Baumer

Passion for Sensors

# Handbuch

## Absolute Drehgeber mit CANopen Lift-Protokoll

Firmware Version ab 1.00

**Baumer IVO GmbH & Co. KG**
Dauchinger Strasse 58-62
DE-78056 Villingen-Schwenningen
Phone +49 (0)7720 942-0
Fax +49 (0)7720 942-900
info.de@baumerivo.com
www.baumer.com

05.11 · 174.01.035/6
Irrtum sowie Änderungen in
Technik und Design vorbehalten.

## Inhalt


Seite

**1. Einleitung 3**
**1.1. Lieferumfang 3**
**1.2. Produktzuordnung 3**
**2. Sicherheits- und Betriebshinweise 4**
**3. CAN-Bus und CANopen-Kommunikation 5**
**3.1. CAN-Bus 5**
3.1.1. CAN-Bus-Eigenschaften 5
**3.2. CANopen 6**
**3.3. CANopen im Aufzugsbau 6**
**3.4. CANopen-Kommunikation 7**
3.4.1. Kommunikationsprofil 7
3.4.2. CANopen Meldungsaufbau 7
3.4.3. Servicedaten-Kommunikation 8
3.4.4. Prozessdaten-Kommunikation 9
3.4.5. Emergency-Dienst 10
3.4.6. Netzwerkmanagement-Dienste 11
3.4.7. Layer Setting Services 13
**3.5. CANopen Lift Drehgeber nach Appl. Profil DSP417 17**
3.5.1. Objekte Übersicht 17
3.5.2. Ausführliche Objektbeschreibung 20
**4. Diagnose und Wissenswertes 32**
**4.1. Fehlerdiagnose Feldbus-Kommunikation 32**
**4.2. Fehlerdiagnose über Feldbus 32**
**4.3. Wissenswertes zum Sensor 33**
**5. Applikationen 34**
**5.1. Objekte setzen und lesen 34**
**5.2. Konfiguration 35**
**5.3. Betrieb 36**
**6. Anschlussbelegung und Inbetriebnahme 38**
**6.1. Mechanischer Anbau 38**
**6.2. Elektrischer Anschluss 38**
6.2.1. Beschreibung der Anschlüsse 38
6.2.2. Anschlussbelegung M12-Stecker 38
6.2.3. Anschlussbelegung D-SUB Stecker 39
**6.3. Anzeigeelemente (Statusanzeige) 39**

**Haftungsausschluss**
Diese Schrift wurde mit grosser Sorgfalt zusammengestellt. Fehler lassen sich jedoch nicht immer vollständig ausschliessen. Baumer IVO GmbH & Co. KG übernimmt daher keine Garantien irgendwelcher Art für die in dieser Schrift zusammengestellten Informationen. In keinem Fall haftet Baumer IVO GmbH & Co. KG oder der Autor für irgendwelche direkten oder indirekten Schäden, die aus der Anwendung dieser Informationen folgen.

Wir freuen uns jederzeit über Anregungen, die der Verbesserung dieses Handbuchs dienen können.

# 1. Einleitung

## 1.1. Lieferumfang

Bitte prüfen Sie vor der Inbetriebnahme die Vollständigkeit der Lieferung.
Je nach Ausführung und Bestellung können zum Lieferumfang gehören:

- Drehgeber
- CD mit Beschreibungsdateien und Handbuch (auch über das Internet zum Download verfügbar)

## 1.2. Produktzuordnung

**Wellen-Drehgeber**

| Produkt | Produkt-Code | Device Name | Eds-Datei | Produktfamilie |
|---|---|---|---|---|
| GXP5W | 0x28 | GXP5 | GXP5_417.eds | Multiturn |
| X 700 | 0x28 | GXP5 | GXP5_417.eds | Multiturn |

Bei den Drehgebern mit Product Code 0x28 handelt es sich um Absolute Drehgeber mit CANopen Schnittstelle und implementiertem Applikationsprofil  DSP417 (Application Profile for Lift Control Systems).



# 2. Sicherheits- und Betriebshinweise

**Zusätzliche Informationen**
- Das Handbuch ist eine Ergänzung zu bereits vorhandenen Dokumentationen (Kataloge, Datenblätter und Montageanleitungen).
- Die Anleitung muss unbedingt vor Inbetriebnahme gelesen werden.

**Bestimmungsgemäßer Gebrauch**
- Der Drehgeber ist ein Präzisionsmessgerät. Er dient ausschließlich zur Erfassung von Winkelpositionen und Umdrehungen, der Aufbereitung und Bereitstellung der Messwerte als elektrische Ausgangssignale für das Folgegerät. Der Drehgeber darf ausschließlich zu diesem Zweck verwendet werden.

**Inbetriebnahme**
- Einbau und Montage des Drehgebers darf ausschließlich durch eine Elektrofachkraft erfolgen.
- Betriebsanleitung des Maschinenherstellers beachten.

**Sicherheitshinweise**
- Vor Inbetriebnahme der Anlage alle elektrischen Verbindungen überprüfen.
- Wenn Montage, elektrischer Anschluss oder sonstige Arbeiten am Drehgeber und an der Anlage nicht fachgerecht ausgeführt werden, kann es zu Fehlfunktion oder Ausfall des Drehgebers führen.
- Eine Gefährdung von Personen, eine Beschädigung der Anlage und eine Beschädigung von Betriebseinrichtungen durch den Ausfall oder Fehlfunktion des Drehgebers muss durch geeignete Sicherheitsmaßnahmen ausgeschlossen werden.
- Drehgeber darf nicht außerhalb der Grenzwerte betrieben werden (siehe weitere Dokumentationen).

*Bei Nichtbeachtung der Sicherheitshinweise kann es zu Fehlfunktionen, Sach- und Personenschäden kommen!*

**Transport und Lagerung**
- Transport und Lagerung ausschließlich in Originalverpackung.
- Drehgeber nicht fallen lassen oder größeren Erschütterungen aussetzen.

**Montage**
- Schläge oder Schocks auf Gehäuse und Welle vermeiden.
- Gehäuse nicht verspannen.
- Keine starre Verbindung von Drehgeberwelle und Antriebswelle vornehmen.
- Drehgeber nicht öffnen oder mechanisch verändern.

*Welle, Kugellager, Glasscheibe oder elektronische Teile können beschädigt werden. Die sichere Funktion ist dann nicht mehr gewährleistet.*

**Elektrische Inbetriebnahme**
- Drehgeber elektrisch nicht verändern.
- Keine Verdrahtungsarbeiten unter Spannung vornehmen.
- Der elektrische Anschluss darf unter Spannung nicht aufgesteckt oder abgenommen werden.
- Die gesamte Anlage EMV gerecht installieren. Einbauumgebung und Verkabelung beeinflussen die EMV des Drehgebers. Drehgeber und Zuleitungen räumlich getrennt oder in großem Abstand zu Leitungen mit hohem Störpegel (Frequenzumrichter, Schütze usw.) verlegen.
- Bei Verbrauchern mit hohen Störpegeln separate Spannungsversorgung für den Drehgeber bereitstellen.
- Drehgebergehäuse und die Anschlusskabel vollständig schirmen.
- Drehgeber an Schutzerde (PE) anschließen. Geschirmte Kabel verwenden. Schirmgeflecht muss mit der Kabelverschraubung oder Stecker verbunden sein. Anzustreben ist ein beidseitiger Anschluss an Schutzerde (PE), Gehäuse über den mechanischen Anbau, Kabelschirm über die nachfolgenden angeschlossenen Geräte. Bei Problemen mit Erdschleifen mindestens eine einseitige Erdung.

*Bei Nichtbeachtung kann es zu Fehlfunktionen, Sach- und Personenschäden kommen!*

# 3. CAN-Bus und CANopen-Kommunikation

## 3.1. CAN-Bus

Der CAN-Bus (CAN: Controller Area Network) wurde ursprünglich von Bosch und Intel für die schnelle, kostengünstige Datenübertragung in der Kraftfahrzeug-Technik entwickelt. Der CAN-Bus wird heute auch in der industriellen Automatisierung verwendet.
Der CAN-Bus ist ein Feldbus (die Normen werden durch die Vereinigung CAN in Automation (CiA) festgelegt) über den Geräte, Aktoren und Sensoren verschiedener Hersteller miteinander kommunizieren.

### 3.1.1. CAN-Bus-Eigenschaften

- Datenrate von 1 MBaud bei einer Netzausdehnung bis zu 40 m
- Beidseitig abgeschlossenes Netzwerk
- Busmedium ist Twisted-Pair-Kabel
- Echtzeitfähigkeit: Definierte max. Wartezeit für Nachrichten hoher Priorität.
- Theoretisch 127 Teilnehmer an einem Bus, physikalisch aber nur 32 (durch den Treiber bedingt).
- Sicherstellung netzweiter Datenkonsistenz. Gestörte Nachrichten werden für alle Netzknoten als fehlerhaft bekannt gemacht.
- Nachrichtenorientierte Kommunikation
  Die Nachricht wird mit einer Nachrichtenkennung (Identifier) gekennzeichnet. Alle Netzknoten prüfen anhand des Identifier, ob die Nachricht für sie relevant ist.
- Broadcasting, Multicasting
  Alle Netzknoten erhalten gleichzeitig jede Nachricht. Daher ist eine Synchronisation möglich.
- Multi-Master-Fähigkeit
  Jeder Teilnehmer im Feldbus kann selbstständig Daten senden und empfangen, ohne dabei auf eine Priorität der Master angewiesen zu sein. Jeder kann seine Nachricht beginnen, wenn der Bus nicht belegt ist. Bei einem gleichzeitigen Senden von Nachrichten setzt sich der Teilnehmer mit der höchsten Priorität durch.
- Priorisierung von Nachrichten
  Der Identifier setzt die Priorität der Nachricht fest. Dadurch können wichtige Nachrichten schnell über den Bus übertragen werden.
- Restfehlerwahrscheinlichkeit
  Sicherungsverfahren im Netzwerk reduzieren die Wahrscheinlichkeit einer unentdeckten, fehlerhaften Datenübertragung auf unter $10^{-11}$. Praktisch kann von einer 100% sicheren Übertragung ausgegangen werden.
- Funktionsüberwachung
  Lokalisation fehlerhafter oder ausgefallener Stationen. Das CAN-Protokoll beinhaltet eine Funktionsüberwachung von Netzknoten. Netzknoten, die fehlerhaft sind, werden in ihrer Funktion eingeschränkt oder ganz vom Netzwerk abgekoppelt.
- Datenübertragung mit kurzer Fehler-Erholzeit
  Durch mehrere Fehlererkennungsmechanismen werden verfälschte Nachrichten mit großer Wahrscheinlichkeit erkannt. Wird ein Fehler erkannt, wird die Nachrichtensendung automatisch wiederholt.

Im CAN-Bus sind mehrere Netzwerkteilnehmer über ein Buskabel miteinander verbunden. Jeder Netzwerkteilnehmer kann Nachrichten senden und empfangen. Die Daten zwischen den Netzwerk-Teilnehmern werden seriell übertragen.

