Baumer

Passion for Sensors

# Bedienungsanleitung
# CANopen Drucksensor




## PDxJ xxxx.xxx.xxxx
## CANopen DS 404

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

Änderungen in Technik und Design vorbehalten.

Ver. 3.11

**1 Allgemeines ..... 3**
1.1 Sicherheitshinweise ..... 3
1.2 Montage und Inbetriebnahme ..... 3

**2 Projektierung ..... 3**
2.1 Maximaler Systemausbau ..... 3

**3 Anschlüsse ..... 3**
3.1 Elektrischer Anschluss ..... 3
3.1.1 Anschlussbelegung ..... 3
3.1.2 Anschluss-Skizze ..... 3
3.2 Potentialverhältnisse ..... 3
3.3 EMV-gerechte Verdrahtung ..... 3
3.3.1 Erdung inaktiver Metallteile ..... 3
3.3.2 Schirmung von Leitungen ..... 3
3.4 Spezifikation der CAN-Leitungen ..... 3

**4 CANopen ..... 3**
4.1 Einleitung ..... 3
4.2 Signale, Aufbau und Bus Topologie ..... 3
4.2.1 Bus-Signale ..... 3
4.2.2 Netzwerk-Topologie ..... 3
4.2.3 Aufbau einer CAN Nachricht ..... 3
4.2.4 Bitweise Busarbitrierung ..... 3
4.2.5 Prioritätsorientierte Nachrichtenübertragung ..... 3
4.2.6 Identifier-Verteilung ..... 3
4.3 Objekte ..... 3
4.4 Kommunikations-Mechanismen ..... 3
4.4.1 Process Data Objects (PDO) ..... 3
4.4.2 Service Data Objects (SDO) ..... 3
4.4.3 Network Management (NMT) ..... 3
4.4.4 Emergency (EMGY) ..... 3
4.4.5 Node-Guarding und Hearbeat ..... 3
4.5 Weitere Begriffserklärung ..... 3
4.5.1 Boot-Up Nachricht ..... 3
4.5.2 EDS ..... 3
4.5.3 DCF ..... 3
4.5.4 LSS ..... 3

**5 CANopen Protokoll ..... 3**
5.1 Allgemeines ..... 3
5.1.1 Bootloader ..... 3
5.2 Network Management ..... 3
5.2.1 Predefined Connection Set ..... 3
5.2.2 Startprozedere ..... 3
5.2.3 Start Node ..... 3
5.2.4 Stop Node ..... 3
5.2.5 Pre-Operational Node ..... 3
5.2.6 Reset Node ..... 3
5.3 Übersicht der unterstützten Objekte ..... 3
5.4 SDO-Struktur ..... 3

**6 Beschreibung der Objekte ..... 3**
6.1 Standard Objekte ..... 3
6.1.1 Geräteprofil ..... 3

6.1.2 Kalibrationsdatum ... 3
6.1.3 Gerätebezeichnung ... 3
6.1.4 Hardware ... 3
6.1.5 Software ... 3
6.1.6 Geräte Identität ... 3
6.2 Parameter Handling (save, load default) ... 3
6.2.1 Speichern ... 3
6.2.2 Defaultwerte laden ... 3
6.3 Gerätespezifische Objekte ... 3
6.3.1 Sensortyp ... 3
6.3.2 Betriebsart ... 3
6.3.3 Einheit Prozesswert ... 3
6.3.4 Dezimalstellen Prozesswert ... 3
6.3.5 Status der Messung ... 3
6.3.6 Abfrage Prozesswert 16bit ... 3
6.3.7 Delta-Prozesswert ... 3
6.4 Herstellerspezifische Objekte ... 3
6.4.1 Mittelungszeit ... 3
6.4.2 Baudrate ... 3
6.4.3 Identifikation ... 3
6.5 PDO-Kommunikation Objekte ... 3
6.5.1 Sende PDO 1 ... 3
6.5.2 Sende PDO 2 ... 3
6.5.3 Sende PDO 1 Mapping Parameter ... 3
6.5.4 Sende PDO 2 Mapping ... 3
6.5.5 Sync ID ... 3

**7 Fehlermeldungen und Dienste ... 3**
7.1 Error Register & History ... 3
7.1.1 Error Register ... 3
7.1.2 Emergency History ... 3
7.2 SDO-Fehlermeldungen ... 3
7.3 Emergency Messages ... 3
7.4 Heartbeat ... 3
7.4.1 Heartbeat Time ... 3
7.5 LSS (Layer Setting Services) ... 3
7.5.1 Aufgedruckte LSS-Daten auf Sensor ... 3
7.5.2 Ansprechen des Sensors über LSS ... 3
7.5.3 Konfigurationsmodus direkte Verbindung (Master-Sensor) ... 3
7.5.4 Konfigurationsmodus von einem Sensor in einem Netzwerk ... 3
7.5.5 Verstellen der ID und der Baudrate ... 3
7.5.6 Speichern der Änderungen ... 3
7.5.7 LSS Mode verlassen ... 3

**8 Anwenderbeispiele CANopen Protokoll ... 3**
8.1 Abfragen eines Prozesswerts über SDO ... 3
8.2 Konfigurieren des Prozesswerts über PDO1 ... 3
8.3 ID Ändern (Objekt 2101 oder LSS) ... 3
8.4 Baudrate ändern (Objekt 21'00h oder LSS) ... 3

**9 Dokument-Revisions-History ... 3**

# 1 Allgemeines

Dieses Manual enthält wichtige Informationen für den sicheren und bestimmungsgemässen Gebrauch des CANopen Drucksensors 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 Drucksensors.

## 1.1 Sicherheitshinweise

- Der Drucksensor ist ein kompaktes und hochsensibles Präzisionsmessgerät. Er dient ausschliesslich zur Erfassung von Druck und Temperatur, der Aufbereitung und Bereitstellung der Messwerte als CANopen Signale für das Folgegerät. Der Drucksensor 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 Drucksensors 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.
- Der Drucksensor darf nicht ausserhalb der Spezifikationen betrieben werden (siehe Datenblatt).
- Keine mechanischen oder elektrischen Veränderungen am Sensor vornehmen.
- Trotz robuster Bauweise, sollten die Drucksensoren keinen harten Stössen ausgesetzt werden.
- Statische und dynamische Überdruck, 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 Montageanleitung MAL 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 Drucksensors. Gerät und Zuleitung getrennt und in grossem Abstand zu Leitungen mit hohen Störpegeln verlegen.
- Den Drucksensor 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 Drucksensor 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 |

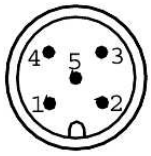
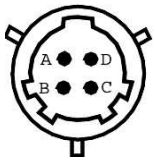
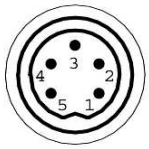
Maximale Gesamt-Busleitungslänge (mit 120Ohm Abschlusswiderstand) und maximale Stichleitungslänge (ohne Abschlusswiderstand) in Abhängigkeit von der Baudrate.

# 3 Anschlüsse

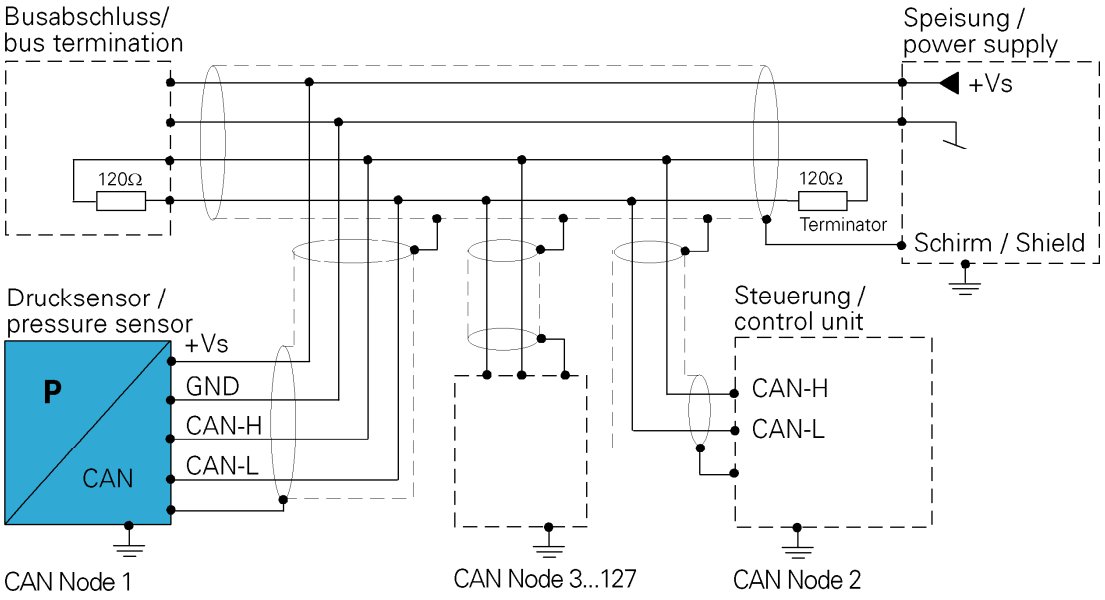
## 3.1 Elektrischer Anschluss

Schliessen Sie den Drucksensor 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

| | |
|---|---|
|  | **Stecker 14C**<br>Pin 1 n.c.<br>Pin 2 +Vs<br>Pin 3 GND<br>Pin 4 CANH<br>Pin 5 CANL<br>Gehäuse Schirm |
|  | **Stecker S40**<br>Pin A CANH<br>Pin B +Vs<br>Pin C CANL<br>Pin D GND<br>Gehäuse Schirm |
|  | **Stecker S50**<br>Pin 1 n.c.<br>Pin 2 +Vs<br>Pin 3 GND<br>Pin 4 CANH<br>Pin 5 CANL<br>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

