen über seine Möglichkeiten die Standardwerte wieder herzustellen.

Bit 0 = 1 bedeutet, dass das Gerät die Wiederherstellung der Standardwerte unterstützt.

Unsigned32

	MSB	LSB
<b>Bits</b>	<b>31-1</b>	<b>0</b>
Wert	= 0	1

Um eine versehentliche Wiederherstellung der Parameterwerte zu vermeiden, wird die Wiederherstellung nur ausgeführt, wenn eine spezielle Signatur in den entsprechenden Subindex geschrieben wird. Die Signatur heißt "load".

Unsigned32

	MSB		LSB
<b>d</b>	<b>a</b>	<b>o</b>	<b>l</b>
64h	61h	6Fh	6Ch

Beim Empfang der richtigen Signatur werden die entsprechenden Standardwerte wieder hergestellt. Schlug die Wiederherstellung fehl, antwortet das Gerät mit Abbruch der Übertragung: Fehlercode 0606 0000 h.

Wurde eine falsche Signatur geschrieben, verweigert das Gerät die Wiederherstellung und antwortet mit Abbruch der Übertragung: Fehlercode 0800 0020 h.

### 8.13 Objekt 1014h: COB-ID EMERGENCY (EMCY)

Dieses Objekt zeigt die konfigurierte COB-ID für den EMCY Schreib-Dienst an.  
Standardwert = 80h + Node-ID.

EMCY Identifier, rw:

31	30	29	28	11	10	0
Valid	0	Frame	0 0000h	11-Bit CAN-ID		
MSB						LSB

Bit(s)	Beschreibung
Valid	0: EMCY existiert / ist gültig 1: EMCY existiert nicht / ist nicht gültig
30	reserviert, immer 0
Frame	0: 11-Bit CAN-ID gültig, normaler CAN Frame 1: 29-Bit CAN-ID gültig, erweiterter CAN Frame (nicht unterstützt)
11-Bit CAN-ID	11-Bit CAN-ID des normalen CAN Frames

Die Bits 0-29 dürfen nicht geändert werden während das Objekt existiert und gültig ist (Bit 31 = 0). Soll ein neuer Wert geschrieben werden, muss das Bit 31 auf 1 gesetzt werden zusammen mit dem neuen Wert. Beim Eintragen ist die Node-ID mit zu berücksichtigen.

### 8.14 Objekt 1016h: Consumer Heartbeat Time

Das Consumer Heartbeat Time Objekt definiert die zu erwartende Producer Heartbeat Zykluszeit. Die Überwachung des Heartbeat Producers beginnt mit dem Erhalt des ersten Heartbeats. Die Consumer Heartbeat Time sollte größer sein, als die entsprechende Producer Heartbeat Time. Wenn der Heartbeat nicht innerhalb der Consumer Heartbeat Time empfangen wird, wird die Emergency 8130h ausgegeben und beide Teilnehmer, Producer/Consumer, in den Zustand *PRE-OPERATIONAL* versetzt. Die Timerwerte von Producer/Consumer werden daraufhin auf 0 gesetzt.

Index	Subindex	Kommentar	Typ	Attribut
1016h	0	größter unterstützte Subindex	Unsigned8	ro
	1	Consumer Heartbeat Time	Unsigned32	rw

Consumer Heartbeat Time:

31	24	23	16	15	0
reserviert, 00h	Node-ID, Default = 1	Heartbeat time [ms], Default = 0			
MSB					LSB

Die Heartbeat time ist als Vielfaches von 1 ms anzugeben. Der Eintrag für die Node-ID entspricht der Node-ID des zu überwachenden Knotens.

## 8.15 Objekt 1017h: Producer Heartbeat Time

Das Producer Heartbeat Time Objekt definiert die Heartbeat Zykluszeit in [ms]. Der Wert 0 deaktiviert den Producer Heartbeat.

Unmittelbar nach der Konfiguration der Producer Heartbeat Time (Wert > 0) wird mit der zyklischen Übertragung der Heartbeat Nachricht begonnen.

Wurde die Producer Heartbeat Time konfiguriert, werden nach dem Einschalten des Gerätes beim Übergang in den Zustand `PRE-OPERATIONAL` bereits Heartbeat Nachrichten übertragen. In diesem Fall wird die Boot-Up-Nachricht schon als erste Heartbeat Nachricht angesehen.

Unsigned16

Producer Heartbeat Time	
Byte 0	Byte 1
$2^7$ bis $2^0$	$2^{15}$ bis $2^8$



*Es ist nicht erlaubt beide Fehler-Kontroll-Mechanismen, „Guarding Protokoll“ und „Heartbeat-Protokoll“, bei einem Knoten zur selben Zeit zu benutzen. Wenn die Heartbeat Producer Time ungleich 0 ist, wird deshalb das Heartbeat Protokoll benutzt.*

## 8.16 Objekt 1018h: Identity Objekt

Dieses Objekt enthält generelle Informationen über das Gerät.

Index	Subindex	Kommentar	Typ	Attribut
1018h	0	größter unterstützter Subindex	Unsigned8	ro
	1	Vendor-ID	Unsigned32	ro
	2	Product Code	Unsigned32	ro
	3	Revision-No.	Unsigned32	ro
	4	Serial-No.	Unsigned32	ro

**Subindex0:** Der Eintrag in Subindex 0 enthält den größten unterstützten Subindex: Wert = 4.

**Subindex1:** Bei Lesezugriff liefert das Gerät die Vendor-ID des Herstellers: 0x0000025C

**Subindex2:** Bei Lesezugriff liefert das Gerät Informationen über den Produktcode: gerätespezifisch

**Subindex3:** Bei Lesezugriff liefert das Gerät Informationen über die Revisions-Nr.: aktuelle Revisions-Nr.

**Subindex4:** Bei Lesezugriff liefert das Gerät Informationen über die Serien-Nr.: aktuelle Serien-Nr.

## 8.17 Objekt 1021h: EDS abspeichern

Über dieses Objekt kann die EDS-Datei segmentiert als ASCII-Code ausgelesen werden.

## 8.18 Objekt 1022h: EDS Speicherformat

Das Objekt zeigt das Speicherformat der über Objekt 0x1021 ausgegebenen EDS-Datei an. Default: 00h (/ISO10646/, nicht komprimiert)

## 8.19 Objekt 1029h: Verhalten im Fehlerfall

Das Objekt steuert das Verhalten wenn ein Netzwerk- oder Gerätefehler auftritt.

Index	Subindex	Kommentar	Default	Typ	Attr.
1029h	0	größter unterstützter Subindex	2	Unsigned8	ro
	1	Verhalten bei Kommunikationsfehlern, siehe Tabelle 11 auf Seite 100	00h	Unsigned8	rw
	2	Verhalten bei Hardwarefehlern, siehe Tabelle 12 auf Seite 100	01h	Unsigned8	rw

Wert	Bedeutung
00h	NMT in PRE-OPERATIONAL-Mode versetzen (nur wenn sich das Gerät im OPERATIONAL-Mode befindet)
01h	keine Änderung des NMT-Status
02h	NMT in STOPPED-Mode versetzen

## 8.20 Firmware-Update

Ein Firmware-Update ist nur mit einer geeigneten Steuerungssoftware möglich, die das CiA-Protokoll 302-3 (ab Version: 4.1.0) und Segmented-SDO-Download unterstützt.

### 8.20.1 Objekt 1F50h: Programmdaten

Mittels dieses Objekts kann ein Firmware-Update des Mess-Systems durchgeführt werden indem die neue Firmware segmentiert auf Subindex 1 geschrieben wird. Um das Update starten zu können, muss das Programm über Objekt 1F51h in den Boot-Loader-Zustand "Stopp Firmware" versetzt werden.

Index	Subindex	Kommentar	Typ	Attribut
1F50h	0	größter unterstützter Subindex	Unsigned8	ro
	1	Programm Nummer 1	Domain	rw

Schlug das Firmware-Update fehl, antwortet das Gerät mit Abbruch der Übertragung: Fehlercode 0606 0000h.

### 8.20.2 Objekt 1F51h: Programmsteuerung

Dieses Objekt wird zur Steuerung des Update-Vorgangs verwendet. Das Gerät muss sich in PRE-OPERATIONAL-Mode befinden.

Index	Subindex	Kommentar	Typ	Attribut
1F51h	0	größter unterstützter Subindex	Unsigned8	ro
	1	Programm Nummer 1	Unsigned8	rw

Wert	Bedeutung	
	Schreibzugriff	Lesezugriff
00h	Stopp Firmware	Firmware gestoppt
01h	Start Firmware <sup>1)</sup>	Firmware gestartet
02h	Reset Firmware <sup>1)</sup>	Firmware gestoppt
03h	Lösche Firmware <sup>1)</sup>	keine Firmware verfügbar

<sup>1)</sup> nur möglich, wenn die Firmware zuvor gestoppt wurde.

Ist die angefragte Aktion nicht vorhanden oder nicht möglich, wird eine Fehlermeldung als SDO Abort-Nachricht ausgegeben. Die Abort-Nachricht 0609 0030h zeigt an, dass die Aktion nicht unterstützt wird. Die Abort-Nachricht 0800 0022h zeigt an, dass die Aktion im Moment nicht ausgeführt werden kann.



### 8.20.3 Objekt 1F56h: Programm Software Identifikation

Dieses Objekt beinhaltet im Subindex 1 eine vom Mess-System generierte Checksumme der Firmware um sie eindeutig zu identifizieren.

Index	Subindex	Kommentar	Typ	Attribut
1F56h	0	größter unterstützter Subindex	Unsigned8	ro
	1	Programm Nummer 1	Unsigned32	ro

### 8.20.4 Objekt 1F57h: Programm Status

Mit Lesezugriff auf dieses Objekt kann der Zustand des Firmwarespeichers ausgelesen werden.

Index	Subindex	Kommentar	Typ	Attribut
1F57h	0	größter unterstützter Subindex	Unsigned8	ro
	1	Programm Nummer 1	Unsigned32	ro

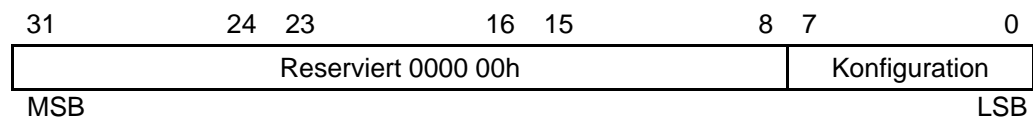
Bit	Wert	Bedeutung
0	0	Status OK, andere Bits gültig, Programmstatus gültig
	1	in Bearbeitung, andere Bits ungültig, Programmstatus ungültig
1	0	Kein Fehler aufgetreten, Firmware gültig
	1	ungültige Firmware
2...7	-	Nicht unterstützt
8...15	-	Reserviert (immer 0)
16...31	-	Nicht unterstützt

## 8.21 Objekt 1F80h: NMT Autostart

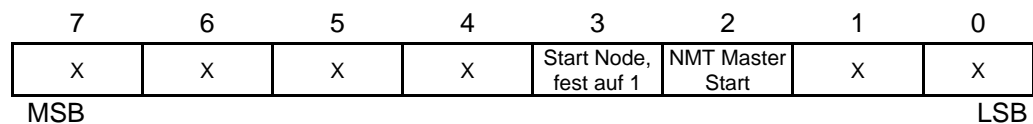
Dieses Objekt konfiguriert das Anlaufverhalten des CANopen Gerätes und legt fest, ob das Gerät automatisch nach der Initialisierung in den Zustand `OPERATIONAL` überführt werden soll:

- Bit 2, NMT Master Start = 0:  
Automatische Überführung in den Zustand `OPERATIONAL`
- Bit 2, NMT Master Start = 1; Standardeinstellung:  
Keine automatische Überführung in den Zustand `OPERATIONAL`

Bitzuordnung:



Aufschlüsselung des Konfigurationsbytes



Der Versuch ein Bit zu ändern das nicht im Mess-System vorhanden ist, veranlasst eine Abort-Nachricht (0609 0030h).

## 9 Parametrierung

M = Mandatory (zwingend)

O = Optional

C = Conditional (bedingt)

Index (h)	Objekt	Name	Datenlänge	Attr.	M/O	Seite
<b>TR Parameter</b>						
<sup>1)</sup> 2000	VAR	Mode-Umschaltung TR / CiA DS-406	Unsigned16	rw	O	61
<sup>1)</sup> 2001	VAR	TR-Betriebsparameter, Zählrichtung	Unsigned16	rw	O	62
<sup>1)</sup> 2002	VAR	TR-Gesamtmesslänge in Schritten	Unsigned32	rw	O	63
<sup>1)</sup> 2003	VAR	TR-Anzahl Umdrehungen, Zähler	Unsigned32	rw	O	64
<sup>1)</sup> 2004	VAR	TR-Anzahl Umdrehungen, Nenner	Unsigned32	rw	O	64
<sup>1)</sup> 2005	VAR	Geschwindigkeitsauflösung	Unsigned16	rw	O	67
2006	VAR	Zusätzliche Parametriermöglichkeit	Unsigned32	rw	O	67
<sup>1)</sup> 2007	VAR	Geschwindigkeitsfaktor	Unsigned16	rw	O	68
<sup>1)</sup> 2008	VAR	Geschwindigkeitsintegrationszeit	Unsigned16	rw	O	68
200A	ARRAY	Geschwindigkeit (INT32)	Integer32	ro	O	69
2010	VAR	Zustandsänderung - Faktor	Unsigned8	rw	O	69
2011	VAR	Zustandsänderung - Hysterese	Unsigned8	rw	O	70
<sup>1)</sup> 2100	VAR	COB-ID für Boot-Up Nachricht	Unsigned16	rw	O	69
<sup>1)</sup> 2101	VAR	Senden von PDO bei Node-Start	Unsigned8	rw	O	71
2110	VAR	LSS-Node ID	Unsigned8	rw	O	71
2111	VAR	LSS-BitTiming	Unsigned8	rw	O	72
<sup>3)</sup> 2200	VAR	DMF abspeichern	Domain	ro	O	72
2500	VAR	SSI-Ausgang verfügbar	Unsigned8	ro	O	72
<sup>1)</sup> 2501	VAR	SSI-Ausgangsdaten	Unsigned8	rw	O	73
<sup>1)</sup> 2502	VAR	SSI-Ausgabecode	Unsigned8	rw	O	73
<sup>1)</sup> 2503	VAR	SSI-Anzahl Datenbits	Unsigned8	rw	O	73
<sup>1)</sup> 2504	VAR	SSI-Monozeit	Unsigned16	rw	O	74
2510	VAR	Preset-Eingänge verfügbar	Unsigned8	ro	O	74
<sup>1)</sup> 2511	VAR	Preset-Wert 1	Unsigned32	rw	O	74
<sup>1)</sup> 2512	VAR	Preset-Wert 2	Unsigned32	rw	O	75
2520	VAR	Inkremental-Ausgang verfügbar	Unsigned8	ro	O	75
<sup>1)</sup> 2521	VAR	Inkremental-Anzahl Impulse	Unsigned32	rw	O	75
<sup>1)</sup> 2522	VAR	Inkremental-Phasenlage	Unsigned8	rw	O	76
<sup>1)</sup> 2523	VAR	Inkremental-K0 Verhalten	Unsigned8	rw	O	76
<sup>1)</sup> 2524	VAR	Inkremental-Pegel	Unsigned8	rw	O	77
<sup>1)</sup> 2525	VAR	Inkremental-K0 Länge	Unsigned8	rw	O	77
<sup>1)</sup> 2526	VAR	Inkremental-K0 setzen	Unsigned32	rw	O	77
2530	VAR	Anzeige verfügbar	Unsigned8	ro	O	78
<sup>1)</sup> 2531	VAR	Anzeige-Schalterfunktion 1	Unsigned8	rw	O	78
<sup>1)</sup> 2532	VAR	Anzeige-Schalterfunktion 2	Unsigned8	rw	O	78
<sup>1)</sup> 2533	VAR	Anzeige-Schalterfunktion 3	Unsigned8	rw	O	78
<sup>1)</sup> 2534	VAR	Anzeige-Schalterfunktion 4	Unsigned8	rw	O	78
<sup>1)</sup> 2535	VAR	Anzeige-Steuerung	Unsigned8	rw	O	78
<sup>1)</sup> 2536	VAR	Anzeige-Daten numerisch	Unsigned32	rw	O	79
<sup>1)</sup> 2537	VAR	Anzeige-Daten ASCII	VisibleString	rw	O	79
2538	VAR	Anzeige-Status	Unsigned8	ro	O	79
2700	VAR	TR-Produktion	Unsigned32	rw	O	79
2710	VAR	TR-Produktion	Unsigned8	rw	O	79
2720	VAR	TR-Produktion	Unsigned8	wo	O	79
2721	VAR	TR-Produktion	Unsigned16	ro	O	79

...

...

Index (h)	Objekt	Name	Datenlänge	Attr.	M/O	Seite
<b>TR Parameter</b>						
2722	VAR	TR-Produktion	Unsigned32	ro	O	79
2750	VAR	TR-Produktion	Unsigned8	rw	O	79
2751	VAR	TR-Produktion	VisibleString	rw	O	79
2752	VAR	TR-Produktion	VisibleString	rw	O	79
2753	VAR	TR-Produktion	VisibleString	rw	O	79
2754	VAR	TR-Produktion	VisibleString	rw	O	79
2755	VAR	TR-Produktion	VisibleString	rw	O	79
2756	VAR	TR-Produktion	VisibleString	rw	O	79
2757	VAR	TR-Produktion	VisibleString	rw	O	79
2758	VAR	TR-Produktion	VisibleString	rw	O	79
2759	VAR	TR-Produktion	VisibleString	rw	O	79
2760	VAR	TR-Produktion	VisibleString	rw	O	79
2761	VAR	TR-Produktion	Unsigned32	rw	O	79
<b>CiA DS-406 Parameter</b>						
<sup>1)</sup> 6000	VAR	Betriebsparameter	Unsigned16	rw	M	80
<sup>1)</sup> 6001	VAR	Mess-Schritte pro Umdrehung	Unsigned32	rw	M	81
<sup>1)</sup> 6002	VAR	Gesamtmesslänge in Schritten	Unsigned32	rw	M	81
<sup>2)</sup> 6003	VAR	Presetwert	Unsigned32	rw	M	83
6004	VAR	Positionswert	Unsigned32	ro	M	83
6030	ARRAY	Geschwindigkeitswerte	Integer16	ro	C	84
<sup>2)</sup> 6200	VAR	Cyclic-Timer	Unsigned16	rw	M	84
6400	ARRAY	Bereichs-Statusregister	Unsigned8	ro	O	85
<sup>1)</sup> 6401	ARRAY	Arbeitsbereich unterer Grenzwert	Integer32	rw	O	85
<sup>1)</sup> 6402	ARRAY	Arbeitsbereich oberer Grenzwert	Integer32	rw	O	85
<b>Diagnose</b>						
6500	VAR	Betriebsstatus	Unsigned16	ro	M	86
6501	VAR	Single-Turn Auflösung	Unsigned32	ro	M	86
6502	VAR	Anzahl der Umdrehungen	Unsigned16	ro	M	87
6503	VAR	Alarme	Unsigned16	ro	C	88
6504	VAR	Unterstützte Alarme	Unsigned16	ro	M	89
6505	VAR	Warnungen	Unsigned16	ro	C	90
6506	VAR	Unterstützte Warnungen	Unsigned16	ro	M	91
6507	VAR	Profil- und Softwareversion	Unsigned32	ro	M	92
6508	VAR	Betriebszeit	Unsigned32	ro	M	92
6509	VAR	Offsetwert	Signed32	ro	M	93
650A	ARRAY	Hersteller-Offsetwert	Signed32	ro	M	93
650B	VAR	Serien-Nummer	Unsigned32	ro	M	93

**Tabelle 9: Encoder-Profilbereich**

<sup>1)</sup> Ist sofort nach Aufruf wirksam und wird erst nach Ausführen von "**Objekt 1010h: Parameter abspeichern**" dauerhaft im EEPROM abgespeichert.

<sup>2)</sup> Ist sofort nach Aufruf wirksam und wird dauerhaft im EEPROM abgespeichert.

<sup>3)</sup> Segmentiertes Lesen

## 9.1 Objekt 2000h: Mode-Umschaltung TR / CiA DS-406

Über die Mode-Umschaltung kann gewählt werden, welche Skalierungsparameter genutzt werden sollen. Standardmäßig werden die Parameter nach dem Encoderprofil CiA DS-406 genutzt. Für besondere Anwendungen kann auf TR-Parameter umgeschaltet werden, um erweiterte Getriebefunktionen zuzulassen.

Index	0x2000
<b>Beschreibung</b>	Mode selection TR / CiA DS-406
<b>Datentyp</b>	UNSIGNED16
<b>Objektcode</b>	VARIABLE
<b>Zugriff</b>	rw
<b>PDO Mapping</b>	nein
<b>Untergrenze</b>	0x0000 = CiA DS-406 - Mode
<b>Obergrenze</b>	0x0001 = TR - Mode
<b>Default</b>	0x0000



*Es können jeweils nur die Parameter im aktiven Mode geändert werden.  
Nicht aufgeführte Objekte gelten für beide Modes.*

CiA DS-406 - Mode	TR - Mode
0x6000, Zählrichtung 0x6001, Mess-Schritte pro Umdrehung 0x6002, Gesamtmesslänge in Schritten	0x2001, Zählrichtung 0x2002, Gesamtmesslänge in Schritten 0x2003, Anzahl Umdrehungen - Zähler 0x2004, Anzahl Umdrehungen - Nenner

### 9.2 TR - Mode

#### 9.2.1 Objekt 2001h: TR-Betriebsparameter, Zählrichtung

Das Objekt mit Index 2001h unterstützt nur die Funktion für die Zählrichtung. Die Zählrichtung definiert, ob steigende oder fallende Positionswerte ausgegeben werden, wenn die Mess-System-Welle im Uhrzeigersinn oder Gegenuhrzeigersinn gedreht wird (Blickrichtung auf die Anflanschung).

Index	0x2001
Beschreibung	TR - Operating parameters
Datentyp	UNSIGNED16
Objektcode	VARIABLE
Zugriff	rw
PDO Mapping	nein
Untergrenze	0x0000 = steigend
Obergrenze	0x0001 = fallend
Default	0x0000

#### 9.2.2 Skalierungsparameter

---

***Gefahr von Körperverletzung und Sachschaden beim Wiedereinschalten des Mess-Systems nach Positionierungen im stromlosen Zustand durch Verschiebung des Nullpunktes!***

Ist die Anzahl der Umdrehungen keine 2-er Potenz oder >4096, kann, falls mehr als 512 Umdrehungen im stromlosen Zustand ausgeführt werden, der Nullpunkt des Multi-Turn Mess-Systems verloren gehen!

**⚠ WARNUNG**

**ACHTUNG**

- Sicherstellen, dass bei einem Multi-Turn Mess-System der Quotient von **Umdrehungen Zähler/Umdrehungen Nenner** eine 2er-Potenz aus der Menge  $2^0, 2^1, 2^2 \dots 2^{12}$  (1, 2, 4...4096) ist.  
oder...
- Sicherstellen, dass sich Positionierungen im stromlosen Zustand bei einem Multi-Turn Mess-System innerhalb von 512 Umdrehungen befinden.

### 9.2.2.1 Objekt 2002h: TR-Gesamtmesslänge in Schritten

Dieses Objekt legt die **Gesamtschrittzahl** des Mess-Systems fest, bevor das Mess-System wieder bei Null beginnt.

Index	0x2002
<b>Beschreibung</b>	TR - Total measuring range in steps
<b>Datentyp</b>	UNSIGNED32
<b>Objektcode</b>	VARIABLE
<b>Zugriff</b>	rw
<b>PDO Mapping</b>	nein
<b>Untergrenze</b>	16 Schritte
<b>Obergrenze</b>	4 294 967 295 = (0xFFFF FFFF) *
<b>Default</b>	<b>16777216</b>

\* Zur Darstellung von 0x100000000 ( $2^{32}$ ) kann „0“ angegeben werden

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 $2^{24}$

Der tatsächlich einzugebende Obergrenzwert für die Messlänge in Schritten ist von der Mess-System-Ausführung abhängig und kann nach untenstehender Formel berechnet werden. Da der Wert "0" bereits als Schritt gezählt wird, ist der Endwert = Messlänge in Schritten – 1.

$$\text{Gesamtmesslänge in Schritten} = \text{Anzahl Schritte pro Umdrehung} * \text{Anzahl Umdrehungen}$$

Zur Berechnung können die Parameter **Schritte/Umdr.** und **Anzahl Umdrehungen** vom Typenschild des Mess-Systems abgelesen werden.

## 9.2.2.2 Objekt 2003h - 2004h: TR-Umdrehungen Zähler / Nenner

Diese beiden Objekte zusammen, legen die **Anzahl der Umdrehungen** fest, bevor das Mess-System wieder bei Null beginnt.

Da Kommazahlen nicht immer endlich (wie z.B. 3,4) sein müssen, sondern mit unendlichen Nachkommastellen (z.B. 3,43535355358774...) behaftet sein können, wird die Umdrehungszahl als Bruch eingegeben.

### Anzahl Umdrehungen Zähler:

Index	0x2003
<b>Beschreibung</b>	TR - Number of revolutions, numerator
<b>Datentyp</b>	UNSIGNED32
<b>Objektcode</b>	VARIABLE
<b>Zugriff</b>	rw
<b>PDO Mapping</b>	nein
<b>Untergrenze Zähler</b>	1
<b>Obergrenze Zähler</b>	256000
<b>Default</b>	<b>4096</b>

### Anzahl Umdrehungen Nenner:

Index	0x2004
<b>Beschreibung</b>	TR - Number of revolutions, denominator
<b>Datentyp</b>	UNSIGNED32
<b>Objektcode</b>	VARIABLE
<b>Zugriff</b>	rw
<b>PDO Mapping</b>	nein
<b>Untergrenze Nenner</b>	1
<b>Obergrenze Nenner</b>	16384
<b>Default</b>	<b>1</b>

### Anzahl der Umdrehungen:

$$\text{Anzahl der Umdrehungen} = \frac{\text{Anzahl Umdrehungen Zähler}}{\text{Anzahl Umdrehungen Nenner}}$$



Sollten bei der Eingabe der Parametrierdaten die zulässigen Bereiche von Zähler und Nenner nicht eingehalten werden können, muss versucht werden diese entsprechend zu kürzen. Ist dies nicht möglich, kann die entsprechende Kommazahl möglicherweise nur annähernd dargestellt werden. Die sich ergebende kleine Ungenauigkeit wird bei echten Rundachsenanwendungen (Endlos-Anwendungen in eine Richtung fahrend) mit der Zeit aufaddiert.

Zur Abhilfe kann z.B. nach jedem Umlauf eine Justage durchgeführt werden, oder man passt die Mechanik bzw. Übersetzung entsprechend an.

Der Parameter **"Anzahl Schritte pro Umdrehung"** darf ebenfalls eine Kommazahl sein, jedoch nicht die **"Messlänge in Schritten"**. Das Ergebnis aus obiger Formel muss auf bzw. abgerundet werden. Der dabei entstehende Fehler verteilt sich auf die programmierte gesamte Umdrehungsanzahl und ist somit vernachlässigbar.

#### **Vorgehensweise bei Linearachsen (Vor- und Zurück-Verfahrbewegungen):**

Der Parameter **"Umdrehungen Nenner"** kann bei Linearachsen fest auf "1" programmiert werden. Der Parameter **"Umdrehungen Zähler"** wird etwas größer als die benötigte Umdrehungsanzahl programmiert. Somit ist sichergestellt, dass das Mess-System bei einer geringfügigen Überschreitung des Verfahrweges keinen Istwertsprung (Nullübergang) erzeugt. Der Einfachheit halber kann auch der volle Umdrehungsbereich des Mess-Systems programmiert werden.

### Das folgende Beispiel soll die Vorgehensweise näher erläutern:

#### **Gegeben:**

- Mess-System mit 4096 Schritte/Umdr. und max. 4096 Umdrehungen
- Auflösung 1/100 mm
- Sicherstellen, dass das Mess-System in seiner vollen Auflösung und Messlänge (4096x4096) programmiert ist:  
Messlänge in Schritten = 16777216,  
Umdrehungen Zähler = 4096  
Umdrehungen Nenner = 1  
Zu erfassende Mechanik auf Linksanschlag bringen
- Mess-System mittels Justage auf „0“ setzen
- Zu erfassende Mechanik in Endlage bringen
- Den mechanisch zurückgelegten Weg in mm vermessen
- Istposition des Mess-Systems an der angeschlossenen Steuerung ablesen

#### **Annahme:**

- zurückgelegter Weg = 2000 mm
- Mess-System-Istposition nach 2000 mm = 607682 Schritte

#### **Daraus folgt:**

Anzahl zurückgelegter Umdrehungen = 607682 Schritte / 4096 Schritte/Umdr.  
= **148,3598633 Umdrehungen**

Anzahl mm / Umdrehung = 2000 mm / 148,3598633 Umdr. = **13,48073499mm / Umdr.**

Bei 1/100mm Auflösung entspricht dies einer **Schrittzahl / Umdrehung** von **1348,073499**

#### **erforderliche Programmierungen:**

Anzahl Umdrehungen Zähler = **4096**  
Anzahl Umdrehungen Nenner = **1**

Messlänge in Schritten = Anzahl Schritte pro Umdrehung \*  $\frac{\text{Anzahl Umdrehungen Zähler}}{\text{Anzahl Umdrehungen Nenner}}$

= 1348,073499 Schritte / Umdr. \*  $\frac{4096 \text{ Umdrehungen Zähler}}{1 \text{ Umdrehung Nenner}}$

= **5521709 Schritte** (abgerundet)

### 9.2.3 Objekt 2005h: Geschwindigkeitsauflösung

Dieses Objekt gibt die Auflösung an, mit der die Geschwindigkeit berechnet und ausgegeben wird, siehe Kap.: 9.2.7 „Objekt 200Ah: Geschwindigkeit INT32“ oder 9.3.5 „Objekt 6030h: Geschwindigkeit“ auf Seite 84.