Netzwerkteilnehmer Beispiele für CAN-Bus-Geräte sind:
- Automatisierungsgeräte, z. B. SPS
- PCs
- Ein- /Ausgangsmodule
- Antriebssteuerungen
- Analysegeräte, z. B. ein CAN-Monitor
- Bedien- und Eingabegeräte als Mensch-Maschine Schnittstelle HMI (HMI, Human Machine Interface)
- Sensoren und Aktoren

## 3.2. CANopen

Unter technischer Leitung des Steinbeis Transferzentrums für Automatisierung wurde auf der Basis der Schicht 7 Spezifikation CAL (CAN-Application Layer) das CANopen-Profil entwickelt. Im Vergleich zu CAL sind in CANopen nur die für diesen Einsatz geeigneten Funktionen enthalten. CANopen stellt somit eine für die Anwendung optimierte Teilmenge von CAL dar und ermöglicht dadurch vereinfachten Systemaufbau und den Einsatz vereinfachter Geräte. CANopen ist optimiert für den schnellen Datenaustausch in Echtzeitsystemen.
Die Organisation CAN in Automation (CiA) ist zuständig für die geltenden Normen der entsprechenden Profile.
CANopen ermöglicht:
• Einfachen Zugriff auf alle Geräte- und Kommunikationsparameter
• Synchronisation von mehreren Geräten
• Automatische Konfiguration des Netzwerkes
• zyklischen und ereignisgesteuerten Prozessdatenverkehr

CANopen besteht aus vier Kommunikationsobjekten (COB) mit unterschiedlichen Eigenschaften:
• Prozess-Daten-Objekte für Echtzeitdaten (PDO)
• Service-Daten-Objekte für Parameter- und Programmübertragung (SDO)
• Netzwerk Management (NMT, Heartbeat)
• Vordefinierte Objekte (für Synchronisation, Notfallnachricht)

Alle Geräte- und Kommunikationsparameter sind in einem Objektverzeichnis gegliedert. Ein Objekt umfasst Name des Objekts, Datentyp, Anzahl Subindexe, Struktur der Parameter und die Adresse. Nach CiA ist dieses Objektverzeichnis in drei verschiedene Teile unterteilt: Kommunikationsprofil, Geräteprofil und ein herstellerspezifisches Profil. (siehe Objektverzeichnis)

## 3.3. CANopen im Aufzugsbau

Bei CANopen handelt es sich um eine standardisierte Anwendung für verteilte, industrielle Automatisierungssysteme auf Basis von CAN sowie des Kommunikationsstandards CAL. CANopen ist ein Standard der CAN-in-Automation (CiA) und hat bereits kurz nach seiner Verfügbarkeit eine sehr weite Verbreitung gefunden. In Europa kann CANopen als der maßgebliche Standard für die Realisierung von industriellen CAN-basierten Systemlösungen betrachtet werden.
Die Federführung wurde der Organisation CAN-in-Automation übergeben. Diese koordiniert alle Arbeiten am System CANopen weltweit. Innerhalb der CiA wurde eine Spezial Interest Group (SIG) "Lift" gegründet. Aufgabe dieser SIG ist die Überprüfung der vorhandenen Profile auf eine Eignung für den Aufzugsbau sowie deren Erweiterung oder Neudefinition. Innerhalb der SIG "Lift" wurden verschiedene Arbeitsgruppen für die einzelnen Profile festgelegt.

## 3.4. CANopen-Kommunikation

### 3.4.1. Kommunikationsprofil

Die Kommunikation zwischen den Netzwerkteilnehmern und dem Master (PC / Steuerung) erfolgt über Objektverzeichnisse und Objekte. Die Objekte werden über einen 16bit-Index adressiert. Das CANopen-Kommunikationsprofil DS 301 standardisiert die verschiedenen Kommunikationsobjekte. Dementsprechend werden sie in mehrere Gruppen unterteilt:

- Prozessdatenobjekte PDO (process data object) zur Echtzeitübertragung von Prozessdaten
- Servicedatenobjekte SDO (service data object) für den Schreib- und Lesezugriff auf das Objektverzeichnis
- Objekte zur Synchronisation und Fehleranzeige von CAN-Teilnehmern:
  - SYNC-Objekt (synchronisation object) zur Synchronisation von Netzwerkteilnehmern
  - EMCY-Objekt (emergency object) zur Fehleranzeige eines Gerätes oder seiner Peripherie
- Netzwerk-Management NMT (network management) zur Initialisierung und Netzwerksteuerung
- Layer Setting Services LSS zur Konfiguration mittels Seriennummer, Revisionsnummer usw. inmitten eines vorhandenen Netzwerks

### 3.4.2. CANopen Meldungsaufbau

Der erste Teil einer Meldung ist die COB-ID (Identifier).
Aufbau der 11-Bit COB-ID:

| Funktions Code | | | | Node-ID | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 4 Bit FunktionsCode | | | | 7 Bit Node-ID | | | | | | |
| | | | | | | | | | | |

Der Funktionscode gibt Aufschluss über die Art der Meldung und die Priorität
Je niedriger die COB-ID, desto höher die Priorität der Meldung.

Broadcast Meldungen:

| Funktionscode | COB-ID |
|---|---|
| NMT | 0 |
| SYNC | 80h |

Peer to Peer Meldungen:

| Funktionscode | COB-ID |
|---|---|
| Emergency | 80h + Node-ID |
| PDO1 (tx)[1)] | 17Bh+ Lift number * 10 + PosUnit |
| | |
| SDO (tx)[1)] | 580h + Node-ID |
| SDO (rx)[1)] | 600h + Node-ID |
| Heartbeat | 700h + Node-ID |
| LSS (tx)[1)] | 7E4h |
| LSS (rx)[1)] | 7E5h |

1): (tx) und (rx) aus der Sicht des Drehgebers

Die Node-ID kann über den CANopen-Bus zwischen 1 und 127 frei gewählt werden (wenn Drehschalter = 0).
Die Drehgeber werden mit Node-ID 1 ausgeliefert.
Eine Änderung erfolgt mit dem Service Daten Objekt 2101h oder über LSS.

Ein CAN-Telegramm besteht aus der COB-ID und bis zu 8 Byte Daten:

| COB-ID | DLC | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
|---|---|---|---|---|---|---|---|---|---|
| Xxx | x | xx | xx | xx | xx | xx | xx | xx | xx |

Die genauen Telegramme werden später noch ausführlich aufgeführt.

### 3.4.3. Servicedaten-Kommunikation

Die Servicedatenobjekte entsprechen den Normen von CiA. Über Index und Subindex kann auf ein Objekt zugegriffen werden. Die Daten können angefordert oder gegebenenfalls ins Objekt geschrieben werden.

**Allgemeines zu den SDO**
Aufbau eines **SDO-Telegramms**:

| COB-ID | DLC | Kommando | Objekt L | Objekt H | Subindex | Data 0 | Data 1 | Data 2 | Data 3 |
|---|---|---|---|---|---|---|---|---|---|

Eine SDO-**COB-ID** setzt sich folgendermaßen zusammen:
Master -> Drehgeber : 600h + Node-ID
Drehgeber -> Master : 580h + Node-ID

**DLC** (Data length code) bezeichnet die Länge des Telegramms. Diese setzt sich wie folgt zusammen:
1 Byte Kommando + 2 Byte Objekt + 1 Byte Subindex + Anzahl Datenbyte (0..4).

Das **Kommando-Byte** legt fest, ob Daten gelesen oder gesetzt werden und um wie viele Datenbyte es sich handelt:

| SDO Kommando | Beschreibung | Datenlänge | |
|---|---|---|---|
| 22h | Download Request | Max. 4 Byte | Parameter an Drehgeber senden |
| 23h | Download Request | 4 Byte | |
| 2Bh | Download Request | 2 Byte | |
| 2Fh | Download Request | 1 Byte | |
| | | | |
| 60h | Download Response | - | Bestätigung der Übernahme an Master |
| 40h | Upload Request | - | Parameter vom Drehgeber anfordern |
| | | | |
| 42h | Upload Response | Max. 4 Byte | Parameter an Master mit max. 4 Byte |
| 43h | Upload Response | 4 Byte | |
| 4Bh | Upload Response | 2 Byte | |
| 4Fh | Upload Response | 1 Byte | |
| | | | |
| 80h | Abort Message | - | Drehgeber meldet Fehlercode an Master |

Eine **Abort Message** zeigt einen Fehler in der CAN-Kommunikation an. Das SDO Kommando-Byte ist 80h. Objekt und Subindex sind die des gewünschten Objektes. In Byte 5..8 steht der Fehler-Code.

| ID | DLC | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Byte 8 |
|---|---|---|---|---|---|---|---|---|---|
| 580h + Node-ID | 8 | 80h | Objekt L | Objekt H | Subindex | ErrByte 0 | ErrByte 1 | ErrByte 2 | ErrByte 3 |

Byte 8..5 ergibt die SDO Abort Meldung (Byte 8 = MSB).
Folgende Meldungen werden unterstützt:

05040001h : Command Byte wird nicht unterstützt
06010000h : Falscher Zugriff auf ein Objekt
06010001h : Lesezugriff auf Write Only
06010002h : Schreibzugriff auf Read Only
06020000h : Objekt wird nicht unterstützt
06090011h : Subindex wird nicht unterstützt
06090030h : Wert außerhalb der Limite
06090031h : Wert zu groß
08000000h : Genereller Error
08000020h : Falsche Speichersignatur ("save")
08000021h : Daten können nicht gespeichert werden

**Beispiele SDO**

**Anfrage** eines Wertes vom Master beim Slave
Eine häufige Anfrage wird diejenige nach der Position sein. → Objekt 6383h Subindex 1

| COB-ID | DLC | Kommando | Objekt L | Objekt H | Subindex | Data 0 | Data 1 | Data 2 | Data 3 |
|---|---|---|---|---|---|---|---|---|---|
| 600h+Node-ID | 8 | 40h | 83h | 63h | 01 | x | x | x | x |

**Antwort** des Slaves **auf die Anfrage** eines Wertes
Die Position ist 4 Byte lang, die genauen Werte sind unter Objekt 6383h Subindex 1 zu finden.

| COB-ID | DLC | Kommando | Objekt L | Objekt H | Subindex | Data 0 | Data 1 | Data 2 | Data 3 |
|---|---|---|---|---|---|---|---|---|---|
| 580h+Node-ID | 8 | 43h | 83h | 63h | 01 | a | b | c | d |

**Schreiben** eines Wertes vom Master in den Slave
Position setzen kann mit Preset erfolgen. → Objekt 6382h

| COB-ID | DLC | Kommando | Objekt L | Objekt H | Subindex | Data 0 | Data 1 | Data 2 | Data 3 |
|---|---|---|---|---|---|---|---|---|---|
| 600h+Node-ID | 8 | 22h | 82h | 63h | 01 | a | b | c | d |

**Antwort** des Slaves **auf das Schreiben** eines Wertes

| COB-ID | DLC | Kommando | Objekt L | Objekt H | Subindex | Data 0 | Data 1 | Data 2 | Data 3 |
|---|---|---|---|---|---|---|---|---|---|
| 580h+Node-ID | 8 | 60h | 82h | 63h | 01 | 0 | 0 | 0 | 0 |

### 3.4.4. Prozessdaten-Kommunikation

Prozessdatenobjekte dienen dem Echtzeit-Datenaustausch für Prozessdaten wie zum Beispiel die Position oder den Betriebsstatus. PDO's können synchron oder zyklisch (asynchron) gesendet werden. Der Drehgeber unterstützt das PDO1 .. Dieses liefert die aktuelle Position des Drehgebers sowie dessen Geschwindigkeit. Es ist in den Objekten 1906h und 1B06hh festgelegt.

**Synchron**
Um die Prozessdaten synchron zu senden, muss im Objekt 1906h Subindex 2 ein Wert zwischen 1 und F0h (=240) eingetragen werden. Wenn nun der Wert 3 beträgt, wird das PDO auf jedes dritte Sync-Telegramm gesendet (beim Wert 1 wird auf jedes Sync-Telegramm gesendet), solange im Objekt 2800h 0 eingetragen ist. Ist dort zum Beispiel eine 5 eingetragen, wird das PDO nach wie vor auf jedes dritte Sync-Telegramm geschrieben, insgesamt aber nur 5 mal. Dem entsprechend folgt auf das 15. Sync-Telegramm das letzte PDO. Der Zähler für die Anzahl der zu übertragenden PDO´s wird bei einer Positions-änderung oder das NMT-Reset zurückgesetzt. D.h. die Position wird, falls sie sich nicht ändert, 5 mal gesendet. Ändert sich die Position, wird sie wieder 5 mal gesendet.

Im synchronen Betrieb werden die PDO vom Master über das Sync-Telegramm angefordert:

| Byte 0 | Byte 1 |
|---|---|
| COB-ID = 80 | 0 |

**Zyklisch (Asynchron)**
Sollen die PDO's zyklisch gesendet werden, muss ins Objekt 1906h Subindex 2 der Wert FFh geschrieben werden. Zusätzlich muss im gleichen Objekt Subindex 5 die Zykluszeit in Millisekunden eingetragen werden. Die eingetragene Zeit wird auf 1ms aufgerundet. Wird der Wert 0ms gespeichert, werden die PDO's nicht gesendet. Die Funktion ist ausgeschaltet.
Eine weitere Möglichkeit bringt das Objekt 2800h: Beträgt der Wert 0, läuft das Zyklische Senden wie oben beschrieben. Beträgt der Wert 1, wird zyklisch geprüft ob eine Änderung des Wertes vorliegt. Wenn nicht, wird nicht gesendet. Beträgt der Wert 4, wird bei jedem Zyklus, falls eine Änderung besteht, das PDO viermal gesendet.