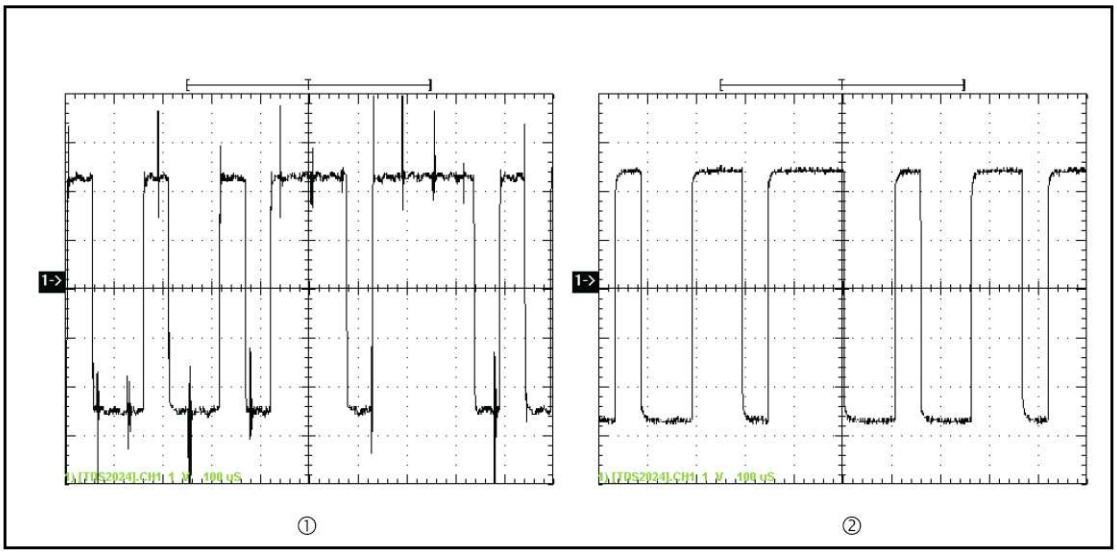
Die Potentialverhältnisse eines CANbus-Systems mit einem CANopen Drucksensor sind durch die folgenden Merkmale gekennzeichnet:

- Der CAN-Bus Anschluss ist nicht potentialgetrennt vom Versorgungsanschluss
- Die einzelnen CANopen Drucksensoren sind nicht galvanisch von der Versorgungsspannung getrennt
- Jeder CANopen Drucksensor 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 Drucksensoren 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 gewährleitstet ist.

Die Erdung der CANopen Drucksensoren erfolgt über den metallischen Gewindeanschluss.

### 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 mm² (paarverseilt mit Abschirmung) | CYPIMF 2 x 2 x 0,5 mm² (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<br>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.

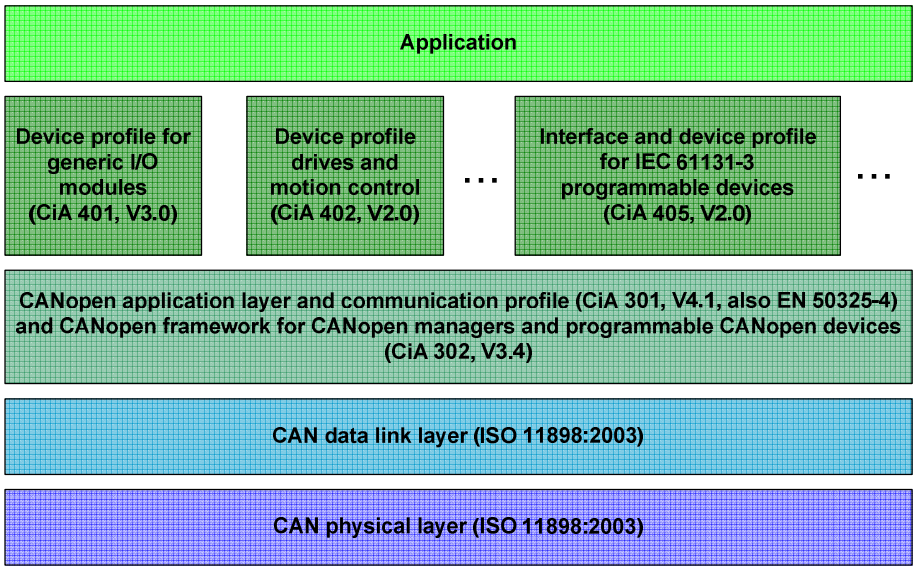
# 4 CANopen

## 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 weisst 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 Übertragungs-protokoll 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.



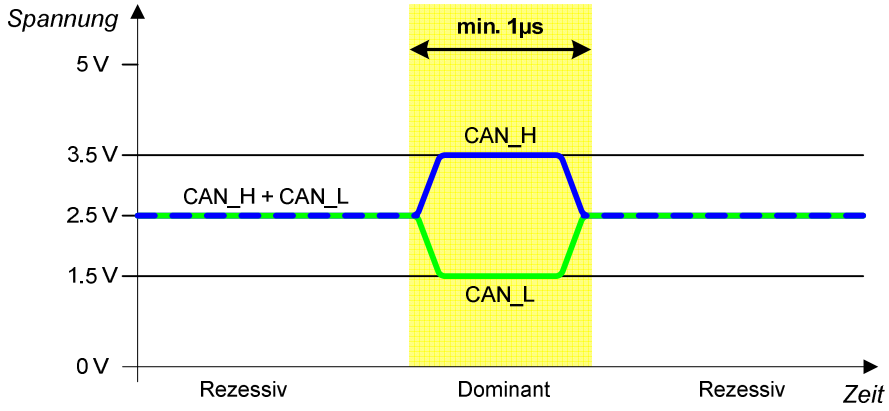

Das CANopen Kommunikationsprofil (CiA DS-301) regelt das „Wie“ der Kommunikation. Hierbei wird zwischen den Echtzeitdaten und den Parameter Daten unterschieden.
Für die Sensoren von Baumer Process Instrumentation wird das Geräteprofil DS404 für Mess- und Regelgeräte angewendet.

## 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.

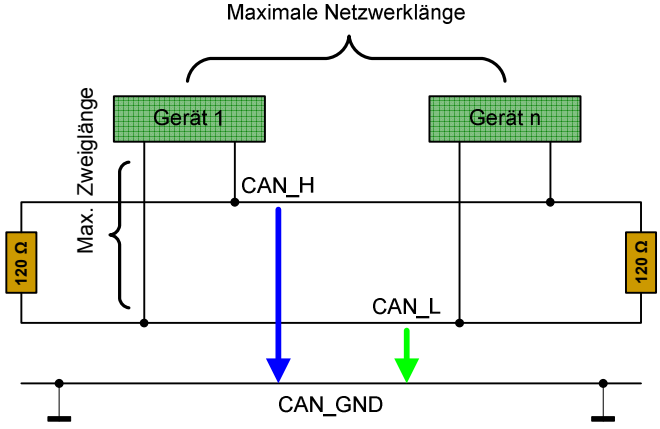
Der Zustand mit zwei unterschiedlichen Pegeln auf CAN-H und CAN-L wird als der dominante Zustand, der Zustand mit zwei gleichen Pegeln als rezessiv bezeichnet.




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 Bedinungen müssen für ein fehlerfrei funktionierendes CANopen Netzwerk gegeben sein:

- Alle Knoten müssen die gleiche Bitrate aufweisen
- Kein 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:

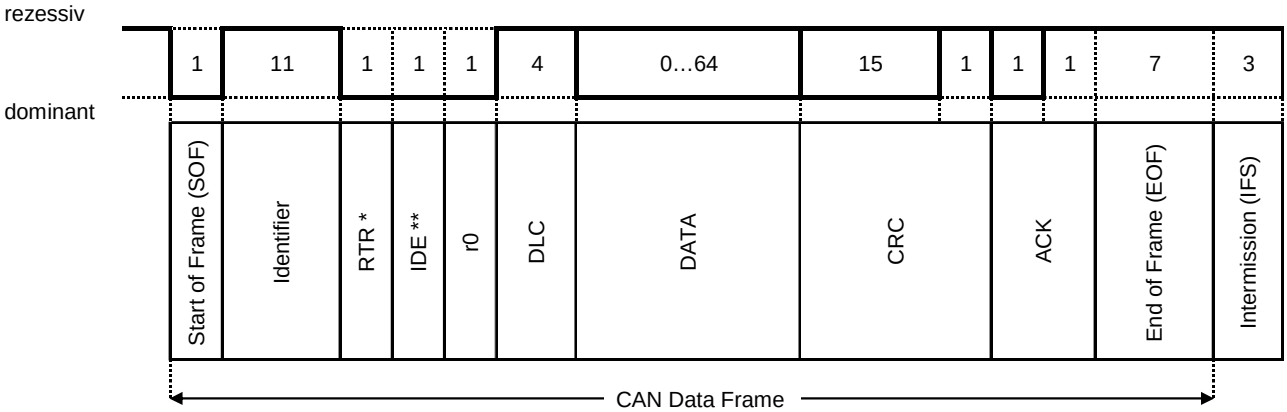
- Start-Condition
- Message Identifier
- Steuerbits
- Daten (0-8 Bytes)
- CRC-Prüfbits
- Acknowledge-Bit
- Stop-Condition

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.

