hohner
AUTOMAZIONE SRL

Technical Manual

CANopen

Absolute Shaft Encoder

CANopen

Item no.: 2 565 392, Version: 3 251005 HOR

© by HOHNER AUTOMAZIONE SRL

HOHNER AUTOMAZIONE SRL claims copyright protection for this documentation.

This documentation must not be modified, amended, copied or given to third parties without prior written approval by HOHNER AUTOMAZIONE SRL.

We reserve the right to make technical modifications and improvements that serve the advancement of our products.

Hohner Automazione srl
P.le Cocchi 10, 21040
Vedano Olona (VA) Italy
Tel. +39 0332 866109
Fax +39 0332 866109
E-Mail: hohner.info@hohner.it
www.hohner.it

Content

1 Definitions

This technical manual describes the software, parameter setting and initial operation of the shaft encoder.

Explanation of symbols:

Passages to which special attention should be paid in order to ensure the correct use and to avoid **dangers** are marked by this symbol.

This Symbol indicates important directions for the **proper use** of the shaft encoder.
The non-observance of these instructions may lead to malfunctions in the shaft encoder or its surrounding parts.

Abbreviations used

CAL	CAN Application Layer. Application layer (layer 7) within the CAN communication model.
CAN	Controller Area Network
CiA	CAN in Automation. International users and manufacturers group of CAN products.
CMS	CAN Message Specification. CAL service element
COB	Communication Object. Transport unit within the CAN network (CAN message). Data is sent via the network within a COB.
COB-ID	COB identifier. Clear identification of a CAN message. The identifier determines the COB's priority in the network.
DBT	Distributor. CAL service element, responsible for the dynamic allocation of identifiers.
DS	Draft Standard
DSP	Draft Standard Proposal
ID	Identifier, see COB-ID
LMT	Layer Management. CAL service element, responsible for the configuration of the parameter within the individual layers of the communication model.
LSB	Least Significant Bit/Byte
MSB	Most Significant Bit/Byte
NMT	Network Management. CAL service element, responsible for the initialization, configuration, and error recovery within the network.
OSI	Open Systems Interconnection. Layer model for the description of the different functional regions of the data communication system.
PDO	Process Data Object. Object for the exchange of process data.
RTR	Remote Transmission Request; data request message
SDO	Service Data Object; Communication object that enables the master to gain access to the object directory of a node.
SYNC	Synchronization message. Stations in the bus respond with their process value to the SYNC command.

Numerical data Unless indicated explicitly, decimal values are represented as figures without additional features (e.g. 1408), binary values are marked **b** (e.g. 1101b), hexadecimal values **h** (e.g. 680h) at the end of the figures.

2 Safety and Operating Instructions

The HOHNER® model series' absolute shaft encoders are quality products that have been manufactured according to recognized electrical engineering regulations. The devices have left the manufacturing company's premises meeting all relevant safety requirements.

Therefore:

- In order to preserve this condition and to ensure an interference-free Operation of the encoders, the technical specifications presented in this documentation must be observed.
- Electrical appliances may only be installed by skilled electricians!
- The devices may only be operated within the limits defined in the technical data.
- The maximum operating voltages must not be exceeded!!
  The devices have been constructed according to DIN EN 61010 Part 1, protection class III.
  In order to avoid dangerous electric shocks, the devices have to be operated with safety extra-low voltage (SELV) and be situated in a field with equipotential bonding.
- For better protection use an external fuse Field of application: industrial processes and control systems.
  Over voltages at the connection terminals have to be restricted to over voltage category II values.
- Shock effects on the housing, especially on the encoder shaft, as well as axial and radial overloading of the encoder shaft should be avoided.
- Only in case an appropriate coupling is used can the maximum precision and life time be guaranteed.
- The proper electromagnetic compatibility values (EMC) are only valid for standard cables and plugs. In the case of screened cables, the screen has to be connected on both sides as well as on large surface to ground. The lines for power supply should also be entirely screened. If this is not possible, appropriate filter methods should be applied.
- The neighbouring parts as well as the installation of the cable system have got a significant influence on the electromagnetic compatibility of the shaft encoder. As a consequence, the electrician has to ensure the EMC of the entire system (device).
- In regions endangered by electrostatic discharges, a good ESD protection for the plugs and the cable to be connected should be provided when installing the shaft encoder.
- For bus resp. connecting cables, only use signal cables with twisted wire pairs (CAN+ with CAN-, +UB with 0 V) with screen.

3 General Information

3.1 Introduction

The HOHNER industry is an absolute shaft encoder (encoder, angle encoder). The version described in this technical manual sends its current position to another station via the "CAN -bus" "transmission medium (physically: screened and twisted two-wire line).

The serial bus system CAN (Controller Area Network), which had been originally developed by Bosch/ Intel for automotive uses, is gaining ground in industrial automation technology. The system is multimaster-compatible, i.e. several CAN- stations are able to request the bus at the same time. The message with the highest priority (determined by the identifier) will be received immediately.