**Übersicht**
In nachfolgender Tabelle werden die verschiedenen Sende-Arten von PDO's zusammengefasst:

| 1906h | | 2800h | Kurzbeschreibung |
|---|---|---|---|
| Sub2 | Sub5 | | |
| FFh | 3ms | 0 | Zyklisches Senden alle 3 ms |
| FFh | 5ms | 2 | Alle 5ms wird das PDO doppelt gesendet, falls eine Änderung vorliegt. |
| FFh | 0ms | 0 | PDO senden ausgeschaltet |
| FFh | 0ms | 2 | PDO senden ausgeschaltet |
| 3 | xxx | 0 | Bei jedem dritten Sync-Telegramm senden |
| 3 | xxx | 2Bh | Auf jedes dritte Sync-Telegramm, aber gesamt nur 43 mal (=2Bh). |

**PDO**
PDO1 Telgrammaufbau:

| ID | DLC | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 |
|---|---|---|---|---|---|---|---|
| 18Ch | 4 | Xx | Xx | Xx | Xx | Xx | Xx |

ID : 17Bh+ Lift number * 10 + PosUnit
Länge : 6 DataByte
Byte1.. 4 : Aktuelle Position in Inkrementen
Byte 5..6 : Geschwindigkeit

### 3.4.5. Emergency-Dienst

Interne Gerätefehler oder Busprobleme lösen eine Emergency-Meldung aus:

| COB-ID | DLC | Byte0 | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 |
|---|---|---|---|---|---|---|---|---|---|
| 80h+Node-ID | 8 | Error Code | | Error Register | Alarms 63C7h | | Warning 63C5h | | n.u. |
| | | 00h | 01h | 1001h | | | | | |

**Byte 0..1: Error Codes**

| Error Code (hex) | Meaning |
|---|---|
| 0000 | Error Reset or No Error |
| 1000 | Generic Error |

**Byte 2: Error Register**

| Bit | Meaning |
|---|---|
| 0 | Generic Error |

**Byte 3..4 Alarms**

| Bit | Meaning | Wert = 0 | Wert = 1 |
|---|---|---|---|
| 0 | Position error aktiv | Nein | Ja |

**Byte 5..6 Warning**

| Bit | Meaning | Wert = 0 | Wert = 1 |
|---|---|---|---|
| 2 | CPU watchdog status | OK | Reset ausgeführt |
| 4 | Battery charge | OK | Ladung zu tief |

**Byte 7: nicht verwendet**

### 3.4.6. Netzwerkmanagement-Dienste

Das Netzwerkmanagement kann in zwei Gruppen unterteilt werden:
Mit den NMT-Diensten für die **Gerätekontrolle** können die Busteilnehmer initialisiert, gestartet und gestoppt werden.
Zusätzlich gibt es die NMT-Dienste zur **Verbindungsüberwachung**.

**Beschreibung der NMT-Kommandos**
Die Kommandos werden als unbestätigte Objekte übertragen und sind folgendermaßen aufgebaut:

| Byte 0 | Byte 1 | Byte 2 |
|---|---|---|
| COB-ID = 0 | Kommando Byte | Knoten Nummer |

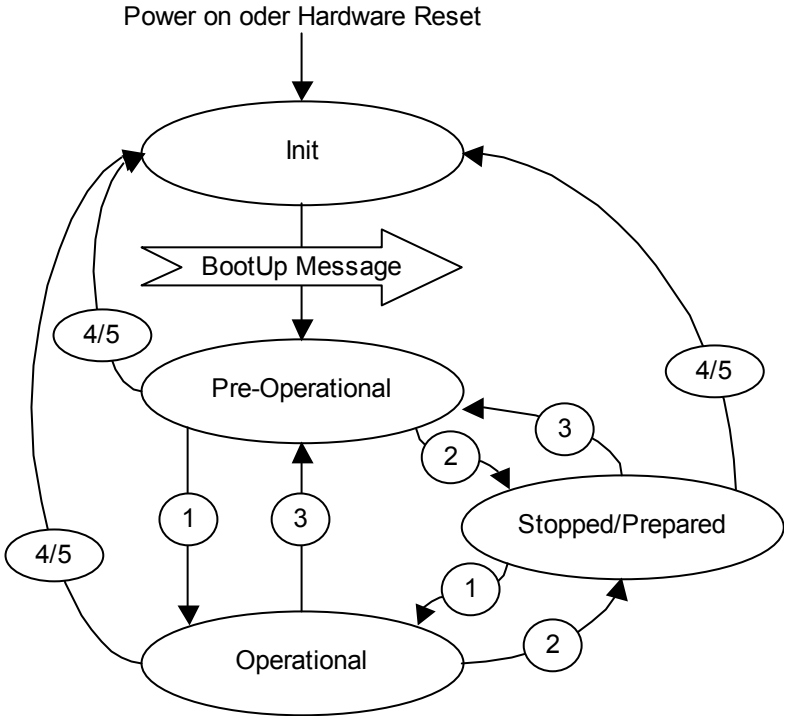
**COB-ID** für NMT-Kommandos ist immer Null. Die Node-ID wird in Byte 2 des NMT-Kommandos übertragen.

**Kommando Byte**

| Kommando Byte | Beschreibung | In State Event Zeichnung |
|---|---|---|
| 01h | Start Remote Node | 1 |
| 02h | Stop Remote Node | 2 |
| 80h | Enter Pre-Operational Mode | 3 |
| 81h, 82h | Reset Remote Node | 4, 5 |

Die **Knotennummer** entspricht der Node-ID des gewünschten Teilnehmers. Mit Knotennummer = 0 werden alle Teilnehmer angesprochen.

**NMT State Event**
Nach dem Initialisieren ist der Drehgeber im Pre-Operational Mode. In diesem Zustand können SDO Parameter gelesen und geschrieben werden. Um PDO Parameter anzufordern, muss der Drehgeber zuerst in den Zustand Operational Mode gefahren werden.



**Die verschiedenen NMT Zustände**

**Init**
Nach dem Initialisieren meldet sich der Drehgeber mit einer BootUp Meldung am CAN-Bus. Danach geht der Drehgeber automatisch in den Zustand PreOperational Mode über.

Die COB-ID der BootUp Meldung setzt sich aus 700h und der Node-ID zusammen.

| COB-ID | Byte 0 |
|---|---|
| 700h + Node-ID | 00 |

**Pre-Operational Mode**
Im Pre-Operational Mode können SDO gelesen und geschrieben werden.

**Operational Mode**
Im Zustand Operational Mode sendet der Drehgeber die gewünschten PDO's. Zudem können SDO gelesen und geschrieben werden.

**Stopped oder Prepared Mode**
Im Stopped Mode ist nur NMT Kommunikation möglich. Es können keine SDO Parameter gelesen oder gesetzt werden. LSS ist nur im Stopped Mode möglich.

**Zustandswechsel**

**Start Remote Node (1)**
Mit dem Startbefehl wird der Drehgeber in den Zustand Operational Mode gebracht.

| COB-ID | Kommando Byte | Knoten Nummer |
|---|---|---|
| 0 | 1h | 0..127 |

**Stop Remote Node (2)**
Mit dem Stoppbefehl wird der Drehgeber in den Zustand Stopped oder Prepared Mode gebracht.

| COB-ID | Kommando Byte | Knoten Nummer |
|---|---|---|
| 0 | 2h | 0..127 |

**Enter Pre-Operational Mode (3)**
Wechsle in den Zustand Pre-Operational Mode.

| COB-ID | Kommando Byte | Knoten Nummer |
|---|---|---|
| 0 | 80h | 0..127 |

**Reset Remote Node (4) oder Reset Kommunikation (5)**
Mit dem Reset-Befehl wird der Drehgeber neu initialisiert.
Reset Remote Node (4):

| COB-ID | Kommando Byte | Knoten Nummer |
|---|---|---|
| 0 | 81h | 0..127 |

Reset Kommunikation (5):

| COB-ID | Kommando Byte | Knoten Nummer |
|---|---|---|
| 0 | 82h | 0..127 |

**Heartbeat**
Im CAL ist ein "Heartbeat"-Prinzip definiert, mit welchem man die Kommunikationsfähigkeit von CANopen-Geräten überwachen kann. Es kann sicher gestellt werden, dass der Master auf einen Ausfall reagieren könnte.
Im Objekt 1017h "Producer Heartbeat Time" kann die Zeit des Heartbeats definiert werden. Sobald der Wert bestätigt wurde, beginnt der Heartbeat zu senden.

| COB-ID | Data/Remote | Byte 0 |
|---|---|---|
| 701h | d | 7Fh (127d) |

Die Heartbeat-Meldungen bestehen aus der COB-ID und einem Byte. In diesem Byte wird der NMT-Zustand überliefert.

0: BootUp-Event
4: Stopped
5: Operational
127: Pre-Operational

D.h. der Drehgeber befindet sich im Pre-Operational Modus (7Fh = 127).

### 3.4.7. Layer Setting Services

Im Frühling 2000 wurde von CiA ein neues Protokoll entworfen, um ein einheitliches Auftreten zu gewährleisten. Beschrieben ist das Vorgehen unter
*Layer Setting Services and Protokoll, CiA Draft Standard Proposal 305* (LSS).
Der Drehgeber wird von uns standardmäßig mit der Node-ID 1 und der Baudrate 50 kBaud ausgeliefert.
Es können mehrere Drehgeber mit der selben Node-ID an das Bussystem angeschlossen werden. Um nun die einzelnen Drehgeber ansprechen zu können, wird LSS verwendet.
Jeder Drehgeber besitzt eine eindeutige Seriennummer und wird über diese Nummer angesprochen. Also können beliebig viele Drehgeber mit gleicher Node-ID an ein Bussystem angeschlossen werden und dann über LSS initialisiert werden. Es können sowohl die Node-ID als auch die Baudrate neu gesetzt werden. LSS kann nur im **Stopped Mode** ausgeführt werden.

**Meldungsaufbau**

**COB-ID:**
Master → Slave : 2021 = 7E5h
Master ← Slave : 2020 = 7E4h
Nach der COB-ID wird ein LSS command specifier gesandt.
Danach werden bis zu sieben Datenbyte angehängt.

| COB-ID | cs | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 |
|---|---|---|---|---|---|---|---|---|

**Switch Mode Global**

| 7E5h → | 04h | Mode | Reserved |
|---|---|---|---|

Mode : 0 → Operationsmode
1 → Konfigurationsmode

**Switch Mode Selektiv**
Mit folgendem Ablauf kann ein ganz bestimmter Drehgeber im Bussystem angesprochen werden.

| 7E5h → | 40h | VendorId | Reserved |
|---|---|---|---|
| 7E5h → | 41h | ProductCode | reserved |
| 7E5h → | 42h | RevisionNumber | reserved |
| 7E5h → | 43h | SerialNumber | reserved |
| 7E4h ← | 44h | Mode | reserved |

VendorId : ECh
ProductCode : Interner Produkt Code für den jeweiligen Drehgeber
RevisionNumber : Aktuelle Revisionsnummer des Drehgebers
SerialNumber : Eindeutige, fortlaufende Seriennummer
Mode : Antwort des Drehgebers ist der neue Mode (0=Operationsmode; 1=Konfigurationsmode)

**Node-ID setzen**

| | | | |
|---|---|---|---|
| 7E5h → | 11h | Node-ID | reserved |

| | | | | |
|---|---|---|---|---|
| 7E4h ← | 11h | ErrCode | Spec Error | reserved |

Node-ID : Die neue Node-ID des Drehgebers
ErrorCode : 0=OK; 1=Node-ID außerhalb des Bereiches; 2..254=reserved; 255→specificError
SpecificError : Falls ErrorCode=255 → Applikationsspezifischer Errorcode.

**BitTiming setzen**

| | | | | |
|---|---|---|---|---|
| 7E5h → | 13h | tableSel | tableInd | reserved |

| | | | | |
|---|---|---|---|---|
| 7E4h ← | 13h | ErrCode | SpecError | reserved |

TableSel : Selektiert die BitTiming Tabelle 0 : Standard CiA Bit Timing Tabelle
1..127 : Reserviert für CiA
128..255 : Herstellerspezifische Tabellen
TableInd : BitTiming Eintrag in selektierter Tabelle (siehe Tabelle unten).
ErrorCode : 0=OK; 1=BitTiming außerhalb des Bereiches; 2..254=reserved; 255→SpecificError
SpecificError : Falls ErrorCode=255 → Applikationsspezifischer Errorcode.