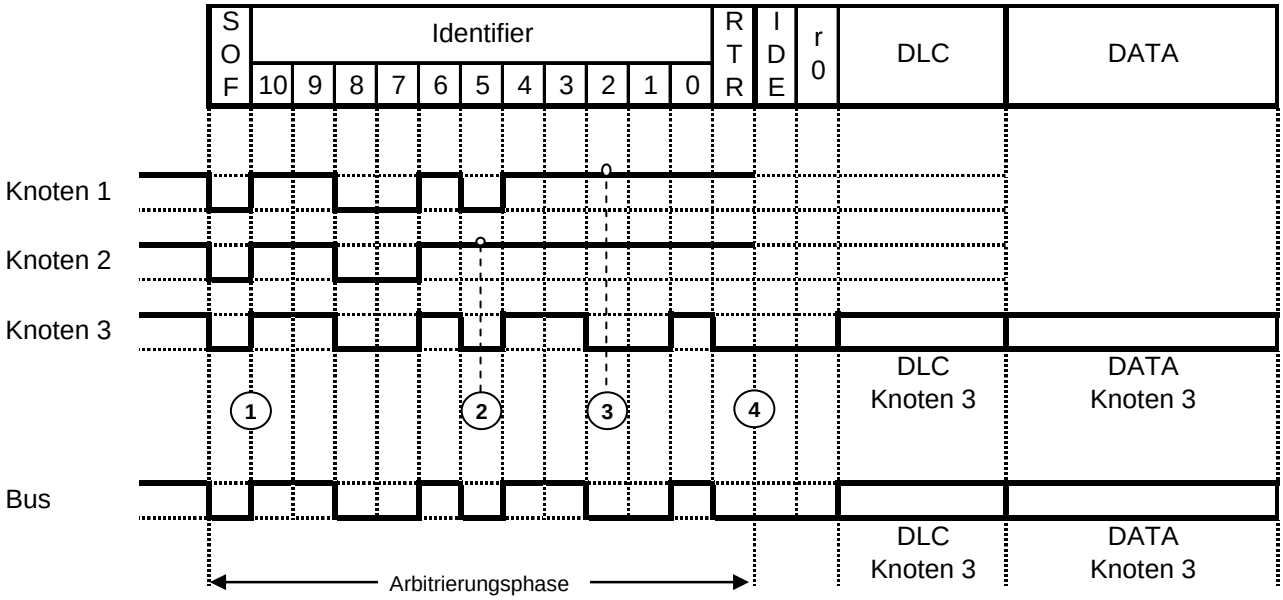



**Bild 1**: Prinzip der bitweisen Busarbitrierung - Knoten 1, 2 und 3 beginnen gleichzeitig einen Arbitrierungsvorgang. Zum Zeitpunkt 2 stellt der Knoten 2 fest, dass der Bus nicht den von ihm gesendeten rezessiven Pegel hat und beendet seinen Arbitrierungsvorgang. Zum Zeitpunkt 3 gibt Knoten 1 auf. 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 höchstpriore 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 niederpriore Nachrichten den Bus mit 100% belegen, ohne die Übertragung höherpriorer Nachrichten maßgeblich zu verzögern. Für die höchstpriore Nachricht resultiert bei einer Übertragungsrate 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 |
| | 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 herstellerspezifische 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 | Netzwerkvariabeln |
| B000h bis BFFFh | Systemvariabeln |
| 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

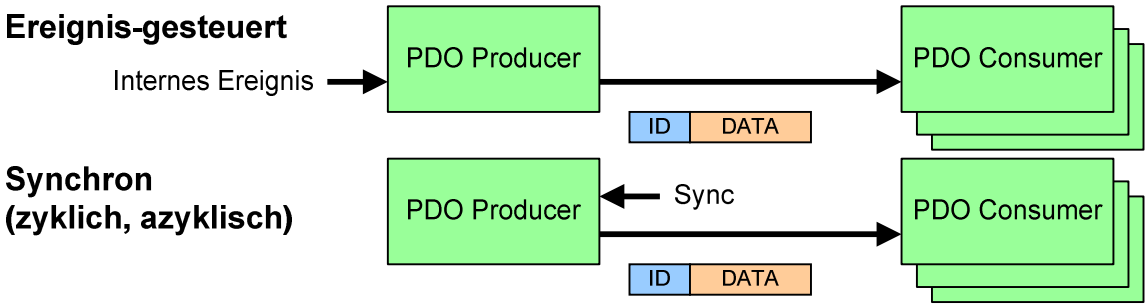
Es wird hauptsächlich zwischen 2 verschiedenen Datenübertragungsarten unterschieden. Die „Process Data Objects“ (PDO) dienen zur Übertragung der Echtzeitdaten/Prozessdaten und die „Service Data Objects“ (SDO) ermöglichen den Zugriff auf das Objektverzeichnis welches sämtliche Einstellungen des Geräts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

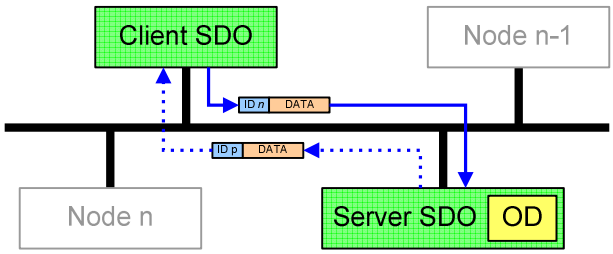
- Ereignisgesteuert
  Die Aussendung der PDO’s wird durch ein internes Ereignis des Knotens ausgelöst. Dies kann durch einen Zeitgeber im Gerät, durch ein Über- oder Unterschreiten einer Limite oder durch weitere interne Ereignisse
- Synchronisiert
  Ein Busteilnehmer (meistens der Master) sendet Synchronisationstelegramme 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 Drucksensoren nicht umgesetzt.

Aufbau einer PDO Nachricht:

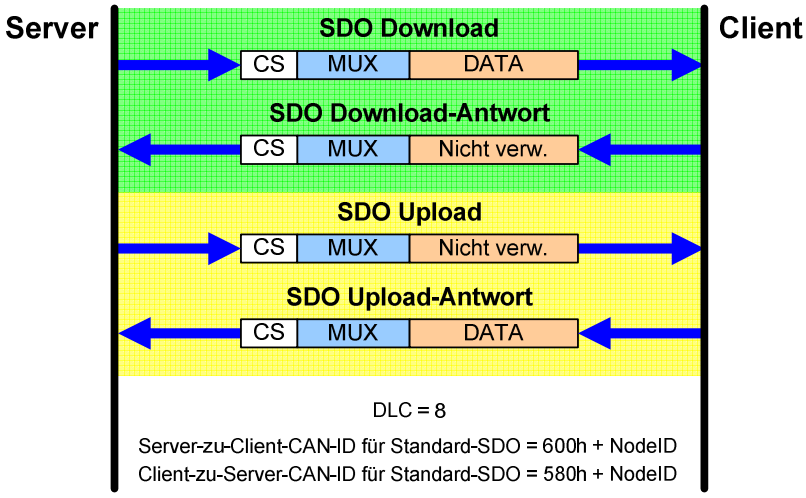



### 4.4.2 Service Data Objects (SDO)

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

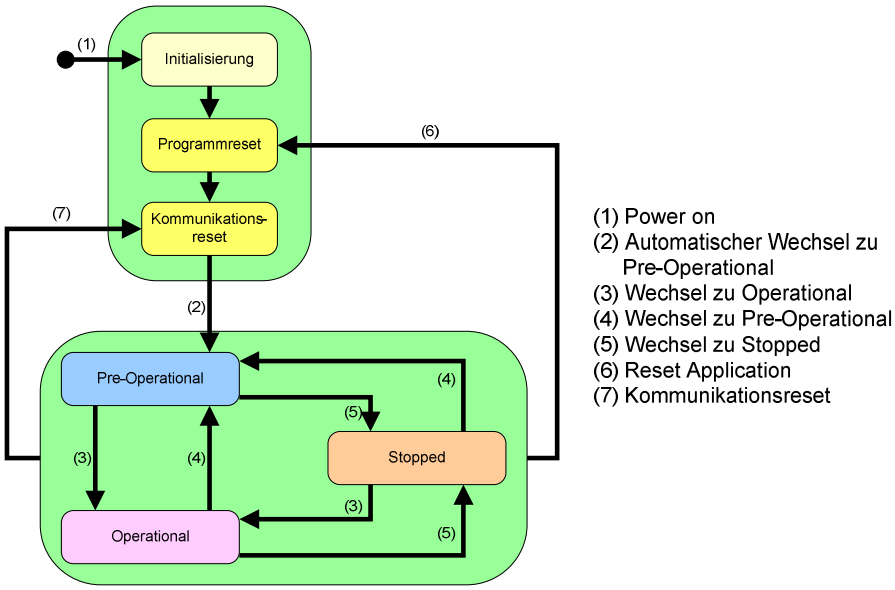



CS = command specifier
MUX = 16-bit Index und 8-Bit Subindex

### 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 ein 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 Hearbeat-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:

| Fehler-code | Fehler-register | Herstellerspezifisches Fehlerfeld |
|---|---|---|

Übersicht über die Fehler-Codes:

| Fehlercode (hex) | Fehlerbeschreibung |
|---|---|
| 00xx | Feherreset / 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 | Geräteprofilspezifisch |
| 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

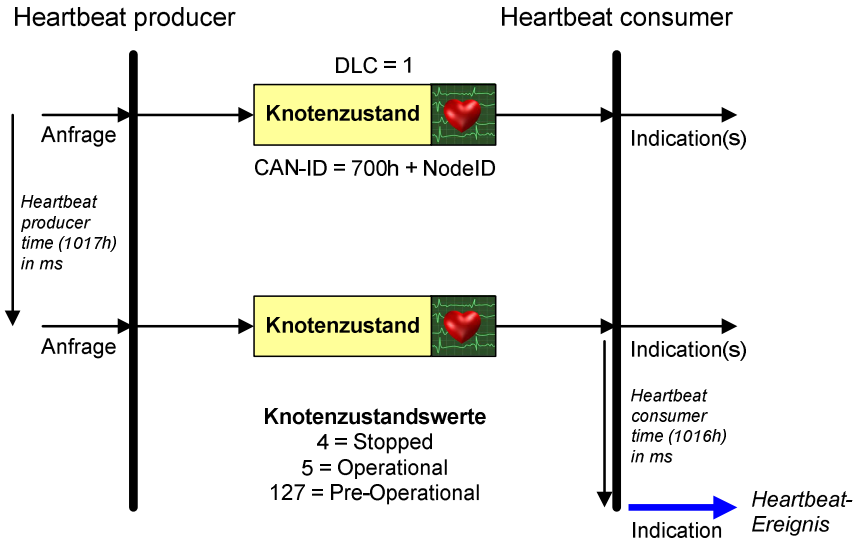
Um die Funktionsfähigkeit der Netzknoten festzustellen hat CANopen folgende Möglichkeiten:

- 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 erkennt 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) beschreit die Funktionalität eines CANopen Gerätes in maschinenlesbarer Form. Diese Dateien in einem standardisierten Textformat beschreiben sowohl alle unterstützen 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 Drucksensor brauchen Sie keine weiteren Hilfsmittel, um die Identifikation und die Baudrate umzustellen. Der Sensor muss 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