Index	0x2005
<b>Beschreibung</b>	Speed unit
<b>Datentyp</b>	UNSIGNED16
<b>Objektcode</b>	VARIABLE
<b>Zugriff</b>	rw
<b>PDO Mapping</b>	nein
<b>Default</b>	<b>100</b>

Einstellbare Auflösungen:

Wert	Auflösung der Geschwindigkeit
<b>8</b>	Schritte/ms bei 8 Bit Auflösung
<b>9</b>	Schritte/ms bei 9 Bit Auflösung
...	...
<b>18</b>	Schritte/ms bei 18 Bit Auflösung
<b>100</b>	Schritte/ms bei Skalierter Auflösung *
<b>101</b>	Schritte/10 ms bei Skalierter Auflösung *
<b>102</b>	Schritte/100 ms bei Skalierter Auflösung *
<b>103</b>	Schritte/s bei Skalierter Auflösung *
<b>200</b>	Umdrehung/Minute
<b>201</b>	Umdrehung/Sekunde

\* Skalierte Auflösung:

- CiA-DS 406-Mode = Objekt 0x6001
- TR-Mode = Ergebnis aus Objekt (0x2002 \* 0x2004) / 0x2003

Siehe Objekt 2000h: Mode-Umschaltung TR / CiA DS-406.

### 9.2.4 Objekt 2006h: Zusätzliche Parameter/Kommandos (gerätespezifisch)

Über dieses Objekt können gerätespezifische Parameter/Kommandos ausgetauscht werden.

### 9.2.5 Objekt 2007h: Geschwindigkeitsfaktor

Dieses Objekt gibt für „Objekt 2005h: Geschwindigkeitsauflösung“ den Faktorwert an, siehe Seite 67.

Index	0x2007
<b>Beschreibung</b>	Speed Factor
<b>Datentyp</b>	UNSIGNED16
<b>Objektcode</b>	VARIABLE
<b>Zugriff</b>	rw
<b>PDO Mapping</b>	nein
<b>Untergrenze</b>	1
<b>Obergrenze</b>	1000
<b>Default</b>	1

### 9.2.6 Objekt 2008h: Geschwindigkeitsintegrationszeit

Dieses Objekt gibt für „Objekt 2005h: Geschwindigkeitsauflösung“ die Integrationszeit in [ms] an, siehe Kapitel 9.2.3 auf Seite 67.

Der Parameter dient zur Berechnung der Geschwindigkeit, welche über die zyklischen Prozessdaten ausgegeben wird. Die Geschwindigkeit wird hierbei in [ (Schritte/Integrationszeit) \* Faktor ] angegeben. Hohe Integrationszeiten ermöglichen hochauflösende Messungen bei geringen Drehzahlen. Niedrige Integrationszeiten zeigen Geschwindigkeitsänderungen schneller an und sind gut geeignet für hohe Drehzahlen und große Dynamik.

Index	0x2008
<b>Beschreibung</b>	Speed Integration Time
<b>Datentyp</b>	UNSIGNED16
<b>Objektcode</b>	VARIABLE
<b>Zugriff</b>	rw
<b>PDO Mapping</b>	nein
<b>Untergrenze</b>	0 ms
<b>Obergrenze</b>	1000 ms
<b>Default</b>	16 ms

#### Beispiel

Gegeben:

- Programmierte Auflösung = 8192 Schritte pro Umdrehung
- Drehzahl = 4800 Umdrehungen pro Minute
- Integrationszeit  $t_i$  = 50 ms = 0,05 s
- Faktor = 1

Gesucht:

- Ausgabewert in (Schritte/Integrationszeit) \* Faktor

$$\text{Anzahl Schritte / s} = \frac{8192 \text{ Schritte} * 4800 \text{ Umdr.}}{\text{Umdr.} * 60 \text{ s}} = \frac{655360 \text{ Schritte}}{1}$$

$$\text{Anzahl Schritte / } t_i = \frac{655360 \text{ Schritte}}{1 \text{ s}} * 0,05 \text{ s} = 32768 \text{ Schritte}$$

$$(\text{Schritte/Integrationszeit}) * \text{Faktor} = \underline{32768 \text{ Schritte} / 50 \text{ ms}}$$

### 9.2.7 Objekt 200Ah: Geschwindigkeit INT32

Das Objekt 200Ah zeigt die Geschwindigkeit des Mess-Systems bei der in „Objekt 2005h: “ festgelegten Auflösung an und unterstützt PDO-Mapping (Standard: Schritte/ms skaliert).

Die Geschwindigkeit wird vorzeichenbehaftet, als Zweierkomplement ausgegeben:

- Zählrichtungseinstellung = steigend
  - Ausgabe positiv, bei Drehung im Uhrzeigersinn (Blickrichtung auf Anflanschung)
- Zählrichtungseinstellung = fallend
  - Ausgabe negativ, bei Drehung im Uhrzeigersinn (Blickrichtung auf Anflanschung)

Index	0x200A
Beschreibung	Speed value INT32
Datentyp	INTEGER32
Objektcode	ARRAY
Zugriff	ro
PDO Mapping	ja

Geschwindigkeitswert INT32			
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 $2^{24}$

Wird der Wertebereich der Geschwindigkeit ( $-2.147.483.648 \dots + 2.147.483.647$ ) über- oder unterschritten, werden die Grenzwerte (0x7FFF FFFF oder 0x8000 0000) ausgegeben.

### 9.2.8 Objekt 2010h: Zustandsänderung - Faktor

Wenn keine Zustandsänderung (Überschreitung der Hysterese Objekt 2011h) erfolgt, antwortet das Mess-System mit dem hier definierten Change of state – Factor multipliziert mit dem in Objekt 6200h: Cyclic-Timer hinterlegten Wert.

Index	0x2010
Beschreibung	Change of state - Factor
Datentyp	UNSIGNED8
Objektcode	VARIABLE
Zugriff	rw
PDO Mapping	nein
Wert	0 = deaktiviert (default) $\geq 1$ = aktiviert
Obergrenze	255

## 9.2.9 Objekt 2011h: Zustandsänderung - Hysterese

Das Objekt 2011h gibt die Anzahl der Schritte -1 an, die das Mess-System machen muss, bevor es eine Zustandsänderung (COS-Meldung) sendet.

Index	0x2011
Beschreibung	Change of state - Hysterese
Datentyp	UNSIGNED8
Objektcode	VARIABLE
Zugriff	rw
PDO Mapping	nein
Untergrenze	0
Obergrenze	255

## 9.2.10 Objekt 2100h: COB-ID für Boot-Up Nachricht

Dieses Objekt konfiguriert die COB-ID, welche das Mess-System beim Anlauf (Einschaltmoment/RESET NODE) ausgibt, nach dem die Initialisierung abgeschlossen wurde. Unterstützt werden die Werte 000h bis 7FFh, Standardwert = 700h.

Index	0x2100
Beschreibung	COB-ID Bootup-Message
Datentyp	UNSIGNED16
Objektcode	VARIABLE
Zugriff	rw
PDO Mapping	nein

Über Bit  $2^{15}$  kann eine Freischaltung vorgenommen werden:

- Bit  $2^{15} = 0$ :  
Geschriebener Wert in den Bits  $2^0$  bis  $2^{10}$  gültig, beim nächsten Anlauf wird die konfigurierte COB-ID verwendet.
- Bit  $2^{15} = 1$ :  
Geschriebener Wert in den Bits  $2^0$  bis  $2^{10}$  nicht gültig, beim nächsten Anlauf wird keine Boot-Up-Nachricht ausgegeben.

COB-ID für Boot-Up Nachricht			
Byte 0	Byte 1		
$2^7$ bis $2^0$	$2^{10}$ bis $2^8$	$2^{11}$ bis $2^{14}$	$2^{15}$
00h – FFh	0h – 7h	0h	0-1

### 9.2.11 Objekt 2101h: Senden von PDO bei Node-Start

Über dieses Objekt kann das einmalige senden des Mess-System Positionswertes nach dem Node-Start Kommando über TPDO1 und TPDO2 eingestellt werden.

Index	0x2101
<b>Beschreibung</b>	Send PDO at Node-Start
<b>Datentyp</b>	UNSIGNED8
<b>Objektcode</b>	VARIABLE
<b>Zugriff</b>	rw
<b>PDO Mapping</b>	nein

Bit	Funktion	Bit = 0	Bit = 1
0	TPDO1 Senden bei Node-Start	AUS	EIN (standard)
1	TPDO2 Senden bei Node-Start	AUS (standard)	EIN
2 - 7	reserviert		

### 9.2.12 Objekt 2110h: LSS Node-ID

Über dieses Objekt kann die über LSS eingestellte Node-ID („Configure Node-ID Protokoll“ Kap.: 4.7.4.1). festgelegt werden. Das Mess-System muss sich dazu im LSS-Mode befinden. Siehe auch Kap. 5.4 „Einstellen der Node-ID und Baudrate“ auf Seite 39.

Index	0x2110
<b>Beschreibung</b>	LSS NodeID
<b>Datentyp</b>	UNSIGNED8
<b>Objektcode</b>	VARIABLE
<b>Zugriff</b>	rw
<b>PDO Mapping</b>	nein
<b>Untergrenze</b>	0x01 = Node-ID 1
<b>Obergrenze</b>	0x7F = Node-ID 127

## 9.2.13 Objekt 2111h: LSS-BitTiming

Über dieses Objekt kann die über LSS eingestellte Baudrate („Configure bit timing parameters Protokoll“ Kap.: 4.7.4.2). festgelegt werden. Das Mess-System muss sich dazu im LSS-Mode befinden. Siehe auch Kap. 5.4 „Einstellen der Node-ID und Baudrate“ auf Seite 39.

Index	0x2111
Beschreibung	LSS BitTiming
Datentyp	UNSIGNED8
Objektcode	VARIABLE
Zugriff	rw
PDO Mapping	nein
Wert	0x00 = 1 Mbit/s 0x01 = 800 kbit/s 0x02 = 500 kbit/s 0x03 = 250 kbit/s 0x04 = 125 kbit/s 0x05 = 100 kbit/s 0x06 = 50 kbit/s 0x07 = 20 kbit/s 0x08 = 10 kbit/s

## 9.2.14 Objekt 2200h: DMF abspeichern

Dieses Objekt dient zu Servicezwecken.

## 9.2.15 SSI-Ausgabe

Die SSI-Ausgabe ist optional und muss hardwaretechnisch vom Mess-System unterstützt werden.

### 9.2.15.1 Objekt 2500h: SSI-Ausgang verfügbar

Dieses Objekt zeigt an, ob die optionale SSI-Schnittstelle verfügbar ist.

Index	0x2500
Beschreibung	SSI-Output Available
Datentyp	UNSIGNED8
Objektcode	VARIABLE
Zugriff	ro
PDO Mapping	nein
Wert	0x00 = SSI-Ausgabe ist nicht verfügbar 0x01 = SSI-Ausgabe ist verfügbar



### 9.2.15.2 Objekt 2501h: SSI-Ausgangsdaten

Legt die Art der Daten fest, die auf der SSI-Schnittstelle ausgegeben werden.

Index	0x2501
Beschreibung	SSI-Output - Data
Datentyp	UNSIGNED8
Objektcode	VARIABLE
Zugriff	rw
PDO Mapping	nein
Wert	0x01 = Position (32Bit Unsigned) 0x02 = Geschwindigkeit (16Bit Signed) 0x03 = Position+Geschwindigkeit (48 Bit, hintereinander)

### 9.2.15.3 Objekt 2502h: SSI-Ausgabecode

Legt den Ausgabecode für die SSI-Schnittstelle fest.

Index	0x2502
Beschreibung	SSI-Output - Code
Datentyp	UNSIGNED8
Objektcode	VARIABLE
Zugriff	rw
PDO Mapping	nein
Wert	0x00 = Gray 0x01 = Binär 0x02 = Gray gekappt

### 9.2.15.4 Objekt 2503h: SSI-Anzahl Datenbits

Legt die Anzahl der Datenbits fest, die auf der SSI-Schnittstelle ausgegeben werden.

Index	0x2503
Beschreibung	SSI-Output - Number of Data-Bits
Datentyp	UNSIGNED8
Objektcode	VARIABLE
Zugriff	rw
PDO Mapping	nein
Untergrenze	1 Bit (0x01)
Obergrenze	64 Bits (0x40)

## 9.2.15.5 Objekt 2504h: SSI-Monozeit

Dieses Objekt legt die Monozeit der SSI-Schnittstelle fest.

Index	0x2504
Beschreibung	SSI-Output - Mono-Timer
Datentyp	UNSIGNED16
Objektcode	VARIABLE
Zugriff	rw
PDO Mapping	nein
Untergrenze	4 $\mu$ s (0x0004)
Obergrenze	999 $\mu$ s (0x03E7)

## 9.2.16 Presetfunktion

**⚠️ WARNUNG**

**ACHTUNG**

***Gefahr von Körperverletzung und Sachschaden durch einen Istwertsprung bei Ausführung der Preset-Justage-Funktion!***

- Die Preset-Justage-Funktion sollte nur im Mess-System-Stillstand ausgeführt werden, bzw. muss der resultierende Istwertsprung programmtechnisch und anwendungstechnisch erlaubt sein!

### 9.2.16.1 Objekt 2510h: Preset-Eingänge verfügbar

Dieses Objekt zeigt an, ob die Presetfunktion über externe Preseteingänge verfügbar ist.

Index	0x2510
Beschreibung	Preset-Inputs Available
Datentyp	UNSIGNED8
Objektcode	VARIABLE
Zugriff	ro
PDO Mapping	nein
Wert	0x00 = Presetfunktion ist nicht verfügbar 0x01 = Presetfunktion ist verfügbar

### 9.2.16.2 Objekt 2511h: Presetwert 1

Dieses Objekt enthält den Presetwert 1 der beim beschalten des externen Preseteingangs 1 als neuer Positionswert gesetzt wird.

Index	0x2511
Beschreibung	Preset-Input Value 1
Datentyp	UNSIGNED32
Objektcode	VARIABLE
Zugriff	rw
PDO Mapping	nein
Untergrenze	0
Obergrenze	programmierte Messlänge in Schritten – 1
Default	0

### 9.2.16.3 Objekt 2512h: Presetwert 2

Dieses Objekt enthält den Presetwert 2 der beim beschalten des externen Preseteingangs 2 als neuer Positionswert gesetzt wird.

Index	0x2512
Beschreibung	Preset-Input Value 2
Datentyp	UNSIGNED32
Objektcode	VARIABLE
Zugriff	rw
PDO Mapping	nein
Untergrenze	0
Obergrenze	programmierte Messlänge in Schritten – 1
Default	0

### 9.2.17 Inkremental-Ausgabe

Die Inkremental-Ausgabe ist optional und muss hardwaretechnisch vom Mess-System unterstützt werden.

#### 9.2.17.1 Objekt 2520h: Inkremental-Ausgang verfügbar

Dieses Objekt zeigt an, ob die optionale Inkremental-Schnittstelle verfügbar ist.

Index	0x2520
Beschreibung	Incremental-Output Available
Datentyp	USINT8
Objektcode	VARIABLE
Zugriff	ro
PDO Mapping	nein
Wert	0x00 = Inkremental-Ausgabe ist nicht verfügbar 0x01 = Inkremental-Ausgabe ist verfügbar

#### 9.2.17.2 Objekt 2521h: Inkremental-Anzahl Impulse

Dieses Objekt legt die Anzahl der Impulse fest, die pro Umdrehung über die inkrementelle Schnittstelle ausgegeben werden können.

Index	0x2521
Beschreibung	Incremental-Output Pulses
Datentyp	UNSIGNED32
Objektcode	VARIABLE
Zugriff	rw
PDO Mapping	nein
Wert	4 ... 36000 *

\* Abhängig von der Hardwareausführung des Mess-Systems.

## 9.2.17.3 Objekt 2522h: Inkremental-Phasenlage

Dieses Objekt legt die Phasenlage für die Inkrementalsignale fest.

<b>Index</b>	<b>0x2522</b>
<b>Beschreibung</b>	Incremental-Output Phase
<b>Datentyp</b>	UNSIGNED8
<b>Objektcode</b>	VARIABLE
<b>Zugriff</b>	rw
<b>PDO Mapping</b>	nein
<b>Wert</b>	0x00 = K1 zu K2 90° voreilend * 0x01 = K1 zu K2 90° nacheilend *

\* Drehrichtung der Mess-System-Welle im Uhrzeigersinn mit Sicht auf den Flansch.

## 9.2.17.4 Objekt 2523h: Inkremental-K0 Verhalten

Dieses Objekt legt den Schalt-Zeitpunkt der Inkremental-Spur K0 fest.

<b>Index</b>	<b>0x2523</b>
<b>Beschreibung</b>	Incremental-Output K0-Condition
<b>Datentyp</b>	UNSIGNED8
<b>Objektcode</b>	VARIABLE
<b>Zugriff</b>	rw
<b>PDO Mapping</b>	nein
<b>Wert</b>	0x00 = K0 wenn K1 high und K2 high 0x01 = K0 wenn K1 low und K2 high 0x02 = K0 wenn K1 high und K2 low 0x03 = K0 wenn K1 low und K2 low

**Beispiel (K0-Länge = ¼ Periode):**

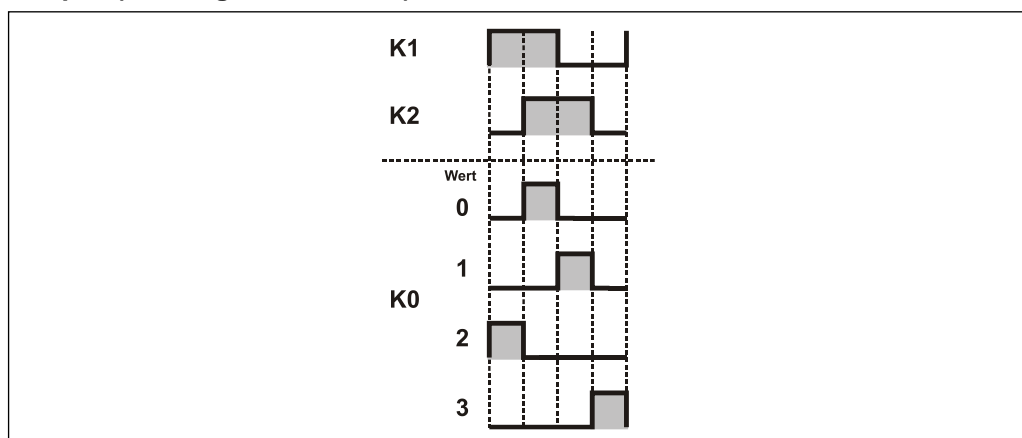


Abbildung 7: Beispiel zu K0 Condition

### 9.2.17.5 Objekt 2524h: Inkremental-Pegel

Dieses Objekt legt den Ausgangspegel der Inkremental-Signale fest.

Index	0x2524
Beschreibung	Incremental-Output Pegel
Datentyp	UNSIGNED8
Objektcode	VARIABLE
Zugriff	rw
PDO Mapping	nein
Wert	0x00 = TTL, 5 VDC Ausgangstreiber: RS422-Ausgangsstufe 0x01 = HTL, US Versorgungsspannung Ausgangstreiber: Gegentakt-Ausgangsstufe Versorgungsspannung muss > 8 VDC betragen.

### 9.2.17.6 Objekt 2525h: Inkremental-K0 Länge

Dieses Objekt legt die Länge des K0-Signals fest.

Index	0x2525
Beschreibung	Incremental-Output K0 Len
Datentyp	UNSIGNED8
Objektcode	VARIABLE
Zugriff	rw
PDO Mapping	nein
Wert	0x00 = K0 dauert ¼ Periode 0x01 = K0 dauert ½ Periode * 0x02 = K0 dauert ¾ Periode * 0x03 = K0 dauert eine Periode * 0x04 = K0 dauert zwei Perioden * 0x05 = K0 dauert vier Perioden *

\* Abhängig von der Hardwareausführung des Mess-Systems.

### 9.2.17.7 Objekt 2526h: Inkremental-K0 setzen

Dieses Objekt setzt den Nullimpuls K0 auf die aktuelle Mess-System-Position plus einen Offsetwert der beim Beschreiben übergeben wird.

Index	0x2526
Beschreibung	Incremental-Output Set K0
Datentyp	UNSIGNED32
Objektcode	VARIABLE
Zugriff	rw
PDO Mapping	nein
Untergrenze	0 = K0 wird auf die aktuelle Position gesetzt
Obergrenze	programmierte Anzahl Impulse (Kap.: 9.2.17.2)

### 9.2.18 Externe-Anzeige

Die Funktionen für eine externe Anzeige sind optional und muss hardwaretechnisch vom Mess-System unterstützt werden.

#### 9.2.18.1 Objekt 2530h: Anzeige verfügbar

Dieses Objekt legt fest, ob eine externe Anzeige verwendet wird. Die Anzeige muss hardwaretechnisch mit dem Mess-System verbunden sein.

Index	0x2530
Beschreibung	Display Available
Datentyp	UNSIGNED8
Objektcode	VARIABLE
Zugriff	ro
PDO Mapping	nein
Wert	0x00 = Externe-Anzeige ist nicht verfügbar 0x01 = Externe-Anzeige ist verfügbar

#### 9.2.18.2 Objekte 2531h ... 2534h: Anzeige-Schalterfunktion 1, 2, 3 und 4

Diese Objekte beinhalten durchgeschleifte Schaltfunktionen für das Display.

Index	0x2531, 0x2532, 0x2533, 0x2534
Beschreibung	Display Button # function
Datentyp	UNSIGNED8
Objektcode	VARIABLE
Zugriff	rw
PDO Mapping	nein

#### 9.2.18.3 Objekt 2535h: Anzeige-Steuerung

Über dieses Objekt, können displayspezifische Einstellungen vorgenommen werden.

Index	0x2535
Beschreibung	Display Control
Datentyp	UNSIGNED8
Objektcode	VARIABLE
Zugriff	rw
PDO Mapping	nein

#### 9.2.18.4 Objekt 2536h: Anzeige-Daten numerisch

Die Mess-System-Position wird numerisch als Dezimalzahl angezeigt.

Index	0x2536
Beschreibung	Display Data Numeric
Datentyp	UNSIGNED32
Objektcode	VARIABLE
Zugriff	rw
PDO Mapping	nein
Untergrenze	- 2 147 483 648 = (0x8000 0000)
Obergrenze	+ 2 147 483 647 = (0x7FFF FFFF)

#### 9.2.18.5 Objekt 2537h: Anzeige-Daten ASCII

Die Mess-System-Position wird im ASCII-Code dargestellt.

Index	0x2537
Beschreibung	Display Data ASCII
Datentyp	VISIBLE_STRING
Objektcode	VARIABLE
Zugriff	rw
PDO Mapping	nein

#### 9.2.18.6 Objekt 2538h: Anzeige-Status

Über dieses Objekt kann der Status des Displays ausgelesen werden.

Index	0x2538
Beschreibung	Display Status
Datentyp	UNSIGNED8
Objektcode	VARIABLE
Zugriff	ro
PDO Mapping	nein

#### 9.2.19 TR-Produktion

Die Objekte „TR-Produktion“ sind für herstellerepezifische Funktionen reserviert.

## 9.3 CiA DS-406 - Mode

### 9.3.1 Objekt 6000h: Betriebsparameter

Das Objekt mit Index 6000h unterstützt nur die Funktion für die Zählrichtung.

Index	0x6000
<b>Beschreibung</b>	Operating parameters
<b>Datentyp</b>	UNSIGNED16
<b>Objektcode</b>	VARIABLE
<b>Zugriff</b>	rw
<b>PDO Mapping</b>	nein

Bit	Funktion	Bit = 0	Bit = 1
0	Zählrichtung	Position steigend	Position fallend
1	reserviert		
2	Skalierungsfunktion	auf 1 gesetzt, kann nicht verändert werden!	
3 – 15	reserviert		

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

### 9.3.2 Skalierungsparameter

**Gefahr von Körperverletzung und Sachschaden beim Wiedereinschalten des Mess-Systems nach Positionierungen im stromlosen Zustand durch Verschiebung des Nullpunktes!**

Ist die Anzahl der Umdrehungen keine 2er Potenz oder >4096, kann, falls mehr als 512 Umdrehungen im stromlosen Zustand ausgeführt werden, der Nullpunkt des Multi-Turn Mess-Systems verloren gehen!

**⚠️ WARNUNG**

**ACHTUNG**

- Sicherstellen, dass bei einem Multi-Turn Mess-System die **Anzahl der Umdrehungen** eine 2er-Potenz aus der Menge  $2^0, 2^1, 2^2 \dots 2^{12}$  (1, 2, 4...4096) ist.  
oder
- Sicherstellen, dass sich Positionierungen im stromlosen Zustand bei einem Multi-Turn Mess-System innerhalb von 512 Umdrehungen befinden.



### 9.3.2.1 Objekt 6001h: Mess-Schritte pro Umdrehung

Das Objekt „Mess-Schritte pro Umdrehung“ legt die Anzahl der Schritte pro Umdrehung fest.

Index	0x6001
<b>Beschreibung</b>	Measuring units per revolution
<b>Datentyp</b>	UNSIGNED32
<b>Objektcode</b>	VARIABLE
<b>Zugriff</b>	rw
<b>PDO Mapping</b>	nein
<b>Untergrenze</b>	1 Schritt / Umdrehung
<b>Obergrenze</b>	max. Wert siehe Mess-System-Typenschild
<b>Default</b>	<b>4096</b>

Mess-Schritte 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 $2^{24}$

### 9.3.2.2 Objekt 6002h: Gesamt Messlänge in Schritten

Dieses Objekt legt die **Gesamtschrittzahl** des Mess-Systems fest, bevor das Mess-System wieder bei Null beginnt.

Index	0x6002
<b>Beschreibung</b>	Total measuring range in measuring units
<b>Datentyp</b>	UNSIGNED32
<b>Objektcode</b>	VARIABLE
<b>Zugriff</b>	rw
<b>PDO Mapping</b>	nein
<b>Untergrenze</b>	16 Schritte
<b>Obergrenze</b>	4 294 967 295 = (0xFFFF FFFF) *
<b>Default</b>	<b>16777216</b>

\* Zur Darstellung von 0x100000000 ( $2^{32}$ ) kann „0“ angegeben werden

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 $2^{24}$

Der tatsächlich einzugebende Obergrenzwert für die Gesamtmesslänge in Schritten ist von der Mess-System-Ausführung abhängig und kann nach untenstehender Formel berechnet werden. Da der Wert "0" bereits als Schritt gezählt wird, ist der Endwert = Messlänge in Schritten – 1.

$$\text{Gesamtmesslänge in Schritten} = \text{Mess-Schritte pro Umdrehung} * \text{Anzahl der Umdrehungen}$$

Zur Berechnung können die Parameter **Schritte/Umdr.** und **Anzahl Umdrehungen** vom Typenschild des Mess-Systems abgelesen werden.

Der Parameter „Anzahl der Umdrehungen“, der sich aus den Eingaben „Gesamtmesslänge in Schritten“ und „Mess-Schritte pro Umdrehung“ ergibt, hat folgende Einschränkung:

Die „Anzahl Umdrehungen“ darf eine Kommazahl sein, die sich mit einem Bruch in folgendem Bereich darstellen lässt:

$$\frac{1...256000}{1...16384} = \text{Anzahl Umdrehungen}$$

### Beispiel 1:

#### **Annahme:**

- Messlänge in Schritten = 16777216
- Schritte pro Umdrehung = 2048

#### **Daraus folgt:**

$$\frac{16777216 \text{ Schritte}}{2048 \text{ Schritte/Umdr.}} = 8192 \text{ Umdr.} = \frac{8192}{1} \text{ Umdr.} \Rightarrow \text{möglich}$$

### Beispiel 2:

#### **Annahme:**

- Messlänge in Schritten = 10000000
- Schritte pro Umdrehung = 3600

#### **Daraus folgt:**

$$\frac{10000000 \text{ Schritte}}{3600 \text{ Schritte/Umdr.}} = 2777,7 \text{ Umdr.} = \frac{25000}{9} \text{ Umdr.} \Rightarrow \text{möglich}$$

Kann die resultierende Anzahl Umdrehungen nicht in diesem Bereich dargestellt werden, so wird die „Messlänge in Schritten“ auf den nächst kleineren Wert korrigiert.



Die neu errechnete Messlänge in Schritten kann durch Rücklesen des Objektes 6002h ausgelesen werden und ist immer kleiner als die vorgegebene Messlänge. Es kann daher vorkommen, dass die tatsächlich benötigte Gesamtschrittzahl unterschritten wird und das Mess-System vor Erreichen des maximalen mechanischen Fahrweges einen Nullübergang generiert.

### 9.3.3 Objekt 6003h: Presetwert

#### ⚠️ WARNUNG

#### ACHTUNG

**Gefahr von Körperverletzung und Sachschaden durch einen Istwertsprung bei Ausführung der Preset-Justage-Funktion!**

- Die Preset-Justage-Funktion sollte nur im Mess-System-Stillstand ausgeführt werden, bzw. muss der resultierende Istwertsprung programmtechnisch und anwendungstechnisch erlaubt sein!

Die Presetfunktion wird verwendet, um den Mess-System-Wert auf einen beliebigen Positionswert innerhalb des Bereiches von 0 bis Messlänge in Schritten — 1 zu setzen. Der Ausgabe-Positionswert wird auf den Parameter "Presetwert" gesetzt, wenn auf dieses Objekt geschrieben wird.

Bei Eingaben eines ungültigen Presetwerts, antwortet das Mess-System mit dem Abort-Code: 0600 0030h.

Index	0x6003
Beschreibung	Preset value
Datentyp	UNSIGNED32
Objektcode	VARIABLE
Zugriff	rw
PDO Mapping	nein

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 $2^{24}$

### 9.3.4 Objekt 6004h: Positionswert

Das Objekt 6004h "Positionswert" definiert den Ausgabe-Positionswert und unterstützt PDO-Mapping.