The data transfer is regulated by the message's priority. Within the CAN system, there are no transport addresses, but message identifiers. The message which is being sent can be received by all stations at the same time (broadcast). By means of a special filter methods, the station only accepts the relevant messages. The identifier transmitted with the message is the basis for the decision as to whether the message will be accepted or not.

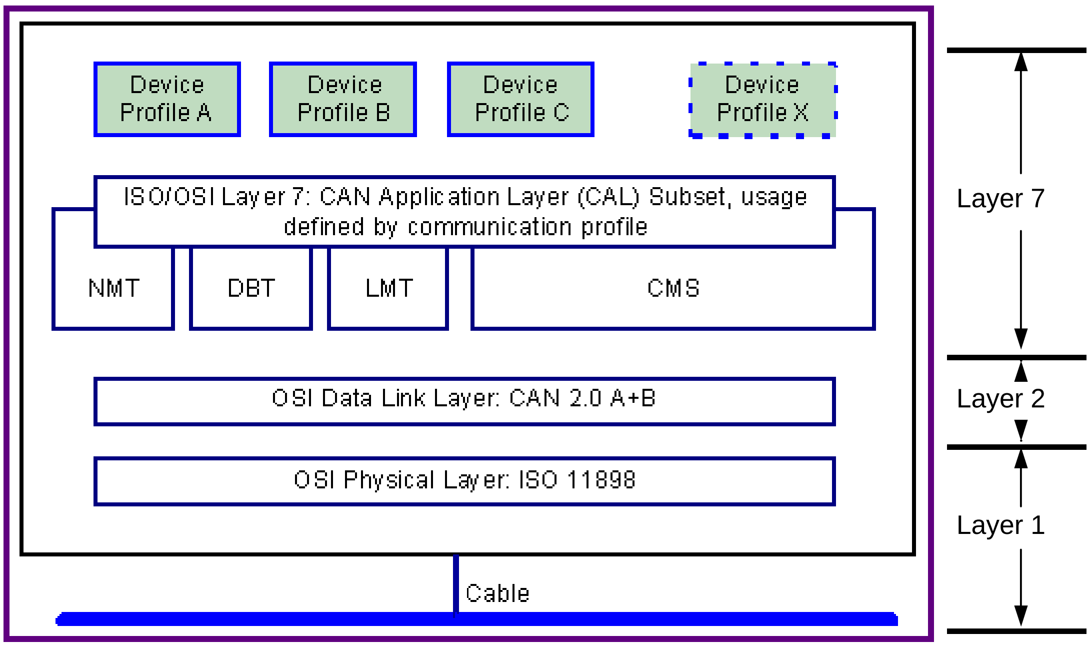
The bus coupler is standardized according to the international standard ISO-DIS 11898 (CAN High Speed) standard and allows data to be transferred at a maximum rate of 1 Mbit/ s. The most significant feature of the CAN-protocol is its high level of transmission reliability(Hamming distance = 6).

The CAN-Controller Intel 82527 used in the encoder is basic as well as full-CAN compatible and supports the CAN-specification 2.0 part B (standard protocol with 11-bit- identifier as well as extended protocol with 29-bit identifier). Up to now, only 11-bit identifiers have been used for CANopen.

3.2 Field of Application

In systems, where the position of a drive or of any other part of a machine has to be recorded and signaled to the control system, the HOHNER industry can assume this function. The HOHNER industry can resolve, for instance, positioning tasks by sending the check-back signal concerning the present drive position via the CANopen to the positioning unit.

3.3 CANopen-Kommunikationsmodell und Profile


Device Profile A
Device Profile B
Device Profile C
Device Profile X
ISO/OSI Layer 7: CAN Application Layer (CAL) Subset, usage defined by communication profile
NMT
DBT
LMT
CMS
OSI Data Link Layer: CAN 2.0 A+B
OSI Physical Layer: ISO 11898
Cable
Layer 7
Layer 2
Layer 1

Layer 1 (Physical Layer):	ISO-DIS 11898 (CAN High Speed)
Layer 2 (Data Link Layer):	ISO-DIS 11898 (CAN High Speed)
Layer 7 (Application Layer):	CiA DS 301 (CANopen CAL-based Communication Profile) + Device profile CiA DS 4xx (CANopen Device Profile for xx)

For the following devices, profiles already exist:

- CiA Draft Standard Proposal 401 for Input/Output Modules
- CiA Draft Standard Proposal 402 for Drives and Motion Control
- CiA Work Item 403 for Human Machine Interfaces
- CiA Work Draft 404 for Closed-Loop Controllers and Transformers
- CiA Work Item 405 for IEC-1131 Interfaces
- **CiA Draft Standard Proposal 406 for Encoders**
- CiA Work Item 407 for Public Transport
- CiA Work Item 408 for Fork-Lifts