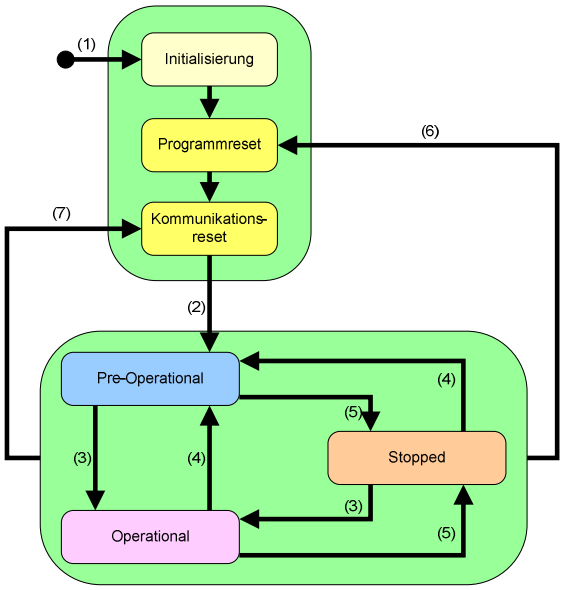
Nach dem Einschalten der Speisung am CANopen-Drucksensor meldet sich der Sensor durch das Senden der CAN-Message “Boot-up Message”. Dies ist eine Mitteilung ohne Datenbytes mit dem COB-Identifier $1792_D$ + Modul-ID ($700_H$ + ID).

### 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 |
| PDO2 (tx) | 641 – 767 | 281h – 2FFh |
| 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 selbständig 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, Emgergency).

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 angegeben Seiten

| Objekt | Sub-Index | Name | Format | Zugriff | Default | PDO mapping | Seite |
|---|---|---|---|---|---|---|---|
| 1000 | 00 | Geräteprofil | U32 | ro | 00‘02‘01‘94h | - | 3 |
| 1001 | 00 | Error Register | U8 | ro | 00h | TPDO | 3 |
| 1002 | 00 | Kalibrationsdatum | U32 | ro | z.B. 10‘07‘14h (14 Juli 2010) | - | 3 |
| 1003 | | Emergency History | ARR | | | | 3 |
| | 00 | Anzahl Fehler | U8 | rw | 00h | - | |
| | 01 | Letzter Fehler | U32 | ro | - | - | |
| | 02-0F | Ältere Fehler | U32 | ro | - | - | |
| 1005 | 00 | Sync ID | U8 | ro | 80h | - | 3 |
| 1008 | 00 | Gerätebezeichnung | U32 | ro | „PDRJ“ | - | 3 |
| 1009 | 00 | Hardware Version | U32 | ro | z.B. „3.02“ | - | 3 |
| 100A | 00 | Software Version | U32 | ro | z.B. „2.02“ | - | 3 |
| 1010 | | Speichern | ARR | | | | 3 |
| | 00 | Anzahl Subindex | U8 | ro | 01h | - | |
| | 01 | Alle Parameter speichern | U32 | rw | „save“ | - | |
| 1011 | | Defaultwerte laden | ARR | | | | 3 |
| | 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 | - | 3 |
| 1017 | 00 | Heartbeat [ms] | U16 | rw | 00‘00h | - | 3 |
| 1018 | | Geräte Identität | REC | | | | 3 |
| | 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‘02h | - | |
| | 04 | Seriennummer | U32 | ro | z.B. 00000000d | - | |
| 1800 | | Sende PDO 1 | REC | | | | 3 |
| | 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 | | | | 3 |
| | 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 | - | |
| 1A00 | | PDO 1 Mapping Parameter | ARR | | | | 3 |
| | 00 | Anzahl Subindex | U8 | rw | 01h | - | |
| | 01 | Erstes übertragene Objekt | U32 | rw | 71‘30‘01‘10h | - | |
| | 02-04 | Weitere übertragene Objekte | U32 | rw | *[2] | - | |
| 1A01 | | PDO 2 Mapping Parameter | ARR | | | | 3 |
| | 00 | Anzahl Subindex | U8 | rw | 01h | - | |
| | 01 | Erstes übertragene Objekt | U32 | rw | 71‘30‘02‘10h | - | |
| | 02-04 | Weitere übertragene Objekte | U32 | rw | *[2] | - | |
| 2000 | 00 | Mittelungszeit [ms] | U16 | rw | 00‘1Eh | - | 3 |
| 2100 | 00 | Baud-Rate | U8 | rw | 03h | - | 3 |

| Objekt | Sub-Index | Name | Format | Zugriff | Default | PDO mapping | Seite |
|---|---|---|---|---|---|---|---|
| 2101 | 00 | Identifikation | U8 | rw | 01h | - | 3 |
| 6110 | | Sensortyp | ARR | | | | 3 |
| | 00 | Anzahl Subindex | U8 | ro | 01h | - | |
| | 01 | Sensortyp | U16 | ro | 46h | - | |
| 6112 | | Betriebsart | ARR | | | | 3 |
| | 00 | Anzahl Subindex | U8 | ro | 01h | - | |
| | 01 | Betriebsart des Sensors | U8 | ro | 01h | - | |
| 6131 | | Einheit Prozesswert | ARR | | | | 3 |
| | 00 | Anzahl Subindex | U8 | ro | 01h | - | |
| | 01 | Physik. Prozesswerteinheit Druck | U32 | ro | 00‘00‘4E‘00h | - | |
| | 02*[1] | Physik. Prozesswerteinheit Temperatur | U32 | ro | 00’00’2D’00h | - | |
| 6132 | | Dezimalstellen | ARR | | | | 3 |
| | 00 | Anzahl Subindex | U8 | ro | 01h | - | |
| | 01 | Anzahl Dezimalstellen Druck | U8 | ro | z.B. 02h | - | |
| | 02*[1] | Anzahl Dezimalstellen Temperatur | U8 | ro | z.B. 01h | - | |
| 6150 | | Status der Messung | ARR | | | | 3 |
| | 00 | Anzahl Subindex | U8 | ro | 01h | - | |
| | 01 | Status Messung Prozesswert 1 | U8 | ro | z.B. 00h | TPDO | |
| | 02*[1] | Status Messung Prozesswert 2 | U8 | ro | z.B. 00h | TPDO | |
| 7130 | | Abfrage des Messwertes 16bit | ARR | | | | 3 |
| | 00 | Anzahl Subindex | U8 | ro | 01h | - | |
| | 01 | Prozesswert 1 in 16bit | I16 | ro | z.B. 01‘2Ch | TPDO | |
| | 02*[1] | Prozesswert 2 in 16bit | I16 | ro | z.B. 00’C8h | TPDO | |
| 7133 | | Delta-Wert 16bit | ARR | | | | 3 |
| | 00 | Anzahl Subindex | U8 | ro | 01h | - | |
| | 01 | Delta-Wert PW1 16bit für PDO | U16 | rw | 00‘00h | - | |
| | 02*[1] | Delta-Wert PW2 16bit für PDO | U16 | rw | 00’00h | - | |

*[1] Subindex ist nur vorhanden, wenn Sensor Temperaturwert ausgeben kann (S- und T- Typen)
*[2] Subindex sind nicht vorhanden, kommen erst zur Anwendung wenn Objekt neu gemappt wird.

## 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-Profils 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 Drucksensor:

| 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 Drucksensor:

| 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 Drucksensor:

| ID | DLC | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 | Byte8 |
|---|---|---|---|---|---|---|---|---|---|
| 580h + ID | 8 | 43h | 08h | 10h | 00h | 50h | 44h | 52h | 4Ah |

Beispiel Drucksensor Sensor:
Byte 5 – 8: 50'44'52'4Ah „PDRJ" 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 Drucksensor:

| ID | DLC | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 | Byte8 |
|---|---|---|---|---|---|---|---|---|---|
| 580h + ID | 8 | 43h | 09h | 10h | 00h | 33h | 2Eh | 30h | 32h |

Beispiel Hardware Version 3.02:
Byte 5 – 8: 33’2E’30’32h „3.02“ 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 Drucksensor:

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

Beispiel Software Version 2.02:
Byte 5 – 8: 32’2E’30’32h „2.02“ 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 | Serienummer | 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 Drucksensor:

| 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 Drucksensor:

| 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 Drucksensor:

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

Beispiel Revisionsnummer 00030202:
Byte 5 – 8: 02’02’03’00h 00’03’02’02 (LSB first) ➔ 00030202h

Die Serienummer (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 Drucksensor:

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

Beispiel Serienummer 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 |
| 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 |
| 1A00 | Mapping Transmit PDO1, Sub-Index 1 | 71’30’01’10h | Objekt 7130 Sub-Index 1 gemappt |
| 1A01 | Mapping Transmit PDO2, Sub-Index 1 | 71’30’02’10h (71’30’01’10h) | Objekt 7130 Sub-Index 2 gemappt nur wenn vorhanden, ansosnten Sub1 |
| 2000 | Mittelungszeit | 00’30h | 30 ms |
| 2100 | Baud-Rate | 03h | 125 kBaud |
| 2101 | Identifikation (7 bit) | 01h | ID 1 |
| 7133 | Interrupt Delta Input Prozesswert 1 16bit | 00h | deaktiviert |
| | Interrupt Delta Input Prozesswert 2 16bit | 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 |

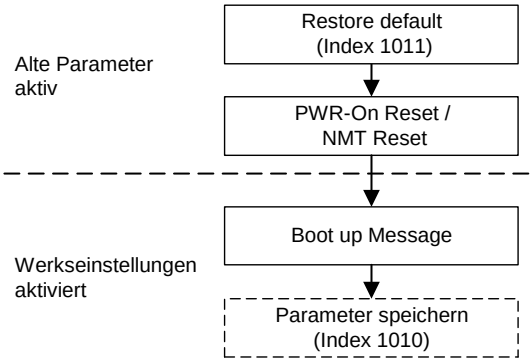
Antwort vom CANopen Drucksensor:

| 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:




Werden die Parameter nicht gespeichert und ein 2. Mal ein Reset durchgeführt, so werden die „alten“ Parameter wieder aktiv!

