Baumer

Passion for Sensors

Manual

Absolute Encoder with CANopen Lift Protocol

Firmaware version from 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 · 174.02.035/4
Subject to technical and design modifications.
Errors and omissions excepted.

Contents

		Page
1.	**Introduction**	**3**
1.1.	**Scope of delivery**	**3**
1.2.	**Product assignment**	**3**
2.	**Safety Precautions and Operating Information**	**4**
3.	**CAN Bus and CANopen Communication**	**5**
3.1.	**CAN bus**	**5**
3.1.1.	CAN bus properties	5
3.2.	**CANopen**	**6**
3.3.	**CANopen in lift construction**	**6**
3.4.	**CANopen communication**	**7**
3.4.1.	Communication profile	7
3.4.2.	CANopen message structure	7
3.4.3.	Service data communication	8
3.4.4.	Process data communication	9
3.4.5.	Emergency service	10
3.4.6.	Network management services	11
3.4.7.	Layer setting services	13
3.5.	**CANopen Lift encoder according to Appl. Profile DSP417**	**17**
3.5.1.	Object overview	17
3.5.2.	Detailed object description	20
4.	**Diagnosis and Important Information**	**34**
4.1.	**Error diagnosis field bus communication**	**34**
4.2.	**Troubleshooting with field bus**	**34**
4.3.	**Important information on sensor**	**35**
5.	**Applications**	**36**
5.1.	**Setting and reading objects**	**36**
5.2.	**Configuration**	**37**
5.3.	**Operation**	**38**
6.	**Connection Assignment and Commissioning**	**40**
6.1.	**Mechanical attachment**	**40**
6.2.	**Electrical connection**	**40**
6.2.1.	Description of connections	40
6.2.2.	Connection assignment of M12 plug	40
6.2.3.	Connection assignment of D-SUB plug	41
6.3.	**Display elements (status display)**	**41**

Disclaimer of liability
The present manual was compiled with utmost care, errors and omissions reserved. For this reason Baumer IVO GmbH & Co. KG rejects any liability for the information compiled in the present manual. Baumer IVO nor the author will accept any liability for direct or indirect damages resulting from the use of the present information.

At any time we should be pleased receiving your comments and proposals for further improvement of the present document.

1. Introduction

1.1. Scope of delivery

Please check the delivery upon completeness prior to commissioning.
Depending on encoder configuration and part number delivery is including:

- Encoder
- CD with describing file and manual (also available as download in the Internet)

1.2. Product assignment

Shaft encoder

Product	Product Code	Device Name	Eds File	Product Family
GXP5W	0x28	GXP5	GXP5_417.eds	Multiturn
X 700	0x28	GXP5	GXP5_417.eds	Multiturn

GXP5W with the Product Code 0x28 is an absolute encoder with a CANopen interface and an implemented application profile DSP417 (Application Profile for Lift Control Systems).



2. Safety Precautions and Operating Information

Additional information
- The manual is a supplement to the documentation which already exists (catalogs, product information and assembly manuals).
- It is imperative that the manual is read before commissioning.

Proper use
- The encoder is a precision measuring device. It is used exclusively for the detection of angle positions and rotations, and the processing and provision of the measured values as electrical output signals for the next device. The encoder may only be used for this purpose.

Commissioning
- The encoder may only be installed and assembled by a qualified electrician.
- Observe the operating manual of the machine manufacturer.

Safety precautions
- Check all electrical connections before commissioning the system.
- If assembly, electrical connection or other work on the encoder and on the system are not carried out properly, malfunction or failure of the encoder may result.
- The endangering of persons, damage to the system and damage to operating equipment due to the failure or malfunction of the encoder must be prevented with suitable safety measures.
- The encoder may not be operated outside the specified limits (see additional documentation).

Failure to observe the safety precautions can lead to malfunctions, damage and injuries!

Transport and storage
- The encoder may only be transported and stored in the original packaging.
- Do not drop the encoder or subject it to major jolts or impacts.

Assembly
- Avoid impacts or shocks to the housing and shaft.
- Do not torque the housing.
- Do not produce a rigid connection between the encoder and drive shafts.
- Do not open the encoder or modify it mechanically.

The shaft, ball bearings, glass plate or electronic parts can be damaged. Safe operation is then no longer ensured.

Electrical commissioning
- Do not modify the encoder electrically.
- Do not carry out wiring work while the encoder or system is energized.
- The electrical connection may not be connected or disconnected while energized.
- Install the entire system so that it is EMC-compliant. The installation environment and wiring affect the EMC of the encoder. Install the encoder and the supply lines in separate locations or at a great distance from wiring with a high interference level (frequency converters, contactors etc.).
- For consumers with high interference levels, provide a separate power supply for the encoder.
- Complete screen the encoder housing and the connection cables.
- Connect the encoder to protective ground and use shielded cables. The cable shield must be connected to the screw cable terminal or plug. Connection of the protective ground at both ends, the housing via the mechanical attachment, and the cable shield via devices connected downstream should be aimed for. If problems occur with ground loops, grounding should be carried out on at least one end.

Failure to observe the above can lead to malfunctions, damage and injuries!

3. CAN Bus and CANopen Communication

3.1. CAN bus

The CAN bus (CAN: Controller Area Network) was originally developed by Bosch and Intel for fast, inexpensive data transmission in motor vehicle technology. Today the CAN bus is also used in industrial automation.
The CAN bus is a field bus (the standards are defined by the association CAN in Automation (CiA)) via which devices, actuators and sensors of different manufacturers can communicate with each other.

3.1.1. CAN bus properties

- Data rate of 1 Mbaud with a network range of up to 40 m (130 ft)
- Network connected at both ends
- Bus medium twisted-pair cables
- Real-time capability: defined maximum wait time for high-priority messages.
- Theoretically 127 nodes to a bus, however physically only 32 (due to the driver used).
- Securing of network-wide data consistency. Faulty messages are made known to all network nodes as faulty.
- Message-oriented communication
  The message is marked with an identifier. All network nodes check whether the message is relevant for them using the identifier.
- Broadcasting, multicasting
  All network nodes receive every message simultaneously, enabling synchronization.
- Multi-master capability
  Every node in the field bus can send and receive data independently without being dependent on a priority of the master. Each can begin its message if the bus is not busy. In the case of simultaneous transmission of messages, the node with the highest priority prevails.
- Assignment of priorities to messages
  The identifier specifies the priority of the message. As a result, important messages can be transmitted quickly via the bus.
- Residual error probability
  Safe-guarding methods in the network reduce the probability of an undetected, faulty data transmission to below 10^{-11}. A virtually 100 % reliable transmission can be assumed.
- Operation monitoring
  Localization of defective or failed stations. The CAN protocol contains an operation monitoring function of network nodes. The operation of defective network nodes is restricted, or they are completely decoupled from the network.
- Data transmission with a short error recovery time
  Due to several error detection mechanisms, falsified messages are recognized with a high probability. If an error is detected, then the message is automatically retransmitted.

Several network nodes are interconnected via a bus cable. Each network node can send and receive messages. The data are transmitted serially between the network nodes.