**Standard CiA Tabelle**

| Baudrate | Tabellen Index |
|---|---|
| 1000 kBaud | 0 |
| 800 kBaud | 1 |
| 500 kBaud | 2 |
| 250 kBaud | 3 |
| 125 kBaud | 4 |
| 100 kBaud | 5 |
| 50 kBaud | 6 |
| 20 kBaud | 7 |
| 10 kBaud | 8 |

**Speichern des Konfigurationsprotokoll**
Dieses Protokoll speichert die Konfigurationsparameter im EEPROM.

| | | |
|---|---|---|
| 7E5h → | 17h | reserved |

| | | | | |
|---|---|---|---|---|
| 7E4h ← | 17h | ErrCode | SpecError | Reserved |

ErrorCode : 0=OK;1=Speichern nicht unterstützt;2=Zugriffsfehler;3..254=reserved;255→SpecificError
SpecificError : Falls ErrorCode=255 → Applikationsspezifischer Errorcode.

**Aktiviere BitTiming Parameter**
Die neuen BitTiming Parameter werden mit dem command specifier 21 aktiviert.

| | | | |
|---|---|---|---|
| 7E5h → | 15h | Switch Delay | Reserved |

Switch Delay : Verzögerung des Resets im Slave in ms.
Nach der Verzögerungszeit meldet sich der Drehgeber mit der neuen Baudrate an.

**VendorId anfordern**
VendorId eines selektierten Drehgebers anfordern

| | | |
|---|---|---|
| 7E5h → | 5Ah | reserved |

| | | | |
|---|---|---|---|
| 7E4h ← | 5Ah | 32 Bit Vendor ID | reserved |

VendorID : = ECh

**Produktcode anfordern**
Produktcode eines selektierten Drehgebers anfordern

| | | |
|---|---|---|
| 7E5h → | 5Bh | reserved |

| | | | |
|---|---|---|---|
| 7E4h ← | 5Bh | ProdukteCode | reserved |

Produktcode : Herstellerabhängiger Produktcode

**Revisionsnummer anfordern**
Revisionsnummer eines selektierten Drehgebers anfordern

| | | |
|---|---|---|
| 7E5h → | 5Ch | reserved |

| | | | |
|---|---|---|---|
| 7E4h ← | 5Ch | 32 Bit Revisionsnummer | reserved |

Revisionsnummer : aktuelle Revision

**Seriennummer anfordern**
Seriennummer eines selektierten Drehgebers anfordern

| | | |
|---|---|---|
| 7E5h → | 5Dh | reserved |

| | | | |
|---|---|---|---|
| 7E4h ← | 5Dh | 32 Bit Seriennummer | reserved |

Seriennummer : eindeutige fortlaufende Seriennummer des Drehgebers

**Bereichsanfrage**
Drehgeber können auch in einem gewissen Bereich gesucht werden. Hierzu werden folgende Objekte nacheinander versandt:

| | | | |
|---|---|---|---|
| 7E5h → | 46h | VendorId | reserved |

| | | | |
|---|---|---|---|
| 7E5h → | 47h | ProductCode | reserved |

| | | | |
|---|---|---|---|
| 7E5h → | 48h | RevisionNumber LOW | reserved |
| 7E5h → | 49h | RevisionNumber HIGH | reserved |

| | | | |
|---|---|---|---|
| 7E5h → | 4Ah | SerialNumber LOW | reserved |
| 7E5h → | 4Bh | SerialNumber HIGH | reserved |

Jeder Drehgeber mit den entsprechenden Parametern meldet sich mit folgender Meldung:

| | | |
|---|---|---|
| 7E4h ← | 4Fh | reserved |

## 3.5. CANopen Lift Drehgeber nach Appl. Profil DSP417

### 3.5.1. Objekte Übersicht

Nach CiA (CAN in Automation) werden die Objekte in drei Gruppen unterteilt:

- **Standard-Objekte**:
  1000h, 1001h, 1018h
- **Herstellerspezifische Objekte** :
  2000h - 5FFFh
- **Applikationsspezifische Objekte**:
  6000h - 67FFh

Folgende Tabelle zeigt eine Zusammenfassung aller vom Gerät unterstützen SDO Objekte.

**Objekt** Objekt Nummer in Hex
**Name** Beschreibung des Objektes
**Type** U/I = Unsigned/Integer , Zahl=Anzahl Bit, ARR=Array
**Attr** ro=ReadOnly, wo=WriteOnly, rw=ReadWrite
**Default** DefaultWert beim ersten Init
**EE** 1= Wird im EEPROM gespeichert
**Info** zusätzliche Info

| Objekt | Name | Type | Attr | Default | EE | Info |
|---|---|---|---|---|---|---|
| 1000h | Device Type | U32 | ro | 060001A1h | | $2^0..2^{15}$ ProfilNr= 1A1h=417 encoder lift<br>$2^{16}..2^{23}$ Communication model<br>$2^{24}..2^{31}$ virtual device code |
| 1001h | Error Register | U8 | ro | 00h | | Bit0=Generic Error |
| 1003h | PreDefined ErrorField | ARR | | | | Enthält die letzten 8 Fehler oder Warnungen |
| 00h | Größter Subindex | U8 | rw | 00h | | Anzahl gespeicherten Meldungen (0..8) |
| 01h | Letzter Eintrag | U32 | ro | | | Fehler oder Warnung |
| .. | .. | .. | .. | .. | | .. |
| 08h | Ältester Eintrag | U32 | ro | | | Fehler oder Warnung |
| 1005h | Sync COB-ID | U32 | rw | 80h | 1 | COB-ID des Sync Objektes |
| 1008h | DeviceName | U32 | ro | "GXP5" | 1 | "GXP5" |
| 1009h | Hardware Version | U32 | ro | "1.00“ | | Hardware Version in ASCII V 1.00 |
| 100Ah | Software Version | U32 | ro | "1.00“ | | Software Version in ASCII V 1.00 |
| 1010h | Store Parameters | ARR | | | | |
| 00h | Größter Subindex | U8 | ro | 04h | | Anzahl Speichermöglichkeiten = 4 |
| 01h | Alle Parameter speichern | U32 | rw | | | =“save“ (0x73617665) zum speichern |
| 02h | Communication Parameters | U32 | rw | | | =“save“ (0x73617665) zum speichern |
| 03h | Application Parameters | U32 | rw | | | =“save“ (0x73617665) zum speichern |
| 04h | Manufacturer Specific Parameters | U32 | rw | | | =“save“ (0x73617665) zum speichern |
| 1011h | Restore Default Parameters | ARR | | | | |
| 00h | Größter Subindex | U8 | ro | 04h | | Anzahl Zurücksetz Möglichkeiten = 4 |
| 01h | Alle Parameter | U32 | rw | | | =“load“ (0x6C6F6164) zum laden |
| 02h | Communication Parameters | U32 | rw | | | =“load“ (0x6C6F6164) zum laden |
| 03h | Application Parameters | U32 | rw | | | =“load“ (0x6C6F6164) zum laden |
| 04h | Manufacturer Specific Parameters | U32 | rw | | | =“load“ (0x6C6F6164) zum laden |
| 1014h | Emergency COB-ID | U32 | rw | 80h + NodeID | 1 | COB-ID des Emergency Objektes |
| 1017h | Producer Heartbeat Time | U16 | rw | 00h | 1 | |
| 1018h | Identity Object | U32 | ro | | | |
| 00h | Größter Subindex | U8 | ro | 04h | | |
| 01h | VendorID | U32 | ro | ECh | 1 | Von CiA vergebene Vendornummer |
| 02h | Product Code | U32 | ro | 1Eh | 1 | |
| 03h | Revision Number | U32 | ro | 00010000h | | Aktuelle Revisionsnummer V1.00 |
| 04h | Serial Number | U32 | ro | xyz | 1 | Eindeutige fortlaufende Seriennummer |
| 1906h | Transmit PDO1 Parameter | REC | | | | |
| 00h | Größter Subindex | U8 | ro | 05h | | |
| 01h | COB-ID | U32 | rw | 18Ch | 1 | PDO ID = 17Bh + LiftNr*10 + PosUnit |
| 02h | PDO Type | U8 | rw | FFh | 1 | FFh= zyklisch |
| 05h | EventTimer | U16 | rw | 00h | 1 | Zykluszeit in ms |
| 1B06h | Transmit PDO1 Mapping | ARR | | | | Nur statisches Mapping !! |
| 00h | Größter Subindex | U8 | ro | 02h | | |
| 01h | Inhalt des PDO1 | U32 | ro | 63830120h | | Positionswert 32Bit |
| 02h | Inhalt des PDO1 | S16 | ro | 63900110h | | Geschwindigkeit 16 Bit |

| | | | | | | |
|---|---|---|---|---|---|---|
| 2100h | Baudrate | U8 | rw | 02h | 1 | Nach setzen der Baudrate muss EEPROM gespeichert und neu initialisiert werden<br>0=10 kBit/s<br>1=20 kBit/s<br>2=50 kBit/s<br>3=100 kBit/s<br>4=125 kBit/s<br>5=250 kBit/s<br>6=500 kBit/s<br>7=800 kBit/s<br>8=1000 kBit/s |
| 2101h | Node Id | U8 | rw | 01h | 1 | Node Nummer 1..127 möglich<br>Nach setzen der Baudrate muss EEPROM gespeichert und neu initialisiert werden. |
| 2110h | Kundenspez. Node ID Version | U32 | ro | 01h | 1 | Behandlung der NodeID definieren |
| 2201h | Statistik | REC | | | | |
| 00h | Größter Subindex | U8 | ro | 03h | | Anzahl Subindexe |
| 01h | Anzahl Position-Fehler | U32 | ro | 00h | 1 | Positionskontrolle |
| 02h | Zeit in Sekunden | U32 | ro | 00h | 1 | Zeit seit letztem Reset |
| 03h | Anzahl TimerReset Watchdog | U32 | ro | 00h | 1 | TimerWatchDog |
| 2300h | Customer EEPROM Bereich | ARR | | | | In diesem Objekt können beliebige Daten |
| 00h | Größter Subindex | U8 | ro | 08h | | Abgespeichert werden |
| 01h | Data0 | U16 | rw | 00h | 1 | |
| 02h | Data1 | U16 | rw | 00h | 1 | |
| 03h | Data2 | U16 | rw | 00h | 1 | |
| 04h | Data3 | U16 | rw | 00h | 1 | |
| 05h | Data4 | U16 | rw | 00h | 1 | |
| 06h | Data5 | U16 | rw | 00h | 1 | |
| 07h | Data6 | U16 | rw | 00h | 1 | |
| 08h | Data7 | U16 | rw | 00h | 1 | |
| | | | | | | |
| 2800h | PDO1-Zusatz /Event Trigger | U8 | rw | 00h | 1 | Wiederholungszähler für PDO1 |
| 6001h | Lift number | U8 | rw | 01h | 1 | Lift Nummer 1..8 (momentan nur 1 möglich) |
| 6380h | Operating Parameter | ARR | | | | |
| 00h | Größter Subindex | U8 | ro | 04h | | |
| 01h | Position unit 1 | U16 | rw | 04h | 1 | Bit0=Drehrichtung 0=CW, 1=CCW<br>Bit2=Skalierungsfunktion 0=aus, 1=ein<br>CW= steigende Positionswerte bei Drehung im Uhrzeigersinn (Sicht auf Flansch<br>CCW steigende Positionswerte bei Drehung im Gegenuhrzeigersinn (Sicht auf Flansch) |
| 02h | Position unit 2 | U16 | rw | 04h | 1 | |
| 03h | Position unit 3 | U16 | rw | 04h | 1 | |
| 04h | Position unit 4 | U16 | rw | 04h | 1 | |
| 6381h | Auflösung | ARR | | | | |
| 00h | Größter Subindex | U8 | ro | 04h | | |
| 01h | Position unit 1 | U32 | rw | 8192 | 1 | Auflösung in Schritte/Umdrehung<br>bzw.<br>mm/Umdrehung |
| 02h | Position unit 2 | U32 | rw | 2000h | 1 | |
| 03h | Position unit 3 | U32 | rw | 13Bit | 1 | |
| 04h | Position unit 4 | U32 | rw | | 1 | |
| 6382h | Preset Wert in Schritten | ARR | | | | |
| 00h | Größter Subindex | U8 | ro | 04h | | |
| 01h | Position unit 1 | U32 | rw | 00h | 1 | Preset in Schritten → Offset |
| 02h | Position unit 2 | U32 | rw | 00h | 1 | |
| 03h | Position unit 3 | U32 | rw | 00h | 1 | |
| 04h | Position unit 4 | U32 | rw | 00h | 1 | |
| 6383h | Position in Schritten | ARR | ro | | | |
| 00h | Größter Subindex | U8 | ro | 04h | | |
| 01h | Position unit 1 | U32 | ro | | | Positionswert inkl. Offset in Schritten |
| 02h | Position unit 2 | U32 | ro | | | |
| 03h | Position unit 3 | U32 | ro | | | |
| 04h | Position unit 4 | U32 | ro | | | |
| 6384h | Encoder measuring step setting | ARR | ro | | | Position unit 1 |
| 00h | Größter Subindex | U8 | ro | 03h | | |
| 01h | Measuring step | U16 | rw | 64h | 1 | Einheit Messschritt , Vielfaches von 10µm |
| 02h | Speed Measuring step | U16 | rw | 0Ah | 1 | Einheit Geschwindigkeit, Vielfaches von 0,1mm/s |
| 03h | Acceleration measuring step | U16 | rw | 0Ah | 1 | Einheit Beschleunigung, Vielfaches von 1mm/s² |