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 Drucksensor:

| 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 |
| 6131 | Einheit Prozesswert |
| 6132 | Dezimalstellen Prozesswert |
| 6150 | Status der Messung |
| 7130 | Prozesswert 16bit |
| 7133 | Interrupt Delta-Prozesswert 16bit |

### 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 Drucksensor:

| ID | DLC | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 | Byte8 |
|---|---|---|---|---|---|---|---|---|---|
| 580h + ID | 8 | 4Bh | 10h | 61h | 01h | 5Ah | 00h | 0 | 0 |

Byte 5: 5Ah Drucksensor

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 Drucksensor:

| 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 Einheit Prozesswert

Physikalische Prozesswerteinheit (Objekt 6131h)

| Objekt | Sub-Index | Parameter | Länge | Access |
|---|---|---|---|---|
| 6131h | 0 | Anzahl der Sub-Indexe | 1 Byte | read |
| | 1 | Einheit Prozesswert 1 | 2 Byte | read |
| | 2* | Einheit Prozesswert 2 | 2 Byte | read |

*nur bei Sensoren mit 2 Prozesswerten (Temperaturmessung) vorhanden: S- & T-Typen

Die physikalische Prozesswerteinheit des CANopen Drucksensors 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 Drucksensor:

| ID | DLC | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 | Byte8 |
|---|---|---|---|---|---|---|---|---|---|
| 580h + ID | 8 | 43h | 31h | 61h | 01h | 00h | 00h | 4Eh | 00h |

| | | |
|---|---|---|
| Byte 4: | 01h | Sub-Index 1 (Einheit Druck) |
| | 02h | Sub-Index 2 (Einheit Temperatur) |
| Byte 5 – 8: | 00’00’4E’00h | 00’4E’00’00 (LSB first) ➔ Einheit “Bar” |
| | 00’00’4E’FDh | FD’4E’00’00 (LSB first) ➔ Einheit “mBar” |
| | 00’00’2D’00h | 00’2D’00’00 (LSB first) ➔ Einheit “℃” |
| | 00’00’AC’00h | 00’AC’00’00 (LSB first) ➔ Einheit “℉” |
| | 00’00’05’00h | 00’05’00’00 (LSB first) ➔ Einheit “Kelvin” |

Einheiten der Prozesswerte sind Sensortyp abhängig und nicht durch den Benutzer einstellbar.

Das Objekt 6131h kann nur gelesen werden.

### 6.3.4 Dezimalstellen Prozesswert

Anzahl der Dezimalstellen des Prozesswerts (Objekt 6132h)

| Objekt | Sub-Index | Parameter | Länge | Access |
|---|---|---|---|---|
| 6132h | 0 | Anzahl der Sub-Indexe | 1 Byte | read |
| | 1 | Dezimalstellen Prozesswert 1 | 1 Byte | read |
| | 2* | Dezimalstellen Prozesswert 2 | 1 Byte | read |

*nur bei Sensoren mit 2 Prozesswerten (Temperaturmessung) vorhanden: S- & T-Typen

Die Anzahl der Dezimalstellen des Prozesswerts 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 Drucksensor:

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

| | | |
|---|---|---|
| Byte 4: | 01h | Sub-Index 1 (Dezimalstellen Druck) |
| | 02h | Sub-Index 2 (Dezimalstellen Temperatur) |
| Byte 5: | 02h | 2 Dezimalstellen |

Das Objekt 6132h kann nur gelesen werden.

### 6.3.5 Status der Messung

Status der Messung (Objekt 6150h)

| Objekt | Sub-Index | Parameter | Länge | Access |
|---|---|---|---|---|
| 6150h | 0 | Anzahl der Sub-Indexe | 1 Byte | read |
| | 1 | Status Messung Prozesswert 1 | 1 Byte | read |
| | 2* | Status Messung Prozesswert 2 | 1 Byte | read |

*nur bei Sensoren mit 2 Prozesswerten (Temperaturmessung) vorhanden: S- & T-Typen

Der Status der Messung 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 Drucksensor:

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

| | | |
|---|---|---|
| Byte 4: | 01h | Sub-Index 1 (Status Druckmessung) |
| | 02h | Sub-Index 2 (Status Temperaturmessung) |
| Byte 5: | 00h | Messwert aktuell |
| | 03h | Überlauf des AD – Wandlers |
| | 05h | Unterlauf des AD – Wandlers |

Das Objekt 6150h kann nur gelesen werden.

### 6.3.6 Abfrage Prozesswert 16bit

16bit Prozesswert (Objekt 7130h)

| Objekt | Sub-Index | Parameter | Länge | Access |
|---|---|---|---|---|
| 7130h | 0 | Anzahl der Sub-Indexe | 1 Byte | read |
| | 1 | Abfrage Prozesswert 1 | 2 Byte | read |
| | 2* | Abfrage Prozesswert 2 | 2 Byte | read |

*nur bei Sensoren mit 2 Prozesswerten (Temperaturmessung) vorhanden: S- & T-Typen

Der 16bit Prozesswert des Drucksensors 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 Drucksensor:

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

| | | |
|---|---|---|
| Byte 4: | 01h | Sub-Index 1 (Druckwert) |
| | 02h | Sub-Index 2 (Temperaturwert) |
| Byte 5 + 6: | 2C'01h | 01'2C (LSB first) ➔ 300 |
| | | 300 / $10^{\text{Anzahl Dezimalstellen}}$(6132) + Einheit(6131) = 3.00 Bar |

Der Prozesswert wird als 16-bit Integer 2-er Komplement ausgegeben.

Beispiel für negativen Temperaturwert:

| ID | DLC | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 | Byte8 |
|---|---|---|---|---|---|---|---|---|---|
| 580h + ID | 8 | 4Bh | 30h | 71h | 02h | ECh | FFh | 0 | 0 |

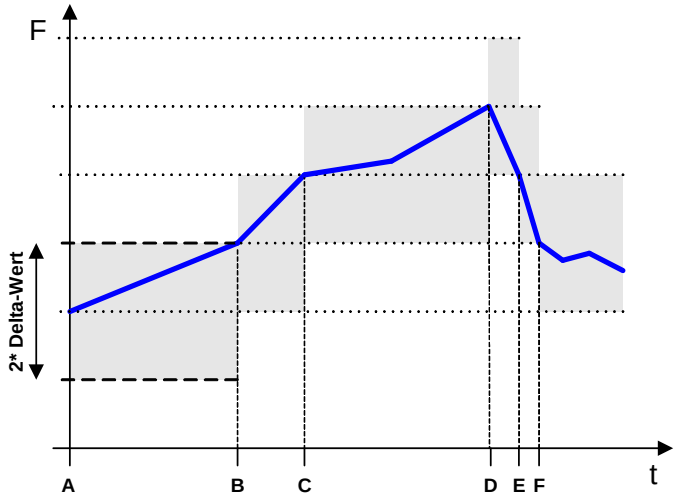
Byte 5 + 6: EC'FFh FF'EC (LSB first) ➔ 65516d = -20
-20 / $10^{\text{Anzahl Dezimalstellen}}$(6132) + Einheit(6131) = -2.0 ℃

Sollte der Prozesswert höher als der mögliche Darstellungsbereich sein, so zeigt der CANopen Drucksensor den Wert 7F'FFh (32767d) an.

Das Objekt 7130h kann nur gelesen werden.

### 6.3.7 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)




A: Aktivieren der Delta-Funktion über PDO Transmission Type

B-F: Prozesswert wird über PDO gesendet, Schwellwert wird neu gesetzt

F: Prozesswert
t: Zeit

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

| Objekt | Sub-Index | Parameter | Länge | Access |
|---|---|---|---|---|
| 7133h | 0 | Anzahl der Sub-Indexe | 1 Byte | read |
| | 1 | Delta-Wert für Prozesswert 1 | 2 Byte | read / write |
| | 2* | Delta-Wert für Prozesswert 2 | 2 Byte | read / write |

*nur bei Sensoren mit 2 Prozesswerten (Temperaturmessung) vorhanden: S- & T-Typen

Die Delta-Funktion wird über den Transmission Type des PDO1 (Objekt 1800h) und PDO2 (Objekt 1801h) 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 Bar $5 * 10^2 = 500 = 01'F4h$ (Delta-Wert)
Byte 4: 01h Sub-Index 1 ➔ Delta-Wert für PDO1
Byte 5 + 6: F4'01h 01'F4 (LSB first) ➔ 500

Antwort vom CANopen Drucksensor:

| 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 Drucksensor:

| 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) ➔ 500 ➔ 5 Bar

## 6.4 Herstellerspezifische Objekte

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

| Objekt | Beschreibung |
|---|---|
| 2000 | Mittelungszeit |
| 2100 | Baudrate |
| 2101 | Identifikation |

### 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.

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 Drucksensor:

| 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 Drucksensor:

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

### 6.4.2 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 Drucksensor 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 |

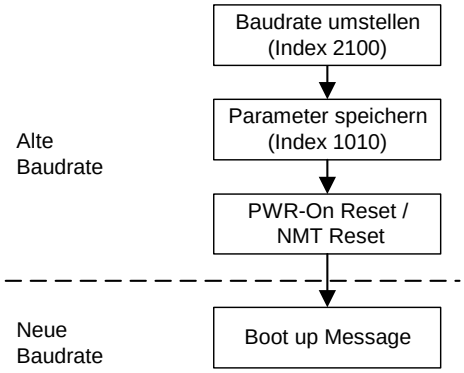
Antwort vom CANopen Drucksensor:

| 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
01h ➔ 20 kBaud
02h ➔ 50 kBaud
03h ➔ 125 kBaud
04h ➔ 250 kBaud
05h ➔ 500 kBaud
06h ➔ 800 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 Drucksensor:

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

### 6.4.3 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 Drucksensor:

| 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:

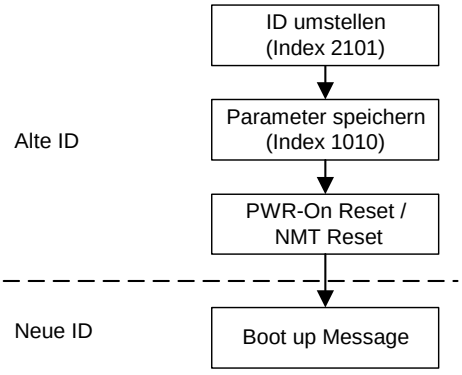
| ID | DLC | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 | Byte8 |
|---|---|---|---|---|---|---|---|---|---|
| 600h + ID | 8 | 22h | 01h | 21h | 00h | 05h | 0 | 0 | 0 |

Byte 5: 05h ID 5

Antwort vom CANopen Drucksensor:

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

Funktionsweise:



## 6.5 PDO-Kommunikation Objekte

Über die Kommunikation mittels PDO (Process Data Objects) ist es möglich bestimmte Werte des CANopen Drucksensors auf einfache und schnelle Weise zu übermitteln. Bei den Drucksensoren 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 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 Drucksensor:

| 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 |

| | | |
|---|---|---|
| Byte 5: | 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 Drucksensor:

| 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... $2^{16}$ – 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 Drucksensor:

| 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.