3.4 The CANopen profile

About two and a half years after the CiA, the association of the user and manufacturer of CAN products, had adopted the CAN-Application Layer (CAL), CANopen and the respective device profiles paved the way for the development of open systems.
CANopen has been developed under the technical direction of the Steinbeis Transfer Centre for Automation (STA Reutlingen; Germany) on the basis of the layer 7 CAL specification.
Compared with CAL, CANopen only provides the functions needed for this special purpose.CANopen is thus a part of CAL which has been optimised for application purposes and allowsfor a simpler system

structure as well as for simpler devices. CANopen has been optimized for a quick transfer of data in real-time systems and has been standardized for different device profiles. The CAN in Automation (CiA) association of users and manufacturers is responsible for the establishing and the standardization of the respective profiles. The Encoder with CANopen meets the requirements laid down in the communication profile (CiA DS 301) and in the device profile for encoders.

CANopen allows for

- Auto configuration of the network,
- Comfortable access to all device parameters.
- Synchronization of the devices,,
- Cyclical and event-controlled process data processing,
- Simultaneous data input and output.

CANopen uses four communication objects(COB)with different features:

- Process Data Objects (PDO) for real-time data
- Service Data Objects (SDO) for the transfer of parameters and programs
- Network Management (NMT, Life-Guarding)
- predefined objects (for synchronization, time stamp, emergency message)

All device parameters are stored in an object directory. The object directory contains the description, data type and structure of the parameters as well as their addresses (index).The directory consists of three parts: communication profile parameters, device profile parameters and manufacturer specific parameters (see Chapter 5 "Object Directory").

3.5 The Encoder Device Profile (CiA DSP 406)

This profile describes a binding, but manufacturer-independent definition of the interface for encoders. The profile not only defines which CANopen functions are to be used, but also how they are to be used. This standard permits an open and manufacturer-independent bus system.

The device profile consists of two object categories:

- the standard category C1 describes all the basic functions the shaft encoder must contain

- the extended category C2 contains a variety of additional functions which either have to be supported by category C2 shaft encoders (mandatory) or which are optional. Category C2devices thus contain all C1 and C2 mandatory functions as well as, depending on the manufacturer, further optional functions.

Furthermore, an addressable area is defined in the profile, to which, depending on the manufacturer, different functions can be assigned.

4 Data transfer

In CANopen, the data is transferred by means of two different communication types(COB=Communication Object) with different features:

- **Process Data Objects (PDO)**
- **Service Data Objects (SDO)**

The **process data objects (PDO)** serve the highly dynamic exchange of real-time data (e.g. position of the shaft encoder) with a maximum length of 8 Byte. This data is transferred with high priority (low COB identifier). PDOs are broadcast messages and put their information simultaneously at the disposal of all desired receivers.
The **service data objects (SDO)** form the communication channel for the transfer of device parameters (e.g. programming of the shaft encoders' resolution). Since these parameters are transferred acyclically (e.g. only once when running up the network), the SDO objects have a low priority (high COB identifier).

The priority of the message objects is determined by the COB identifier.

4.1 COB-Identifier

For an easier administration of the identifiers, CANopen uses the "Predefined master/Slave Connection Set"). In this case, all identifiers with standard values are defined in the object directory. However, these identifiers can be modified according to the customers' needs via SDO access..

The 11-bit identifier consists of a 4 bit function code and a 7 bit node number.

Bit-Nr.	10	9	8	7	6	5	4	3	2	1	0
Art	Function code				Node number						
Belegung[1]	x	x	x	x	0	0	x	x	x	x	x

[1] x = binary value can be selected freely (0 or 1); 0 = value 0 is fixed

The higher the value of the COB identifier, the lower the identifier's priority!

4.1.1 Node number

The 7-bit node number is set, as far as the hardware is concerned, by means of the 5 DIP switch on the rear of the shaft encoder (see Chapter 8.1.1 "Setting of the node number").The 5 DIP switches determine the bits 0 through 4, the remaining bits 5 and 6 are irrevocably set to 0.

Node number 0 is reserved and may not be used by any node. The resulting node numbers range from 1 through 31. A new node number is not accepted unless the encoder is run up again (Reset/ Power-on).All identifiers will be reset to their standard values.

4.1.2 Message Objects and Function Codes

The following function codes are defined in the Predefined master/Slave Connection Set:

Object[1]	**Function code**	**Resulting COB ID**[2]		**CMS Priority-**
	binary	**hexadecimal**	**decimal**	**group**
NMT	0000	0	0	0
SYNC[3]	0001	80	128	0
TIME STAMP	0010	100	256	1
EMERGENCY[3]	0001	81 - FF	129 - 255	0, 1
PDO 1 (tx)[3]	0011	181 - 1FF	385 - 511	1, 2
PDO 2 (tx)[3]	0101	281 - 2FF	641 - 767	2, 3
SDO (tx)	1011	581 - 5FF	1409 - 1535	6
SDO (rx)	1100	601 - 67F	1537 - 1663	6, 7
Nodeguard	1110	701 - 77F	1793 - 1919	-

[1] NMT, SYNC, TIME STAMP are broadcast objects that are directed to all nodes; all the others are peer-to-peer objects which are only accepted by addressed nodes.