Index	0x6004
Beschreibung	Position value
Datentyp	INTEGER32
Objektcode	VARIABLE
Zugriff	ro
PDO Mapping	ja

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 $2^{24}$

## 9.3.5 Objekt 6030h: Geschwindigkeit

Das Objekt 6030h zeigt im Subindex 1 die Geschwindigkeit des Mess-System bei der in „Objekt 2005h: “ festgelegten Auflösung an (Standard: Schritte/ms skaliert).

Die Geschwindigkeit wird vorzeichenbehaftet, als Zweierkomplement ausgegeben:

- Zählrichtungseinstellung = steigend
  - Ausgabe positiv, bei Drehung im Uhrzeigersinn (Blickrichtung auf Anflanschung)
- Zählrichtungseinstellung = fallend
  - Ausgabe negativ, bei Drehung im Uhrzeigersinn (Blickrichtung auf Anflanschung)

Index	0x6030
Beschreibung	Speed value
Datentyp	INTEGER16
Objektcode	ARRAY
Zugriff	ro
PDO Mapping	ja

Geschwindigkeitswert	
Byte 0	Byte 1
2 <sup>7</sup> bis 2 <sup>0</sup>	2 <sup>15</sup> bis 2 <sup>8</sup>

Wird der Wertebereich der Geschwindigkeit (-32768...+32767) über- oder unterschritten, werden die Grenzwerte (0x7FFF oder 0x8000) ausgegeben.

## 9.3.6 Objekt 6200h: Cyclic-Timer

Dieses Objekt definiert den Parameter „Cyclic-Timer“. Eine asynchrone Übertragung des Positionswertes wird eingestellt, wenn der Cyclic-Timer auf > 0 programmiert wird. Es können Werte zwischen 1 ms und 65535 ms ausgewählt werden. Standardwert = 0.

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

Wenn das Mess-System mit dem Kommando NODE-START gestartet wird und der Wert des Cyclic-Timers > 0 ist, überträgt in den Standarteinstellungen das erste Sende-Prozessdaten-Objekt (Objekt 1800h) die Mess-System-Position.

Index	0x6200
Beschreibung	Cyclic timer
Datentyp	UNSIGNED16
Objektcode	VARIABLE
Zugriff	rw
PDO Mapping	nein



*Der Event-Timer, Subindex 5 des Kommunikationsparameters 1800h, ist fest verknüpft mit dem Cyclic-Timer. Dies bedeutet, dass eine Änderung des Event-Timers sich auch im Cyclic Timer auswirkt und umgekehrt. Die Kommunikationsparameter 1801h benutzt ausschließlich seinen eigenen Timer, Zugriff über Subindex 5.*

### 9.3.7 Objekt 6400h: Bereichs-Statusregister

Mit den Objekten „Objekt 6401h: Arbeitsbereich unterer Grenzwert“ und „Objekt 6402h: Arbeitsbereich oberer Grenzwert“ zusammen, kann innerhalb des Messbereichs ein Arbeitsbereich definiert werden.

Das „Objekt 6400h: Bereichs-Statusregister“ beinhaltet in Subindex 1 den aktuellen Bereichsstatus der Mess-System-Position. Durch diese Funktion können externe Lageschalter eingespart werden.

Index	0x6400
Beschreibung	Area state register
Datentyp	UNSIGNED8
Objektcode	ARRAY
Zugriff	ro
PDO Mapping	ja

Bit	Funktion
0	reserviert
1	1 = überhalb des Arbeitsbereichs
2	1 = unterhalb des Arbeitsbereichs
3 – 7	reserviert

### 9.3.8 Objekt 6401h: Arbeitsbereich unterer Grenzwert

Das Objekt „Arbeitsbereich unterer Grenzwert“ definiert die Untergrenze des Arbeitsbereichs. Der Wert muss innerhalb des Messbereichs und unter dem „Arbeitsbereich oberer Grenzwert“ liegen.

Index	0x6401
Beschreibung	Work area low limit
Datentyp	INTEGER32
Objektcode	ARRAY
Zugriff	rw
PDO Mapping	nein
Untergrenze	- 2 147 483 648 = (0x8000 0000)
Obergrenze	+ 2 147 483 647 = (0x7FFF FFFF)

### 9.3.9 Objekt 6402h: Arbeitsbereich oberer Grenzwert

Das Objekt „Arbeitsbereich oberer Grenzwert“ definiert die Obergrenze des Arbeitsbereichs. Der Wert muss innerhalb des Messbereichs und über dem „Arbeitsbereich unterer Grenzwert“ liegen.

Index	0x6402
Beschreibung	Work area high limit
Datentyp	INTEGER32
Objektcode	ARRAY
Zugriff	rw
PDO Mapping	nein
Untergrenze	- 2 147 483 648 = (0x8000 0000)
Obergrenze	+ 2 147 483 647 = (0x7FFF FFFF)

## 9.4 Mess-System Diagnose

### 9.4.1 Objekt 6500h: Betriebsstatus

Dieses Objekt beinhaltet den Betriebsstatus des Mess-Systems und Informationen über die intern programmierten Parameter.

Index	0x6500
Beschreibung	Operating status
Datentyp	UNSIGNED16
Objektcode	VARIABLE
Zugriff	ro
PDO Mapping	nein

Bit	Funktion	Bit = 0	Bit = 1
0	Zählrichtung	steigend	fallend
1	reserviert		
2	Konstant		X
3 - 15	reserviert		

### 9.4.2 Objekt 6501h: Single-Turn Auflösung

Das Objekt 6501h enthält die maximale Anzahl der Mess-Schritte pro Umdrehung welche durch das Mess-System ausgegeben werden können.

Index	0x6501
Beschreibung	Singleturn resolution
Datentyp	UNSIGNED32
Objektcode	VARIABLE
Zugriff	ro
PDO Mapping	nein

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: abhängig von der Kapazität, siehe Typenschild

### 9.4.3 Objekt 6502h: Anzahl der Umdrehungen

Dieses Objekt beinhaltet die Anzahl der Umdrehungen, welche das Mess-System ausgeben kann.

Index	0x6502
Beschreibung	Number of distinguishable revolutions
Datentyp	UNSIGNED32
Objektcode	VARIABLE
Zugriff	ro
PDO Mapping	nein

Für ein Multi-Turn Mess-System ergibt sich aus der Anzahl der Umdrehungen und der Single-Turn Auflösung die Gesamtmesslänge, welche sich nach der unten stehenden Formel berechnen lässt. Die max. Anzahl der Umdrehungen ist 256.000 (18 Bit).

$\text{Gesamtmesslänge in Schritten} = \text{Anzahl der Umdrehungen} \times \text{Single-Turn Auflösung}$
---

Standardwert: 59392 = E800h Umdrehungen.

Da dieses Objekt nur einen 16 Bit-Wert speichern kann, wird der höherwertige Anteil der Zahl  $\boxed{3}$ E800h (256.000) nicht dargestellt.

## 9.4.4 Objekt 6503h: Alarme

Das Objekt 6503h beinhaltet zusätzlich zur „Emergency-Meldung“ weitere Alarm-Meldungen welche über das PDO-Mapping auch in den Prozessdaten eingeblendet werden können. Ein Alarm wird gesetzt, wenn eine Störung im Mess-System zum falschen Positionswert führen könnte. Falls ein Alarm auftritt, wird das zugehörige Bit solange auf logisch „High“ gesetzt, bis der Alarm gelöscht und das Mess-System bereit ist, einen richtigen Positionswert auszugeben.

<b>Index</b>	<b>0x6503</b>
<b>Beschreibung</b>	Alarms
<b>Datentyp</b>	UNSIGNED16
<b>Objektcode</b>	VARIABLE
<b>Zugriff</b>	ro
<b>PDO Mapping</b>	nein

Bit	Funktion	Bit = 0	Bit = 1
0	Positionsfehler	Nein	Ja
1	Reserviert für weitere Verwendung		
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	Reserviert für weitere Verwendung		
14	herstellerspezifische Funktionen		
15	herstellerspezifische Funktionen		

### Positionsfehler

Das Bit wird gesetzt, wenn das Mess-System eine Störung des Systems erkennt.

### EE-PROM-Fehler

Das Mess-System hat eine falsche Checksumme im EEPROM-Bereich erkannt, oder ein Schreibvorgang in das EEPROM konnte nicht erfolgreich abgeschlossen werden.



### 9.4.5 Objekt 6504h: Unterstützte Alarmer

Das Objekt 6504h beinhaltet Informationen über die Alarmer, die durch das Mess-System unterstützt werden.

<b>Index</b>	<b>0x6504</b>
<b>Beschreibung</b>	Supported alarms
<b>Datentyp</b>	UNSIGNED16
<b>Objektcode</b>	VARIABLE
<b>Zugriff</b>	ro
<b>PDO Mapping</b>	nein

Bit	Funktion	Bit = 0	Bit = 1
0	Positionsfehler	Nein	Ja
1	Reserviert für weitere Verwendung		
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	Reserviert für weitere Verwendung		
14	herstellerspezifische Funktionen		
15	herstellerspezifische Funktionen		

## 9.4.6 Objekt 6505h: Warnungen

Das Objekt 6505h beinhaltet Informationen über die Warnungen und zeigen an, dass bestimmte Betriebsparameter des Mess-Systems überschritten wurden. Im Gegensatz zu den Alarmen beinhalten die Warnungen keine Anzeige für fehlerhafte Positionswerte.

Index	0x6505
<b>Beschreibung</b>	Warnings
<b>Datentyp</b>	UNSIGNED16
<b>Objektcode</b>	VARIABLE
<b>Zugriff</b>	ro
<b>PDO Mapping</b>	nein

Bit	Funktion	Bit = 0	Bit = 1
0	Geschwindigkeitswarnung	Nein	Ja
1	Reserviert für weitere Verwendung		
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	Temperaturwarnung	Nein	Ja
13	herstellerspezifische Funktionen		
14	herstellerspezifische Funktionen		
15	herstellerspezifische Funktionen		

### Grenzwerte:

Die Grenzwerte können dem gerätespezifischen Datenblatt entnommen werden.



*Alle Warnungen werden automatisch gelöscht, sobald sich die Betriebsparameter wieder im Normalbereich befinden.*

### 9.4.7 Objekt 6506h: Unterstützte Warnungen

Das Objekt 6506h beinhaltet Informationen über die Warnungen, die durch das Mess-System unterstützt werden.

Index	0x6506
<b>Beschreibung</b>	Supported warnings
<b>Datentyp</b>	UNSIGNED16
<b>Objektcode</b>	VARIABLE
<b>Zugriff</b>	ro
<b>PDO Mapping</b>	nein

Bit	Funktion	Bit = 0	Bit = 1
0	Geschwindigkeitswarnung	Nein	Ja
1	Reserviert für weitere Verwendung		
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	Temperaturwarnung	Nein	Ja
13	herstellerspezifische Funktionen		
14	herstellerspezifische Funktionen		
15	herstellerspezifische Funktionen		

## 9.4.8 Objekt 6507h: Profil- und Softwareversion

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

Index	0x6507
<b>Beschreibung</b>	Profile and software version
<b>Datentyp</b>	UNSIGNED32
<b>Objektcode</b>	VARIABLE
<b>Zugriff</b>	ro
<b>PDO Mapping</b>	nein

z.B.:    Profilversion:            3.02  
           Binärcode:            0000 0011    0000 0010  
           Hexadezimal:        03            02

Die zweiten 16 Bit enthalten den Index der Softwareversion aus Objekt 100Ah.

z.B.:    Softwareversions-Index: 1.06  
           Binärcode:            0000 0001    0000 0110  
           Hexadezimal:        01            06

Die Softwareversion ohne Versionsindex ist in Objekt 100Ah enthalten, siehe Seite 49.

Profilversion		Softwareversions-Index	
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 $2^8$

## 9.4.9 Objekt 6508h: Betriebsdauer

Dieses Objekt speichert die Betriebsdauer in den nichtflüchtigen Speicher solange das Mess-System mit Strom versorgt wird.

Die Betriebsdauer wird in 0,1 Std. pro Digit erfasst.

Index	0x6508
<b>Beschreibung</b>	Operating time
<b>Datentyp</b>	UNSIGNED32
<b>Objektcode</b>	VARIABLE
<b>Zugriff</b>	ro
<b>PDO Mapping</b>	nein

#### 9.4.10 Objekt 6509h: Offsetwert

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

Index	0x6509
Beschreibung	Offset value
Datentyp	UNSIGNED32
Objektcode	VARIABLE
Zugriff	ro
PDO Mapping	nein

#### 9.4.11 Objekt 650Ah: Hersteller-Offsetwert

Dieses Objekt enthält in Subindex 1 den internen Mess-System Offsetwert.

Index	0x650A
Beschreibung	Manufacturer offset value
Datentyp	UNSIGNED32
Objektcode	ARRAY
Zugriff	ro
PDO Mapping	nein

#### 9.4.12 Objekt 650Bh: Serien-Nummer

Dieses Objekt enthält die aktuelle Serien-Nr. des Gerätes und entspricht dem Identity-Objekt 1018h, Subindex 4.

Index	0x650B
Beschreibung	Serial number
Datentyp	UNSIGNED32
Objektcode	VARIABLE
Zugriff	ro
PDO Mapping	nein

### 10 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 Objekt 1003h, Byte 0-1		Fehler-Register Objekt 1001h	Alarme Objekt 6503		Warnungen Objekt 6505		0

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

Wenn das Mess-System einen internen Fehler erkennt, wird eine Emergency-Meldung mit dem Fehlercode des Objekts 1003h (Vordefiniertes Fehlerfeld) und dem Fehler-Register (Objekt 1001h) übertragen.

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

## 11 Fehlerursachen und Abhilfen

### 11.1 Optische Anzeigen

Die Anzahl, Lage und Zuordnung der Status-LEDs ist abhängig von der Geräteausführung und kann der gerätespezifischen Steckerbelegung entnommen werden.



*Es kann über das „Objekt 1029h: Verhalten im Fehlerfall“ gesteuert werden, ob das Mess-System beim Auftreten eines Fehlers vom OPERATIONAL in den PRE-OPERATIONAL- oder den STOPPED-Mode überführt werden soll, siehe Seite 55. In diesem Fall ist die abgesetzte Emergency-Meldung auszuwerten, siehe Emergency-Fehlercodes ab Seite 99.*

#### 11.1.1 Mess-System mit zwei Status-LEDs

##### LED1, Device Status:

LED-Status	Ursache	Abhilfe
OFF	Spannungsversorgung fehlt oder wurde unterschritten	<ul style="list-style-type: none"> <li>- Spannungsversorgung, Verdrahtung prüfen.</li> <li>- Liegt die Spannungsversorgung im zulässigen Bereich?</li> </ul>
	Hardwarefehler, Mess-System defekt	<ul style="list-style-type: none"> <li>- Mess-System tauschen.</li> </ul>
ON (grün)	Gerät funktioniert fehlerfrei	-
Blinking (grün)	Gerät wird ausserhalb der Spezifikation betrieben (z.B. Drehgeschwindigkeit, Temperatur)	<ul style="list-style-type: none"> <li>- Parameter und Spezifikationen prüfen.</li> </ul>
ON (rot)	Gerät nicht betriebsbereit	<ul style="list-style-type: none"> <li>- Versorgungsspannung ausschalten, danach wieder einschalten. Führt diese Maßnahme nicht zum Erfolg, muss das Mess-System ausgetauscht werden.</li> </ul>
Blinking (rot)	Gerät hat einen Fehler festgestellt (z.B. EEPROM-Zugriff fehlerhaft)	
Flickering (rot)	Gerät hat einen Defekt festgestellt (z.B. Positionsfehler)	

### LED2, Net Status:

LED-Status	Ursache	Abhilfe
OFF	Spannungsversorgung fehlt oder wurde unterschritten	<ul style="list-style-type: none"> <li>- Spannungsversorgung, Verdrahtung prüfen.</li> <li>- Liegt die Spannungsversorgung im zulässigen Bereich?</li> </ul>
	Hardwarefehler, Mess-System defekt	<ul style="list-style-type: none"> <li>- Mess-System tauschen.</li> </ul>
ON (grün)	Gerät befindet sich im <i>OPERATIONAL</i> Zustand	-
Blinking (grün)	Gerät befindet sich im <i>PRE-OPERATIONAL</i> Zustand	<ul style="list-style-type: none"> <li>- Kein Fehler, Mess-System kann in den <i>OPERATIONAL</i> Mode überführt werden.</li> </ul>
Single flash (grün)	Gerät befindet sich im <i>STOPPED</i> Zustand	
Flickering (grün)	Mess-System befindet sich im LSS-Configuration-Zustand	<ul style="list-style-type: none"> <li>- Kein Fehler, Mess-System kann in den LSS-Waiting-Zustand überführt werden.</li> </ul>
ON (rot)	Bus offline aufgrund von <ul style="list-style-type: none"> <li>- vertauschter CAN-Leitungen</li> <li>- unterbrochener CAN-Leitungen</li> <li>- doppelter NODE-ID im Netzwerk</li> </ul>	<ul style="list-style-type: none"> <li>- Buskabel überprüfen.</li> <li>- Steckverbindungen überprüfen.</li> <li>- Eingestellte Baudrate muss mit der Master-Baudrate übereinstimmen.</li> </ul>
Single flash (rot)	zu viele Fehler im CAN-Controller	<ul style="list-style-type: none"> <li>- Versorgungsspannung ausschalten, danach wieder einschalten.</li> <li>- Physikalischen Aufbau des Bus-Systems überprüfen.</li> </ul>
Double flash (rot)	Node Guarding-Fehler	<ul style="list-style-type: none"> <li>- Generelle Busauslastung <math>\leq 85\%</math> !</li> <li>- Versuchen, die Baudrate zu erhöhen.</li> <li>- Zykluszeit über die Objekte 100Ch und 100Dh für das Node-Guarding-Protokoll erhöhen.</li> <li>- Versorgungsspannung ausschalten, danach wieder einschalten.</li> </ul>
	Heartbeat	<ul style="list-style-type: none"> <li>- Generelle Busauslastung <math>\leq 85\%</math> !</li> <li>- Versuchen, die Baudrate zu erhöhen.</li> <li>- Zykluszeit über die Objekte 1016h bzw. 1017h für das Heartbeat-Protokoll anpassen.</li> </ul>



### 11.1.2 Mess-System mit einer Status-LED

LED-Status	Ursache	Abhilfe
OFF	Spannungsversorgung fehlt oder wurde unterschritten	<ul style="list-style-type: none"> <li>- Spannungsversorgung, Verdrahtung prüfen.</li> <li>- Liegt die Spannungsversorgung im zulässigen Bereich?</li> </ul>
	Hardwarefehler, Mess-System defekt	<ul style="list-style-type: none"> <li>- Mess-System tauschen.</li> </ul>
ON (grün)	Gerät befindet sich im <i>OPERATIONAL</i> Zustand	-
Blinking (grün)	Gerät befindet sich im <i>PRE-OPERATIONAL</i> Zustand	<ul style="list-style-type: none"> <li>- Kein Fehler, Mess-System kann in den <i>OPERATIONAL</i> Mode überführt werden.</li> </ul>
Single flash (grün)	Gerät befindet sich im <i>STOPPED</i> Zustand	
Flickering (grün)	Mess-System befindet sich im LSS-Configuration-Zustand	<ul style="list-style-type: none"> <li>- Kein Fehler, Mess-System kann in den LSS-Waiting-Zustand überführt werden.</li> </ul>
ON (rot)	Bus offline aufgrund von <ul style="list-style-type: none"> <li>- vertauschter CAN-Leitungen</li> <li>- unterbrochener CAN-Leitungen</li> <li>- doppelter NODE-ID im Netzwerk</li> </ul>	<ul style="list-style-type: none"> <li>- Buskabel überprüfen.</li> <li>- Steckverbindungen überprüfen.</li> <li>- Eingestellte Baudrate muss mit der Master-Baudrate übereinstimmen.</li> </ul>
Single flash (rot)	zu viele Fehler im CAN-Controller	<ul style="list-style-type: none"> <li>- Versorgungsspannung ausschalten, danach wieder einschalten.</li> <li>- Physikalischen Aufbau des Bus-Systems überprüfen.</li> </ul>
Double flash (rot)	Node Guarding-Fehler	<ul style="list-style-type: none"> <li>- Generelle Busauslastung <math>\leq 85\%</math> !</li> <li>- Versuchen, die Baudrate zu erhöhen.</li> <li>- Zykluszeit über die Objekte 100Ch und 100Dh für das Node-Guarding-Protokoll erhöhen.</li> <li>- Versorgungsspannung ausschalten, danach wieder einschalten.</li> </ul>
	Heartbeat	<ul style="list-style-type: none"> <li>- Generelle Busauslastung <math>\leq 85\%</math> !</li> <li>- Versuchen, die Baudrate zu erhöhen.</li> <li>- Zykluszeit über die Objekte 1016h bzw. 1017h für das Heartbeat-Protokoll anpassen.</li> </ul>
Flickering (rot)	Hardwarefehler, Mess-System defekt	<ul style="list-style-type: none"> <li>- Mess-System tauschen.</li> </ul>

## 11.2 SDO-Fehlercodes

Im Fall eines Fehlers (SDO Response CCD = 0x80) enthält der Datenbereich einen 4-Byte-Fehlercode. Folgende Fehler-Codes werden vom Mess-System unterstützt:

Fehlercode	Bedeutung	Abhilfe
0x0503 0000	Togglebit hat sich nicht geändert.	Beim segmentierten Übertragen eines SDOs ist ein Telegramm nicht richtig übertragen worden. - Vorgang wiederholen
0x0504 0001	Kein gültiger oder unbekannter Kommando-Code (CCD)	Liste der gültigen CCD's siehe Tabelle 3 auf Seite 17.
0x0601 0001	Lesezugriff auf ein Objekt, welches nur beschrieben werden kann.	Falscher Kommando-Code (CCD), es sind nur Schreib - Kommandos (0x2x) erlaubt, siehe Tabelle 3 auf Seite 17.
0x0601 0002	Schreibzugriff auf ein Objekt, welches nur gelesen werden kann.	Falscher Kommando-Code (CCD), es sind nur Lese - Kommandos (0x4x) erlaubt, siehe Tabelle 3 auf Seite 17.
0x0602 0000	Objekt im Objekt-Verzeichnis nicht vorhanden.	Gültige Objekte siehe Tabelle 8 und Tabelle 9 auf Seite 46 und 60.
0x0604 0041	Objekt kann nicht gemappt werden	Unterstützte Mapping-Objekte siehe Kapitel 7.2.1 auf Seite 44.
0x0604 0042	Anzahl und Länge der gemappten Objekte übersteigt die zulässige PDO Länge	Überprüfen - Mapping Objekte $\leq 8$ Byte Datenlänge pro TPDO - Anzahl Mapping Objekte $\leq 2$ pro TPDO
0x0607 0010	Datentyp bzw. Länge der Service-Parameter stimmt nicht.	Der benutzte Kommando-Code (CCD) stimmt nicht mit der Datenlänge des übertragenen Objekts überein. Vergleiche Kommando-Codes Seite 17 mit den Objekten, siehe Tabelle 8 und Tabelle 9 auf Seite 46 und 60.
0x0607 0012	Datentyp bzw. Länge der Service-Parameter zu groß.	Der benutzte Kommando-Code (CCD) ist länger als das übertragene Objekt. Vergleiche Kommando-Codes Seite 17 mit den Objekten, siehe Tabelle 8 und Tabelle 9 auf Seite 46 und 60.
0x0607 0013	Datentyp bzw. Länge der Service-Parameter zu klein.	Der benutzte Kommando-Code (CCD) ist kürzer als das übertragene Objekt. Vergleiche Kommando-Codes Seite 17 mit den Objekten, siehe Tabelle 8 und Tabelle 9 auf Seite 46 und 60.
0x0609 0011	Subindex nicht vorhanden.	Überprüfen, welche Subindexe das entsprechende Objekt unterstützt.
0x0609 0030	Ungültiger Parameterwert (nur Download)	Zulässigen Wertebereich für das entsprechende Objekt überprüfen.
0x0609 0031	Gesendeter Parameterwert zu groß	Gültiger Bereich des Objekts beachten.
0x0609 0032	Gesendeter Parameterwert zu klein	Gültiger Bereich des Objekts beachten.
0x0800 0020	Daten können nicht übertragen bzw. gespeichert werden	Falsche Signatur beim Abspeichern/ Wiederherstellen der Parameter geschrieben, siehe Objekte 1010h/1011h, Seite 51/52.
0x0800 0021	Daten können aufgrund der lokalen Ansteuerung nicht gesendet oder gespeichert werden.	Falsche Mode-Ansteuerung, siehe Objekt 2000h: Mode-Umschaltung TR / CiA DS-406 auf Seite 61 oder falscher Zustand für Objekt 1F51h: Programmsteuerung auf Seite 56.
0x0800 0022	Daten können aufgrund des Gerätezustandes (Status) nicht übertragen bzw. gespeichert werden	Falsche Vorgehensweise bei der Mapping-Konfiguration vorgenommen, siehe Ändern der Mappingeinstellung auf Seite 44.
0x0800 0024	Keine Daten verfügbar	Hinweis, dass keine Fehler mehr vorhanden sind, bei Lesezugriff auf Objekt 1003h Subindex 01, siehe Seite 48.

**Tabelle 10: SDO-Fehlercodes**

## 11.3 Emergency-Fehlercodes

Emergency-Meldungen werden beim Auftreten einer geräteinternen Störung ausgelöst, Übertragungsformat siehe Kapitel „Emergency-Meldung“, Seite 94.

Die Fehleranzeige wird über die Objekte

- Fehlerregister 0x1001, siehe Seite 47 und
  - Vordefiniertes Fehlerfeld 0x1003, siehe Seite 48
- vorgenommen.

### 11.3.1 Objekt 1001h: Fehlerregister

Das Fehlerregister zeigt bitkodiert den Fehlerzustand des Mess-Systems an. Es können auch mehrere Fehler gleichzeitig durch ein gesetztes Bit angezeigt werden. Der Fehlercode des zuletzt aufgetretenen Fehlers wird in Objekt 0x1003, Subindex 1 hinterlegt, die Anzahl der Fehler im Subindex 0. Im Moment des Auftretens wird ein Fehler durch eine EMCY-Nachricht signalisiert. Durch Lesen des Objekts 1001h wird der zuletzt gespeicherte Fehler in Objekt 0x1003, Subindex 0 gelöscht. Jede weitere Leseanforderung löscht einen weiteren Fehler aus der Liste. Mit Löschen des letzten Fehlers wird das Fehlerregister zurückgesetzt und eine EMCY-Nachricht mit Fehlercode „0x000“ übertragen.

Bit	Bedeutung
0	generischer Fehler
1	0
2	0
3	0
4	Kommunikationsfehler (Überlauf, Fehlerstatus)
5	0
6	0
7	0

### 11.3.2 Objekt 1003h: Vordefiniertes Fehlerfeld, Bits 0 – 15

Über das Emergency-Objekt wird immer nur der zuletzt aufgetretene Fehler angezeigt. Für jede EMCY-Nachricht die gelöscht wurde, wird eine Emergency-Meldung mit Fehlercode „0x0000“ übertragen. Das Ergebnis kann dem Objekt 0x1003 entnommen werden. Wenn kein Fehler mehr vorliegt, zeigt auch das Fehlerregister keinen Fehler mehr an.

Die Fehlerliste in Objekt 0x1003 kann auf verschiedene Arten gelöscht werden:

1. Schreiben des Wertes „0“ auf Subindex 0 im Objekt 0x1003
2. Ausführen des NMT-Dienstes „Reset Communication“, Kommando 0x82
3. Durch Lesen des Objekts 0x1001, nach dem der letzte Fehler gelöscht wurde

Fehlercode	Bedeutung	Abhilfe
0x0000	Fehler rückgesetzt / kein Fehler	-
0x5000	Sensor/Hardware-Fehler	-
0x8100	Kommunikationsfehler, die vom CAN-Controller ausgelöst werden. (BUS-OFF-Zustand)	<ul style="list-style-type: none"> <li>- Knoten zurücksetzen mit Kommando 0x81, danach Knoten neu starten mit Kommando 0x01</li> <li>- Versuchen, das Gerät neu zu starten durch Spannung AUS/EIN.</li> </ul>
0x8130	Node Guarding-Fehler	<ul style="list-style-type: none"> <li>- Generelle Busauslastung <math>\leq 85\%</math> !</li> <li>- Versuchen, die Baudrate zu erhöhen.</li> <li>- Zykluszeit über die Objekte 100Ch und 100Dh für das Node-Guarding-Protokoll erhöhen.</li> <li>- Versuchen, das Gerät neu zu starten durch Spannung AUS/EIN.</li> </ul>
	Heartbeat Fehler	<ul style="list-style-type: none"> <li>- Generelle Busauslastung <math>\leq 85\%</math> !</li> <li>- Versuchen, die Baudrate zu erhöhen.</li> <li>- Zykluszeit über die Objekte 1016h bzw. 1017h für das Heartbeat-Protokoll anpassen.</li> </ul>

**Tabelle 11: Emergency-Fehlercodes**

### 11.4 Alarm-Meldungen

Über das Objekt 6503h werden zusätzlich zur Emergency-Meldung weitere Alarm-Meldungen ausgegeben. Das entsprechende Fehlerbit wird gelöscht, wenn der Fehler nicht mehr anliegt.

Fehler	Ursache	Abhilfe
Bit 0 = 1, Positionsfehler	Ausfall von Abtastelementen im Mess-System	Versorgungsspannung eventuell ausschalten, danach wieder einschalten. Wenn der Fehler trotz dieser Maßnahme wiederholt auftritt, muss das Mess-System getauscht werden.
Bit 12 = 1, EE-PROM-Fehler	Speicherbereich im internen EEPROM defekt	

**Tabelle 12: Alarm-Meldungen**

## 11.5 Sonstige Störungen

Störung	Ursache	Abhilfe
Positionssprünge des Mess-Systems	starke Vibrationen	Vibrationen, Schläge und Stöße z.B. an Pressen, werden mit so genannten „Schockmodulen“ gedämpft. Wenn der Fehler trotz dieser Maßnahmen wiederholt auftritt, muss das Mess-System getauscht werden.
	elektrische Störungen EMV	Gegen elektrische Störungen helfen eventuell isolierende Flansche und Kupplungen aus Kunststoff, sowie Kabel mit paarweise verdrehten Adern für Datenleitungen. Die Schirmung und die Leitungsführung müssen nach den Aufbaurichtlinien für das jeweilige Feldbus-System ausgeführt sein.
	übermäßige axiale und radiale Belastung der Welle oder einen Defekt der Abtastung.	Kupplungen vermeiden mechanische Belastungen der Welle. Wenn der Fehler trotz dieser Maßnahme weiterhin auftritt, muss das Mess-System getauscht werden.