### 6.5.2 Sende PDO 2

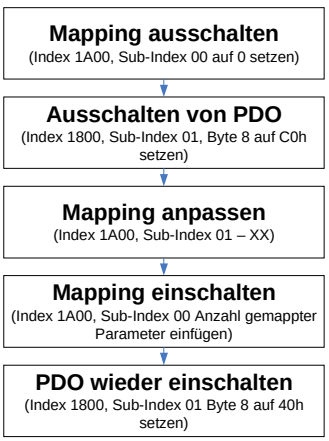
Objekt 1801h (Transmit PDO Communication Parameter):
Über das Objekt 1801h werden die Einstellungen für die PDO 2 vorgenommen. Die Parameter sind identisch mit dem PDO 1. Detaillierte Angaben können von PDO 1 entnommen werden.

### 6.5.3 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 (Druckwert) | 2 Bytes |
| 7130 | 02* | Prozesswert 2 (Temperaturwert) | 2 Bytes |
| 6150 | 01 | Status Prozesswert 1 | 1 Byte |
| 6150 | 02* | Status Prozesswert 2 | 1 Byte |
| 1001 | 00 | Error Register | 1 Byte |

*nur bei Sensoren mit 2 Prozesswerten (Temperaturmessung) vorhanden: S&T Typen

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 Drucksensor:

| 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

### 6.5.4 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. Die COB-ID des Sende PDO 2 hat 280h + ID (PDO 1 hat 180h + ID).

### 6.5.5 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 Drucksensor:

| 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 Drucksensor:

| 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 Drucksensor:

| 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 Drucksensor:

| 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 Drucksensor:

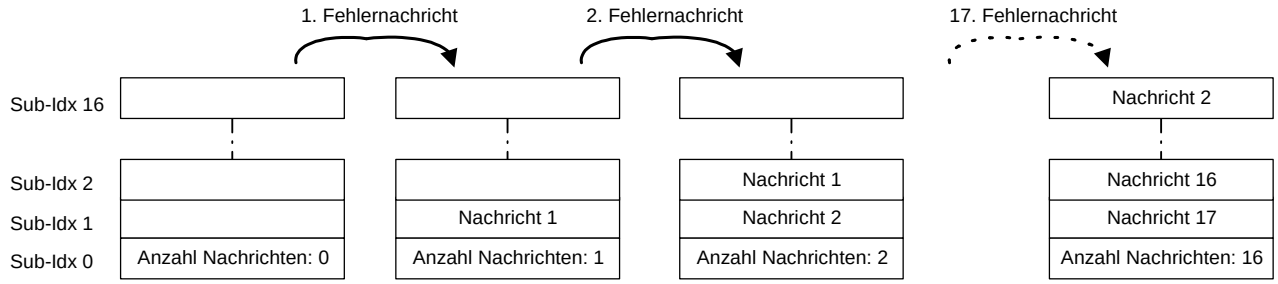
| ID | DLC | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 | Byte8 |
|---|---|---|---|---|---|---|---|---|---|
| 580h + ID | 8 | 43h | 03h | 10h | 01h | 00h | FFh | 81h | 14h |

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 | Additional Code | | Error Code | Error Class |

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 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 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 |

## 7.3 Emergency Messages

Emergency Messages werden im Fehlerfall vom Sensor selbständig gesendet. Beim ersten Auftreten eines Fehlers wird eine Fehlernachricht gesendet. Wird der Fehlergrund behoben oder liegt der Fehler nicht mehr an, wird ebenfalls eine entsprechende Fehlernachricht gesendet.
Die letzten 16 Fehlernachrichten werden in der Emergency History gespeichert.

Die Fehlermeldungen sind wie folgt aufgebaut:

| ID | DLC | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 | Byte8 |
|---|---|---|---|---|---|---|---|---|---|
| 80h + ID | 8 | Emergency Error Code | | Error Register | Herstellerspezifischer Error Code | | | | |

Folgende Error Codes werden unterstützt:

| **Emergency Error Code** | **Error Register** | **Herster. Error Code** | **Beschreibung** |
|---|---|---|---|
| 50’01h | 81h | 10h | EEPROM Lesefehler (Hardware Fehler)<br>Beim Aufstarten: Fehler wird mit ID1 und 125kBaud gesendet |
| 00’00h | 00h | 20h | EEPROM Schreibfehler behoben<br>Erfolgreiche Speicherung von Tariernullpunkt im EEPROM |
| 50’01h | 81h | 30h | EEPROM Schreibfehler (Hardware Fehler)<br>Beim Speichern des Tariernullpunkts im EEPROM |
| 00’00h | 00h | 11h | Druck-Signal hat zulässigen Bereich wieder erreicht |
| FF’00h | 81h | 12h | maximale Druck-Signalgrösse überschritten |
| FF’00h | 81h | 14h | minimale Druck-Signalgrösse unterschritten |
| 00’00h | 00h | 21h | Temperatur-Signal hat zulässigen Bereich wieder erreicht |
| FF’00h | 81h | 22h | maximale Temperatur-Signalgrösse überschritten |
| FF’00h | 81h | 24h | minimale Temperatur-Signalgrösse unterschritten |

Objekt 1014h (COB-ID Emergency Object):
In diesem Objekt ist die ID bei einem Fehlerfall festgehalten. Anhand dieser ID kann die Fehlermeldung einem Sensor zugeordnet werden.

Abfrage der ID (Objekt 1014h) für die Emergency Message. Die ID kann wie folgt gelesen werden:

| ID | DLC | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 | Byte8 |
|---|---|---|---|---|---|---|---|---|---|
| 600h + ID | 8 | 40h | 14 | 10h | 00h | 0 | 0 | 0 | 0 |

Antwort vom CANopen Drucksensor:

| ID | DLC | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 | Byte8 |
|---|---|---|---|---|---|---|---|---|---|
| 580h + ID | 8 | 43h | 14h | 10h | 00h | 81h | 0 | 0 | 0 |

Byte 5: 81h ID = 80h + Node ID

Das Objekt 1014h kann nur gelesen werden und hat keinen Sub-Index.

## 7.4 Heartbeat

Der CANopen Drucksensor 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. Default-mässig 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 Drucksensor:

| 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 Drucksensor:

| 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 |

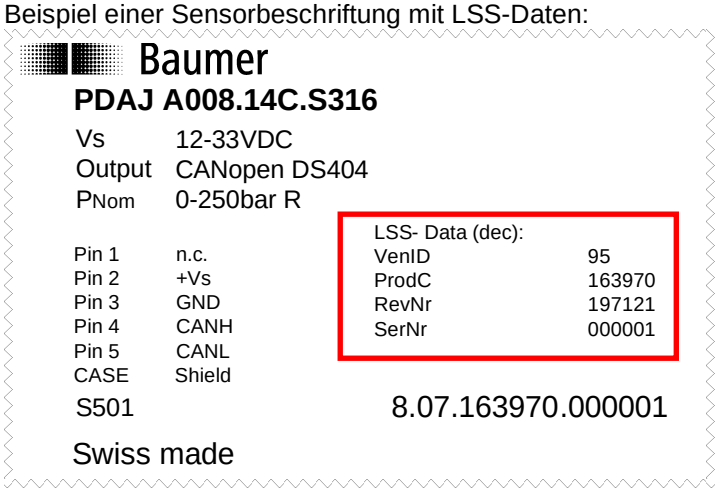
| | | |
|---|---|---|
| Byte 1: | 00h | Aufstarten (Boot up) |
| | 04h | Stop Mode |
| | 05h | Operate |
| | 7Fh | Pre-operational |

## 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 Drucksensor aufgedruckt.

Beispiel einer Sensorbeschriftung mit LSS-Daten:




Die LSS-Daten sind im dezimalen Zahlensystem dargestellt.