[2] COB ID = Function code • 128 + node number (DIP switch + 1). However, the standard identifiers can be modified via SDO access to the respective object directory entries.

[3] After being modified, the parameters can be stored via the bus by means of the object 1010h in the electrically erasable programmable read only memory (EEPROM) of the shaft encoder.

4.2 Transfer of process data

There are two PDO services PDO1 (tx) and PDO2 (tx) available. A PDO transfer can be initialized by various events (see Object directory index 1800h on page 27):

- asynchronously (event-controlled) by means of the internal device timer or by means of a modification of the process values
- synchronously as response to a SYNC message; (The SYNC command makes the CANopen nodes store their data synchronously in order to put them one after the other and according to their priority on the bus)
- as response to an RTR message; (by means of the remote frame = recessive RTR bit, precisely the message with the transferred identifier will be requested)

PDO message have the following structure:

COB-ID	**Process Data in Binary Code**							
11 Bit	Byte 0 $2^7 ... 2^0$	Byte 1 $2^{15} ... 2^8$	Byte 2 $2^{23} ... 2^{16}$	Byte 3 $2^{31} ... 2^{24}$	Byte 4	Byte 5	Byte 6	Byte 7
Default:	Value of position							
Typ1:	Value of position				Flags[1)]			
Typ2:	Value of position				Speed[2)]		Acceleration[3)]	

[1)] signal flags of the warning position, see also object directory 2004h index
[2)] see object directory 2002h index

[3)] see object directory 2003h index

The PDO-type (Default, type 1, type 2) is set in the object directory in the **2005h index**.

The COB ID as well as the transfer mode for **PDO1** is determined in the object directory **index 1800h** (see page 27). The setting can be modified by means of the SDO access
Standard settings:

Release:	PDO **active** (enabled)
COB-ID:	**180h+node number**
Transfer mode:	**254 =** asynchronous, i.e. the output of the process value is initiated by the shaft encoder (event-controlled by an internal device timer or by a modification of the process value). The timer takes priority, if a cycle time > 0 has been defined in index 6200. In addition, the request via RTR is possible..

The COB ID as well as the transfer mode for **PDO2** is determined in the object directory**1802h index** (see page 29). The setting can be modified by means of the SDO access.

Standard settings:

Release:	PDO **inactive** (disabled)
COB-ID:	**280h+node number**
Transfer mode:	**1 =** synchronous, i.e. the output of the process value is initiated by SYNC messages. In addition, the request via RTR is possible.

4.3 Transfer of service data (Parameter Setting)

All the device parameters are filed in this object directory (see Chapter 5) under standardized addresses (index) and can be described and read via SDOs. SDOs are exchanged between two stations (configuration master and shaft encoder) by means of the handshake procedure (request or release).

There are two SDO services available:
- one transfer SDO (tx) for messages (parameter or release) from the shaft encoder to the master
- one receiver SDO (rx) for messages (request or parameter)from the master to the shaft encoder.

SDO messages have the following structure (here: expedited protocol for data volumes up to4 bytes):

COB-ID	**Command**	**Index**		**Subindex**	**Service Data (Parameters)**			
11 Bit	Byte 0	Byte 1 (LSB)	Byte 2 (MSB)	Byte 3	Byte 4 (LSB)	Byte 5	Byte 6	Byte 7 (MSB)

Service data longer than 4 bytes (Objects 1008h, 1009h, 100Ah) are transferred by the segmented protocol.
The meaning of index, subindex and data will be explained in Chapter 5 "Object Directory".

4.3.1 SDO-COB-ID

SDO (tx) (Encoder →Master): 580h (1408) + Node number
SDO (rx) (Master□→ Encoder): 600h (1536) + Node number
SDO identifiers cannot be modified!

The following identifiers are standard for the SDO services:

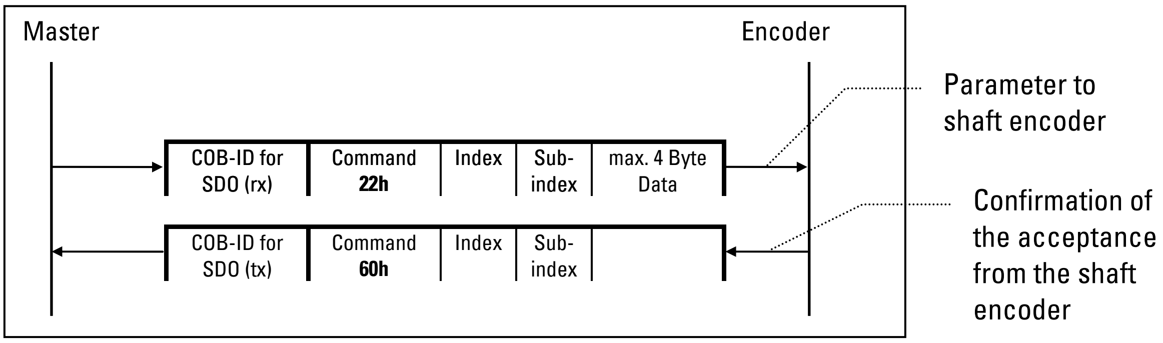
4.3.2 Command

The command byte describes the type of SDO message:

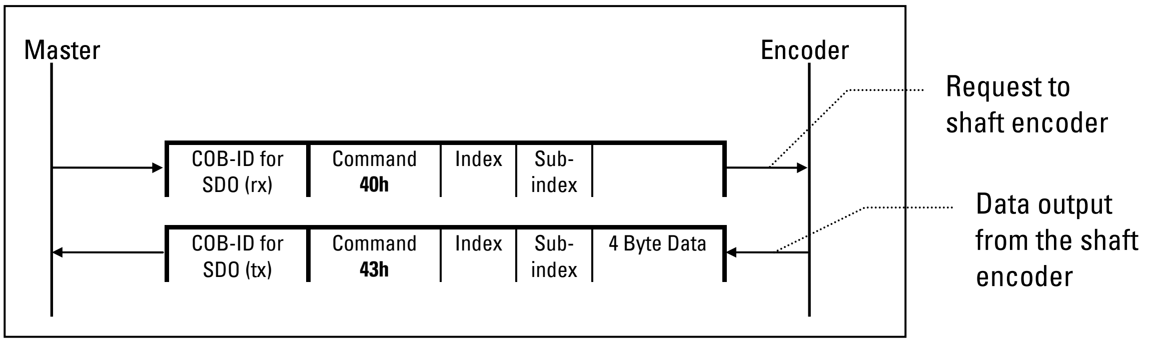
Command (Expedited Protocol)	**Type**	**Function**
22h	SDO(rx), Initiate Download Request	Send parameters to shaft encoder (Data length max. 4 Byte)
23h	SDO(rx), Initiate Download Request	Send parameters to shaft encoder (Data length max. 4 Byte)
2Bh	SDO(rx), Initiate Download Request	Send parameters to shaft encoder (Data length max. 2 Byte)
2Fh	SDO(rx), Initiate Download Request	Send parameters to shaft encoder Data length max. 1 Byte)
60h	SDO(tx), Initiate Download Response	Confirmation of acceptance to master
40h	SDO(rx), Initiate Upload Request	Request parameter from encoder
43h	SDO(tx), Initiate Upload Response	Parameter to master with Data length =4 Byte (Unsigned 32)
4Bh	SDO(tx), Initiate Upload Response	Parameter to master with Data length =2 Byte (Unsigned 16)
4Fh	SDO(tx), Initiate Upload Response	Parameter to master with Data length =1 Byte (Unsigned 8)
80h	SDO(tx), Abort Domain Transfer	Encoder signals error code to master

- An error message (Command 80h) replaces the normal confirmation in case of an error (Response).
- The error message does not only include communication protocol errors (e.g. the wrong command byte) but also errors concerning the access to the object directory-(e.g. wrong index, write attempt on read only object, wrong data length etc.).The error codes are described in the CANopen profile (DS 301) and the device profile (DSP 406) respectively.

4.3.3 Example for the Transfer of service data



Master sends parameters to the encoder



Master requests parameters from the shaft encoder

5 Object directory

All features and parameters of an CANopen device are stored in its object directory.

The entire data of the object directory are stored in the EEPROM of the shaft encoder to be protected against power failures and will be copied into the main memory (RAM) incase the Power-on or Reset. If data in the object directory are changed, the modification will only become effective in the main memory. If the data is to be saved permanently, the data must by all means be imported in the EEPROM via the object 1010h (Save parameters).

The existent data in the EEPROM will be overwritten!

Access to the object directory (read or write parameters) is gained via the SDO services which are described in Chapter 4.3 "Transfer of Service Data (Parameter Setting)".

The object directory consists of different parts:

- features that are valid for all CANopen devices (DS 301)
- features that are valid for shaft encoders (device profile DSP 406)
- features depending on the manufacturer

The address (index), which points at any entry in the object directory, is, except for the features depending on the manufacturer, standardized within the profiles. It is thus ensured that all devices always execute the functions described in the profile (standard and optional functions) within the same index. This is one of the requirements for an open system and the exchangeability of the devices.

The entries in the object directory are addressed by means of a 16 bit index. Each index entry can further be subdivided by a subindex.