**Tabelle 13: Sonstige Störungen**



**CAN**open®

# Encoder

## Series:

- 362

- 582

- 802

- 1102

- \_ Additional safety instructions
- \_ Installation
- \_ Commissioning
- \_ Parameterization
- \_ Cause of faults and remedies

**User Manual  
Interface**

---

## 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)

---

### Copyright protection

This Manual, including the illustrations contained therein, is subject to copyright protection. Use of this Manual by third parties in contravention of copyright regulations is not permitted. Reproduction, translation as well as electronic and photographic archiving and modification require the written content of the manufacturer. Violations shall be subject to claims for damages.

---

### Subject to modifications

The right to make any changes in the interest of technical progress is reserved.

---

### Document information

Release date / Rev. date:	03/06/2024
Document / Rev. no.:	TR-ECE-BA-DGB-0147 v07
File name:	TR-ECE-BA-DGB-0147-07.docx
Author:	STB

---

### Font styles

*Italic* or **bold** font styles are used for the title of a document or are used for highlighting.

`Courier` font displays text, which is visible on the display or screen and software menu selections.

" < " > " indicates keys on your computer keyboard (such as <RETURN>).

---

### Brand names

CANopen® and CiA® are registered community trademarks of CAN in Automation e.V.

---



# Contents

<b>Revision index .....</b>	<b>109</b>
<b>1 General information .....</b>	<b>110</b>
1.1 Applicability .....	110
1.2 References.....	111
1.3 Abbreviations and definitions.....	112
<b>2 Additional safety instructions .....</b>	<b>113</b>
2.1 Definition of symbols and instructions .....	113
2.2 Additional instructions for proper use .....	113
<b>3 Optional interface variants .....</b>	<b>114</b>
<b>4 CANopen information .....</b>	<b>115</b>
4.1 CANopen – Communication profile .....	116
4.2 Process- and Service-Data-Objects .....	117
4.3 Object Dictionary .....	118
4.4 CANopen default identifier.....	118
4.5 Transmission of SDO messages .....	119
4.5.1 SDO message format .....	119
4.5.2 Read SDO.....	121
4.5.3 Write SDO .....	122
4.6 Network management, NMT .....	123
4.6.1 Network management services .....	124
4.6.1.1 NMT device control services.....	124
4.6.1.2 NMT Node / Life guarding services.....	125
4.7 Layer setting services (LSS) and protocols .....	126
4.7.1 Finite state automaton, FSA .....	127
4.7.2 Transmission of LSS services .....	128
4.7.2.1 LSS message format.....	128
4.7.3 Switch mode protocols.....	129
4.7.3.1 Switch state global protocol .....	129
4.7.3.2 Switch state selective protocol .....	129
4.7.4 Configuration protocols .....	130
4.7.4.1 Configure Node-ID protocol .....	130
4.7.4.2 Configure bit timing parameters protocol .....	131
4.7.4.3 Activate bit timing parameters protocol.....	132
4.7.4.4 Store configuration protocol .....	132
4.7.5 Inquire LSS address protocols.....	133
4.7.5.1 Inquire identity Vendor-ID protocol.....	133
4.7.5.2 Inquire identity Product-Code protocol.....	133
4.7.5.3 Inquire identity Revision-Number protocol .....	134
4.7.5.4 Inquire identity Serial-Number protocol.....	134
4.7.6 Inquire Node-ID protocol.....	135
4.7.7 Identification protocols .....	136
4.7.7.1 LSS identify remote slave protocol.....	136
4.7.7.2 LSS identify slave protocol.....	136
4.7.7.3 LSS identify non-configured remote slave protocol .....	137
4.7.7.4 LSS identify non-configured slave protocol.....	137
4.8 Device profile .....	138

<b>5 Installation / Preparation for start-up .....</b>	<b>139</b>
5.1 Connection – notes .....	140
5.2 Bus termination .....	140
5.3 Switching on the supply voltage .....	140
5.4 Setting the Node-ID and Baud rate .....	141
5.4.1 Setting by means of DIP switches .....	141
5.4.2 Setting by means of LSS services .....	141
5.4.2.1 Configuration of the Node-ID .....	141
5.4.2.2 Configuration of the Baud rate .....	142
5.4.3 Setting by means of SDOs.....	142
<b>6 Commissioning.....</b>	<b>143</b>
6.1 CAN – interface.....	143
6.1.1 EDS file .....	143
6.1.2 Bus status .....	143
<b>7 Communication profile .....</b>	<b>144</b>
7.1 Structure of the communication parameter, 1800h-1801h .....	144
7.2 Structure of the mapping parameter, 1A00h-1A01h.....	145
7.2.1 Procedure for re-mapping .....	146
7.3 Transmission of the measuring system position value .....	146
7.3.1 1st Transmit Process-Data-Object.....	147
7.3.2 2nd Transmit Process-Data-Object .....	147
<b>8 Communication specific standard objects (CiA DS-301) .....</b>	<b>148</b>
8.1 Object 1000h: Device type.....	149
8.2 Object 1001h: Error register .....	149
8.3 Object 1002h: Manufacturer status register .....	150
8.4 Object 1003h: Pre-defined error field .....	150
8.5 Object 1005h: COB-ID SYNC message .....	151
8.6 Object 1008h: Device name .....	151
8.7 Object 1009h: Hardware version .....	151
8.8 Object 100Ah: Software version .....	151
8.9 Object 100Ch: Guard time .....	152
8.10 Object 100Dh: Life time factor .....	152
8.11 Object 1010h: Store parameters.....	153
8.12 Object 1011h: Restore default parameter values .....	154
8.13 Object 1014h: COB-ID EMERGENCY (EMCY).....	155
8.14 Object 1016h: Consumer heartbeat time.....	155
8.15 Object 1017h: Producer heartbeat time.....	156
8.16 Object 1018h: Identity Object .....	156
8.17 Object 1021h: Store EDS .....	157
8.18 Object 1022h: Store format.....	157
8.19 Object 1029h: Error behavior object.....	157

8.20 Firmware update .....	158
8.20.1 Object 1F50h: Program data .....	158
8.20.2 Object 1F51h: Program control.....	158
8.20.3 Object 1F56h: Program software identification.....	159
8.20.4 Object 1F57h: Flash status identification.....	159
8.21 Object 1F80h: NMT Autostart .....	160
<b>9 Parameterization.....</b>	<b>161</b>
9.1 Object 2000h: Mode selection TR / CiA DS-406 .....	163
9.2 TR - Mode .....	164
9.2.1 Object 2001h: TR-Operating parameters, code sequence.....	164
9.2.2 Scaling parameters .....	164
9.2.2.1 Object 2002h: TR-Total measuring range in steps .....	165
9.2.2.2 Object 2003h - 2004h: TR-Number of revolutions, numerator / denominator .....	166
9.2.3 Object 2005h: Speed unit .....	169
9.2.4 Object 2006h: Additional Parameter/Commands (device specific) .....	169
9.2.5 Object 2007h: Speed Factor .....	170
9.2.6 Object 2008h: Speed Integration Time .....	170
9.2.7 Object 200Ah: Speed value INT32 .....	171
9.2.8 Object 2010h: Change of state - Factor.....	171
9.2.9 Object 2011h: Change of state - Hysteresis.....	172
9.2.10 Object 2100h: COB-ID for bootup message .....	172
9.2.11 Object 2101h: Send PDO at Node-Start.....	173
9.2.12 Object 2110h: LSS Node-ID .....	173
9.2.13 Object 2111h: LSS BitTiming.....	174
9.2.14 Object 2200h: Store DMF .....	174
9.2.15 SSI-Output .....	174
9.2.15.1 Object 2500h: SSI-Output Available .....	174
9.2.15.2 Object 2501h: SSI-Output - Data .....	175
9.2.15.3 Object 2502h: SSI-Output - Code .....	175
9.2.15.4 Object 2503h: SSI-Output - Number of Data-Bits .....	175
9.2.15.5 Object 2504h: SSI-Output - Mono-Timer .....	176
9.2.16 Preset function .....	176
9.2.16.1 Object 2510h: Preset-Inputs Available.....	176
9.2.16.2 Object 2511h: Preset-Input Value 1 .....	176
9.2.16.3 Object 2512h: Preset-Input Value 2 .....	177
9.2.17 Incremental-Output .....	177
9.2.17.1 Object 2520h: Incremental-Output Available .....	177
9.2.17.2 Object 2521h: Incremental-Output Pulses .....	177
9.2.17.3 Object 2522h: Incremental-Output Phase.....	178
9.2.17.4 Object 2523h: Incremental-Output K0-Condition .....	178
9.2.17.5 Object 2524h: Incremental-Output Level .....	179
9.2.17.6 Object 2525h: Incremental-Output K0 Len.....	179
9.2.17.7 Object 2526h: Incremental-Output Set K0 .....	179
9.2.18 External Display .....	180
9.2.18.1 Object 2530h: Display available.....	180
9.2.18.2 Objects 2531h ... 2534h: Display Button 1, 2, 3 and 4 function .....	180
9.2.18.3 Object 2535h: Display Control .....	180
9.2.18.4 Object 2536h: Display Data Numeric .....	181
9.2.18.5 Object 2537h: Display Data ASCII .....	181
9.2.18.6 Object 2538h: Display Status.....	181
9.2.19 TR-Production.....	181
9.3 CiA DS-406 - Mode.....	182
9.3.1 Object 6000h: Operating parameters .....	182
9.3.2 Scaling parameters .....	182
9.3.2.1 Object 6001h: Measuring units per revolution.....	183
9.3.2.2 Object 6002h: Total measuring range in measuring units.....	183

---

9.3.3 Object 6003h: Preset value.....	185
9.3.4 Object 6004h: Position value .....	185
9.3.5 Object 6030h: Speed value.....	186
9.3.6 Object 6200h: Cyclic timer .....	186
9.3.7 Object 6400h: Area state register .....	187
9.3.8 Object 6401h: Work area low limit .....	187
9.3.9 Object 6402h: Work area high limit.....	187
9.4 Measuring system diagnostics.....	188
9.4.1 Object 6500h: Operating status .....	188
9.4.2 Object 6501h: Single turn resolution.....	188
9.4.3 Object 6502h: Number of distinguishable revolutions .....	189
9.4.4 Object 6503h: Alarms .....	190
9.4.5 Object 6504h: Supported alarms .....	191
9.4.6 Object 6505h: Warnings .....	192
9.4.7 Object 6506h: Supported warnings .....	193
9.4.8 Object 6507h: Profile and software version .....	194
9.4.9 Object 6508h: Operating time .....	194
9.4.10 Object 6509h: Offset value .....	195
9.4.11 Object 650Ah: Manufacturer offset value .....	195
9.4.12 Object 650Bh: Serial number.....	195
<b>10 Emergency Message .....</b>	<b>196</b>
<b>11 Causes of faults and remedies .....</b>	<b>197</b>
11.1 Optical displays.....	197
11.1.1 Measuring system with two status LEDs .....	197
11.1.2 Measuring system with one status LED.....	199
11.2 SDO Error codes .....	200
11.3 Emergency Error codes .....	201
11.3.1 Object 1001h: Error register .....	201
11.3.2 Object 1003h: Pre-defined Error field, bits 0 – 15 .....	202
11.4 Alarm messages .....	202
11.5 Other faults .....	203

## Revision index

Revision	Date	Index
First release	05/22/2019	00
Additional SSI interface added	10/31/2019	01
Additional incremental interface added	03/06/2020	02
Object 6003h: Preset value adapted	11/06/2020	03
Chapter "Other faults" no twisted pair wires for supply	01/27/2022	04
Objects 200Ah, 2010h and 2011h added	08/31/2022	05
- Validity for measuring system series 362 added - Error code 5000h added - Chapter "Measuring system with one status LED" added	07/05/2023	06
Chepter: 10 „Emergency Message“ byte 3 to 6 edited	03/06/2024	07

## 1 General information

The User Manual includes the following topics:

- Safety instructions in addition to the basic safety instructions defined in the Assembly Instructions
- Installation
- Commissioning
- Parameterization
- Causes of faults and remedies

As the documentation is arranged in a modular structure, this User Manual is supplementary to other documentation, such as product datasheets, dimensional drawings, leaflets and the assembly instructions etc.

The User Manual may be included in the customer's specific delivery package or it may be requested separately.

### 1.1 Applicability

This User Manual applies exclusively to measuring system models according to the following type designation code with **CANopen** interface and optional additional Interfaces:

- 362
- 582
- 802
- 1102

The products are labeled with affixed nameplates and are components of a system.

The following documentation therefore also applies:

- see chapter "Other applicable documents" in the Assembly Instructions
  - Series 362: [www.tr-electronic.com/f/TR-ECE-BA-DGB-0108](http://www.tr-electronic.com/f/TR-ECE-BA-DGB-0108)
  - Series 582: [www.tr-electronic.com/f/TR-ECE-BA-DGB-0035](http://www.tr-electronic.com/f/TR-ECE-BA-DGB-0035)
  - Series 802: [www.tr-electronic.com/f/TR-ECE-BA-DGB-0075](http://www.tr-electronic.com/f/TR-ECE-BA-DGB-0075)
  - Series 1102: [www.tr-electronic.com/f/TR-ECE-BA-DGB-0081](http://www.tr-electronic.com/f/TR-ECE-BA-DGB-0081)
- Product data sheets
  - Series 362: [www.tr-electronic.com/s/S025378](http://www.tr-electronic.com/s/S025378)
  - Series 582: [www.tr-electronic.com/s/S025379](http://www.tr-electronic.com/s/S025379)
  - Series 802: [www.tr-electronic.com/s/S025380](http://www.tr-electronic.com/s/S025380)
  - Series 1102: [www.tr-electronic.com/s/S025381](http://www.tr-electronic.com/s/S025381)

## 1.2 References

1.	ISO 11898: Road Vehicles Interchange of Digital Information - Controller Area Network (CAN) for high-speed Communication, November 1993
2.	Robert Bosch GmbH, CAN Specification 2.0 Part A and B, September 1991
3.	CiA DS-201 V1.1, CAN in the OSI Reference Model, February 1996
4.	CiA DS-202-1 V1.1, CMS Service Specification, February 1996
5.	CiA DS-202-2 V1.1, CMS Protocol Specification, February 1996
6.	CiA DS-202-3 V1.1, CMS Encoding Rules, February 1996
7.	CiA DS-203-1 V1.1, NMT Service Specification, February 1996
8.	CiA DS-203-2 V1.1, NMT Protocol Specification, February 1996
9.	CiA DS-204-1 V1.1, DBT Service Specification, February 1996
10.	CiA DS-204-2 V1.1, DBT Protocol Specification, February 1996
11.	CiA DS-206 V1.1, Recommended Layer Naming Conventions, February 1996
12.	CiA DS-207 V1.1, Application Layer Naming Conventions, February 1996
13.	CiA DS-301 V4.2, CANopen Communication Profile based on CAL, December 2007
14.	CiA DS-302 V4.1, Additional application layer functions, February 2009
15.	CiA DS-303-3 V1.3, Indicator specification, August 2006
16.	CiA DS-305 V2.2.5, Layer Setting Services (LSS) and Protocols, November 2010
17.	CiA DS-406 V3.2, CANopen Profile for Encoder, December 2006

### 1.3 Abbreviations and definitions

EMC	<b>E</b> lectro <b>M</b> agnetic <b>C</b> ompatibility
CAL	CAN Application Layer. The application layer for CAN-based networks as specified by CiA in Draft Standard 201 ... 207.
CAN	Controller Area Network. Data link layer protocol for serial communication as specified in ISO 11898.
CiA	CAN in Automation international manufacturer and user organization e.V.: non-profit association for Controller Area Network (CAN).
CMS	CAN-based Message Specification. One of the service elements of the application layer in the CAN Reference Model.
COB	Communication Object. (CAN Message) A unit of transportation in a CAN Network. Data must be sent across a Network inside a COB.
COB-ID	COB-Identifier. Identifies a COB uniquely in a Network. The identifier determines the priority of that COB in the MAC sub-layer too.
DBT	Distributor. One of the service elements of the application in the CAN Reference Model. It is the responsibility of the DBT to distribute COB-ID's to the COB's that are used by CMS.
EDS	<b>E</b> lectronic- <b>D</b> ata- <b>S</b> heet
FSA	Finite state automata. State machine to control LSS services.
LSS	Layer Setting Services. Services and protocols for the configuration of the Node-ID and Baud rate about the CAN Network.
NMT	Network Management. One of the service elements of the application in the CAN Reference Model. It performs initialization, configuration and error handling in a CAN network.
PDO	Process Data Object. Object for data exchange between several devices.
SDO	Service Data Object. Peer to peer communication with access to the Object Dictionary of a device.
SSI	<b>S</b> ynchronous- <b>S</b> erial- <b>I</b> nterface



## 2 Additional safety instructions

### 2.1 Definition of symbols and instructions



means that death or serious injury can occur if the required precautions are not met.

---



means that minor injuries can occur if the required precautions are not met.

---

---

**NOTICE**

means that damage to property can occur if the required precautions are not met.

---



indicates important information or features and application tips for the product used.

---

### 2.2 Additional instructions for proper use

The measurement system is designed for operation with CANopen networks according to the International Standard ISO/DIS 11898 and 11519-1 up to max. 1 Mbit/s. The profile corresponds to the **"CANopen Device Profile for Encoder CiA DS-406 V2.0A"**.

The technical guidelines for the structure of the CANopen network from the CAN User Organization CiA are always to be observed in order to ensure safe operation.

### 3 Optional interface variants

The functions and the connection technology varies at interface variants. Only the device-specific data sheets, pin assignments and technical drawings should be used.

Only the functions, parameters and options from this user manual which also are supported by the measuring system, are valid. The optional functionalities are indicated in an appropriate place with "optional".

Which options by the measuring system are supported, points can be derived by the followings:

- Type of the pin assignment
- Corresponding details on the type plate
- Performance range of the necessary EDS file
- Firmware no.
- Declaration between TR electronic and the customer

## 4 CANopen information

CANopen was developed by the CiA and is standardized since at the end of 2002 in the European standard EN 50325-4.

As communication method CANopen uses the layers 1 and 2 of the CAN standard which was developed originally for the use in road vehicles (ISO 11898-2). In the automation technology these are extended by the recommendations of the CiA industry association with regard to the pin assignment and transmission rates. In the area of the application layer CiA has developed the standard CAL (CAN Application Layer).

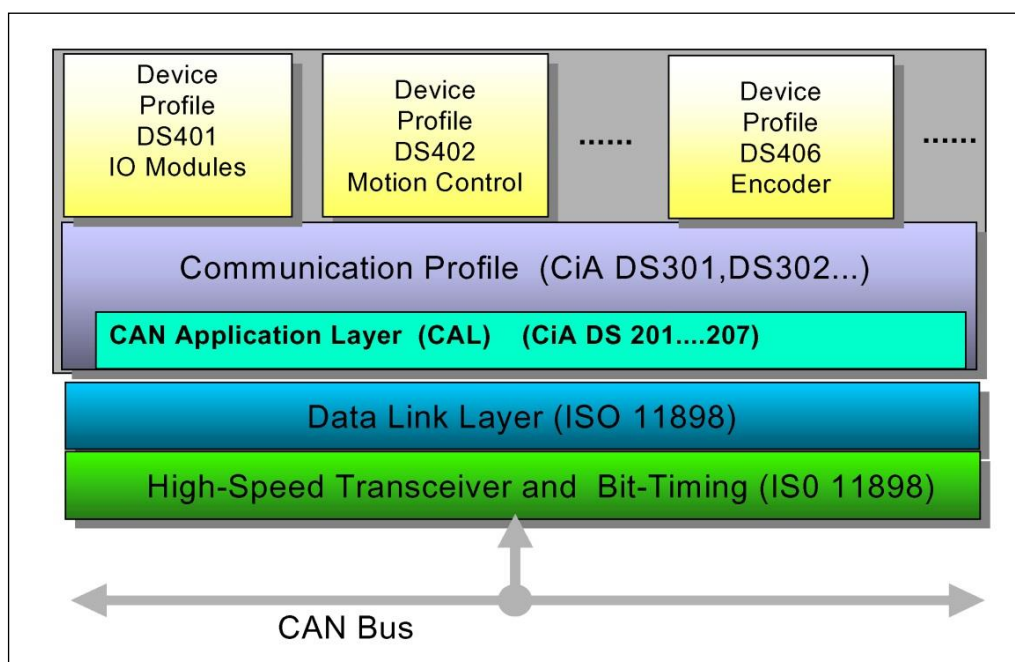


Figure 1: CANopen classified in the ISO/OSI reference model

In case of CANopen at first the communication profile as well as a "Build instructions" for device profiles was developed, in which with the structure of the object dictionary and the general coding rules the common denominator of all device profiles is defined.

## 4.1 CANopen – Communication profile

The CANopen communication profile (defined in CiA DS-301) regulates the devices data exchange. Here real time data (e.g. position value) and parameter data (e.g. code sequence) will be differentiated. To the data types, which are different from the character, CANopen assigns respectively suitable communication elements.

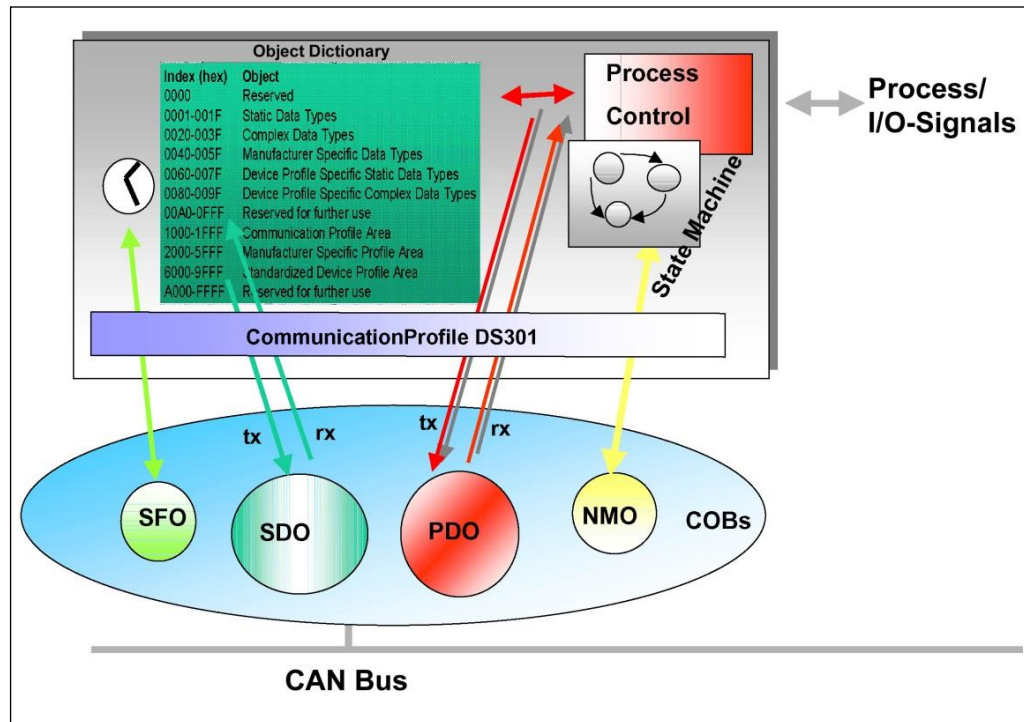


Figure 2: Communication profile

### Special Function Object (SFO)

- Synchronization (SYNC)
- Emergency (EMCY) Protocol

### Network Management Object (NMO)

e.g.

- Life / Node-Guarding
- Boot-Up,...
- Error Control Protocol

## 4.2 Process- and Service-Data-Objects

### Process-Data-Object (PDO)

Process-Data-Objects manage the process data exchange, e.g. the cyclical transmission of the position value.

The process data exchange with the CANopen PDOs is "CAN pure", therefore without protocol overhead. All broadcast characteristics of CAN remain unchanged. A message can be received and evaluated by all devices at the same time.

From the measuring system the two transmitting process data objects 1800h for asynchronous (event-driven) position transmission and 1801h for the synchronous (upon request) position transmission are used.

### Service-Data-Object (SDO)

Service-Data-Objects manage the parameter data exchange, e.g. the non-cyclical execution of the Preset function.

For parameter data of arbitrary size with the SDO an efficient communication mechanism is available. For this between the configuration master and the connected devices a service data channel for the parameter communication is available. The device parameters can be written with only one telegram handshake into the object dictionary of the devices or can be read out from this.

### Important characteristics of the SDO and PDO

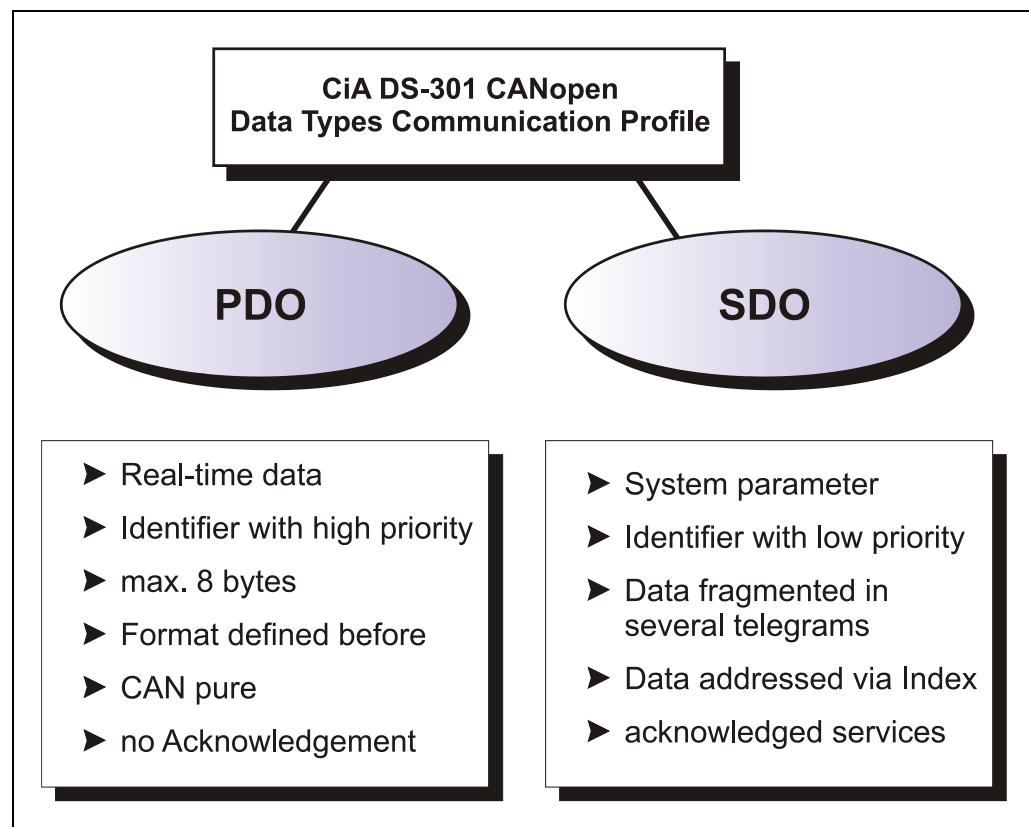


Figure 3: Comparison of PDO/SDO characteristics

### 4.3 Object Dictionary

The object dictionary structures the data of a CANopen device in a clear tabular arrangement. It contains all device parameters as well as all current process data, which are accessible thereby also about the SDO.

Index	Object	
0000 <sub>h</sub>	not used	Common to all devices
0001 <sub>h</sub> - 025F <sub>h</sub>	Data type definitions	
0260 <sub>h</sub> - 0FFF <sub>h</sub>	Reserved	
1000 <sub>h</sub> - 1FFF <sub>h</sub>	Communication profile area	
2000 <sub>h</sub> - 5FFF <sub>h</sub>	Manufacturer specific profile area	Device specific
6000 <sub>h</sub> - 9FFF <sub>h</sub>	Standardized device profile area	
A000 <sub>h</sub> - BFFF <sub>h</sub>	Standardized interface profile area	
C000 <sub>h</sub> - FFFF <sub>h</sub>	Reserved	

Figure 4: Structure of the Object Dictionary

### 4.4 CANopen default identifier

CANopen devices can be used without configuration in a CANopen network. Just the setting of a bus address and the baud rate is required. From this node address the identifier allocation for the communication channels is derived.

$$\text{COB-Identifier} = \text{Function Code} + \text{Node-ID}$$

10				0						
1	2	3	4	1	2	3	4	5	6	7
Function Code				Node-ID = Adjustment of the address switches + 1 or LSS settings						

#### Examples

Object	Function Code	COB-ID	Index Communication Parameter
NMT	0000bin	0	–
SYNC	0001bin	80h	1005
PDO1 (tx)	0011bin	181h – 1FFh	1800h

## 4.5 Transmission of SDO messages

The transmission of SDO messages is done by the CMS “Multiplexed Domain” protocol (CIA DS202-2).

With SDOs objects from the object dictionary can be read or written. It is an acknowledged service. The so-called **SDO client** specifies in its request the parameter, the access method (read/write) and if necessary the value. The so-called **SDO server** performs the write or read access and answers the request with a response. In the error case an error code gives information about the cause of error. Transmit-SDO and Receive-SDO are distinguished by their function codes.

