

Betriebsanleitung Dehnungssensoren CANopen DS 404



DSRT 22DJ-S5-xxxx

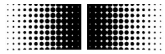


DST55R-28.xxx.TC1.A5

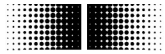
Baumer Electric AG
Hummelstrasse 17
Postfach
CH-8501 Frauenfeld
www.baumer.com

Änderungen in Technik und Design vorbehalten.

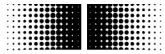
Ver. 3.07



1	Allgemeines	5
1.1	Sicherheitshinweise.....	5
1.2	Montage und Inbetriebnahme	5
2	Projektierung	6
2.1	Maximaler Systemausbau	6
3	Anschlüsse	7
3.1	Elektrischer Anschluss	7
3.1.1	Anschlussbelegung	7
3.1.2	Anschluss-Skizze	7
3.2	Potentialverhältnisse	8
3.3	EMV-gerechte Verdrahtung	9
3.3.1	Erdung inaktiver Metallteile	9
3.3.2	Schirmung von Leitungen.....	9
3.4	Spezifikation der CAN-Leitungen	9
4	CANopen	10
4.1	Einleitung.....	10
4.2	Signale, Aufbau und Bus Topologie.....	10
4.2.1	Bus-Signale	10
4.2.2	Netzwerk-Topologie	11
4.2.3	Aufbau einer CAN Nachricht	12
4.2.4	Bitweise Busarbitrierung.....	13
4.2.5	Prioritätsorientierte Nachrichtenübertragung	14
4.2.6	Identifizier-Verteilung	14
4.3	Objekte	15
4.4	Kommunikations-Mechanismen	16
4.4.1	Process Data Objects (PDO)	16
4.4.2	Service Data Objects (SDO)	17
4.4.3	Network Management (NMT).....	18
4.4.4	Emergency (EMGY)	20
4.4.5	Node-Guarding und Heartbeat.....	21
4.5	Weitere Begriffserklärung	22
4.5.1	Boot-Up Nachricht	22
4.5.2	EDS	22
4.5.3	DCF	22
4.5.4	LSS.....	22
5	CANopen Protokoll	23
5.1	Allgemeines	23
5.1.1	Bootloader	23
5.2	Network Management	23
5.2.1	Predefined Connection Set	23
5.2.2	Startprozedere	24
5.2.3	Start Node	25
5.2.4	Stop Node.....	25
5.2.5	Pre-Operational Node	25
5.2.6	Reset Node.....	25
5.3	Übersicht der unterstützten Objekte.....	26
5.4	SDO-Struktur	28
6	Beschreibung der Objekte.....	29
6.1	Standard Objekte	29
6.1.1	Geräteprofil.....	29



6.1.2	Kalibrationsdatum.....	29
6.1.3	Gerätebezeichnung	29
6.1.4	Hardware	30
6.1.5	Software	30
6.1.6	Geräte Identität.....	30
6.2	Parameter Handling (save, load default).....	32
6.2.1	Speichern	32
6.2.2	Defaultwerte laden	33
6.3	Gerätespezifische Objekte	34
6.3.1	Sensortyp	34
6.3.2	Betriebsart	34
6.3.3	Tarierung.....	35
6.3.4	Einheit Prozesswert.....	35
6.3.5	Dezimalstellen Prozesswert	35
6.3.6	Status der Messung	36
6.3.7	Abfrage Prozesswert 16bit	36
6.3.8	Abfrage Prozesswert 24bit	37
6.3.9	Delta-Prozesswert	37
6.4	Herstellerspezifische Objekte.....	39
6.4.1	Mittelungszeit.....	39
6.4.2	Tarierung speichern	40
6.4.3	IIR Filter Grenzfrequenz	40
6.4.4	Autozero	41
6.4.5	Status Autozero	41
6.4.6	Baudrate	41
6.4.7	Identifikation	42
6.4.8	Datentyp 16/24bit	43
6.5	PDO-Kommunikation Objekte	44
6.5.1	Empfangs PDO 1 (Tarierung).....	44
6.5.2	Empfangs PDO 1 Mapping Parameter.....	45
6.5.3	Sende PDO 1	45
6.5.4	Sende PDO 2	47
6.5.5	Sende PDO 3	47
6.5.6	Sende PDO 1 Mapping Parameter.....	47
6.5.7	Sende PDO 2 Mapping	48
6.5.8	Sende PDO 3 Mapping	48
6.5.9	Sync ID	48
7	Fehlermeldungen und Dienste.....	49
7.1	Error Register & History	49
7.1.1	Error Register	49
7.1.2	Emergency History	49
7.2	SDO-Fehlermeldungen	51
7.3	Emergency Messages.....	52
7.4	Heartbeat.....	53
7.4.1	Heartbeat Time.....	53
7.5	LSS (Layer Setting Services).....	54
7.5.1	Aufgedruckte LSS-Daten auf Sensor	54
7.5.2	Ansprechen des Sensors über LSS	54
7.5.3	Konfigurationsmodus direkte Verbindung (Master-Sensor)	56
7.5.4	Konfigurationsmodus von einem Sensor in einem Netzwerk.....	56
7.5.5	Verstellen der ID und der Baudrate	57
7.5.6	Speichern der Änderungen	58



7.5.7	LSS Mode verlassen	58
8	Anwenderbeispiele CANopen Protokoll	59
8.1	Tarieren des Prozesswerts über SDO und PDO	59
8.2	Abfragen eines Prozesswerts über SDO (16 und 24bit)	60
8.3	Konfigurieren des Prozesswerts über PDO1 (16 und 24bit)	61
8.4	ID Ändern (Objekt 2101 oder LSS)	63
8.5	Baudrate ändern (Objekt 21'00h oder LSS)	64
9	Dokument-Revisions-History	65

1 Allgemeines

Diese Betriebsanleitung enthält wichtige Informationen für den sicheren und bestimmungsgemässen Gebrauch des CANopen Dehnungssensoren und muss vor Inbetriebnahme unbedingt gelesen werden. Es wurde für Personal erarbeitet, welches im Umgang mit elektrischen Geräten geschult und qualifiziert ist. Weiter findet sich eine kleine Einführung und Begriffserklärung von CANopen und nützliche Hinweise für eine einwandfreie Funktionalität des Dehnungssensors.

1.1 Sicherheitshinweise

- Der Dehnungssensor ist ein kompaktes und hochsensibles Präzisionsmessgerät. Er dient ausschliesslich zur Erfassung von Dehnungen auf Druck und Zug, der Aufbereitung und Bereitstellung der Messwerte als CANopen Signale für das Folgegerät. Der Dehnungssensor darf ausschliesslich zu diesem Zweck verwendet werden.
- Der einwandfreie und sichere Betrieb setzt sachgemässen Transport, Lagerung, Montage sowie sorgfältige Bedienung und Wartung voraus.
- Einbau und Montage des Dehnungssensors darf ausschliesslich durch eine Fachkraft erfolgen.
- Vor der Inbetriebnahme der Anlage alle elektrischen Verbindungen überprüfen.
- Bei der Inbetriebnahme der Sensoren auf die jeweils geltenden Sicherheits- und Unfallverhütungsvorschriften achten.
- Es müssen hard- und softwareseitige Sicherheitsvorkehrungen getroffen werden, damit ein Leitungsbruch nicht zu undefinierten Zuständen der Automatisierungseinrichtung führt.
- Bei Anlagen, die aufgrund einer Fehlfunktion grosse Sachschäden oder sogar Personenschäden verursachen können, müssen Sicherheitsvorkehrungen getroffen werden, die im Fehlerfall einen sicheren Betriebszustand gewährleisten. Dies kann z. B. durch Grenzwertschalter, mechanische Verriegelungen usw. erfolgen.
- Der Dehnungssensor darf nicht ausserhalb der Spezifikationen betrieben werden (siehe Datenblatt).
- Keine mechanischen oder elektrischen Veränderungen am Sensor vornehmen.
- Trotz robuster Bauweise sollten die Dehnungssensoren keinen harten Stössen ausgesetzt werden.
- Statische und dynamische Überdehnungen, die 200% des Nennbereiches überschreiten, müssen vermieden werden.

1.2 Montage und Inbetriebnahme

- Für Informationen zur Montage und den Anschluss ans Messsystem wird auf die Montage- und Betriebsanleitung verwiesen, welche dem Sensor beigelegt ist.
- Verdrahtungsarbeiten immer im spannungslosen Zustand vornehmen.
- Elektrischer Anschluss darf nicht unter Spannung aufgesteckt oder abgenommen werden.
- Die gesamte Anlage EMV gerecht installieren. Einbauumgebung und Verkabelung beeinflussen die EMV des Dehnungssensors. Gerät und Zuleitung getrennt und in grossem Abstand zu Leitungen mit hohen Störpegeln verlegen.
- Den Dehnungssensor an Schutzerde anschliessen und geschirmte Kabel verwenden. Schirmgeflecht des Kabels muss mit der Kabelverschraubung verbunden sein.

2 Projektierung

2.1 Maximaler Systemausbau

Um einen lauffähigen Bus aufzubauen, muss mindestens ein Master (oder übergeordnetes System) auf dem Bus vorhanden sein. Dieser Master kann eine SPS-Steuerung oder ein PC mit entsprechender CAN-Karte sein. Jeder CANopen Dehnungssensor stellt einen aktiven CAN-Knoten dar.

Ein Busstrang mit jeweils einem Master des CAN-Netzwerkes kann aus maximal 127 Teilnehmern bestehen. Jeder Teilnehmer erhält eine eigene Adresse.

Werkseinstellungen dieses Sensors entnehmen sie dem Kapitel 6.2 [Parameter Handling \(save, load default\)](#).

Halten Sie unbedingt die zulässigen Bus- und Stichleitungslängen nach Tabelle 1 ein

Die maximal zulässige Gesamt-Leitungslänge und Gesamt-Stichleitungslänge

- ist abhängig von der Baudrate und
- kann in mehrere Segmente bzw. Einzelstichleitungen aufgeteilt werden.

Tabelle 1:

Baudrate [kBit/s]	10	20	50	100	125	250	500	800	1000
Gesamt-Buslänge	5000 m	3000 m	1000 m	500 m	400 m	200 m	75 m	30 m	25 m
Gesamt-Stichleitungslänge	1360 m	875 m	350 m	175 m	140 m	70 m	35 m	20 m	17 m
Einzel-Stichleitungslänge	270 m	175 m	70 m	35 m	28 m	14 m	7 m	4 m	3 m

Maximale Gesamt-Busleitungslänge (mit 120Ohm Abschlusswiderstand) und maximale Stichleitungslänge (ohne Abschlusswiderstand) in Abhängigkeit von der Baudrate.

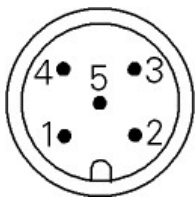
3 Anschlüsse

3.1 Elektrischer Anschluss

Schliessen Sie den Dehnungssensor gemäss untenstehender Skizze an. Achten Sie auf die richtige Polung. Verwenden Sie abgeschirmte Kabel.

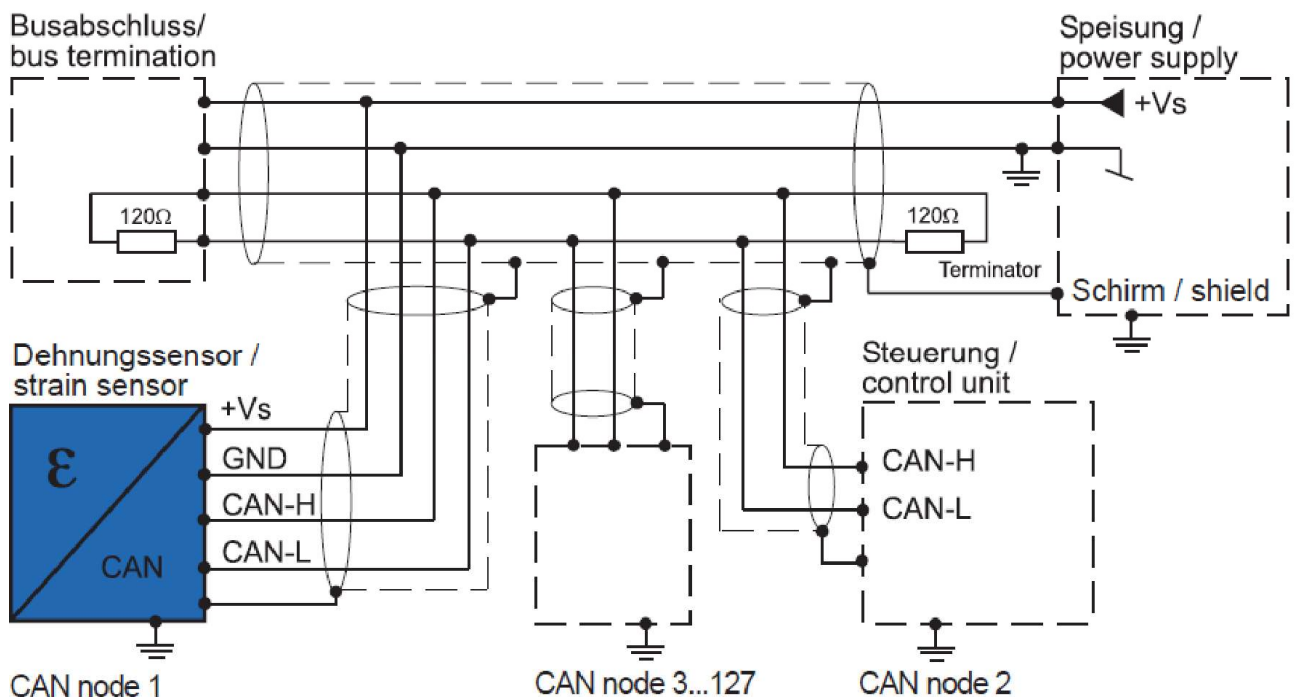
Das Sensorgehäuse und der Kabelschirm müssen geerdet werden. Vermeiden Sie Potentialdifferenzen zwischen Anlagenteilen und Messketten.

3.1.1 Anschlussbelegung



1 n.c.
 2 +Vs
 3 GND
 4 CANH
 5 CANL
 Gehäuse Schirm

3.1.2 Anschluss-Skizze



Um die PELV Anforderungen zu erfüllen gemäss EN 60204-1 §6.4.1, empfehlen wir 0V (GND) an einem Punkt im System mit Schutzerde zu verbinden.

3.2 Potentialverhältnisse

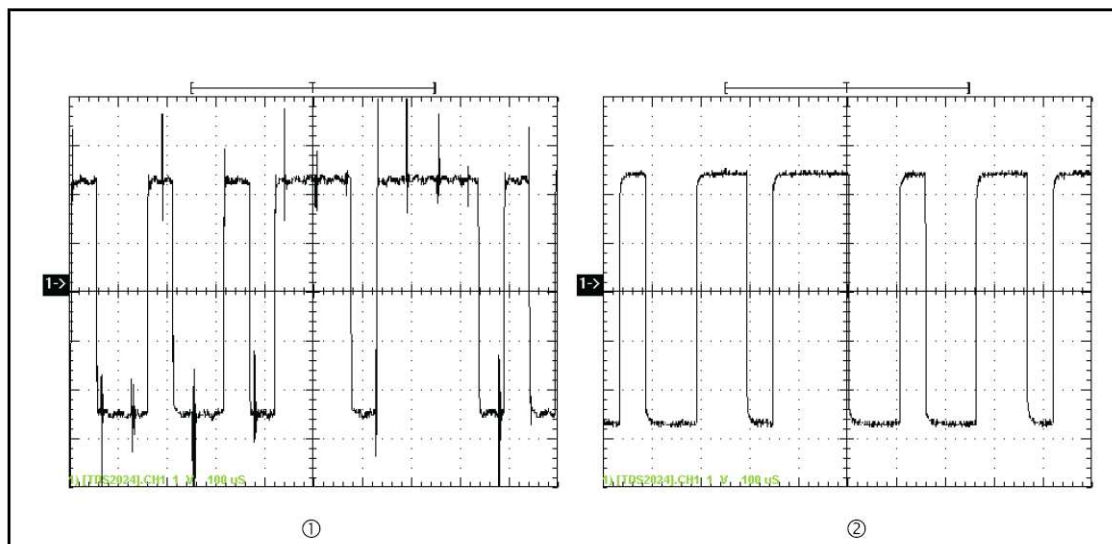
Die Potentialverhältnisse eines CANbus-Systems mit einem CANopen Dehnungssensor sind durch die folgenden Merkmale gekennzeichnet:

- Der CAN-Bus Anschluss ist nicht potentialgetrennt vom Versorgungsanschluss
- Die einzelnen CANopen Dehnungssensoren sind nicht galvanisch von der Versorgungsspannung getrennt
- Jeder CANopen Dehnungssensor kann separat versorgt werden

Vermeiden Sie Potentialdifferenzen, indem Sie

- jeden CAN-Teilnehmer auf dem kürzesten, möglichst niederohmigen Weg mit dem gleichen Erd-Bezugspotential (PE) der Maschine/Anlage verbinden.
- eine Potentialausgleichsleitung zwischen den Kommunikationsteilnehmern verwenden.
- den Erdbezug der Maschine/Anlage zur Gesamterde niederohmig ausführen.

Erkennen von EMV-Störungen im Signal-Oszillogramm



Oszillogramme der CAN-Signale (1) mit und (2) ohne Störspannung (Messpunkte: CAN_HIGH zu CAN_LOW).

Zur Quantifizierung von Störungen sind Messungen mit einem CAN-Analyser erforderlich. Hiermit können wichtige Bus-Parameter wie beispielsweise die Buslast oder die Anzahl von Errorframes ermittelt und weitergehende Analysen durchgeführt werden.

3.3 EMV-gerechte Verdrahtung

EMV (Elektromagnetische Verträglichkeit) ist die Fähigkeit eines Gerätes in einer gegebenen elektromagnetischen Umgebung fehlerfrei zu arbeiten, ohne selbst die Umgebung in einer nicht zulässigen Weise zu beeinflussen.

Alle CANopen Dehnungssensoren werden diesen Anforderungen gerecht, da sämtliche Sensoren auf die Einhaltung der gesetzlich vorgeschriebenen Grenzwerte (Industrienorm) getestet werden.

3.3.1 Erdung inaktiver Metallteile

Alle inaktiven Metallteile müssen grossflächig und impedanzarm verbunden werden (Erdung). Diese Massnahme stellt sicher, dass ein einheitliches Bezugspotential für alle Elemente des Systems gewährleistet ist.

Die Erdung der CANopen Dehnungssensoren erfolgt über die zwei oder vier Befestigungsschrauben.

3.3.2 Schirmung von Leitungen

Die Schirmleitung sollte, wenn möglich, beidseitig mit EMV gerechtem Schirmanschluss geerdet werden.

3.4 Spezifikation der CAN-Leitungen

Das Kabel, welches Sie für die Verbindung der Busteilnehmer am CAN-Bus verwenden, muss der Norm ISO 11898 entsprechen. Die Leitungen müssen demnach folgende elektrischen Eigenschaften aufweisen:

Spezifikation der CAN-Leitungen.

Gesamtlänge Bussystem	< 300 m	< 1000 m
Kabeltyp	LIYCY 2 x 2 x 0,5 mm ² (paarverseilt mit Abschirmung)	CYPIMF 2 x 2 x 0,5 mm ² (paarverseilt mit Abschirmung)
Leitungswiderstand	≤ 40 Ohm/km	≤ 40 Ohm/km
Kapazitätsbelag	≤ 130 nF/km	≤ 60 nF/km
Anschluss	Paar 1 (weiss / braun): CAN-GND und +Vs Paar 2 (grün / gelb): CAN-HIGH und CAN-LOW	

- Nur Leitungen verwenden, die ein zusätzliches Aderpaar für CAN-GND haben.
- Nur mit korrekt angeschlossenem CAN-GND ist ein störungsfreier Busbetrieb möglich.

Schliessen Sie die Bus-Abschlusswiderstände an

Am physikalischen Anfang und am physikalischen Ende des Bussystems muss jeweils ein Abschlusswiderstand von 120 Ohm angeschlossen sein.

4 CANopen

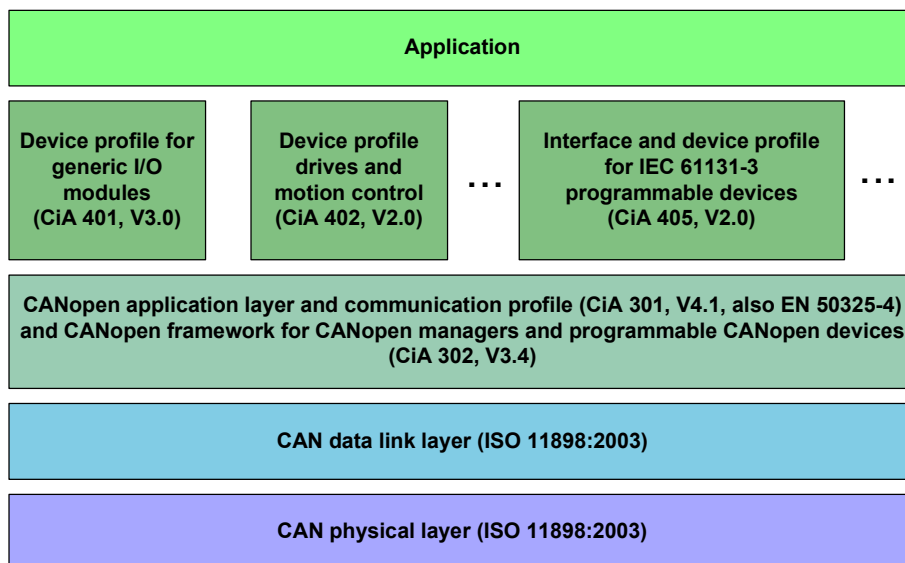
4.1 Einleitung

Das CANopen Protokoll ist ein offenes und standardisiertes ISO/OSI-Schicht-7 Protokoll basierend auf dem Controller Area Network (CAN) Application Layer. CANopen ist entwickelt, international genormt und gewartet von der Nutzerorganisation CAN in Automation (CiA).

CANopen weist folgende Leistungsmerkmale aus:

- Übertragung von zeitkritischen Prozessdaten nach dem Producer-Consumer Prinzip. Nachrichten können von allen Busteilnehmern empfangen werden. Sie werden nicht mit der Zieladresse versehen, sondern weisen einen Identifier auf.
- Standardisierte Gerätebeschreibung (Daten, Parameter, Funktionen, Programme) im sogenannten "Objektverzeichnis". Zugriff auf alle "Objekte" eines Gerätes mit standardisiertem Übertragungsprotokoll nach dem Client-Server-Prinzip.
- Standardisierte Knoten-Überwachung (Node-Guarding & Heartbeat), Fehler Signalisierung (Emergency-Nachrichten) und Netzwerk-Koordination (Netzwerk-Management).
- Standardisiertes System für synchrone Operationen (Synchronisations-Nachricht).
- Standardisierte Funktion zur Konfiguration der Baudrate und der Geräte-ID über den Bus mittels LSS.

CANopen besteht aus einem Kommunikationsprofil (regelt die Kommunikation) sowie aus verschiedenen Geräteprofilen für die typischen Anwendungsprofile.



Das CANopen Kommunikationsprofil (CiA DS-301) regelt das „Wie“ der Kommunikation. Hierbei wird zwischen den Echtzeitdaten und den Parameter Daten unterschieden.

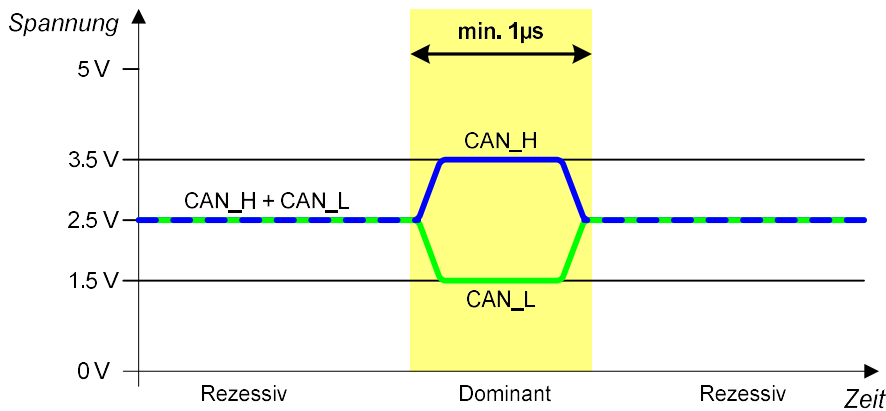
Für die Sensoren von Baumer Process Instrumentation wird das Geräteprofil DS404 für Mess- und Regelgeräte angewendet.

