Baumer

Passion for Sensors

Manual

Absolute Encoder with CANopen (bus cover and integrated interface)

Revision number from 1.04

Baumer Electric AG
Hummelstrasse 17
CH-8501 Frauenfeld
Phone +41 52 728 11 22
Fax +41 52 728 11 44
sales.ch@baumer.com
www.baumer.com

12.03.18 174.02.026/9
Subject to modification in technic and design.
Errors and ommissions excepted

Content

Page

1. Introduction 3
1.1. Product assignment 3
2. Safety and operating instructions 4
3. Product families 5
4. CAN bus and CANopen communication 6
4.1. CAN bus 6
4.1.1. CAN bus characteristics 6
4.2. CANopen 7
4.3. CANopen communication 8
4.3.1. Communication profile 8
4.3.2. CANopen message structure 8
4.3.3. Service data communication 9
4.3.4. Process data communication 10
4.3.5. Emergency service 12
4.3.6. Network management services 13
4.3.7. Layer Setting Services 17
4.4. Encoder profile 20
4.4.1. Overview of encoder objects 20
5. Diagnosis and useful information 25
5.1. Error diagnosis field bus communication 25
5.2. Error diagnosis via field bus 25
5.3. Useful information relating to the sensor 26
6. Applications 27
6.1. Setting and reading objects 27
6.2. Configuration 28
6.3. Operation 29
7. Terminal assignment and commissioning 31
7.1. Mechanical mounting 31
7.2. Electrical connection 31
7.2.1. Setting the user address (Node ID) 31
7.2.2. Setting the baud rate 33
7.2.3. Terminating resistor 33
7.2.4. Bus cover connection 34
7.2.5. Terminal assignment 36
7.3. Display elements (status display) 36
8. Connection and start-up with integrated fieldbus interface 37
8.1.1. Setting of node-ID and baud rate 37
8.1.2. Terminating resistor 38
8.1.3. Pin assignment D-Sub connector 38
8.1.4. Pin assignment connector M12 38

1. Introduction

1.1. Product assignment

Solid shaft encoders

Product	Product-Code	Device Name	EDS-file	Product family	Variation
BPSV 58	0x0B	BPSx	BPSxBusC.eds	Procoder – Singleturn	Bus cover
BPMV 58	0x0A	BPMx	BPMxBusC.eds	Procoder – Multiturn	Bus cover
BEMV 58	0x0A	BPMx	BPMxBusC.eds	Procoder – Multiturn	Bus cover
BMSV 42	0x17	BMSx	BMSxInt.eds	MAGRES – Singleturn	Integrated
BMMV xx	0x16	BMMx	BMMxInt.eds	MAGRES – Multiturn	Integrated
BOSV 58	0x0F	BOSx	BOSxBusC.eds	Dignalizer – Singleturn	Bus cover
BOMV 58	0x0E	BOMx	BOxBusC.eds	Dignalizer – Multiturn	Bus cover

Blind hollow shaft encoders

Product	Product-Code	Device Name	EDS-file	Product family	Variation
BPSH 58	0x0B	BPSx	BPSxBusC.eds	Procoder – Singleturn	Bus cover
BPMH 58	0x0A	BPMx	BPMxBusC.eds	Procoder – Multiturn	Bus cover
BMSH 58	0x0D	BMSx	BMSxBusC.eds	MAGRES – Singleturn	Bus cover
BMMH 58	0x0C	BMMx	BMMxBusC.eds	MAGRES – Multiturn	Bus cover
BMSHxx	0x17	BMSx	BMSxInt.eds	MAGRES – Singleturn	Integrated
BMMHxx	0x16	BMMx	BMMxInt.eds	MAGRES – Multiturn	Integrated
BOSH 58	0x0F	BOSx	BOSxBusC.eds	Dignalizer – Singleturn	Bus cover
BOMH 58	0x0E	BOMx	BOMxBusC.eds	Dignalizer – Multiturn	Bus cover
BOSH 58	0x19	BOSx	BOSxInt.eds	Dignalizer - Singleturn	Integrated
BOMH 58	0x18	BOMx	BOMxInt.eds	Dignalizer - Multiturn	Integrated
BFF	0x02	BFx	BFx13Bit.eds	BFF	Integrated
BFG	0x02	BFx	BFx13Bit.eds	BFG	Integrated

Through hollow shaft encoders

Product	Product-Code	Device Name	EDS-file	Product family	Variation
BISD 58	0x0B	BPSx	BPSxBusC.eds	Procoder – Singleturn	Bus cover
BIMD 58	0x0A	BPMx	BPMxBusC.eds	Procoder – Multiturn	Bus cover
BPSD 58	0x0B	BPSx	BPSxBusC.eds	Procoder – Singleturn	Bus cover
BPMD 58	0x0A	BPMx	BPMxBusC.eds	Procoder – Multiturn	Bus cover
BPSD 14	0x0B	BPSx	BPSxBusC.eds	Procoder – Singleturn	Bus cover
BPMD 14	0x0A	BPMx	BPMxBusC.eds	Procoder – Multiturn	Bus cover
BPSD 25	0x0B	BPSx	BPSxBusC.eds	Procoder – Singleturn	Bus cover
BPMD 25	0x0A	BPMx	BPMxBusC.eds	Procoder – Multiturn	Bus cover
BPSD 50	0x0B	BPSx	BPSxBusC.eds	Procoder – Singleturn	Bus cover
BPMD 50	0x0A	BPMx	BPMxBusC.eds	Procoder – Multiturn	Bus cover

2. Safety and operating instructions

Supplementary information
- This manual is intended as a supplement to already existing documentation (catalogues, data sheets and assembly instructions).
- The manual must be read without fail before initial commissioning of the equipment.

Intended purpose of the equipment
- The encoder is a precision measurement device. It is used to determine angular positions and revolutions, and to prepare and supply measured values in the form of electrical output signals for the follow-on device systems. The encoder may only be used for this purpose.

Commissioning
- The encoder may only be installed and assembled by suitably qualified experts.
- Observe the operating instructions of the machine manufacturer.

Safety remarks
- Prior to commissioning the equipment, check all electrical connections.
- If installation, electrical connection or any other work performed at the encoder or at the equipment is not correctly executed, this can result in a malfunction or failure of the encoder.
- Steps must be taken to exclude any risk of personal injury, damage to the plant or to the operating equipment as a result of encoder failure or malfunction by providing suitable safety precautions.
- Encoders must not be operated outside the specified limited values (see detailed product documentation).

Failure to comply with the safety remarks can result in malfunctions, personal injury or damage to property.

Transport and storage
- Only ever transport or store encoders in their original packaging.
- Never drop encoders or expose them to major vibrations.

Assembly
- Avoid impacts or shocks on the housing and shaft / hollow shaft.
- Avoid any twist or torsion on the housing.
- Do not open the encoder or make any mechanical changes to it.

The shaft, ball bearings, glass pane or electronic components can be damaged. In this case, safe and reliable operation cannot be guaranteed.

Electrical commissioning
- Do not make any electrical changes at the encoder.
- Do not carry out any wiring work when the encoder is live.
- Never plug or unplug the electrical connection when the encoder is live.
- Ensure that the entire plant is installed in line with EMC requirements. The installation environment and wiring affect the electromagnetic compatibility of the encoder. Install the encoder and supply cables separately or at a long distance from cables with high interference emissions (frequency converters, contactors etc.)
- Where working with consumers which have high interference emissions, make available a separate power supply for the encoder.
- Completely shield the encoder housing and connecting cable.
- Connect the encoder to the protective earth (PE) conductor using shielded cable. The braided shield must be connected to the cable gland or plug. Ideally, aim at bilateral connection to protective earth (PE), the housing via the mechanical assembly, the cable shield via the downstream connected devices. In case of earth loop problems, earth on one side only as a minimum requirement.

Failure to observe these instructions can result in malfunctions, material damage or personal injury.

3. Product families

The structure of the product family is modular. Depending on what is required of the encoder, the basic encoder and bus covers can be combined at will with the selected bus system. The basic encoders differ in terms of accuracy, ambient conditions and the sampling system used. In the MAGRES product family there is in addition the so-called integrated version featuring a connector output without bus cover.

Bus cover
The bus cover accommodates the entire electronic circuitry for measured value processing and for the field bus. Communication with the CAN bus takes place via the CAN controller integrated in the microcontroller. The CAN controller used has full CAN capability and supports the CAN specification 2.0B. The bus interface is standardized in accordance with ISO/DIS 11898. The maximum data rate is 1 Mbit/s.