The measuring system (slave) corresponds to the SDO server and uses the following function codes:

Function codes	COB-ID	Meaning
11 (1011 bin)	0x580 + Node ID	Slave → SDO Client
12 (1100 bin)	0x600 + Node ID	SDO Client → Slave

Table 1: COB-IDs for Service Data Object (SDO)

### 4.5.1 SDO message format

The data field with max. 8 byte length of a CAN message is used by a SDO as follows:

CCD	Index		Sub-Index	Data			
Byte 0	Byte 1 Low	Byte 2 High	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7

Table 2: SDO message

The **command code (CCD)** identifies whether the SDO is to be read or written. In addition with a writing order, the number of bytes which can be written is encoded in the CCD.

At the SDO response the CCD reports whether the request was successful. In the case of a reading order the CCD gives additionally information about the number of bytes, which could be read:

CCD	Meaning	Valid for
0x22	Write n bytes	SDO Request
0x23	Write 4 bytes	SDO Request
0x2B	Write 2 bytes	SDO Request
0x2F	Write 1 byte	SDO Request
0x60	Writing successfully	SDO Response
0x80	Error	SDO Response
0x40	Reading request	SDO Request
0x43	4 byte data read	SDO response upon reading request
0x4B	2 byte data read	SDO response upon reading request
0x4F	1 byte data read	SDO response upon reading request

Table 3: SDO command codes

In the case of an error (SDO response CCD = 0x80) the data field contains a 4-byte error code, which gives information about the error cause. Meaning of the error codes, see Table 10 on page 200.

### Segment Protocol, Data segmentation

---

Some objects contain data which are larger than 4 bytes. To be able to read these data, the "Segment Protocol" must be used.

As a usual SDO service, at first the read operation is started with the command code = 0x40. About the response the number of data segments and the number of bytes to be read is reported. With following reading requests the individual data segments can be read. A data segment consists respectively of 7 bytes.

Example of reading a data segment:

Telegram 1

CCD	Meaning	Valid for
0x40	Reading request, initiation	SDO Request
0x41	1 data segment available The number of bytes which can be read is indicated in the bytes 4 to 7.	SDO Response

Telegram 2

CCD	Meaning	Valid for
0x60	Reading request	SDO Request
0x01	No further data segment available. The bytes 1 to 7 contain the requested data.	SDO Response



## 4.5.2 Read SDO

Initiate Domain Upload

**Request Protocol format:**

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

Read SDOs								
Byte	0	1	2	3	4	5	6	7
Contents	Code	Index		Sub-Index	Data 0	Data 1	Data 2	Data 3
	40h	Low	High	Byte	0	0	0	0

The Read SDO telegram has to be send to the slave.

The slave answers with the following telegram:

**Response Protocol format:**

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

Read SDOs								
Byte	0	1	2	3	4	5	6	7
Contents	Code	Index		Sub-Index	Data 0	Data 1	Data 2	Data 3
	4xh	Low	High	Byte	Data	Data	Data	Data

**Format Byte 0:**

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

n = number of data bytes (bytes 4-7) that does not contain data

If only 1 data byte (Data 0) contains data the value of byte 0 is "4FH".

If byte 0 = 80h the transfer has been aborted.

### 4.5.3 Write SDO

Initiate Domain Download

**Request Protocol format:**

$$\text{COB-Identifier} = 600\text{h} + \text{Node-ID}$$

Write SDOs								
Byte	0	1	2	3	4	5	6	7
Contents	Code	Index		Sub-Index	Data 0	Data 1	Data 2	Data 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 = number of data bytes (bytes 4-7) that does not contain data.

If only 1 data byte (Data 0) contains data the value of byte 0 is "2FH".

The Write SDO telegram has to be send to the slave.

The slave answers with the following telegram:

**Response Protocol format:**

$$\text{COB-Identifier} = 580\text{h} + \text{Node-ID}$$

Read SDOs								
Byte	0	1	2	3	4	5	6	7
Contents	Code	Index		Sub-Index	Data 0	Data 1	Data 2	Data 3
	60h	Low	High	Byte	0	0	0	0

If byte 0 = 80h the transfer has been aborted.

## 4.6 Network management, NMT

The network management supports a simplified Boot-Up of the net. With only one telegram for example all devices can be switched into the Operational condition.

After Power on the measuring system is first in the "Pre-Operational" condition (2).

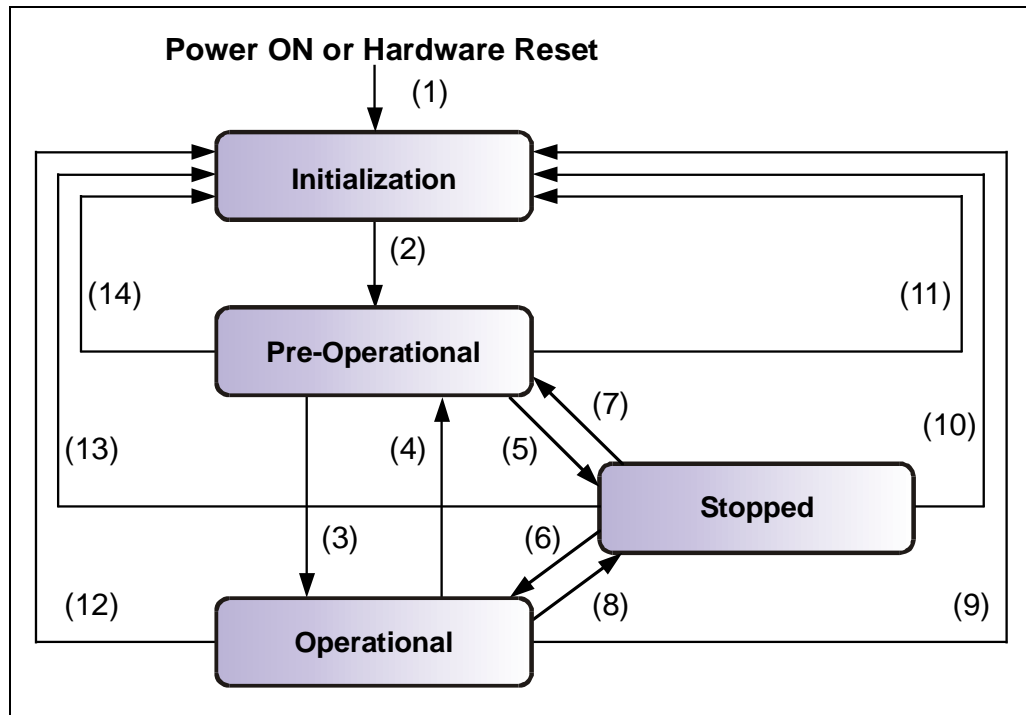


Figure 5: Boot-Up mechanism of the network management

State	Description
(1)	At Power on the initialization state is entered autonomously
(2)	Initialization finished - enter PRE-OPERATIONAL automatically
(3),(6)	Start_Remote_Node --> Operational
(4),(7)	Enter_PRE-OPERATIONAL_State --> Pre-Operational
(5),(8)	Stop_Remote_Node
(9),(10),(11)	Reset_Node
(12),(13),(14)	Reset_Communication

#### 4.6.1 Network management services

The **network management (NMT)** has the function to initialize, start, stop and monitor nodes of a CANopen network.

NMT services are initiated by a **NMT master**, which identifies individual nodes (**NMT slave**) about their Node-ID. A NMT message with the Node ID 0 refers to **all** NMT slaves.

**The measuring system corresponds to a NMT slave.**

##### 4.6.1.1 NMT device control services

The NMT services for device control use the **COB-ID 0** and get thus the highest priority.

By the data field of the CAN message only the first two bytes are used:

CCD	Node ID
Byte 0	Byte 1

The following commands are defined:

CCD	Meaning	State
-	At Power on the initialization state is entered autonomously	(1)
-	Initialization finished - enter <code>PRE-OPERATIONAL</code> automatically	(2)
0x01	<b>Start Remote Node</b> Node is switched into the <code>OPERATIONAL</code> state and the normal net-operation is started.	(3),(6)
0x02	<b>Stop Remote Node</b> Node is switched into the <code>STOPPED</code> state and the communication is stopped. An active connecting monitoring remains active.	(5),(8)
0x80	<b>Enter PRE-OPERATIONAL</b> Node is switched into the <code>PRE-OPERATIONAL</code> state. All messages can be used, but no PDOs.	(4),(7)
0x81	<b>Reset Node</b> Set values of the profile parameters of the object on default values. Afterwards transition into the <code>RESET COMMUNICATION</code> state.	(9),(10), (11)
0x82	<b>Reset Communication</b> Node is switched into the <code>RESET COMMUNICATION</code> state. Afterwards transition into the <code>INITIALIZATION</code> state, first state after Power on.	(12),(13), (14)

**Table 4: NMT device control services**

#### 4.6.1.2 NMT Node / Life guarding services

With the Node/Life guarding a NMT master can detect the failure of a NMT slave and/or a NMT slave can detect the failure of a NMT master:

- **Node Guarding and Life Guarding:**  
With these services a NMT master monitors a NMT slave

At the **Node Guarding** the NMT master requests the state of a NMT slave in regular intervals. The toggle bit 2<sup>7</sup> in the “Node Guarding Protocol” toggles after each request:

Example:

0x85, 0x05, 0x85 ... --> no error

0x85, 0x05, 0x05 ... --> error

Additionally if the **Life Guarding** is active, the NMT slave requests the state of a NMT master in regular intervals, otherwise the slave changes into the PRE-OPERATIONAL state.

The NMT services for Node/Live guarding use the function code 1110 bin: **COB-ID 0x700+Node ID**.

Index	Description	
0x100C	<b>Guard Time [ms]</b>	At termination of the time interval
0x100D	<b>Life Time Factor</b>	<b>Life Time = Guard Time x Life Time Factor [ms]</b> the NMT slave expects a state request by the master.  Guard Time = 0: No monitoring active Life Time = 0: Life guarding disabled

**Table 5: Parameter for NMT services**

### 4.7 Layer setting services (LSS) and protocols

The LSS-services and protocols, documented in CiA DS-305 V2.2.5, are used to inquire or to change the settings of several parameters of the data link layer and application layer of a LSS slave by a LSS master via the CAN network.

Following parameters are supported:

- Node-ID
- Baud rate
- LSS address compliant to the identity object (1018h)

Thus it isn't necessary to adjust the Node-ID or Baud rate by means of DIP-switches. Access to the LSS slave is made thereby by its LSS address, consisting of:

- Vendor-ID
- Product-Code
- Revision-No. and
- Serial-No.

The measuring system supports the following services:

Switch state services

- Switch state selective
- Switch state global

Configuration services

- Configure Node-ID
- Configure bit timing parameters
- Activate bit timing parameters
- Store configured parameters

Inquiry services

- Inquire LSS address
- Inquire Node-ID

Identification services

- LSS identify remote slave
- LSS identify slave
- LSS identify non-configured remote slave
- LSS identify non-configured slave

#### 4.7.1 Finite state automaton, FSA

The FSA corresponds to a state machine and defines the behavior of a LSS slave. The state machine is controlled by LSS COBs produced by the LSS master, or NMT COBs produced by the NMT master, or local NMT state transitions.

The LSS FSA supports the following states:

- (0) Initial: Pseudo state, indicating the activation of the FSA
- (1) LSS waiting: In this state, all services are supported as defined below
- (2) LSS configuration: In this state, all services are supported as defined below
- (3) Final: Pseudo state, indicating the deactivation of the FSA

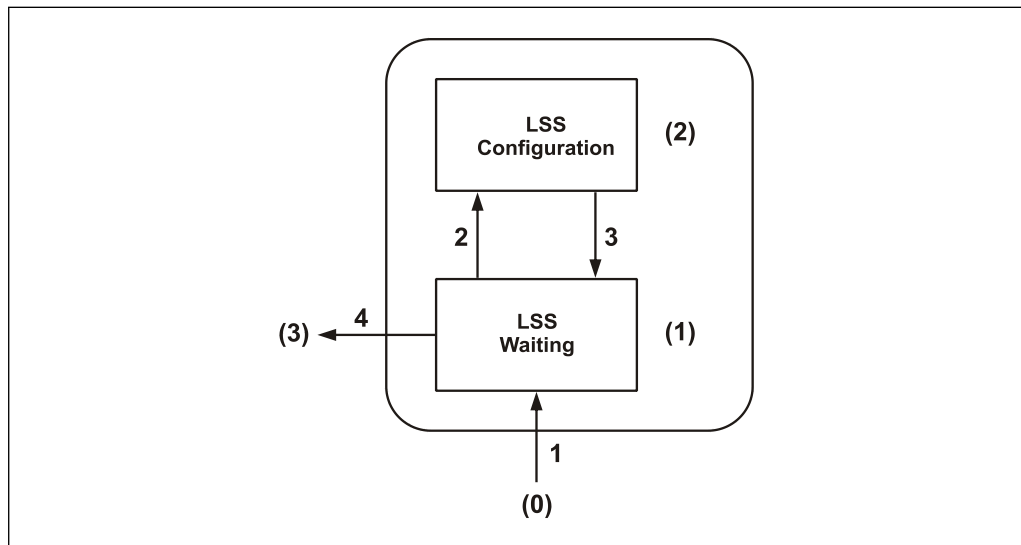


Figure 6: LSS FSA state machine

State behavior of the supported services

Services	LSS Waiting	LSS Configuration
Switch state global	Yes	Yes
Switch state selective	Yes	No
Activate bit timing parameters	No	Yes
Configure bit timing parameters	No	Yes
Configure Node-ID	No	Yes
Store configured parameters	No	Yes
Inquire LSS address	No	Yes
Inquire Node-ID	No	Yes
LSS identify remote slave	Yes	Yes
LSS identify slave	Yes	Yes
LSS identify non-configured remote slave	Yes	Yes
LSS identify non-configured slave	Yes	Yes

## LSS FSA state transitions

Transition	Events	Actions
1	Automatic transition after initial entry into either NMT PRE-OPERATIONAL state, or NMT STOPPED state, or NMT RESET COMMUNICATION state with Node-ID equals FFh.	none
2	LSS switch state global command with parameter 'configuration switch' or 'switch state selective' command.	none
3	LSS switch state global command with parameter 'waiting switch'.	none
4	Automatic transition if invalid Node-ID has been changed and the new Node-ID has been successfully stored in non-volatile memory AND state switch to LSS waiting was commanded.	none

Once the LSS FSA is entered further state transitions in the NMT FSA from NMT PRE-OPERATIONAL to NMT STOPPED state and vice versa does not lead to re-entering the LSS FSA.

#### 4.7.2 Transmission of LSS services

By means of LSS services, the LSS master requests services to be performed by the LSS slave. Communication between LSS master and LSS slave is made by means of implemented LSS protocols.

Similar as in the case of SDO transmitting, also here two COB-IDs for sending and receiving are used:

COB-ID	Meaning
0x7E4	LSS slave → LSS master
0x7E5	LSS master → LSS slave

Table 6: COB-IDs for Layer Setting Services (LSS)

##### 4.7.2.1 LSS message format

The data field with max. 8 byte length of a CAN message is used by a LSS service as follows:

CS	Data						
Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7

Table 7: LSS message

Byte 0 contains the **Command-Specifier** (CS), afterwards 7 byte data are following.



## 4.7.3 Switch mode protocols

### 4.7.3.1 Switch state global protocol

The given protocol has implemented the *Switch state global service* and controls the LSS state machine of the LSS slave. By means of the LSS master all LSS slaves in the network can be switched into the *LSS waiting* or *LSS configuration* state.

LSS-Master --> LSS-Slave

	0	1	2	3	4	5	6	7
COB-ID	CS	Mode		Reserved by CiA				
0x7E5	0x04	0 = waiting mode 1 = configuration mode						

### 4.7.3.2 Switch state selective protocol

The given protocol has implemented the *Switch state selective service* and controls the LSS state machine of the LSS slave. By means of the LSS master only this LSS slave in the network can be switched into the *LSS configuration* state, whose LSS address attributes equals the LSS address.

LSS-Master --> LSS-Slave

	0	1	2	3	4	5	6	7
COB-ID	CS	Vendor-ID ( $\triangleq$ Index 1018h:01)				Reserved by CiA		
0x7E5	0x40	LSB		MSB				

	0	1	2	3	4	5	6	7
COB-ID	CS	Product-Code ( $\triangleq$ Index 1018h:02)				Reserved by CiA		
0x7E5	0x41	LSB		MSB				

	0	1	2	3	4	5	6	7
COB-ID	CS	Revision-No. ( $\triangleq$ Index 1018h:03)				Reserved by CiA		
0x7E5	0x42	LSB		MSB				

	0	1	2	3	4	5	6	7
COB-ID	CS	Serial-No. ( $\triangleq$ Index 1018h:04)				Reserved by CiA		
0x7E5	0x43	LSB		MSB				

LSS-Slave --> LSS-Master

	0	1	2	3	4	5	6	7
COB-ID	CS	Reserved by CiA						
0x7E4	0x44							

## 4.7.4 Configuration protocols

### 4.7.4.1 Configure Node-ID protocol

The given protocol has implemented the *Configure Node-ID service*. By means of the LSS master the Node-ID of a single LSS slave in the network can be configured. Only one device is to be switched into *LSS configuration state*. For storage of the new Node-ID the *Store configuration protocol* must be transmitted to the LSS slave. To activate the new Node-ID the NMT service *Reset Communication* (0x82) must be called.

LSS-Master --> LSS-Slave

	0	1	2	3	4	5	6	7
COB-ID	CS	Node-ID	Reserved by CiA					
0x7E5	0x11	0x01...0x7F, 0xFF						

LSS-Slave --> LSS-Master

	0	1	2	3	4	5	6	7
COB-ID	CS	Error Code	Spec. Error	Reserved by CiA				
0x7E4	0x11							

Node-ID

1...127: valid addresses

Error Code

0: Protocol successfully completed  
 1: Node-ID out of range, 1...127  
 2...254: reserved  
 255: application specific error occurred

Specific Error

if Error Code = 255 --> application specific error occurred,  
 otherwise reserved by CiA



*The Node-ID specified here can also be changed via "Object 2110h: LSS Node-ID" using SDO. See also chapter 5.4 "Setting the Node-ID and Baud rate" on page 141 and chap. 9.2.12 "Object 2110h: LSS Node-ID" on page 173.*

#### 4.7.4.2 Configure bit timing parameters protocol

The given protocol has implemented the *Configure bit timing parameters service*. By means of the LSS master the Baud rate of a single LSS slave in the network can be configured. Only one device is to be switched into *LSS configuration state*. For storage of the new Baud rate the *Store configuration protocol* must be transmitted to the LSS slave.

LSS-Master --> LSS-Slave

	0	1	2	3	4	5	6	7
COB-ID	CS	Table Selector	Table Index	Reserved by CiA				
0x7E5	0x13	0	0x00...0x08					

LSS-Slave --> LSS-Master

	0	1	2	3	4	5	6	7
COB-ID	CS	Error Code	Spec. Error	Reserved by CiA				
0x7E4	0x13							

Table Selector

0: Standard CiA Baud rate table

Table Index

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

Error Code

0: Protocol successfully completed  
 1: selected Baud rate not supported  
 2...254: reserved  
 255: application specific error occurred

Specific Error

if Error Code = 255 --> application specific error occurred,  
 otherwise reserved by CiA



The Baud rate specified here can also be changed via "Object 2111h: LSS BitTiming" using SDO. See also chapter 5.4 "Setting the Node-ID and Baud rate" on page 141 and chap. 9.2.13 "Object 2111h: LSS BitTiming" on page 174.

#### 4.7.4.3 Activate bit timing parameters protocol

The given protocol has implemented the *Activate bit timing parameters service*. The protocol activates the Baud rate which was configured about the *Configure bit timing parameters protocol* and is performed with all LSS slaves in the network which are in the state *LSS configuration*.

LSS-Master --> LSS-Slave

	0	1	2	3	4	5	6	7
COB-ID	CS	Switch Delay [ms]		Reserved by CiA				
0x7E5	0x15	LSB	MSB					

##### Switch Delay

The parameter *Switch Delay* defines the length of two delay periods (D1, D2) with equal length. These are necessary to avoid operating the bus with differing Baud rate parameters.

After the time D1 and an individual processing duration, the switching internally in the LSS slave is performed. After the time D2 the LSS slave responses with CAN-messages and the new configured Baud rate.

It is necessary:

Switch Delay > longest occurring processing duration of a LSS slave

#### 4.7.4.4 Store configuration protocol

The given protocol has implemented the *Store configuration service*. By means of the LSS master the configured parameters of a single LSS slave in the network can be stored into the non-volatile memory. Only one device is to be switched into *LSS configuration* state.

LSS-Master --> LSS-Slave

	0	1	2	3	4	5	6	7
COB-ID	CS	Reserved by CiA						
0x7E5	0x17							

LSS-Slave --> LSS-Master

	0	1	2	3	4	5	6	7
COB-ID	CS	Error Code	Spec. Error	Reserved by CiA				
0x7E4	0x17							

##### Error Code

- 0: Protocol successfully completed
- 1: *Store configuration* not supported
- 2: Storage media access error
- 3...254: reserved
- 255: application specific error occurred

##### Specific Error

if Error Code = 255 --> application specific error occurred,  
otherwise reserved by CiA

## 4.7.5 Inquire LSS address protocols

### 4.7.5.1 Inquire identity Vendor-ID protocol

The given protocol has implemented the *Inquire LSS address service*. By means of the LSS master the Vendor-ID of a single LSS slave in the network can be read-out. Only one device is to be switched into *LSS configuration state*.

LSS-Master --> LSS-Slave

	0	1	2	3	4	5	6	7
<b>COB-ID</b>	<b>CS</b>	Reserved by CiA						
0x7E5	0x5A							

LSS-Slave --> LSS-Master

	0	1	2	3	4	5	6	7
<b>COB-ID</b>	<b>CS</b>	<b>Vendor-ID</b> ( $\triangleq$ Index 1018h:01)				Reserved by CiA		
0x7E4	0x5A	LSB		MSB				

### 4.7.5.2 Inquire identity Product-Code protocol

The given protocol has implemented the *Inquire LSS address service*. By means of the LSS master the Product-Code of a single LSS slave in the network can be read-out. Only one device is to be switched into *LSS configuration state*.

LSS-Master --> LSS-Slave

	0	1	2	3	4	5	6	7
<b>COB-ID</b>	<b>CS</b>	Reserved by CiA						
0x7E5	0x5B							

LSS-Slave --> LSS-Master

	0	1	2	3	4	5	6	7
<b>COB-ID</b>	<b>CS</b>	<b>Product-Code</b> ( $\triangleq$ Index 1018h:02)				Reserved by CiA		
0x7E4	0x5B	LSB		MSB				

## 4.7.5.3 Inquire identity Revision-Number protocol

The given protocol has implemented the *Inquire LSS address service*. By means of the LSS master the Revision-No. of a single LSS slave in the network can be read-out. Only one device is to be switched into *LSS configuration* state.

LSS-Master --> LSS-Slave

	0	1	2	3	4	5	6	7
COB-ID	CS	Reserved by CiA						
0x7E5	0x5C							

LSS-Slave --> LSS-Master

	0	1	2	3	4	5	6	7
COB-ID	CS	Revision-No. ( $\triangleq$ Index 1018h:03)				Reserved by CiA		
0x7E4	0x5C	LSB		MSB				

## 4.7.5.4 Inquire identity Serial-Number protocol

The given protocol has implemented the *Inquire LSS address service*. By means of the LSS master the Serial-No. of a single LSS slave in the network can be read-out. Only one device is to be switched into *LSS configuration* state.

LSS-Master --> LSS-Slave

	0	1	2	3	4	5	6	7
COB-ID	CS	Reserved by CiA						
0x7E5	0x5D							

LSS-Slave --> LSS-Master

	0	1	2	3	4	5	6	7
COB-ID	CS	Serial-No. ( $\triangleq$ Index 1018h:04)				Reserved by CiA		
0x7E5	0x5D	LSB		MSB				

#### 4.7.6 Inquire Node-ID protocol

The given protocol has implemented the *Inquire Node-ID service*. By means of the LSS master the Node-ID of a single LSS slave in the network can be read-out. Only one device is to be switched into *LSS configuration* state.

LSS-Master --> LSS-Slave

	0	1	2	3	4	5	6	7
COB-ID	CS	Reserved by CiA						
0x7E5	0x5E							

LSS-Slave --> LSS-Master

	0	1	2	3	4	5	6	7
COB-ID	CS	Node-ID	Reserved by CiA					
0x7E4	0x5E	0x00...0x7F, 0xFF						

#### Node-ID

Corresponds the Node-ID of the selected device. If the Node-ID currently was changed by means of the *Configure Node-ID service*, the original Node-ID is reported. Only after execution of the NMT service *Reset Communication* (0x82) the actual Node-ID is reported.

## 4.7.7 Identification protocols

### 4.7.7.1 LSS identify remote slave protocol

The given protocol has implemented the *LSS identify remote slave service*. By means of the LSS master LSS slaves in the network can be identified within a certain range. All LSS slaves with matching Vendor-ID, Product-Code, Revision-No. Range and Serial-No. Range, response with the *LSS identify slave protocol*.

LSS-Master --> LSS-Slave

	0	1	2	3	4	5	6	7
COB-ID	CS	Vendor-ID ( $\triangle$ Index 1018h:01)				Reserved by CiA		
0x7E5	0x46	LSB		MSB				

	0	1	2	3	4	5	6	7
COB-ID	CS	Product-Code ( $\triangle$ Index 1018h:02)				Reserved by CiA		
0x7E5	0x47	LSB		MSB				

	0	1	2	3	4	5	6	7
COB-ID	CS	Revision-No. LOW				Reserved by CiA		
0x7E5	0x48	LSB		MSB				

	0	1	2	3	4	5	6	7
COB-ID	CS	Revision-No. HIGH				Reserved by CiA		
0x7E5	0x49	LSB		MSB				

	0	1	2	3	4	5	6	7
COB-ID	CS	Serial-No. LOW				Reserved by CiA		
0x7E5	0x4A	LSB		MSB				

	0	1	2	3	4	5	6	7
COB-ID	CS	Serial-No. HIGH				Reserved by CiA		
0x7E5	0x4B	LSB		MSB				

### 4.7.7.2 LSS identify slave protocol

The given protocol has implemented the *LSS identify slave service*. All LSS slaves with matching LSS attributes given in the *LSS identify remote slave protocol*, response with this protocol.

LSS-Slave --> LSS-Master

	0	1	2	3	4	5	6	7
COB-ID	CS	Reserved by CiA						
0x7E4	0x4F							



#### 4.7.7.3 LSS identify non-configured remote slave protocol

The given protocol has implemented the *LSS identify non-configured remote slave service*. By means of the LSS master all non-configured LSS slaves (Node-ID = FFh) in the network are identified. The relevant LSS slaves response with the *LSS identify non-configured slave protocol*.

LSS-Master --> LSS-Slave

	0	1	2	3	4	5	6	7
COB-ID	CS	Reserved by CiA						
0x7E5	0x4C							

#### 4.7.7.4 LSS identify non-configured slave protocol

The given protocol has implemented the *LSS identify non-configured slave service*. After execution of the *LSS identify non-configured remote slave protocol* all non-configured LSS slaves with Node-ID FFh response with this protocol.

LSS-Slave --> LSS-Master

	0	1	2	3	4	5	6	7
COB-ID	CS	Reserved by CiA						
0x7E4	0x50							

### 4.8 Device profile

The CANopen device profiles describe the "what" of the communication. In the profiles the meaning of the transmitted data is unequivocal and manufacturer independently defined. So the basic functions of each device class

e.g. for encoder: **CiA DS-406**

can be responded uniformly. On the basis of these standardized profiles CANopen devices can be accessed in an identical way over the bus. Therefore devices which support the same device profile are exchangeable with each other.

You can obtain further information on CANopen from the **CAN in Automation** User- and Manufacturer Association:

---

#### **CAN in Automation**

Am Weichselgarten 26  
DE-91058 Erlangen

Tel. +49-9131-69086-0  
Fax +49-9131-69086-79

Website: [www.can-cia.org](http://www.can-cia.org)  
e-mail: [headquarters@can-cia.org](mailto:headquarters@can-cia.org)

---

## 5 Installation / Preparation for start-up

The CANopen system is wired in bus topology with terminating resistors (120 ohms) at the beginning and at the end of the bus line. If it is possible, drop lines should be avoided. The cable is to be implemented as shielded twisted pair cable and should have an impedance of 120 ohms and a resistance of 70 mΩ/m. The data transmission is carried out about the signals CAN-H and CAN-L with a common GND as data reference potential. Optionally also a 24 V supply voltage can be carried.

In a CANopen network max. **127** slaves can be connected. The measuring system supports the Node-ID range from 1...127. The transmission rate can be adjusted via Dip-switches, LSS protocol or SDO and supports the following baud rates:

- 10 kbit/s
- 20 kbit/s
- 50 kbit/s
- 100 kbit/s (*only LSS or SDO*)
- 125 kbit/s
- 250 kbit/s
- 500 kbit/s
- 800 kbit/s
- 1 Mbit/s

The length of a CANopen network is depending on the transmission rate and is represented in the following:

Cable cross section	10 kbit/s	20 kbit/s	50 kbit/s	100 kbit/s	125 kbit/s	250 kbit/s	500 kbit/s	800 kbit/s	1 Mbit/s
0.25 mm <sup>2</sup> – 0.34 mm <sup>2</sup>	5000 m	2500 m	1000 m	ca. 600 m	500 m	250 m	100 m	50 m	25 m

The



- ISO 11898,
- the recommendations of the CiA DR 303-1 (CANopen cabling and connector pin assignment)
- and other applicable standards and guidelines are to be observed to insure safe and stable operation!

*In particular, the applicable EMC directive and the shielding and grounding guidelines must be observed!*

### 5.1 Connection – notes

Mainly, the electrical characteristics are defined by the variable connection technique and are defined by the device-specific pin assignment.