| | | | | | | |
|---|---|---|---|---|---|---|
| 6390h | Speed value car | ARR | | | | |
| 00h | Größter Subindex | U8 | ro | 04h | | |
| 01h | Position unit 1 | S16 | ro | | | Geschwindigkeit (Einheit siehe Objekt 6384 ) |
| 02h | Position unit 2 | S16 | ro | | | |
| 03h | Position unit 3 | S16 | ro | | | |
| 04h | Position unit 4 | S16 | ro | | | |
| 63C0h | Operating Status | ARR | | | | |
| 00h | Größter Subindex | U8 | ro | 04h | | |
| 01h | Position unit 1 | U16 | ro | 04h | | Bit0=Drehrichtung 0=CW, 1=CCW<br>Bit2=Skalierungsfunktion 0=aus, 1=ein<br>CW: steigende Positionswerte bei Drehung im Uhrzeigersinn (Sicht auf Flansch)<br>CCW :steigende Positionswerte bei Drehung im Gegenuhrzeigersinn (Sicht auf Flansch) |
| 02h | Position unit 2 | U16 | ro | 04h | | |
| 03h | Position unit 3 | U16 | ro | 04h | | |
| 04h | Position unit 4 | U16 | ro | 04h | | |
| 63C1h | Max. Auflösung | ARR | | | | |
| 00h | Größter Subindex | U8 | ro | 04h | | |
| 01h | Position unit 1 | U32 | ro | 2000h=GXP5 | | Maximale Auflösung in Schritte/Umdrehung |
| 02h | Position unit 2 | U32 | ro | 2000h=GXP5 | | |
| 03h | Position unit 3 | U32 | ro | 2000h=GXP5 | | |
| 04h | Position unit 4 | U32 | ro | 2000h=GXP5 | | |
| 63C2h | Max. Umdrehungen | ARR | | | | |
| 00h | Größter Subindex | U8 | ro | 04h | | |
| 01h | Position unit 1 | U16 | ro | 10000h | | Max. Anzahl Umdrehungen |
| 02h | Position unit 2 | U16 | ro | 10000h | | |
| 03h | Position unit 3 | U16 | ro | 10000h | | |
| 04h | Position unit 4 | U16 | ro | 10000h | | |
| 63C4h | Unterstützte Warnungen | ARR | | | | |
| 00h | Größter Subindex | U8 | ro | 04h | | |
| 01h | Position unit 1 | U16 | ro | 14h | | Folgende Warnungen werden unterstützt:<br>Bit2 = CPU watchdog status<br>Bit4 = Batterieladung |
| 02h | Position unit 2 | U16 | ro | 14h | | |
| 03h | Position unit 3 | U16 | ro | 14h | | |
| 04h | Position unit 4 | U16 | ro | 14h | | |
| 63C5h | Warnungen | ARR | | | | |
| 00h | Größter Subindex | U8 | ro | 04h | | |
| 01h | Position unit 1 | U16 | ro | 00h | | Folgende Warnungen werden ausgewertet:<br>Bit2 = CPU watchdog status<br>Bit4 = Batterieladung |
| 02h | Position unit 2 | U16 | ro | 00h | | |
| 03h | Position unit 3 | U16 | ro | 00h | | |
| 04h | Position unit 4 | U16 | ro | 00h | | |
| 63C6h | Unterstützte Alarme | ARR | | | | |
| 00h | Größter Subindex | U8 | ro | 04h | | |
| 01h | Position unit 1 | U16 | ro | 01h | | Folgende Alarme werden unterstützt:<br>Bit0=PositionsError |
| 02h | Position unit 2 | U16 | ro | 01h | | |
| 03h | Position unit 3 | U16 | ro | 01h | | |
| 04h | Position unit 4 | U16 | ro | 01h | | |
| 63C7h | Alarme | ARR | | | | |
| 00h | Größter Subindex | U8 | ro | 04h | | |
| 01h | Position unit 1 | U16 | ro | 00h | | Folgende Alarme werden ausgewertet:<br>Bit0=PositionsError |
| 02h | Position unit 2 | U16 | ro | 00h | | |
| 03h | Position unit 3 | U16 | ro | 00h | | |
| 04h | Position unit 4 | U16 | ro | 00h | | |
| 63C8h | Operating Time | ARR | | | | |
| 00h | Größter Subindex | U8 | ro | 4h | | |
| 01h | Position unit 1 | U32 | ro | 0h | | Zeit in 1/10 Stunden seit letztem Reset |
| 02h | Position unit 2 | U32 | ro | 0h | | |
| 03h | Position unit 3 | U32 | ro | 0h | | |
| 04h | Position unit 4 | U32 | ro | 0h | | |
| 63C9h | Offset Wert | ARR | | | | |
| 00h | Größter Subindex | U8 | ro | 4h | | |
| 01h | Position unit 1 | U32 | ro | 0h | | Offset aus Preset berechnet → 6382h |
| 02h | Position unit 2 | U32 | ro | 0h | | |
| 03h | Position unit 3 | U32 | ro | 0h | | |
| 04h | Position unit 4 | U32 | ro | 0h | | |

### 3.5.2. Ausführliche Objektbeschreibung

**Objekt 1000 Device Type**

| | |
|---|---|
| SubIndex | 0 |
| DatenTyp | Unsigned 32 |
| Zugriff | ReadOnly |
| Default | 060001A1h |
| EEPROM | No |
| Beschreibung | Information über Geräteprofil und Gerätetyp |
| Werte | $2^0..2^{15}$ ProfilNr= 1A1h=417 (Application Profile for Lift Control Systems).<br>$2^{16}..2^{23}$ Communication model<br>$2^{24}..2^{31}$ virtual device code |

**Objekt 1001 Error Register**

| | |
|---|---|
| SubIndex | 0 |
| DatenTyp | Unsigned 8 |
| Zugriff | ReadOnly |
| Default | 0h |
| EEPROM | No |
| Beschreibung | Aktueller FehlerCode |
| Werte | Bit 0 1 = Generic Error |

**Objekt 1003 Vor Definiertes Error Feld**

CiA (CAN in Automation) definiert hier etwa 200 verschiedene Error Codes.
In diesem Dokument werden nur die für den Sensor relevanten Error Codes beschrieben.
Dieses Objekt speichert die letzten 8 aufgetretenen Fehler oder Warnungen.

| | |
|---|---|
| SubIndex | 0 |
| DatenTyp | Unsigned 8 |
| Zugriff | ReadWrite |
| Default | 0 |
| EEPROM | No |
| Beschreibung | Lesen: Anzahl Fehler oder Warnungen<br>Schreiben von 0: Fehler zurücksetzen |
| Werte | 0..8 |

| | |
|---|---|
| SubIndex | 1..8 |
| DatenTyp | Unsigned 32 |
| Zugriff | ReadOnly |
| Default | 0 |
| EEPROM | No |
| Beschreibung | Aufgetretene Fehler oder Warnungen wobei SubIndex 1 der letzte, SubIndex 2 der vorletzte, ...., Eintrag ist |
| Werte | Noch nicht definiert |

**Objekt 1005 COB-ID SYNC Message**

| | |
|---|---|
| SubIndex | 0 |
| DatenTyp | Unsigned 32 |
| Zugriff | ReadWrite |
| Default | 80h |
| EEPROM | Yes |
| Beschreibung | Definiert COB-ID des Synchronisations Objekt (SYNC) |
| Werte | Bit 31 not defined<br>Bit 30 1=Sensor generiert SYNC Meldungen, 0=generiert keine SYNC Meldung<br>Bit 29 1=29 Bit SYNC COB-ID (CAN 2.0B), 0=28 Bit SYNC COB-ID (CAN 2.0A)<br>Bit 28..11 Bit 28..11 der 29 Bit SYNC COB-ID<br>Bit 10..0 Bit 10..0 der SYNC COB-ID |

**Objekt 1008 Hersteller Device Name**

| | |
|---|---|
| SubIndex | 0 |
| DatenTyp | Unsigned 32 |
| Zugriff | ReadOnly |
| Default | "GXP5" |
| EEPROM | No |
| Beschreibung | GeräteBezeichnung in ASCII |
| Werte | Data 0..3:<br>"GXP5" = 47h 58h 50h 35h → GXP5W Multiturn |

**Objekt 1009 Hersteller Hardware Version**

| | |
|---|---|
| SubIndex | 0 |
| DatenTyp | Unsigned 32 |
| Zugriff | ReadOnly |
| Default | |
| EEPROM | No |
| Beschreibung | Hardware-Version in ASCII |
| Werte | Data 0..3 Beispiel: 31h 2Eh 30h 30h = "1.00“ |

**Objekt 100A Hersteller Software Version**

| | |
|---|---|
| SubIndex | 0 |
| DatenTyp | Unsigned 32 |
| Zugriff | ReadOnly |
| Default | |
| EEPROM | No |
| Beschreibung | Software-Version in ASCII |
| Werte | Data 0..3 siehe Typenschild |

### Objekt 1010 Parameter speichern

Über das Objekt 1010h wird das Speichern untenstehender Objekte in den nichtflüchtigen Speicher (EEPROM) ausgelöst.
Um ein unabsichtliches Speichern zu verhindern muss die Botschaft „save“ in den SubIndex 1 geschrieben werden.

| COB-ID | DLC | Kommando | Objekt L | Objekt H | Subindex | Data 0 | Data 1 | Data 2 | Data 3 |
|---|---|---|---|---|---|---|---|---|---|
| 600h+Node-ID | 8 | 23h | 10h | 10h | 01 | 73h 's' | 61h 'a' | 76h 'v' | 65h 'e' |

Im EEPROM gespeicherte Objekte:

| Objekt | SubIndex | Beschreibung | Default Wert (nach Objekt 1011) |
|---|---|---|---|
| 1005h | 0h | Sync ID | 80h |
| 1008h | 0h | Device Name | "GXP5" |
| 1014h | 0h | Emergency COB-ID | 80h+Node-ID |
| 1017h | 0h | Producer Heartbeat time | 0h (disabled) |
| 1018h | 1h | VendorID | ECh |
| 1018h | 2h | Product Code | 28h → GXP5W Multiturn |
| 1018h | 4h | Serial Number | Xyz |
| 1906h | 1h | PDO1 ID | PDO ID = 17Bh + LiftNr * 10 + PosUnit |
| 1906h | 2h | PDO1 Type | FFh -> asynchron, zyklisch |
| 1906h | 5h | PDO1 EventTimer asynchron Betrieb | 0h (disabled) |
| | | | |
| 2100h | 0h | Baudrate | 2h = 50 kBaud |
| 2101h | 0h | Node-ID | 1h |
| 2110h | 0h | Manufacturer_Options | 8h |
| 2201h | 1h | Anzahl Position-Fehler | 0h |
| 2201h | 2h | Betriebszeit Total in Sekunden | 0h |
| 2201h | 3h | Anzahl TimerReset vom WatchDog | 0h |
| 2300h | 1h | Kundenspezifischer EEProm Bereich Data0 | 0h |
| 2300h | 2h | Kundenspezifischer EEProm Bereich Data1 | 0h |
| 2300h | 3h | Kundenspezifischer EEProm Bereich Data2 | 0h |
| 2300h | 4h | Kundenspezifischer EEProm Bereich Data3 | 0h |
| 2300h | 5h | Kundenspezifischer EEProm Bereich Data4 | 0h |
| 2300h | 6h | Kundenspezifischer EEProm Bereich Data5 | 0h |
| 2300h | 7h | Kundenspezifischer EEProm Bereich Data6 | 0h |
| 2300h | 8h | Kundenspezifischer EEProm Bereich Data7 | 0h |
| 2800h | 0h | PDO1-Zusatz (EventTrigger) | 0h |
| | | | |
| 6001h | 0h | Lift number | 1h |
| 6380h | 1h | Operating Parameter | 0004h |
| 6381 | 1h | Auflösung in Schritte/Umdrehung | 2000h = GXP5 |
| 6382h | 1h | Preset Wert in Schritten | 0h |
| 6384h | 01h | Encoder measuring step setting<br>Einheit f. Messschritt: Vielfaches von 10 µm | 64h |
| 6384h | 02h | Encoder measuring step setting<br>Einheit f. Geschwindigkeit: Vielfaches von 0,1 mm/s | 0Ah |
| 6384h | 03h | Encoder measuring step setting<br>Einheit f. Beschleunigung: Vielfaches von 1 $mm/s^2$ | 0Ah |
| | | | |