Examples of network nodes for CAN bus devices are:
- Automation devices, e.g. PLC
- PCs
- Input/output modules
- Drive controllers
- Analysis devices, e.g. a CAN monitor
- Operating and input devices as human-machine interface (HMI)
- Sensors and actuators

3.2. CANopen

Under the technical direction of the Steinbeis Transfer Center for Automation, the CANopen profile was developed on the basis of the Layer 7 specification CAL (CAN Application Layer). Compared to CAL, CANopen only contains the functions suitable for this application. CANopen therefore represents a subset of CAL optimized for the application and enables simplified system design and the use of simplified devices. CANopen is optimized for rapid data exchange in real-time systems.
The CAN in Automation (CiA) organization is responsible for the applicable standards of the corresponding profiles.
CANopen enables:
• Simple access to all device and communication parameters
• Synchronization of several devices
• Automatic network configuration
• Cyclical and event-controlled process data traffic

CANopen consists of four communication objects (COB) with different properties:
• Process data objects for real-time data (PDO)
• Service data objects for parameter and program transmission (SDO)
• Network management (NMT, Heartbeat)
• Predefined objects (for synchronization, emergency message)

All device and communication parameters are structured in an object directory. An object comprises the name of the object, data type, number of sub-indices, structure of the parameters and the address. According to the CiA, this object directory is divided into three different parts: communication profile, device profile and a manufacturer-specific profile (see Object directory).

3.3. CANopen in lift construction

CANopen is a standardized application for distributed, industrial automation systems based on CAN and on the communication standard CAL. CANopen is a standard of the CAN in Automation (CiA) and already found extremely broad acceptance shortly after it became available. In Europe CANopen can be considered the decisive standard for the realization of industrial CAN-based system solutions.
The responsibility was turned over to the organization CAN-in-Automation, which coordinates all work on the CANopen system worldwide. Within the CiA a special interest group (SIG) "Lift" was founded. The task of this SIG is the checking of the existing profiles for their suitability for lift construction and their expansion or redefinition. Within the SIG "Lift" various working groups for the individual profiles were specified.

3.4. CANopen communication

3.4.1. Communication profile

The communication between the network nodes and the master (PC/controller) is carried out via object directories and objects. The objects are addressed via a 16-bit index. The CANopen communication profile DS 301 standardizes the various communication objects. Accordingly, they are divided into several groups:

- Process data objects (PDO) for the real-time transmission of process data
- Service data objects (SDO) for the write and read access to the object directory
- Objects for the synchronization and error display of CAN nodes:
  - SYNC object (synchronization object) for the synchronization of network nodes
  - EMCY object (emergency object) for the error display of a device or its periphery
- Network management (NMT) for the initialization and network control
- Layer setting services (LSS) for the configuration by means of a serial number, revision number etc. within an existing network

3.4.2. CANopen message structure

The first part of a message is the COB-ID (identifier).
Structure of the 11-bit COB-ID:

Function Code				Node-ID						
4-bit Function Code				7-bit Node-ID						

The function code provides information on the type of message and the priority
The lower the COB-ID, the higher the priority of the message.

Broadcast messages:

Function Code	COB-ID
NMT	0
SYNC	80h

Peer to peer messages:

Function Code	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) and (rx) from the standpoint of the encoder

The node ID can be freely selected via the CANopen bus between 1 and 127 (if rotary switch = 0). The encoders are shipped with a node ID = 1.
A change is made with the service data object 2101h or via LSS.

A CAN telegram consists of the COB-ID and up to 8 bytes of data:

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

The exact telegrams will be listed in detail later.

3.4.3. Service data communication

The service data objects comply with the CiA standards. An object can be accessed via an index and subindex. The data can be requested or, if necessary, written to the object.

General information on the SDOs
Structure of an **SDO telegram**:

COB-ID	DLC	Command	Object L	Object H	Subindex	Data 0	Data 1	Data 2	Data 3

An SDO **COB-ID** consists of the following:
Master -> Encoder : 600h + Node-ID
Encoder -> Master : 580h + Node-ID

DLC (data length code) designates the length of the telegram. This consists of the following:
1 byte command + 2 bytes object + 1 byte subindex + number of data bytes (0 - 4).

The **command byte** specifies whether data are read or set and how many data bytes are concerned:

SDO Command	Description	Data Length	
22h	Download Request	Max. 4 bytes	Send parameters to encoder
23h	Download Request	4 bytes	
2Bh	Download Request	2 bytes	
2Fh	Download Request	1 byte	
60h	Download Response	-	Confirmation of adoption to Master
40h	Upload Request	-	Request parameters from encoder
42h	Upload Response	Max. 4 bytes	Parameters to Master with max. 4 bytes
43h	Upload Response	4 bytes	
4Bh	Upload Response	2 bytes	
4Fh	Upload Response	1 byte	
80h	Abort Message	-	Encoder signals error code to Master

An **abort message** indicates an error in the CAN communication. The SDO command byte is 80h. The object and the subindex are those of the desired object. The error code is present in byte 5 - 8.

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

Byte 8 - 5 results in the SDO abort message (Byte 8 = MSB).
The following messages are supported:

05040001h : Command byte is not supported
06010000h : Incorrect access to an object
06010001h : Read access to Write Only
06010002h : Write access to Read Only
06020000h : Object is not supported
06090011h : Subindex is not supported
06090030h : Value outside limits
06090031h : Value too large
08000000h : General error
08000020h : Incorrect memory signature ("save")
08000021h : Data cannot be saved

Example SDOs

Request for a value from the Master by the Slave
A frequent request is for the position. → Object 6383h Subindex 1

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

Response of the Slave **to the request** for a value
The position is 4 bytes long, and the exact values are specified under Object 6383h Subindex 1.

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

Writing of a value from the Master to the Slave
A position can be set with Preset. → Object 6382h

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

Response of the Slave **to the writing** of a value

COB-ID	DLC	Command	Object L	Object 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. Process data communication

Process data objects are used for real-time data exchange for process data, e.g. the position or the operating status. PDOs can be transmitted synchronously or cyclically (asynchronously). The encoder supports PDO1 .. This supplies the current encoder position and its speed, and is defined in Objects 1906h and 1B06hh.

Synchronous
To transmit the process data synchronously, a value between 1 and F0h (=240) must be entered in the object 1906h Subindex 2. If the value is now 3, the PDO is transmitted to every third Sync telegram (with a value of 1 it is transmitted to every Sync telegram) until 0 is entered in the object 2800h 0. If, for example, a 5 is entered there, the PDO is still written to every third Sync telegram, however a total of only 5 times. In accordance with this, the 15th Sync telegram is followed by the last PDO. The counter for the number of PDOs to be transmitted is reset via a position change or the NMT reset. This means the position is transmitted 5 times unless it changes. If the position changes, it is transmitted again 5 times.

In the synchronous mode, the PDOs are requested by the Master via the Sync telegram:

Byte 0	Byte 1
COB-ID = 80	0

Cyclical (asynchronous)
If the PDOs are to be transmitted cyclically, the value FFh must be written to Object 1906h Subindex 2. In addition, the cycle time in milliseconds must be entered in the same object, Subindex 5. The entered time is rounded off to 1ms. If the value 0 ms is saved, the PDOs are not transmitted and the function is deactivated. Another option is provided by Object 2800h. If the value is 0, the cyclical transmission runs as described above. If the value is 1, it is cyclically checked whether a change has been made in the value. If not, no transmission takes place. If the value is 4, the PDO is transmitted 4 times per cycle if a change exists.