*The connection can be made only in connection with the device specific pin assignment!*

*At the delivery of the measuring system one device specific pin assignment in printed form is enclosed and it can be downloaded afterwards from the page „[www.tr-electronic.com/service/downloads/pin-assignments.html](http://www.tr-electronic.com/service/downloads/pin-assignments.html)“. The number of the pin assignment is noted on the nameplate of the measuring system.*

---

### 5.2 Bus termination

If the measuring system is the last slave in the CAN segment, the bus is to be terminated with the termination switch = ON. In this state, the subsequent CAN-bus is decoupled. The position and orientation of the termination switch can be found in the device-specific pin assignment.

For measuring systems without a termination switch, the continuing CAN-bus must be provided with a 120  $\Omega$  termination resistor.

### 5.3 Switching on the supply voltage

After the connection and all settings have been carried out, the supply voltage can be switched on.

After power on and finishing the initialization, the measuring system goes into the `PRE-OPERATIONAL` state. In the default setting, this status is acknowledged by the Boot-Up message "**COB-ID 0x700+Node ID**". If the measuring system detects an internal error, an emergency message with the error code will be transmitted (see chapter "Emergency Message", page 196).

In the `PRE-OPERATIONAL` state first only a parameter setting about Service-Data-Objects is possible. But it is possible to configure PDOs with the help of SDOs. If the measuring system was transferred into the `OPERATIONAL` state, also a transmission of PDOs is possible.

## 5.4 Setting the Node-ID and Baud rate

The Node-ID and Baud rate can be set via DIP switches, LSS services or SDOs.

On delivery, the Node-ID and the Baud rate must be set via DIP switches. In order to be able to make the settings by means of LSS or SDOs, the DIP switches 1...7 must be set to "ON" before switching on the supply voltage in order to activate the LSS or SDO services.

If the measuring system has no DIP switches, the Node-ID and Baud rate set via LSS or SDO services are automatically active.



- A Node-ID may only be assigned only once in the CAN bus.
- If the LSS services are active, the baud rate set via DIP switches is ineffective.

### 5.4.1 Setting by means of DIP switches

If the measuring system has DIP switches, the node ID is set via DIP switches 1 ... 7 and the baud rate via DIP switches 8 ... 10 according to the pin assignment.



- The Node-ID corresponds to the switch position as a binary value + 1.
- The switch position is read only at the moment of switch-on, subsequent changes are not recognized.

### 5.4.2 Setting by means of LSS services

The sequence for setting the Node-ID and Baud rate using LSS services is described below. See also chapters: 4.7.4.1 "Configure Node-ID protocol" and 4.7.4.2 "Configure bit timing parameters protocol".

#### 5.4.2.1 Configuration of the Node-ID

Assumption:

- LSS address unknown
- only one LSS slave should be in the network
- the Node-ID 12 dec. shall be adjusted

Procedure:

- Perform service 04 *Switch state global protocol*, Mode = 1, to switch the LSS slave into *Configuration state*.
- Perform service 17 *Configure Node-ID protocol*, Node-ID = 12.  
--> Wait for acknowledgement and check successfully execution,  
--> Error Code = 0.
- Perform service 23 *Store configuration protocol*.  
--> Wait for acknowledgement and check successfully execution,  
--> Error Code = 0.
- Perform service 04 *Switch state global protocol*, Mode = 0, to switch the LSS slave into *Waiting state*.
- Perform NMT service *Reset Communication* (0x82), to switch the new Node-ID active.

### 5.4.2.2 Configuration of the Baud rate

Assumption:

- LSS address unknown
- only one LSS slave should be in the network
- the Baud rate 125 kbit/s shall be adjusted

Procedure:

- Perform NMT service *Stop Remote Node* (0x02), to switch the LSS slave into *Stopped state*. The LSS slave shouldn't sent any CAN-messages  
--> Heartbeat switched off.
- Perform service 04 *Switch state global protocol*,  
Mode = 1, to switch the LSS slave into *Configuration state*.
- Perform service 19 *Configure bit timing parameters protocol*,  
Table Selector = 0, Table Index = 4  
--> Wait for acknowledgement and check successfully execution,  
--> Error Code = 0.
- Perform service 21 *Activate bit timing parameters protocol*, to  
switch the new Baud rate active.
- Perform service 23 *Store configuration protocol*.  
--> Wait for acknowledgement and check successfully execution,  
--> Error Code = 0.
- Perform service 04 *Switch state global protocol*,  
Mode = 0, to switch the LSS slave into *Waiting state*.

### 5.4.3 Setting by means of SDOs

The Node-ID is set via "Object 2110h: LSS Node-ID" and the Baud rate via "Object 2111h: LSS BitTiming". See chapters 9.2.12 and 9.2.13.



*The Node-ID and Baud rate set via SDOs correspond to the values set via LSS services and overwrite each other.*

---

## 6 Commissioning

### 6.1 CAN – interface

The CAN-Bus-Interface is defined by the international norm ISO/DIS 11898 and specifies the two lowest layers of the ISO/DIS CAN Reference Model.

The CAN-BUS-Interface is galvanic isolated of the measuring system electronic and becomes the power over internal DC/DC-converter. There is no external power supply necessary for the CAN-BUS-Driver.

The CANopen Communication Profile (CiA standard DS 301) is a subset of CAN Application Layer (CAL) and describes, how the services are used by devices. The CANopen Profile allows the definition of device profiles for decentralized I/O.

The measuring system with CANopen-protocol support the Device Profile for Encoder (CIA Draft Standard Proposal 406 V3.2.0). **The measuring systems support the extended functions in Class C2.**

The communication functionality and objects, which are used in the encoder profile, are described in a EDS-File (Electronic Data Sheet).

When using a CANopen Configuration Tool (e.g.: CANSETTER), the user can read the objects of the measuring system (SDOs) and program the functionality.

The selection of transmission rate and node number is done by switches or LSS-service.

#### 6.1.1 EDS file

The EDS (electronic datasheet) contains all information on the measuring system-specific parameters and the measuring system's operating modes. The EDS file is integrated using the CANopen network configuration tool to correctly configure or operate the measuring system.

**Download:** [www.tr-electronic.de/f/TR-ECE-ID-MUL-0064](http://www.tr-electronic.de/f/TR-ECE-ID-MUL-0064)

#### 6.1.2 Bus status

The position, assignment and flashing frequency of the status display (LED's) can be found in the device-specific pin assignment.



*At the delivery of the measuring system one device specific pin assignment in printed form is enclosed and it can be downloaded afterwards from the page „[www.tr-electronic.com/service/downloads/pin-assignments.html](http://www.tr-electronic.com/service/downloads/pin-assignments.html)“. The number of the pin assignment is noted on the nameplate of the measuring system.*

---

Corresponding measures in case of an error see chapter “Optical displays”, page 197.

## 7 Communication profile

Generally there are two kinds of process data objects (PDO):

1. Transmit-PDOs (TPDO), to send data
2. Receive-PDOs (RPDO), to receive data

By the measuring system only Transmit-PDOs are supported, to transmit the actual value or the speed value.

The TPDOs are described by the TPDO communication parameter 1800h-1801h and the TPDO mapping parameter 1A00h-1A01h. The communication parameter describes the communication capabilities of the TPDO and the mapping parameter contains information about the contents of the TPDO.

### 7.1 Structure of the communication parameter, 1800h-1801h

Sub-index 0 contains the number of valid object entries.

Sub-index 1 contains the COB-ID of the TPDO:

31	30	29	28	11	10	0
Valid	RTR	Frame	0 0000h	11-Bit CAN-ID		
MSB				LSB		

Bit(s)	Description
Valid	0: PDO exists / is valid 1: PDO does not exist / is not valid
RTR	0: Remote Frame allowed on this PDO 1: no Remote Frame allowed on this PDO
Frame	0: 11-Bit CAN-ID valid, CAN base frame 1: 29-Bit CAN-ID valid, CAN extended frame (not supported)
11-Bit CAN-ID	11-Bit CAN-ID of the CAN base frame

Sub-index 2 defines the transmission character of the TPDO:

Value	Description
01h	Actual value is transferred synchronously over a remote frame or SYNC telegram
02h	Actual value is transferred synchronously over a remote frame or cyclically after each 2. SYNC telegram
03h	Actual value is transferred synchronously over a remote frame or cyclically after each 3. SYNC telegram
...	...
F0h	Actual value is transferred synchronously over a remote frame or cyclically after each 240. SYNC telegram
FDh	Actual value can be transferred only over a remote frame
FEh	Actual value is transferred asynchronously with the timer value from object 1800h and 1801h (sub-index 5)



Sub-index 3 contains the inhibit time for the TPDO. The time is the minimum interval for PDO transmission if the transmission type is set to FEh. The value is defined as multiple of 100  $\mu$ s. The value of 0 disables the inhibit time.

The value must not be changed while the PDO exists (bit 31 of sub-index 1 is set to 0)

Sub-index 4 is not supported.

Sub-index 5 contains the event-timer. The time is the maximum interval for PDO transmission if the transmission type is set to FEh. The value is defined as multiple of 1 ms. The value of 0 disables the event-timer.

The event-timer sub-index 5 of the communication parameter 1800h is hard-wired with Object 6200h: Cyclic timer. That means that a change in the event timer causes a change in the cyclic timer and vice versa.

The communication parameter 1801h exclusively uses its own timer, access over sub-index 5.

## 7.2 Structure of the mapping parameter, 1A00h-1A01h

Sub-index 0 contains the number of valid object entries. The value of 0 disables the mapping function.

The following entries contain the information of the mapped application objects. The object describes the content of the PDO by their index, sub-index and length in bit:

31	16	15	8	7	0
Index			Sub-index		Length in bit
MSB			LSB		

### 7.2.1 Procedure for re-mapping

- Destroy TPDO by setting bit “valid” to 1 of sub-index 1 of the according communication parameter 1800h-1801h.
- Disable mapping by setting sub-index 0 to 0 in the according mapping parameter 1A00h-1A01h.
- Modify mapping by changing the values of the corresponding mapping parameter 1A00h-1A01h (from sub-index 1).
- Enable mapping by setting sub-index 0 to the number of mapped objects in the corresponding mapping parameter 1A00h-1A01h.
- Create TPDO by setting bit “valid” to 0 of sub-index 1 of the according communication parameter 1800h-1801h. The desired COB-ID and the bit “valid” must be set together with one write command!
- Save mapping configuration by means of “Object 1010h: Store parameters”.

Provided mapping objects:

- Object 200Ah: Speed value INT32, see page 171
- Object 6004h: Position value, see page 185
- Object 6030h: Speed, see page 186
- Object 6400h: Area state register, see page 187
- Object 6503h: Alarms, see page 190

### 7.3 Transmission of the measuring system position value

Before the measuring system position can be transferred the measuring system has to be started with the “Node Start” command.

#### Node-Start Protocol

COB-Identifier = 0	
Byte 0	Byte 1
1	Node-ID

Node Start command with the Node-ID of the measuring system (slave) starts only this device.

Node Start command with **Node-ID = 0** starts all slaves connected to the bus.

After the Node Start command the measuring system transmit in the default setting the position value one time with the COB-ID of object 1800h. This service can be prevented via the "Object 2101h: Send PDO at Node-Start" on page 173.

**In the default setting the measuring system position value can be transmitted in different ways:**

### 7.3.1 1st Transmit Process-Data-Object

In the default setting this TPDO transmits the position value of the measuring system in an asynchronous way. The value of the timer is stored in sub-index 5 or in index 6200h. The default setting of the timer is 0, that means the timer is disabled.

Index	Sub-Index	Comment	Default value	Attr.
1800h	0	largest supported Sub-Index	5	ro
	1	COB-ID used by TPDO 1	180h + Node-ID	rw
	2	transmission type	254	rw
	3	inhibit time	0	rw
	4	-	-	-
	5	event timer <--> cyclic timer	0	rw
1A00h	0	largest supported Sub-Index	1 (max. 8)	rw
	1	1. mapped object	6004 0020h	rw
	2	2. mapped object	0000 0000h	rw
	3	3. mapped object	0000 0000h	rw
	4	4. mapped object	0000 0000h	rw
	5	5. mapped object	0000 0000h	rw
	6	6. mapped object	0000 0000h	rw
	7	7. mapped object	0000 0000h	rw
	8	8. mapped object	0000 0000h	rw

### 7.3.2 2nd Transmit Process-Data-Object

In the default setting this TPDO transmits one-time the position value of the measuring system in a cyclic way (on request). Request by remote frame (default COB-ID: 280h+Node-ID) or SYNC telegram (default COB-ID: 080h).

Index	Sub-Index	Comment	Default value	Attr.
1801h	0	largest supported Sub-Index	5	ro
	1	COB-ID used by TPDO 2	280 + Node-ID	rw
	2	transmission type	1	rw
	3	inhibit time	0	rw
	4	-	-	-
	5	event timer	0	rw
1A01h	0	largest supported Sub-Index	1 (max. 8)	rw
	1	1. mapped object	6004 0020h	rw
	2	2. mapped object	0000 0000h	rw
	3	3. mapped object	0000 0000h	rw
	4	4. mapped object	0000 0000h	rw
	5	5. mapped object	0000 0000h	rw
	6	6. mapped object	0000 0000h	rw
	7	7. mapped object	0000 0000h	rw
	8	8. mapped object	0000 0000h	rw

## 8 Communication specific standard objects (CiA DS-301)

Following table gives an overview on the supported indices in the Communication Profile Area:

M = Mandatory

O = Optional

Index (h)	Object	Name	Type	Attr.	M/O	Page
1000	VAR	Device type	Unsigned32	ro	M	149
1001	VAR	Error register	Unsigned8	ro	M	149
1002	VAR	Manufacturer status register	Unsigned32	ro	O	150
1003	ARRAY	Pre-defined error field	Unsigned32	rw	O	150
1005	VAR	COB-ID SYNC message	Unsigned32	rw	O	151
<sup>1)</sup> 1008	VAR	Device name	Vis-String	const	O	151
<sup>1)</sup> 1009	VAR	Hardware version	Vis-String	const	O	151
<sup>1)</sup> 100A	VAR	Software version	Vis-String	const	O	151
100C	VAR	Guard time	Unsigned16	rw	O	152
100D	VAR	Life time factor	Unsigned8	rw	O	152
1010	ARRAY	Store parameters	Unsigned32	rw	O	153
1011	ARRAY	Restore default parameters	Unsigned32	rw	O	154
1014	VAR	COB-ID EMCY	Unsigned32	rw	O	155
1016	ARRAY	Consumer heartbeat time	Unsigned32	rw	O	155
1017	VAR	Producer heartbeat time	Unsigned16	rw	O	156
1018	RECORD	Identity Object	Identity (23h)	ro	M	156
<sup>1)</sup> 1021	VAR	Store EDS	Domain	ro	O	157
1022	VAR	Store format	Unsigned8	ro	M	157
1029	ARRAY	Error behavior object	Unsigned8	rw	O	157
<sup>1)</sup> 1F50	ARRAY	Program data	Domain	rw	O	158
1F51	ARRAY	Program control	Unsigned8	rw	M	158
1F56	ARRAY	Program software identification	Unsigned32	ro	M	159
1F57	ARRAY	Flash status identification	Unsigned32	ro	M	159
1F80	VAR	NMT startup	Unsigned32	rw	O	160

**Table 8: Communication specific standard objects**

<sup>1)</sup> segmented reading

## 8.1 Object 1000h: Device type

Contains information about the device type. The object at index 1000h describes the type of device and its functionality. It is composed of a 16 bit field which describes the device profile that is used (Device Profile Number 406 = 196h) and a second 16 bit field which gives information on the type of encoder.

Unsigned32

Device Type			
Device Profile Number		Encoder Type	
Byte 0	Byte 1	Byte 2	Byte 3
196h		$2^7$ to $2^0$	$2^{15}$ to $2^8$

### Encoder Type

Code	Definition
01	Single-Turn absolute rotary encoder
02	Multi-Turn absolute rotary encoder

## 8.2 Object 1001h: Error register

The error register displays bit coded the error state of the measuring system. Also several errors at the same time can be displayed by a set bit. The more exact error cause can be taken from the bits 0-15 of the object 0x1003. An error is signaled at the moment of the occurrence by an EMCY message.

Unsigned8

Bit	Meaning
0	generic error
1	0
2	0
3	0
4	communication error (overflow, error state)
5	0
6	0
7	0

### 8.3 Object 1002h: Manufacturer status register

This object is not used by the measuring system, by read access the value is always "0".

### 8.4 Object 1003h: Pre-defined error field

This object saves the measuring system error occurred last and displays the error via the Emergency object. Each new error overwrites an error which was stored before in sub-index 1. Sub-index 0 contains the number of the occurred errors. Meanings of the error codes, see Table 11 on page 202.

Index	Sub-Index	Comment	Type	Attribute
1003h	0	number of errors	Unsigned8	ro
	1	standard error field	Unsigned32	rw

Sub-index 0: The entry at sub-index 0 contains the number of errors that have occurred and recorded in sub-index 1.

Sub-index 1: The error are composed of a 16 bit error code and a 16 bit additional error information.

Unsigned32

Standard Error Field			
Byte 0	Byte 1	Byte 2	Byte 3
Error code		Additional Information, not supported	

## 8.5 Object 1005h: COB-ID SYNC message

This object defines the COB-ID of the Synchronization Object (SYNC). Further, it defines whether the device consumes the SYNC or whether the device generates the SYNC. However, the measuring system supports only the processing of SYNC-messages and uses the 11-bit identifier.

Unsigned32

MSB

LSB

31	30	29	28-11	10-0
X	0	0	0	00 1000 0000

Bit 31 not relevant

Bit 30 = 0, device does not generate SYNC message

Bit 29 = 0, 11-bit ID ( CAN 2.0A )

Bit 28 – 11 = 0

Bit 10 – 0 = 11-bit SYNC-COB-IDENTIFIER, default Value = 080H

If a SYNC-telegram with the identifier, defined in this object (080H), and data length = 0 has been received by the device, in the default setting the position value of the measuring system is transmitted by the 2nd Transmit PDO (object 1801h), non-recurrent triggering.

Object	Function Code	COB-ID
SYNC	0001	80h

## 8.6 Object 1008h: Device name

Contains the manufacturer device name (visible string), transmission via “Segment Protocol”.

## 8.7 Object 1009h: Hardware version

Contains the manufacturer hardware version (visible string), transmission via “Segment Protocol”.

## 8.8 Object 100Ah: Software version

Contains the manufacturer software version (visible string), transmission via “Segment Protocol”.

## 8.9 Object 100Ch: Guard time

The objects at index 100Ch and 100Dh include the guard time in milli-seconds and the life time factor. The life time factor multiplied with the guard time gives the live time for the Node Guarding Protocol. Default value = 0.

Unsigned16

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

## 8.10 Object 100Dh: Life time factor

The life time factor multiplied with the guard time gives the life time for the node guarding protocol. Default value = 0.

Unsigned8

Life Time Factor
Byte 0
$2^7$ to $2^0$



## 8.11 Object 1010h: Store parameters

This object supports the saving of parameters in nonvolatile memory (EEPROM).

Index	Sub-Index	Comment	Type	Attr.
1010h	0	largest supported Sub-Index	Unsigned8	ro
	1	save all parameters	Unsigned32	rw
	2	save communication parameters (objects: 1000h...1FFFh)	Unsigned32	rw
	3	save application parameters (objects: 6000h...9FFFh)	Unsigned32	rw
	4	save manufacturer defined parameters (objects: 2000h...5FFFh)	Unsigned32	rw

**Sub-Index 0:** The entry at sub-index 0 contains the largest Sub-Index that is supported. Value = 4.

**Sub-Index 1...4:** Contains the save command.

By read access the device provides information about its saving capability.

Bit 0 = 1, the device saves parameters only on command. That means, if parameters have been changed by the user and no "Store Parameter Command" had been executed, at the next power on, the parameters will have their old values.

Unsigned32

MSB		LSB	
Bits	31-2	1	0
Value	= 0	0	1

In order to avoid storage of parameters by mistake, storage is only executed when a specific signature is written to the appropriate sub-index. The signature is "save".

Unsigned32

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

On reception of the correct signature, the device saves the parameters. If saving failed, the device responds with abort domain transfer: Error code 0606 0000h.

If a wrong signature is written, the device refuses to save and responds with abort domain transfer: Error code 0800 0020h.

## 8.12 Object 1011h: Restore default parameter values

This object supports the restoring of the default values from the writable parameters.

Index	Sub-Index	Comment	Type	Attr.
1011h	0	largest supported Sub-Index	Unsigned8	ro
	1	restore all parameters	Unsigned32	rw
	2	restore communication parameters (objects: 1000h...1FFFh)	Unsigned32	rw
	3	restore application parameters (objects: 6000h...9FFFh)	Unsigned32	rw
	4	restore manufacturer defined parameters (objects: 2000h...5FFFh)	Unsigned32	rw

**Sub-Index 0:** The entry at sub-index 0 contains the largest Sub-Index that is supported. Value = 4.

**Sub-Index 1...4:** Contains the restore command.

By read access on subindex 1 the device provides information about its restoring capability.

Bit 0 = 1 means that the device supports the restoring of default values.

Unsigned32

MSB		LSB
Bits	31-1	0
Value	= 0	1

In order to avoid restoring of parameter values by mistake, restoring is only executed when a specific signature is written to the appropriate sub-index. The signature is "load".

Unsigned32

MSB			LSB
d	a	o	l
64h	61h	6Fh	6Ch

On reception of the correct signature, the device restores the appropriate default parameters. If restoring failed, the device responds with abort domain transfer: Error code 0606 0000h.

If a wrong signature is written, the device refuses to restore the defaults and responds with abort domain transfer: Error code 0800 0020h.

### 8.13 Object 1014h: COB-ID EMERGENCY (EMCY)

This object indicates the configured COB-ID for the EMCY write service.  
Default value = 80h + Node-ID.

EMCY Identifier, rw:

31	30	29	28	11	10	0
Valid	0	Frame	0 0000h	11-Bit CAN-ID		
MSB						LSB

Bit(s)	Description
Valid	0: EMCY exists / is valid 1: EMCY does not exist / is not valid
30	reserved, always 0
Frame	0: 11-Bit CAN-ID valid, CAN base frame 1: 29-Bit CAN-ID valid, CAN extended frame (not supported)
11-Bit CAN-ID	11-Bit CAN-ID of the CAN base frame

The bits 0 to 29 must not be changed, while the object exists and is valid (bit 31 = 0).  
If a new value shall be written, bit 31 must be set to 1 together with the new value. In this connection the Node-ID must be considered.

### 8.14 Object 1016h: Consumer heartbeat time

The consumer heartbeat time object indicates the expected heartbeat cycle time. Monitoring of the heartbeat producer starts after the reception of the first heartbeat. The consumer heartbeat time should be higher than the corresponding producer heartbeat time. If the heartbeat is not received within the heartbeat consumer time, the emergency 8130h is transmitted and both nodes, Producer/Consumer, will be set into PRE-OPERATIONAL state. Hereupon, the timer values of the Producer/Consumer are set to 0.

Index	Sub-Index	Comment	Type	Attribute
1016h	0	largest supported Sub-Index = 1	Unsigned8	ro
	1	Consumer heartbeat time	Unsigned32	rw

Consumer heartbeat time:

31	24	23	16	15	0
reserved, 00h	Node-ID, default = 1		Heartbeat time [ms], Default = 0		
MSB					LSB

The heartbeat time is given in multiples of 1 ms. The entry for the Node-ID corresponds to the Node-ID of the node to be guarded.

## 8.15 Object 1017h: Producer heartbeat time

The producer heartbeat time indicates the configured cycle time of the heartbeat in [ms]. The value 0 disables the producer heartbeat.

If the heartbeat producer time (value > 0) is configured the heartbeat protocol, cyclic sending of heartbeat messages, begins immediately.

If the heartbeat producer time was configured the heartbeat protocol starts on the transition from the NMT state `INITIALIZATION` to the NMT state `PRE-OPERATIONAL`. In this case the boot-up message is regarded as first heartbeat message.

Unsigned16

Producer Heartbeat Time	
Byte 0	Byte 1
$2^7$ to $2^0$	$2^{15}$ to $2^8$



*It is not allowed to use both error control mechanisms "Guarding protocol" and "Heartbeat protocol" on one NMT slave at the same time. Thus, if the heartbeat producer time is unequal 0 the heartbeat protocol is used.*

## 8.16 Object 1018h: Identity Object

This object provides general identification information of the CANopen device.

Index	Sub-Index	Comment	Type	Attribute
1018h	0	highest sub-index supported	Unsigned8	ro
	1	Vendor-ID	Unsigned32	ro
	2	Product-Code	Unsigned32	ro
	3	Revision-No.	Unsigned32	ro
	4	Serial-No.	Unsigned32	ro

**Sub-index0:** The entry at sub-index 0 contains the largest Sub-Index that is supported:  
Value = 4.

**Sub-index1:** Contains the Vendor-ID of the manufacturer:  
0x0000025C

**Sub-index2:** Provides information about the Product-Code:  
device specific

**Sub-index3:** Provides information about the Revision-No.:  
current Revision-No.

**Sub-index4:** Provides information about the Serial-No.:  
current Serial-No.

## 8.17 Object 1021h: Store EDS

Via this object the EDS file can be read out segmented as an ASCII-code value.

## 8.18 Object 1022h: Store format

This object shows the store format of the EDS file in object 1021h.  
Default 00h ( /ISO10646/, not compressed)

## 8.19 Object 1029h: Error behavior object

This object controls the behavior of the measuring system in case of a communication or device error.

Index	Sub-Index	Comment	Def.	Type	Attr.
1029h	0	largest supported Sub-Index	2	Unsigned8	ro
	1	communication error (see Table 11 on page 202)	00h	Unsigned8	rw
	2	profile- or manufacturer-specific error (see Table 12 on page 202)	01h	Unsigned8	rw

Code	Definition
00h	Change to NMT state PRE-OPERATIONAL (only if currently in NMT state OPERATIONAL)
01h	No change of the NMT state
02h	Change to NMT state STOPPED

## 8.20 Firmware update

A firmware update is only possible with a suitable CAN remote software which supports the CiA-protocol 302-3 (as of version: 4.1.0) and the segmented SDO download.

### 8.20.1 Object 1F50h: Program data

With this object a firmware update of the measuring system can be realized by writing the new firmware segmented to the sub-index 1. Before the update can be started, the program status "boot loader" must be set via object 1F51h (Stop firmware).

Index	Sub-Index	Comment	Type	Attribute
1F50h	0	largest supported Sub-Index	Unsigned8	ro
	1	Program number 1	Domain	rw

If the firmware update fails for any reason, the device responds with abort domain transfer: Error code 0606 0000h.

### 8.20.2 Object 1F51h: Program control

This object shall be used for the control of the update process. Device must be in PRE-OPERATIONAL mode.

Index	Sub-Index	Comment	Type	Attribute
1F51h	0	largest supported Sub-Index	Unsigned8	ro
	1	Program number 1	Unsigned8	rw

Code	Definition	
	write access	read access
00h	Stop firmware	Firmware stopped
01h	Start firmware <sup>1)</sup>	Firmware started
02h	Reset firmware <sup>1)</sup>	Firmware stopped
03h	Clear firmware <sup>1)</sup>	no firmware available

<sup>1)</sup> Only possible after the firmware is stopped.

If the requested action is not supported or not possible for any reason the transfer shall be responded with the SDO abort message. The SDO abort code 0609 0030h shall indicate a not supported action. The SDO abort code 0800 0022h shall indicate that a requested action can currently not be performed.

### 8.20.3 Object 1F56h: Program software identification

Implies in sub-index 1 a measuring system generated checksum of the firmware for identification.

Index	Sub-Index	Comment	Type	Attribute
1F56h	0	largest supported Sub-Index	Unsigned8	ro
	1	Program number 1	Unsigned32	ro

### 8.20.4 Object 1F57h: Flash status identification

By reading access to this object, it can be read out the status of the firmware memory.

Index	Sub-Index	Comment	Type	Attribute
1F57h	0	largest supported Sub-Index	Unsigned8	ro
	1	Program number 1	Unsigned32	ro

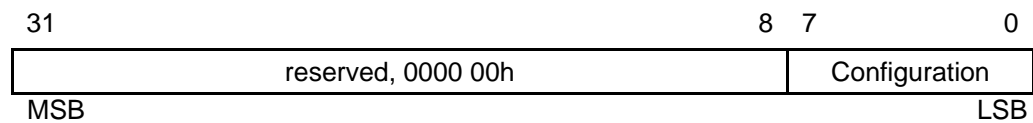
Bit	Code	Definition
0	0	Status ok, other bits valid, program software identification valid
	1	In progress, other bits not valid, program software identification not valid
1	0	No error occurred, program valid
	1	Program not valid
2..7	-	Not supported
8...15	-	Reserved (always 0)
16...31	-	Not supported

## 8.21 Object 1F80h: NMT Autostart

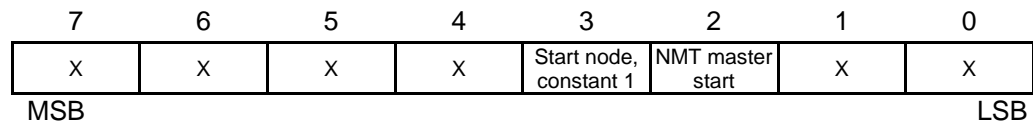
This object configures the startup behavior of the CANopen device and defines whether the device shall be switched automatically after the initialization into the `OPERATIONAL` state:

- Bit 2, NMT master start = 0:  
Device is switched into `OPERATIONAL` automatically
- Bit 2, NMT master start = 1; default setting:  
Device is not switched into `OPERATIONAL` automatically

Bit structure of the value:



Bit structure of the configuration byte



An attempt to change a bit of a functionality that is not supported by the CANopen device shall be responded with the abort message 0609 0030h.