4.2 Signale, Aufbau und Bus Topologie

4.2.1 Bus-Signale

Eine gute elektrische Störsicherheit wird unter anderem dadurch erreicht, dass ein Bit auf zwei Leitungen differenziell übertragen wird. Die Leitung CAN-High und CAN-Low enthalten das invertierte und das nicht invertierte serielle Datensignal.

Der Zustand mit zwei unterschiedlichen Pegeln auf CAN-H und CAN-L wird als der dominante Zustand, der Zustand mit zwei gleichen Pegeln als rezessiv bezeichnet.



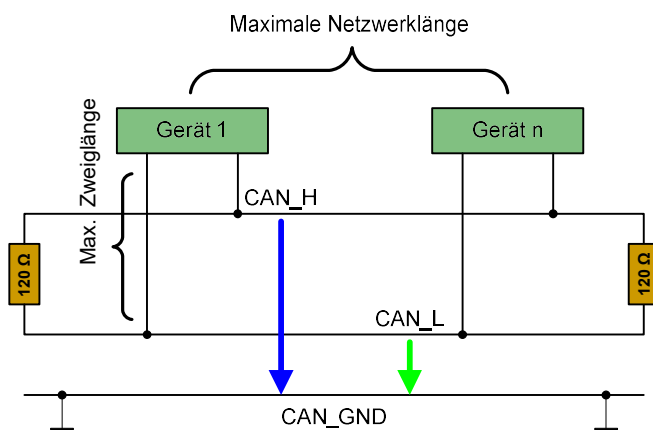
Der dominante Zustand entspricht per CAN Definition einer logischen Null (Bustreiber haben einen „Open-Kollektor“ Ausgang).

Legt ein Knoten eine logische Null auf den Bus, überschreibt er den Zustand einer logischen Eins eines anderen Knotens.

4.2.2 Netzwerk-Topologie

Die zugrunde liegende CAN-Architektur definiert die physikalischen Strukturen des CANopen-Netzwerks. Diese beruht auf einer Bus- (Linien-)Topologie. Zur Vermeidung von Reflexionen der Signale müssen die Enden des Netzes mit einem Abschlusswiderstand (120Ohm) geschlossen werden.

Zusätzlich sind auch die maximal zulässigen Zweig-Leitungslängen (max. stub length) für den Anschluss der einzelnen Netzwerkknoten zu beachten.



Die zulässigen Bitraten/Leitungslängen für ein CANopen-Netzwerk (CiA 301):

Data rate bus length	Sample point (TQ)	Max. stub length	Accumu- lated stub length
1 Mbit/s 25 m	87,5% (125 ns)	1,5 m	7,5 m
800 kbit/s 50 m	87,5% (125 ns)	2,5 m	12,5 m
500 kbit/s 100 m	87,5% (125 ns)	5,5 m	27,5 m
250 kbit/s 250 m	87,5% (250 ns)	11 m	55 m
125 kbit/s 500 m	87,5% (500 ns)	22 m	110 m
50 kbit/s 1000 m	87,5% (1,25 µs)	55 m	275 m
20 kbit/s 2500 m	87,5% (3,125 µs)	137,5 m	687,5 m
10 kbit/s 5000 m	87,5% (6,25 µs)	275 m	1375 m

Zwei Bedingungen müssen für ein fehlerfrei funktionierendes CANopen Netzwerk gegeben sein:

- Alle Knoten müssen die gleiche Bitrate aufweisen
- Keine Knoten-ID darf zweimal vorhanden sein

Der Systemintegrator ist für das Einhalten der gleichen Bitrate und der unterschiedlichen Knoten-ID verantwortlich.

Sensoren werden von Baumer standardmässig mit 125kBaud und ID =1 ausgeliefert und können über die Objekte 2100H und 2101H oder mit Hilfe von „LSS-Service“ (CiA 305) konfiguriert werden.

4.2.3 Aufbau einer CAN Nachricht

Eine CAN-Botschaft, auch Frame genannt, besteht aus folgenden 7 Kennfeldern:

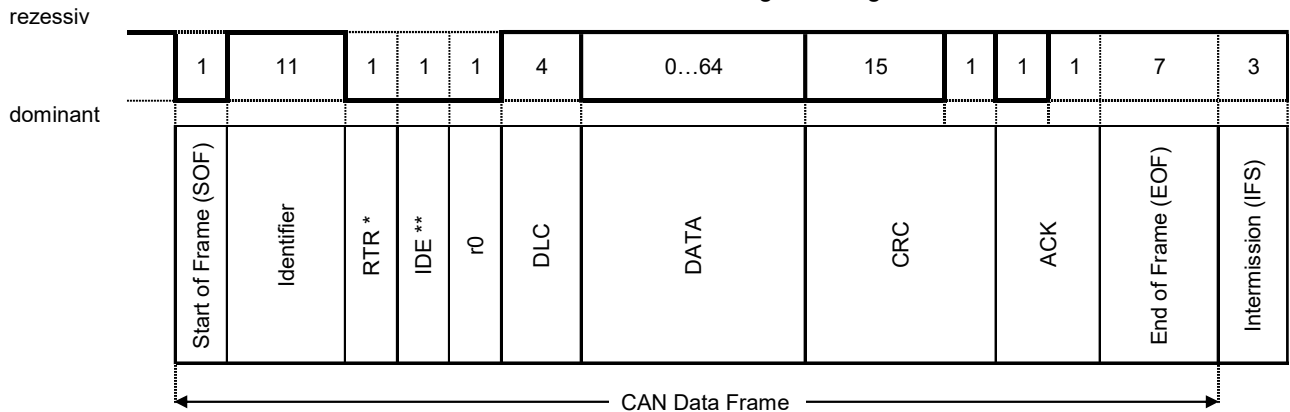
- Start-Condition
- Message Identifier
- Steuerbits
- Daten (0-8 Bytes)
- CRC-Prüfbits
- Acknowledge-Bit
- Stop-Condition

Man unterscheidet die Frames nach der Länge des Identifiers:

- Standard Frame (11 Bit Identifier)
- Extended Frame (29 Bit Identifier)

Bei Baumer PI werden nur Standard Frames unterstützt.

Den Aufbau des Standard-Frames nach Standard CAN 2.0A zeigt das folgende Bild:



* RTR = 0 => Data Frame
RTR = 1 => Remote Frame

** IDE = 0 => 11Bit Identifier
IDE = 1 => 29 Bit Identifier

- Start: dominant (logisch 0), dient der Synchronisation
- Identifier: Information für den Empfänger und Prioritätsinformation für die Busarbitrierung
- RTR: rezessiv, unterscheidet zwischen Daten- (dominant) und Datenanforderungstelegramm (rezessiv)
- IDE: Identifier Extension
- r0: reserviert
- DLC: enthält die Längeninformation der nachfolgenden Daten
- DATA: enthält die Daten des Telegramms
- CRC: kennzeichnet den Fehlercode für die vorangegangenen Informationen. Die CRC Prüfsumme wird zur Fehlererkennung verwendet
- ACK: enthält Rückmeldung von anderen Teilnehmern bei korrektem Empfang der Nachricht
- EOF: kennzeichnet das Ende des Datentelegramms (7 rezessive Bits)
- IFS: kennzeichnet den Zeitraum für das Übertragen einer korrekt empfangenen Nachricht

4.2.4 Bitweise Busarbitrierung

Im Rahmen der Arbitrierungsphase wird ermittelt, welche von den gleichzeitig arbitrierenden Nachrichten die höchste Priorität aufweist. Höchste Priorität hat die Nachricht mit dem niedrigsten Wert des Nachrichten-Identifiers. Die Arbitrierungsphase umfasst das Senden des Nachrichten-Identifiers sowie des sog. RTR-Bits ("Remote-Transmission-Request"-Bit). Erkennt ein Netzknoten einen dominanten Buspegel (logisch 0), obwohl er selbst einen rezessiven Pegel (rezessives Bit) aufgeschaltet hat, so bricht er den Sendevorgang sofort ab und geht in den Empfangszustand, da in diesem Falle offensichtlich gleichzeitig eine Nachricht höherer Priorität gesendet wird.

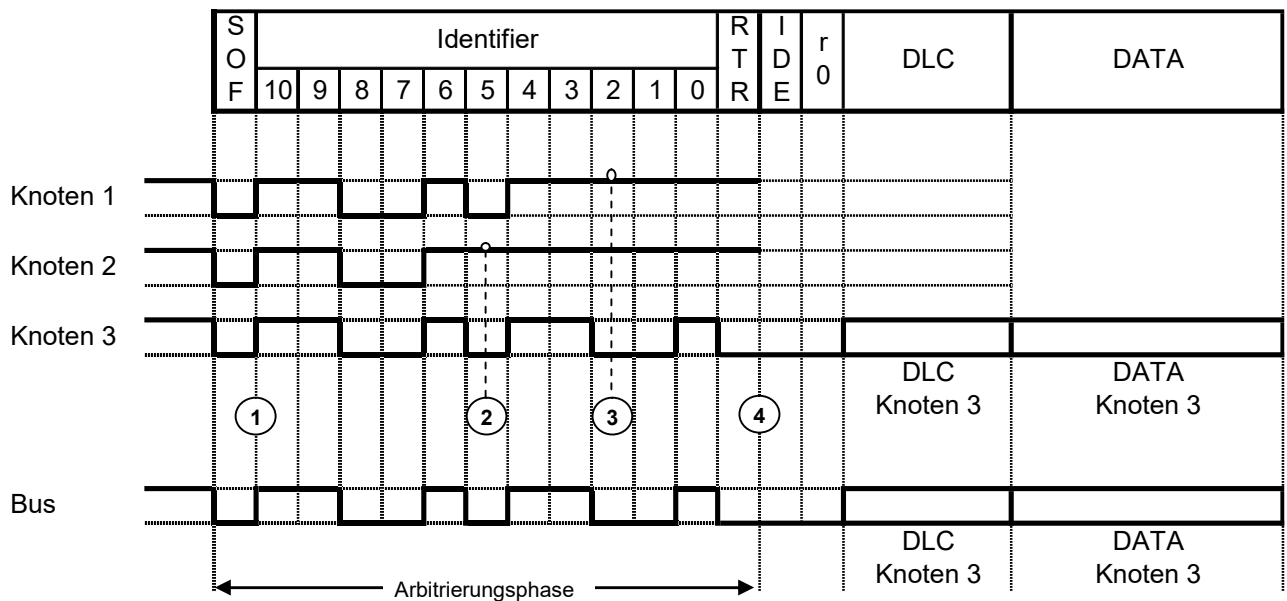


Bild 1: Prinzip der bitweisen Busarbitrierung - Knoten 1, 2 und 3 beginnen gleichzeitig einen Arbitrierungsvorgang. Zum Zeitpunkt 2 stellt der Knoten 2 fest, dass der Bus nicht den von ihm gesendeten rezessiven Pegel hat und beendet seinen Arbitrierungsvorgang. Zum Zeitpunkt 3 gibt Knoten 1 auf. Zum Zeitpunkt 4 (Ende des Arbitrierungsprozesses) sendet Knoten 3 seine Daten.

4.2.5 Prioritätsorientierte Nachrichtenübertragung

Das zuvor beschriebene Arbitrierungsverfahren garantiert zu jeder Zeit, dass die jeweils höchstpriorie Nachricht gesendet wird sobald der Bus frei ist. Die Priorität der Nachricht ist hierbei über den Wert des Nachrichten-Identifiers bestimmt. Je kleiner dieser Wert ist, desto höher ist die Priorität der Nachricht. Das Prinzip der prioritätsorientierten Nachrichten ermöglicht eine sehr effiziente Nutzung der für die Datenübertragung zur Verfügung stehenden Bandbreite. Hierbei ist es möglich, dass niederpriorie Nachrichten den Bus mit 100% belegen, ohne die Übertragung höherpriorer Nachrichten maßgeblich zu verzögern. Für die höchstpriorie Nachricht resultiert bei einer Übertragungsrage von 1 Mbit/s eine maximale Latenzzeit von ca. 130 µs.

4.2.6 Identifier-Verteilung

Standardmässig werden bei der Kommunikation über CANopen Nachrichten-Identifier mit 11 Bit Länge verwendet. Somit steht der Bereich von 0 bis 7FF_H zur Verfügung. Die Identifier-Verteilung ist so ausgelegt, dass in einem CANopen- Netzwerk maximal 128 Geräte vorhanden sind: ein NMT-Master und bis zu 127 NMT-Slaves. Vordefinierte Identifier Zuweisung:

Kommunikationsobjekt	COB-ID(s) hex	Slave-Knoten
NMT node control	000	nur empfangen
Sync	080	nur empfangen
Emergency	080 + NodeID	senden
TimeStamp	100	nur empfangen
PDO	180 + NodeID	1. PDO senden
	200 + NodeID	1. PDO empfangen
	280 + NodeID	2. PDO senden
	300 + NodeID	2. PDO empfangen
	380 + NodeID	3. PDO senden
	400 + NodeID	3. PDO empfangen
SDO	480 + NodeID	4. PDO senden
	500 + NodeID	4. PDO empfangen
SDO	580 + NodeID	senden
	600 + NodeID	empfangen
NMT node monitoring (node guarding/heartbeat)	700 + NodeID	senden
LSS	7E4	senden
	7E5	empfangen

Der "Master" im Netzwerk besitzt die Fähigkeit, die Betriebsart der "Slaves" zu ändern. Er hat also die Kontrolle über das CANopen-Netzwerk. Vielfach wird der „Master“ daher auch als CANopen-Network-Manager bezeichnet. Typischerweise wird ein CANopen-Master durch eine SPS oder einen PC realisiert. Die CANopen-Slaves können die Adressen 1 bis 127 belegen. Durch die Geräteadresse ergibt sich dann automatisch eine Anzahl von Identifiern, welche dieses Gerät dann belegt.

4.3 Objekte

Das Objektverzeichnis beschreibt die komplette Funktionalität der CANopen-Geräte und ist in Tabellenform organisiert. Im Objektverzeichnis sind nicht nur die standardisierten Datentypen und Objekte des CANopen-Kommunikationsprofils sowie der Geräteprofile enthalten, sondern gegebenenfalls auch hersteller-spezifische Objekte und Datentypen. Die Adressierung der Einträge erfolgt mit Hilfe eines 16-Bit-Indizes (Reihenadresse der Tabelle, maximal 65536 Einträge) und eines 8-Bit-Subindizes (Spaltenadresse der Tabelle, maximal 256 Einträge). Somit lassen sich zusammengehörige Objekte leicht gruppieren. Die Struktur dieses CANopen Objektverzeichnisses ist in der folgenden Tabelle dargestellt.

Übersicht über das gesamte Objektverzeichnis:

Indexbereich	Beschreibung
0000h	Reserviert
0001h bis 025Fh	Datentypen
0260h bis 0FFFh	Reserviert
1000h bis 1FFFh	Kommunikationsprofilbereich
2000h bis 5FFFh	Herstellerspezifischer Profilbereich
6000h bis 9FFFh	Standardisierter Profilbereich
A000h bis AFFFh	Netzwerkvariablen
B000h bis BFFFh	Systemvariablen
C000h bis FFFFh	Reserviert

Ausschnitt aus dem Objekt-Bereich für die Kommunikation (1000H ... 1FFFH)

Indexbereich	Beschreibung
1000h bis 1029h	Generelle Kommunikationsobjekte
1200h bis 12FFh	SDO Parameterobjekte
1300h bis 13FFh	CANopen Sicherheitsobjekte
1400h bis 1BFFh	PDO Parameterobjekte
1F00h bis 1F11h	SDO Managerobjekte
1F20h bis 1F27h	Konfigurationsmanagerobjekte
1F50h bis 1F54h	Programmsteuerungsobjekte
1F80h bis 1F89h	NMT Masterobjekte

4.4 Kommunikations-Mechanismen

Es wird hauptsächlich zwischen 2 verschiedenen Datenübertragungsarten unterschieden. Die „Process Data Objects“ (PDO) dienen zur Übertragung der Echtzeitdaten/Prozessdaten und die „Service Data Objects“ (SDO) ermöglichen den Zugriff auf das Objektverzeichnis welches sämtliche Einstellungen des Geräts beinhaltet.

Zusätzlich zu den Standard-Übertragungs Mechanismen gibt es noch weitere Kommunikations-Mechanismen. Dabei handelt es sich um „Network Management“ (NMT), „Emergency“ (EMGY), „Node-Guarding“ und „Heartbeat“.

4.4.1 Process Data Objects (PDO)

Die Hauptaufgabe eines CANopen-Systems ist der Austausch von Prozessdaten.

Für die Übertragung von Prozessdaten wird auf Protokoll-Overhead verzichtet und die Übertragung erfolgt nach dem sogenannten "Producer-Consumer Prinzip". Dies bedeutet, dass eine Nachricht von einem Knoten (der "Producer") versendet von allen anderen Knoten (die "Consumer") empfangen werden kann. Dieses Prinzip wird auch als "Rundfunk" bezeichnet und stellt ein sehr effiziente Prinzip der Datenübertragung dar. PDO Nachrichten werden nicht bestätigt um die Buslast vor allem bei zeitkritischen Anwendungen möglichst tief zu halten. Dieser Dienst ist somit nicht ein Frage-Antwort Mechanismus.

Die Übertragung von PDOs ist nur im Zustand "Operational" möglich und die Übertragungspakete haben keine feste Datenlänge. Die Datenlänge der PDO kann zwischen einem und acht Byte lang sein.

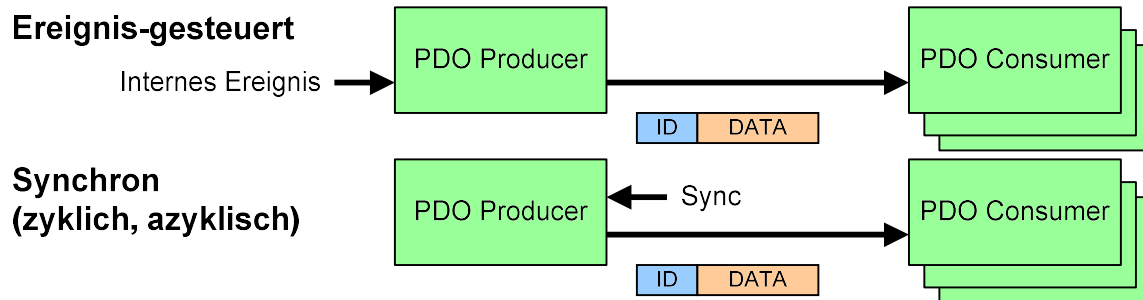
Bezüglich der Zusammensetzung der Datenpakete muss sowohl der Sender als auch der Empfänger wissen wie der Inhalt zu interpretieren respektive zusammenzustellen ist. Der Absender des PDOs ist nur durch die COB-ID erkennbar.

Das sogenannte "PDO-Mapping" beschreibt die einzelnen Prozessvariablen die im Datenfeld eines PDO übertragen werden, wie sie angeordnet sind und welche Daten Typ und Länge sie haben. Der Inhalt und die Bedeutung der übertragenen Daten eines PDOs ist in einer PDO-Mapping Liste definiert; sowohl auf der Sende- wie auch auf der Empfangsseite.

Die Übertragung der Prozessdaten kann durch verschiedene Ereignisse ausgelöst werden:

- Ereignisgesteuert
Die Aussendung der PDO's wird durch ein internes Ereignis des Knotens ausgelöst. Dies kann durch einen Zeitgeber im Gerät, durch ein Über- oder Unterschreiten einer Limite oder durch weitere interne Ereignisse
- Synchronisiert
Ein Busteilnehmer (meistens der Master) sendet Synchronisationstelegramme auf den Bus. Bei der synchronen Übertragung werden die PDOs durch das empfangene Sync-Telegramm ausgelöst. Damit ist es möglich eine Momentaufnahme (Prozesswerte zur gleichen Zeit) des Systems zu erhalten.
- Anforderungsgesteuert
In diesem Fall fordert ein Busteilnehmer die Prozessdaten über ein Remote-Transmission-Request (RTR) an. Dieser Mechanismus wird nicht weiter empfohlen und bei den Dehnungssensoren nicht umgesetzt.

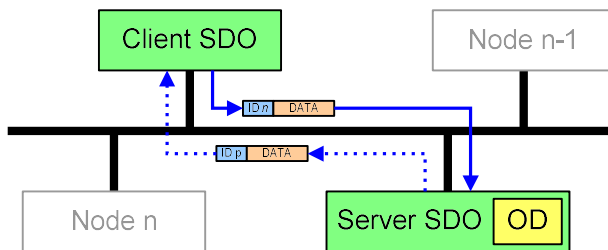
Aufbau einer PDO Nachricht:



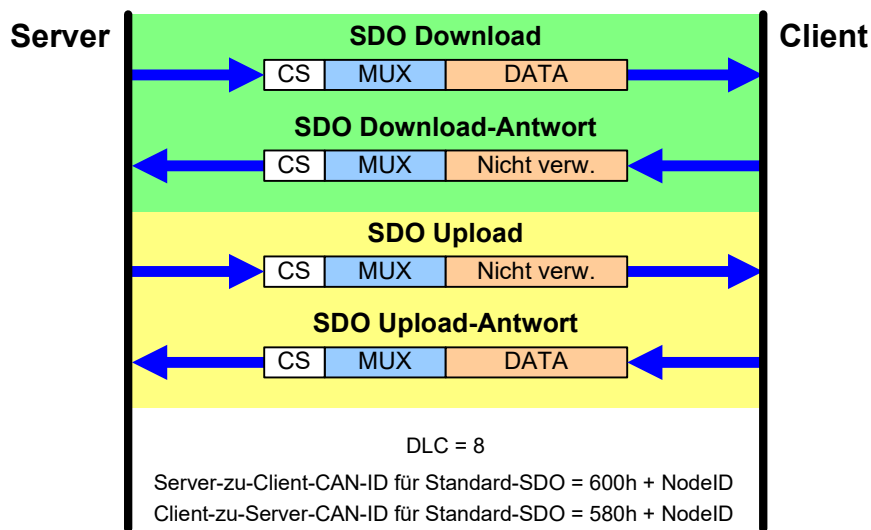
4.4.2 Service Data Objects (SDO)

Spezifische Kommunikation Objekte, sogenannte "Service Data Objects" (SDO) werden für den direkten Zugriff auf CANopen Geräte verwendet. Mit diesen SDOs können in Objektverzeichnis Einträge gelesen und geschrieben werden. Die Kommunikation erfolgt stets als logische 1:1-Verbindung (Peer-to-Peer) zwischen zwei Knoten (meistens ist der Master der Konfigurierer und ein normaler Busteilnehmer der zu Konfigurierende).

Aufgrund der direkten Verbindung wird auf jede Anfrage auch eine Antwort erwartet. Dies ist zu vergleichen mit einer Funkverbindung. Jede Anfrage muss beantwortet werden, auch wenn das Gerät nicht in der Lage ist die Anfrage auszuführen oder zu beantworten oder wenn die Anfrage selbst bereits fehlerhaft ist. Eine solche negative Antwort heißt "Abbruch". In der Abbruchmeldung wird neben dem 4-Byte lange Fehlercode (Ursache für den Abbruch) auch die Objektadresse auf welche zugegriffen werden sollte angegeben.