**Objekt 1011 Restore Parameters**

Über das Objekt 1011h werden die Werte im RAM mit den Default Werten (siehe Objekt 1010h) überschrieben. Zudem wird der Inhalt des EEPROM als ungültig markiert. Dies bedeutet, dass bis zum nächsten Speichern der Daten im EEPROM, jeweils die Default Werte geladen werden.
Um ein unabsichtliches Überschreiben zu verhindern muss die Botschaft „load“ in den SubIndex 1 geschrieben werden.

| COB-ID | DLC | Kommando | Objekt L | Objekt H | Subindex | Data 0 | Data 1 | Data 2 | Data 3 |
|---|---|---|---|---|---|---|---|---|---|
| 600h+Node-ID | 8 | 23h | 11h | 10h | 01 | 6Ch 'l' | 6Fh 'o' | 61h 'a' | 64h 'd' |

**Objekt 1014 COB-ID Emergency Message**

| | |
|---|---|
| SubIndex | 0 |
| DatenTyp | Unsigned 32 |
| Zugriff | ReadWrite |
| Default | 80h+Node-ID |
| EEPROM | Yes |
| Beschreibung | Definiert COB-ID des Emergency Objekt |
| Werte | 80h + Node-ID |

**Objekt 1017 Producer Heartbeat Time**

| | |
|---|---|
| SubIndex | 0 |
| DatenTyp | Unsigned 16 |
| Zugriff | ReadWrite |
| Default | 0h |
| EEPROM | Yes |
| Beschreibung | Definiert die Wiederholzeit des Überwachungsdienstes Heartbeat |
| Werte | 0 = Disabled<br>1..65535 = Wiederholzeit in ms |

**Objekt 1018 Identity Object**

| | |
|---|---|
| SubIndex | 0 |
| DatenTyp | Unsigned 8 |
| Zugriff | ReadOnly |
| Default | 4 |
| EEPROM | No |
| Beschreibung | Größter unterstützter Subindex |
| Werte | 4 = Größter unterstützter SubIndex |

| | |
|---|---|
| SubIndex | 1 |
| DatenTyp | Unsigned 32 |
| Zugriff | ReadOnly |
| Default | ECh |
| EEPROM | Yes |
| Beschreibung | Von CiA vergebene VendorID für Baumer IVO GmbH & Co. KG |
| Werte | ECh (im Internet unter www.can-cia.de) |
| SubIndex | 2 |
| DatenTyp | Unsigned 32 |
| Zugriff | ReadOnly |
| Default | 28h = GXP5 |
| EEPROM | Yes |
| Beschreibung | Product Code |
| Werte | 28h → GXP5W Multiturn |

| | |
|---|---|
| SubIndex | 3 |
| DatenTyp | Unsigned 32 |
| Zugriff | ReadOnly |
| Default | |
| EEPROM | No |
| Beschreibung | Revision Number des Sensors |
| Werte | <table><tr><td>Data 0 = Lauf-nummer LOW</td><td>Data 1 = Lauf-nummer HIGH</td><td>Data 2 = Version LOW</td><td>Data 3 = Version HIGH</td></tr></table> Version der aktuellen Software = xxyy (xx=Version, yy=Laufnummer)<br><br>siehe Typenschild |

| | |
|---|---|
| SubIndex | 4 |
| DatenTyp | Unsigned 32 |
| Zugriff | ReadOnly |
| Default | 0 |
| EEPROM | Yes |
| Beschreibung | Fortlaufende eindeutige Seriennummer des Sensors |
| Werte | Wird im Werk während des Endtests definiert |

**Objekt 1906 PDO1 Parameter**

| | |
|---|---|
| SubIndex | 0 |
| DatenTyp | Unsigned 32 |
| Zugriff | ReadOnly |
| Default | 5 |
| EEPROM | No |
| Beschreibung | Größter unterstützter Subindex |
| Werte | 5 |

| | |
|---|---|
| SubIndex | 1 |
| DatenTyp | Unsigned 32 |
| Zugriff | ReadWrite |
| Default | 18Ch |
| EEPROM | Yes |
| Beschreibung | COB-ID des PDO |
| Werte | 17Bh + LiftNr * 10 + PosUnit |
| SubIndex | 2 |
| DatenTyp | Unsigned 8 |
| Zugriff | ReadWrite |
| Default | FFh |
| EEPROM | Yes |
| Beschreibung | PDO Type |
| Werte | 1..n..F0h = PDO hat synchrone Charakteristik (auf jedes n-te SYNC-Telegramm wird das PDO gesendet)<br>FFh = PDO hat asynchrone Charakteristik (PDO's werden zyklisch in Abhängigkeit vom EventTimer und EventTrigger gesendet) |

| | |
|---|---|
| SubIndex | 5 |
| DatenTyp | Unsigned 16 |
| Zugriff | ReadWrite |
| Default | 203h |
| EEPROM | Yes |
| Beschreibung | Event Timer für Prozess Daten Objekt |
| Werte | 0 = zyklisches Senden ausgeschaltet<br>1..n..65535 =Wiederholzeit zyklisches Senden beträgt n ms. |

**Objekt 1B06 PDO1 Mapping**

| | |
|---|---|
| SubIndex | 0 |
| DatenTyp | Unsigned 8 |
| Zugriff | ReadOnly |
| Default | 0 |
| EEPROM | No |
| Beschreibung | Größter unterstützter Subindex |
| Werte | 1 |

| | |
|---|---|
| SubIndex | 1 |
| DatenTyp | Unsigned 32 |
| Zugriff | ReadOnly |
| Default | 63830120h |
| EEPROM | No |
| Beschreibung | Beschreibt den Inhalt der PDO1-Meldung |
| Werte | 6383h = Position (Subindex 01, Länge 0x20) |

| | |
|---|---|
| SubIndex | 2 |
| DatenTyp | Signed 16 |
| Zugriff | ReadOnly |
| Default | 63900110h |
| EEPROM | No |
| Beschreibung | Beschreibt den Inhalt der PDO1-Meldung |
| Werte | 6390h = Geschwindigkeit (Subindex 01, Länge 0x10) |

**Objekt 2100 Baudrate**

| | |
|---|---|
| SubIndex | 0 |
| DatenTyp | Unsigned 8 |
| Zugriff | ReadWrite |
| Default | 2 = 50kBaud |
| EEPROM | Yes |
| Beschreibung | Baudrate des Sensors lesen oder neu setzen.<br>→ Nach dem Setzen müssen Parameter mit dem Objekt 1010h im EEPROM gespeichert und danach der Sensor neu initialisiert werden |
| Werte | 0 10 kBaud<br>1 20 kBaud<br>2 50 kBaud<br>3 100 kBaud<br>4 125 kBaud<br>5 250 kBaud<br>6 500 kBaud<br>7 800 kBaud<br>8 1000 kBaud |

**Objekt 2101 Node-ID**

| | |
|---|---|
| SubIndex | 0 |
| DatenTyp | Unsigned 8 |
| Zugriff | ReadWrite |
| Default | 1 |
| EEPROM | Yes |
| Beschreibung | Node-ID des Sensors lesen oder neu setzen.<br>→ Nach dem Setzen müssen Parameter mit dem Objekt 1010h im EEPROM gespeichert und danach der Sensor neu initialisiert werden |
| Werte | 1..127 |

**Objekt 2110 Manufacturer_Options**

| | |
|---|---|
| SubIndex | 0 |
| DatenTyp | Unsigned 32 |
| Zugriff | ReadWrite |
| Default | 8h |
| EEPROM | Yes |
| Beschreibung | Einstellungen, um die Kompatibilität zu älteren Sensoren zu gewährleisten |
| Werte | Bit 3 =1 : Reset nach Bus Off |

**Objekt 2201 Statistik**

| | |
|---|---|
| SubIndex | 0 |
| DatenTyp | Unsigned 8 |
| Zugriff | ReadOnly |
| Default | 3h |
| EEPROM | No |
| Beschreibung | Größter unterstützter Subindex |
| Werte | 3 |

| | |
|---|---|
| SubIndex | 1 |
| DatenTyp | Unsigned 32 |
| Zugriff | ReadOnly |
| Default | 0h |
| EEPROM | Yes |
| Beschreibung | Anzahl der Positions-Fehler insgesamt |
| Werte | 0...4294967295 |

| | |
|---|---|
| SubIndex | 2 |
| DatenTyp | Unsigned 32 |
| Zugriff | ReadOnly |
| Default | 0h |
| EEPROM | Yes |
| Beschreibung | Betriebszeit in Sekunden Total (Objekt 6508h Zeit seit letztem Reset) |
| Werte | 0... 4294967295 |

| | |
|---|---|
| SubIndex | 3 |
| DatenTyp | Unsigned 32 |
| Zugriff | ReadOnly |
| Default | 0h |
| EEPROM | Yes |
| Beschreibung | WatchDog TimerReset Zähler |
| Werte | 0... 4294967295 |

**Objekt 2300 Customer EEPROM Bereich**

| | |
|---|---|
| SubIndex | 0 |
| DatenTyp | Unsigned 8 |
| Zugriff | ReadOnly |
| Default | 8h |
| EEPROM | No |
| Beschreibung | In diesem Objekt können beliebige Daten abgespeichert werden |
| Werte | 8 |

| | |
|---|---|
| SubIndex | 1...8 |
| DatenTyp | Unsigned 16 |
| Zugriff | ReadWrite |
| Default | 0h |
| EEPROM | Yes |
| Beschreibung | Pro Subindex kann ein 16 Bit Wert abgespeichert werden<br>(speichern im EEPROM über Objekt 1010h) |
| Werte | 0 |

**Objekt 2800 PDO1-Zusatz (EventTrigger)**

| | |
|---|---|
| SubIndex | 0 |
| DatenTyp | Unsigned 8 |
| Zugriff | ReadWrite |
| Default | 0h |
| EEPROM | Yes |
| Beschreibung | Der Wert EventTrigger bestimmt, wie oft der gleiche PDO-Wert übertragen wird |
| Werte | 0 = PDO-Zähler ist ausgeschaltet → ständiges übermitteln (Zeitbasis vom EventTimer)<br>1..n..255 = der gleiche PDO-Wert wird n-mal gesendet (Zeitbasis vom EventTimer) |

**Objekt 6001 Lift number**

| | |
|---|---|
| SubIndex | 0 |
| DatenTyp | Unsigned 8 |
| Zugriff | ReadWrite |
| Default | 1 |
| EEPROM | Yes |
| Beschreibung | Lift Nummer |
| Werte | 1 |

**Objekt 6380 Operating Parameter**

| | |
|---|---|
| SubIndex | 0 |
| DatenTyp | Unsigned 16 |
| Zugriff | ReadWrite |
| Default | 4 |
| EEPROM | Yes |
| Beschreibung | Betriebsparameter |
| Werte | Bit 0 Drehrichtung = 0<br>→ Uhrzeigersinn; 1 → Gegenuhrzeigersinn<br>Bit 2 Skalierungsfunktion = 0<br>→ max. Auflösung; 1 → gespeicherte Auflösung |

**Objekt 6381 Auflösung**

| | |
|---|---|
| SubIndex | 1 |
| DatenTyp | Unsigned 32 |
| Zugriff | ReadWrite |
| Default | 2000h = 8192 = 13Bit → GXP5 |
| EEPROM | Yes |
| Beschreibung | Anzahl Schritte pro Umdrehung frei wählbar.<br>! Offsetwert wird bei einer Änderung der Auflösung zurückgesetzt ! |
| Werte | 1.. 8192 → GXP5 |