Overview
The following table provides a summary of the various transmission types of PDOs:

1906h		2800h	Brief Description
Sub2	Sub5		
FFh	3ms	0	Cyclical transmission every 3 ms
FFh	5ms	2	The PDO is sent double every 5 ms if a change is present.
FFh	0ms	0	PDO transmission is deactivated
FFh	0ms	2	PDO transmission is deactivated
3	xxx	0	Transmit for every third Sync telegram
3	xxx	2Bh	Every third Sync telegram, but only 43 times total (=2Bh).

PDO
PDO1 telegram structure:

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
Length : 6 data bytes
Byte1 - 4 : Current position in increments
Byte 5 - 6 : Speed

3.4.5. Emergency service

Internal device error or bus problems trigger an emergency message:

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	Value = 0	Value = 1
0	Position error active	No	Yes

Byte 5..6 Warning

Bit	Meaning	Value = 0	Value = 1
2	CPU watchdog status	OK	Reset carried out
4	Battery charge	OK	Charge too low

Byte 7: Not in use

3.4.6. Network management services

The network management can be divided into two groups:
With the NMT services for **device control**, the bus nodes can be initialized, started and stopped.
In addition, there are also NMT services for **connection monitoring**.

Description of NMT commands
The commands are transferred as unconfirmed objects and are structured as follows:

Byte 0	Byte 1	Byte 2
COB-ID = 0	Command Byte	Node Number

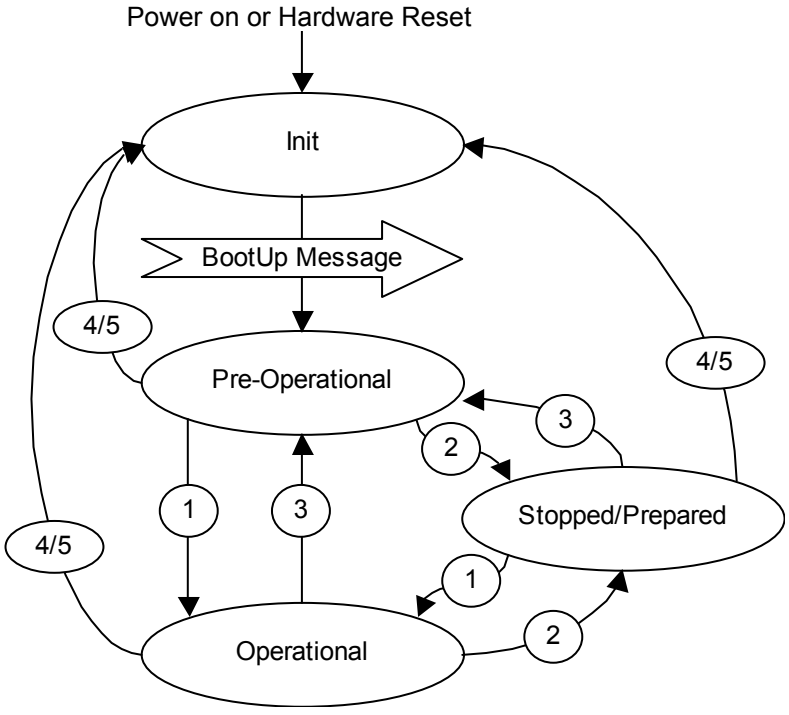
COB-ID for NMT commands is always zero. The Node-ID is transmitted in Byte 2 of the NMT command.

Command Byte

Command Byte	Description	In State Event Drawing
01h	Start Remote Node	1
02h	Stop Remote Node	2
80h	Enter Pre-Operational Mode	3
81h, 82h	Reset Remote Node	4, 5

The **node number** corresponds to the Node-ID of the desired node. With none number = 0 all nodes are addressed.

NMT State Event
Following initialization the encoder is in the Pre-Operational mode. In this state SDO parameters can be read and written. To request PDO parameters, the encoder must first be run in the Operational Mode state.



Different NMT states

Init
Following initialization the encoder sends a BootUp message on the CAN bus. Then the encoder automatically switches into PreOperational mode.

The COB-ID of the BootUp message consists of 700h an the Node-ID.

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

Pre-Operational Mode
In the Pre-Operational mode, SDOs can be read and written.

Operational Mode
In the Operational mode the encoder sends the desired PDOs. In addition, SDOs can be read and written.

Stopped or Prepared Mode
In the Stopped mode only NMT communication is possible. No SDO parameters can be read or written. LSS is only possible in the Stopped mode.

State change

Start Remote Node (1)
With the Start command, the encoder is switched into the Operational mode.

COB-ID	Command Byte	Node Number
0	1h	0 - 127

Stop Remote Node (2)
With the Stop command the encoder is switched into the Stopped or Prepared mode.

COB-ID	Command Byte	Node Number
0	2h	0 - 127

Enter Pre-Operational Mode (3)
Switches into the Pre-Operational mode.

COB-ID	Command Byte	Node Number
0	80h	0 - 127

Reset Remote Node (4) or Reset Communication (5)
With the Reset command the encoder is reinitialized.
Reset Remote Node (4):

COB-ID	Command Byte	Node Number
0	81h	0 - 127

Reset Communication (5):

COB-ID	Command Byte	Node Number
0	82h	0 - 127

Heartbeat
A "Heartbeat" principle is defined in the CAL with which the communication capability of CANopen devices can be monitored. It can be ensured that the Master could react to a failure.
In Object 1017h "Producer Heartbeat Time" the time of the heartbeat can be defined. As soon as the value has been confirmed, the heartbeat begins to transmit.

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

The heartbeat messages consist of the COB-ID and a byte. This byte is used to transmit the NMT state.

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

That means the encoder is in the Pre-Operational mode (7Fh = 127).

3.4.7. Layer setting services

In the spring of 2000 a new protocol was drafted by CiA to ensure a uniform occurrence. The procedure is described under *Layer Setting Services and Protocol, CiA Draft Standard Proposal 305* (LSS).
In the standard configuration we ship the encoder with the Node-ID 1 and the baud rate 50 kBaud.
Several encoders with the same Node-ID can be connected to the bus system. To now be able to address the individual encoders, LSS is used.
Each encoder has a unique serial number and is addressed via this number. This means any desired number of encoders with the same Node-ID can be connected to a bus system and then initialized with LSS. Both the Node-ID and the baud rate can be reset. LSS can only be carried out in the **Stopped Mode**.

Message structure

COB-ID:
Master → Slave : 2021 = 7E5h
Master ← Slave : 2020 = 7E4h
After the COB-ID an LSS command specifier is transmitted.
Then up to seven data bytes are added on.

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 → Operation mode
1 → Configuration mode

Switch Mode Selective
With the following procedure a very specific encoder can be addressed in the bus system.

7E5h →	40h	VendorID	Reserved
7E5h →	41h	ProductCode	Reserved
7E5h →	42h	RevisionNumber	Reserved
7E5h →	43h	SerialNumber	Reserved
7E4h ←	44h	Mode	Reserved

VendorID : ECh
ProductCode : Internal product code for the respective encoder
RevisionNumber : Current revision number of the encoder
SerialNumber : Unique, consecutive serial number
Mode : Response of the encoder is the new mode (0 = Operation mode; 1 = Configuration mode)

Set Node-ID

7E5h →	11h	Node-ID	Reserved

7E4h ←	11h	ErrCode	Spec Error	Reserved

Node-ID : The new Node-ID of the encoder
ErrorCode : 0=OK; 1=Node-ID outside range; 2..254 = reserved; 255→specificError
SpecificError : If ErrorCode = 255 → application-specific error code.