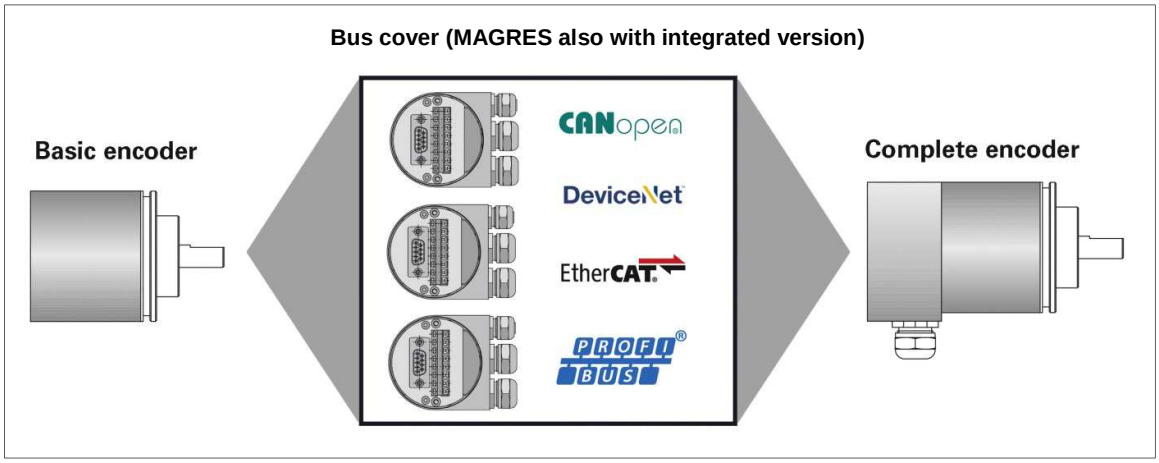
MAGRES
Has a resolution of 8192 steps per revolution with 13 bit, features a magnetic sampling system and is suitable for operation in extreme ambient conditions. The MAGRES product family comprises two designs, first the so-called integrated version with connector and mating output without bus cover and second the modular bus cover system.

Procoder
Has a resolution of 8192 steps per revolution with 13 bit, features an optical/magnetic sampling system and is suitable for standard applications.

Dignalizer
Has a resolution of 262144 steps per revolution with 18 bit, features an optical/magnetic sampling system with integrated analogue/digital conversion and is suitable for high-precision measurements.

The basic encoders are subdivided once again into a singleturn and a multiturn encoder. The multiturn encoder is capable of a resolution of up to 16 bit or 65536 revolutions, or 18 bit corresponding to 266144 revolutions (Dignalizer). The bus covers are differentiated by the respective bus interfaces. Available interfaces are CANopen, EtherCAT, DeviceNet and Profibus-DP.
All encoders can be parameterized via the bus interface.

Functional principle: MAGRES / Procoder / Dignalizer for solid shaft or hollow shaft respectively



BFF
has a resolution of 8192 steps per revolution, features a optical sampling system and is suitable for standard applications.

4. CAN bus and CANopen communication

4.1. CAN bus

The CAN bus (CAN: Controller Area Network) was originally developed by Bosch and Intel as a means of fast, low-cost data transmission in automotive applications. The CAN bus is used today also in industrial automation applications.
The CAN bus is a field bus (the standards are defined by the CAN in Automation (CiA) Association) through which devices, actuators and sensors from different manufacturers can communicate with each other.

4.1.1. CAN bus characteristics

- Data rate of 1 MBaud with network expansion up to 20 m
- Network connected on both sides
- The bus medium is a twisted-pair cable
- Real time capability: Defined maximum waiting time for high-priority messages.
- Theoretically 127 users at one bus, but physically only 32 are possible (due to the driver).
- Ensures data consistency across the network. Damaged messages are notified as faulty for all network nodes.
- Message-oriented communication
  The message is identified by a message identifier. All network nodes use the identifier to test whether the message is of relevance for them.
- Broadcasting, multicasting
  All network nodes receive each message simultaneously. Synchronization is therefore possible.
- Multimaster capability
  Each user in the field bus is able to independently transmit and receive data without being dependent upon the priority of the master. Each user is able to start its message when the bus is not occupied. When messages are sent simultaneously, the user with the highest priority prevails.
- Prioritization of messages
  The identifier defines the priority of the message. This ensures that important messages are transmitted quickly via the bus.
- Residual error probability
  Safety procedures in the network reduce the probability of an undiscovered faulty data transmission to below 10^{-11}. In practical terms, it is possible to ensure a 100% reliable transmission.
- Function monitoring
  Localization of faulty or failed stations. The CAN protocol encompasses a network node monitoring function. The function of network nodes which are faulty is restricted, or they are completely uncoupled from the network.
- Data transmission with short error recovery time
  By using several error detection mechanisms, falsified messages are detected to a high degree of probability.
  If an error is detected, the message transmission is automatically repeated.

In the CAN Bus, several network users are connected by means of a bus cable. Each network user is able to transmit and receive messages. The data between network users is serially transmitted.

Examples of network users for CAN bus devices are:
- Automation devices such as PLCs
- PCs
- Input and output modules
- Drive control systems
- Analysis devices, such as a CAN monitor
- Control and input devices as Human Machine Interfaces (HMI)
- Sensors and actuators

4.2. CANopen

Under the technical management of the Steinbeis Transfer Centre for Automation, the CANopen profile was developed on the basis of the Layer 7 specification CAL (CAN Application Layer). In comparison with CAL, CANopen only contains the functions suitable for this application. CANopen thus represents only a partial function of CAL optimized for the application in hand, so permitting a simplified system structure and the use of simplified devices. CANopen is optimized for fast data exchange in real time systems.
The organization CAN in Automation (CiA) is responsible for the applicable standards of the relevant profiles.
CANopen permits:
• Simplified access to all device and communication parameters
• Synchronization of several devices
• Automatic configuration of the network
• Cyclical and event-controlled process data communication

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

All device and communication parameters are subdivided into an object directory. An object directory encompasses the name of the object, data type, number of subindexes, structure of the parameters and the address. According to CiA, this object directory is subdivided into three different parts. Communication profile, device profile and a manufacturer-specific profile (see object directory).

4.3. CANopen communication

4.3.1. Communication profile

Communication between the network users and the Master (PC / Control) takes place by means of object directories and objects. The objects are addressed via a 16 bit index. The CANopen communication profile DS 301 standardizes the various communication objects. They are accordingly divided into several groups:

- Process data objects PDO for real time transmission of process data
- Service data objects SDO for read/write access to the object directory
- Objects for synchronization and error display of CAN users:
  - SYNC object (synchronization object) for synchronization of network users
  - EMCY object (emergency object) for error display of a device or its peripherals
- Network management NMT for initialization and network control
- Layer Setting Services LSS for configuration by means of serial numbers, revision numbers etc. in the middle of an existing network

4.3.2. 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 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)]	180h + Node ID
PDO2 (tx)[1)]	280h + Node ID
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 viewpoint of the encoder

The node ID can be freely selected by means of the CANopen bus between 1 and 127 (if encoder = 0). The encoders are supplied with the Node ID 1.
This can be changed with the service data object 2101h or using LSS.

A CAN telegram is made up 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 precise telegram is outlined in more detail at a later point.

4.3.3. Service data communication

The service data objects correspond to the standards of the CiA. It is possible to access an object via index and subindex. The data can be requested or where applicable written into the object.

General information on the SDO
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** is composed as follows:
Master -> Encoder : 600h + Node ID
Encoder -> Master : 580h + Node ID

DLC (data length code) describes the length of the telegram. This is composed as follows:
1 byte command + 2 bytes object + 1 byte subindex + no. of data bytes (0 - 4).

The **command byte** defines whether data is read or set, and how many data bytes are involved.

SDO command	Description	Data length	
22h	Download request	Max. 4 Byte	Transmits parameter to encoder
23h	Download request	4 byte	
2Bh	Download request	2 byte	
2Fh	Download request	1 byte	
60h	Download response	-	Confirms receipt to master
40h	Upload request	-	Requests parameter from encoder
42h	Upload response	Max. 4 byte	Parameter to master with max. 4 byte
43h	Upload response	4 byte	
4Bh	Upload response	2 byte	
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 subindex are those of the requested object. The error code is contained in bytes 8…5.