Aufbau einer SDO Nachricht



CS = command specifier
 MUX = 16-bit Index und 8-Bit Subindex

4.4.3 Network Management (NMT)

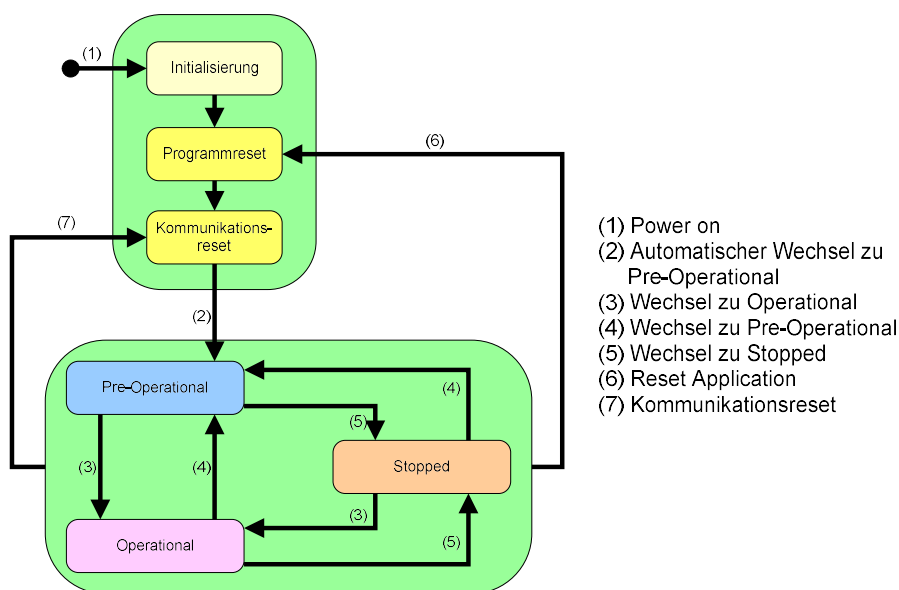
In einem CANopen-Netzwerk gibt es einen NMT-Master und 1 ... 127 NMT-Slaves.

Der NMT-Master hat die komplette Kontrolle über alle Geräte und kann deren Zustand verändern.

Die NMT-Messages haben in einem CANopen-Netzwerk die höchste Priorität und besitzen die ID=0. Ein NMT-Befehl hat jeweils nur 2 Daten-Bytes. Der NMT-Master kann den Zustand eines einzelnen Slaves (z.B. ID=2) oder das ganze Netzwerk (ID=0) steuern.

Die Zustände in einem CANopen-Netzwerk werden meistens in einem Zustandsdiagramm dargestellt. Folgende Zustände sind in einem CANopen-Netzwerk möglich:

- Initialization
- Pre-Operational
- Operational
- Stopped



Initialisierung

Nach einem NMT-Reset oder einem Power-On ist der Knoten im „Initialization“ Zustand. In diesem Zustand wird die Geräte-Applikation und die Kommunikation initialisiert. Nach Abschluss der Initialisierung überträgt der Knoten eine „Boot-up Nachricht“ und schaltet automatisch in den Pre-Operational Zustand.

Pre-Operational

In diesem Zustand kann mit dem Knoten über SDO kommuniziert werden. Es können keine PDO Meldungen gesendet werden. Dieser Zustand wird in erster Linie für die Konfiguration der CANopen Geräte verwendet.

Operational

In diesem Zustand hat der Knoten die volle Betriebsbereitschaft und kann selbstständig Nachrichten übertragen.

Stopped

Bis auf Node-Guarding- und Heartbeat-Nachrichten können in diesem Zustand keine Nachrichten vom Knoten gesendet werden. Einzig die LSS-Konfiguration funktioniert in diesem Zustand.

Start-Remote Node => Wechsel in den Operational Mode

ID	DLC	Byte1	Byte2
0	2	01h	Node

Node = Moduladresse, 0 = alle Knoten

Stop-Remote Node => Wechsel in den Stopped Mode

ID	DLC	Byte1	Byte2
0	2	02h	Node

Node = Moduladresse, 0 = alle Knoten

Pre-Operational-Remote Node => Wechsel in den pre-Operational Mode

ID	DLC	Byte1	Byte2
0	2	80h	Node

Node = Moduladresse, 0 = alle Knoten

Reset Node => Software Reset des Knotens

ID	DLC	Byte1	Byte2
0	2	81h	Node

Node = Moduladresse, 0 = alle Knoten

4.4.4 Emergency (EMGY)

Emergency Botschaften signalisieren Fehler eines Knotens. Das Emergency-Telegramm, beinhaltet einen Code, der den Fehler eindeutig identifiziert (definiert in DS-301 und den Geräteprofilen).

Die Emergency-Botschaften werden von den CANopen-Geräten selbstständig ausgesendet.

Zusammenstellung der Emergency-Botschaft:

Fehlercode	Fehlerregister	Herstellerspezifisches Fehlerfeld
-------------------	-----------------------	--

Übersicht über die Fehler-Codes:

Fehlercode (hex)	Fehlerbeschreibung
00xx	Fehlerreset / kein Fehler
10xx	Genereller Fehler
2xxx	Strom
3xxx	Spannung
4xxx	Temperatur
50xx	Gerätehardware
6xxx	Gerätesoftware
70xx	Zusätzliche Module
8xxx	Monitoring
90xx	Externer Fehler
F0xx	Zusätzliche Funktionen
FFxx	Gerätespezifisch

Übersicht über das Error Register:

Bit	Fehlerursache
0	Genereller Fehler
1	Strom
2	Spannung
3	Temperatur
4	Kommunikationsfehler
5	Geräteprofilsspezifisch
6	Reserviert (immer 0)
7	Herstellerspezifisch

Gleichzeitig werden die Fehler-Codes auch in die Emergency-History (Objekt 1003h) geschrieben. Die COB-ID der Emergency-Botschaft kann im Objekt 1014h nachgeschaut werden.

4.4.5 Node-Guarding und Hearbeat

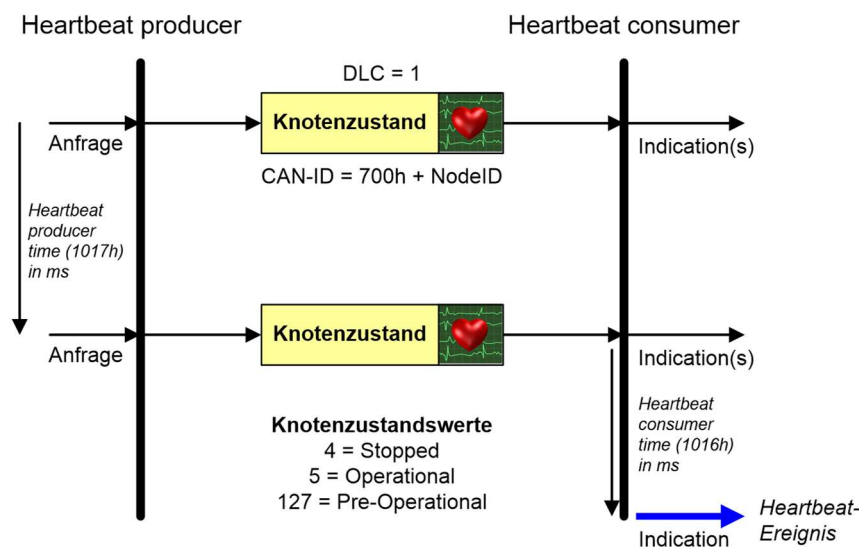
Um die Funktionsfähigkeit der Netzknoten festzustellen hat CANopen folgende Möglichkeiten:

- Automatische Übertragung eines "Heartbeat-Nachricht" von den Netzknoten ("Heartbeat"-Prinzip)
- Zyklische Abfrage des Knoten Zustands durch den "NMT-Master" ("Node-Guarding"-Prinzip)

Bei der Knotenüberwachung mit Heartbeat sendet jeder Knoten autark eine Botschaft in zyklischen Abständen aus. Diese Nachricht kann von jedem Teilnehmer im Netzwerk überwacht werden. Das Intervall zwischen zwei Heartbeat-Botschaften kann im Objekt 1017h eingestellt werden.

Beim Node-Guarding-Protokoll sendet der NMT-Master Botschaften an die CANopen-Slaves, die innerhalb einer definierten Zeit antworten. Ein Ausbleiben der Antwort kann nur durch den NMT-Master erkannt werden. Fällt der NMT-Master aus, so ist das gesamte Netzwerk lahmgelegt. Aus diesem Grund und der höhere Buslast (durch 2 CAN-Nachrichten pro Überwachungsintervall) wird das Node-Guarding fast komplett durch die Heartbeat-Überwachung abgelöst.

Die Überwachungs-Botschaft der Knoten enthält die COB-ID 700h + Node-ID des Senders. Das einzige übertragene Daten-Byte beinhaltet den Geräte-Zustand (Pre-Operational, Operational, Stopped) des Senders.



4.5 Weitere Begriffserklärung

4.5.1 Boot-Up Nachricht

Die Boot-up Nachricht ist das erste Lebenszeichen eines CANopen Geräts nach einem Power-up oder einem Reset. Diese Botschaft signalisiert, dass Knoten die Initialisierung abgeschlossen haben und in den Pre-Operational Status wechseln.

4.5.2 EDS

Das „Electronic Data Sheet“ (EDS) beschreibt die Funktionalität eines CANopen Gerätes in maschinen-lesbarer Form. Diese Dateien in einem standardisierten Textformat beschreiben sowohl alle unterstützten Objekte aus dem Objektverzeichnis des Gerätes, diverse Informationen über das Gerät und den Hersteller als auch physikalische Parameter wie zum Beispiel die unterstützten Baudraten.

Fast alle CANopen Steuerungen können EDS Dateien einlesen und erleichtern dem System-Integrator die Parametrisierung des Systems.

4.5.3 DCF

Das „Device Configuration File“ (DCF) hat das EDS File als Grundlage und enthält zusätzlich die Werte jedes Objekts.

Dieses File kann zur automatischen Konfiguration von CANopen Geräten verwendet werden.

4.5.4 LSS

Der Layer Setting Services (LSS) ist ein Dienst mit dem einem Gerät die ID und die Bitrate eingestellt werden kann.

Dafür sind die Identifier 7E4_H und 7E5_H reserviert. Der Dienst kann in einer 1:1 Verbindung vom Master zum Gerät oder über den Bus genutzt werden.

5 CANopen Protokoll

5.1 Allgemeines

Diese Bedienungsanleitung gibt den aktuellen Stand der implementierten Funktionen der Module wieder (beschrieben in den nachfolgenden Kapiteln).

Weiterführende Literatur erhältlich bei der Nutzerorganisation:

CAN in Automation (CiA)
Kontumazgarten 3
DE-90429 Nürnberg
headquarters@can-cia.org
www.can-cia.org

Zu diesem CANopen Dehnungssensor brauchen Sie keine weiteren Hilfsmittel um die Identifikation und die Baudrate umzustellen. Der Sensor kann auch nicht geöffnet werden. Die Kommunikationsparameter können über die Software definiert und gespeichert werden.

Die Angaben zu dem CANopen-Master entnehmen Sie bitte den Dokumentationen der jeweils eingesetzten Geräte.

5.1.1 Bootloader

Im Sensor ist ein Bootloader implementiert. Auf Wunsch kann damit die Firmware des Sensors beim Kunden im CAN-Netzwerk aktualisiert werden.

5.2 Network Management

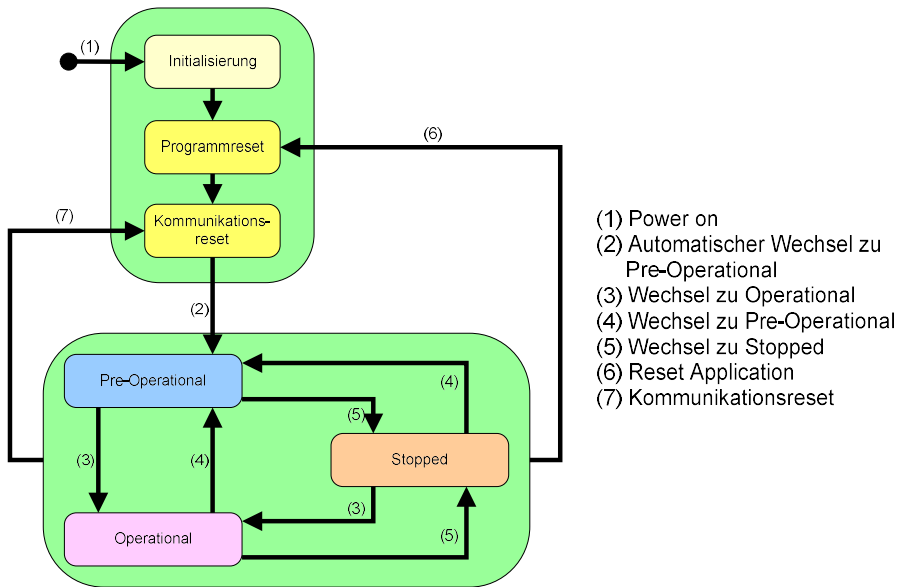
Nach dem Einschalten der Speisung am CANopen-Dehnungssensor meldet sich der Sensor durch das Senden der CAN-Message "Boot-up Message". Dies ist eine Mitteilung ohne Datenbytes mit dem COB-Identifizier $1792_D + \text{Modul-ID} (700_H + \text{ID})$.

5.2.1 Predefined Connection Set

COB-ID = Function Code (4 bit) + Modul-ID (7 bit)

Objekt	COB-ID (dezimal)	COB-ID (hex)
Network Management	0	0
Sync	128	80h
Emergency	129 – 255	81h – FFh
PDO1 (tx)	385 – 511	181h – 1FFh
PDO1 (rx)	512 – 640	201h – 27Fh
PDO2 (tx)	641 – 767	281h – 2FFh
PDO3 (tx)	897 – 1023	381h – 3FFh
SDO (tx)	1409 – 1535	581h – 5FFh
SDO (rx)	1537 – 1663	601h – 67Fh
Heart beat	1793 – 1919	701h – 77Fh

5.2.2 Startprozedere



- (1) Power on
- (2) Automatischer Wechsel zu Pre-Operational
- (3) Wechsel zu Operational
- (4) Wechsel zu Pre-Operational
- (5) Wechsel zu Stopped
- (6) Reset Application
- (7) Kommunikationsreset

Initialisierung

Dies ist der Zustand, den ein Knoten nach dem Einschalten durchläuft. Innerhalb dieser Phase erfolgt eine Initialisierung der Geräteapplikation sowie der Gerätekommunikation. Anschliessend geht der Knoten selbstständig in den Zustand Pre-Operational.

Pre-Operational

In diesem Zustand wartet der Knoten auf das Freigeben des Operational-Modus. Die Kommunikationsmöglichkeiten sind in der untenstehenden Tabelle dargestellt.

Operational

In diesem Zustand hat der CANopen-Knoten die volle Betriebsbereitschaft und kann selbständig Nachrichten übertragen (PDO's, Emergency).

Kommunikationsmöglichkeiten während den verschiedenen Modi:

	Initialisierung	Pre-Operational Modus	Operational Modus	Stop Modus
PDO			X	
SDO		X	X	
Sync Indexe			X	
Emergency Indexe		X	X	
Boot – Up Indexe	X			
Network management		X	X	X
Heart beat		X	X	X
LSS				X

5.2.3 Start Node

ID	DLC	Byte1	Byte2
0	2	01h	Node

Node = Moduladresse, 0 = alle Knoten

Der Befehl "Start Node" setzt den Knoten wieder in den Operational Modus.

5.2.4 Stop Node

ID	DLC	Byte1	Byte2
0	2	02h	Node

Node = Moduladresse, 0 = alle Knoten

Der Befehl "Stop Node" setzt den Knoten wieder in den Pre-Operational Modus.

5.2.5 Pre-Operational Node

ID	DLC	Byte1	Byte2
0	2	80h	Node

Node = Moduladresse, 0 = alle Knoten

5.2.6 Reset Node

ID	DLC	Byte1	Byte2
0	2	81h	Node

Node = Moduladresse, 0 = alle Knoten

Der Befehl "Reset Node" entspricht einem Power On Reset.

5.3 Übersicht der unterstützten Objekte

Folgende Tabelle zeigt eine Zusammenstellung aller unterstützten SDO Objekten.

Objekt	Objektnummer in Hex
Sub-Index	Sub-Index in Hex
Name	Name des Objekts/Sub-Index
Format	U/I = Unsigned/Integer, Zahl = Anzahl Datenbit, ARR = Array, REC = Record
Zugriff	ro = read only, wo = write only, rw = read & write
Default	Defaultwerte beim ersten Initialisieren oder bei Load Default
PDO mapping	Kann Objekt/Sub-Index gemappt werden, TPDO = Sende PDO, RPDO = Empfangs PDO
Seite	Weitere Informationen zu den Objekten befindet sich auf den angegebenen Seiten

Objekt	Sub-Index	Name	Format	Zugriff	Default	PDO mapping	Seite
1000	00	Geräteprofil	U32	ro	00'02'01'94h	-	29
1001	00	Error Register	U8	ro	00h	TPDO	49
1002	00	Kalibrationsdatum	U32	ro	z.B. 10'07'14h (14 Juli 2010)	-	29
1003		Emergency History	ARR				49
	00	Anzahl Fehler	U8	rw	00h	-	
	01	Letzter Fehler	U32	ro	-	-	
	02-0F	Ältere Fehler	U32	ro	-	-	
1005	00	Sync ID	U8	ro	80h	-	48
1008	00	Gerätebezeichnung	U32	ro	„DSRT“	-	29
1009	00	Hardware Version	U32	ro	z.B. „3.03“	-	30
100A	00	Software Version	U32	ro	z.B. „2.08“	-	30
1010		Speichern	ARR				32
	00	Anzahl Subindex	U8	ro	01h	-	
	01	Alle Parameter speichern	U32	rw	„save“	-	
1011		Defaultwerte laden	ARR				33
	00	Anzahl Subindex	U8	ro	01h	-	
	01	Alle Default Daten laden	U32	rw	„load“	-	
1014	00	Emergency Message ID	U32	ro	00'00'00'81h	-	52
1017	00	Heartbeat [ms]	U16	rw	00'00h	-	53
1018		Geräte Identität	REC				30
	00	Anzahl Subindex	U8	ro	04h	-	
	01	Vendor ID	U32	ro	00'00'00'5Fh	-	
	02	Produktcode	U32	ro	z.B. 11038931d	-	
	03	Revisionsnummer	U32	ro	z.B. 00'03'02'08h	-	
	04	Seriennummer	U32	ro	z.B. 00000000d	-	
1400		Empfang PDO 1	REC				44
	00	Anzahl Subindex	U8	ro	02h	-	
	01	ID & Aktivierung RPDO	U32	rw	40'00'02'01h	-	
	02	Übertragungsart	U8	rw	FEh	-	
1600		PDO 1 Mapping Parameter	ARR				45
	00	Anzahl Subindex	U8	ro	01h	-	
	01	Mapping Objekt	U32	ro	20'03'00'20h	-	
1800		Sende PDO 1	REC				45
	00	Anzahl Subindex	U8	ro	05h	-	
	01	ID & Aktivierung TPDO1	U32	rw	40'00'01'81h	-	
	02	Übertragungsart	U8	rw	FFh	-	
	05	Event Time [ms]	U16	rw	03'E8h	-	
1801		Sende PDO 2	REC				47
	00	Anzahl Subindex	U8	ro	05h	-	
	01	ID & Aktivierung TPDO2	U32	rw	40'00'02'81h	-	
	02	Übertragungsart	U8	rw	02h	-	
	05	Event Time [ms]	U16	rw	03'E8h	-	
1802		Sende PDO 3	REC				47
	00	Anzahl Subindex	U8	ro	05h	-	
	01	ID & Aktivierung TPDO3	U32	rw	40'00'03'81h	-	
	02	Übertragungsart	U8	rw	FEh	-	
	05	Event Time [ms]	U16	rw	03'E8h	-	

Objekt	Sub-Index	Name	Format	Zugriff	Default	PDO mapping	Seite
1A00		PDO 1 Mapping Parameter	ARR				47
	00	Anzahl Subindex	U8	rw	01h	-	
	01	Erstes übertragene Objekt	U32	rw	71'30'01'10h	-	
	02-04	Weitere übertragene Objekte	U32	rw	*1	-	
1A01		PDO 2 Mapping Parameter	ARR				48
	00	Anzahl Subindex	U8	rw	01h	-	
	01	Erstes übertragene Objekt	U32	rw	71'30'01'10h	-	
	02-04	Weitere übertragene Objekte	U32	rw	*1	-	
1A02		PDO 3 Mapping Parameter	ARR				48
	00	Anzahl Subindex	U8	rw	01h	-	
	01	Erstes übertragene Objekt	U32	rw	20'04'00'10h	-	
	02-04	Weitere übertragene Objekte	U32	rw	*1	-	
2000	00	Mittelungszeit [ms]	U16	rw	00'1Eh	-	39
2001	00	Autozero speichern	U8	rw	00h	-	40
2002	00	IIR Filter Grenzfrequenz	U16	rw	00'00h	-	40
2003	00	Autozero	U32	wo	„zero“	RPDO*2	41
2004	00	Status Autozero	U16	ro	00'00h	TPDO	41
2100	00	Baud-Rate	U8	rw	03h	-	41
2101	00	Identifikation	U8	rw	01h	-	42
2112		Datentyp 16/24bit	ARR			-	43
	00	Anzahl Subindex	U8	ro	01h	-	
	01	Datentypauswahl 16/24bit	U16	rw	71'30h	-	
6110		Sensortyp	ARR				34
	00	Anzahl Subindex	U8	ro	01h	-	
	01	Sensortyp	U16	ro	46h	-	
6112		Betriebsart	ARR				34
	00	Anzahl Subindex	U8	ro	01h	-	
	01	Betriebsart des Sensors	U8	ro	01h	-	
6125		Autozero	ARR				35
	00	Anzahl Subindex	U8	ro	01h	-	
	01	Tarieren von Prozesswert	string	rw	„zero“	-	
6131		Einheit Prozesswert	ARR				35
	00	Anzahl Subindex	U8	ro	01h	-	
	01	Physik. Prozesseinheit	U32	ro	FA'01'01'00h	-	
6132		Dezimalstellen	ARR				35
	00	Anzahl Subindex	U8	ro	01h	-	
	01	Anzahl Dezimalstellen	U8	ro	z.B. 02h	-	
6150		Status der Messung	ARR				36
	00	Anzahl Subindex	U8	ro	01h	-	
	01	Status Messung Prozesswert	U8	ro	z.B. 00h	TPDO	
7130		Abfrage des Messwertes 16bit	ARR				36
	00	Anzahl Subindex	U8	ro	01h	-	
	01	Messwert in 16bit	I16	ro	z.B. 01'2Ch	TPDO	
7133		Delta-Wert 16bit	ARR				37
	00	Anzahl Subindex	U8	ro	01h	-	
	01	Delta-Wert 16bit für PDO	U16	rw	00'00h	-	
8130		Abfrage des Messwertes 24bit	ARR				37
	00	Anzahl Subindex	U8	ro	01h	-	
	01	Messwert 24bit	I24	ro	z.B. 01'38'80h	TPDO	
8133		Delta-Wert 24bit	ARR				38
	00	Anzahl Subindex	U8	ro	01h	-	
	01	Delta-Wert 24bit für PDO	U24	rw	00'00'00h	-	

*1 Subindex sind nicht vorhanden, kommen erst zur Anwendung wenn Objekt neu gemappt wird.

*2 Objekt ist auf Receive PDO gemappt, RPDO hat aber kein dynamisches Mapping.

5.4 SDO-Struktur

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
-	8	CMD	Objekt		Sub-Index	Datenbytes			

Vorgang	CMD	Bemerkung
Master fordert Daten vom Slave	40h	
Slave antwortet	42h	(Gültige Datenbytes nicht spezifiziert)
	43h	(4 gültige Datenbytes)
	47h	(3 gültige Datenbytes)
	4Bh	(2 gültige Datenbytes)
	4Fh	(1 gültige Datenbytes)
Master schreibt zum Slave	22h	(Gültige Datenbytes nicht spezifiziert)
	23h	(4 gültige Datenbytes)
	27h	(3 gültige Datenbytes)
	2Bh	(2 gültige Datenbytes)
	2Fh	(1 gültige Datenbytes)
Slave antwortet	60h	

Bei Objekt und Datenbytes wird das LSB zuerst übertragen. Werden ASCII Zeichen übertragen, so werden die Zeichen leserlich übertragen (erstes Zeichen zuerst).