Set BitTiming

7E5h →	13h	tableSel	tableInd	Reserved

7E4h ←	13h	ErrCode	SpecError	Reserved

TableSel : Selects the BitTiming table 0 : Standard CiA bit timing table
1 - 127 : Reserved for CiA
128 - 255: Manufacturer-specific tables
TableInd : BitTiming entry in selected table (see table below).
ErrorCode : 0=OK; 1=BitTiming outside range; 2 - 254 = reserved; 255→SpecificError
SpecificError : Falls ErrorCode=255 → Application-specific error code.

Standard CiA table

Baud Rate	Table 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

Saving configuration protocol
This protocol saves the configuration parameters in the EEPROM.

7E5h →	17h	Reserved

7E4h ←	17h	ErrCode	SpecError	Reserved

ErrorCode : 0=OK;1 = Saving not supported; 2 = access error; 3 - 254 = Reserved;255→specificError
SpecificError : If ErrorCode = 255 → Application-specific error code.

Activate BitTiming Parameters
The new BitTiming parameters are activated with the command specifier 21.

7E5h →	15h	Switch Delay	Reserved

Switch Delay : Delay of the reset in the Slave in ms.
After the delay time, the encoder signals the new baud rate.

Request VendorID
Request the VendorID of a selected encoder

7E5h →	5Ah	Reserved

7E4h ←	5Ah	32-Bit Vendor ID	Reserved

VendorID : = ECh

Request ProductCode
Request the ProductCode of a selected encoder

7E5h →	5Bh	Reserved

7E4h ←	5Bh	ProductCode	Reserved

ProductCode : Manufacturer-dependent product code

Request Revision Number
Request the RevisionNumber of a selected encoder

7E5h →	5Ch	Reserved

7E4h ←	5Ch	32-Bit Revision Number	Reserved

RevisionNumber : Current revision

Request SerialNumber
Request the SerialNumber of a selectable encoder

7E5h →	5Dh	Reserved

7E4h ←	5Dh	32-Bit Serial Number	Reserved

SerialNumber : Unique consecutive serial number of the encoder

Range Request
Encoders can also be searched for in a certain range. For this purpose, the following objects are sent consecutively:

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

Each encoder with the corresponding parameters responds by transmitting the following message:

7E4h ←	4Fh	Reserved

3.5. CANopen Lift encoder according to Appl. Profile DSP417

3.5.1. Object overview

According to CiA (CAN in Automation), the objects are divided into three groups:

- **Standard objects**:
  1000h, 1001h, 1018h
- **Manufacturer-specific objects**:
  2000h - 5FFFh
- **Application-specific objects**:
  6000h - 67FFh

The following table shows a summary of all SDO objects supported by the device.

Object Object number in hex
Name Description of object
Type U/I = Unsigned/Integer , No. = Number of bits, ARR = Array
Attr ro = ReadOnly, wo = WriteOnly, rw = ReadWrite
Default Default value for first initialization
EE 1= If saving to EEPROM is carried out
Info Additional information

Object	Name	Type	Attr	Default	EE	Info
1000h	Device Type	U32	ro	060001A1h		$2^0..2^{15}$ Profile No. = 1A1h=417 encoder lift $2^{16}..2^{23}$ Communication model $2^{24}..2^{31}$ virtual device code
1001h	Error Register	U8	ro	00h		Bit0=Generic Error
1003h	Predefined Error Field	ARR				Contains last 8 errors or warnings
00h	Largest Subindex	U8	rw	00h		Number of messages saved (0 - 8)
01h	Last Entry	U32	ro			Error or warning
..	..	..	..	..		..
08h	Oldest Entry	U32	ro			Error or warning
1005h	Sync COB-ID	U32	rw	80h	1	COB-ID of Sync object
1008h	Device Name	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	Largest Subindex	U8	ro	04h		Number of memory possibilities = 4
01h	Save All Parameters	U32	rw			=“save“ (0x73617665) to save
02h	Communication Parameters	U32	rw			=“save“ (0x73617665) to save
03h	Application Parameters	U32	rw			=“save“ (0x73617665) to save
04h	Manufacturer Specific Parameters	U32	rw			=“save“ (0x73617665) to save
1011h	Restore Default Parameters	ARR				
00h	Largest Subindex	U8	ro	04h		Number of reset possibilities = 4
01h	All Parameters	U32	rw			=“load“ (0x6C6F6164) to load
02h	Communication Parameters	U32	rw			=“load“ (0x6C6F6164) to load
03h	Application Parameters	U32	rw			=“load“ (0x6C6F6164) to load
04h	Manufacturer Specific Parameters	U32	rw			=“load“ (0x6C6F6164) to load
1014h	Emergency COB-ID	U32	rw	80h + Node-ID	1	COB-ID of Emergency object
1017h	Producer Heartbeat Time	U16	rw	00h	1	
1018h	Identity Object	U32	ro			
00h	Largest Subindex	U8	ro	04h		
01h	VendorID	U32	ro	ECh	1	Vendor number assigned by CiA
02h	Product Code	U32	ro	1Eh	1	
03h	Revision Number	U32	ro	00010000h		Current revision number V1.00
04h	Serial Number	U32	ro	xyz	1	Unique consecutive serial number

1906h	Transmit PDO1 Parameter	REC				
00h	Largest Subindex	U8	ro	05h		
01h	COB-ID	U32	rw	18Ch	1	PDO ID = 17Bh + Lift No10 + PosUnit
02h	PDO Type	U8	rw	FFh	1	FFh= cyclical
05h	Event Timer	U16	rw	00h	1	Cycle time in ms
1B06h	Transmit PDO1 Mapping	ARR				Static mapping only!!
00h	Largest Subindex	U8	ro	02h		
01h	Content of PDO1	U32	ro	63830120h		Position valve 32-bit
02h	Content of PDO1	S16	ro	63900110h		Speed 16-bit
2100h	Baud Rate	U8	rw	02h	1	After baud rate is set, EEPROM must be saved and reinitialized 0=10 kBit/s 1=20 kBit/s 2=50 kBit/s 3=100 kBit/s 4=125 kBit/s 5=250 kBit/s 6=500 kBit/s 7=800 kBit/s 8=1000 kBit/s
2101h	Node-ID	U8	rw	01h	1	Node number 1 - 127 possible. After baud rate is set, EEPROM must be saved and reinitialized.
2110h	Customer-Specific Node-ID Version	U32	ro	01h	1	Define treatment of Node-ID
2201h	Statistics	REC				
00h	Largest Subindex	U8	ro	03h		Number of subindexes
01h	Number of Position Errors	U32	ro	00h	1	Position check
02h	Time in Seconds	U32	ro	00h	1	Time since last reset
03h	Number of Timer Reset Watchdog	U32	ro	00h	1	Timer Watchdog
2300h	Customer EEPROM Area	ARR				Any desired data can be saved
00h	Largest Subindex	U8	ro	08h		in this object
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 Addition/Event Trigger	U8	rw	00h	1	Repetition counter for PDO1
6001h	Lift Number	U8	rw	01h	1	Lift number 1 - 8 (currently only 1 possible)
6380h	Operating Parameters	ARR				
00h	Largest Subindex	U8	ro	04h		
01h	Position Unit 1	U16	rw	04h	1	Bit0 = Rotating direction 0 = CW, 1 = CCW Bit2 = Scaling Function 0 = off, 1 = on CW = Increasing position value during clockwise rotation (when viewing flange) CCW = Increasing position value during counter-clockwise rotation (when viewing flange)
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	Resolution	ARR				
00h	Largest Subindex	U8	ro	04h		
01h	Position Unit 1	U32	rw	8192	1	Resolution in steps/rotation or mm/rotation
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 Value in Steps	ARR				
00h	Largest Subindex	U8	ro	04h		
01h	Position Unit 1	U32	rw	00h	1	Preset in steps → 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 Steps	ARR	ro			
00h	Largest Subindex	U8	ro	04h		
01h	Position Unit 1	U32	ro			Position value incl. offset in steps
02h	Position Unit 2	U32	ro			
03h	Position Unit 3	U32	ro			
04h	Position Unit 4	U32	ro			