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 the limit
06090031h : Value too great
08000000h : General error
08000020h : Incorrect save signature
08000021h : Data cannot be stored

SDO examples

Request of a value by the master from the slave
A frequent request will be a request for position. → Object 6004h

COB ID	DLC	Command	Object L	Object H	Subindex	Data 0	Data 1	Data 2	Data 3
600h+node ID	8	40h	04h	60h	0	x	x	x	x

Response by the slave **to the request** for a value
The position is 4 bytes long, the precise values can be found under object 6004h.

COB ID	DLC	Command	Object L	Object H	Subindex	Data 0	Data 1	Data 2	Data 3
580h+node ID	8	43h	04h	60h	0	a	b	c	d

Writing of a value by the master into the slave
Position setting can be performed with preset. → Object 6003h

COB ID	DLC	Command	Object L	Object H	Subindex	Data 0	Data 1	Data 2	Data 3
600h+node ID	8	22h	03h	60h	0	a	b	c	d

Slave's **response** 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	03h	60h	0	0	0	0	0

4.3.4. Process data communication

Process data objects are used for real time data exchange for process data, for example position or operating status. PDOs can be transmitted synchronously or cyclically (asynchronously). The encoder supports the PDO1 and the PDO2. Both PDOs supply the current position of the encoder and are defined in the objects 1800h, 1801h, 1A00h, 1A01, 2800h, 2801h and 6200h.

Synchronous
In order to transmit the process data synchronously, a value between 1 and F0h (=240) must be written into the object 1800h / 1801h Subindex 2. If the value is 3, the PDO is transmitted on every third sync telegram (if the value 1 is entered, transmission takes place on every sync telegram), as long as there is a 0 written into the object 2800h / 2801h. If it contains for example a 5, the PDO will continue to be written as before on every third Sync telegram, but only a total of 5 times. Accordingly, the last PDO is written on the 15th sync telegram. The counter for the number of PDOs to be transmitted is reset in the event of a position change or NMT reset, i.e. unless it is changed, the position is transmitted five times. If the position changes, it is transmitted a further five times.
In synchronous operation, the PDO is requested by the master via the Sync telegram.

Byte 0	Byte 1
COB ID = 80	0

Cyclical (asynchronous)
If you wish the PDOs to be transmitted cyclically, the value FEh must be written into the object 1800h / 1801h 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 1 ms. If the value is stored for 0 ms, the PDOs are not transmitted. The function is switched off.
The object 2800h / 2801h offers another possibility: If the value is 0, cyclical transmission runs as described above. If the value is 1, a cyclical test is performed as to whether a change of the value has occurred. If not, no transmission takes place. If the value is 4, the PDO is transmitted four times with each cycle if there is a change.

Overview
In the following table, the different transmission modes for PDOs are summarized:

1800h		2800h	Summarized description
Sub2	**Sub5**		
FEh	3ms	0	Cyclical transmission every 3ms
FEh	5ms	2	Every 5ms, the PDO is sent twice if there is a change
FEh	0ms	xxx	Transmit PDO switched off
3	xxx	0	Transmit with every third sync telegram
3	xxx	2Bh	On every third sync telegram, but only 43 times in total (=2Bh).

PDO (Position)
PDO1 telegram structure:

ID	DLC	Byte 1	Byte 2	Byte 3	Byte 4
181h	4	Xx	Xx	Xx	Xx

ID : 180h + node ID
Length : 4 Data Byte
Byte1 - 4 : Current position in increments

PDO2 telegram structure:

ID	DLC	Byte 1	Byte 2	Byte 3	Byte 4
281h	4	Xx	Xx	Xx	Xx

ID : 280h + node ID
Length : 4 Data Byte
Byte1 - 4 : Current position in increments

4.3.5. Emergency service

Internal device error or bus problems initiate 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 6503h		Warning 6505h		n.u.
		00h	01h	1001h					

Byte 0..1: Error codes

Error Code (hex)	Meaning
0000	Error reset or no error
1000	Generic error
5530	EEPROM error
6010	Software reset (Watchdog)
7320	Position error
7510	Internal communication error
8130	Life Guard error or Heartbeat error
FF00	Battery charge to low

Byte 2: Error register

Bit	Meaning
0	Generic Error
4	Communication error
7	manufacturer specific

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 executed
4	Battery charge	OK	Charge too deep

Byte 7: Not used

4.3.6. Network management services

Network management can be divided into two groups.
Using the NMT services for **device monitoring**, bus users can be initialized, started and stopped.
In addition, NMT services exist for **connection monitoring**.

Description of the NMT command
The commands are transmitted as unconfirmed objects and are structured as follows:

Byte 0	Byte 1	Byte 2
COB ID = 0	Command byte	Node number

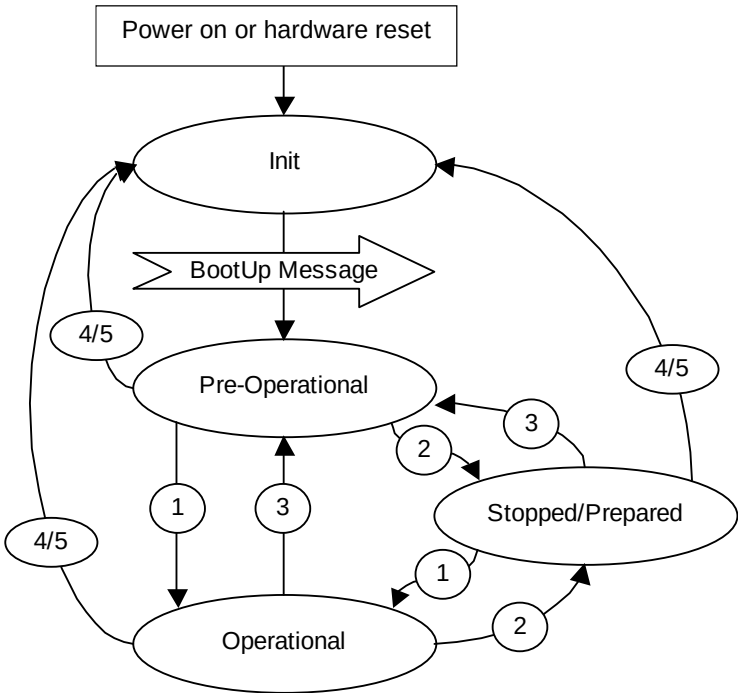
The **COB ID** for NMT commands is always zero. The node ID is transmitted in byte 2 of the NMT command.

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 required users. With node number = 0, all users are addressed.

NMT state event
Following initialization, the encoder is in the pre-operational mode. In this status, SDO parameters can be read and written. In order to request PDO parameters, the encoder must first be moved to the operational mode status.



The various NMT states

Init
Following initalization, the encoder logs on to the CAN bus with a BootUp message. The encoder then goes automatically to the pre-operational mode status.

The COB ID of the BootUp message is made up of 700h and 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 transmits the requested 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 set. LSS is only possible in the stopped mode.

Status change

Start remote node (1)
With the start command, the encoder is switched to the operational mode status.

COB ID	Command byte	Node number
0	1h	0..127

Stop remote node (2)
With the stop command, the encoder is switched to the stopped or prepared mode status.

COB ID	Command byte	Node number
0	2h	0..127

Enter pre-operational mode (3)
Change to the pre-operational mode status.

COB ID	Command byte	Node number
0	80h	0..127

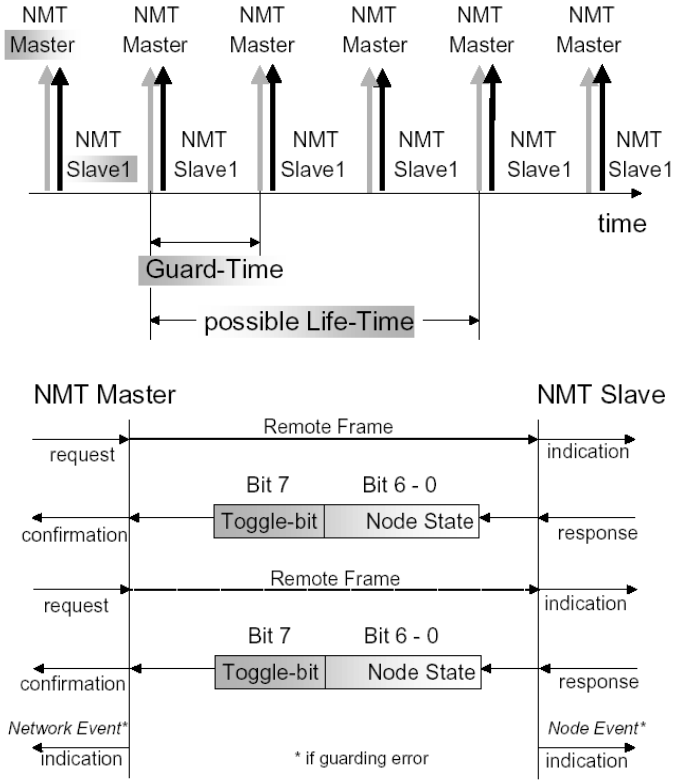
Reset remote node (4) or reset communication (5)
With the reset command, the encoder is re-initialized.
Reset remote node (4):

COB ID	Command byte	Node number
0	81h	0..127

Reset communication (5):

COB ID	Command byte	Node number
0	82h	0..127

Node and Life Guarding



"Communication error Object 1029h-1h".