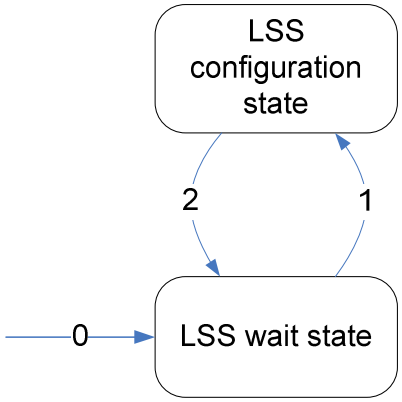
| **Aufschrift:** | **Bezeichnung:** | **Objekt:** | **Sub-Index:** |
|---|---|---|---|
| VenID | Vendor-ID | 1018h | 01h |
| ProdC | Produkt-Code | 1018h | 02h |
| RevNr | Revisions-Nummer | 1018h | 03h |
| SerNr | Serie-Nummer | 1018h | 04h |

### 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.

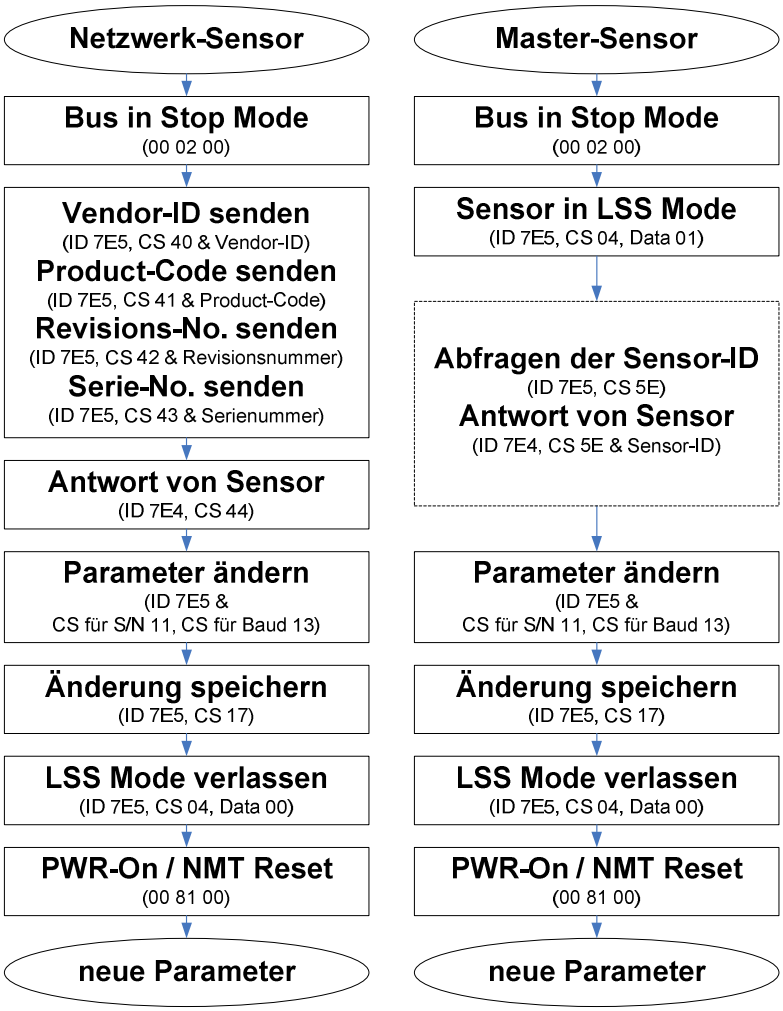
Alle Sensoren in den Stop Modus bringen:

| 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 Drucksensor:

| 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 unterstützen 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 | 02h | 02h | 03h | 00h | 0 | 0 | 0 |

Revisions-Nummer:

| | | |
|---|---|---|
| Byte 1: | 42h | Systembyte Revisions-Nummer |
| Byte 2 – 5: | 02’02’03’00h | 00’03’02’02 (LSB first) ➔ 00030202h |

Übermitteln der Serienummer:

| ID | DLC | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 | Byte8 |
|---|---|---|---|---|---|---|---|---|---|
| 7E5 | 8 | 43h | 7Bh | 00h | 00h | 00h | 0 | 0 | 0 |

Serienummer:

| | | |
|---|---|---|
| Byte 1: | 43h | Systembyte Serienummer |
| Byte 2 – 5: | 7B'00'00'00h | 00'00'00'7B (LSB first) ➔ 123 |

Antwort vom CANopen Drucksensor:

| ID | DLC | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 | Byte8 |
|---|---|---|---|---|---|---|---|---|---|
| 7E4 | 8 | 44h | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Mit dieser Antwort bestätigt der Drucksensor, dass er erkannt wurde. Nun können die ID und die Baudrate vom Sensor geändert werden.

### 7.5.5 Verstellen der ID und der Baudrate

Anpassen der ID:

| ID | DLC | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 | Byte8 |
|---|---|---|---|---|---|---|---|---|---|
| 7E5 | 8 | 11h | 01h | 0 | 0 | 0 | 0 | 0 | 0 |

Beispiel ID = 1:

| | | |
|---|---|---|
| Byte 2: | 01h | ID = 1 |

Antwort vom CANopen Drucksensor:

| ID | DLC | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 | Byte8 |
|---|---|---|---|---|---|---|---|---|---|
| 7E4 | 8 | 11h | 00h | 0 | 0 | 0 | 0 | 0 | 0 |

| | | |
|---|---|---|
| Byte 2: | 00h | ID wurde erfolgreich umgestellt |
| | 01h | ID ausserhalb gültigem Bereich |
| | FFh | spezifischer Fehler |

Anpassen der Baudrate:

| ID | DLC | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 | Byte8 |
|---|---|---|---|---|---|---|---|---|---|
| 7E5 | 8 | 13h | 00h | 04h | 0 | 0 | 0 | 0 | 0 |

Beispiel Baudrate = 125kBaud:

| | | |
|---|---|---|
| Byte 2: | 00h | CiA Bauraten Tabelle |
| Byte 3: | 04h | 4 = 125kBau |

Baudraten:

| | | |
|---|---|---|
| 0=1Mbaud | 3=250kBaud | 7=20kBaud |
| 1=800kBaud | 4=125kBaud | 8=10kBaud |
| 2=500kBaud | 6=50kBaud | |

Antwort vom CANopen Drucksensor:

| ID | DLC | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 | Byte8 |
|---|---|---|---|---|---|---|---|---|---|
| 7E4 | 8 | 13h | 00h | 0 | 0 | 0 | 0 | 0 | 0 |

| | | |
|---|---|---|
| Byte 2: | 00h | Baudrate wurde erfolgreich umgestellt |
| | 01h | Baudrate ausserhalb gültigem Bereich |
| | FFh | spezifischer Fehler |

### 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 Drucksensor:

| 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 Abfragen eines Prozesswerts über SDO

### Messinitialisierung:

**1. Einheit abfragen (Index 61'31h)**

Master / Slave

| 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 | 00h | 4Eh | 00h |

00'4E'00'00h => Bar

**2. Dezimalstellen abfragen (Index 61'32h)**

Master / Slave

| 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 |

02h => 2 Dezimalstellen => Division mit $10^2$

### Messzyklus:

**Index 71'30h, positiver Druck**

Master / Slave

| 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 |

01'2Ch => $300_{Dez}$

Dezimalpunkt verschieben: 300 / $10^2$ = 3.00

Einheit hinzufügen: 3.00 => 3.00 Bar

**Index 71'30h, negativer Druck**

Master / Slave

| 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 | D4h | FEh | 0 | 0 |

FE'D4h => 1111'1110'1101'0100b (inv.)
- 0000'0001'0010'1011b (+1)
- 0000'0001'0010'1100b = $-300_{Dez}$

Dezimalpunkt verschieben: -300 / $10^2$ = -3.00

Einheit hinzufügen: -3.00 => -3.00 Bar

## 8.2 Konfigurieren des Prozesswerts über PDO1

### Messinitialisierung:

**1. Einheit abfragen (Index 61'31h)**

Master / Slave

| 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 | 00h | 4Eh | 00h |

00'4E'00'00h => Bar

**2. Dezimalstellen abfragen (Index 61'32h)**

Master / Slave

| 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 |

02h => 2 Dezimalstellen => Division mit $10^2$

## Konfiguration PDO Einstellungen:

### 1. PDO1 Transmission Type einstellen

Master

| 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 |

FFh => ausgelöst nach Ablauf der Event Time

### 2. PDO1 Event Time einstellen (1sec)

Master Slave

| 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 |

03'E8h => $1000_{Dez}$ => Event Time = 1000ms

### 3. Mapping PDO1 deaktivieren

Master Slave

| 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 |

### 4. PDO1 ausschalten

Master Slave

| ID | DLC | Byte1 | Index | | Sub. | gem. Index | | Byte7 | Aktiv |
|---|---|---|---|---|---|---|---|---|---|
| 600h + ID | 8 | 22h | 00h | 18h | 01h | 180h + ID | | 00h | C0h |

| ID | DLC | Byte1 | Index | | Sub. | Byte5 | Byte6 | Byte7 | Byte8 |
|---|---|---|---|---|---|---|---|---|---|
| 580h + ID | 8 | 60h | 00h | 18h | 01h | 0 | 0 | 0 | 0 |

### 5. Mapping PDO1 anpassen

Master Slave

| ID | DLC | Byte1 | Index | | No. | Länge | Sub. | gem. Index | |
|---|---|---|---|---|---|---|---|---|---|
| 600h + ID | 8 | 22h | 00h | 1Ah | 01h | 10h | 01h | 30h | 71h |

| ID | DLC | Byte1 | Index | | No. | Byte5 | Byte6 | Byte7 | Byte8 |
|---|---|---|---|---|---|---|---|---|---|
| 580h + ID | 8 | 60h | 00h | 1Ah | 01h | 0 | 0 | 0 | 0 |

No.: 01h => 1. gemapptes Objekt in PDO1
Länge: 10h => 16 Bit Datenlänge des gemappten Index
Sub.: 01h => gemappter Subindex = 01h
Gem. Index: 30'71h => gemappter Index = 71'30h

### 6. Mapping PDO1 aktivieren

Master Slave

| ID | DLC | Byte1 | Index | | Sub. | Anz. | Byte6 | Byte7 | Byte8 |
|---|---|---|---|---|---|---|---|---|---|
| 600h + ID | 8 | 22h | 00h | 1Ah | 00h | 01h | 0 | 0 | 0 |

| ID | DLC | Byte1 | Index | | Sub. | Byte5 | Byte6 | Byte7 | Byte8 |
|---|---|---|---|---|---|---|---|---|---|
| 580h + ID | 8 | 60h | 00h | 1Ah | 00h | 0 | 0 | 0 | 0 |

01h => 1 gemapptes Objekt in PDO1

### 7. PDO1 einschalten

Master Slave

| ID | DLC | Byte1 | Index | | Sub. | COB-ID | Byte7 | Aktiv |
|---|---|---|---|---|---|---|---|---|
| 600h + ID | 8 | 22h | 00h | 18h | 01h | 180h + ID | 0 | 40h |

| ID | DLC | Byte1 | Index | | Sub. | Byte5 | Byte6 | Byte7 | Byte8 |
|---|---|---|---|---|---|---|---|---|---|
| 580h + ID | 8 | 60h | 00h | 18h | 01h | 0 | 0 | 0 | 0 |