The description of the object directory entries is structured as follows:

Index (hex)	Sub-Index (hex)	Object	Name	Type	Attr.	M/O

Index:	16 bit address of the entry	
Sub-Index:	8 bit pointer on the subentry; is only used with complex data structures (e.g. Record, Array) ;if there is no subentry: SubIndex=0	
Object:	NULL	Entry without data
	DOMAIN	larger variable data set, e.g. program code
	DEFTYPE	Definition of the data types, e.g. Boolean, float, unsigned16 etc.
	DEFSTRUCT	Definition of e record-entry, e.g. PDO Mapping Structure
	VAR	single data value, e.g. Boolean, float, unsigned16, string etc.
	ARRAY	Field with similar data, e.g. unsigned16 data
	RECORD	Field with an arbitrary mixture of data
Name:	short description of the function	
Type:	data type, e.g. Boolean, float, unsigned16, integer etc.	
Attr.:	attribute grants access rights to the object:	
	rw	read and write access
	ro	read only access
	const	read only access, value is a constant parameter
M/O	M	Mandatory: Object has to be implemented in the device
	O	Optional: Object does not have to be implemented in the device
	(The M/O-Indications in the following chapters refer to a category 2 shaft encoder according to the device profile DSP 406)	

Organisation of the entire object directory:

Index (hex)	Object
0000	not used
0001 - 001F	static data types
0020 - 003F	complex data types
0040 - 005F	manufacturer specific data types
0060 - 0FFF	reserved
1000 - 1FFF	communication profile
2000 - 5FFF	manufacturer specific profile
6000 - 9FFF	standardised device profile
A000 - FFFF	reserved

5.1 Encoder Entries in the object directory

5.1.1 Overview

The following entries are realized in the object directory of the Encoder according to the CANopen profile DS 301 and the device profile DSP 406:

Index (hex)	Object	Name	Type	Attr.	M/O	See page
Communication parameters:						
1000	VAR	Device type	Unsigned32	ro	M	*21*
1001	VAR	Error register	Unsigned8	ro	M	*22*
1003	ARRAY	Predefined error field	Unsigned32	ro	O	*22*
1004	ARRAY	Number of the supported PDOs	Unsigned32	ro	O	*22*
1005[1]	VAR	COB ID for SYNC message	Unsigned32	rw	O	*23*
1008	VAR	Manufacturer device name	Vis-String	ro	O	*23*
1009	VAR	Hardware version	Vis-String	ro	O	*23*
100A	VAR	Software version	Vis-String	ro	O	*23*
100B	VAR	Node number	Unsigned32	ro	O	*24*
100C	VAR	Monitoring time	Unsigned16	rw	O	*24*
100D	VAR	Life time factor	Unsigned8	rw	O	*24*
100E	VAR	COB-ID for node guarding	Unsigned32	rw	O	*24*
100F	VAR	Number of the supported SDOs	Unsigned32	ro	O	*25*
1010	ARRAY	Save parameters	Unsigned32	rw	O	*25*
1011	ARRAY	Load standard values	Unsigned32	rw	O	*25*
1014[1]	VAR	COB-ID for emergency objects	Unsigned32	rw	O	*26*
1018	RECORD	Identity object		ro	M	*26*
1800[2]	RECORD	PDO1 parameter		rw	M	*27*
1802[2]	RECORD	PDO2 parameter		rw	M	*29*
Encoder parameters (manufacturer specific):						
2000[1]	ARRAY	Warning positions	Unsigned32	rw	O	*30*
2001[1]	VAR	Offset value	Signed32	rw	O	*30*
2002	VAR	Speed	Signed16	rw	O	*30*
2003	VAR	Acceleration	Signed16	rw	O	*31*
2004	VAR	Signal flags of the warning position	Unsigned8	ro	O	*31*
2005[1]	VAR	PDO type	Unsigned8	rw	O	*32*

Encoder parameters (gen.):

6000[1]	VAR	Service parameters	Unsigned16	rw	M/O	*33*
6001[1]	VAR	Measuring steps per revolution	Unsigned32	rw	M	*33*
6002[1]	VAR	Total number of measuring steps	Unsigned32	rw	M	*34*
6003[1]	VAR	Preset value	Unsigned32	rw	M	*35*
6004	VAR	Value of position	Unsigned32	ro	M	*35*
6200[1]	VAR	Cycle timer	Unsigned16	rw	M	*35*

Encoder diagnosis parameters:

6500	VAR	Operating status	Unsigned16	ro	M	*36*
6501	VAR	phys. Resolution single turn	Unsigned32	ro	M	*36*
6502	VAR	phys. Number of revolution	Unsigned16	ro	M	*37*
6503	VAR	Alarm message	Unsigned16	ro	M	*37*
6504	VAR	Supported alarm message	Unsigned16	ro	M	*37*
6505	VAR	Warning message	Unsigned16	ro	M	*37*
6506	VAR	Supported warning message	Unsigned16	ro	M	*38*
6507	VAR	Profile- and Software version	Unsigned32	ro	M	*38*
6508	VAR	Operating time counter	Unsigned32	ro	M	*39*
6509	VAR	Offset value	Signed32	ro	M	*39*
650A	VAR	Manufacturer offset	Signed32	ro	M	*39*
650B	VAR	Serial number	Unsigned32	ro	M	*39*

[1] After being modified, the parameters can be stored via the bus by means of the object 1010h in the electrically erasable programmable read only memory (EEPROM) of the shaft encoder.