The „CAN in Automation“ association CiA recommend to use the new heartbeat protocol (see next chapter).
To use the node guarding instead of heartbeat protocol bit 5 of object 2110h has to be set.

To detect absent devices (e.g. because of bus-off) that do not transmit PDOs regularly, the NMT Master can manage a database, where besides other information the expected states of all connected devices are recorded, which is known as Node Guarding. With cyclic node guarding the NMT master regularly polls its NMT slaves. To detect the absence of the NMT master, the slaves test internally, whether the Node Guarding is taking place in the defined time interval (Life Guarding). The Node Guarding is initiated by the NMT Master in Pre-Operational state of the slave by transmitting a Remote Frame.
The NMT Master regularly retrieves the actual states of all devices on the network by a Remote Frame and compares them to the states recorded in the network database. Mismatches are indicated first locally on the NMT Master through the Network Event Service. Consequently the application must take appropriate actions to ensure that all devices on the bus will got to a save state

Example for a nodeguarding protocol:

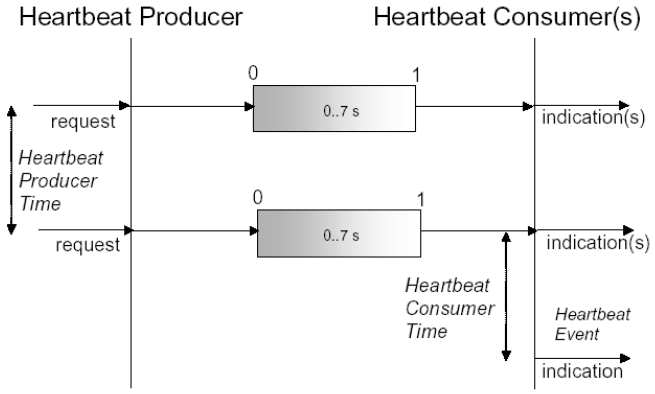
COB-ID	Data/ Remote	Byte 0
701h	r	00h (0d)
701h	d	FFh (255d)
701h	r	00h (0d)
701h	d	7Fh (127d)

Possible NMT node states:

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

in other words, the encoder is in the pre-operational mode (7Fh = 127).

Heartbeat protocol



The optional heartbeat protocol should substitute the life/node guarding protocol. Heartbeat ist aktiv, wenn im Objekt 2110h Bit5 auf '0' ist. It is highly recommend to implement for new device designs the heartbeat protocol. A Heartbeat Producer transmits the Heartbeat message cyclically with the frequency defined in Heartbeat producer time object. One or more Heartbeat Consumer may receive the indication. The relationship between producer and consumer is configurable via Object Dictionary entries. The Heartbeat Consumer guards the reception of the Heartbeat within the Heartbeat consumer time. If the Heartbeat is not received within this time a Heartbeat Event will be generated "Communication error object 1029h-1h".

Example for a heartbeat protocol

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

The heartbeat messages consist of the COB ID and one byte. In this byte, the NMT status is supplied.

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

in other words, the encoder is in the pre-operational mode (7Fh = 127).

4.3.7. Layer Setting Services

In the spring of 2000, CiA drafted a new protocol intended to ensure standardized occurrence. The procedure is described under
Layer Setting Services and Protocol, CiA Draft Standard Proposal 305 (LSS).
The encoder is supplied by us as standard with the node ID 1 and a baud rate of 50 kBaud. Several encoders can be connected to the bus system with the same node ID. To allow individual encoders to be addressed, LSS is used.
Each encoder is fitted with its own unique serial number and is addressed using this number. In other words, an optional number of encoders with the same node ID can be connected to one bus system, and then initialized via LSS. Both the node ID and also the baud rate can be reset. LSS can only be executed 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.
This is followed by up to seven attached data bytes.

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

Selective switch mode
The following procedure can be used to address a certain encoder in the bus system.

7E5h →	40h	Vendor ID	reserved

7E5h →	41h	Product code	reserved

7E5h →	42h	Revision number	reserved

7E5h →	43h	Serial number	reserved

7E4h ←	44h	Mode	reserved

Vendor Id : 5Fh
Product code : Internal product code for the respective encoder
Revision number : Current revision number of the encoder
Serial number : Unique, consecutive serial number
Mode : The encoder's response is the new mode (0=operating mode; 1=configuration mode)

Setting the node ID

7E5h →	17	Node ID	reserved

7E4h ←	11h	ErrCode	Spec error	reserved

Node ID : The encoder's new node ID
Error code : 0=OK; 1=Node ID outside range; 2..254=reserved; 255→Specific error
Specific error : If Error code=255 → application-specific error code.

Setting the bit timing

7E5h →	13h	tableSel	tableInd	reserved

7E4h ←	13h	ErrCode	SpecError	reserved

TableSel : Selects the bit timing table
0 : Standard CiA bit timing table
1..127 : Reserved for CiA
128..255 : Manufacturer-specific tables
TableInd : Bit timing entry in selected table (see table below).
Error code : 0=OK; 1=Bit timing outside range; 2..254=reserved; 255→Specific error
Specific error : If Error code=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 the configuration protocol
This protocol saves the configuration parameters in the EEPROM.

7E5h →	17h	reserved

7E4h ←	17h	ErrCode	SpecError	Reserved

Error code : 0=OK;1=Saving not supported;2=Access error;3..254=reserved;255→Specific error
Specific error : If error code=255 → Application-specific error code.

Activate bit timing parameters
The new bit timing parameters are activated with the command specifier 21.

7E5h →	15h	16 bit Switch delay	Reserved

Switch Delay : Reset delay in the slave in ms.
After the delay, the encoder logs on with the new baud rate.

Request vendor ID
Requesting the vendor ID of a selected encoder

7E5h →	5Ah	reserved

7E4h ←	5Ah	32 bit vendor ID	reserved

Vendor ID : = 5Fh

Request product code
Request product code of a selected encoder

7E5h →	5Bh	reserved	

7E4h ←	5Bh	Product code	reserved

Product code : Manufacturer-dependent product code

Request revision number
Request revision number of a selected encoder

7E5h →	5Ch	reserved	

7E4h ←	5Ch	32 bit revision number	reserved

Revision number : Current revision

Request serial number
Request serial number of a selected encoder

7E5h →	5Dh	reserved	

7E4h ←	5Dh	32 bit serial number	reserved

Serial number : Unique consecutive serial number of the encoder

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

7E5h →	46h	Vendor ID	reserved

7E5h →	47h	Product code	reserved

7E5h →	48h	Revision number LOW	reserved
7E5h →	49h	Revision number HIGH	reserved

7E5h →	4Ah	Serial number LOW	reserved
7E5h →	4Bh	Serial number HIGH	reserved

Each encoder with the relevant parameters logs on with the following message:

7E4h ←	4Fh	reserved	

4.4. Encoder profile

4.4.1. Overview of encoder objects

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

- **Standard objects**:
  1000h, 1001h, 1018h
- **Manufacturer-specific objects**:
  2000h - 5FFFh
- **Device-specific objects**:
  All other objects from 1000h - 1FFFh, 6000h - FFFFh

The following table provides a summary of all SDO objects supported by the encoder.