## Abfrage des Prozesswerts:

### Sensor in Operational Mode bringen

Master Slave

| ID | DLC | Byte1 | Byte2 |
|---|---|---|---|
| 0 | 2 | 01h | ID |

Master Slave

| ID | DLC | Prozesswert | |
|---|---|---|---|
| 180h + ID | 2 | XXh | YYh |

nach 1s

| ID | DLC | Prozesswert | |
|---|---|---|---|
| 180h + ID | 2 | XXh | YYh |

nach 1s

| ID | DLC | Prozesswert | |
|---|---|---|---|
| 180h + ID | 2 | XXh | YYh |

## 8.3 ID Ändern (Objekt 2101 oder LSS)

**Bus in Stop Mode** Obj. 21'01h

nicht nötig

LSS mit Sensor im Netzwerk

Master → Slave

| ID | DLC | Byte1 | Byte2 |
|---|---|---|---|
| 0 | 2 | 02h | 0 |

**Sensor in Konfigurationsmodus**

nicht nötig

Master → Slave

| ID | DLC | Byte1 | | | | | Byte6 | Byte7 | Byte8 |
|---|---|---|---|---|---|---|---|---|---|
| 7E5h | 8 | 40h | Vendor-ID | | | | 0 | 0 | 0 |
| 7E5h | 8 | 41h | Produktcode | | | | 0 | 0 | 0 |
| 7E5h | 8 | 42h | Revisionsnummer | | | | 0 | 0 | 0 |
| 7E5h | 8 | 43h | Serienummer | | | | 0 | 0 | 0 |
| 7E4h | 8 | 44h | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Produktcode, Revisions- und Serienummer sich auf dem Sensor ersichtlich und können im Objekt 1018h ausgelesen werden

**ID ändern**

Master → Slave

| ID | DLC | Byte1 | Index | | Sub. | ID | Byte6 | Byte7 | Byte8 |
|---|---|---|---|---|---|---|---|---|---|
| 600h + ID | 8 | 22h | 01h | 21h | 00h | ID | 0 | 0 | 0 |

| ID | DLC | Byte1 | Index | | Sub. | Byte5 | Byte6 | Byte7 | Byte8 |
|---|---|---|---|---|---|---|---|---|---|
| 580h + ID | 8 | 60h | 01h | 21h | 00h | 0 | 0 | 0 | 0 |

Master → Slave

| ID | DLC | Byte1 | ID | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 | Byte8 |
|---|---|---|---|---|---|---|---|---|---|
| 7E5h | 8 | 11h | ID | 0 | 0 | 0 | 0 | 0 | 0 |

| ID | DLC | Byte1 | Error | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 | Byte8 |
|---|---|---|---|---|---|---|---|---|---|
| 7E4h | 8 | 11h | 00h | 0 | 0 | 0 | 0 | 0 | 0 |

**Änderungen speichern**

Master → Slave

| ID | DLC | Byte1 | Index | | Sub. | „s“ | „a“ | „v“ | „e“ |
|---|---|---|---|---|---|---|---|---|---|
| 600h + ID | 8 | 22h | 25h | 61h | 01h | 73h | 61h | 76h | 65h |

| ID | DLC | Byte1 | Index | | Sub. | Byte5 | Byte6 | Byte7 | Byte8 |
|---|---|---|---|---|---|---|---|---|---|
| 580h + ID | 8 | 60h | 25h | 61h | 01h | 0 | 0 | 0 | 0 |

Master → Slave

| ID | DLC | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 | Byte8 |
|---|---|---|---|---|---|---|---|---|---|
| 7E5h | 8 | 17h | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| ID | DLC | Byte1 | Error | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 | Byte8 |
|---|---|---|---|---|---|---|---|---|---|
| 7E4h | 8 | 17h | 00h | 0 | 0 | 0 | 0 | 0 | 0 |

**Konfigurationsmodus verlassen**

nicht nötig

Master → Slave

| ID | DLC | Byte1 | Byte2 | Byte3 | Byte4 | Byte5 | Byte6 | Byte7 | Byte8 |
|---|---|---|---|---|---|---|---|---|---|
| 7E5h | 8 | 04h | 00h | 0 | 0 | 0 | 0 | 0 | 0 |

**Sensor resetieren oder neu aufstarten**
**=> ACHTUNG: Alte ID verwenden**

Master → Slave

| ID | DLC | Byte1 | Byte2 |
|---|---|---|---|
| 0 | 2 | 81h | ID |

Master → Slave

| ID | DLC | Byte1 | Byte2 |
|---|---|---|---|
| 0 | 2 | 81h | ID |

**Sensor arbeitet mit neuer ID**

**Bootup Message mit neuer ID**

Master → Slave

| ID | DLC | Status |
|---|---|---|
| 700 + ID | 1 | 00h |

Master → Slave

| ID | DLC | Status |
|---|---|---|
| 700 + ID | 1 | 00h |

## 8.4 Baudrate ändern (Objekt 21'00h oder LSS)

| **Bus in Stop Mode** Obj. 21'00h | LSS mit Sensor im Netzwerk |
|---|---|
| nicht nötig | Master → Slave: ID 0 \| DLC 2 \| Byte1 02h \| Byte2 0 |
| **Sensor in Konfigurationsmodus** | |
| nicht nötig | Master → Slave:<br>ID 7E5h \| DLC 8 \| Byte1 40h \| Vendor-ID 5Fh 00h 00h 00h \| Byte6 0 \| Byte7 0 \| Byte8 0<br>ID 7E5h \| DLC 8 \| Byte1 41h \| Produktcode AAh BBh CCh DDh \| Byte6 0 \| Byte7 0 \| Byte8 0<br>ID 7E5h \| DLC 8 \| Byte1 42h \| Revisionsnummer EEh FFh GGh HHh \| Byte6 0 \| Byte7 0 \| Byte8 0<br>ID 7E5h \| DLC 8 \| Byte1 43h \| Serienummer JJh KKh LLh MMh \| Byte6 0 \| Byte7 0 \| Byte8 0<br>ID 7E4h \| DLC 8 \| Byte1 44h \| Byte2 0 \| Byte3 0 \| Byte4 0 \| Byte5 0 \| Byte6 0 \| Byte7 0 \| Byte8 0<br>Produktcode, Revisions- und Serienummer sich auf dem Sensor ersichtlich und können im Objekt 1018h ausgelesen werden |
| **Baudrate ändern (auf 250kBaud)** | |
| Master → Slave:<br>ID 600h + ID \| DLC 8 \| Byte1 22h \| Index 00h 21h \| Sub. 00h \| Baudr. 04h \| Byte6 0 \| Byte7 0 \| Byte8 0<br>Slave → Master:<br>ID 580h + ID \| DLC 8 \| Byte1 60h \| Index 00h 21h \| Sub. 00h \| Byte5 0 \| Byte6 0 \| Byte7 0 \| Byte8 0<br>Baudr.: 04h => 250kBaud | Master → Slave:<br>ID 7E5h \| DLC 8 \| Byte1 13h \| Baudr. 03h \| Byte3 0 \| Byte4 0 \| Byte5 0 \| Byte6 0 \| Byte7 0 \| Byte8 0<br>Slave → Master:<br>ID 7E4h \| DLC 8 \| Byte1 13h \| Error 00h \| Byte3 0 \| Byte4 0 \| Byte5 0 \| Byte6 0 \| Byte7 0 \| Byte8 0<br>Baudr.: 03h => 250kBaud |
| **Änderungen speichern** | |
| Master → Slave:<br>ID 600h + ID \| DLC 8 \| Byte1 22h \| Index 25h 61h \| Sub. 01h \| „s" 73h \| „a" 61h \| „v" 76h \| „e" 65h<br>Slave → Master:<br>ID 580h + ID \| DLC 8 \| Byte1 60h \| Index 25h 61h \| Sub. 01h \| Byte5 0 \| Byte6 0 \| Byte7 0 \| Byte8 0 | Master → Slave:<br>ID 7E5h \| DLC 8 \| Byte1 17h \| Byte2 0 \| Byte3 0 \| Byte4 0 \| Byte5 0 \| Byte6 0 \| Byte7 0 \| Byte8 0<br>Slave → Master:<br>ID 7E4h \| DLC 8 \| Byte1 17h \| Error 00h \| Byte3 0 \| Byte4 0 \| Byte5 0 \| Byte6 0 \| Byte7 0 \| Byte8 0 |
| **Konfigurationsmodus verlassen** | |
| nicht nötig | Master → Slave:<br>ID 7E5h \| DLC 8 \| Byte1 04h \| Byte2 00h \| Byte3 0 \| Byte4 0 \| Byte5 0 \| Byte6 0 \| Byte7 0 \| Byte8 0 |
| **Weitere Busteilnehmer auf neue Baudrate konfigurieren** | |
| **Bus resetieren oder neu aufstarten<br>=> ACHTUNG: Startet mit neuer Baudrate** | |
| Master → Slave:<br>ID 0 \| DLC 2 \| Byte1 81h \| Byte2 00h | Master → Slave:<br>ID 0 \| DLC 2 \| Byte1 81h \| Byte2 00h |
| **Sensor kommuniziert mit neuer Baudrate** | |
| **Bootup Message mit neuer Baudrate** | |
| Slave → Master:<br>ID 700 + ID \| DLC 1 \| Status 00h | Slave → Master:<br>ID 700 + ID \| DLC 1 \| Status 00h |

# 9 Dokument-Revisions-History

V2.10 Manual vor Software Version 2.00 & Revisionsnummer 30200h

V3.10 Manual zu Software ab Version 2.00 & Revisionsnummer 30200h
Überprüfen der Kommunikation und berichtigen der Befehle.
Implementieren der Neuerungen der Software Version 2.00
- LSS Dienst
- dynamisches PDO mapping
- 2. Prozesswert (Temperatur)
- Bootloader implementiert

V3.11 Manual zu Software ab Version 2.00 & Revisionsnummer 30200h
Überarbeitung des Manuals und ergänzen von allgemeinen Informationen