**Objekt 6382 Preset Wert**

| | |
|---|---|
| SubIndex | 1 |
| DatenTyp | Unsigned 32 |
| Zugriff | ReadWrite |
| Default | 0h |
| EEPROM | Yes |
| Beschreibung | Frei wählbarer Positionswert. Preset und interne Position ergeben Offset (→ Objekt 63C9h) |
| Werte | 0.. 536870911 → GXP5 |

**Objekt 6383 Position in Schritten**

| | |
|---|---|
| SubIndex | 1 |
| DatenTyp | Unsigned 32 |
| Zugriff | ReadOnly |
| Default | |
| EEPROM | No |
| Beschreibung | Aktuelle Position inkl. Offset |
| Werte | 0.. 536870911 → GXP5 |

**Objekt 6384 Encoder measuring Step setting**

| | |
|---|---|
| SubIndex | 0 |
| DatenTyp | Unsigned 8 |
| Zugriff | ReadOnly |
| Default | 3h |
| EEPROM | No |
| Beschreibung | Größter unterstützter Subindex |
| Werte | 3 |

| | |
|---|---|
| SubIndex | 1 |
| DatenTyp | Unsigned 16 |
| Zugriff | ReadOnly |
| Default | 64h |
| EEPROM | Yes |
| Beschreibung | Encoder measuring step setting |
| Werte | Einheit für Messschritt: Vielfaches von 10 µm |

| | |
|---|---|
| SubIndex | 2 |
| DatenTyp | Unsigned 16 |
| Zugriff | ReadOnly |
| Default | 0Ah |
| EEPROM | Yes |
| Beschreibung | Speed measuring Step |
| Werte | Einheit für Geschwindigkeit: Vielfaches von 0,1 mm/s |

| | |
|---|---|
| SubIndex | 3 |
| DatenTyp | Unsigned 16 |
| Zugriff | ReadOnly |
| Default | 0Ah |
| EEPROM | Yes |
| Beschreibung | Acceleration measuring Step |
| Werte | Einheit f. Beschleunigung: Vielfaches von 1 $mm/s^2$ |

**Objekt 6390 Speed value car**

| | |
|---|---|
| SubIndex | 1 |
| DatenTyp | Signed 16 |
| Zugriff | ReadOnly |
| Default | |
| EEPROM | no |
| Beschreibung | Geschwindigkeit |
| Werte | |

**Objekt 63C0 Operating Status**

| | |
|---|---|
| SubIndex | 1 |
| DatenTyp | Unsigned 16 |
| Zugriff | ReadOnly |
| Default | 4h |
| EEPROM | No |
| Beschreibung | Betriebsdaten, die in Objekt 6380h geschrieben wurden |
| Werte | Bit 0 Drehrichtung = 0<br>→ Uhrzeigersinn; 1 → Gegenuhrzeigersinn<br>Bit 2 Skalierungsfunktion = 0<br>→ max. Auflösung; 1 → gespeicherte Auflösung |

**Objekt 63C1 Max. Auflösung in Schritten**

| | |
|---|---|
| SubIndex | 1 |
| DatenTyp | Unsigned 32 |
| Zugriff | ReadOnly |
| Default | 2000h = 8192 = 13Bit → GXP5 |
| EEPROM | No |
| Beschreibung | Maximale Singleturn-Auflösung in Schritten |
| Werte | 2000h = 8192 = 13Bit → GXP5 |

**Objekt 63C2 Max. Umdrehungen**

| | |
|---|---|
| SubIndex | 1 |
| DatenTyp | Unsigned 16 |
| Zugriff | ReadOnly |
| Default | 10000h = 65536 = 16Bit |
| EEPROM | No |
| Beschreibung | Maximale Anzahl Umdrehungen |
| Werte | 10000h = 65536= 16Bit |

**Objekt 63C4 Unterstützte Warnungen**

| | |
|---|---|
| SubIndex | 1 |
| DatenTyp | Unsigned 16 |
| Zugriff | ReadOnly |
| Default | Multiturn:<br>14h |
| EEPROM | No |
| Beschreibung | Vom Objekt 63C5 unterstützte Warnungen |
| Werte | Multiturn:<br>Bit 2 = CPU Watchdog Status<br>Bit 4 = Batterie-Ladung |

**Objekt 63C5 Warnungen**

| | |
|---|---|
| SubIndex | 1 |
| DatenTyp | Unsigned 16 |
| Zugriff | ReadOnly |
| Default | 0h |
| EEPROM | No |
| Beschreibung | Warnungen gem. Objekt 63C4 |
| Werte | Multiturn:<br>Bit 2 = 1 → CPU Watchdog geresetet<br>Bit 4 = 1 → Batterie-Ladung zu tief |

**Objekt 63C6 Unterstützte Alarme**

| | |
|---|---|
| SubIndex | 1 |
| DatenTyp | Unsigned 16 |
| Zugriff | ReadOnly |
| Default | 1h |
| EEPROM | No |
| Beschreibung | Vom Objekt 63C7 unterstützte Alarmmeldungen |
| Werte | Bit 0 = Positions-Fehler |

**Objekt 63C7 Alarme**

| | |
|---|---|
| SubIndex | 1 |
| DatenTyp | Unsigned 16 |
| Zugriff | ReadOnly |
| Default | 0h |
| EEPROM | No |
| Beschreibung | Alarmmeldungen gem. Objekt 63C6h |
| Werte | Bit 0 = 1 → Positions-Fehler aktiv |

**Objekt 63C8 Operating Time**

| | |
|---|---|
| SubIndex | 1 |
| DatenTyp | Unsigned 32 |
| Zugriff | ReadOnly |
| Default | 0h |
| EEPROM | No |
| Beschreibung | Betriebszeit in 1/10 Stunden, seit dem letzten Reset des Sensors |
| Werte | 0..n..4294967295 = n * 6 Minuten Betriebszeit ohne Reset |

**Objekt 63C9 Offset**

| | |
|---|---|
| SubIndex | 1 |
| DatenTyp | Unsigned 32 |
| Zugriff | ReadOnly |
| Default | 0h |
| EEPROM | Yes |
| Beschreibung | Aus Preset (→ Objekt 6382h) errechnet |
| Werte | 0..aktueller Gesamtmessbereich-1 |

# 4. Diagnose und Wissenswertes

## 4.1. Fehlerdiagnose Feldbus-Kommunikation

- Falls der Drehgeber über den CANopen-Bus nicht angesprochen werden kann, sollten Sie als erstes die Anschlüsse überprüfen.

  Sind die Anschlüsse in Ordnung, sollte als nächstes der Feldbusbetrieb getestet werden. Dazu wird ein CAN-Monitor benötigt, welcher die CANopen-Kommunikation aufzeichnet und die Telegramme darstellt.

- Nun sollte der Drehgeber beim Aus- und wieder Einschalten der Spannungsversorgung eine BootUp-Message absetzen.

  Sollte keine BootUp-Meldung erscheinen, prüfen Sie, ob die Baudraten des Drehgebers, des CAN-Monitors und des Bussystems übereinstimmen.

- Wenn Sie Schwierigkeiten haben die Verbindung zu einem Teilnehmer aufzunehmen, prüfen Sie die Knotennummer und die Baudrate.

  Die Baudrate muss überall gleich eingestellt werden. Die Knotennummer (Node-ID, Knotenadresse) muss zwischen 1 und 127 liegen. Jeder Busteilnehmer muss eindeutig mit einer Node-ID definiert werden. D.h. es darf auf keinen Fall mehrere Male die selbe NodeID zugeordnet werden.

  Node-ID und Baudrate können auch bequem über den LSS-Dienst eingestellt werden.

## 4.2. Fehlerdiagnose über Feldbus

Der Drehgeber verfügt über mehrere Objekte und Meldungen, welche den Status oder Fehlerzustände des Drehgebers umschreiben:

- Objekt 1001h: Dieses Objekt ist ein Error-Register für den Fehlerzustand des Gerätes.
- Objekt 1003h: In diesem Objekt werden die letzten acht Fehlercodes und Warnungen gespeichert.
- Objekt Emergency (80h + Node-ID): Hochpriore Fehlermeldung eines Teilnehmers mit Error code und Error register.
- SDO Abort Message: Falls die SDO-Kommunikation nicht korrekt abläuft, enthält die SDO-Antwort einen Abort code.

**Objekt 1001h Error register**
In diesem Register wird das Vorhandensein eines Gerätefehlers sowie dessen Art angezeigt.

Bit 0: Generic error
Die restlichen Bits werden durch unseren Drehgeber nicht unterstützt.

**Objekt 1003h Predefined error field**
In diesem Objekt werden die acht zuletzt aufgetretenen Error codes aus den Objekten 63C5h und 63C7h gespeichert, wobei der letzte Error im Subindex1 und der älteste Error unter Subindex8 eingetragen ist.

**Objekt Emergency**
Fehlermeldung eines Teilnehmers.

**SDO Abort Message**
Erfolgt die SDO-Kommunikation nicht problemlos, wird als SDO-Antwort ein Abort code gesendet:

| | |
|---|---|
| 05040001h | : Command Byte wird nicht unterstützt |
| 06010000h | : Falscher Zugriff auf ein Objekt |
| 06010001h | : Lesezugriff auf Write Only |
| 06010002h | : Schreibzugriff auf Read Only |
| 06020000h | : Objekt wird nicht unterstützt |
| 06090011h | : Subindex wird nicht unterstützt |
| 06090030h | : Wert außerhalb der Limite |
| 06090031h | : Wert zu groß |
| 08000000h | : Genereller Error |
| 08000020h | : Falsche Speichersignatur ("save") |
| 08000021h | : Daten können nicht gespeichert werden |

## 4.3. Wissenswertes zum Sensor

**Node-ID neu setzen**
1. Die Node-ID wird mit dem Baumer IVO spezifischen Objekt 2100h neu gesetzt.
2. Nach dem Setzen der Node-ID muss diese mit dem Objekt 1010h im EEPROM gespeichert werden.
3. Beim nächsten Initialisieren wird sich der Sensor mit der neuen Node-ID melden.

**Baudrate neu setzen**
1. Die Baudrate wird mit dem Baumer IVO spezifischen Objekt 2101h neu gesetzt.
2. Nach dem Setzen der Baudrate muss diese mit dem Objekt 1010h im EEPROM gespeichert werden.
3. Beim nächsten Initialisieren wird sich der Sensor auf der neuen Baudrate melden.
4. ! NICHT VERGESSEN DEN MASTER AUF DIE NEUE BAUDRATE EINSTELLEN !

**Abschirmung**
Da der Drehgeber je nach Einbaulage nicht immer auf einem definierten Erdpotential liegt, sollte der Drehgeber-Flansch zusätzlich immer mit Erdpotential verbunden werden. Grundsätzlich sollte der Drehgeber über eine abgeschirmte Leitung angeschlossen werden.
Wenn möglich sollte der Kabelschirm beidseitig aufgelegt werden. Es ist darauf zu achten, dass keine Ausgleichströme über den Drehgeber abgeleitet werden.

# 5. Applikationen

## 5.1. Objekte setzen und lesen

Um ein Objekt (SDO) zu überschreiben oder zu lesen werden immer zwei Telegramme gesendet.

**Objekt setzen**
Zuerst sendet der Master den zu setzenden Wert. Anschließend sendet der Drehgeber die Bestätigung.

Wert (ba) wird gesendet:

| COB-ID | DLC | Kommando | Objekt L | Objekt H | Subindex | Data 0 | Data 1 | Data 2 | Data 3 |
|---|---|---|---|---|---|---|---|---|---|
| 600h+Node-ID | 8 | 2Bh | 00h | 23h | 3h | a | b | x | x |

Bestätigung:

| COB-ID | DLC | Kommando | Objekt L | Objekt H | Subindex | Data 0 | Data 1 | Data 2 | Data 3 |
|---|---|---|---|---|---|---|---|---|---|
| 580h+Node-ID | 8 | 60h | 00h | 23h | 3h | 0 | 0 | 0 | 0 |

**Objekt lesen**
Zuerst sendet der Master eine Aufforderung des gewünschten Objekts. Dann sendet der Drehgeber den geforderten Wert.