Object Object number in Hex
Name **---**
Format U/I = Unsigned/Integer , No. = no of bits, ARR = Array, REC = Record
Access ro = read only, wo = write only, rw = read write
Default Default value on first init
Save yes = is stored in the EEPROM
Description Additional info

Object Subindex	Name	Format	Access	Default	Save	Description
1000h	Device type	U32	ro	00020196h 00010196h		Multiturn encoder: Byte 0..1: Profile no=196h=406 Byte 2..3: Encoder type =2 (Multiturn, absolute) Singleturn encoder: Byte 0..1: Profile no=196h=406 Byte 2..3: Encoder type =1 (Singleturn, absolute)
1001h	Error register	U8	ro	0h		Bit0 = Generic error Bit4 = Communication error (overrun, ...) Bit7 = Manufacturer specific error
1003h	Predefined error field	ARR				Contains the last 8 errors or warnings
00h	Biggest subindex	U8	rw	0h		Number of stored messages (0 - 8)
01h	Last entry	U32	ro			Last error or warning 1000h Generic error 5530h EEProm error 6010h Software reset (watchdog) 7320h Position error 7510h Internal communication error 8130h Life Guard error or Heartbeat error FF00h Battery charge to low
..	..	..	..	..		..
08h	Oldest entry	U32	ro			Error or warning, see Subindex 01h
1005h	Sync COB ID	U32	rw	80h	yes	COB ID of the sync object
1008h	Device name	U32	ro	”BFx” "BPMx" "BPSx" "BMMx“ "BMSx" "BOMx" "BOSx"	yes	Device name = “BFx” 13Bit Singleturn "BPMx" Procoder Multiturn "BPSx" Procoder Singleturn "BMMx“ MAGRES Mulitturn "BMSx“ MAGRES Singleturn "BOMx" Dignalizer Multiturn "BOSx" Dignalizer Singleturn
1009h	Hardware version	U32	ro	factory		Hardware version in ASCII
100Ah	Software version	U32	ro	factory		Software version in ASCII
100Ch	Guard time	U16	rw	0h	yes	Timer for nodeguarding
100Dh	Life time factor	U8	rw	0h	yes	Multiplication of Guard time
1010h	Store parameters	ARR				
00h	Biggest subindex	U8	ro	4h		No. of save possibilities 4

Object Subindex	Name	Format	Access	Default	Save	Description
01h	Save all parameters	U32	rw			=“evas “ (0x65766173) to save
02h	Communication parameters	U32	rw			=“evas “ (0x65766173) to save
03h	Application parameters	U32	rw			=“evas “ (0x65766173) to save
04h	Manuf. specific parameters	U32	rw			=“evas “ (0x65766173) to save
1011h	Restore default parameters	ARR				
00h	Biggest subindex	U8	ro	4h		No. of reset possibilities = 4
01h	All parameters	U32	rw			=“daol“ (0x64616F6C) to load
02h	Communication parameters	U32	rw			=“daol“ (0x64616F6C) to load
03h	Application parameters	U32	rw			=“daol“ (0x64616F6C) to load
04h	Manufacturer specific parameters	U32	rw			=“daol“ (0x64616F6C) to load
1014h	Emergency COB ID	U32	rw	80h + Node ID	yes	COB ID of the emergency object
1016h	Consumer heartbeat	ARR				
00h	Biggest subindex	U8	ro	1h		
01h	Consumer heartbeat time	U32	rw	10000h	yes	Bit0..15 Consumer heartbeat time in ms Bit16..23 Node-ID
1017h	Producer heartbeat time	U16	rw	0h	yes	Producer heartbeat time in ms
1018h	Identity object	REC	ro			
00h	Biggest subindex	U8	ro	4h		
01h	Vendor ID	U32	ro	5Fh	yes	Vendor no. issued by CiA
02h	Product code	U32	ro	02h 0Ah 0Bh 0Ch 0Dh 0Eh 0Fh 16h 17h 18h 19h	1	Product Code: 02h = BFx 0Ah = Procoder multiturn bus cover 0Bh = Procoder singleturn bus cover 0Ch = MAGRES multiturn bus cover 0Dh = MAGRES singleturn bus cover 0Eh = Dignalizer multiturn bus cover 0Fh = Dignalizer singleturn bus cover 16h = MAGRES multiturn integrated 17h = MAGRES singleturn integrated 18h = Dignalizer multiturn integrated 19h = Dignalizer singleturn integrated
03h	Revision number	U32	ro	factory		Current revision number
04h	Serial number	U32	ro	factory	yes	Unique consecutive serial number
1029h	Error behaviour	ARR				
00h	Biggest subindex	U8	ro	1h		
01h	Communication error	U8	rw	1h	yes	0h = Change to pre-operational mode 1h = No state change 2h = Change to stopped mode 3h = Reset remote node
1800h	Transmit PDO1 parameter	REC				
00h	Biggest subindex	U8	ro	5h		
01h	COB ID	U32	rw	180h+id	yes	PDO ID = 180h + node ID
02h	PDO type	U8	rw	FEh	yes	FEh=User defined, cyclical
05h	Event timer	U16	rw	203h	yes	Cycle time in ms
1801h	Transmit PDO2 parameter	REC				
00h	Biggest subindex	U8	ro	5h		
01h	COB ID	U32	rw	280h+id	yes	PDO ID = 280h + Node ID
02h	PDO type	U8	rw	2h	yes	2h= synchronous operation
05h	Event timer	U16	rw	100h	yes	Cycle time in ms
1A00h	Transmit PDO1 mapping	ARR				
00h	Biggest subindex	U8	ro	1h		

Object Subindex	Name	Format	Access	Default	Save	Description
01h	Content of PDO1	U32	ro	6004h		Read only, although from CiA as read write
1A01h	Transmit PDO2 mapping	ARR				
00h	Biggest subindex	U8	ro	1h		
01h	Content of PDO2	U32	ro	6004h		Read only, although from CiA as read write
2100h	Baud rate	U8	rw	2h	yes	After setting the baud rate, the 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	1h	yes	Node number 1..127 possible After setting the baud rate, the EEPROM must be saved and reinitialized.
2110h	Versions control	U32	rw	8h	yes	Bit1 = The sense of rotation (object 6000h Bit0) 0 Not inverted 1 Inverted Bit2 = Scaling function 0 Enabled 1 Disabled Bit3 = 0 Not reseted after busOFF 1 Reseted after busOFF Bit5 = 0 Heartbeat protocol active 1 Node guarding protocol active (only active after restart of bus) Bit6 = 0 The encoder transmitted the actual position value after a SYNC telegram 1 The encoder transmitted a new evaluated position value after a SYNC telegram Bit7 = For SYNC minimal jitter of position 0 Cyclic position evaluation 1 The position will be evaluated after a SYNC telegram (Bit6 must be also activated) → minimal jitter of position Bit8 = PDO1 timing delay 2ms 0 1800h-5h = 6200h 1 1800h-5h = 6200h + 2ms Bit9 = Responce by write to object Resolution/overall resolution 0 Offset reset 1 Offset not reset (Version from V1.08)
2201h	Statistics	REC				
00h	Biggest subindex	U8	ro	3h		No. of subindexes
01h	No. of position errors	U32	ro		yes	Position control
02h	Time in seconds	U32	ro		yes	Time since last reset
03h	Number timer reset watchdog	U32	ro		yes	Timer watchdog
2300h	Customer EEPROM range	ARR				Optional data can be stored in this object
00h	Biggest subindex	U8	ro	7h		
01h	Data0	U16	rw	0h	yes	
02h	Data1	U16	rw	0h	yes	
03h	Data2	U16	rw	0h	yes	
04h	Data3	U16	rw	0h	yes	
05h	Data4	U16	rw	0h	yes	
06h	Data5	U16	rw	0h	yes	