Der Bereich des Kommunikations-Profiles befindet sich in den Indexen 1000h-1FFFh und beinhaltet alle Parameter, welche das CAN-Netzwerk betreffen. Dieser Bereich ist in allen CANopen Geräten gleich definiert.

Die minimale Zeitdifferenz zwischen zwei SDO Botschaften darf 20ms nicht unterschreiten. Eine schnellere SDO-Kommunikation kann das Gerät in undefinierte Zustände setzen.

6 Beschreibung der Objekte

6.1 Standard Objekte

6.1.1 Geräteprofil

Das Geräteprofil (Objekt 1000h) kann wie folgt gelesen werden:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
600h + ID	8	40h	00h	10h	00h	0	0	0	0

Antwort vom CANopen Dehnungssensor:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
580h + ID	8	43h	00h	10h	00h	94h	01h	02h	00h

Byte 5 + 6: 01'94h = 404d (Geräteprofil Nummer)
 Byte 7 + 8: 00'02h (zusätzliche Information, Analog Input)

Das Objekt 1000h kann nur gelesen werden und hat keinen Sub-Index.

6.1.2 Kalibrationsdatum

Das Kalibrationsdatum (Objekt 1002h) des Sensors kann wie folgt gelesen werden:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
600h + ID	8	40h	02h	10h	00h	0	0	0	0

Antwort vom CANopen Dehnungssensor:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
580h + ID	8	43h	02h	10h	00h	09h	0Bh	14h	00h

Beispiel Kalibrationsdatum 20.11.09:
 Byte 5: 09h Jahr 09
 Byte 6: 0Bh Monat 11 (November)
 Byte 7: 14h Tag 20
 Byte 8: Reserviert

Das Objekt 1002h kann nur gelesen werden und hat keinen Sub-Index.

6.1.3 Gerätebezeichnung

Die Gerätebezeichnung (Objekt 1008h) kann wie folgt gelesen werden:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
600h + ID	8	40h	08h	10h	00h	0	0	0	0

Antwort vom CANopen Dehnungssensor:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
580h + ID	8	43h	08h	10h	00h	44h	53h	52h	54h

Beispiel DSRT Sensor:
 Byte 5 – 8: 44'53'52'54h „DSRT“ im ASCII Format

Das Objekt 1008h kann nur gelesen werden und hat keinen Sub-Index.

6.1.4 Hardware

Die aktuelle Hardware Version (Objekt 1009h) kann wie folgt gelesen werden:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
600h + ID	8	40h	09h	10h	00h	0	0	0	0

Im ASCII-Format codierte Antwort vom CANopen Dehnungssensor:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
580h + ID	8	43h	09h	10h	00h	33h	2Eh	30h	33h

Beispiel Hardware Version 3.03:

Byte 5 – 8: 33'2E'30'33h „3.03“ im ASCII Format

Das Objekt 1009h kann nur gelesen werden und hat keinen Sub-Index.

6.1.5 Software

Die aktuelle Software Version (Objekt 100Ah) kann wie folgt gelesen werden:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
600h + ID	8	40h	0Ah	10h	00h	0	0	0	0

Im ASCII-Format codierte Antwort vom CANopen Dehnungssensor:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
580h + ID	8	43h	0Ah	10h	00h	32h	2Eh	30h	38h

Beispiel Software Version 2.08:

Byte 5 – 8: 32'2E'30'38h „2.08“ im ASCII Format

Das Objekt 100Ah kann nur gelesen werden und hat keinen Sub-Index.

6.1.6 Geräte Identität

Das Objekt 1018h enthält generelle Informationen über das Gerät und hat folgenden Aufbau:

Objekt	Sub-Index	Parameter	Länge	Access
1018h	0	Anzahl der Sub-Indexe	1 Byte	Read
	1	Vendor-ID	4 Byte	Read
	2	Produkt Code	4 Byte	Read
	3	Revisionsnummer	4 Byte	Read
	4	Seriennummer	4 Byte	Read

Die Vendor-ID (Sub-Index 1) kann wie folgt gelesen werden:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
600h + ID	8	40h	18h	10h	01h	0	0	0	0

Antwort vom CANopen Dehnungssensor:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
580h + ID	8	43h	18h	10h	01h	5Fh	00h	00h	00h

Byte 5 – 8: 5F'00'00'00h 00'00'00'5F (LSB first) → Firma Baumer

Der Produktcode (Sub-Index 2) kann wie folgt gelesen werden:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
600h + ID	8	40h	18h	10h	02h	0	0	0	0

Antwort vom CANopen Dehnungssensor:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
580h + ID	8	43h	18h	10h	02h	44h	0Dh	A8h	00h

Beispiel Product Code 11013444:

Byte 5 – 8: 44'0D'A8'00h 00'A8'0D'44 (LSB first) → 11013444 in Dezimal

Die Revisionsnummer (Sub-Index 3) kann wie folgt gelesen werden:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
600h + ID	8	40h	18h	10h	03h	0	0	0	0

Antwort vom CANopen Dehnungssensor:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
580h + ID	8	43h	18h	10h	03h	01h	02h	03h	00h

Beispiel Revisionsnummer 00030201:

Byte 5 – 8: 01'02'03'00h 00'03'02'01 (LSB first) → 00030201h

Die Seriennummer (Sub-Index 4) kann wie folgt gelesen werden:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
600h + ID	8	40h	18h	10h	04h	0	0	0	0

Antwort vom CANopen Dehnungssensor:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
580h + ID	8	43h	18h	10h	04h	7Bh	00h	00h	00h

Beispiel Seriennummer 123:

Byte 5 – 8: 7B'00'00'00h 00'00'00'7B (LSB first) → 123

Das Objekt 1018h kann nur gelesen werden.

6.2 Parameter Handling (save, load default)

Folgende veränderbaren Objekte sind in einem EEPROM gespeichert:

Objekt	Objektname	Default-Werte	
1017	Producer Heartbeat Time	00h	deaktiviert
1400	Empfang PDO1 – COB-ID + PDO valid, Sub-Index 1	40'00'02'xxh	COB-ID = 200h+ID, Empfang PDO ist aktiviert
1800	PDO1 – COB-ID + PDO valid , Sub-Index 1	40'00'01'xxh	COB-ID = 180h+ID, PDO ist aktiviert
	PDO1 – Transmission Type , Sub-Index 2	FFh	Sendet nach Ablauf des Event Timer
	PDO1 – Event Time , Sub-Index 5	03'E8h	1000 ms
1801	PDO2 – COB-ID + PDO valid , Sub-Index 1	40'00'02'xxh	COB-ID = 280h+ID, PDO ist aktiviert
	PDO2 – Transmission Type , Sub-Index 2	02h	Sendet nach 2. Sync Message
	PDO2 – Event Time , Sub-Index 5	03'E8h	1000 ms
1802	PDO3 – COB-ID + PDO valid , Sub-Index 1	40'00'03'xxh	COB-ID = 380h+ID, PDO ist aktiviert
	PDO3 – Transmission Type , Sub-Index 2	FEh	Asynchron, sendet nach Wertveränderung
	PDO3 – Event Time , Sub-Index 5	03'E8h	1000 ms
1A00	Mapping Transmit PDO1, Sub-Index 1	71'30'01'10h	Objekt 7130 Sub-Index 1 gemappt
1A01	Mapping Transmit PDO2, Sub-Index 1	71'30'01'10h	Objekt 7130 Sub-Index 1 gemappt
1A02	Mapping Transmit PDO3, Sub-Index 1	20'04'00'10h	Objekt 2004 Sub-Index 0 gemappt
2000	Mittelungszeit	00'30h	30 ms
2001	Save Autozero automatically	00h	Keine automatische Speicherung
2002	IIR-Filter Grenzfrequenz	00'00h	IIR Filter ausgeschaltet
2100	Baud-Rate	03h	125 kBaud
2101	Identifikation (7 bit)	01h	ID 1
2112	Datentyp 16/24bit	71'30h	16bit Datentyp
7133	Interrupt Delta Input Prozesswert 16bit	00h	deaktiviert
8133	Interrupt Delta Input Prozesswert 24bit	00h	deaktiviert

6.2.1 Speichern

Mit dem Objekt 1010h können die aktuellen Parameter im EEPROM gespeichert werden. Die Objekte die gespeichert werden sind in der Tabelle zu Beginn des Kapitels 6.2 ersichtlich.

Das Speichern erfolgt, indem die Botschaft "save" als ASCII-Code auf das Objekt 1010h, Sub-Index1 gesendet wird. Die Botschaft hat folgenden Aufbau:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
600h + ID	8	22h	10h	10h	01h	73h	61h	76h	65h

Antwort vom CANopen Dehnungssensor:

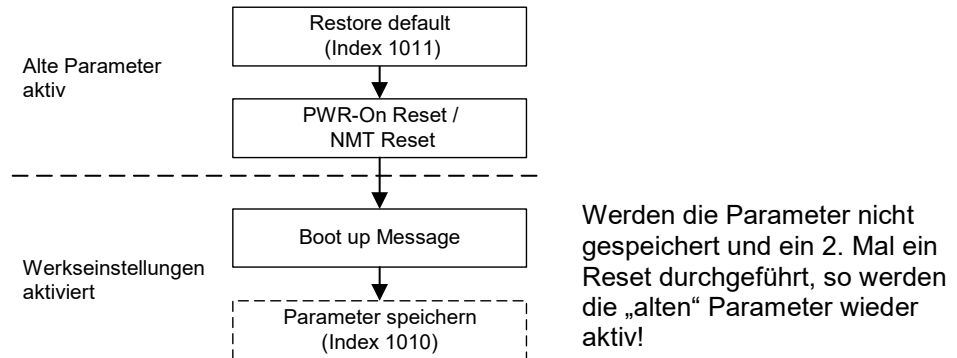
ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
580h + ID	8	60h	10h	10h	01h	0	0	0	0

6.2.2 Defaultwerte laden

Mit dem Objekt 1011h können die Werkseinstellungen geladen werden.

Die Parameter und die entsprechenden Werte befinden sich zu Beginn des Kapitels 6.2.

Funktionsweise:



Das Laden der Default-Daten erfolgt, indem die Botschaft "load" in ASCII-Code auf das Objekt 1011h, Sub-Index1 gesendet wird. Die Botschaft hat folgenden Aufbau:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
600h + ID	8	22h	11h	10h	01h	6Ch	6Fh	61h	64h

Antwort vom CANopen Dehnungssensor:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
580h + ID	8	60h	11h	10h	01h	0	0	0	0

Das Laden der Default-Werte bedeutet, dass die Werte ins RAM geladen werden. Sollen die Werte beim nächsten Reset erhalten bleiben, müssen die Parameter mit dem Objekt 1010h im EEPROM abgespeichert werden.

6.3 Gerätespezifische Objekte

Einstellung und Abfrage der sensorspezifischen Werte. Nachfolgende Objekte werden unterstützt:

Objekt	Beschreibung
6110	Sensortyp
6112	Betriebsart
6125	Tarierung
6131	Einheit Prozesswert
6132	Dezimalstellen Prozesswert
6150	Status der Messung
7130	Prozesswert 16bit
7133	Interrupt Delta-Prozesswert 16bit
8130	Prozesswert 24bit
8133	Interrupt Delta-Prozesswert 24bit

6.3.1 Sensortyp

Der Sensortyp (Objekt 6110h) kann wie folgt gelesen werden:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
600h + ID	8	40h	10h	61h	01h	0	0	0	0

Antwort vom CANopen Dehnungssensor:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
580h + ID	8	4Bh	10h	61h	01h	46h	00h	0	0

Byte 5: 46h Dehnungssensor

Das Objekt 6110h kann nur gelesen werden.

6.3.2 Betriebsart

Dieser Sensor weist zwei grundlegende Betriebsmodi auf:

- Mess-Modus
- Abgleich-Modus

Bei der Auslieferung ist der Sensor immer im Mess-Modus.

Die Betriebsart (Objekt 6112h) kann wie folgt gelesen werden:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
600h + ID	8	40h	12h	61h	01h	0	0	0	0

Antwort vom CANopen Dehnungssensor:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
580h + ID	8	4Fh	12h	61h	01h	01h	0	0	0

Byte 5: 01h Mess-Modus
 02h – 09h Reserviert
 0Ah (Abgleich-Modus)

Das Objekt 6112h kann nur gelesen werden.

6.3.3 Tarierung

Eine Tarierung (Prozesswert auf Null setzen) erfolgt, indem die Botschaft „zero“ in ASCII-Code auf den Sub-Index 1 des Objekts 6125h gesendet wird. Es kann nur der 1. Prozesswert (Dehnung) tariert werden.

Die Tarierungs-Botschaft (Objekt 6125h) hat folgenden Aufbau:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
600h + ID	8	22h	25h	61h	01h	7Ah	65h	72h	6Fh

Nachdem die Tarierung des Dehnungssensors erfolgreich war, wird folgende Antwort gesendet:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
580h + ID	8	60h	25h	61h	01h	0	0	0	0

Falls das Sensorsignal ausserhalb des Tarierbereichs liegt, wird eine SDO Fehlermeldung „Time out“ gesendet.

Die Einstellungen einer Tarierung werden mit einem Speicherbefehl (Objekt 1010h) im EEPROM gespeichert. Ist das Objekt 2001h (Save Autozero automatically) aktiviert, werden die Einstellungen nach jeder Tarierung im EEPROM gespeichert. Wird häufig tariert, soll diese Funktion deaktiviert werden.

Info: Die Tarierung kann auch über den Empfangs PDO1 oder über das Objekt 2003h ausgelöst werden.

6.3.4 Einheit Prozesswert

Die physikalische Prozesswerteinheit (Objekt 6131h) des CANopen Dehnungssensors kann wie folgt gelesen werden:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
600h + ID	8	40h	31h	61h	01h	0	0	0	0

Antwort vom CANopen Dehnungssensor:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
580h + ID	8	43h	31h	61h	01h	00h	01h	01h	FAh

Byte 5 – 8: 00'01'01'FAh FA'01'01'00 (LSB first) → Einheit „ $\mu\epsilon$ “ oder „ $\mu\text{m}/\text{m}$ “

Das Objekt 6131h kann nur gelesen werden.

6.3.5 Dezimalstellen Prozesswert

Die Anzahl der Dezimalstellen des Prozesswerts (Objekt 6132h) kann wie folgt gelesen werden:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
600h + ID	8	40h	32h	61h	01h	0	0	0	0

Antwort vom CANopen Dehnungssensor:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
580h + ID	8	4Fh	32h	61h	01h	02h	0	0	0

Byte 5: 02h 2 Dezimalstellen

Das Objekt 6132h kann nur gelesen werden. Wird der Datentyp umgestellt (16/24bit über 2112h) werden die Dezimalstellen in diesem Objekt auch angepasst.

6.3.6 Status der Messung

Der Status der Messung (Objekt 6150h) des CANopen Dehnungssensors kann wie folgt gelesen werden:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
600h + ID	8	40h	50h	61h	01h	0	0	0	0

Antwort vom CANopen Dehnungssensor:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
580h + ID	8	4Fh	50h	61h	01h	00h	0	0	0

Byte 5: 00h Messwert aktuell
 03h Überlauf des AD – Wandlers
 05h Unterlauf des AD – Wandlers

Das Objekt 6150h kann nur gelesen werden.

6.3.7 Abfrage Prozesswert 16bit

Der 16bit Prozesswert – Dehnung (Objekt 7130h) des Dehnungssensors kann wie folgt gelesen werden:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
600h + ID	8	40h	30h	71h	01h	0	0	0	0

Antwort vom CANopen Dehnungssensors:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
580h + ID	8	4Bh	30h	71h	01h	2Ch	01h	0	0

Byte 5 + 6: 2C'01h 01'2C (LSB first) → 300
 $300 / 10^{\text{Anzahl Dezimalstellen}(6132)} + \text{Einheit}(6131) = 3.00 \mu\text{E}$

Der Prozesswert wird als 16-bit Integer 2-er Komplement ausgegeben. Mit dem Objekt 2112h kann zwischen dem 16bit (Objekt 7130h) und dem 24bit (Objekt 8130h) Prozesswert umgeschaltet werden.

Beispiel für negative Dehnung:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
580h + ID	8	4Bh	30h	71h	01h	D4h	FEh	0	0

Byte 5 + 6: D4'FEh FE'D4 (LSB first) → -300
 $-300 / 10^{\text{Anzahl Dezimalstellen}(6132)} + \text{Einheit}(6131) = -3.00 \mu\text{E}$

Das Objekt 7130h kann nur gelesen werden.

6.3.8 Abfrage Prozesswert 24bit

Der 24bit Prozesswert – Dehnung (Objekt 8130h) des Dehnungssensors kann wie folgt gelesen werden:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
600h + ID	8	40h	30h	81h	01h	0	0	0	0

Antwort vom CANopen Dehnungssensors:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
580h + ID	8	47h	30h	81h	01h	80h	38h	01h	0

Byte 5 - 7: 80'38'01h 01'38'80 (LSB first) → 80000
 $80000/10^{\text{Anzahl Dezimalstellen}(6132)} + \text{Einheit}(6131) = 800.00 \mu\epsilon$

Der Prozesswert wird als 24-bit Integer 2-er Komplement ausgegeben. Mit dem Objekt 2112h kann zwischen dem 16bit (Objekt 7130h) und dem 24bit (Objekt 8130h) Prozesswert umgeschaltet werden.

Beispiel für negative Dehnung:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
580h + ID	8	4Bh	30h	81h	01h	80h	C7h	FEh	0

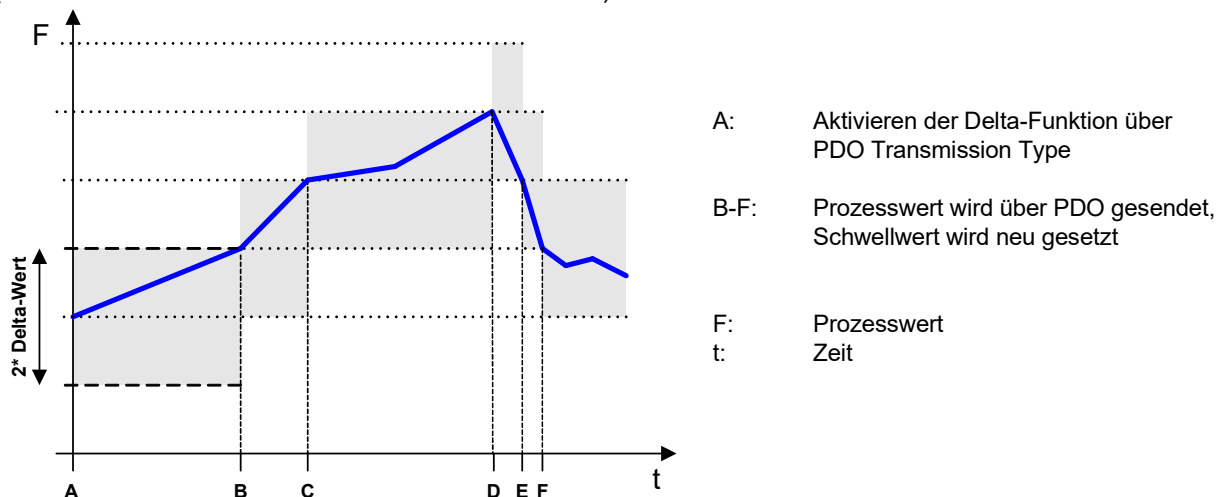
Byte 5 - 7: 80'C7'FEh FE'C7'80 (LSB first) → -80000
 $-80000/10^{\text{Anzahl Dezimalstellen}(6132)} + \text{Einheit}(6131) = -800.00 \mu\epsilon$

Das Objekt 8130h kann nur gelesen werden.

6.3.9 Delta-Prozesswert

Nach dem Aktivieren der Deltafunktion wird beim Überschreiten der definierten Schwelle der aktuelle Prozesswert über einen PDO geschickt und die Schwelle neu gesetzt. Beim nächsten Über- oder Unterschreiten der Schwelle wird erneut ein PDO gesendet.

(Schwelle = momentaner Prozesswert +/- Deltawert)



Über das Objekt 7133h (16bit) oder 8133h (24bit) kann der Delta-Wert gelesen und geschrieben werden.

Die Delta-Funktion wird über den Transmission Type des PDO1 (Objekt 1800h), PDO2 (Objekt 1801h) und PDO3 (Objekt 1802h) aktiviert oder deaktiviert.

Der Delta-Wert (Objekt 7133h) kann wie folgt geändert werden:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
600h + ID	8	22h	33h	71h	01h	F4h	01h	0	0

Gewünschter Deltawert 5 μE

$5 * 10^2 = 500 = 01'F4h$ (Delta-Wert)

Byte 4: 01h

Sub-Index 1 \rightarrow Delta-Wert für PDO1

Byte 5 + 6: F4'01h

01'F4 (LSB first) \rightarrow 500

Antwort vom CANopen Dehnungssensor:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
580h + ID	8	60h	33h	71h	01h	0	0	0	0

Wird der Wert 00h in das Objekt 7133h Sub-Index 01h geschrieben, ist die Delta-Wert Funktion ausgeschaltet.

Der Delta-Wert (Objekt 7133h) kann wie folgt gelesen werden:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
600h + ID	8	40h	33h	71h	01h	0	0	0	0

Antwort vom CANopen Dehnungssensor:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
580h + ID	8	4Bh	33h	71h	01h	F4h	01h	0	0

Byte 5 + 6: F4'01h

01'F4 (LSB first) \rightarrow 500 \rightarrow 5 μE

Der Deltawert des 24bit Prozesswerts (8133h) verhält sich gleich wie der Deltawert des 16bit Prozesswerts (7133h). Es können aber 24bit anstelle von 16bit eingegeben werden.

6.4 Herstellerspezifische Objekte

Die nachfolgenden Objekte sind herstellerspezifisch definiert. Sie dienen der Einstellung des CANopen Dehnungssensors. Folgende Objekte werden unterstützt:

Objekt	Beschreibung
2000	Mittelungszeit
2001	Tarierung speichern
2002	IIR Filter Grenzfrequenz
2003	Autozero
2004	Status Autozero
2100	Baudrate
2101	Identifikation
2112	Datentyp 16/24bit

6.4.1 Mittelungszeit

Die Mittelungszeit bestimmt die Zeit in ms über welche die Messwerte arithmetisch gemittelt werden. Die Mittelungszeit ist zwischen 0...1000 (0...3E8h) wählbar. Für die Mittelung wird intern mit einem 32bit Buffer gearbeitet. Ist die Mittelungszeit nun >32ms so wird nur jeder 2te Messwert abgespeichert, für Mittelungszeiten >64ms nur jeder 3te Messwert, usw.

Ist der IIR Filter eingeschaltet, sollte die Mittelungszeit ausgeschaltet werden.

Die aktuelle Mittelungszeit (Objekt 2000h) kann wie folgt gelesen werden:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
600h + ID	8	40h	00h	20h	00h	0	0	0	0

Antwort vom CANopen Dehnungssensor:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
580h + ID	8	4Bh	00h	20h	00h	1Eh	00h	0	0

Byte 5 + 6: 1E'00h 00'1Eh (LSB first) → 30ms (Default-Wert)

Ändern der Mittelungszeit:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
600h + ID	8	22h	00h	20h	00h	64h	00h	0	0

Byte 5 + 6: 64'00h 00'64h (LSB first) → 100ms

Antwort vom CANopen Dehnungssensor:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
580h + ID	8	60h	00h	20h	00h	0	0	0	0

6.4.2 Tarierung speichern

Bei einer zyklischen Tarierungen ist empfohlen "Tarierung speichern" auszuschalten.