Anfrage vom Master:

| COB-ID | DLC | Kommando | Objekt L | Objekt H | Subindex | Data 0 | Data 1 | Data 2 | Data 3 |
|---|---|---|---|---|---|---|---|---|---|
| 600h+Node-ID | 8 | 40h | 81h | 63h | 01h | x | x | x | x |

Antwort (dcba) des Drehgebers auf die Anfrage:

| COB-ID | DLC | Kommando | Objekt L | Objekt H | Subindex | Data 0 | Data 1 | Data 2 | Data 3 |
|---|---|---|---|---|---|---|---|---|---|
| 580h+Node-ID | 8 | 43h | 81h | 63h | 01h | a | b | c | d |

**Inbetriebnahme**
Wenn der Drehgeber an den Bus angeschlossen wird, meldet er sich mit einer BootUp-Meldung. Nun muss der Drehgeber an seine Umgebung angepasst und konfiguriert werden.

**Node-ID und Baudrate ändern mit LSS**
Node-ID und Baudrate können geändert werden, ohne den Drehgeber über diese ansprechen zu müssen. Mit dem LSS-Dienst werden die Sensoren über ProductCode, RevisionNr, VendorID und Seriennummer angesprochen und konfiguriert.

**Node-ID (Knotennr.) ändern**
Die Node-ID kann im Objekt 2101h zwischen 1 und 127 geändert werden. Anschließend sollte ein Speichervorgang mittels Objekt 1010h durchgeführt werden. Beim nächsten Initialisieren meldet sich der Drehgeber mit der neuen Node-ID an.

**Baudrate ändern**
Die Baudrate kann im Objekt 2100h geändert werden. Es wird ein Index ins Objekt geschrieben, nicht die effektive Baudrate:

| | **Baudrate** |
|---|---|
| 0 | 10 kBaud |
| 1 | 20 kBaud |
| 2 | 50 kBaud |
| 3 | 100 kBaud |
| 4 | 125 kBaud |
| 5 | 250 kBaud |
| 6 | 500 kBaud |
| 7 | 800 kBaud |
| 8 | 1000 kBaud |

Nun muss die Baudrate noch über Objekt 1010-1 gespeichert werden. Beim nächsten Initialisieren meldet sich der Drehgeber auf der neuen Baudrate an. Vorher sollte man aber noch die Baudrate des Masters ändern.

## 5.2. Konfiguration

**Position setzen (Preset, Referenzierung)**
Wert wird gesendet:

| COB-ID | DLC | Kommando | Objekt L | Objekt H | Subindex | Data 0 | Data 1 | Data 2 | Data 3 |
|---|---|---|---|---|---|---|---|---|---|
| 600h+Node-ID | 8 | 23h | 82h | 63h | 01h | a | b | c | d |

Bestätigung:

| COB-ID | DLC | Kommando | Objekt L | Objekt H | Subindex | Data 0 | Data 1 | Data 2 | Data 3 |
|---|---|---|---|---|---|---|---|---|---|
| 580h+Node-ID | 8 | 60h | 82h | 63h | 01h | 0 | 0 | 0 | 0 |

**Drehrichtung und Skalierung ändern**
Die Drehrichtung kann auf CW (Uhrzeigersinn) oder CCW (Gegenuhrzeigersinn) eingestellt werden. Zusätzlich wird im gleichen Objekt (6380) die Skalierung ein oder ausgeschaltet. Bei eingeschalteter Skalierung wird die eingestellte Auflösung verwendet. Wird jedoch die Skalierung ausgeschaltet, arbeitet der Drehgeber mit der maximalen Auflösung.

Bit 0: 0 -> CW (Uhrzeigersinn) Wert: 0
1 -> CCW (Gegenuhrzeigersinn) Wert: 1
Bit 2: 0 -> Skalierung aus Wert: 0
1 -> Skalierung ein Wert: 4

Gegenuhrzeigersinn und Skalierung ein:

| COB-ID | DLC | Kommando | Objekt L | Objekt H | Subindex | Data 0 | Data 1 | Data 2 | Data 3 |
|---|---|---|---|---|---|---|---|---|---|
| 600h+Node-ID | 8 | 23h | 80h | 63h | 01h | 5h | x | x | x |

Bestätigung:

| COB-ID | DLC | Kommando | Objekt L | Objekt H | Subindex | Data 0 | Data 1 | Data 2 | Data 3 |
|---|---|---|---|---|---|---|---|---|---|
| 580h+Node-ID | 8 | 60h | 80h | 63h | 01h | 0 | 0 | 0 | 0 |

**Singleturnauflösung ändern**
Im Objekt 6381h kann die Singleturnauflösung konfiguriert werden. Zum Beispiel 1024 (10bit) Schritte pro Umdrehung (1024 = 400h):

| COB-ID | DLC | Kommando | Objekt L | Objekt H | Subindex | Data 0 | Data 1 | Data 2 | Data 3 |
|---|---|---|---|---|---|---|---|---|---|
| 600h+Node-ID | 8 | 23h | 81h | 63h | 01h | 00 | 04 | 00 | 00 |

Bestätigung:

| COB-ID | DLC | Kommando | Objekt L | Objekt H | Subindex | Data 0 | Data 1 | Data 2 | Data 3 |
|---|---|---|---|---|---|---|---|---|---|
| 580h+Node-ID | 8 | 60h | 81h | 63h | 01h | 0 | 0 | 0 | 0 |

**Einstellungen ins EEPROM speichern**
Über das Objekt 1010h wird das Speichern untenstehender Objekte in den nichtflüchtigen Speicher (EEPROM) ausgelöst. Um ein unabsichtliches Speichern zu verhindern muss die Botschaft "save“ in den Subindex 1 geschrieben werden.

| COB-ID | DLC | Kommando | Objekt L | Objekt H | Subindex | Data 0 | Data 1 | Data 2 | Data 3 |
|---|---|---|---|---|---|---|---|---|---|
| 600h+Node-ID | 8 | 23h | 10h | 10h | 01h | 73 's’ | 61 'a’ | 76 'v’ | 65 'e’ |

| COB-ID | DLC | Kommando | Objekt L | Objekt H | Subindex | Data 0 | Data 1 | Data 2 | Data 3 |
|---|---|---|---|---|---|---|---|---|---|
| 580h+Node-ID | 8 | 60h | 10h | 10h | 01h | 0 | 0 | 0 | 0 |

## 5.3. Betrieb

**NMT-Zustände**
Nachdem der Drehgeber initialisiert wurde, befindet er sich im **Pre-Operational Mode**. In diesem Zustand können SDO gelesen und geschrieben werden.

Um die PDO-Kommunikation noch zu starten, müssen Sie einen **NMT-Start** senden. Dann befindet sich der Drehgeber im **Operational Mode**. Nun werden gewünschte PDO's gesendet. Zudem können SDO gelesen und geschrieben werden.

Wenn der Drehgeber mit einem **NMT-Stop** gestoppt wird, befindet sich der Drehgeber im **Stopped Mode**. In diesem Zustand ist nur noch NMT-Kommunikation möglich, also auch Heartbeat.

Durch einen **NMT-Reset** wird der Drehgeber wieder initialisiert und befindet sich erneut im **Pre-Operational Mode**.

**Position lesen**
Anfrage vom Master:

| COB-ID | DLC | Kommando | Objekt L | Objekt H | Subindex | Data 0 | Data 1 | Data 2 | Data 3 |
|---|---|---|---|---|---|---|---|---|---|
| 600h+Node-ID | 8 | 40h | 83h | 63h | 01h | 0 | 0 | 0 | 0 |

Antwort (dcba) des Drehgebers auf die Anfrage:

| COB-ID | DLC | Kommando | Objekt L | Objekt H | Subindex | Data 0 | Data 1 | Data 2 | Data 3 |
|---|---|---|---|---|---|---|---|---|---|
| 580h+Node-ID | 8 | 43h | 83h | 63h | 01h | a | b | c | d |

**PDO's konfigurieren**
Nach folgender Tabelle können die PDO's konfiguriert werden:

| 1906h | | 2800h | Kurzbeschreibung |
|---|---|---|---|
| Sub2 | Sub5 | | |
| FFh | 3ms | 0 | Zyklisches Senden alle 3ms |
| FFh | 5ms | 2 | Alle 5ms wird das PDO doppelt gesendet, falls eine Änderung vorliegt. |
| FFh | 0ms | 0 | PDO senden ausgeschaltet |
| FFh | 0ms | 2 | PDO senden ausgeschaltet |
| 3 | xxx | 0 | Bei jedem dritten Sync-Telegramm senden |
| 3 | xxx | 2Bh | Auf jedes dritte Sync-Telegramm, aber insgesamt nur 43mal (=2Bh). |

**Heartbeat Time festlegen**
Um die Kommunikationsfähigkeit zu überwachen muss im Objekt 1017h "Producer Heartbeat Time" die Zeit des Herzschlages definiert werden. Sobald der Wert bestätigt wurde, beginnt der Dienst zu senden. Beispiel: Alle 100ms soll der Drehgeber einen Heartbeat senden (100 = 64h):

| COB-ID | DLC | Kommando | Objekt L | Objekt H | Subindex | Data 0 | Data 1 |
|---|---|---|---|---|---|---|---|
| 600h+Node-ID | 8 | 2Bh | 17h | 10h | 0h | 64h | 0h |

Bestätigung:

| COB-ID | DLC | Kommando | Objekt L | Objekt H | Subindex | Data 0 | Data 1 |
|---|---|---|---|---|---|---|---|
| 580h+Node-ID | 8 | 60h | 17h | 10h | 0h | 0 | 0 |

| COB-ID | Data/ Remote | Byte 0 |
|---|---|---|
| 701h | d | 7Fh |

Die Heartbeat-Meldungen bestehen aus der COB-ID und einem Byte. In diesem Byte wird der NMT-Zustand überliefert.

0: BootUp-Event
4: Stopped
5: Operational
127: Pre-Operational

D.h. der Drehgeber befindet sich im Pre-Operational Modus (7Fh = 127).

# 6. Anschlussbelegung und Inbetriebnahme

## 6.1. Mechanischer Anbau

**Wellen-Drehgeber**

- Drehgebergehäuse an den Befestigungsbohrungen flanschseitig mit drei Schrauben (quadratischer Flansch mit 4 Schrauben) montieren. Gewindedurchmesser und Gewindetiefe beachten.
- Alternativ kann der Drehgeber mit Befestigungsexzentern in jeder Winkelposition montiert werden, siehe Zubehör.
- Antriebswelle und Drehgeberwelle über eine geeignete Kupplung verbinden. Die Wellenenden dürfen sich nicht berühren. Die Kupplung muss Verschiebungen durch Temperatur und mechanisches Spiel ausgleichen. Zulässige axiale oder radiale Achsbelastung beachten. Geeignete Verbindungen siehe Zubehör.
- Befestigungsschrauben fest anziehen.

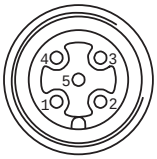
## 6.2. Elektrischer Anschluss

### 6.2.1. Beschreibung der Anschlüsse

| Pin | Belegung |
|---|---|
| CAN_L | CAN Bus Signal (dominant Low) |
| CAN_H | CAN Bus Signal (dominant High) |
| UB | Versorgungsspannung 10...30 VDC |
| GND B | Masseanschluss für UB |
| CAN_GND | Optionaler GND-Bezug für CAN Interface |

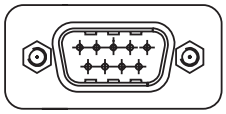
### 6.2.2. Anschlussbelegung M12-Stecker

| Pin | Belegung |
|---|---|
| 1 | GND B |
| 2 | UB |
| 3 | CAN_GND |
| 4 | CAN_H |
| 5 | CAN_L |
| | |
| | |
| | |
| | |



### 6.2.3. Anschlussbelegung D-SUB Stecker

| Pin | Belegung |
|---|---|
| 1 | -- |
| 2 | CAN_L |
| 3 | CAN_GND |
| 4 | -- |
| 5 | -- |
| 6 | GND B |
| 7 | CAN_H |
| 8 | -- |
| 9 | UB |



## 6.3. Anzeigeelemente (Statusanzeige)

Auf der Rückseite des Drehgebers kann, je nach Ausführung eine DUO-LED integriert sein.

| LED grün | LED rot | Status |
|---|---|---|
| aus | aus | Versorgungsspannung nicht angeschlossen |
| blinkt | aus | Preoperational Mode |
| ein | aus | Operational Mode |
| ein | ein | Stopped/Prepared Mode |
| aus | blinkt | Warning |
| aus | ein | Error |