Object Subindex	Name	Format	Access	Default	Save	Description
07h	Data6	U16	rw	0h	yes	
2800h	PDO1 addition event trigger	U8	rw	0h	yes	Repeat counter for PDO1
2801h	PDO2 addition event trigger	U8	rw	0h	yes	Repeat counter for PDO2
6000h	Operating parameter	U16	rw	4h	yes	Bit0 = look to the shaft, Position increasing 0 CW 1 CCW Bit2 = 0 Scaling function disabled 1 Scaling function enabled
6001h	Resolution	U32	rw	 2000h 1000h 40000h	yes	Resolution in steps / revolution: 13Bit = Procoder, BFx 12Bit = MAGRES 18bit = Dignalizer
6002h	Overall measuring range in increments	U32	rw	 20000000h 2000h 40000000h 1000h (1)00000000h 40000h	yes	Overall measuring range in increments 29Bit = Procoder multiturn 13Bit = Procoder singleturn, BFx 30Bit = MAGRES multiturn 12Bit = MAGRES singleturn 32Bit = Dignalizer multiturn 18Bit = Dignalizer singleturn
6003h	Preset value in increments	U32	rw	0h	yes	Preset in increments → Offset
6004h	Position in increments	U32	ro			Position value including offset in increments
6200h	Cyclic timer for PDO1	U16	rw	203h	yes	In ms, identical object 1800h, subindex 5
6500h	Operating status	U16	ro	4h		Bit0 = look to the shaft, Position increasing 0 CW 1 CCW Bit2 = 0 Scaling function disabled 1 Scaling function enabled
6501h	Max. resolution	U32	ro	 2000h 4000h 40000h		Max. resolution in steps / revolution: 13Bit = Procoder, BFx 14Bit = MAGRES 18Bit = Dignalizer
6502h	Overall measuring range in increments	U32	ro	 20000000h 2000h (1)00000000h 4000h (1)00000000h 40000h		(is outside the specification of CiA) Overall measuring range in increments: 29Bit = Procoder multiturn 13Bit = Procoder singleturn, BFx 32Bit = MAGRES multiturn 14Bit = MAGRES singleturn 32Bit = Dignalizer multiturn 18Bit = Dignalizer singleturn
6503h	Alarms	U16	ro	0h		The following alarms are evaluated: Bit0 = Position error
6504h	Supported alarms	U16	ro	1h		The following alarms are supported: Bit0 = Position error
6505h	Warnings	U16	ro	0h		The following warnings are evaluated: Multiturn encoder: Bit2 = CPU watchdog status Bit4 = Battery charge Singleturn encoder: Bit2 = CPU watchdog status
6506h	Supported warnings	U16	ro	 14h 04h		The following warnings are supported: Multiturn encoder: Bit2 = CPU watchdog status Bit4 = Battery charge Singleturn encoder: Bit2 = CPU watchdog status
6507h	Profile & software version	U32	ro	factory		Byte 0..1: Profile version =2.01 = 0201h Byte 2..3: Software version = 1.05 = 0105h
6508h	Operating time	U32	ro	0h		Time in 1/10 hours since last reset

Object Subindex	Name	Format	Access	Default	Save	Description
6509h	Offset	U32	ro	0h		Offset calculated from preset → 6003h
650Bh	Serial number	U32	ro	factory	yes	Linked with serial number object 1018-4

5. Diagnosis and useful information

5.1. Error diagnosis field bus communication

- If the encoder cannot be addressed via the CANopen bus, first of all check the terminals.

  If the terminals are not in order, field bus operation should be tested next. For this purpose, a CAN monitor is required which records CANopen communication and shows the telegrams.

- The encoder should now place a BootUp message when switching the power supply off and on again.

  Should no BootUp message appear, check whether the baud rates of the drive system, the CAN monitor and the bus system are in agreement.

- If you have difficulty in establishing the connection to the user, check the node number and baud rate.

  The baud rate must be set the same throughout. The node number (node ID, node address) must be between 1 and 127. Each bus user must be unambiguously assigned a node ID, i.e. it is strictly prohibited to assign the same node ID more than once.

  The node ID and baud rate can also be set conveniently using the LSS service.

5.2. Error diagnosis via field bus

The encoder has at its disposal several objects and messages which transcribe the status or error status of the encoder.

- Object 1001h: This object is an error register for the device error status.
- Object 1003h: In this object, the last eight error codes and warnings are stored.
- Object Emergency (80h + Node ID): High-priority error message of a user with error code and error register.
- SDO abort message: If SDO communication does not run correctly, the SDO response contains an abort code.

Object 1001h error register
The existence of a device error and its type are indicated in this register.

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

Object 1003h predefined error field
In this object, the eight last occurring error codes from objects 6503h and 6505h are saved, whereby the latest error is stored in subindex 1 and the oldest error in subindex 8.

Object emergency
Error message of a user.

SDO abort message
If SDO communication is not running smoothly, an abort code is transmitted as the SDO response:

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 great
08000000h	: General error
08000020h	: Incorrect save signature ("save")
08000021h	: Data cannot be saved

Service address

Baumer Electric AG
Postfach, Hummelstrasse 17
CH-8501 Frauenfeld
Service-Tel.-No.: +41(0)52 728 11 22 (Mo.-Fr. 8.00-17.00 h)
sales.ch@baumer.com
www.baumer.com

5.3. Useful information relating to the sensor

Resetting the node ID
1. The node ID is reset using the Baumer-specific object 2100h.
2. After setting the node ID, this must be saved in the EEPROM with object 1010h.
3. On next initialization, the sensor logs on with the new node ID.

Resetting the baud rate
1. The baud rate is reset with the Baumer-specific object 2101h.
2. After setting the baud rate this must be saved in the EEPROM with object 1010h.
3. On next initialization, the sensor logs on with the new baud rate.
4. ! DO NOT FORGET TO SET THE MASTER TO THE NEW BAUD RATE !

Shielding
As the encoder is not always connected to a defined earth potential depending on its mounting position, the encoder flange should always be additionally linked to earth potential. The encoder should always on principle be connected to a shielded conductor.
If possible the cable shield should be in place at both ends. Ensure that no equalizing currents are discharged via the encoder.

6. Applications

6.1. Setting and reading objects

In order to overwrite an object (SDO) or to read it, two telegrams always have to be transmitted.

Object setting
First, the master transmits the value to be set. The encoder then transmits the confirmation.

Value (ba) is transmitted:

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 transmits a request for the required object. Then the encoder transmits 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	04h	60h	0h	x	x	x	x

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	04h	60h	0h	a	b	c	d

Commissioning
When the encoder is connected to the bus, it logs on with a BootUp message. The encoder must now be adjusted to its environment and configured.

Changing the node ID and baud rate with LSS
The node ID and baud rate can be changed without having to use these to address the encoder. With the LSS service, the sensors are addressed and configured via the product code, revision no., vendor ID and serial number.

Changing the node ID (node no.)
The node ID can be changed in object 2101h between 1 and 127. A save routine should then be executed using object 1010h. On the next initialization, the encoder logs on with the new node ID.

Changing the baud rate
The baud rate can be changed in the object 2100h. An index is written into 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

The baud rate now still has to be saved using object 1010-1. On next initialization, the encoder logs on to the new baud rate. However, before this the baud rate of the master should be changed.

6.2. Configuration

Position setting
The value is transmitted:

COB ID	DLC	Command	Object L	Object H	Subindex	Data 0	Data 1	Data 2	Data 3
600h+node ID	8	23h	03h	60h	0h	a	b	c	d

Conformation:

COB ID	DLC	Command	Object L	Object H	Subindex	Data 0	Data 1	Data 2	Data 3
580h+node ID	8	60h	03h	60h	0h	0	0	0	0

Changing the sense of rotation and scaling
The sense of rotation can be set to CW (clockwise) or CCW (counterclockwise). In addition, the scaling can be switched on or off in the same object (6000h). With the scaling switched on, the set resolutions are used. However, if the scaling is switched off, the encoder works with the maximum resolution settings (6501h and 6502h).

Bit 0: 0 -> CW (clockwise)
1 -> CCW (counterclockwise)
Bit 2: 0 -> Scaling off
1 -> Scaling on

Counterclockwise rotation and scaling on:

COB ID	DLC	Command	Object L	Object H	Subindex	Data 0	Data 1	Data 2	Data 3
600h+node ID	8	23h	00h	60h	0h	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	00h	60h	0h	0	0	0	0

Changing singleturn resolution
In object 6001h, the singleturn resolution can be configured. For example 1024 (10bit) steps per revolution (1024 = 400h):

COB ID	DLC	Command	Object L	Object H	Subindex	Data 0	Data 1	Data 2	Data 3
600h+node ID	8	23h	01h	60h	0h	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	01h	60h	0h	0	0	0	0

Changing the overall resolution
In object 6002h, the overall resolution can be set. The overall resolution and the singleturn resolution result in the number of revolutions. Example: The singleturn resolution is set at 10 bit (1024 steps) and the overall resolution at 22 bit (4194304), resulting in 4096 (12bit) revolutions of 1024 (10bit) steps each.

Setting the overall resolution to 4194304 (4194304 = 400000h)

COB ID	DLC	Command	Object L	Object H	Subindex	Data 0	Data 1	Data 2	Data 3
600h+node ID	8	23h	02h	60h	0h	00	00	40	00

Confirmation:

COB ID	DLC	Command	Object L	Object H	Subindex	Data 0	Data 1	Data 2	Data 3
580h+node ID	8	60h	02h	60h	0h	0	0	0	0

Saving the setting in the EEPROM
Object 1010h initiates the save routine for the objects below in the non-volatile memory (EEPROM). In order to prevent unintentional saving, the message "Save" must be written in 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'

Confirmation:

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

6.3. Operation

NMT statuses
Once the encoder has been initialized, it is then in the **Pre-operational mode**. In this mode, SDO can be read and written.