[2] The SubIndex parameters COB-ID and Inhibit time can be stored to the nonvolatile encoder storage (EEPROM) via Object 1010h.

5.1.2 Detailed description of the communication parameters

5.1.2.1 Object 1000h: Device type

Provides information on the device profile and the device type used.

1000	VAR	Device type	Unsigned32	ro	M

Data contents

Device type number		Encoder type	
Byte 0 (LSB)	Byte 1	Byte 2	Byte 3 (MSB)
96h*	01h*	01h (absolute shaft encoder single turn) 02h (absolute shaft encoder multi turn)	00h

* 196h = 406 decimal (encoder profile)

5.1.2.2 Object 1001h: Error register

Occurring device errors will be indicated here

1001	VAR	Error register	Unsigned8	ro	M

Data content:

Bit 0 = 1:	general error (Shaft encoder-alarm signal)
Bit 1...7:	unsigned

5.1.2.3 Object 1003h: Predefined error field

The occurring errors will be indicated here. The last 8 errors will be stored in the error field.

1. The entry in subindex 0 contains the number of the errors stored.
2. Every new error will be stored in subindex 1, the existing entries will be displaced by one position.
3. The entire error list will be deleted if the 0 value is written in the subindex 0.

1003	ARRAY	Predefined error field	Unsigned32	ro	O

5.1.2.4 Object1004h: Number of supported PDOs

The object contains information on the maximum rate of PDOs supported by the device. Subindex 0h contains the total number of transmit (byte 0 and 1) and receive (byte 2 and 3)PDOs. Subindex 1h contains the number of synchronous transmit (byte 0 and 1) and receive (byte 2and 3) PDOs. Subindex 2h contains the number of asynchronous transmit (byte 0 and 1) and receive (byte 2and 3) PDOs.

1004	ARRAY	Number of supported PDOs	Unsigned32	ro	O

Data content:

Sub-Index 0h:	2h (2 Transmit-PDO´s)
Sub-Index 1h:	1h (1 synchronous Transmit-PDO)
Sub-Index 2h:	1h (1 asynchronous Transmit-PDO)

5.1.2.5 Object 1005h: COB-ID for SYNC-Nachricht

The object defines the COB ID for the SYNC message. Additionally, it is defined, whether the device generates or receives SYNC objects.

1005	VAR	COB-ID for SYNC-Nachricht	Unsigned32	rw	0

Data contents:

Bit 0...10:	11 Bit Identifier; **Standard-ID = 80h**
Bit 11...29:	0 (reserved for 29 Bit Identifier devices)
Bit 30:	0 (device does not generate SYNC message)
Bit 31:	1 (device receives SYNC message)

5.1.2.6 Object 1008h: Manufacturer device name

Contains the manufacturer device name.

1008	VAR	Manufacturer device name	Vis-String	ro	0

Data contents:

"EncoderCAN" in the ASCII-Code

5.1.2.7 Object 1009h: Hardware version

Contains the hardware version number.

1009	VAR	Hardware version	Vis-String	ro	0

Data contents:

e.g. "HW-V2" in the ASCII code

5.1.2.8 Object 100Ah: Software version

Contains the software version number.

100A	VAR	Software version	Vis-String	ro	0

Data contents:

e.g. "SV03.00" in the ASCII code

5.1.2.9 Object 100Bh: Node number

The Object indicates the preset node number. This value is calculated via the binary value +1 set at the DIP switches (see Chapter 8.1.1 "Setting the node number").

100B	VAR	Node number	Unsigned32	ro	0

Data contents:

Node number in the 1h... 1Fh range (1...31 decimal);

5.1.2.10 Object 100Ch und 100Dh: Monitoring time and life time factor

The objects 100Ch, 100Dh, and 100Eh define the parameters for node guarding (see Chapter 6.2 Node guarding).

The objects in index 100Ch and 100Dh contain the monitoring time in milliseconds and the lifetime factor. The life time factor multiplied by the monitoring time is the life time for the nodeguarding protocol.

100C	VAR	Monitoring time	Unsigned16	rw	0
100D	VAR	Life time factor	Unsigned8	rw	0

Data contents:

Monitoring time 0000...FFFFh [ms]; standard value: = 0h
Life time factor: 00...FFh; standard value = 0h

5.1.2.11 Object 100Eh: COB-ID for node guarding

The objects 100Ch, 100Dh, and 100Eh define the parameters for node guarding (see Chapter 6.2 Node guarding).

The object 100Eh defines the COB ID for the node guarding protocol.