6384h	Encoder Meas. Step Setting	ARR	ro			Position of Unit 1
00h	Largest Subindex	U8	ro	03h		
01h	Measuring Step	U16	rw	64h	1	Unit of measuring step, multiple of 10 µm
02h	Speed Measuring Step	U16	rw	0Ah	1	Unit of speed, multiple of 0.1 mm/s
03h	Acceleration Measuring Step	U16	rw	0Ah	1	Unit of acceleration, multiple of 1 mm/s²
6390h	Speed Value Car	ARR				
00h	Largest Subindex	U8	ro	04h		
01h	Position Unit 1	S16	ro			Speed (see Object 6384 for unit)
02h	Position Unit 2	S16	ro			
03h	Position Unit 3	S16	ro			
04h	Position Unit 4	S16	ro			
63C0h	Operating Status	ARR				
00h	Largest Subindex	U8	ro	04h		
01h	Position Unit 1	U16	ro	04h		Bit0 = Rotating direction 0 = CW, 1 = CCW Bit2 = Scaling Function 0 = off, 1 = on CW = Increasing position value during clockwise rotation (when viewing flange) CCW = Increasing position value during counter-clockwise rotation (when viewing flange)
02h	Position Unit 2	U16	ro	04h		
03h	Position Unit 3	U16	ro	04h		
04h	Position Unit 4	U16	ro	04h		
63C1h	Max. Resolution	ARR				
00h	Largest Subindex	U8	ro	04h		
01h	Position Unit 1	U32	ro	2000h		Maximum resolution in steps/rotation
02h	Position Unit 2	U32	ro	2000h		
03h	Position Unit 3	U32	ro	2000h		
04h	Position Unit 4	U32	ro	2000h		
63C2h	Max. Rotations	ARR				
00h	Largest Subindex	U8	ro	04h		
01h	Position Unit 1	U16	ro	10000h		Max. number of rotations
02h	Position Unit 2	U16	ro	10000h		
03h	Position Unit 3	U16	ro	10000h		
04h	Position Unit 4	U16	ro	10000h		
63C4h	Supported Warnings	ARR				
00h	Largest Subindex	U8	ro	04h		
01h	Position Unit 1	U16	ro	14h		Following warnings are supported: Bit2 = CPU watchdog status Bit4 = Battery charge
02h	Position Unit 2	U16	ro	14h		
03h	Position Unit 3	U16	ro	14h		
04h	Position Unit 4	U16	ro	14h		
63C5h	Warnings	ARR				
00h	Largest Subindex	U8	ro	04h		
01h	Position Unit 1	U16	ro	00h		Following warnings are evaluated: Bit2 = CPU watchdog status Bit4 = Battery charge
02h	Position Unit 2	U16	ro	00h		
03h	Position Unit 3	U16	ro	00h		
04h	Position Unit 4	U16	ro	00h		
63C6h	Supported Alarms	ARR				
00h	Largest Subindex	U8	ro	04h		
01h	Position Unit 1	U16	ro	01h		Following alarms are supported: Bit0=Position error
02h	Position Unit 2	U16	ro	01h		
03h	Position Unit 3	U16	ro	01h		
04h	Position Unit 4	U16	ro	01h		
63C7h	Alarms	ARR				
00h	Largest Subindex	U8	ro	04h		
01h	Position Unit 1	U16	ro	00h		Following alarms are evaluated: Bit0=Position error
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	Largest Subindex	U8	ro	4h		
01h	Position Unit 1	U32	ro	0h		Time in 1/10 hours since last Reset
02h	Position Unit 2	U32	ro	0h		
03h	Position Unit 3	U32	ro	0h		
04h	Position Unit 4	U32	ro	0h		
63C9h	Offset Value	ARR				
00h	Largest Subindex	U8	ro	4h		
01h	Position Unit 1	U32	ro	0h		Offset calculated from Preset → 6382 h
02h	Position Unit 2	U32	ro	0h		
03h	Position Unit 3	U32	ro	0h		
04h	Position Unit 4	U32	ro	0h		

3.5.2. Detailed object description

Object 1000 Device Type

Subindex	0
Data Type	Unsigned 32
Access	ReadOnly
Default	060001A1h
EEPROM	No
Description	Information on device profile and device model
Values	$2^0..2^{15}$ Profile No.= 1A1h=417 (Application Profile for Lift Control Systems). $2^{16}..2^{23}$ Communication model $2^{24}..2^{31}$ Virtual device code

Object 1001 Error Register

Subindex	0
Data Type	Unsigned 8
Access	ReadOnly
Default	0h
EEPROM	No
Description	Current Error Code
Values	Bit 0 1 = Generic Error

Object 1003 Predefined Error Field

Here CiA (CAN in Automation) defines approximately 200 different Error Codes.
This document only describes the Error Codes relevant for the sensor.
This object saves the last 8 errors or warnings which have occurred.

Subindex	0
Data Type	Unsigned 8
Access	ReadWrite
Default	0
EEPROM	No
Description	Read: Number of errors or warnings Write 0: Reset errors
Values	0 - 8

Subindex	1..8
Data Type	Unsigned 32
Access	ReadOnly
Default	0
EEPROM	No
Description	Errors or warnings which have occurred, whereby Subindex 1 is the last entry, Subindex 2 the next to the last etc.
Values	Not yet defined

Object 1005 COB-ID SYNC Message

Subindex	0
Data Type	Unsigned 32
Access	ReadWrite
Default	80h
EEPROM	Yes
Description	Defines COB-ID of synchronization object (SYNC)
Values	Bit 31 Not defined Bit 30 1=Sensor generates SYNC messages, 0 = generates no SYNC message Bit 29 1 = 29-bit SYNC COB-ID (CAN 2.0B), 0 = 28-bit SYNC COB-ID (CAN 2.0A) Bit 28..11 Bit 28 - 11 of 29-bit SYNC COB-ID Bit 10..0 Bit 10 - 0 of SYNC COB-ID

Object 1008 Manufacturer Device Name

Subindex	0
Data Type	Unsigned 32
Access	ReadOnly
Default	"GXP5"
EEPROM	No
Description	Device designation in ASCII
Values	Data 0 - 3: "GXP5" = 47h 58h 50h 35h → GXP5W Multiturn

Object 1009 Manufacturer Hardware Version

Subindex	0
Data Type	Unsigned 32
Access	ReadOnly
Default	
EEPROM	No
Description	Hardware version in ASCII
Values	Data 0 - 3 example: 31h 2Eh 30h 30h = "1.00“

Object 100A Manufacturer Software Version

Subindex	0
Data Type	Unsigned 32
Access	ReadOnly
Default	
EEPROM	No
Description	Software version in ASCII
Values	Data 0 - 3 See rating plate

Object 1010 Save Parameters

With Object 1010h the saving of the objects below in the non-volatile memory (EEPROM) is triggered. To prevent accidental saving, the message "save" must be written to Subindex 1.