Der aktuelle Status der Tarierung-Speicherung (Objekt 2001h) kann wie folgt gelesen werden:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
600h + ID	8	40h	01h	20h	00h	0	0	0	0

Antwort vom CANopen Dehnungssensor:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
580h + ID	8	4Fh	01h	20h	00h	01h	0	0	0

Byte 5: 01h 0 → autom. Speicherung nach Tarierung ausgeschaltet
 1 → autom. Speicherung nach Tarierung eingeschaltet

Ändern des Status:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
600h + ID	8	22h	01h	20h	00h	00h	0	0	0

Byte 5: 00h 0 → autom. Speicherung nach Tarierung ausgeschaltet

Antwort vom CANopen Dehnungssensor:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
580h + ID	8	60h	01h	20h	00h	0	0	0	0

6.4.3 IIR Filter Grenzfrequenz

Dieses Objekt beinhaltet die Grenzfrequenz für ein IIR-Filter 1ster Ordnung. Die Filterkonstantenberechnung ist auf eine Abtastrate von 1000 Samples pro Sekunde. abgestimmt. Die Grenzfrequenz kann zwischen 0...499 (0...1F3h) eingestellt werden, wobei der Wert 0 das Filter ausschaltet (Default). Ist das IIR Filter eingeschaltet, sollte die Mittelungszeit ausgeschaltet werden und umgekehrt.

Die aktuelle IIR-Filter Grenzfrequenz (Objekt 2002h) kann wie folgt gelesen werden:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
600h + ID	8	40h	02h	20h	00h	0	0	0	0

Antwort vom CANopen Dehnungssensor:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
580h + ID	8	4Bh	02h	20h	00h	0Ah	00h	0	0

Byte 5 + 6: 00'00h 00'00h → ausgeschaltet (Default-Wert)

Ändern der IIR-Filter Grenzfrequenz:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
600h + ID	8	22h	02h	20h	00h	64h	00h	0	0

Byte 5 + 6: 64'00h 00'64h → 100Hz

Antwort vom CANopen Dehnungssensor:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
580h + ID	8	60h	02h	20h	00h	0	0	0	0

6.4.4 Autozero

Dieses Autozero (Objekt 2003h) hat die gleiche Funktion wie das Objekt 6125h.

Dieses Objekt ist jedoch mappbar und wird für den Receive-PDO verwendet. Der Receive-PDO hat kein dynamisches Mapping.

6.4.5 Status Autozero

Der Status Autozero (Objekt 2004h) widerspiegelt den aktuellen Status der Tarierung, die über den Receive PDO gestartet wurde. Dieses Objekt ist mappbar und ist Default mässig im Transmit PDO3 konfiguriert. Ist PDO 3 aktiviert, wird ein Feedback über Start und Abschluss der Tarierung ausgegeben.

- 00'00h es ist noch keine Tarierung über Receive PDO ausgeführt worden
- 75'00h („u“ in ASCII) entspricht einer laufenden Tarierung
- 66'00h („f“ in ASCII) entspricht einer erfolgreichen Tarierung
- 65'72h („er“ in ASCII) entspricht einer fehlgeschlagenen Tarierung

6.4.6 Baudrate

Die Baudrate bestimmt die Geschwindigkeit mit welcher der gesamte Bus betrieben wird. Im Bus müssen alle Teilnehmer mit der gleichen Baudrate konfiguriert sein. Der CANopen Dehnungssensor wird mit dem Default Wert von 3 (entspricht 125kBaud) ausgeliefert.

Die aktuelle Baudrate (Objekt 2100h) kann wie folgt gelesen werden:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
600h + ID	8	40h	00h	21h	00h	0	0	0	0

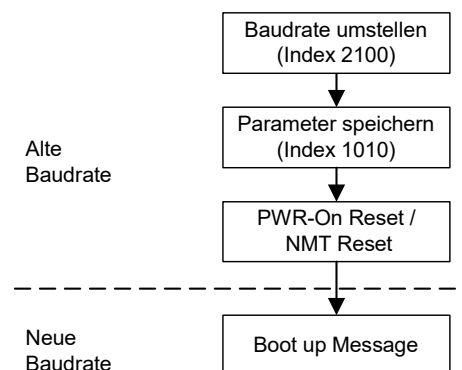
Antwort vom CANopen Dehnungssensor:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
580h + ID	8	4Fh	00h	21h	00h	03h	0	0	0

Byte 5:	00h → 10 kBaud	04h → 250 kBaud
	01h → 20 kBaud	05h → 500 kBaud
	02h → 50 kBaud	06h → 800 kBaud
	03h → 125 kBaud	07h → 1000 kBaud

Funktionsweise:

Ändern der Baudrate:



Ändern der Baudrate auf 500 kBaud:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
600h + ID	8	22h	00h	21h	00h	05h	0	0	0

Byte 5: 05h 5 → 500 kBaud

Antwort vom CANopen Dehnungssensor:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
580h + ID	8	60h	00h	21h	00h	0	0	0	0

6.4.7 Identifikation

Die ID eines CANopen Knoten ist die Identifikation im Netzwerk, dadurch darf jede ID in einem Netzwerk nur für ein Gerät vergeben werden. Ansonsten werden jeweils 2 Sensoren angesprochen. Es können Identifikationsnummer von 1 ... 127 vergeben werden.

Die aktuelle Identifikationsnummer (Objekt 2101h) kann wie folgt gelesen werden:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
600h + ID	8	40h	01h	21h	00h	0	0	0	0

Antwort vom CANopen Dehnungssensor:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
580h + ID	8	4Fh	01h	21h	00h	01h	0	0	0

Byte 5: 01h ID 1 (Default-Wert)

Ändern der Identifikation auf Nummer 5:

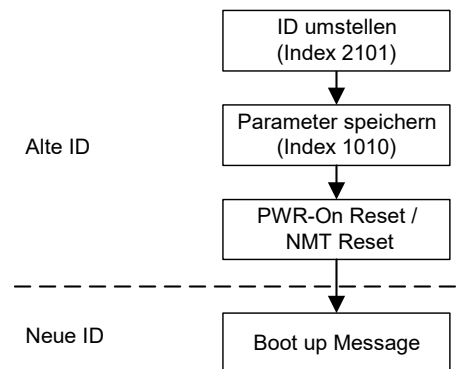
ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
600h + ID	8	22h	01h	21h	00h	05h	0	0	0

Byte 5: 05h ID 5

Antwort vom CANopen Dehnungssensor:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
580h + ID	8	60h	01h	21h	00h	0	0	0	0

Funktionsweise:



6.4.8 Datentyp 16/24bit

Mit dem Objekt 2112h kann der Benutzer die Datenlänge des Prozesswerts auswählen. Abhängig vom eingestellten Wert kann der Prozesswert auf dem Objekt 7130h oder 8130h abgefragt werden. Beim Umstellen des Datentyps (Länge des Prozesswerts) werden die Dezimalstellen (6132h) und das PDO mapping neu geladen.

Der aktuelle Datentyp (Objekt 2112h) kann wie folgt gelesen werden:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
600h + ID	8	40h	12h	21h	01h	0	0	0	0

Antwort vom CANopen Dehnungssensor:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
580h + ID	8	4Bh	12h	21h	01h	30h	71h	0	0

Byte 5 + 6: 30'71h 71'30h → 16bit Prozesswert

Ändern des Datentyps von 16bit auf 24bit:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
600h + ID	8	22h	12h	21h	01h	30h	81h	0	0

Byte 5 + 6: 30'81h 81'30h → 24bit Prozesswert

Antwort vom CANopen Dehnungssensor:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
580h + ID	8	60h	12h	21h	01h	0	0	0	0

6.5 PDO-Kommunikation Objekte

Über die Kommunikation mittels PDO (Process Data Objects) ist es möglich bestimmte Werte des CANopen Dehnungssensors auf einfache und schnelle Weise zu übermitteln. Über die gleiche Kommunikationsart kann ein Autozero im Dehnungssensor durchgeführt werden. Bei den Dehnungssensoren ist ein dynamisches PDO-Mapping für die Sende PDOs realisiert. Die gesendeten Parameter wie auch die Bedingung für das Senden eines PDO's können vom Anwender definiert werden.

Der PDO kann über eine Sync-Nachricht, nach Ablauf der Event-Time oder beim Überschreiten eines Delta-Werts ausgelöst werden.

Die Funktionalität der PDOs ist nur im Operational-Modus des Sensors aktiviert.

6.5.1 Empfangs PDO 1 (Tarierung)

Über das Objekt 1400h werden die Einstellungen vorgenommen, um mit den Empfangs-PDOs zu arbeiten.

Das Objekt 1400h (Receive PDO Communication Parameter) hat folgenden Aufbau:

Objekt	Sub-Index	Parameter	Länge	Access
1400h	0	Anzahl der Sub-Indexe	1 Byte	read
	1	Aktivierung & ID der PDO 1 (COB-ID)	4 Byte	read / write
	2	Art der Sendung (Transmission Type)	1 Byte	read / write

Die PDO wird wie folgt aktiviert/deaktiviert (Sub-Index 1):

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
600h + ID	8	22h	00h	14h	01h	01h	02h	0	40h

Byte 5+6 01'02h 02'01h (LSB first) → 200h + ID 1 (COB-ID)
 Byte 8 40h PDO aktiviert und RTR nicht unterstützt
 C0h PDO deaktiviert und RTR nicht unterstützt

Der Transmission Type (Sub-Index 2) kann wie folgt abgefragt werden:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
600h + ID	8	40h	00h	14h	02h	00h	0	0	0

Antwort vom CANopen Dehnungssensor:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
580h + ID	8	4F	00h	14h	02h	FEh	0h	0h	0h

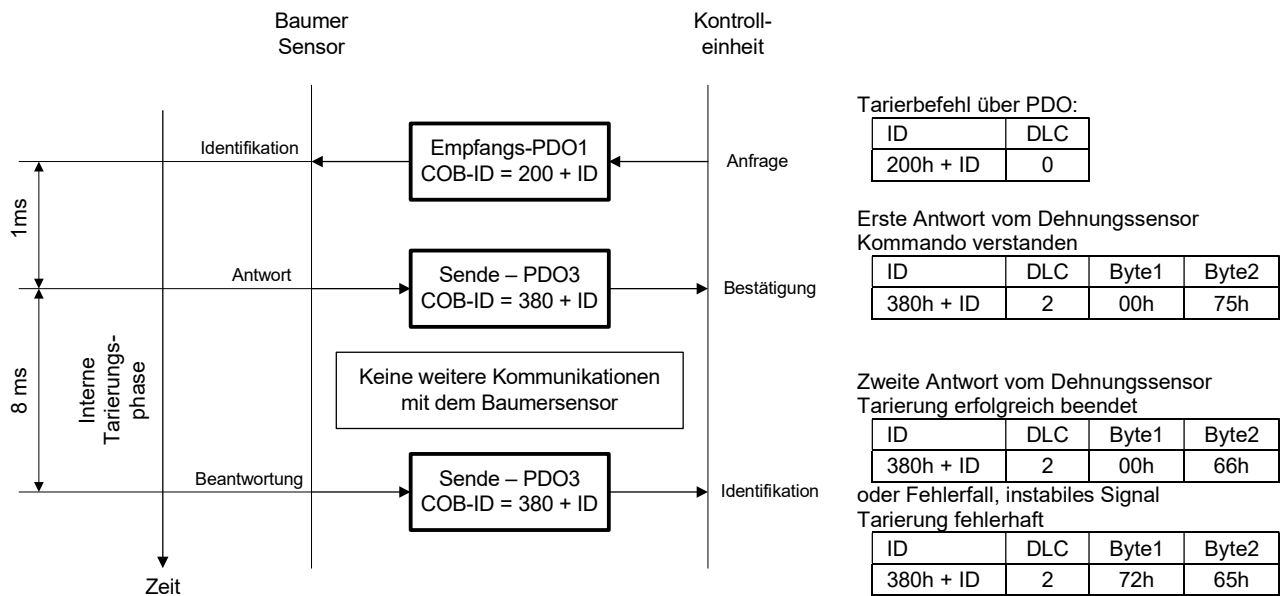
Byte 5 FEh 254d → senden nach Änderung (siehe unten)

Funktionsbeschreibung: Tarierung über Empfangs PDO 1

Über den Empfangs PDO 1 kann eine Tarierung (Autozero) ausgelöst werden.

Für einen definierten Ablauf sind folgende Punkte zu beachten:

- Sensor befindet sich im „Operational Mode“
- Empfang PDO 1 ist aktiviert und Transmission Type vom RPDO ist FEh
- Sende PDO 3 aktiviert und Transmission Type von TPDO3 ist FEh
- Mapping von Sende PDO 3: Objekt 2004, Sub-Index 0



6.5.2 Empfangs PDO 1 Mapping Parameter

Objekt 1600h (Receive PDO Mapping Parameter)

Über das Objekt 1600h kann das Mapping der Empfangs-PDO 1 abgefragt werden. Der Empfangs-PDO hat ein fixes Mapping. D.h. das Mapping kann nicht angepasst werden.

Abfrage über SDO-Kommandos:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
600h + ID	8	40h	00h	16h	01h	0	0	0	0

Antwort vom CANopen Dehnungssensor:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
580h + ID	8	43h	00h	16h	01h	00h	00h	03h	20h

Byte 4: 01h 1. gemapptes Objekt in Empfang PDO 1
 02h – 04h keine weiteren gemappte Objekte
 Byte 5: 20h 20h → 32 Bit Datenlänge des Objekts
 Byte 6: 00h 00h → gemappter Sub-Index 0
 Byte 7 + 8: 03'20h 20'03h → gemapptes Objekt → Autozero Funktion

Das Objekt 1600h kann nur gelesen werden.

6.5.3 Sende PDO 1

Über den Objekt 1800h werden die Einstellungen vorgenommen, um mit den Sende-PDO's zu arbeiten.

Die PDO's sollten nicht schneller als die entsprechende Messrate angefordert werden.
 (1000 Messungen / Sekunde) > 1 ms

Das Objekt hat den folgenden Aufbau:

Objekt	Sub-Index	Parameter	Länge	Access
1800h	0	Anzahl der Sub-Indexe	1 Byte	read
	1	Aktivierung & ID der PDO 1 (COB-ID)	4 Byte	read / write
	2	Art der Sendung (Transmission Type)	1 Byte	read / write
	5	Event time in ms	2 Byte	read / write

Die PDO wird wie folgt aktiviert/deaktiviert (Sub-Index 1):

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
600h + ID	8	22h	00h	18h	01h	81h	01h	0	40h

Byte 5 + 6: 81'01h 01'81h (LSB first) → 180h + ID 1 (COB-ID)
 Byte 8: 40h PDO aktiviert und RTR nicht unterstützt
 C0h PDO deaktiviert und RTR nicht unterstützt

Antwort vom CANopen Dehnungssensor:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
580h + ID	8	60h	00h	18h	01h	0	0	0	0

Die Art der Sendung (Sub-Index 2) wird wie folgt verändert:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
600h + ID	8	22h	00h	18h	02h	FFh	0	0	0

Byte5 01h – F0h ausgelöst nach dem n^{ten} Sync (1 – 240)
 FEh ausgelöst wenn Messwert +/-Delta-Wert übersteigt
 (Delta-Wert wird über Objekt 7133 definiert)
 FFh ausgelöst nach Ablauf der Event Time
 (Event-Time definiert in Sub-Index 05h)

Antwort vom CANopen Dehnungssensor:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
580h + ID	8	60h	00h	18h	02h	0	0	0	0

Die Event Time legt fest, in welchem Zyklus der PDO gesendet werden soll. Für das Aussenden der PDO ist ein interner Timer mit der Zeitbasis von Event Time verantwortlich.
 Der Wertebereich ist zwischen $0 \dots 2^{16} - 1 = 65535$ [ms] festgelegt.

Die Event Time (Sub-Index 5) wird wie folgt verändert:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
600h + ID	8	22h	00h	18h	05h	E8h	03h	0	0

Byte 5 + 6: E8'03h 03'E8h (LSB first) → Event Time 1000 [ms] (Default-Wert)

Antwort vom CANopen Dehnungssensor:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
580h + ID	8	60h	00h	18h	05h	0	0	0	0

Zu kleine Werte machen keinen Sinn, da der Bus zu stark belastet und die Messwerte eventuell noch nicht aktualisiert sind (siehe Beginn dieses Kapitels).

Das Abspeichern der eingestellten PDO-Parameter erfolgt nicht automatisch, sondern müssen manuell über das Objekt 1010h getätigt werden.

6.5.4 Sende PDO 2

Objekt 1801h (Transmit PDO Communication Parameter):

Über das Objekt 1801h werden die Einstellungen für die PDO 2 vorgenommen. Die Parameter sind identisch mit dem PDO 1. Detaillierte Angaben können von PDO 1 entnommen werden.

6.5.5 Sende PDO 3

Objekt 1802h (Transmit PDO Communication Parameter):

Über das Objekt 1802h werden die Einstellungen für die PDO 3 vorgenommen. Die Parameter sind identisch mit dem PDO 1. Detaillierte Angaben können von PDO 1 entnommen werden.

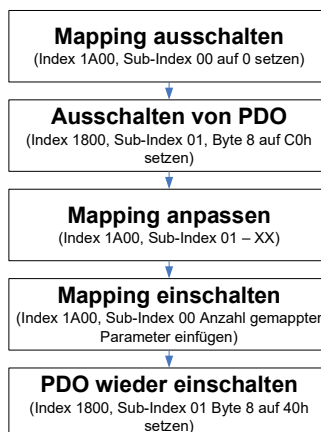
6.5.6 Sende PDO 1 Mapping Parameter

Über das Objekt 1A00h kann das Mapping der Sende-PDO 1 festgelegt und abgefragt werden.

Das Objekt hat folgenden Aufbau:

Objekt	Sub-Index	Parameter	Länge	Access
1A00h	0	Anzahl gemappter Objekte im PDO 1	1 Byte	read / write
	1	erstes gemapptes Objekt im PDO 1	4 Byte	read / write
	2	2tes bis 4tes gemapptes Objekt im PDO 1	4 Byte	read / write

Vorgehen beim Einstellen des Mappings:



Der Sende-PDO 1 hat eine Grösse von 8 Byte, die der Benutzer frei konfigurieren kann.

Aus folgenden Objekten kann ausgewählt werden:

Objekt	Sub-Index	Parameter	Länge
7130	01	Prozesswert 1 16bit (Dehnungswert)	2 Bytes
8130	01	Prozesswert 1 24bit (Dehnungswert)	3 Bytes
6150	01	Status Prozesswert 1	1 Byte
1001	00	Error Register	1 Byte
2004	00	Status Autozero	2 Bytes

Bei der Zuweisung der gemappten Objekte muss auch die Datenlänge in Byte 5 definiert werden. In der untenstehenden Tabelle ist der entsprechende Code dafür aufgelistet:

Datenlänge	Byte 5	Beschreibung
1 Byte	08h	08h => 8dez: 8bit werden gemappt
2 Bytes	10h	10h => 16dez: 16bit werden gemappt
3 Bytes	18h	18h => 24dez: 24bit werden gemappt
4 Bytes	20h	20h => 32dez: 32bit werden gemappt

Mapping des PDO 1 über SDO-Befehl:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
600h + ID	8	22h	00h	1Ah	01h	10h	01h	30h	71h

Beispiel PDO 1 mit Prozesswert 1:

Byte 4:	01h	1. gemapptes Objekt in PDO 1
	02h – 04h	2. bis 4. gemapptes Objekt in PDO 1 (Achtung nur 8Byte)
Byte 5:	10h	16 Bit Datenlänge des gemappten Objekt (2 Byte)
Byte 6:	01h	gemappter Sub-Index 01
Byte 7 + 8:	30'71h	7130 → gemappter Objekt

Antwort vom CANopen Dehnungssensor:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
580h + ID	8	60h	00h	1Ah	01h	0	0	0	0

Aufbau der Botschaft beim Senden des PDO 1:

ID	DLC	Byte1	Byte2*	Byte3*	Byte4*	Byte5*	Byte6*	Byte7*	Byte8*
180h + ID	2	68h	10h						

*Byte werden nur gesendet wenn diese im Objekt 1A00 gemappt wurden

6.5.7 Sende PDO 2 Mapping

Objekt 1A01h (Transmit PDO Mapping Parameter):

Über das Objekt 1A01h kann das Mapping der Sende-PDO 2 abgefragt werden und hat den gleichen Aufbau wie der PDO 1.

6.5.8 Sende PDO 3 Mapping

Objekt 1A02h (Transmit PDO Mapping Parameter):

Über das Objekt 1A02h kann das Mapping der Sende-PDO 3 abgefragt werden und hat den gleichen Aufbau wie der PDO 1.

6.5.9 Sync ID

Beim Sensor ist die Sync-Generierung ausgeschaltet und es werden nur PDO Meldungen gesendet wenn das Objekt 180xh Sub-Index 2 entsprechende eingestellt wurde.

Im Objekt 1005h kann die ID für die Sync-Nachrichten abgefragt werden. Liegt eine Sync-Nachricht mit der folgenden ID auf dem Bus, so kann ein PDO ausgelöst werden (vgl. PDO-Kommunikation).

Die Sync-ID (Objekt 1005h) kann wie folgt gelesen werden:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
600h + ID	8	40h	05h	10h	00h	0	0	0	0

Antwort vom CANopen Dehnungssensor:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
580h + ID	8	43h	05h	10h	00h	80h	0	0	0

Die ID ist auf 80h festgelegt. Dies gewährleistet den Sync-Botschaften eine hohe Priorität auf dem CAN-Bus.

Das Objekt 1005h kann nur gelesen werden und hat kein Sub-Index.

7 Fehlermeldungen und Dienste

7.1 Error Register & History

7.1.1 Error Register

Das Error Register (Objekt 1001h) kann wie folgt gelesen werden:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
600h + ID	8	40h	01h	10h	00h	0	0	0	0

Antwort vom CANopen Dehnungssensor:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
580h + ID	8	4Fh	01h	10h	00h	81h	0	0	0

Byte 5: 81h herstellerspezifischer Fehler aufgetreten
 Bit 0=1 → Fehler aufgetreten
 Bit 7=1 → Fehler ist herstellerspezifisch

Das Objekt 1001h kann nur gelesen werden und hat keinen Sub-Index.

7.1.2 Emergency History

Das Objekt 1003h speichert die letzten 16 Fehlernachrichten (Emergency Messages) im RAM, die im Betrieb aufgetreten sind. Nach dem nächsten Reset oder Power Down werden die Daten gelöscht. Wird 00h auf den Sub-Index 0 geschrieben, so wird die Aufzeichnung ebenfalls gelöscht.

Das Objekt hat folgenden Aufbau:

Objekt	Sub-Index	Parameter	Länge	Access
1003h	0	Anzahl Fehlernachrichten	1 byte	Read/Write
	1...16	Fehlernachrichten	8 byte	Read

Der Sub-Index 0 (Anzahl aufgetretene Fehlernachrichten) kann wie folgt gelesen werden:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
600h + ID	8	40h	03h	10h	00h	0	0	0	0

Antwort vom CANopen Dehnungssensor:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
580h + ID	8	4Fh	03h	10h	00h	02h	0	0	0

Byte 5: 02h 2 Fehlernachrichten aufgezeichnet

Die Emergency Messages werden gelöscht, indem eine 0 auf den Sub-Index 0 geschrieben wird:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
600h + ID	8	22h	03h	10h	00h	00h	0	0	0

Antwort vom CANopen Dehnungssensor:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
580h + ID	8	60h	03h	10h	00h	0	0	0	0

Die Fehlernachrichten (Sub-Index 1...16) können wie folgt gelesen werden:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
600h + ID	8	40h	03h	10h	01h	0	0	0	0

Byte 4: 01h letzte gespeicherte Fehlernachricht
 02h – 10h ältere Fehlernachrichten

Antwort vom CANopen Dehnungssensor:

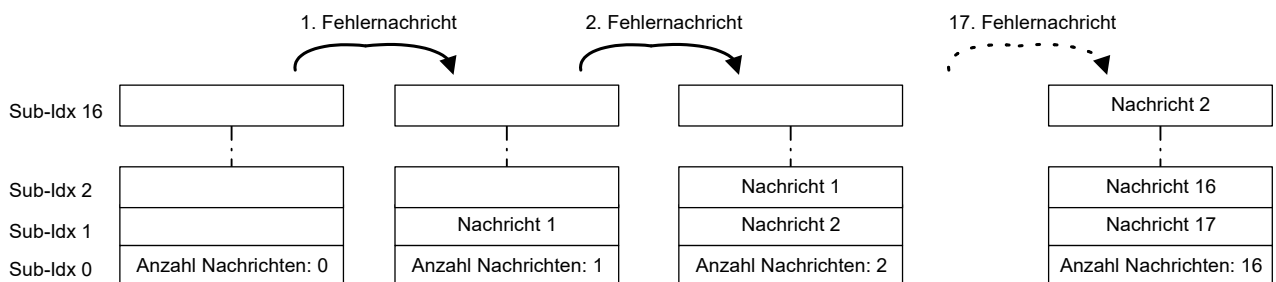
ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
580h + ID	8	43h	03h	10h	01h	00h	FFh	81h	14h

Der Fehlercode wird im Kapitel „Emergency Messages“ beschrieben.