In order to start PDO communication, you must transmit an **NMT start**. The encoder is then in the **Operational mode**. Any required PDOs are then transmitted. SDOs can also be read and written.

If the encoder is stopped with an **NMT stop**, the encoder is then in the **stopped mode**. In this mode, only NMT communication is the possible, i.e. also heartbeat.

By means of an **NMT reset** the encoder is re-initialized and is then once again in the **pre-operational mode**.

Reading the 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	04h	60h	0	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	04h	60h	0	a	b	c	d

Configuring PDOs
The PDOs can be configured in accordance with the following table:

1800h		2800h	Summarized description
Sub2	Sub5		
FEh	3ms	0	Cyclical transmission every 3 ms
FEh	5ms	2	Every 5ms the PDO is sent double if a change has occurred.
FEh	0ms	xxx	Transmit PDO switched off
3	xxx	0	Transmit with each third sync telegram
3	xxx	2Bh	With each sync telegram but in total only 43 times (=2Bh).

Defining heartbeat time
In order to monitor communication capability, the heartbeat time must be defined in object 1017h with "Producer heartbeat time". As soon as the value has been confirmed, the service begins transmission.
Example: Every 100ms, the encoder should transmit a heartbeat (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 are made up of the COB ID and one byte. IN this byte, the NMT status is supplied.

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

i.e. the encoder is in the pre-operational modus (7Fh = 127).

7. Terminal assignment and commissioning

7.1. Mechanical mounting

Shaft encoder

- Mount the encoder housing using the fastening holes on the flange side with three screws (square flange with four screws), paying attention to the thread diameter and thread depth.
- Alternatively, the encoder can be mounted in any angular position using three eccentric fastenings - see accessories.
- Connect the drive shaft and encoder shaft using a suitable coupling. The ends of the shafts must not be touching. The coupling must be capable of compensating for displacement due to temperature and mechanical backlash. Pay attention to the admissible axial or radial shaft loads. For suitable connecting devices, see under accessories.
- Tighten the fastening screws.

Hollow shaft / end shaft encoder

- Clamping ring fixture
  Place the encoder on the drive shaft and tighten the clamping ring.
- Encoder torque pin
  Slide encoder onto the drive shaft and insert torque pin into the adjusting element provided by customer.
- Adjusting element with rubberized spring element
  Push the encoder on to the drive shaft and insert the parallel pin into the mounted adjusting element (not supplied) (with rubberized spring element)
- Adjusting bracket
  Push the encoder over the drive shaft. Insert the adjusting bracket into the rubberized spring element of the encoder and fasten the adjusting bracket on the contact surface (not supplied).
- Shoulder screw
  Push the encoder over the drive shaft and insert the shoulder screw (not supplied) in the rubberized spring element of the encoder.
- Coupling spring
  Mount the coupling spring with screws onto the fixing holes of the encoder housing.
  Push the encoder over the drive shaft and fasten the coupling spring on the contact surface.

7.2. Electrical connection

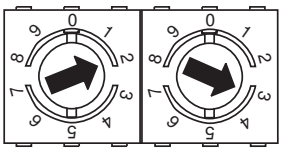
Only ever store or transport the bus cover in the ESD bag. The bus cover must rest fully against the housing and be firmly screwed in place.
For electrical connection, pull off the bus cover using the following method:

- Release the fastening screws of the bus cover
- Carefully loosen the bus cover and lift off in the axial direction

7.2.1. Setting the user address (Node ID)

The user address is set via the EEPROM. The node ID (user address) is defined in object 2101h. In addition, it is possible to set the user address decimally using two rotary switches in the bus cover. If the switches are at 0, the node ID from the EEPROM is used. As soon as the switch is set to a value, this set value is used as the user address. The maximum number of users is 99.

- Set the user address decimally using the two rotary switches 1 and 2 (default setting 01).



Example: 23

For BOxH 58S1x24Bxx/xxxxH (Dignalizer CANopen integrated)
Set the Node-ID and the terminal resistor using the Dip-Switch.

Position S1	Position S2	Position S3	Position S4	Position S5	Position S6	Position S7	Position S8	Node-ID	Terminating resistor
$2^0 = 1$	$2^1 = 2$	$2^2 = 4$	$2^3 = 8$	$2^4 = 16$	$2^5 = 32$	$2^6 = 64$			
OFF	OFF	OFF	OFF	OFF	OFF	OFF	x	Factory default	
ON	OFF	OFF	OFF	OFF	OFF	OFF	x	1	
OFF	ON	OFF	OFF	OFF	OFF	OFF	x	2	
ON	ON	OFF	OFF	OFF	OFF	OFF	x	3	
OFF	OFF	ON	OFF	OFF	OFF	OFF	x	4	
ON	OFF	ON	OFF	OFF	OFF	OFF	x	5	
OFF	ON	ON	OFF	OFF	OFF	OFF	x	6	
ON	ON	ON	OFF	OFF	OFF	OFF	x	7	
OFF	OFF	OFF	ON	OFF	OFF	OFF	x	8	
ON	OFF	OFF	ON	OFF	OFF	OFF	x	9	
OFF	ON	OFF	ON	OFF	OFF	OFF	x	10	
ON	ON	OFF	ON	OFF	OFF	OFF	x	11	
OFF	OFF	ON	ON	OFF	OFF	OFF	x	12	
ON	OFF	ON	ON	OFF	OFF	OFF	x	13	
...							x	...	
ON	ON	ON	ON	ON	ON	ON	x	127h	
x	x	x	x	x	x	x	OFF	x	open
x	x	x	x	x	x	x	ON	x	120 Ω

OFF = 0
ON = 1
x = no influence

7.2.2. Setting the baud rate

The baud rate is defined in the object 2100h. In addition, it is possible here to set the baud rate using a switch. The baud rate setting is performed on a binary basis via switches 1 to 3 of the 3-pin DIP switch in the bus cover. The baud rate used from the EEPROM is ignored as soon as the switch for the user address is not set to 0.

Baud rate	Setting DIP switches		
	1	2	3
10 kBit/s	OFF	OFF	OFF
20 kBit/s	OFF	OFF	ON
50 kBit/s *	OFF	ON	OFF
125 kBit/s	OFF	ON	ON
250 kBit/s	ON	OFF	OFF
500 kBit/s	ON	OFF	ON
800 kBit/s	ON	ON	OFF
1 MBit/s	ON	ON	ON

* Factory setting:



For BOxH 58S1x24Bxx/ xxxxH (Dignalizer CANopen integrated)

Set the baud rate using the rotary switch.

Rotary switch	Baud rate [kBit/s]
0	10
1	20
2	50
3	125
4	250
5	500
6	800
7	1000

7.2.3. Terminating resistor

If the connected encoder is the last device in the bus line, the bus must be terminated with a resistor. The resistor is in the bus cover and is connected using a one-pole DIP switch. The terminating resistor must be switched to "ON" at the last user with a 1-pole DIP switch (default setting OFF).



ON = Final user
OFF = User X

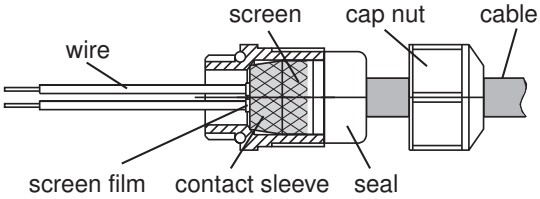


Switch 1: ON = Final user
OFF = User X
Switch 2: Without function

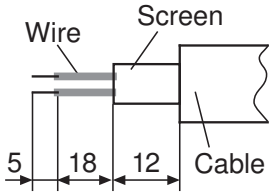
7.2.4. Bus cover connection

- Release the cap nut of the cable gland.
- Push the cap nut and seal insert with contact sleeve onto the cable sheath.
- Strip the cable sheath and cores, shorten the shield film where this exists (see Fig.)
- Bend over the braided screen by approx. 90°.
- Push the sealing insert with contact sleeve along as far as the braided shield. Insert the sealing insert with contact sleeve and cable flush into the cable gland and tighten the cap nut.