COB-ID	DLC	Command	Object L	Object 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'

Objects saved in the EEPROM:

Object	Subindex	Description	Default Value (as per Object 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
1018h	4h	Serial Number	Xyz
1906h	1h	PDO1 ID	PDO ID = 17Bh + LiftNr * 10 + PosUnit
1906h	2h	PDO1 Type	FFh -> asynchronous, cyclical
1906h	5h	PDO1 Event Timer Asynchronous Operation	0h (disabled)
2100h	0h	Baud Rate	2h = 50 kBaud
2101h	0h	Node-ID	1h
2110h	0h	Manufacturer_Options	8h
2201h	1h	Number of Position Errors	0h
2201h	2h	Total Operating Time in Seconds	0h
2201h	3h	Number of Timer Resets from Watchdog	0h
2300h	1h	Customer-Specific EEPROM Range Data0	0h
2300h	2h	Customer-Specific EEPROM Range Data1	0h
2300h	3h	Customer-Specific EEPROM Range Data2	0h
2300h	4h	Customer-Specific EEPROM Range Data3	0h
2300h	5h	Customer-Specific EEPROM Range Data4	0h
2300h	6h	Customer-Specific EEPROM Range Data5	0h
2300h	7h	Customer-Specific EEPROM Range Data6	0h
2300h	8h	Customer-Specific EEPROM Range Data7	0h
2800h	0h	PDO1 Addition (Event Trigger)	0h
6001h	0h	Lift Number	1h
6380h	1h	Operating Parameter	0004h
6381	1h	Triggering in Steps/Rotation	2000h
6382h	1h	Preset Value in Steps	0h
6384h	01h	Encoder Measuring Step Setting Unit for Measuring Step: Multiple of 10 µm	64h
6384h	02h	Encoder Measuring Step Setting Unit for Speed: Multiple of 0.1 mm/s	0Ah
6384h	03h	Encoder Measuring Step Setting Unit for Acceleration: Multiple of 1 mm/s^2	0Ah

Object 1011 Restore Parameters

With Object 1011h the values in the RAM are overwritten with the Default Values (see Object 1010h). In addition, the content of the EEPROM is marked as invalid. This means that up to the next saving of the data in the EEPROM, the respective Default Values are loaded.
To prevent accidental overwriting, the message "load" must be written to Subindex 1.

COB-ID	DLC	Command	Object L	Object 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'

Object 1014 COB-ID Emergency Message

Subindex	0
Data Type	Unsigned 32
Access	ReadWrite
Default	80h+Node-ID
EEPROM	Yes
Description	Defines COB-ID of Emergency object
Values	80h + Node-ID

Object 1017 Producer Heartbeat Time

Subindex	0
Data Type	Unsigned 16
Access	ReadWrite
Default	0h
EEPROM	Yes
Description	Defines repetition time of Heartbeat monitoring service
Values	0 = Disabled 1..65535 = Repetition time in ms

Object 1018 Identity Object

Subindex	0
Data Type	Unsigned 8
Access	ReadOnly
Default	4
EEPROM	No
Description	Largest supported Subindex
Values	4 = Largest supported Subindex

Subindex	1
Data Type	Unsigned 32
Access	ReadOnly
Default	ECh
EEPROM	Yes
Description	VendorID for IVO GmbH & Co. KG assigned by CiA
Values	ECh (on Internet at www.can-cia.de)

Subindex	2
Data Type	Unsigned 32
Access	ReadOnly
Default	28h
EEPROM	Yes
Description	Product Code
Values	28h → GXP5W Multiturn

Subindex	3
Data Type	Unsigned 32
Access	ReadOnly
Default	
EEPROM	No
Description	Revision Number des Sensors
Values	<table><tr><td>Data 0 = Build number LOW</td><td>Data 1 = Build number HIGH</td><td>Data 2 = Version LOW</td><td>Data 3 = Version HIGH</td></tr></table> Version of current software = xxyy (xx = Version, yy = Serial number) See rating plate

Subindex	4
Data Type	Unsigned 32
Access	ReadOnly
Default	0
EEPROM	Yes
Description	Consecutive unique sensor serial number
Values	Defined at factory during final test

Object 1906 PDO1 Parameter

Subindex	0
Data Type	Unsigned 32
Access	ReadOnly
Default	5
EEPROM	No
Description	Largest supported Subindex
Values	5

Subindex	1
Data Type	Unsigned 32
Access	ReadWrite
Default	18Ch
EEPROM	Yes
Description	COB-ID of PDO
Values	17Bh + Lift No * 10 + PosUnit

Subindex	2
Data Type	Unsigned 8
Access	ReadWrite
Default	FFh
EEPROM	Yes
Description	PDO Type
Values	1..n..F0h = PDO has synchronous characteristic (PDO is sent for each nth SYNC telegram) FFh = PDO has asynchronous characteristic (PDOs are sent cyclically in dependence on Event Timer and Event Trigger)

Subindex	5
Data Type	Unsigned 16
Access	ReadWrite
Default	203h
EEPROM	Yes
Description	Event Timer for Process Data Object
Values	0 = Cyclical transmission deactivated 1..n..65535 =Repetition time of cyclical transmission is n ms.

Object 1B06 PDO1 Mapping

Subindex	0
Data Type	Unsigned 8
Access	ReadOnly
Default	0
EEPROM	No
Description	Largest supported Subindex
Values	1

Subindex	1
Data Type	Unsigned 32
Access	ReadOnly
Default	63830120h
EEPROM	No
Description	Describes content of PDO1 message
Values	6383h = Position (Subindex 01, Length 0x20)

Subindex	2
Data Type	Signed 16
Access	ReadOnly
Default	63900110h
EEPROM	No
Description	Describes content of PDO1 message
Values	6390h = Speed (Subindex 01, Length 0x10)

Object 2100 Baud Rate

Subindex	0
Data Type	Unsigned 8
Access	ReadWrite
Default	2 = 50kBaud
EEPROM	Yes
Description	Read or set new sensor baud rate. → After setting, parameters must be saved in EEPROM with Object 1010h and then sensor must be reinitialized
Values	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

Object 2101 Node-ID

Subindex	0
Data Type	Unsigned 8
Access	ReadWrite
Default	1
EEPROM	Yes
Description	Read or set new sensor Node-ID. → After setting, parameters must be saved in EEPROM with Object 1010h and then sensor must be reinitialized
Values	1 - 127

Object 2110 Manufacturer_Options

Subindex	0
Data Type	Unsigned 32
Access	ReadWrite
Default	8h
EEPROM	Yes
Description	Settings for ensuring compatibility to older sensors
Values	Bit 3 =1 : Reset after Bus Off