## 9 Parameterization

M = Mandatory  
O = Optional  
C = Conditional

Index (h)	Object	Name	Data length	Attr.	M/O	Page
<b>TR Parameter</b>						
<sup>1)</sup> 2000	VAR	Mode selection TR / CiA DS-406	Unsigned16	rw	O	163
<sup>1)</sup> 2001	VAR	TR - Operating parameters	Unsigned16	rw	O	164
<sup>1)</sup> 2002	VAR	TR - Total measuring range in steps	Unsigned32	rw	O	165
<sup>1)</sup> 2003	VAR	TR - Number of revolution, numerator	Unsigned32	rw	O	166
<sup>1)</sup> 2004	VAR	TR - Number of revolution, denominator	Unsigned32	rw	O	166
<sup>1)</sup> 2005	VAR	Speed unit	Unsigned16	rw	O	169
2006	VAR	Additional parameterization	Unsigned32	rw	O	169
<sup>1)</sup> 2007	VAR	Speed Factor	Unsigned16	rw	O	170
<sup>1)</sup> 2008	VAR	Speed Integration Time	Unsigned16	rw	O	170
200A	ARRAY	Speed value INT32	Integer32	ro	O	171
2010	VAR	Change of state - Factor	Unsigned8	rw	O	171
2011	VAR	Change of state - Hysterese	Unsigned8	rw	O	172
<sup>1)</sup> 2100	VAR	COB-ID Bootup-Message	Unsigned16	rw	O	172
<sup>1)</sup> 2101	VAR	Send PDO at Node Start	Unsigned8	rw	O	173
2110	VAR	LSS NodeID	Unsigned8	rw	O	173
2111	VAR	LSS BitTiming	Unsigned8	rw	O	174
<sup>3)</sup> 2200	VAR	Store DMF	Domain	ro	O	174
2500	VAR	SSI-Output Available	Unsigned8	ro	O	174
<sup>1)</sup> 2501	VAR	SSI-Output - Data	Unsigned8	rw	O	175
<sup>1)</sup> 2502	VAR	SSI-Output - Code	Unsigned8	rw	O	175
<sup>1)</sup> 2503	VAR	SSI-Output - Number of Data-Bits	Unsigned8	rw	O	175
<sup>1)</sup> 2504	VAR	SSI-Output - Mono-Timer	Unsigned16	rw	O	176
2510	VAR	Preset-Inputs Available	Unsigned8	ro	O	176
<sup>1)</sup> 2511	VAR	Preset-Input Value 1	Unsigned32	rw	O	176
<sup>1)</sup> 2512	VAR	Preset-Input Value 2	Unsigned32	rw	O	177
2520	VAR	Incremental-Output Available	Unsigned8	ro	O	177
<sup>1)</sup> 2521	VAR	Incremental-Output Pulses	Unsigned32	rw	O	177
<sup>1)</sup> 2522	VAR	Incremental-Output Phase	Unsigned8	rw	O	178
<sup>1)</sup> 2523	VAR	Incremental-Output K0-Condition	Unsigned8	rw	O	178
<sup>1)</sup> 2524	VAR	Incremental-Output Level	Unsigned8	rw	O	179
<sup>1)</sup> 2525	VAR	Incremental-Output K0 Len	Unsigned8	rw	O	179
<sup>1)</sup> 2526	VAR	Incremental-Output Set K0	Unsigned32	rw	O	179
2530	VAR	Display Available	Unsigned8	ro	O	180
<sup>1)</sup> 2531	VAR	Display Button 1 function	Unsigned8	rw	O	180
<sup>1)</sup> 2532	VAR	Display Button 2 function	Unsigned8	rw	O	180
<sup>1)</sup> 2533	VAR	Display Button 3 function	Unsigned8	rw	O	180
<sup>1)</sup> 2534	VAR	Display Button 4 function	Unsigned8	rw	O	180
<sup>1)</sup> 2535	VAR	Display Control	Unsigned8	rw	O	180
<sup>1)</sup> 2536	VAR	Display Data Numeric	Unsigned32	rw	O	181
<sup>1)</sup> 2537	VAR	Display Data ASCII	VisibleString	rw	O	181
2538	VAR	Display Status	Unsigned8	ro	O	181
2700	VAR	TR-Production	Unsigned32	rw	O	181
2710	VAR	TR-Production	Unsigned8	rw	O	181
2720	VAR	TR-Production	Unsigned8	wo	O	181
2721	VAR	TR-Production	Unsigned16	ro	O	181

...

...

Index (h)	Object	Name	Data length	Attr.	M/O	Page
<b>TR Parameter</b>						
2722	VAR	TR-Production	Unsigned32	ro	O	181
2750	VAR	TR-Production	Unsigned8	rw	O	181
2751	VAR	TR-Production	VisibleString	rw	O	181
2752	VAR	TR-Production	VisibleString	rw	O	181
2753	VAR	TR-Production	VisibleString	rw	O	181
2754	VAR	TR-Production	VisibleString	rw	O	181
2755	VAR	TR-Production	VisibleString	rw	O	181
2756	VAR	TR-Production	VisibleString	rw	O	181
2757	VAR	TR-Production	VisibleString	rw	O	181
2758	VAR	TR-Production	VisibleString	rw	O	181
2759	VAR	TR-Production	VisibleString	rw	O	181
2760	VAR	TR-Production	VisibleString	rw	O	181
2761	VAR	TR-Production	Unsigned32	rw	O	181
<b>CiA DS-406 Parameter</b>						
<sup>1)</sup> 6000	VAR	Operating parameters	Unsigned16	rw	M	182
<sup>1)</sup> 6001	VAR	Measuring units per revolution	Unsigned32	rw	M	183
<sup>1)</sup> 6002	VAR	Total measuring range in measuring units	Unsigned32	rw	M	183
<sup>2)</sup> 6003	VAR	Preset value	Unsigned32	rw	M	185
6004	VAR	Position value	Unsigned32	ro	M	185
6030	ARRAY	Speed value	Integer16	ro	C	186
<sup>2)</sup> 6200	VAR	Cyclic timer	Unsigned16	rw	M	186
6400	ARRAY	Area state register	Unsigned8	ro	O	187
<sup>1)</sup> 6401	ARRAY	Work area low limit	Integer32	rw	O	187
<sup>1)</sup> 6402	ARRAY	Work area high limit	Integer32	rw	O	187
<b>Diagnostics</b>						
6500	VAR	Operating status	Unsigned16	ro	M	188
6501	VAR	Single-Turn resolution	Unsigned32	ro	M	188
6502	VAR	Number of distinguishable revolutions	Unsigned16	ro	M	189
6503	VAR	Alarms	Unsigned16	ro	C	190
6504	VAR	Supported alarms	Unsigned16	ro	M	191
6505	VAR	Warnings	Unsigned16	ro	C	192
6506	VAR	Supported warnings	Unsigned16	ro	M	193
6507	VAR	Profile and software version	Unsigned32	ro	M	194
6508	VAR	Operating time	Unsigned32	ro	M	194
6509	VAR	Offset value	Integer32	ro	M	195
650A	ARRAY	Manufacturer offset value	Integer32	ro	M	195
650B	VAR	Serial number	Unsigned32	ro	M	195

**Table 9: Encoder profile area**

- <sup>1)</sup> Is immediately active after a write command and is durably stored in the EEPROM after execute the object **"1010, Store parameters"**.
- <sup>2)</sup> Is immediately active and durably stored in the EEPROM after a write command.
- <sup>3)</sup> segmented reading

## 9.1 Object 2000h: Mode selection TR / CiA DS-406

With the mode selection can be selected which scaling parameter should be used. Normally the parameters according to the encoder profile CiA DS-406 are used. For special applications it can be switched over to TR parameter to use expanded gear functions.

Index	0x2000
<b>Description</b>	Mode selection TR / CiA DS-406
<b>Data type</b>	UNSIGNED16
<b>Object code</b>	VARIABLE
<b>Access</b>	rw
<b>PDO mapping</b>	no
<b>Lower limit</b>	0x0000 = CiA DS-406 mode
<b>Upper limit</b>	0x0001 = TR mode
<b>Default</b>	0x0000



*Only the parameters in the active mode can be changed. Not listed objects apply for both modes.*

CiA DS-406 mode	TR mode
0x6000, Counting direction	0x2001, Counting direction
0x6001, Single measuring range	0x2002, Total measuring range in steps
0x6002, Total measuring range in steps	0x2003, Number of revolution - numerator
	0x2004, Number of revolution - denominator

## 9.2 TR - Mode

### 9.2.1 Object 2001h: TR-Operating parameters, code sequence

The object with index 2001h supports only the function for the code sequence. The code sequence defines whether increasing or decreasing position values are output when the measuring system shaft rotates clockwise or counter clockwise as seen on the flange side.

Index	0x2001
Description	TR - Operating parameters
Data type	UNSIGNED16
Object code	VARIABLE
Access	rw
PDO mapping	no
Lower limit	0x0000 = increasing
Upper limit	0x0001 = decreasing
Default	0x0000

### 9.2.2 Scaling parameters

***Danger of personal injury and damage to property exists if the measuring system is restarted after positioning in the de-energized state by shifting of the zero point!***

If the number of revolutions is not an exponent of 2 or is >4096, it can occur, if more than 512 revolutions are made in the de-energized state, that the zero point of the multi-turn measuring system is lost!

**⚠ WARNING**

**NOTICE**

- Ensure that the quotient of **Revolutions Numerator / Revolutions Denominator** for a multi-turn measuring system is an exponent of 2 of the group  $2^0, 2^1, 2^2 \dots 2^{12}$  (1, 2, 4...4096).
- or
- Ensure that every positioning in the de-energized state for a multi-turn measuring system is within 512 revolutions.

### 9.2.2.1 Object 2002h: TR-Total measuring range in steps

This object defines the **total number of steps** of the measuring system before the measuring system restarts at zero.

Index	0x2002
Description	TR - Total measuring range in steps
Data type	UNSIGNED32
Object code	VARIABLE
Access	rw
PDO mapping	no
Lower limit	16 steps
Upper limit	4 294 967 295 = (0xFFFF FFFF) *
Default	16777216

\* It can be set "0" to display the value 0x100000000 ( $2^{32}$ ).

Total measuring range			
Byte 0	Byte 1	Byte 2	Byte 3
$2^7$ to $2^0$	$2^{15}$ to $2^8$	$2^{23}$ to $2^{16}$	$2^{31}$ to $2^{24}$

The actual upper limit for the measurement length to be entered in steps is dependent on the measuring system version and can be calculated with the formula below. As the value "0" is already counted as a step, the end value = measurement length in steps - 1.

$$\text{Total measuring range} = \text{Steps per revolution} * \text{Number of revolutions}$$

To calculate, the parameters **steps/rev.** and **the number of revolutions** can be read on the measuring system nameplate.

### 9.2.2.2 Object 2003h - 2004h: TR-Number of revolutions, numerator / denominator

Together, these two objects define the **number of revolutions** before the measuring system restarts at zero.

As decimal numbers are not always finite (as is e.g. 3.4), but they may have an infinite number of digits after the decimal point (e.g. 3.43535355358774...) the number of revolutions is entered as a fraction.

#### Number of revolutions numerator:

Index	0x2003
Description	TR - Number of revolutions, numerator
Data type	UNSIGNED32
Object code	VARIABLE
Access	rw
PDO mapping	no
Lower limit	1
Upper limit	256000
Default	4096

#### Number of revolutions denominator:

Index	0x2004
Description	TR - Number of revolutions, denominator
Data type	UNSIGNED32
Object code	VARIABLE
Access	rw
PDO mapping	no
Lower limit	1
Upper limit	16384
Default	1

#### Number of revolutions:

$$\text{Number of revolutions} = \frac{\text{Number of revolutions numerator}}{\text{Number of revolutions denominator}}$$

*If it is not possible to enter parameter data in the permitted ranges of numerator and denominator, the attempt must be made to reduce these accordingly. If this is not possible, it may only be possible to represent the decimal number affected approximately. The resulting minor inaccuracy accumulates for real round axis applications (infinite applications with motion in one direction).*

*A solution is e.g. to perform adjustment after each revolution or to adapt the mechanics or gearbox accordingly.*

*The parameter **"Steps per revolution"** may also be decimal number, however the **"Total measuring range"** may not. The result of the above formula must be rounded up or down. The resulting error is distributed over the total number of revolutions programmed and is therefore negligible.*

***Preferably for linear axes (forward and backward motions):***

*The parameter **"Revolutions denominator"** can be programmed as a fixed value of "1". The parameter **"Revolutions numerator"** is programmed slightly higher than the required number of revolutions. This ensures that the measuring system does not generate a jump in the actual value (zero transition) if the distance travelled is exceeded. To simplify matters the complete revolution range of the measuring system can also be programmed.*

**The following example serves to illustrate the approach:**

**Given:**

- Measuring system with 4096 steps/rev. and max. 4096 revolutions
- Resolution 1/100 mm
- Ensure the measuring system is programmed in its full resolution and total measuring length (4096x4096):  
Total number of steps = 16777216,  
Revolutions numerator = 4096  
Revolutions denominator = 1
- Set the mechanics to be measured to the left stop position
- Set measuring system to "0" using the adjustment
- Set the mechanics to be measured to the end position
- Measure the mechanical distance covered in mm
- Read off the actual value of the measuring system from the controller connected

**Assumed:**

- Distance covered = 2000 mm
- Measuring system actual position after 2000 mm = 607682 steps

**Derived:**

Number of revolutions covered = 607682 steps / 4096 steps/rev.  
= **148.3598633 revolutions**

Number of mm / revolution = 2000 mm / 148.3598633 revs. = **13.48073499mm / rev.**

For 1/100mm resolution this equates to a **Number of steps per revolution of 1348.073499**

**Required programming:**

Number of Revolutions numerator = **4096**  
Number of Revolutions denominator = **1**

$$\begin{aligned}\text{Total number of steps} &= \text{Number of steps per revolution} * \frac{\text{Number of revolutions numerator}}{\text{Number of revolutions denominator}} \\ &= 1348.073499 \text{ steps / rev.} * \frac{4096 \text{ revolutions numerator}}{1 \text{ revolution denominator}} \\ &= \textbf{5521709 steps} \text{ (rounded off)}\end{aligned}$$



### 9.2.3 Object 2005h: Speed unit

This object indicates the resolution in bit whereby the speed in „Object 6030h“ is calculated and output, see chapter 9.2.7 “Object 200Ah: Speed value INT32” or 9.3.5 „Object 6030h: Speed” on page 186.

Index	0x2005
<b>Description</b>	Speed unit
<b>Data type</b>	UNSIGNED16
<b>Object code</b>	VARIABLE
<b>Access</b>	rw
<b>PDO mapping</b>	no
<b>Default</b>	100

Selectable resolutions:

Value	Speed resolution
8	Steps/ms at 8 bit resolution
9	Steps/ms at 9 bit resolution
...	...
18	Steps/ms at 18 bit resolution
100	Steps/ms at scaled resolution *
101	Steps/10 ms at scaled resolution *
102	Steps/100 ms at scaled resolution *
103	Steps/s at scaled resolution *
200	rpm
201	rps

\* scaled resolution:

- CiA-DS 406-Mode = object 0x6001
- TR-Mode = result of object (0x2002 \* 0x2004) / 0x2003

See Object 2000h: Mode selection TR / CiA DS-406.

### 9.2.4 Object 2006h: Additional Parameter/Commands (device specific)

About this object device specific parameters and commands can be exchanged.

## 9.2.5 Object 2007h: Speed Factor

This object defines the factor value for "Object 2005h: Speed unit", see page 169.

Index	0x2007
Description	Speed Factor
Data type	UNSIGNED16
Object code	VARIABLE
Access	rw
PDO mapping	no
Lower limit	1
Upper limit	1000
Default	1

## 9.2.6 Object 2008h: Speed Integration Time

Defines the integration time in [ms] for "Object 2005h: Speed unit", see chapter 9.2.3 on page 169.

The parameter serves to calculate the velocity, which is output via the cyclic process input data. The velocity is specified in [ (Steps/Integration Time) \* Factor ]. High integration times enable high-resolution measurements at low speeds. Low integration times show velocity changes more quickly and are suitable for high speeds and high dynamics.

Index	0x2008
Description	Speed Integration Time
Data type	UNSIGNED16
Object code	VARIABLE
Access	rw
PDO mapping	no
Lower limit	0 ms
Upper limit	1000 ms
Default	16 ms

### Example

Given:

- Programmed resolution = 8192 steps per revolution
- Velocity = 4800 revolutions per minute
- Integration time  $t_i$  = 50 ms = 0.05 s
- Factor = 1

Find:

- Output value in (Steps/Integration Time) \* Factor

$$\text{Number of steps / s} = \frac{8192 \text{ steps} * 4800 \text{ rev.}}{\text{rev.} * 60 \text{ s}} = \frac{655360 \text{ steps}}{1 \text{ s}}$$

$$\text{Number of steps / } t_i = \frac{655360 \text{ steps}}{1 \text{ s}} * 0.05 \text{ s} = 32768 \text{ steps}$$

$$(\text{Steps/integration time}) * \text{factor} = \underline{32768 \text{ steps} / 50 \text{ ms}}$$

### 9.2.7 Object 200Ah: Speed value INT32

The object 200Ah shows the output speed value of the measuring system in the unit which is given in "Object 2005h: Speed unit" and supports PDO mapping (default: Steps/ms at scaled resolution).

The speed value is signed and is output as a two's complement.

- Code Sequence setting = increasing
  - Output positive, turning clockwise (view onto flange side)
- Code Sequence setting = decreasing
  - Output negative, turning clockwise (view onto flange side)

<b>Index</b>	<b>0x200A</b>
<b>Description</b>	Speed value INT32
<b>Data type</b>	INTEGER32
<b>Object code</b>	ARRAY
<b>Access</b>	ro
<b>PDO mapping</b>	yes

Speed value INT32			
Byte 0	Byte 1	Byte 2	Byte 3
$2^7$ to $2^0$	$2^{15}$ to $2^8$	$2^{23}$ to $2^{16}$	$2^{31}$ to $2^{24}$

If the value range of the speed is under or over the limits (-2.147.483.648... + 2.147.483.647), the limit values (0x7FFF FFFF or 0x8000 0000) will be output.

### 9.2.8 Object 2010h: Change of state - Factor

If there is no change of state (exceeding of hysteresis object 2011h), the measuring system responds with the change of state factor defined here multiplied by the value stored in Object 6200h: Cyclic timer.

<b>Index</b>	<b>0x2010</b>
<b>Description</b>	Change of state - Factor
<b>Data type</b>	UNSIGNED8
<b>Object code</b>	VARIABLE
<b>Access</b>	rw
<b>PDO mapping</b>	no
<b>Value</b>	0 = inactive (default) ≥1 = active
<b>Upper limit</b>	255

### 9.2.9 Object 2011h: Change of state - Hysteresis

Object 2011h specifies the number of steps -1 that the measuring system must take before it sends a change of state (COS message).

Index	0x2011
Description	Change of state - Hysteresis
Data type	UNSIGNED8
Object code	VARIABLE
Access	rw
PDO mapping	no
Lower limit	0
Upper limit	255

### 9.2.10 Object 2100h: COB-ID for bootup message

This object configures the COB-ID which is transmitted when the measuring system starts up (Switch-on time/RESET NODE) and is used to signal that the slave has entered the state PRE-OPERATIONAL after the state INITIALIZING. Valid values are 000h to 7FFh, default value = 700h.

Index	0x2100
Description	COB-ID Bootup-Message
Data type	UNSIGNED16
Object code	VARIABLE
Access	rw
PDO mapping	no

By means of bit  $2^{15}$  this function can be enabled/disabled:

- Bit  $2^{15} = 0$ :  
Written value into bits  $2^0$  to  $2^{10}$  is valid, with the next start up the COB-ID configured is used.
- Bit  $2^{15} = 1$ :  
Written value into bits  $2^0$  to  $2^{10}$  is not valid, with the next start up no boot-up message is transmitted.

COB-ID for Boot-Up Message			
Byte 0	Byte 1		
$2^7$ to $2^0$	$2^{10}$ to $2^8$	$2^{11}$ to $2^{14}$	$2^{15}$
00h – FFh	0h – 7h	0h	0-1

### 9.2.11 Object 2101h: Send PDO at Node-Start

About this object the one-time transmission of the measuring system position value via TPDO1 and TPDO2 at node start can be prevented.

Index	0x2101
Description	Send PDO at Node-Start
Data type	UNSIGNED8
Object code	VARIABLE
Access	rw
PDO mapping	no

Bit	Function	Bit = 0	Bit = 1
0	Send TPDO1 at Node-Start	off	on (default)
1	Send TPDO2 at Node-Start	off (default)	on
2 - 7	Reserved for further use		

### 9.2.12 Object 2110h: LSS Node-ID

This object can be used to set the Node-ID via LSS ("Configure Node-ID protocol" chapter 4.7.4.1) be determined. The measuring system must be in LSS mode. See also chapter 5.4 "Setting the Node-ID and Baud rate" on page 141.

Index	0x2110
Description	LSS NodeID
Data type	UNSIGNED8
Object code	VARIABLE
Access	rw
PDO mapping	no
Lower limit	0x01 = Node-ID 1
Upper limit	0x7F = Node-ID 127

### 9.2.13 Object 2111h: LSS BitTiming

This object can be used to set the baud rate via LSS ("Configure bit timing parameters protocol" chapter 4.7.4.2) be determined. The measuring system must be in LSS mode. See also chapter 5.4 "Setting the Node-ID and Baud rate" on page 141.

<b>Index</b>	<b>0x2111</b>
<b>Description</b>	LSS BitTiming
<b>Data type</b>	UNSIGNED8
<b>Object code</b>	VARIABLE
<b>Access</b>	rw
<b>PDO mapping</b>	No
<b>Value</b>	0x00 = 1 Mbit/s 0x01 = 800 kbit/s 0x02 = 500 kbit/s 0x03 = 250 kbit/s 0x04 = 125 kbit/s 0x05 = 100 kbit/s 0x06 = 50 kbit/s 0x07 = 20 kbit/s 0x08 = 10 kbit/s

### 9.2.14 Object 2200h: Store DMF

This object is for service purposes.

### 9.2.15 SSI-Output

The SSI output is optional and must be supported by the measurement system hardware.

#### 9.2.15.1 Object 2500h: SSI-Output Available

This object indicates if the optional SSI interface is available.

<b>Index</b>	<b>0x2500</b>
<b>Description</b>	SSI-Output Available
<b>Data type</b>	UNSIGNED8
<b>Object code</b>	VARIABLE
<b>Access</b>	ro
<b>PDO mapping</b>	no
<b>Value</b>	0x00 = SSI output is not available 0x01 = SSI output is available

### 9.2.15.2 Object 2501h: SSI-Output - Data

Defines the type of data that is output on the SSI interface.

Index	0x2501
Description	SSI-Output - Data
Data type	UNSIGNED8
Object code	VARIABLE
Access	rw
PDO mapping	no
Value	0x01 = Position (32 bit Unsigned) 0x02 = Speed (16 bit Signed) 0x03 = Position+Speed (48 bit, behind each other)

### 9.2.15.3 Object 2502h: SSI-Output - Code

Defines the output code for the SSI interface.

Index	0x2502
Description	SSI-Output - Code
Data type	UNSIGNED8
Object code	VARIABLE
Access	rw
PDO mapping	no
Value	0x00 = Gray 0x01 = Binary 0x02 = Gray cut

### 9.2.15.4 Object 2503h: SSI-Output - Number of Data-Bits

Defines the number of data bits that are output on the SSI interface.

Index	0x2503
Description	SSI-Output - Number of Data-Bits
Data type	UNSIGNED8
Object code	VARIABLE
Access	rw
PDO mapping	no
Lower limit	1 bit (0x01)
Upper limit	64 bits (0x40)

## 9.2.15.5 Object 2504h: SSI-Output - Mono-Timer

This object defines the mono time of the SSI interface.

Index	0x2504
Description	SSI-Output - Mono-Timer
Data type	UNSIGNED16
Object code	VARIABLE
Access	rw
PDO mapping	no
Lower limit	4 $\mu$ s (0x0004)
Upper limit	999 $\mu$ s (0x03E7)

## 9.2.16 Preset function

### ⚠ WARNING

### NOTICE

***Risk of injury and damage to property by an actual value jump when the Preset adjustment function is performed!***

- The preset adjustment function should only be performed when the measuring system is at rest, otherwise the resulting actual value jump must be permitted in the program and application!

### 9.2.16.1 Object 2510h: Preset-Inputs Available

This object indicates whether the preset function is available via external preset inputs.

Index	0x2510
Description	Preset-Inputs Available
Data type	UNSIGNED8
Object code	VARIABLE
Access	ro
PDO mapping	no
Value	0x00 = Preset function is not available 0x01 = Preset function is available

### 9.2.16.2 Object 2511h: Preset-Input Value 1

This object contains the preset value 1, which is set as the new position value when the external preset input 1 is connected.

Index	0x2511
Description	Preset-Input Value 1
Data type	UNSIGNED32
Object code	VARIABLE
Access	rw
PDO mapping	no
Lower limit	0
Upper limit	programmed measuring length in steps – 1
Default	0



### 9.2.16.3 Object 2512h: Preset-Input Value 2

This object contains the preset value 2, which is set as the new position value when the external preset input 2 is connected.

Index	0x2512
Description	Preset-Input Value 2
Data type	UNSIGNED32
Object code	VARIABLE
Access	rw
PDO mapping	no
Lower limit	0
Upper limit	programmed measuring length in steps – 1
Default	0

### 9.2.17 Incremental-Output

The incremental output is optional and must be supported by the measurement system hardware.

#### 9.2.17.1 Object 2520h: Incremental-Output Available

This object indicates whether the optional incremental interface is available.

Index	0x2520
Description	Incremental-Output Available
Data type	USINT8
Object code	VARIABLE
Access	ro
PDO mapping	no
Value	0x00 = Incremental output is not available 0x01 = Incremental output is available

#### 9.2.17.2 Object 2521h: Incremental-Output Pulses

This object specifies the number of pulses that can be output per revolution via the incremental interface.

Index	0x2521
Description	Incremental-Output Pulses
Data type	UNSIGNED32
Object code	VARIABLE
Access	rw
PDO mapping	no
Value	4 ... 36000 *

\* Depending on the hardware version of the measuring system.

## 9.2.17.3 Object 2522h: Incremental-Output Phase

This object defines the phase position for the incremental signals.

Index	0x2522
Description	Incremental-Output Phase
Data type	UNSIGNED8
Object code	VARIABLE
Access	rw
PDO mapping	no
Value	0x00 = K1 leads to K2 by 90° * 0x01 = K1 lagging to K2 by 90° *

\* Turning direction clockwise with view on the flange.

## 9.2.17.4 Object 2523h: Incremental-Output K0-Condition

This object defines the switching time of the Incremental track K0.

Index	0x2523
Description	Incremental-Output K0-Condition
Data type	UNSIGNED8
Object code	VARIABLE
Access	rw
PDO mapping	no
Value	0x00 = K0 if K1 high and K2 high 0x01 = K0 if K1 low and K2 high 0x02 = K0 if K1 high and K2 low 0x03 = K0 if K1 low and K2 low

Example (K0 length = ¼ period):

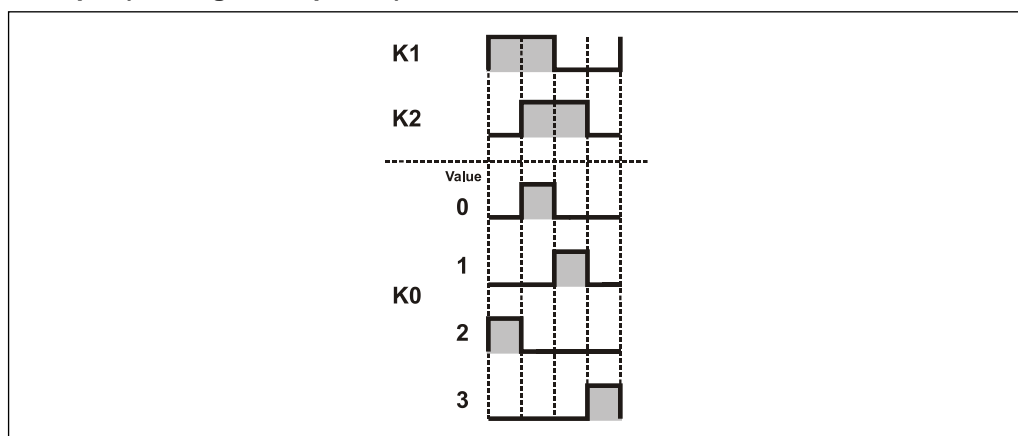


Figure 7: Example to K0 Condition

### 9.2.17.5 Object 2524h: Incremental-Output Level

This object defines the output level of the incremental signals.

Index	0x2524
Description	Incremental-Output Level
Data type	UNSIGNED8
Object code	VARIABLE
Access	rw
PDO mapping	no
Value	0x00 = TTL, 5 VDC; Output driver: RS422 0x01 = HTL, US Supply voltage; Output driver: Push-Pull The supply voltage must be > 8 VDC

### 9.2.17.6 Object 2525h: Incremental-Output K0 Len

This object defines the length of the K0 signal.

Index	0x2525
Description	Incremental-Output K0 Len
Data type	UNSIGNED8
Object code	VARIABLE
Access	rw
PDO mapping	no
Value	0x00 = K0 lasts ¼ period 0x01 = K0 lasts ½ period * 0x02 = K0 lasts ¾ period * 0x03 = K0 lasts one period * 0x04 = K0 lasts two periods * 0x05 = K0 lasts four periods *

\* Depending on the hardware version of the measuring system.

### 9.2.17.7 Object 2526h: Incremental-Output Set K0

This object sets the zero pulse K0 to the current measuring system position plus an offset value that is transferred with write access.

Index	0x2526
Description	Incremental-Output Set K0
Data type	UNSIGNED32
Object code	VARIABLE
Access	rw
PDO mapping	no
Lower limit	0 = K0 is set to the current position
Upper limit	programmed number of pulses (chapter: 9.2.17.2)

### 9.2.18 External Display

The functions for an external display are optional and must be supported by the measuring system hardware.

#### 9.2.18.1 Object 2530h: Display available

This object indicates if an external display is available.

Index	0x2530
Description	Display Available
Data type	UNSIGNED8
Object code	VARIABLE
Access	ro
PDO mapping	no
Value	0x00 = External display is not available 0x01 = External display is available

#### 9.2.18.2 Objects 2531h ... 2534h: Display Button 1, 2, 3 and 4 function

These objects contain looped switching functions for the display.