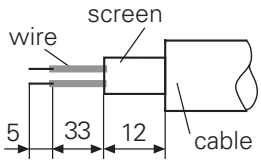


For standard encoder

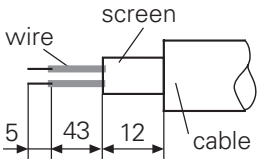


For BISD and BIMD

Bus cable



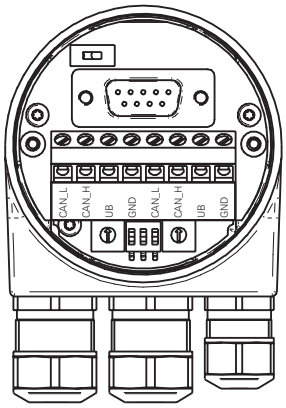
Supply voltage cable

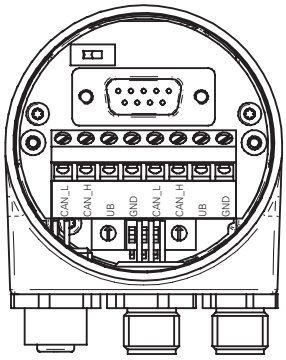


- Terminals with the same designation are internally interconnected.
- For the power supply, use only cable gland 3. For the bus lines, cable gland 1 or 2 can be optionally selected. For the bus lines, cable glands 1 or 2 can be freely selected. Observe the admissible cable cross sections.
- Insert the cores using the shortest route from the cable gland to the terminal strip. Observe the admissible core cross-section. Use isolated core end sleeves.
- Avoid crossing over data lines with the supply voltage line.

Bus cover - axial



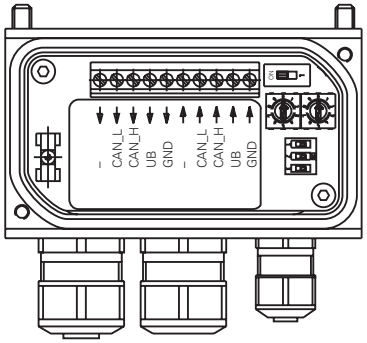


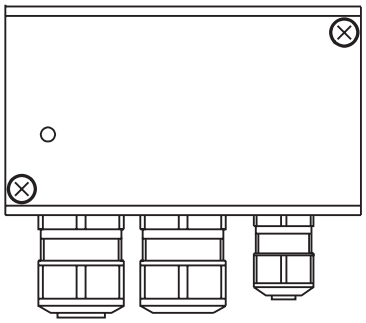


1 2 3 Cable gland M12 connector

Bus cover – radial

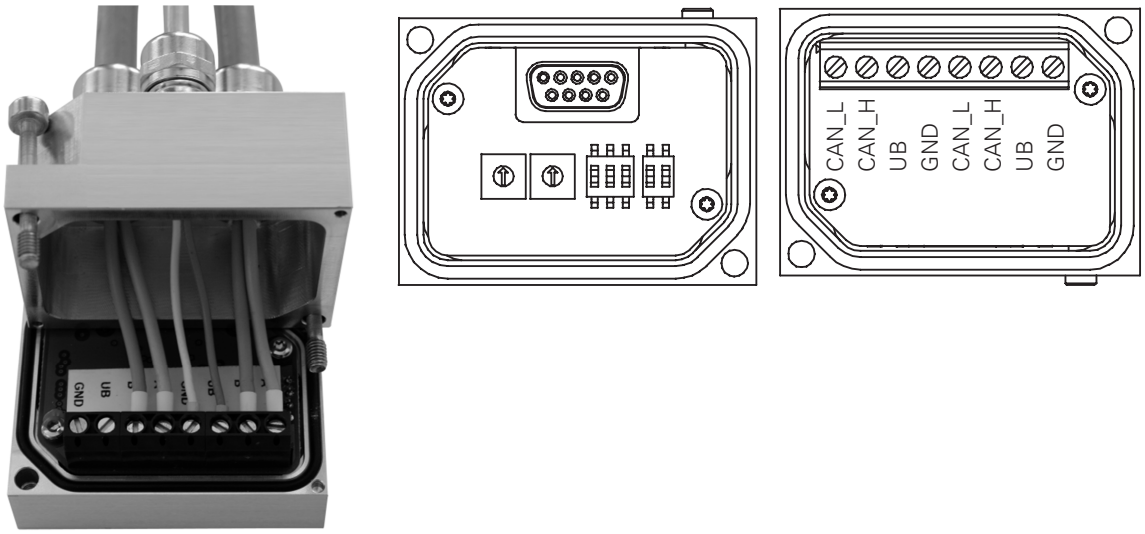






1 2 3

Bus cover – radial (BISD and BIMD)



7.2.5. Terminal assignment

Terminal	Explanation
CAN_L	CAN bus signal (dominant low)
CAN_H	CAN bus signal (dominant high)
UB	Supply voltage 10 - 30 VDC
GND	Ground terminal for UB

Terminals with the same designation are internally interconnected.

- Carefully plug the bus cover onto the D-SUB plug of the basic encoder, then press only via the sealing rubber, taking care not to tilt it. The bus cover must rest fully against the basic encoder.
- Tighten both the fastening screws firmly in the same direction.

The encoder housing and braided shield of the connecting cable are only ideally connected if the bus cover is resting fully on the basic encoder (positive locking).

7.3. Display elements (status display)

A dual LED is integrated at the back of the bus cover.

LED green	LED red	Status
Off	Off	Supply voltage not connected
Flashing	Off	Pre-operational mode
On	Off	Operational mode
On	Off	Stopped/Prepared mode
Off	Flashing	Warning
Off	On	Error

8. Connection and start-up with integrated fieldbus interface

8.1.1. Setting of node-ID and baud rate

The node-ID can be set via software only. This is done by writing the effective node-ID in object 2101h. For further details, see object description.

Example: Set node-ID to 23h:

COB-ID	DLC	command	object L	object H	subindex	data 0	data 1	data 2	data 3
600h+node-ID	8	2Fh	01h	21h	0h	23h	Xx	Xx	Xx

Confirmation:

COB-ID	DLC	command	object L	object H	subindex	data 0	data 1	data 2	data 3
580h+node-ID	8	60h	01h	21h	0h	0h	0h	0h	0h

The baud rate can be set via software only. This is done in object 2100h by selecting the index which corresponds to the effective baud rate as shown in the following table. For further details, see object description.

	baud rate
0h	10 kbit/s
1h	20 kbit /s
2h	50 kbit /s
3h	100 kbit /s
4h	125 kbit /s
5h	250 kbit /s
6h	500 kbit /s
7h	800 kbit /s
8h	1000 kbit /s

Example: Set baud rate to 250 kbit/s (5h):

COB-ID	DLC	command	object L	object H	subindex	data 0	data 1	data 2	data 3
600h+ node-ID	8	2Fh	00h	21h	0h	05h	Xx	Xx	Xx

Confirmation:

COB-ID	DLC	command	object L	object H	subindex	data 0	data 1	data 2	data 3
580h+ node-ID	8	60h	00h	21h	0h	0h	0h	0h	0h

For non-volatile storage of settings, please write message "save“ into subindex 1 of object 1010h. For further details, see following example or object description.

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’

Confirmation:

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	0h	0h	0h	0h

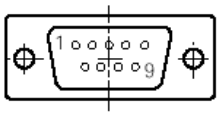
New node-ID or new baud rate becomes active after next initialisation. Please note, that baud rate of master must be set accordingly.

8.1.2. Terminating resistor

In order to allow optimum bus communication with baud rate up to 1 Mbit/s, the bus must be terminated correctly. Beginning and end of bus must be terminated by a resistor (120 Ω, ¼ W).

8.1.3. Pin assignment D-Sub connector

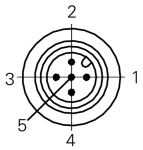
pin no.	signal	description	Cable color
1		n.c.	
2	CAN_L	CAN bus signal (dominant LOW)	yellow
3	CAN_GND	CAN bus ground	gray
4		n.c.	
5		n.c.	
6	0 V	Supply voltage ground	white
7	CAN_H	CAN bus Signal (dominant HIGH)	green
8		n.c.	
9	UB	Supply voltage 10...30 VDC	brown



View to encoder

8.1.4. Pin assignment connector M12

Pin no.	Signal	Description	
1		n.c.	
2	+Vs	Supply voltage 10...30 VDC	
3	CAN_GND	CAN bus ground / 0 V supply voltage	
4	CAN_H	CAN bus signal (dominant High)	
5	CAN_L	CAN bus signal (dominant Low)	



View to encoder