Object 2201 Statistics

Subindex	0
Data Type	Unsigned 8
Access	ReadOnly
Default	3h
EEPROM	No
Description	Largest supported Subindex
Values	3

Subindex	1
Data Type	Unsigned 32
Access	ReadOnly
Default	0h
EEPROM	Yes
Description	Total number of position errors
Values	0...4294967295

Subindex	2
Data Type	Unsigned 32
Access	ReadOnly
Default	0h
EEPROM	Yes
Description	Total operating time in seconds (Object 6508h Time since last Reset)
Values	0... 4294967295

Subindex	3
Data Type	Unsigned 32
Access	ReadOnly
Default	0h
EEPROM	Yes
Description	Watchdog Timer Reset counter
Values	0 - 4294967295

Object 2300 Customer EEPROM Area

Subindex	0
Data Type	Unsigned 8
Access	ReadOnly
Default	8h
EEPROM	No
Description	Any desired data can be saved in this object
Values	8

Subindex	1 - 8
Data Type	Unsigned 16
Access	ReadWrite
Default	0h
EEPROM	Yes
Description	One 16-bit value can be saved per Subindex (saving in EEPROM with Object 1010h)
Values	0

Object 2800 PDO1 Addition (Event Trigger)

Subindex	0
Data Type	Unsigned 8
Access	ReadWrite
Default	0h
EEPROM	Yes
Description	Event Trigger value determines how often same PDO value is transmitted
Values	0 = PDO counter is deactivated → constant transmission (time basis from Event Timer) 1..n..255 = Same PDO value is transmitted n times (time basis from Event Timer)

Object 6001 Lift Number

Subindex	0
Data Type	Unsigned 8
Access	ReadWrite
Default	1
EEPROM	Yes
Description	Lift number
Values	1

Object 6380 Operating Parameters

Subindex	0
Data Type	Unsigned 16
Access	ReadWrite
Default	4
EEPROM	Yes
Description	Operating parameters
Values	Bit 0 Rotating direction = 0 → Clockwise; 1 → Counter-clockwise Bit 2 Scaling function = 0 → Max. resolution; 1 → Saved resolution

Object 6381 Resolution

Subindex	1
Data Type	Unsigned 32
Access	ReadWrite
Default	2000h = 8192 = 13Bit
EEPROM	Yes
Description	Number of steps per rotation is freely selectable. !Offset value is reset when resolution is changed!
Values	 1 - 8192

Object 6382 Preset Value

Subindex	1
Data Type	Unsigned 32
Access	ReadWrite
Default	0h
EEPROM	Yes
Description	Freely selectable position value. Preset and internal position result in offset (→ Object 63C9h)
Values	0 - 536870911

Object 6383 Position in Steps

Subindex	1
Data Type	Unsigned 32
Access	ReadOnly
Default	
EEPROM	No
Description	Current position incl. offset
Values	0 - 536870911

Object 6384 Encoder Measuring-Step Setting

Subindex	0
Data Type	Unsigned 8
Access	ReadOnly
Default	3h
EEPROM	No
Description	Largest supported Subindex
Values	3

Subindex	1
Data Type	Unsigned 16
Access	ReadOnly
Default	64h
EEPROM	Yes
Description	Encoder measuring-step setting
Values	Unit for measuring step: multiple of 10 µm

Subindex	2
Data Type	Unsigned 16
Access	ReadOnly
Default	0Ah
EEPROM	Yes
Description	Speed measuring step
Values	Unit for speed: multiple of 0.1 mm/s

Subindex	3
Data Type	Unsigned 16
Access	ReadOnly
Default	0Ah
EEPROM	Yes
Description	Acceleration measuring step
Values	Unit for acceleration: multiple of 1 mm/s^2

Object 6390 Speed Value Car

Subindex	1
Data Type	Signed 16
Access	ReadOnly
Default	
EEPROM	no
Description	Speed
Values	

Object 63C0 Operating Status

Subindex	1
Data Type	Unsigned 16
Access	ReadOnly
Default	4h
EEPROM	No
Description	Operating data written in Object 6380h
Values	Bit 0 Rotating direction = 0 → Clockwise; 1 → Counter-clockwise Bit 2 Scaling function = 0 → Max. resolution; 1 → Saved resolution

Object 63C1 Max. Resolution in Steps

Subindex	1
Data Type	Unsigned 32
Access	ReadOnly
Default	2000h = 8192 = 13-bit
EEPROM	No
Description	Maximal single-turn resolution in steps
Values	2000h = 8192 = 13-bit

Object 63C2 Max. Rotations

Subindex	1
Data Type	Unsigned 16
Access	ReadOnly
Default	10000h = 65536 = 16-bit
EEPROM	No
Description	Maximum number of rotations
Values	10000h = 65536= 16-bit

Object 63C4 Supported Warnings

Subindex	1
Data Type	Unsigned 16
Access	ReadOnly
Default	Multiturn: 14h
EEPROM	No
Description	Warnings supported by Object 63C5
Values	Multiturn: Bit 2 = CPU Watchdog status Bit 4 = Battery charge

Object 63C5 Warnings

Subindex	1
Data Type	Unsigned 16
Access	ReadOnly
Default	0h
EEPROM	No
Description	Warnings as per Object 63C4
Values	Multiturn: Bit 2 = 1 → CPU Watchdog reset Bit 4 = 1 → Battery charge is insufficient

Object 63C6 Supported Alarms

Subindex	1
Data Type	Unsigned 16
Access	ReadOnly
Default	1h
EEPROM	No
Description	Alarm messages supported by Object 63C7
Values	Bit 0 = Position error

Object 63C7 Alarms

Subindex	1
Data Type	Unsigned 16
Access	ReadOnly
Default	0h
EEPROM	No
Description	Alarm messages as per Object 63C6h
Values	Bit 0 = 1 → Position error active

Object 63C8 Operating Time

Subindex	1
Data Type	Unsigned 32
Access	ReadOnly
Default	0h
EEPROM	No
Description	Operating time in 1/10 hour since last sensor reset
Values	0..n..4294967295 = n * 6 minutes operating time without reset

Object 63C9 Offset

Subindex	1
Data Type	Unsigned 32
Access	ReadOnly
Default	0h
EEPROM	Yes
Description	Calculated from preset (→ Object 6382h)
Values	0 - current total measuring range - 1

4. Diagnosis and Important Information

4.1. Error diagnosis field bus communication

- If the encoder cannot be addressed via the CANopen bus, you should first check the connections.

  If the connections are OK, next the field bus operation should be tested. A CAN monitor is required for this purpose which records the CANopen communication and displays the telegrams.

- Now the encoder should output a BootUp message when the power supply is switched off and then on again.

  If no BootUp message appears, check whether the baud rates of the encoder, the CAN monitor and the bus system match.

- If you have difficulties establishing a connection to a node, check the node number and the baud rate.

  The same baud rate must be set everywhere. The node number (Node-ID, node address) must be between 1 and 127. Each bus node must be defined with a unique Node-ID. This means the same Node-ID may never be assigned multiple times.

  The Node-ID and baud rate can also be conveniently set via the LSS service.

4.2. Troubleshooting with field bus

The encoder has several objects and messages which describe the status or error states of the encoder:

- Object 1001h: This object is an error register for the error state of the device.
- Object 1003h: The last eight error codes and warnings are saved in this object.
- Object Emergency (80h + Node-ID): High-priority error message of a node with an Error Code and an Error Register.
- SDO Abort Message: If the SDO communication does not run correctly, the SDO response contains an Abort Code.