Wird ein Sub-Index ohne aufgetretene Fehlermeldung abgefragt so sendet der Sensor folgende Antwort:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
580h + ID	8	80h	03h	10h	01h	11h	00h	09h	06h

Funktionsweise:



Der Sensor kann die letzten 16 Fehlernachrichten speichern. Die letzte Nachricht wird unter dem Sub-Index 1 gespeichert, alle vorherigen werden um eine Position nach oben geschoben.

Ist der Speicher voll und es tritt eine neue Nachricht auf wird die älteste Nachricht (Sub-Index 16) aus dem Speicher geschoben.

7.2 SDO-Fehlermeldungen

Bei fehlerhaftem Zugriff auf einen Objekt erhalten Sie eine Fehlermeldung als Antwort. Eine Fehlermeldung hat den folgenden Aufbau:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
580h + ID	8	80h	Objekt		Sub-Index	Abort Code			

Die ID der Botschaft sowie das Objekt und der Sub-Index beziehen sich auf die Botschaft, welche den Fehler verursacht hat.

Die Fehlermeldungen können folgende Inhalte aufweisen:

Abort Code	Bedeutung
05 04 00 00h	Time out (bei Tarierung: Signal ausserhalb Tarierbereich)
05 04 00 01h	Client / Server-Befehl ist nicht gültig oder unbekannt
06 01 00 01h	Nur Schreibzugriff möglich
06 01 00 02h	Nur Lesezugriff möglich
06 02 00 00h	Objekt existiert nicht
06 04 00 41h	Objekt kann nicht gemappt werden oder Mapping nicht deaktiviert
06 04 00 42h	Anzahl und Länge gemappter Objekte übersteigt zulässige PDO Länge
06 04 00 43h	Inkompatibilität von allgemeinen Parametern. Objekt kann nicht auf Sende PDO gemappt werden
06 07 00 10h	Datentyp stimmt nicht, die Länge des Serviceparameter stimmt nicht
06 09 00 11h	Sub-Index existiert nicht
06 09 00 30h	Wertebereich des Parameters überschritten
06 09 00 31h	Geschriebener Wert zu hoch
06 09 00 32h	Geschriebener Wert zu niedrig
08 00 00 20h	Daten können nicht gespeichert werden, da die Signatur des SDO-Befehls nicht stimmt.
08 00 00 21h	Daten können nicht gespeichert werden, da die Kommunikation mit dem Speicherbaustein fehlgeschlagen ist.
08 00 00 22h	Daten können nicht gespeichert werden, da zurzeit bereits auf den Speicherbaustein zugegriffen wird.
08 00 00 24h	Keine Daten vorhanden

7.4 Heartbeat

Der CANopen Dehnungssensor bietet ebenfalls die Möglichkeit des Heartbeat Protokolls. Das Heartbeat Protokoll ermöglicht ein Error-Kontrollsystem, ohne dass dafür eine Anfrage nötig ist. Der Heartbeat Produzent sendet die Statusmeldung zyklisch (definiert im Objekt 1017h). Sollte die Meldung nicht innerhalb der definierten Zeit eintreffen, so sendet der CANbus – Controller eine entsprechende Reaktion senden (Networkmanagement Commands).

Die Zeit des Heartbeats kann zwischen 1 und 65535 (1ms bis 65.535sec) eingestellt werden. Defaultmässig ist der Heartbeat ausgeschaltet (00h im Objekt 1017h).

7.4.1 Heartbeat Time

Die Heartbeat Zeit (Objekt 1017h) kann wie folgt gelesen werden:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
600h + ID	8	40h	17h	10h	00h	0	0	0	0

Antwort vom CANopen Dehnungssensor:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
580h + ID	8	4Bh	17h	10h	00h	00h	00h	0	0

Byte 5 + 6: 00'00h 00'00h (LSB first) → ausgeschaltet (Default-Wert)

Ändern der Heartbeat Time auf 1000ms (1000d → 3E8h):

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
600h + ID	8	22h	17h	10h	00h	E8h	03h	0	0

Byte 5 + 6: E8'03h 03'E8h (LSB first) → 1000d → 1000ms

Antwort vom CANopen Dehnungssensor:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
580h + ID	8	60h	17h	10h	00h	0	0	0	0

Wird diese Zeit auf 0 gesetzt, so ist der Heartbeat ausgeschaltet. (Default Einstellung)

Der Wert kann mit dem Objekt 1010h in das EEPROM gespeichert werden.

Nach dem Aktivieren des Heartbeat Protokolls werden durch den Sensor die folgenden Mitteilungen gesendet:

ID	DLC	Byte1
700h + ID	1	04h

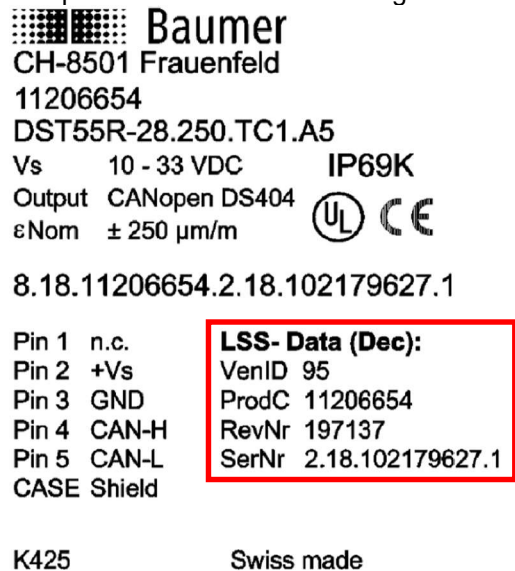
Byte 1: 00h Aufstarten (Boot up)
 04h Stop Mode
 05h Operate
 7Fh Pre-operational




7.5 LSS (Layer Setting Services)

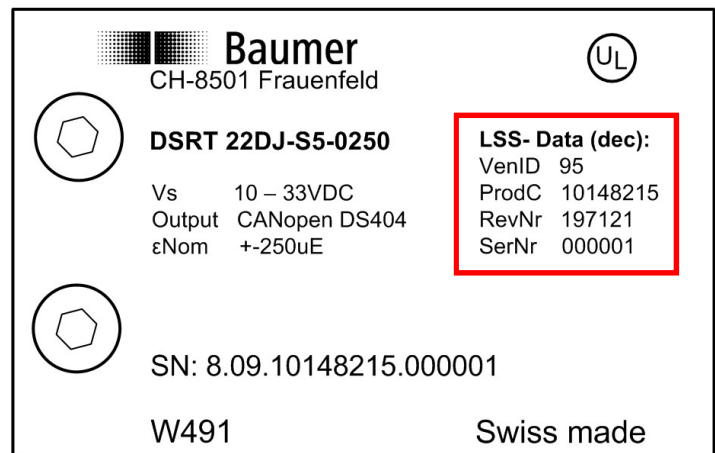
7.5.1 Aufgedruckte LSS-Daten auf Sensor





Um die LSS-Funktionalität zu nutzen, werden LSS-Daten benötigt, welche den Sensor eindeutig identifizieren. Dabei handelt es sich um die Geräte Identität (Objekt 1018h). Diese Daten sind auf jedem Dehnungssensor aufgedruckt.

Beispiel einer Sensorbeschriftung mit LSS-Daten:



 **Baumer**
 CH-8501 Frauenfeld
 11206654
 DST55R-28.250.TC1.A5
 Vs 10 - 33 VDC IP69K
 Output CANopen DS404
 εNom ± 250 µm/m  
 8.18.11206654.2.18.102179627.1
 Pin 1 n.c.
 Pin 2 +Vs
 Pin 3 GND
 Pin 4 CAN-H
 Pin 5 CAN-L
 CASE Shield
LSS- Data (Dec):
 VenID 95
 ProdC 11206654
 RevNr 197137
 SerNr 2.18.102179627.1
 K425 Swiss made



 **Baumer**
 CH-8501 Frauenfeld 
 **DSRT 22DJ-S5-0250**
 Vs 10 – 33VDC
 Output CANopen DS404
 εNom +-250uE

 SN: 8.09.10148215.000001
 W491 Swiss made
LSS- Data (dec):
 VenID 95
 ProdC 10148215
 RevNr 197121
 SerNr 000001

Aufschrift:	Bezeichnung:	Objekt:	Sub-Index:
VenID	Vendor-ID	1018h	01h
ProdC	Produkt-Code	1018h	02h
RevNr	Revisions-Nummer	1018h	03h
SerNr	Serie-Nummer	1018h	04h

Die LSS-Daten sind im dezimalen Zahlensystem dargestellt.

7.5.2 Ansprechen des Sensors über LSS

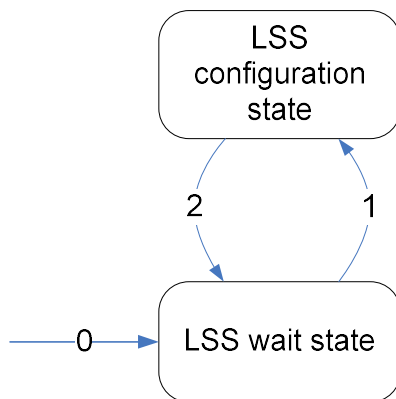
Es bestehen 2 Möglichkeiten einen Sensor mit LSS anzusprechen. Der LSS Service wird für einen Sensor in einem bestehenden Netzwerk, wie auch für einen Sensor 1:1 mit einem Master unterstützt.

In beiden Fällen muss der Bus in den Stop Mode gebracht werden.

Im LSS Betrieb werden die Sensoren mit der ID 7E5h angesprochen (es könnten verschiedene Sensoren die ID 1 haben). Der angesprochene Sensor antwortet mit der ID 7E4h.

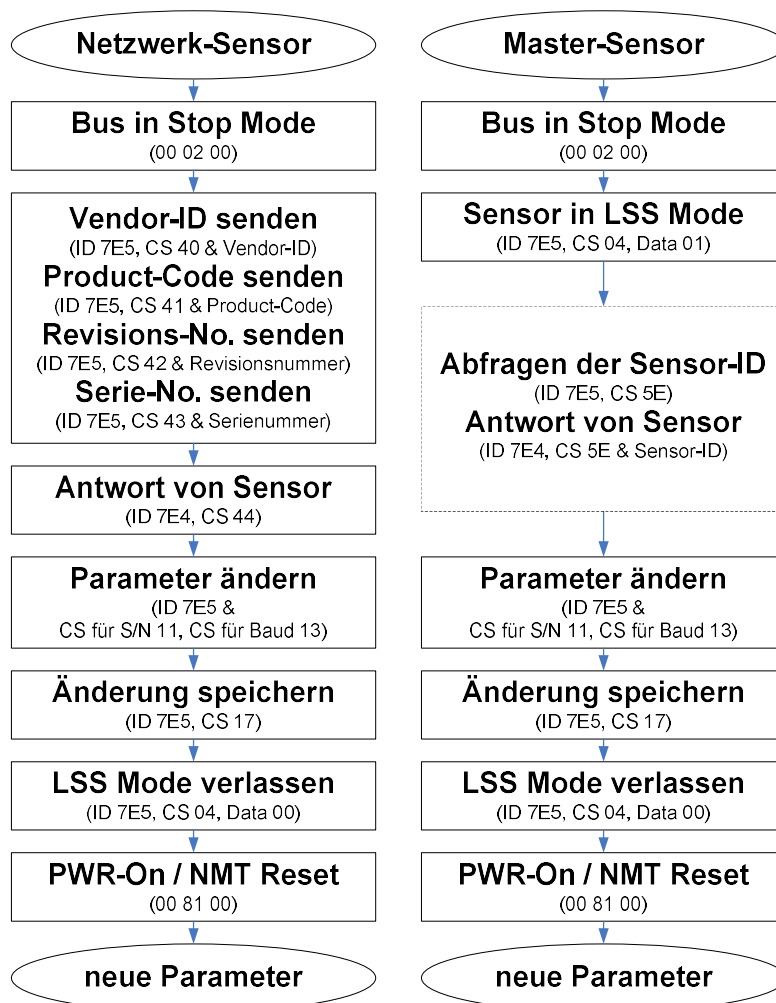
Alle Sensoren in den Stop Modus bringen:

ID	DLC	Byte1	Byte2
0	2	02h	00h



0 = automatisch in Wait State (Sensor im Stop Mode)
 1 = eintreten Configuration State (cs=04, Data=01)
 2 = zurück in Wait State (cs=04, Data=00)

Nun können die verschiedenen Wege gegangen werden, um den gewünschten Sensor in den Konfigurationsmodus zu bringen. (Direkte Verbindung Master-Sensor oder Sensor in Netzwerk)



7.5.3 Konfigurationsmodus direkte Verbindung (Master-Sensor)

Um LSS gebrauchen zu können, muss sich der Sensor im LSS Konfigurationsmodus befinden. Der Sensor kann mit folgendem Befehl in den LSS Konfigurationsmodus gebracht werden:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
7E5	8	04h	01h	0	0	0	0	0	0

Um zu überprüfen ob der Sensor in den Konfigurationsmodus gewechselt hat, kann die momentan eingestellte ID abgefragt werden:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
7E5	8	5Eh	0	0	0	0	0	0	0

Antwort vom CANopen Dehnungssensor:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
7E4	8	5Eh	01h	0	0	0	0	0	0

Beispiel "ID 1"

Byte 2: 01h angesprochener Sensor hat ID 1

Gibt der Sensor keine Antwort, so unterstützt er LSS nicht oder die Baudrate ist nicht korrekt. Für das sichere Ansprechen, muss der Service mit allen unterstützten Baudraten getestet werden. Nun können die ID und die Baudrate vom Sensor geändert werden.

7.5.4 Konfigurationsmodus von einem Sensor in einem Netzwerk

Mit den folgenden Befehlen kann der Sensor erkannt werden:

Übermitteln der Vendor-ID:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
7E5	8	40h	5Fh	00h	00h	00h	0	0	0

Vender-ID:

Byte 1: 40h Systembyte Vender-ID
 Byte 2 – 5: 5F'00'00'00h 00'00'00'5F (LSB first) → Firma Baumer electric

Übermitteln des Produkt-Code:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
7E5	8	41h	44h	0Dh	A8h	00h	0	0	0

Produkt-Code:

Byte 1: 41h Systembyte Produkt-Code
 Byte 2 – 5: 44'0D'A8'00h 00'A8'0D'44 (LSB first) → 11013444 in Dezimal

Übermitteln der Revisions-Nummer:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
7E5	8	42h	01h	02h	03h	00h	0	0	0

Revisions-Nummer:

Byte 1: 42h Systembyte Revisions-Nummer
 Byte 2 – 5: 01'02'03'00h 00'03'02'01 (LSB first) → 00030201h

7.5.6 Speichern der Änderungen

Um die Einstellungen zu übernehmen müssen diese gespeichert werden.

Speichern der Einstellungen von LSS:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
7E5	8	17h	0	0	0	0	0	0	0

Antwort vom CANopen Dehnungssensor:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
7E4	8	17h	00h	0	0	0	0	0	0

Byte 2: 00h Speicherung erfolgreich abgeschlossen
 01h Speicherung war nicht erfolgreich
 FFh spezifischer Fehler

Die Änderungen werden erst nach einem Reset übernommen.

7.5.7 LSS Mode verlassen

Der Sensor kann mit folgendem Befehl zurück in den Stop Mode gebracht werden:

ID	DLC	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7	Byte8
7E5	8	04h	00h	0	0	0	0	0	0

Auf diesen Befehl gibt der Sensor keine Antwort als Bestätigung.

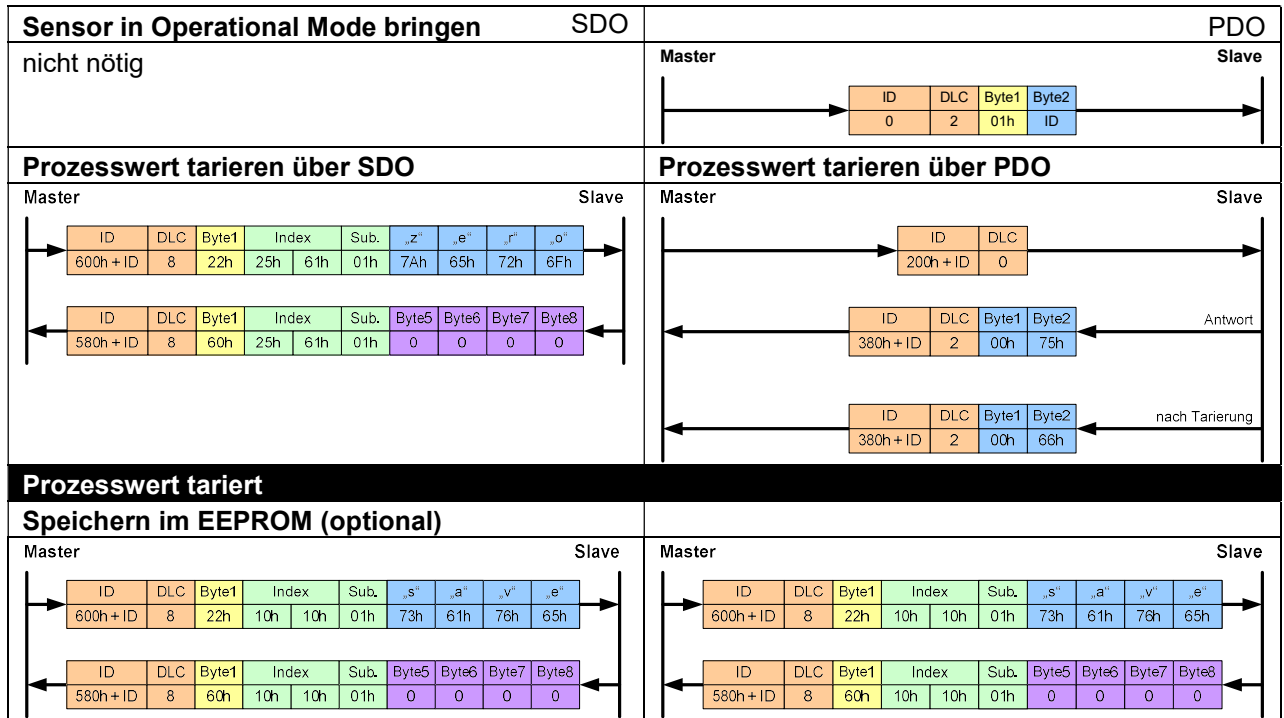
8 Anwenderbeispiele CANopen Protokoll

Dieses Kapitel soll den Umgang mit dem CANopen Produkt erleichtern. Es werden typische kundenseitige Anwendungen aufgezeigt, welche auch einfach mit dem Sensor nachgebildet werden können.

8.1 Tarieren des Prozesswerts über SDO und PDO

Der Prozesswert kann über SDO und PDO tariert werden.

Für die Tariierung über PDO muss sich der Sensor im Operational-Mode befinden.



8.2 Abfragen eines Prozesswerts über SDO (16 und 24bit)

Messinitialisierung:

1. Datentyp einstellen (Objekt 21'12h) 16-Bit	24-Bit																																																																								
<p>Master → Slave</p> <table border="1"> <tr> <td>ID</td><td>DLC</td><td>Byte1</td><td>Index</td><td>Sub.</td><td>Datentyp</td><td>Byte7</td><td>Byte8</td> </tr> <tr> <td>600h+ID</td><td>8</td><td>22h</td><td>12h 21h</td><td>01h</td><td>30h 71h</td><td>0</td><td>0</td> </tr> </table> <p>← Slave →</p> <table border="1"> <tr> <td>ID</td><td>DLC</td><td>Byte1</td><td>Index</td><td>Sub.</td><td>Byte5</td><td>Byte6</td><td>Byte7</td><td>Byte8</td> </tr> <tr> <td>580h+ID</td><td>8</td><td>60h</td><td>12h 21h</td><td>01h</td><td>0</td><td>0</td><td>0</td><td>0</td> </tr> </table> <p>71'30h => 16-Bit-Prozesswert</p>	ID	DLC	Byte1	Index	Sub.	Datentyp	Byte7	Byte8	600h+ID	8	22h	12h 21h	01h	30h 71h	0	0	ID	DLC	Byte1	Index	Sub.	Byte5	Byte6	Byte7	Byte8	580h+ID	8	60h	12h 21h	01h	0	0	0	0	<p>Master → Slave</p> <table border="1"> <tr> <td>ID</td><td>DLC</td><td>Byte1</td><td>Index</td><td>Sub.</td><td>Datentyp</td><td>Byte7</td><td>Byte8</td> </tr> <tr> <td>600h+ID</td><td>8</td><td>22h</td><td>12h 21h</td><td>01h</td><td>30h 81h</td><td>0</td><td>0</td> </tr> </table> <p>← Slave →</p> <table border="1"> <tr> <td>ID</td><td>DLC</td><td>Byte1</td><td>Index</td><td>Sub.</td><td>Byte5</td><td>Byte6</td><td>Byte7</td><td>Byte8</td> </tr> <tr> <td>580h+ID</td><td>8</td><td>60h</td><td>12h 21h</td><td>01h</td><td>0</td><td>0</td><td>0</td><td>0</td> </tr> </table> <p>81'30h => 24-Bit-Prozesswert</p>	ID	DLC	Byte1	Index	Sub.	Datentyp	Byte7	Byte8	600h+ID	8	22h	12h 21h	01h	30h 81h	0	0	ID	DLC	Byte1	Index	Sub.	Byte5	Byte6	Byte7	Byte8	580h+ID	8	60h	12h 21h	01h	0	0	0	0				
ID	DLC	Byte1	Index	Sub.	Datentyp	Byte7	Byte8																																																																		
600h+ID	8	22h	12h 21h	01h	30h 71h	0	0																																																																		
ID	DLC	Byte1	Index	Sub.	Byte5	Byte6	Byte7	Byte8																																																																	
580h+ID	8	60h	12h 21h	01h	0	0	0	0																																																																	
ID	DLC	Byte1	Index	Sub.	Datentyp	Byte7	Byte8																																																																		
600h+ID	8	22h	12h 21h	01h	30h 81h	0	0																																																																		
ID	DLC	Byte1	Index	Sub.	Byte5	Byte6	Byte7	Byte8																																																																	
580h+ID	8	60h	12h 21h	01h	0	0	0	0																																																																	
<p>2. Einheit abfragen (Objekt 61'31h) 16-Bit</p> <p>Master → Slave</p> <table border="1"> <tr> <td>ID</td><td>DLC</td><td>Byte1</td><td>Index</td><td>Sub.</td><td>Byte5</td><td>Byte6</td><td>Byte7</td><td>Byte8</td> </tr> <tr> <td>600h+ID</td><td>8</td><td>40h</td><td>31h 61h</td><td>01h</td><td>0</td><td>0</td><td>0</td><td>0</td> </tr> </table> <p>← Slave →</p> <table border="1"> <tr> <td>ID</td><td>DLC</td><td>Byte1</td><td>Index</td><td>Sub.</td><td colspan="4">Einheit</td> </tr> <tr> <td>580h+ID</td><td>8</td><td>43h</td><td>31h 61h</td><td>01h</td><td>00h</td><td>01h</td><td>01h</td><td>FAh</td> </tr> </table> <p>FA'01'01'00h => µε oder µm/m</p>	ID	DLC	Byte1	Index	Sub.	Byte5	Byte6	Byte7	Byte8	600h+ID	8	40h	31h 61h	01h	0	0	0	0	ID	DLC	Byte1	Index	Sub.	Einheit				580h+ID	8	43h	31h 61h	01h	00h	01h	01h	FAh	<p>Master → Slave</p> <table border="1"> <tr> <td>ID</td><td>DLC</td><td>Byte1</td><td>Index</td><td>Sub.</td><td>Byte5</td><td>Byte6</td><td>Byte7</td><td>Byte8</td> </tr> <tr> <td>600h+ID</td><td>8</td><td>40h</td><td>31h 61h</td><td>01h</td><td>0</td><td>0</td><td>0</td><td>0</td> </tr> </table> <p>← Slave →</p> <table border="1"> <tr> <td>ID</td><td>DLC</td><td>Byte1</td><td>Index</td><td>Sub.</td><td colspan="4">Einheit</td> </tr> <tr> <td>580h+ID</td><td>8</td><td>43h</td><td>31h 61h</td><td>01h</td><td>00h</td><td>01h</td><td>01h</td><td>FAh</td> </tr> </table> <p>FA'01'01'00h => µε oder µm/m</p>	ID	DLC	Byte1	Index	Sub.	Byte5	Byte6	Byte7	Byte8	600h+ID	8	40h	31h 61h	01h	0	0	0	0	ID	DLC	Byte1	Index	Sub.	Einheit				580h+ID	8	43h	31h 61h	01h	00h	01h	01h	FAh
ID	DLC	Byte1	Index	Sub.	Byte5	Byte6	Byte7	Byte8																																																																	
600h+ID	8	40h	31h 61h	01h	0	0	0	0																																																																	
ID	DLC	Byte1	Index	Sub.	Einheit																																																																				
580h+ID	8	43h	31h 61h	01h	00h	01h	01h	FAh																																																																	
ID	DLC	Byte1	Index	Sub.	Byte5	Byte6	Byte7	Byte8																																																																	
600h+ID	8	40h	31h 61h	01h	0	0	0	0																																																																	
ID	DLC	Byte1	Index	Sub.	Einheit																																																																				
580h+ID	8	43h	31h 61h	01h	00h	01h	01h	FAh																																																																	
<p>3. Dezimalstellen abfragen (Objekt 61'32h) 16-Bit</p> <p>Master → Slave</p> <table border="1"> <tr> <td>ID</td><td>DLC</td><td>Byte1</td><td>Index</td><td>Sub.</td><td>Byte5</td><td>Byte6</td><td>Byte7</td><td>Byte8</td> </tr> <tr> <td>600h+ID</td><td>8</td><td>40h</td><td>32h 61h</td><td>01h</td><td>0</td><td>0</td><td>0</td><td>0</td> </tr> </table> <p>← Slave →</p> <table border="1"> <tr> <td>ID</td><td>DLC</td><td>Byte1</td><td>Index</td><td>Sub.</td><td>Dez.</td><td>Byte6</td><td>Byte7</td><td>Byte8</td> </tr> <tr> <td>580h+ID</td><td>8</td><td>4Fh</td><td>32h 61h</td><td>01h</td><td>02h</td><td>0</td><td>0</td><td>0</td> </tr> </table> <p>02h => 2 Dezimalstellen => Division mit 10²</p>	ID	DLC	Byte1	Index	Sub.	Byte5	Byte6	Byte7	Byte8	600h+ID	8	40h	32h 61h	01h	0	0	0	0	ID	DLC	Byte1	Index	Sub.	Dez.	Byte6	Byte7	Byte8	580h+ID	8	4Fh	32h 61h	01h	02h	0	0	0	<p>Master → Slave</p> <table border="1"> <tr> <td>ID</td><td>DLC</td><td>Byte1</td><td>Index</td><td>Sub.</td><td>Byte5</td><td>Byte6</td><td>Byte7</td><td>Byte8</td> </tr> <tr> <td>600h+ID</td><td>8</td><td>40h</td><td>32h 61h</td><td>01h</td><td>0</td><td>0</td><td>0</td><td>0</td> </tr> </table> <p>← Slave →</p> <table border="1"> <tr> <td>ID</td><td>DLC</td><td>Byte1</td><td>Index</td><td>Sub.</td><td>Dez.</td><td>Byte6</td><td>Byte7</td><td>Byte8</td> </tr> <tr> <td>580h+ID</td><td>8</td><td>4Fh</td><td>32h 61h</td><td>01h</td><td>02h</td><td>0</td><td>0</td><td>0</td> </tr> </table> <p>02h => 2 Dezimalstellen => Division mit 10²</p>	ID	DLC	Byte1	Index	Sub.	Byte5	Byte6	Byte7	Byte8	600h+ID	8	40h	32h 61h	01h	0	0	0	0	ID	DLC	Byte1	Index	Sub.	Dez.	Byte6	Byte7	Byte8	580h+ID	8	4Fh	32h 61h	01h	02h	0	0	0
ID	DLC	Byte1	Index	Sub.	Byte5	Byte6	Byte7	Byte8																																																																	
600h+ID	8	40h	32h 61h	01h	0	0	0	0																																																																	
ID	DLC	Byte1	Index	Sub.	Dez.	Byte6	Byte7	Byte8																																																																	
580h+ID	8	4Fh	32h 61h	01h	02h	0	0	0																																																																	
ID	DLC	Byte1	Index	Sub.	Byte5	Byte6	Byte7	Byte8																																																																	
600h+ID	8	40h	32h 61h	01h	0	0	0	0																																																																	
ID	DLC	Byte1	Index	Sub.	Dez.	Byte6	Byte7	Byte8																																																																	
580h+ID	8	4Fh	32h 61h	01h	02h	0	0	0																																																																	

Messzyklus:

Objekt 71'30h, positive Dehnung 16-Bit	Objekt 81'30h, positive Dehnung 24-Bit																																																																						
<p>Master → Slave</p> <table border="1"> <tr> <td>ID</td><td>DLC</td><td>Byte1</td><td>Index</td><td>Sub.</td><td>Byte5</td><td>Byte6</td><td>Byte7</td><td>Byte8</td> </tr> <tr> <td>600h+ID</td><td>8</td><td>40h</td><td>30h 71h</td><td>01h</td><td>0</td><td>0</td><td>0</td><td>0</td> </tr> </table> <p>← Slave →</p> <table border="1"> <tr> <td>ID</td><td>DLC</td><td>Byte1</td><td>Index</td><td>Sub.</td><td>Prozesswert</td><td>Byte7</td><td>Byte8</td> </tr> <tr> <td>580h+ID</td><td>8</td><td>4Bh</td><td>30h 71h</td><td>01h</td><td>2Ch 01h</td><td>0</td><td>0</td> </tr> </table> <p>01'2Ch => 300_{Dez} Dezimalpunkt verschieben: 300 / 10² = 3.00 Einheit hinzufügen: 3.00 => <u>3.00µε</u></p>	ID	DLC	Byte1	Index	Sub.	Byte5	Byte6	Byte7	Byte8	600h+ID	8	40h	30h 71h	01h	0	0	0	0	ID	DLC	Byte1	Index	Sub.	Prozesswert	Byte7	Byte8	580h+ID	8	4Bh	30h 71h	01h	2Ch 01h	0	0	<p>Master → Slave</p> <table border="1"> <tr> <td>ID</td><td>DLC</td><td>Byte1</td><td>Index</td><td>Sub.</td><td>Byte5</td><td>Byte6</td><td>Byte7</td><td>Byte8</td> </tr> <tr> <td>600h+ID</td><td>8</td><td>40h</td><td>30h 81h</td><td>01h</td><td>0</td><td>0</td><td>0</td><td>0</td> </tr> </table> <p>← Slave →</p> <table border="1"> <tr> <td>ID</td><td>DLC</td><td>Byte1</td><td>Index</td><td>Sub.</td><td colspan="3">Prozesswert</td><td>Byte8</td> </tr> <tr> <td>580h+ID</td><td>8</td><td>47h</td><td>30h 81h</td><td>01h</td><td>80h 38h 01h</td><td>0</td><td>0</td> </tr> </table> <p>01'38'80h => 80'000_{Dez} Dezimalpunkt verschieben: 80'000 / 10² = 800 Einheit hinzufügen: 800 => <u>800µε</u></p>	ID	DLC	Byte1	Index	Sub.	Byte5	Byte6	Byte7	Byte8	600h+ID	8	40h	30h 81h	01h	0	0	0	0	ID	DLC	Byte1	Index	Sub.	Prozesswert			Byte8	580h+ID	8	47h	30h 81h	01h	80h 38h 01h	0	0	
ID	DLC	Byte1	Index	Sub.	Byte5	Byte6	Byte7	Byte8																																																															
600h+ID	8	40h	30h 71h	01h	0	0	0	0																																																															
ID	DLC	Byte1	Index	Sub.	Prozesswert	Byte7	Byte8																																																																
580h+ID	8	4Bh	30h 71h	01h	2Ch 01h	0	0																																																																
ID	DLC	Byte1	Index	Sub.	Byte5	Byte6	Byte7	Byte8																																																															
600h+ID	8	40h	30h 81h	01h	0	0	0	0																																																															
ID	DLC	Byte1	Index	Sub.	Prozesswert			Byte8																																																															
580h+ID	8	47h	30h 81h	01h	80h 38h 01h	0	0																																																																
<p>Objekt 71'30h, negative Dehnung 16-Bit</p> <p>Master → Slave</p> <table border="1"> <tr> <td>ID</td><td>DLC</td><td>Byte1</td><td>Index</td><td>Sub.</td><td>Byte5</td><td>Byte6</td><td>Byte7</td><td>Byte8</td> </tr> <tr> <td>600h+ID</td><td>8</td><td>40h</td><td>30h 71h</td><td>01h</td><td>0</td><td>0</td><td>0</td><td>0</td> </tr> </table> <p>← Slave →</p> <table border="1"> <tr> <td>ID</td><td>DLC</td><td>Byte1</td><td>Index</td><td>Sub.</td><td colspan="3">Prozesswert</td><td>Byte8</td> </tr> <tr> <td>580h+ID</td><td>8</td><td>4Bh</td><td>30h 71h</td><td>01h</td><td>D4h FEh</td><td>0</td><td>0</td> </tr> </table> <p>FE'D4h => 1111'1110'1100'0100b $\overset{\text{inv.}}{\curvearrowright}$ - 0000'0001'0010'1011b $\overset{+1}{\curvearrowright}$ - 0000'0001'0010'1100b $\overset{\text{inv.}}{\curvearrowright}$ = -300_{Dez} Dezimalpunkt verschieben: -300 / 10² = -3.00 Einheit hinzufügen: -3.00 => <u>-3.00µε</u></p>	ID	DLC	Byte1	Index	Sub.	Byte5	Byte6	Byte7	Byte8	600h+ID	8	40h	30h 71h	01h	0	0	0	0	ID	DLC	Byte1	Index	Sub.	Prozesswert			Byte8	580h+ID	8	4Bh	30h 71h	01h	D4h FEh	0	0	<p>Objekt 81'30h, negative Dehnung 24-Bit</p> <p>Master → Slave</p> <table border="1"> <tr> <td>ID</td><td>DLC</td><td>Byte1</td><td>Index</td><td>Sub.</td><td>Byte5</td><td>Byte6</td><td>Byte7</td><td>Byte8</td> </tr> <tr> <td>600h+ID</td><td>8</td><td>40h</td><td>30h 81h</td><td>01h</td><td>0</td><td>0</td><td>0</td><td>0</td> </tr> </table> <p>← Slave →</p> <table border="1"> <tr> <td>ID</td><td>DLC</td><td>Byte1</td><td>Index</td><td>Sub.</td><td colspan="3">Prozesswert</td><td>Byte8</td> </tr> <tr> <td>580h+ID</td><td>8</td><td>47h</td><td>30h 81h</td><td>01h</td><td>80h C7h FEh</td><td>0</td><td>0</td> </tr> </table> <p>FE'C7'80h => 1111'1110'1100'0111'1000'0000b $\overset{\text{inv.}}{\curvearrowright}$ - 0000'0001'0011'1000'0111'1111b $\overset{+1}{\curvearrowright}$ - 0000'0001'0011'1000'1000'0000b $\overset{\text{inv.}}{\curvearrowright}$ = -80'000_{Dez} Dezimalpunkt verschieben: -80'000 / 10² = -800 Einheit hinzufügen: -800 => <u>-800µε</u></p>	ID	DLC	Byte1	Index	Sub.	Byte5	Byte6	Byte7	Byte8	600h+ID	8	40h	30h 81h	01h	0	0	0	0	ID	DLC	Byte1	Index	Sub.	Prozesswert			Byte8	580h+ID	8	47h	30h 81h	01h	80h C7h FEh	0	0
ID	DLC	Byte1	Index	Sub.	Byte5	Byte6	Byte7	Byte8																																																															
600h+ID	8	40h	30h 71h	01h	0	0	0	0																																																															
ID	DLC	Byte1	Index	Sub.	Prozesswert			Byte8																																																															
580h+ID	8	4Bh	30h 71h	01h	D4h FEh	0	0																																																																
ID	DLC	Byte1	Index	Sub.	Byte5	Byte6	Byte7	Byte8																																																															
600h+ID	8	40h	30h 81h	01h	0	0	0	0																																																															
ID	DLC	Byte1	Index	Sub.	Prozesswert			Byte8																																																															
580h+ID	8	47h	30h 81h	01h	80h C7h FEh	0	0																																																																

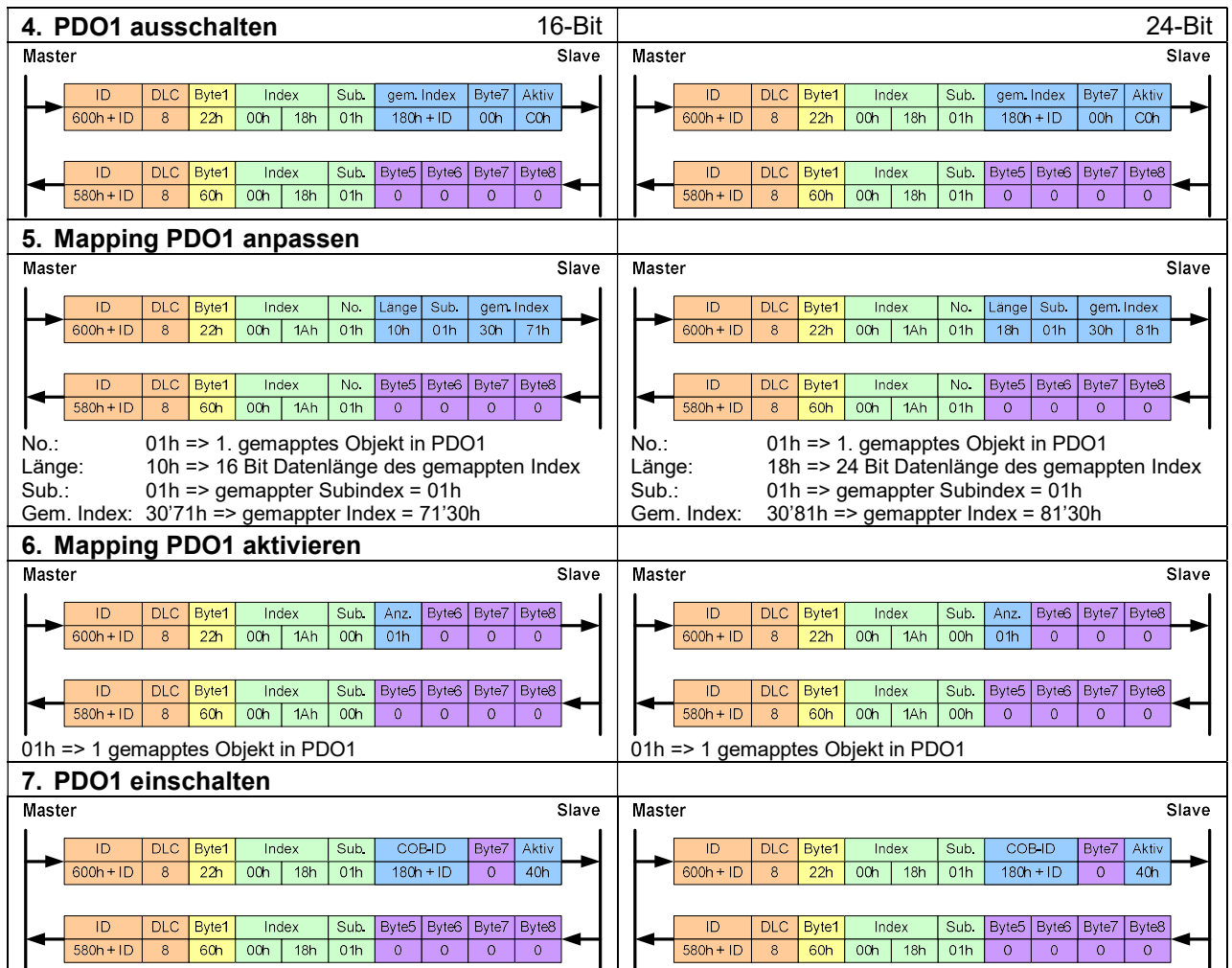
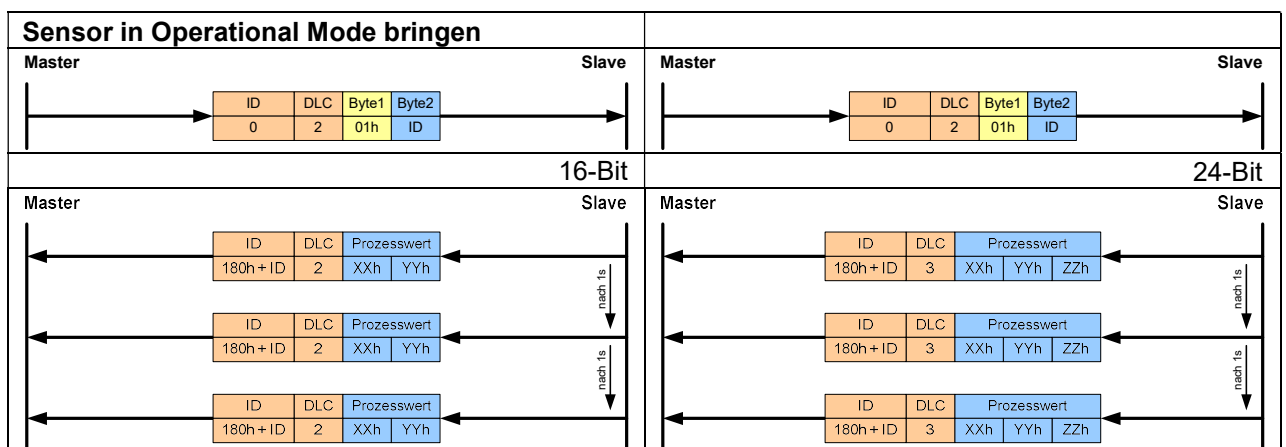
8.3 Konfigurieren des Prozesswerts über PDO1 (16 und 24bit)

Messinitialisierung:

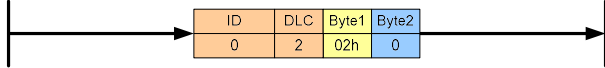
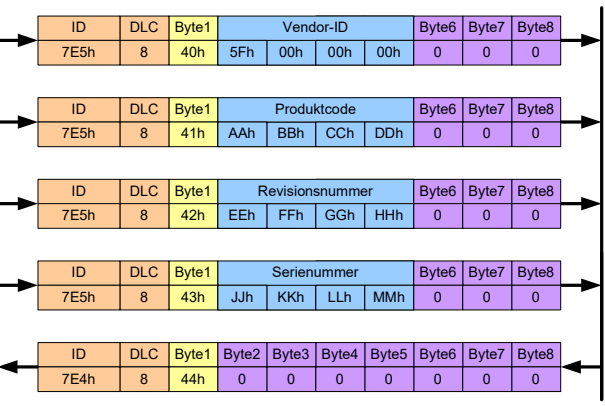
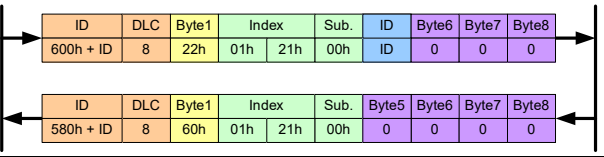
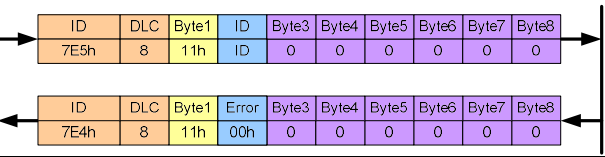
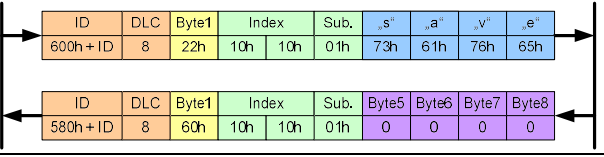
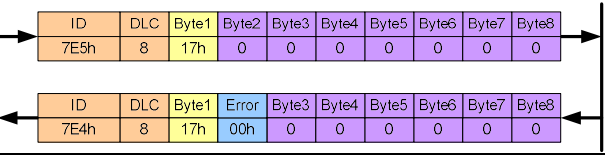
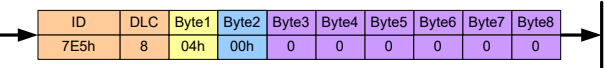
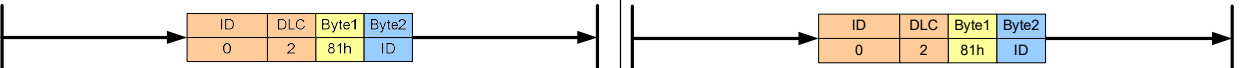
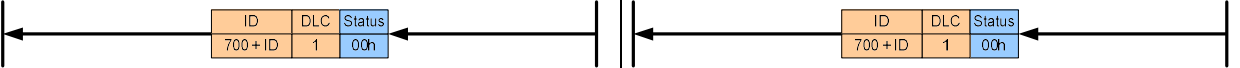
1. Datentyp einstellen (Objekt 21'12h) 16-Bit	24-Bit																																																																								
<p>Master → Slave</p> <table border="1"> <tr> <td>ID</td><td>DLC</td><td>Byte1</td><td>Index</td><td>Sub.</td><td>Datentyp</td><td>Byte7</td><td>Byte8</td> </tr> <tr> <td>600h+ID</td><td>8</td><td>22h</td><td>12h 21h</td><td>01h</td><td>30h 71h</td><td>0</td><td>0</td> </tr> </table> <p>← Slave</p> <table border="1"> <tr> <td>ID</td><td>DLC</td><td>Byte1</td><td>Index</td><td>Sub.</td><td>Byte5</td><td>Byte6</td><td>Byte7</td><td>Byte8</td> </tr> <tr> <td>580h+ID</td><td>8</td><td>60h</td><td>12h 21h</td><td>01h</td><td>0</td><td>0</td><td>0</td><td>0</td> </tr> </table> <p>71'30h => 16-Bit-Prozesswert</p>	ID	DLC	Byte1	Index	Sub.	Datentyp	Byte7	Byte8	600h+ID	8	22h	12h 21h	01h	30h 71h	0	0	ID	DLC	Byte1	Index	Sub.	Byte5	Byte6	Byte7	Byte8	580h+ID	8	60h	12h 21h	01h	0	0	0	0	<p>Master → Slave</p> <table border="1"> <tr> <td>ID</td><td>DLC</td><td>Byte1</td><td>Index</td><td>Sub.</td><td>Datentyp</td><td>Byte7</td><td>Byte8</td> </tr> <tr> <td>600h+ID</td><td>8</td><td>22h</td><td>12h 21h</td><td>01h</td><td>30h 81h</td><td>0</td><td>0</td> </tr> </table> <p>← Slave</p> <table border="1"> <tr> <td>ID</td><td>DLC</td><td>Byte1</td><td>Index</td><td>Sub.</td><td>Byte5</td><td>Byte6</td><td>Byte7</td><td>Byte8</td> </tr> <tr> <td>580h+ID</td><td>8</td><td>60h</td><td>12h 21h</td><td>01h</td><td>0</td><td>0</td><td>0</td><td>0</td> </tr> </table> <p>81'30h => 24-Bit-Prozesswert</p>	ID	DLC	Byte1	Index	Sub.	Datentyp	Byte7	Byte8	600h+ID	8	22h	12h 21h	01h	30h 81h	0	0	ID	DLC	Byte1	Index	Sub.	Byte5	Byte6	Byte7	Byte8	580h+ID	8	60h	12h 21h	01h	0	0	0	0				
ID	DLC	Byte1	Index	Sub.	Datentyp	Byte7	Byte8																																																																		
600h+ID	8	22h	12h 21h	01h	30h 71h	0	0																																																																		
ID	DLC	Byte1	Index	Sub.	Byte5	Byte6	Byte7	Byte8																																																																	
580h+ID	8	60h	12h 21h	01h	0	0	0	0																																																																	
ID	DLC	Byte1	Index	Sub.	Datentyp	Byte7	Byte8																																																																		
600h+ID	8	22h	12h 21h	01h	30h 81h	0	0																																																																		
ID	DLC	Byte1	Index	Sub.	Byte5	Byte6	Byte7	Byte8																																																																	
580h+ID	8	60h	12h 21h	01h	0	0	0	0																																																																	
<p>2. Einheit abfragen (Objekt 61'31h)</p> <p>Master → Slave</p> <table border="1"> <tr> <td>ID</td><td>DLC</td><td>Byte1</td><td>Index</td><td>Sub.</td><td>Byte5</td><td>Byte6</td><td>Byte7</td><td>Byte8</td> </tr> <tr> <td>600h+ID</td><td>8</td><td>40h</td><td>31h 61h</td><td>01h</td><td>0</td><td>0</td><td>0</td><td>0</td> </tr> </table> <p>← Slave</p> <table border="1"> <tr> <td>ID</td><td>DLC</td><td>Byte1</td><td>Index</td><td>Sub.</td><td colspan="4">Einheit</td> </tr> <tr> <td>580h+ID</td><td>8</td><td>43h</td><td>31h 61h</td><td>01h</td><td>00h</td><td>01h</td><td>01h</td><td>FAh</td> </tr> </table> <p>FA'01'01'00h => µε oder µm/m</p>	ID	DLC	Byte1	Index	Sub.	Byte5	Byte6	Byte7	Byte8	600h+ID	8	40h	31h 61h	01h	0	0	0	0	ID	DLC	Byte1	Index	Sub.	Einheit				580h+ID	8	43h	31h 61h	01h	00h	01h	01h	FAh	<p>Master → Slave</p> <table border="1"> <tr> <td>ID</td><td>DLC</td><td>Byte1</td><td>Index</td><td>Sub.</td><td>Byte5</td><td>Byte6</td><td>Byte7</td><td>Byte8</td> </tr> <tr> <td>600h+ID</td><td>8</td><td>40h</td><td>31h 61h</td><td>01h</td><td>0</td><td>0</td><td>0</td><td>0</td> </tr> </table> <p>← Slave</p> <table border="1"> <tr> <td>ID</td><td>DLC</td><td>Byte1</td><td>Index</td><td>Sub.</td><td colspan="4">Einheit</td> </tr> <tr> <td>580h+ID</td><td>8</td><td>43h</td><td>31h 61h</td><td>01h</td><td>00h</td><td>01h</td><td>01h</td><td>FAh</td> </tr> </table> <p>FA'01'01'00h => µε oder µm/m</p>	ID	DLC	Byte1	Index	Sub.	Byte5	Byte6	Byte7	Byte8	600h+ID	8	40h	31h 61h	01h	0	0	0	0	ID	DLC	Byte1	Index	Sub.	Einheit				580h+ID	8	43h	31h 61h	01h	00h	01h	01h	FAh
ID	DLC	Byte1	Index	Sub.	Byte5	Byte6	Byte7	Byte8																																																																	
600h+ID	8	40h	31h 61h	01h	0	0	0	0																																																																	
ID	DLC	Byte1	Index	Sub.	Einheit																																																																				
580h+ID	8	43h	31h 61h	01h	00h	01h	01h	FAh																																																																	
ID	DLC	Byte1	Index	Sub.	Byte5	Byte6	Byte7	Byte8																																																																	
600h+ID	8	40h	31h 61h	01h	0	0	0	0																																																																	
ID	DLC	Byte1	Index	Sub.	Einheit																																																																				
580h+ID	8	43h	31h 61h	01h	00h	01h	01h	FAh																																																																	
<p>3. Dezimalstellen abfragen (Objekt 61'32h)</p> <p>Master → Slave</p> <table border="1"> <tr> <td>ID</td><td>DLC</td><td>Byte1</td><td>Index</td><td>Sub.</td><td>Byte5</td><td>Byte6</td><td>Byte7</td><td>Byte8</td> </tr> <tr> <td>600h+ID</td><td>8</td><td>40h</td><td>32h 61h</td><td>01h</td><td>0</td><td>0</td><td>0</td><td>0</td> </tr> </table> <p>← Slave</p> <table border="1"> <tr> <td>ID</td><td>DLC</td><td>Byte1</td><td>Index</td><td>Sub.</td><td>Dez.</td><td>Byte6</td><td>Byte7</td><td>Byte8</td> </tr> <tr> <td>580h+ID</td><td>8</td><td>4Fh</td><td>32h 61h</td><td>01h</td><td>02h</td><td>0</td><td>0</td><td>0</td> </tr> </table> <p>02h => 2 Dezimalstellen => Division mit 10²</p>	ID	DLC	Byte1	Index	Sub.	Byte5	Byte6	Byte7	Byte8	600h+ID	8	40h	32h 61h	01h	0	0	0	0	ID	DLC	Byte1	Index	Sub.	Dez.	Byte6	Byte7	Byte8	580h+ID	8	4Fh	32h 61h	01h	02h	0	0	0	<p>Master → Slave</p> <table border="1"> <tr> <td>ID</td><td>DLC</td><td>Byte1</td><td>Index</td><td>Sub.</td><td>Byte5</td><td>Byte6</td><td>Byte7</td><td>Byte8</td> </tr> <tr> <td>600h+ID</td><td>8</td><td>40h</td><td>32h 61h</td><td>01h</td><td>0</td><td>0</td><td>0</td><td>0</td> </tr> </table> <p>← Slave</p> <table border="1"> <tr> <td>ID</td><td>DLC</td><td>Byte1</td><td>Index</td><td>Sub.</td><td>Dez.</td><td>Byte6</td><td>Byte7</td><td>Byte8</td> </tr> <tr> <td>580h+ID</td><td>8</td><td>4Fh</td><td>32h 61h</td><td>01h</td><td>02h</td><td>0</td><td>0</td><td>0</td> </tr> </table> <p>02h => 2 Dezimalstellen => Division mit 10²</p>	ID	DLC	Byte1	Index	Sub.	Byte5	Byte6	Byte7	Byte8	600h+ID	8	40h	32h 61h	01h	0	0	0	0	ID	DLC	Byte1	Index	Sub.	Dez.	Byte6	Byte7	Byte8	580h+ID	8	4Fh	32h 61h	01h	02h	0	0	0
ID	DLC	Byte1	Index	Sub.	Byte5	Byte6	Byte7	Byte8																																																																	
600h+ID	8	40h	32h 61h	01h	0	0	0	0																																																																	
ID	DLC	Byte1	Index	Sub.	Dez.	Byte6	Byte7	Byte8																																																																	
580h+ID	8	4Fh	32h 61h	01h	02h	0	0	0																																																																	
ID	DLC	Byte1	Index	Sub.	Byte5	Byte6	Byte7	Byte8																																																																	
600h+ID	8	40h	32h 61h	01h	0	0	0	0																																																																	
ID	DLC	Byte1	Index	Sub.	Dez.	Byte6	Byte7	Byte8																																																																	
580h+ID	8	4Fh	32h 61h	01h	02h	0	0	0																																																																	

Konfiguration PDO Einstellungen:

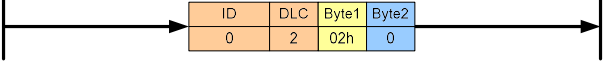
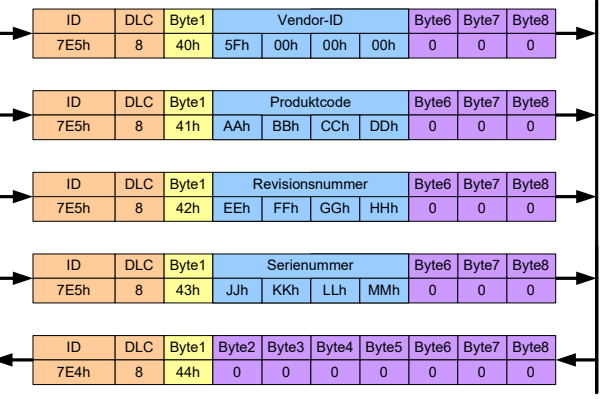
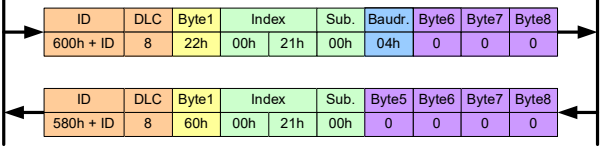
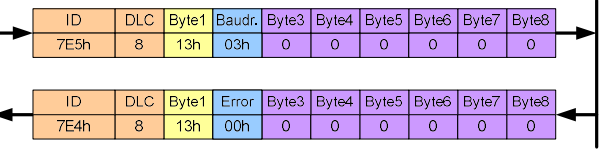
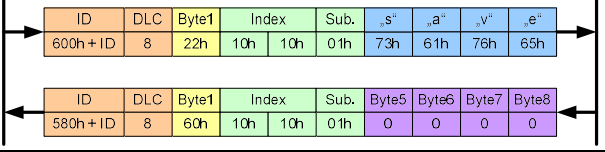
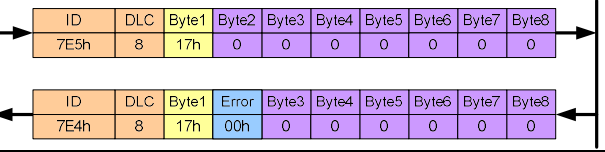
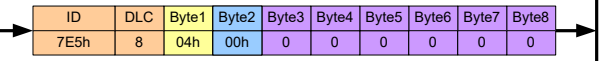

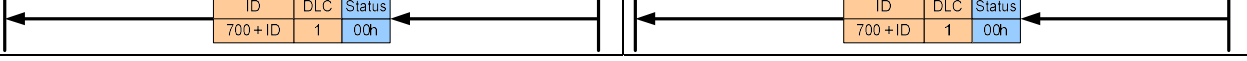
1. PDO1 Transmission Type einstellen 16-Bit	24-Bit																																																																								
<p>Master → Slave</p> <table border="1"> <tr> <td>ID</td><td>DLC</td><td>Byte1</td><td>Index</td><td>Sub.</td><td>Type</td><td>Byte6</td><td>Byte7</td><td>Byte8</td> </tr> <tr> <td>600h+ID</td><td>8</td><td>22h</td><td>00h 18h</td><td>02h</td><td>FFh</td><td>0</td><td>0</td><td>0</td> </tr> </table> <p>← Slave</p> <table border="1"> <tr> <td>ID</td><td>DLC</td><td>Byte1</td><td>Index</td><td>Sub.</td><td>Byte5</td><td>Byte6</td><td>Byte7</td><td>Byte8</td> </tr> <tr> <td>580h+ID</td><td>8</td><td>60h</td><td>00h 18h</td><td>02h</td><td>0</td><td>0</td><td>0</td><td>0</td> </tr> </table> <p>FFh => ausgelöst nach Ablauf der Event Time</p>	ID	DLC	Byte1	Index	Sub.	Type	Byte6	Byte7	Byte8	600h+ID	8	22h	00h 18h	02h	FFh	0	0	0	ID	DLC	Byte1	Index	Sub.	Byte5	Byte6	Byte7	Byte8	580h+ID	8	60h	00h 18h	02h	0	0	0	0	<p>Master → Slave</p> <table border="1"> <tr> <td>ID</td><td>DLC</td><td>Byte1</td><td>Index</td><td>Sub.</td><td>Type</td><td>Byte6</td><td>Byte7</td><td>Byte8</td> </tr> <tr> <td>600h+ID</td><td>8</td><td>22h</td><td>00h 18h</td><td>02h</td><td>FFh</td><td>0</td><td>0</td><td>0</td> </tr> </table> <p>← Slave</p> <table border="1"> <tr> <td>ID</td><td>DLC</td><td>Byte1</td><td>Index</td><td>Sub.</td><td>Byte5</td><td>Byte6</td><td>Byte7</td><td>Byte8</td> </tr> <tr> <td>580h+ID</td><td>8</td><td>60h</td><td>00h 18h</td><td>02h</td><td>0</td><td>0</td><td>0</td><td>0</td> </tr> </table> <p>FFh => ausgelöst nach Ablauf der Event Time</p>	ID	DLC	Byte1	Index	Sub.	Type	Byte6	Byte7	Byte8	600h+ID	8	22h	00h 18h	02h	FFh	0	0	0	ID	DLC	Byte1	Index	Sub.	Byte5	Byte6	Byte7	Byte8	580h+ID	8	60h	00h 18h	02h	0	0	0	0
ID	DLC	Byte1	Index	Sub.	Type	Byte6	Byte7	Byte8																																																																	
600h+ID	8	22h	00h 18h	02h	FFh	0	0	0																																																																	
ID	DLC	Byte1	Index	Sub.	Byte5	Byte6	Byte7	Byte8																																																																	
580h+ID	8	60h	00h 18h	02h	0	0	0	0																																																																	
ID	DLC	Byte1	Index	Sub.	Type	Byte6	Byte7	Byte8																																																																	
600h+ID	8	22h	00h 18h	02h	FFh	0	0	0																																																																	
ID	DLC	Byte1	Index	Sub.	Byte5	Byte6	Byte7	Byte8																																																																	
580h+ID	8	60h	00h 18h	02h	0	0	0	0																																																																	
<p>2. PDO1 Event Time einstellen (1sec)</p> <p>Master → Slave</p> <table border="1"> <tr> <td>ID</td><td>DLC</td><td>Byte1</td><td>Index</td><td>Sub.</td><td>Event Time</td><td>Byte7</td><td>Byte8</td> </tr> <tr> <td>600h+ID</td><td>8</td><td>22h</td><td>00h 18h</td><td>05h</td><td>E8h 03h</td><td>0</td><td>0</td> </tr> </table> <p>← Slave</p> <table border="1"> <tr> <td>ID</td><td>DLC</td><td>Byte1</td><td>Index</td><td>Sub.</td><td>Byte5</td><td>Byte6</td><td>Byte7</td><td>Byte8</td> </tr> <tr> <td>580h+ID</td><td>8</td><td>60h</td><td>00h 18h</td><td>05h</td><td>0</td><td>0</td><td>0</td><td>0</td> </tr> </table> <p>03'E8h => 1000_{Dez} => Event Time = 1000ms</p>	ID	DLC	Byte1	Index	Sub.	Event Time	Byte7	Byte8	600h+ID	8	22h	00h 18h	05h	E8h 03h	0	0	ID	DLC	Byte1	Index	Sub.	Byte5	Byte6	Byte7	Byte8	580h+ID	8	60h	00h 18h	05h	0	0	0	0	<p>Master → Slave</p> <table border="1"> <tr> <td>ID</td><td>DLC</td><td>Byte1</td><td>Index</td><td>Sub.</td><td>Event Time</td><td>Byte7</td><td>Byte8</td> </tr> <tr> <td>600h+ID</td><td>8</td><td>22h</td><td>00h 18h</td><td>05h</td><td>E8h 03h</td><td>0</td><td>0</td> </tr> </table> <p>← Slave</p> <table border="1"> <tr> <td>ID</td><td>DLC</td><td>Byte1</td><td>Index</td><td>Sub.</td><td>Byte5</td><td>Byte6</td><td>Byte7</td><td>Byte8</td> </tr> <tr> <td>580h+ID</td><td>8</td><td>60h</td><td>00h 18h</td><td>05h</td><td>0</td><td>0</td><td>0</td><td>0</td> </tr> </table> <p>03'E8h => 1000_{Dez} => Event Time = 1000ms</p>	ID	DLC	Byte1	Index	Sub.	Event Time	Byte7	Byte8	600h+ID	8	22h	00h 18h	05h	E8h 03h	0	0	ID	DLC	Byte1	Index	Sub.	Byte5	Byte6	Byte7	Byte8	580h+ID	8	60h	00h 18h	05h	0	0	0	0				
ID	DLC	Byte1	Index	Sub.	Event Time	Byte7	Byte8																																																																		
600h+ID	8	22h	00h 18h	05h	E8h 03h	0	0																																																																		
ID	DLC	Byte1	Index	Sub.	Byte5	Byte6	Byte7	Byte8																																																																	
580h+ID	8	60h	00h 18h	05h	0	0	0	0																																																																	
ID	DLC	Byte1	Index	Sub.	Event Time	Byte7	Byte8																																																																		
600h+ID	8	22h	00h 18h	05h	E8h 03h	0	0																																																																		
ID	DLC	Byte1	Index	Sub.	Byte5	Byte6	Byte7	Byte8																																																																	
580h+ID	8	60h	00h 18h	05h	0	0	0	0																																																																	
<p>3. Mapping PDO1 deaktivieren</p> <p>Master → Slave</p> <table border="1"> <tr> <td>ID</td><td>DLC</td><td>Byte1</td><td>Index</td><td>Sub.</td><td>Anz.</td><td>Byte6</td><td>Byte7</td><td>Byte8</td> </tr> <tr> <td>600h+ID</td><td>8</td><td>22h</td><td>00h 1Ah</td><td>00h</td><td>00h</td><td>0</td><td>0</td><td>0</td> </tr> </table> <p>← Slave</p> <table border="1"> <tr> <td>ID</td><td>DLC</td><td>Byte1</td><td>Index</td><td>Sub.</td><td>Byte5</td><td>Byte6</td><td>Byte7</td><td>Byte8</td> </tr> <tr> <td>580h+ID</td><td>8</td><td>60h</td><td>00h 1Ah</td><td>00h</td><td>0</td><td>0</td><td>0</td><td>0</td> </tr> </table>	ID	DLC	Byte1	Index	Sub.	Anz.	Byte6	Byte7	Byte8	600h+ID	8	22h	00h 1Ah	00h	00h	0	0	0	ID	DLC	Byte1	Index	Sub.	Byte5	Byte6	Byte7	Byte8	580h+ID	8	60h	00h 1Ah	00h	0	0	0	0	<p>Master → Slave</p> <table border="1"> <tr> <td>ID</td><td>DLC</td><td>Byte1</td><td>Index</td><td>Sub.</td><td>Anz.</td><td>Byte6</td><td>Byte7</td><td>Byte8</td> </tr> <tr> <td>600h+ID</td><td>8</td><td>22h</td><td>00h 1Ah</td><td>00h</td><td>00h</td><td>0</td><td>0</td><td>0</td> </tr> </table> <p>← Slave</p> <table border="1"> <tr> <td>ID</td><td>DLC</td><td>Byte1</td><td>Index</td><td>Sub.</td><td>Byte5</td><td>Byte6</td><td>Byte7</td><td>Byte8</td> </tr> <tr> <td>580h+ID</td><td>8</td><td>60h</td><td>00h 1Ah</td><td>00h</td><td>0</td><td>0</td><td>0</td><td>0</td> </tr> </table>	ID	DLC	Byte1	Index	Sub.	Anz.	Byte6	Byte7	Byte8	600h+ID	8	22h	00h 1Ah	00h	00h	0	0	0	ID	DLC	Byte1	Index	Sub.	Byte5	Byte6	Byte7	Byte8	580h+ID	8	60h	00h 1Ah	00h	0	0	0	0
ID	DLC	Byte1	Index	Sub.	Anz.	Byte6	Byte7	Byte8																																																																	
600h+ID	8	22h	00h 1Ah	00h	00h	0	0	0																																																																	
ID	DLC	Byte1	Index	Sub.	Byte5	Byte6	Byte7	Byte8																																																																	
580h+ID	8	60h	00h 1Ah	00h	0	0	0	0																																																																	
ID	DLC	Byte1	Index	Sub.	Anz.	Byte6	Byte7	Byte8																																																																	
600h+ID	8	22h	00h 1Ah	00h	00h	0	0	0																																																																	
ID	DLC	Byte1	Index	Sub.	Byte5	Byte6	Byte7	Byte8																																																																	
580h+ID	8	60h	00h 1Ah	00h	0	0	0	0																																																																	


Abfrage des Prozesswerts:


8.4 ID Ändern (Objekt 2101 oder LSS)

Bus in Stop Mode	Obj. 21'01h	LSS mit Sensor im Netzwerk	
nicht nötig		Master	Slave
			
Sensor in Konfigurationsmodus		Master	Slave
nicht nötig		 <p>Produktcode, Revisions- und Seriennummer sich auf dem Sensor ersichtlich und können im Objekt 1018h ausgelesen werden</p>	
ID ändern		Master	Slave
		Master	Slave
		 	
Änderungen speichern		Master	Slave
		Master	Slave
		 	
Konfigurationsmodus verlassen		Master	Slave
nicht nötig			
Sensor resettieren oder neu aufstarten => ACHTUNG: Alte ID verwenden		Master	Slave
		Master	Slave
			
Sensor arbeitet mit neuer ID			
Bootup Message mit neuer ID			
		Master	Slave
		Master	Slave
			

8.5 Baudrate ändern (Objekt 21'00h oder LSS)

Bus in Stop Mode	Obj. 21'00h	LSS mit Sensor im Netzwerk	
nicht nötig		Master	Slave
			
Sensor in Konfigurationsmodus		Master	Slave
nicht nötig		 <p>Produktcode, Revisions- und Seriennummer sich auf dem Sensor ersichtlich und können im Objekt 1018h ausgelesen werden</p>	
Baudrate ändern (auf 250kBaud)		Master	Slave
		Master	Slave
		 <p>Baudr.: 04h => 250kBaud</p>	
		 <p>Baudr.: 03h => 250kBaud</p>	
Änderungen speichern		Master	Slave
		Master	Slave
			
			
Konfigurationsmodus verlassen		Master	Slave
nicht nötig		Master	Slave
			
Weitere Busteilnehmer auf neue Baudrate konfigurieren			
Bus resettieren oder neu aufstarten			
=> ACHTUNG: Startet mit neuer Baudrate			
Master		Slave	Master
			
Sensor kommuniziert mit neuer Baudrate			
Bootup Message mit neuer Baudrate			
Master		Slave	Master
			

9 Dokument-Revisions-History

- V2.10 Manual vor Software Version 2.00 & Revisionsnummer 30200h
- V3.00 Manual ab Software Version 2.00 & Revisionsnummer 30200h
Angepasst an überarbeitete CANopen Kommunikation
Implementieren der Neuerungen der Software Version 2.00
- LSS Dienst
 - Kommunikationsüberarbeitung
 - IIR Filter implementiert
- V3.01 Manual ab Software Version 2.03 & Revisionsnummer 30203h
Angepassen der Emergency Message an überlaufenden Wertebereich
- V3.02 Manual ab Software Version 2.04 & Revisionsnummer 30204h
Erweiterte PDO Funktionalität (RPDO1, TPDO2, TPDO3)
Erweiterter Objekt 2003, 2004 für Receive PDO Mapping und Antwort von Autozero über Transmit PDO
- V3.03 Manual ab Software Version 2.05 & Revisionsnummer 30205h
Ergänzen von 16/24bit Datentyp. Erweiterung mit Objekt 8130, 8133, 2112
- V3.04 Manual ab Software Version 2.07 & Revisionsnummer 30207h
Anpassungen an kleine Softwareänderungen
- V3.05 Manual ab Software Version 2.07 & Revisionsnummer 30207h
Einfügen einer CANopen Einführung und einer Objekt Tabelle
- V3.06 Manual ab Software Version 2.07 & Revisionsnummer 30207h
Bereinigen von „save“ Objekt in Anwenderbeispielen, ergänzen von Datenlänge in Mapping
- V3.07 Manual sprachlich für bestehenden DSRT 22DJ und neuen DST55R C1 angepasst.