Index	0x2531, 0x2532, 0x2533, 0x2534
Description	Display Button # function
Data type	UNSIGNED8
Object code	VARIABLE
Access	rw
PDO mapping	no

#### 9.2.18.3 Object 2535h: Display Control

Via this object, display-specific settings can be made.

Index	0x2535
Description	Display Control
Data type	UNSIGNED8
Object code	VARIABLE
Access	rw
PDO mapping	no

#### 9.2.18.4 Object 2536h: Display Data Numeric

The measuring system position is displayed numerically as a decimal number.

Index	0x2536
Description	Display Data Numeric
Data type	UNSIGNED32
Object code	VARIABLE
Access	rw
PDO mapping	no
Lower limit	- 2 147 483 648 = (0x8000 0000)
Upper limit	+ 2 147 483 647 = (0x7FFF FFFF)

#### 9.2.18.5 Object 2537h: Display Data ASCII

The measuring system position is displayed in ASCII code.

Index	0x2537
Description	Display Data ASCII
Data type	VISIBLE_STRING
Object code	VARIABLE
Access	rw
PDO mapping	no

#### 9.2.18.6 Object 2538h: Display Status

The status of the display can be read out via this object.

Index	0x2538
Description	Display Status
Data type	UNSIGNED8
Object code	VARIABLE
Access	ro
PDO mapping	no

### 9.2.19 TR-Production

The objects "TR-Produktion" are reserved for manufacturer-specific functions.

## 9.3 CiA DS-406 - Mode

### 9.3.1 Object 6000h: Operating parameters

The object with index 6000h supports only the function for the code sequence.

<b>Index</b>	<b>0x6000</b>
<b>Description</b>	Operating parameters
<b>Data type</b>	UNSIGNED16
<b>Object code</b>	VARIABLE
<b>Access</b>	rw
<b>PDO mapping</b>	no

Bit	Function	Bit = 0, default	Bit = 1
0	Code Sequence	increasing	decreasing
1	Reserved for further use		
2	Scaling function	set on 1, can't be changed	
3 - 15	Reserved for further use		

The code sequence defines whether increasing or decreasing position values are output when the measuring system shaft rotates clockwise or counter clockwise as seen on the shaft.

### 9.3.2 Scaling parameters

---

***Danger of personal injury and damage to property exists if the measuring system is restarted after positioning in the de-energized state by shifting of the zero point!***

If the number of revolutions is not an exponent of 2 or is >4096, it can occur, if more than 512 revolutions are made in the de-energized state, that the zero point of the multi-turn measuring system is lost!

#### **WARNING**

#### **NOTICE**

- Ensure that the **Number of Revolutions** for a multi-turn measuring system is an exponent of 2 of the group  $2^0, 2^1, 2^2 \dots 2^{12}$  (1, 2, 4...4096).  
or...
  - Ensure that every positioning in the de-energized state for a multi-turn measuring system is within 512 revolutions.
-

### 9.3.2.1 Object 6001h: Measuring units per revolution

The object "Measuring units per revolution" sets the number of steps per revolution.

Index	0x6001
Description	Measuring units per revolution
Data type	UNSIGNED32
Object code	VARIABLE
Access	rw
PDO mapping	no
Lower limit	1 step / revolution
Upper limit	See max. value on the measuring system nameplate
Default	4096

Measuring units per revolution			
Byte 0	Byte 1	Byte 2	Byte 3
$2^7$ to $2^0$	$2^{15}$ to $2^8$	$2^{23}$ to $2^{16}$	$2^{31}$ to $2^{24}$

### 9.3.2.2 Object 6002h: Total measuring range in measuring units

This object defines the **total number of steps** of the measuring system before the measuring system restarts at zero.

Index	0x6002
Description	Total measuring range in measuring units
Data type	UNSIGNED32
Object code	VARIABLE
Access	rw
PDO mapping	no
Lower limit	16 steps
Upper limit	$4\,294\,967\,295 = (0xFFFF\,FFFF) *$
Default	16777216

\* It can be set "0" to display the value  $0x100000000$  ( $2^{32}$ ).

Total measuring range			
Byte 0	Byte 1	Byte 2	Byte 3
$2^7$ to $2^0$	$2^{15}$ to $2^8$	$2^{23}$ to $2^{16}$	$2^{31}$ to $2^{24}$

The actual upper limit for the measurement length to be entered in steps is dependent on the measuring system version and can be calculated with the formula below. As the value "0" is already counted as a step, the end value = measurement length in steps - 1.

$$\text{Total measuring range} = \text{Steps per revolution} * \text{Number of revolutions}$$

To calculate, the parameters **steps/rev.** and **the number of revolutions** can be read on the measuring system nameplate.

The Parameter „Number of revolutions“, which results out of the „total measuring range in steps“ and „measuring steps per revolution“ has following restriction:

The "number of revolutions" may be a decimal number which can be represented as fraction in the following area:

$$\frac{1...256000}{1...16384} = \text{Number of revolutions}$$

### Example 1:

#### **Assumed:**

- Measuring range in steps = 16777216
- Steps per revolution = 2048

#### **Derived:**

$$\frac{16777216 \text{ steps}}{2048 \text{ steps/revolution.}} = 8192 \text{ revolutions} = \frac{8192}{1} \text{ revolutions} \Rightarrow \text{possible}$$

### Example 2:

#### **Assumed:**

- Measuring range in steps = 10000000
- Steps per revolution = 3600

#### **Derived:**

$$\frac{10000000 \text{ steps}}{3600 \text{ steps/revolution.}} = 2777,7 \text{ revolutions} = \frac{25000}{9} \text{ revolutions} \Rightarrow \text{possible}$$

---

If the resulting number of revolutions cannot be represented in this area, then the "Measuring range in steps" is corrected to the next smaller value.



The newly calculated total measuring range can be read from the Object 6002h and is always shorter than the specified measurement length. It may therefore occur that the total number of steps actually required is not achieved and the measuring system generates a zero transition before it reaches the maximum mechanical distance.



### 9.3.3 Object 6003h: Preset value

#### ⚠ WARNING

#### NOTICE

***Risk of injury and damage to property by an actual value jump when the Preset adjustment function is performed!***

- The preset adjustment function should only be performed when the measuring system is at rest, otherwise the resulting actual value jump must be permitted in the program and application!

The Preset Function can be used to adjust the measuring system to any position value within a range of 0 to measuring length in increments –1. The output position value is set to the parameter "Preset value" when writing to this object.

At writing of an invalid preset value, the measuring system outputs the abort code 0600 0030h.

Index	0x6003
Description	Preset value
Data type	INTEGER32
Object code	VARIABLE
Access	rw
PDO mapping	no

Preset value			
Byte 0	Byte 1	Byte 2	Byte 3
$2^7$ to $2^0$	$2^{15}$ to $2^8$	$2^{23}$ to $2^{16}$	$2^{31}$ to $2^{24}$

### 9.3.4 Object 6004h: Position value

The object 6004h "Position value" defines the output position value and the supported PDO mapping.

Index	0x6004
Description	Position value
Data type	INTEGER32
Object code	VARIABLE
Access	ro
PDO mapping	yes

Position value			
Byte 0	Byte 1	Byte 2	Byte 3
$2^7$ to $2^0$	$2^{15}$ to $2^8$	$2^{23}$ to $2^{16}$	$2^{31}$ to $2^{24}$

### 9.3.5 Object 6030h: Speed value

The object 6030h shows in sub-index 1 the output speed value, which is given in steps per ms, the resolution is given in "Object 2005h: Speed unit".

The speed value is signed and is output as a two's complement.

- Code Sequence setting = increasing
  - Output positive, turning clockwise (view onto flange side)
- Code Sequence setting = decreasing
  - Output negative, turning clockwise (view onto flange side)

Index	0x6030
Description	Speed value
Data type	INTEGER16
Object code	ARRAY
Access	ro
PDO mapping	yes

Speed value	
Byte 0	Byte 1
$2^7$ to $2^0$	$2^{15}$ to $2^8$

If the value range of the speed is under or over the limits of -32768...+32767, the limit values (0x7FFF or 0x8000) will be output.

### 9.3.6 Object 6200h: Cyclic timer

Defines the parameter "Cyclic timer". A Cyclic transmission of the position value is set, when the cyclic timer is programmed > 0. Values between 1 ms and 65535 ms can be selected. Default value = 0.

e.g.: 1 ms = 1 h  
256 ms = 100 h

When the measuring system is started with the NODE START Command and the value of the cyclic timer is > 0, in the default setting the 1st transmit PDO (object 1800h) transmit the measuring system position.

Index	0x6200
Description	Cyclic timer
Data type	UNSIGNED16
Object code	VARIABLE
Access	rw
PDO mapping	no



*The event-timer sub-index 5 of the communication parameter 1800h is hard-wired with the cyclic timer. That means that a change in the event timer causes a change in the cyclic timer and vice versa. The communication parameter 1801h exclusively uses its own timer, access over sub-index 5.*

### 9.3.7 Object 6400h: Area state register

With the objects "Object 6401h: Work area low limit" and "Object 6402h: Work area high limit" together, a work area can be defined within the measuring range. The "Object 6400h: Area state register" contains the current area status of the measuring system position in sub-index 1. This function can be used to save external position switches.

Index	0x6400
Description	Area state register
Data type	UNSIGNED8
Object code	ARRAY
Access	ro
PDO mapping	yes

Bit	Function
0	reserved
1	1 = above the work area
2	1 = below the work area
3 – 7	reserved

### 9.3.8 Object 6401h: Work area low limit

The "Work area low limit" object defines the lower limit of the working range. The value must be within the measuring range and below the "Work area high limit".

Index	0x6401
Description	Work area low limit
Data type	INTEGER32
Object code	ARRAY
Access	rw
PDO mapping	no
Lower limit	- 2 147 483 648 = (0x8000 0000)
Upper limit	+ 2 147 483 647 = (0x7FFF FFFF)

### 9.3.9 Object 6402h: Work area high limit

The "Work area high limit" object defines the upper limit of the working range. The value must be within the measuring range and above the "working range low limit".

Index	0x6402
Description	Work area high limit
Data type	INTEGER32
Object code	ARRAY
Access	rw
PDO mapping	no
Lower limit	- 2 147 483 648 = (0x8000 0000)
Upper limit	+ 2 147 483 647 = (0x7FFF FFFF)

## 9.4 Measuring system diagnostics

### 9.4.1 Object 6500h: Operating status

This object contains the operating status of the measuring system. It gives information on measuring system internal programmed parameters.

Index	0x6500
Description	Operating status
Data type	UNSIGNED16
Object code	VARIABLE
Access	ro
PDO mapping	no

Bit	Function	Bit = 0	Bit = 1
0	Code Sequence	increasing	decreasing
1	Reserved for further use		
2	Constant		X
3 - 15	Reserved for further use		

### 9.4.2 Object 6501h: Single turn resolution

The object 6501h contains the number of measuring steps per revolution which can be output by the measuring system.

Index	0x6501
Description	Singleturn resolution
Data type	UNSIGNED32
Object code	VARIABLE
Access	ro
PDO mapping	no

Single turn resolution			
Byte 0	Byte 1	Byte 2	Byte 3
$2^7$ to $2^0$	$2^{15}$ to $2^8$	$2^{23}$ to $2^{16}$	$2^{31}$ to $2^{24}$

Standard value: depending on capacity marked on nameplate

### 9.4.3 Object 6502h: Number of distinguishable revolutions

This object contains the number of distinguishable revolutions that the measuring system can output.

Index	0x6502
Description	Number of distinguishable revolutions
Data type	UNSIGNED32
Object code	VARIABLE
Access	ro
PDO mapping	no

For a Multi-Turn measuring system the number of distinguishable revolutions and the Single-Turn resolution gives the measuring range according to the formula below. The maximum number of distinguishable revolutions is 256.000 (18 bits).

$\text{Measuring range} = \text{Number of distinguishable revolutions} \times \text{Single-Turn resolution}$
--

Standard value: 59392 = E800h revolutions.

Since this object only can store a 16-bit value, the high-order word with the number 3E800h (256.000) is not represented.

## 9.4.4 Object 6503h: Alarms

Additionally to the emergency message, object 6503h provides further alarm messages which also can show over the PDO mapping on the process data. An alarm is set if a malfunction in the measuring system could lead to an incorrect position value. If an alarm occurs, the according bit is set to logical high until the alarm is cleared and the measuring system is able to provide an accurate position value.

Index	0x6503
Description	Alarms
Data type	UNSIGNED16
Object code	VARIABLE
Access	ro
PDO mapping	no

Bit	Function	Bit = 0	Bit = 1
0	Position error	No	Yes
1	Reserved for further use		
2	Reserved for further use		
3	Reserved for further use		
4	Reserved for further use		
5	Reserved for further use		
6	Reserved for further use		
7	Reserved for further use		
8	Reserved for further use		
9	Reserved for further use		
10	Reserved for further use		
11	Reserved for further use		
12	EE-PROM Error	OK	Error
13	Reserved for further use		
14	Manufacturer specific functions		
15	Manufacturer specific functions		

### Position error

The bit is set, if the measuring system detects a malfunction of the system

### EE-PROM error

The measuring system detects a wrong checksum in the EEPROM area or a write process into the EEPROM could not be finished successfully.

### 9.4.5 Object 6504h: Supported alarms

The object 6504h contains the information on supported alarms by the measuring system.

Index	0x6504
Description	Supported alarms
Data type	UNSIGNED16
Object code	VARIABLE
Access	ro
PDO mapping	no

Bit	Function	Bit = 0	Bit = 1
0	Position error	No	Yes
1	Reserved for further use		
2	Reserved for further use		
3	Reserved for further use		
4	Reserved for further use		
5	Reserved for further use		
6	Reserved for further use		
7	Reserved for further use		
8	Reserved for further use		
9	Reserved for further use		
10	Reserved for further use		
11	Reserved for further use		
12	EE-PROM error	No	Yes
13	Manufacturer specific functions		
14	Manufacturer specific functions		
15	Manufacturer specific functions		

### 9.4.6 Object 6505h: Warnings

This object contains the warnings and indicates that tolerance for certain internal parameters of the measuring system have been exceeded. In contrast to alarm and emergency messages warnings do not imply incorrect position values.

Index	0x6505
Description	Warnings
Data type	UNSIGNED16
Object code	VARIABLE
Access	ro
PDO mapping	no

Bit	Function	Bit = 0	Bit = 1
0	Speed warning	No	Yes
1	Reserved for further use		
2	Reserved for further use		
3	Reserved for further use		
4	Reserved for further use		
5	Reserved for further use		
6	Reserved for further use		
7	Reserved for further use		
8	Reserved for further use		
9	Reserved for further use		
10	Reserved for further use		
11	Reserved for further use		
12	Temperature warning	No	Yes
13	Manufacturer specific functions		
14	Manufacturer specific functions		
15	Manufacturer specific functions		

#### Limits:

The limit values can be seen on the device specific data sheet.



*All warnings are cleared if the tolerances are again within normal parameters.*



### 9.4.7 Object 6506h: Supported warnings

The object 6506h contains the information of supported warnings by the measuring system.

Index	0x6506
Description	Supported warnings
Data type	UNSIGNED16
Object code	VARIABLE
Access	ro
PDO mapping	no

Bit	Function	Bit = 0	Bit = 1
0	Speed warning	No	Yes
1	Reserved for further use		
2	Reserved for further use		
3	Reserved for further use		
4	Reserved for further use		
5	Reserved for further use		
6	Reserved for further use		
7	Reserved for further use		
8	Reserved for further use		
9	Reserved for further use		
10	Reserved for further use		
11	Reserved for further use		
12	Temperature warning	No	Yes
13	Manufacturer specific functions		
14	Manufacturer specific functions		
15	Manufacturer specific functions		

### 9.4.8 Object 6507h: Profile and software version

This object contains in the 1st 16 bits the profile version which is implemented in the measuring system. It is combined to a revision number and an index.

Index	0x6507
Description	Profile and software version
Data type	UNSIGNED32
Object code	VARIABLE
Access	ro
PDO mapping	no

Example: Profile version: 3.2  
 Binary code: 0000 0011 0000 0010  
 Hexadecimal: 03 02

The 2nd 16 bits contain the index of the software version out of object 100Ah.

Example: Software version index: 1.02  
 Binary code: 0000 0001 0000 0110  
 Hexadecimal: 01 06

The software version without the index is contained in object 100Ah, see page 151.

Profile version		Software version index	
Byte 0	Byte 1	Byte 2	Byte 3
$2^7$ to $2^0$	$2^{15}$ to $2^8$	$2^7$ to $2^0$	$2^{15}$ to $2^8$

### 9.4.9 Object 6508h: Operating time

The operating time is stored in the encoder nonvolatile memory as long as the encoder is power supplied.

The value is given in 0.1 hours per digit.

Index	0x6508
Description	Operating time
Data type	UNSIGNED32
Object code	VARIABLE
Access	ro
PDO mapping	no

#### 9.4.10 Object 6509h: Offset value

This object contains the offset value calculated by the preset function. The offset value is stored and can be read from the measuring system.

Index	0x6509
Description	Offset value
Data type	UNSIGNED32
Object code	VARIABLE
Access	ro
PDO mapping	no

#### 9.4.11 Object 650Ah: Manufacturer offset value

This object contains in sub-index 1 the measuring system offset value.

Index	0x650A
Description	Manufacturer offset value
Data type	UNSIGNED32
Object code	ARRAY
Access	ro
PDO mapping	no

#### 9.4.12 Object 650Bh: Serial number

This object contains the current Serial-No. of the device and corresponds to the Identity-Object 1018h, Sub-index 4.

Index	0x650B
Description	Serial number
Data type	UNSIGNED32
Object code	VARIABLE
Access	ro
PDO mapping	no

### 10 Emergency Message

Emergency messages are triggered by the occurrence of a device internal malfunction and are transmitted from the concerned application device to the other devices with highest priority.

Emergency Message								
Byte	0	1	2	3	4	5	6	7
Contents	Emergency Error Code Object 1003h, Byte 0-1		Error Register Object 1001h	Alarms Object 6503		Warnings Object 6505		0

#### Default COB-Identifier = 080h + Node-ID

If the measuring system detects an internal error, an emergency message will be transmitted with the error code of object 1003h (pre-defined error field) and the error register object 1001h.

If the error disappears, the measuring system transmits an emergency message with error code "0" (reset error / no error) and error register "0".

## 11 Causes of faults and remedies

### 11.1 Optical displays

The number, position and assignment of the status LEDs depends on the device version and can be found in the device specific pin assignment.



*Via the "Object 1029h: Error behavior object" it can be set, whether the measuring system change after an error from the OPERATIONAL in the PRE-OPERATIONAL or in the STOPPED-Mode, see page 157. In this case the emergency message should be evaluated, see "Emergency Error codes" on page 201.*

#### 11.1.1 Measuring system with two status LEDs

##### LED1, Device Status:

LED status	Cause	Remedy
OFF	Voltage supply absent or was fallen below	<ul style="list-style-type: none"> <li>- Check voltage supply wiring</li> <li>- Does the voltage supply is in the permitted range?</li> </ul>
	Hardware fault, measuring system defective	Replace measuring system
ON green	Device operates fault free	-
Blinking green	Device is operated out of specification (e.g., rotational speed, temperature)	Check parameters and specifications
ON red	Device is not ready for operation	Switch off the supply voltage, then switch it on again. If this measure does not succeed, the measuring system must be replaced.
Blinking red	Device has detected an error (e.g. EEPROM access error)	
Flickering red	Device has detected a defect (e.g. position error)	

### LED2, Net Status:

LED status	Cause	Remedy
OFF	Voltage supply absent or too low	<ul style="list-style-type: none"> <li>- Check voltage supply, wiring</li> <li>- Is the voltage supply in the permissible range?</li> </ul>
	Hardware fault, measuring system defective	<ul style="list-style-type: none"> <li>- Replace measuring system</li> </ul>
ON (green)	Measuring system is in the <i>OPERATIONAL</i> mode	<ul style="list-style-type: none"> <li>- -</li> </ul>
Blinking (green)	Measuring system is in <i>PRE OPERATIONAL</i> mode	<ul style="list-style-type: none"> <li>- No error, measuring system can be transferred into <i>OPERATIONAL</i> mode</li> </ul>
Single flash (green)	Measuring system is in <i>SOPPED</i> mode	
Flickering (green)	LSS configuration active	<ul style="list-style-type: none"> <li>- No error, measuring system can be transferred into LSS waiting state</li> </ul>
ON (red)	Bus offline by reason of <ul style="list-style-type: none"> <li>- interchanged CAN lines</li> <li>- interrupted CAN lines</li> <li>- incorrect adjusted baud rate</li> </ul>	<ul style="list-style-type: none"> <li>- Check bus line</li> <li>- Check plug connector</li> <li>- Adjusted baud rate must agree with the master baud rate</li> </ul>
Single flash (red)	To much errors on the CAN controller	<ul style="list-style-type: none"> <li>- Switch off the supply voltage, then switch it oagain.</li> <li>- Check the physical structure of the bus system</li> </ul>
Double flash (red)	Node Guarding error	<ul style="list-style-type: none"> <li>- General bus load <math>\leq 85\%</math> !</li> <li>- Attempt to increase the baud rate</li> <li>- Increase cycle time for node guarding protocol by means of the objects 100Ch and 100Dh</li> <li>- Switch off the supply voltage, then switch it on again.</li> </ul>
	Heartbeat error	<ul style="list-style-type: none"> <li>- General bus load <math>\leq 85\%</math> !</li> <li>- Attempt to increase the baud rate</li> <li>- Adapt cycle time for heartbeat protocol by means of the objects 1016h or 1017h</li> </ul>

### 11.1.2 Measuring system with one status LED

LED status	Cause	Remedy
OFF	Voltage supply absent or too low	<ul style="list-style-type: none"> <li>- Check voltage supply, wiring</li> <li>- Is the voltage supply in the permissible range?</li> </ul>
	Hardware fault, measuring system defective	<ul style="list-style-type: none"> <li>- Replace measuring system</li> </ul>
ON (green)	Measuring system is in the <i>OPERATIONAL</i> mode	-
Blinking (green)	Measuring system is in <i>PRE OPERATIONAL</i> mode	<ul style="list-style-type: none"> <li>- No error, measuring system can be transferred into OPERATIONAL mode</li> </ul>
Single flash (green)	Measuring system is in <i>SOPPED</i> mode	
Flickering (green)	LSS configuration active	<ul style="list-style-type: none"> <li>- No error, measuring system can be transferred into LSS waiting state</li> </ul>
ON (red)	Bus offline by reason of <ul style="list-style-type: none"> <li>- interchanged CAN lines</li> <li>- interrupted CAN lines</li> <li>- incorrect adjusted baud rate</li> </ul>	<ul style="list-style-type: none"> <li>- Check bus line</li> <li>- Check plug connector</li> <li>- Adjusted baud rate must agree with the master baud rate</li> </ul>
Single flash (red)	To much errors on the CAN controller	<ul style="list-style-type: none"> <li>- Switch off the supply voltage, then switch it on again.</li> <li>- Check the physical structure of the bus system</li> </ul>
Double flash (red)	Node Guarding error	<ul style="list-style-type: none"> <li>- General bus load <math>\leq 85\%</math> !</li> <li>- Attempt to increase the baud rate</li> <li>- Increase cycle time for node guarding protocol by means of the objects 100Ch and 100Dh</li> <li>- Switch off the supply voltage, then switch it on again.</li> </ul>
	Heartbeat error	<ul style="list-style-type: none"> <li>- General bus load <math>\leq 85\%</math> !</li> <li>- Attempt to increase the baud rate</li> <li>- Adapt cycle time for heartbeat protocol by means of the objects 1016h or 1017h</li> </ul>
Flickering (red)	Hardware fault, measuring system defective	<ul style="list-style-type: none"> <li>- Replace measuring system</li> </ul>

## 11.2 SDO Error codes

In the case of an error (SDO response CCD = 0x80) the data field contains a 4-byte error code. By the measuring system the following error codes are supported:

Error code	Meaning	Remedy
0x0503 0000	Toggle bit not alternated.	An error occurred at segmented transmission of a SDO - repeat the procedure
0x0504 0001	Not valid or unknown command code (CCD)	List of valid CCD's see Table 3 on page 119.
0x0601 0001	Attempt to read a write only object.	False Command Code (CCD), only write commands (0x2x) are permitted, see Table 3 on page 119.
0x0601 0002	Attempt to write a read only object.	False Command Code (CCD), only read commands (0x4x) are permitted, see Table 3 on page 119.
0x0602 0000	Object does not exist in the object dictionary.	Valid objects see Table 8 and Table 9 on page 148 and 162.
0x0604 0041	Object cannot be mapped	Provided mapping objects see chapter 7.2.1 on page 146
0x0604 0042	The number and length of the objects to be mapped would exceed PDO length	Check - Mapping objects $\leq 8$ byte data length per TPDO - Number of mapping objects $\leq 2$ per TPDO
0x0607 0010	Data type does not match, length of service parameter does not match.	The used Command Code (CCD) does not match with the data length of the transferred object. Compare Command Codes on page 119 with the objects, see Table 8 and Table 9 on page 148 and 162.
0x0607 0012	Data type or length of service parameter too big	The used Command Code (CCD) is longer than the data length of the transferred object. Compare Command Codes on page 119 with the objects, see Table 8 and Table 9 on page 148 and 162.
0x0607 0013	Data type or length of service parameter too small	The used Command Code (CCD) is shorter than the data length of the transferred object. Compare Command Codes on page 119 with the objects, see Table 8 and Table 9 on page 148 and 162.
0x0609 0011	Sub-index does not exist.	Check which sub-indices the corresponding object supports.
0x0609 0031	Invalid parameter value (download only)	Check the valid value of the object
0x0609 0031	Transmitted value of parameter too high.	Check the valid range of the object
0x0609 0032	Transmitted value of parameter too low.	Check the valid range of the object
0x0800 0020	Data cannot be transferred or stored to the application	Wrong signature written when storing/restoring the parameters, see objects 1010h/1011h, page 153/154.
0x0800 0021	In case of the local control, the data cannot be transferred or stored to the application	Wrong Mode Selection, see Object 2000h: Mode selection TR / CiA DS-406 on page 163 or wrong state for the Object 1F51h: Program control on page 158.
0x0800 0022	In case of the device status, data cannot be transferred	Wrong procedure at mapping configuration, see Procedure for re-mapping on page 146.
0x0800 0024	No data available	Indication that no more errors are existing at read access on object 1003h sub-index 1, see page 150.

Table 10: SDO Error codes



## 11.3 Emergency Error codes

Emergency objects are triggered by the occurrence of a device internal error situation, transmission format see chapter "Emergency Message", page 196.

The error indication is carried out about the objects

- Error register 0x1001, page 149 and
- Pre-defined error field 0x1003, page 150

### 11.3.1 Object 1001h: Error register

The error register displays bit coded the error state of the measuring system. Also several errors at the same time can be displayed by a set bit. The error code of the error occurred last is stored in object 0x1003, sub-index 1, the number of errors in sub-index 0. An error is signaled at the moment of the occurrence by an EMCY-message. By reading of the object 1001h the error stored last in object 0x1003, sub-index 0, is cleared. Each further read request clears a further error from the list. With the clearing of the last error the error register is set back and an EMCY-message with error code "0x000" is transferred.

Bit	Meaning
0	generic error
1	0
2	0
3	0
4	communication error (overflow, error state)
5	0
6	0
7	0

### 11.3.2 Object 1003h: Pre-defined Error field, bits 0 – 15

About the Emergency object only the error occurred last is indicated. For each EMCY-message which could be deleted an EMCY-report with error code "0x0000" is transmitted. The result can be taken from object 0x1003. If no more error is present, the error register indicates also no more error.

The error list in object 0x1003 can be deleted in different ways:

1. Writing a "0" to sub-index 0 in object 0x1003
2. Execution of the NMT-service "Reset Communication", command 0x82
3. Reading the object 0x1001, after the last error was deleted

Error code	Meaning	Remedy
0x0000	reset error / no error	-
0x5000	Device hardware – generic error	-
0x8100	Communication errors, which are triggered by the CAN-controller.	<ul style="list-style-type: none"> <li>- Reset node with command 0x81, after that start the node again with command 0x01.</li> <li>- Try to restart the device: Voltage OFF/ON.</li> </ul>
0x8130	Life guard error	<ul style="list-style-type: none"> <li>- General bus load <math>\leq 85\%</math> !</li> <li>- Attempt to increase the baud rate</li> <li>- Increase cycle time for node guarding protocol by means of the objects 100Ch and 100Dh</li> <li>- Try to restart the device: Voltage OFF/ON.</li> </ul>
	Heartbeat error	<ul style="list-style-type: none"> <li>- General bus load <math>\leq 85\%</math> !</li> <li>- Attempt to increase the baud rate</li> <li>- Adapt cycle time for heartbeat protocol by means of the objects 1016h or 1017h</li> </ul>

Table 11: Emergency Error codes

### 11.4 Alarm messages

About the object 6503h additionally to the EMCY-message further alarm messages are output. The corresponding error bit is deleted, if the error is present no more.

Error	Cause	Remedy
Bit 0 = 1, Position error	Failure of scanning elements in the measuring system	Possibly shut-off measuring system voltage then switch on again. If the error recurs despite this measure, the measuring system must be replaced.
Bit 12 = 1, EE-PROM error	Memory area in internal EE-PROM defective	

Table 12: Alarm messages

## 11.5 Other faults

Fault	Cause	Remedy
Position skips of the measuring system	Strong vibrations	Vibrations, impacts and shocks, e.g. on presses, are dampened with "shock modules". If the error recurs despite these measures, the measuring system must be replaced.
	Electrical faults EMC	Perhaps isolated flanges and couplings made of plastic help against electrical faults, as well as cables with twisted pair wires for data lines. Shielding and wire routing must be performed according to the respective field-bus system construction guidelines.
	Extreme axial and radial load on the shaft or a defective scanning unit.	Couplings prevent mechanical stress on the shaft. If the error still occurs despite these measures, the measuring system must be replaced.

**Table 13: Other faults**