Object 1001h Error register
In this register the presence of a device error and its type are displayed.

Bit 0: Generic error
The remaining bits are not supported by our encoder.

Object 1003h Predefined error field
In this object the last eight Error Codes which have occurred in the objects 63C5h and 63C7h are saved, whereby the last error is entered in Subindex1 and the oldest error in Subindex8.

Object Emergency
Error message of a node.

SDO Abort Message
If the SDO communication is not carried out problem-free, an Abort Code is transmitted as the SDO response:

05040001h	: Command byte is not supported
06010000h	: Incorrect accessing of an object
06010001h	: Read access to Write Only
06010002h	: Write access to Read Only
06020000h	: Object is not supported
06090011h	: Subindex is not supported
06090030h	: Value outside limits
06090031h	: Value too large
08000000h	: General error
08000020h	: Incorrect memory signature ("save")
08000021h	: Data cannot be saved

4.3. Important information on sensor

Set new Node-ID
1. A new Node-ID is set with the IVO-specific Object 2100h.
2. After the Node-ID is set, it must be saved in the EEPROM with Object 1010h.
3. During the next initialization the sensor will report with the new Node-ID.

Set new baud rate
1. A new baud rate is set with the IVO-specific Object 2101h.
2. After the baud rate is set, it must be saved in the EEPROM with Object 1010h.
3. During the next initialization the sensor will report at the new baud rate.
4. ! REMEMBER TO SET THE MASTER TO THE NEW BAUD RATE !

Shielding
As, depending on the installation position, the encoder is not always connected to a defined ground potential, the encoder flanges should also always be connected to ground potential. The encoder should always be connected via a shielded cable.
If possible, the cable shield should be connected on both sides. It must be ensured that no transient currents are discharged via the encoder.

5. Applications

5.1. Setting and reading objects

To overwrite or read an object (SDO), two telegrams are always sent.

Set object
First the Master sends the value to be set. Then the encoder sends the confirmation.

Value (ba) is sent:

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

Confirmation:

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

Read object
First the Master sends a request for the desired object. Then the encoder sends the requested value.

Request from Master:

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

Response (dcba) of encoder to the request:

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

Commissioning
When the encoder is connected to the bus, it reports with a BootUp message. Now the encoder must be adapted to its environment and configured.

Change Node-ID and baud rate with LSS
The Node-ID and baud rate can be changed without having to address the encoder via these. With the LSS service the sensors are addressed and configured via ProductCode, RevisionNo, VendorID and SerialNumber.

Change Node-ID (Node No.)
The Node-ID can be changed in Object 2101h between 1 and 127. Then a save should be carried out with Object 1010h. During the next initialization the encoder then reports with the new Node-ID.

Change baud rate
The baud rate can be changed in Object 2100h. An index is written to the object, not the effective baud rate:

	Baud Rate
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

Now the baud rate must still be saved with Object 1010-1. During the next initialization the encoder then reports at the new baud rate. However, the baud rate of the Master should be changed.

5.2. Configuration

Set position (Preset, Referencing)
Value is sent:

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

Confirmation:

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

Change rotating direction and scaling
The rotating direction can be set to CW (clockwise) or CCW (counter-clockwise). In addition, the scaling can be activated or deactivated in the same object (6380). With the scaling activated, the set resolution is used. However, if the scaling is deactivated, the encoder operates with the maximum resolution.

Bit 0: 0 -> CW (clockwise) Value: 0
1 -> CCW (counter-clockwise) Value: 1
Bit 2: 0 -> Scaling off Value: 0
1 -> Scaling on Value: 4

Counter-clockwise and scaling activated:

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

Confirmation:

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

Change single-turn resolution
In Object 6381h the singleturn resolution can be configured. For example, 1024 (10-bit) steps per rotation (1024 = 400h):

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

Confirmation:

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

Save settings to EEPROM
With Object 1010h the saving of the objects below to the non-volatile memory (EEPROM) is triggered. To prevent accidental saving, the message "save“ must be written to Subindex 1.

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

5.3. Operation

NMT states
After the encoder has been initialized, it is in the **Pre-Operational Mode**. In this state SDOs can be read and written.

To also start PDO communication, an **NMT Start** must be sent. Then the encoder is in the **Operational Mode**. Now desired PDOs are sent. In addition, SDOs can also be read and written.

When the encoder is stopped with an **NMT Stop**, the encoder is in the **Stopped Mode**. In this state only NMT communication is possible, i.e. also the Heartbeat.

With an **NMT Reset** the encoder is reinitialized and is then in the **Pre-Operational Mode** again.

Read position
Request from the Master:

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

Response (dcba) of the encoder to the request:

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

Configure PDOs
The PDOs can be configured according to the following table:

1906h		2800h	Brief Description
Sub2	Sub5		
FFh	3ms	0	Cyclical transmission every 3 ms
FFh	5ms	2	PDO is sent double every 5 ms if a change exists.
FFh	0ms	0	Send PDO deactivated
FFh	0ms	2	Send PDO deactivated
3	xxx	0	Send during every third Sync telegram
3	xxx	2Bh	Every third Sync telegram, but total of only 43 times (=2Bh).

Specify Heartbeat Time
To monitor the communication capability, the time of the heartbeat must be defined in Object 1017h "Producer Heartbeat Time". As soon as the value has been confirmed, the service begins to transmit. For example, the encoder is to send a heartbeat every 100 ms (100 = 64h):

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

Confirmation:

COB-ID	DLC	Command	Object L	Object 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

The Heartbeat messages consist of the COB-ID and a byte. The NMT state is transmitted in this byte.

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

This means the encoder is in the Pre-Operational mode (7Fh = 127).

6. Connection Assignment and Commissioning

6.1. Mechanical attachment

Shaft encoder

- Mount the encoder housing on the flange mounting holes with three screws (square flange with 4 screws). Observe the thread diameter and thread depth.
- As an alternative the encoder can be mounted in any angled position with eccentric mountings, see Accessories.
- Connect the drive shaft and encoder shaft via a suitable coupling. The shaft ends may not touch each other. The coupling must compensate shifting caused by temperature and mechanical play. Observe the permissible axial or radial axis load. For suitable connections, see Accessories.
- Firmly tighten the mounting screws.

6.2. Electrical connection

6.2.1. Description of connections

Pin	Assignment
CAN_L	CAN Bus Signal (dominant Low)
CAN_H	CAN Bus Signal (dominant High)
UB	Supply voltage 10 - 30 VDC
GND B	Ground connection for UB
CAN_GND	Optional GND reference for CAN interface

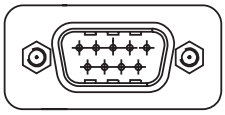
6.2.2. Connection assignment of M12 plug

Pin	Assignment
1	GND B
2	UB
3	CAN_GND
4	CAN_H
5	CAN_L



6.2.3. Connection assignment of D-SUB plug

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



6.3. Display elements (status display)

Depending on the design, a DUO-LED may be integrated on the back of the encoder.

LED green	LED red	Status
off	off	Supply Voltage Not Connected
flashing	off	Preoperational Mode
on	off	Operational Mode
on	on	Stopped/Prepared Mode
off	flashing	Warning
off	on	Error