100E	VAR	COB-ID for node guarding	Unsigned32	rw	0

Data contents:

Bit 0...10:	11 Bit Identifier; **Standard-ID = 700h + node number**
Bit 11...29:	0 (reserved for 29 Bit Identifier-device)
Bit 30, 31:	reserved

5.1.2.12 Object 100Fh: Number of supported SDOs

The object contains information on the number of SDOs supported by the device. Byte 0 and 1 contain the number of server SDOs, byte 2 and 3 contain the number of client SDOs.

100F	VAR	Number of supported SDOs	Unsigned32	ro	0

Data contents:

00000001h: 1 server-SDO with 2 directions (SDO (rx) and SDO (tx))

5.1.2.13 Object 1010h: Save parameters

By writing the command "save" in the subindex 1h, the parameters will be saved in the electrically erasable programmable read only memory (EEPROM).The following objects will be saved by means of this command: 1005h, 1014h, 1800h(subindex 1 and 3), 1802h (subindex 1 and 3), 2000h, 2001h, 2005h, 6000h, 6001h, 6002h,6003h, 6200h. Objects which are not subject to the save command have to be sent to the encoder again after each Reset/Power-on!

In order to avoid that data are saved accidentally, the command will only be executed if the string "save" is entered as the code word in this subindex.

The values stored in the EEPROM (Power-on values) will be irretrievably overwritten by means of this command!

A read only access to subindex 1h provides information on the memory functionality.

1010	ARRAY	Save parameters	Unsigned32	rw	0

Data content:

Write access:
Byte 0: 73h (ASCII-Code for "s")
Byte 1: 61h (ASCII-Code for "a")
Byte 2: 76h (ASCII-Code for "v")
Byte 3: 65h (ASCII-Code for "e")

Read only access:

Bit 0 = 1:	Device saves parameters on command
Bit 1 = 0:	Device does not save automatically

Bit 2...31 = 0: reserved

5.1.2.14 Object 1011h: Load standard values

By means of writing the command "load" in the subindex 1h , all parameters within the encoder RAM will be reset to their standard values. In order to avoid that the standard values are loaded by accident, the command will only be executed if the string "load" is entered as the code word in this subindex.

The values stored in the EEPROM (Power-on values) will be irretrievably overwritten by means of this command!

A read only access to **subindex 1h** provides information whether it is possible at all to load the standard values.

1011	ARRAY	Load standard values	Unsigned32	rw	0

Data contents:

Write access:
Byte 0: 6Ch (ASCII-Code for "l")
Byte 1: 6Fh (ASCII-Code for "o")
Byte 2: 61h (ASCII-Code for "a")
Byte 3: 64h (ASCII-Code for "d")

Read only access:

Bit 0 = 1:	Device supports loading of standard values
Bit 1...31 = 0:	reserved

The standard values will only be valid after a "Reset Node"(see Chapter 6.1). If the standard values are also to be copied in the EEPROM, the command "Save Parameters" (see object 1010h) must be executed after the "Reset Node".

5.1.2.15 Object 1014h: COB-ID for emergency objects

The object COB ID for emergency messages.

1014	VAR	COB-ID for emergency objects	Unsigned32	rw	0

Data contents:

Bit 0...10:	11 Bit Identifier; **Standard-ID = 80h + node number**
Bit 11...29:	0 (reserved for 29 Bit Identifier device)
Bit 30, 31:	reserved

5.1.2.16 Object 1018h: Identity Object

The object for reading the Device –Identifikation

1018	RECORD	Device – Identifikation		ro	M

Sub-Index 0h :	only „read“ Delivers constantly the value 4
Sub-Index 1h:	only „read“ Delivers the Vendor-ID (00000008h)
Sub-Index 2h:	Delivers the product code (e.g. 2 for HOHNER CANopen)
Sub-Index 3h:	only „read“ Delivers the SW revision number (e.g.02 00 04 00 → 04. 02)
Sub-Index 4h:	only „read“ Delivers 8-digit serial number of the encoder

5.1.2.17 Object 1800h: PDO1 parameter (asynchronous)

The object contains the parameter for the process data object PDO1. In the standard setting, the process data of the encoder will be read out asynchronously via this service, initiated by the internal device timer (only if the **device timer** was set by means of **object 6200h**), by modification of the process values or by an RTR request. The **PDO-type** will be set in **object 2005h**.

SUB Index 0h: only „read“; delivers constantly the value „3“
SUB-Index 1h: COB ID and release
SUB-Index 2h: transfer mode
SUB-Index 3h: Inhibit time; i.e. minimum waiting time until this PDO can be sent again (Unit : 0,1 ms)
Example: Value 400 is equivalent to 40 ms waiting time

1800	RECORD	PDO1- Parameter		rw	M

Data contents:

Sub-Index 1h:	Bit 0...10:	11 Bit Identifier; **Standard-ID = 180h + node number**
(unsigned32)	Bit 11...29:	0 (reserved for 29 Bit Identifier devices)
	Bit 30:	0 = RTR allowed (unchangeable)
	Bit 31:	0 (PDO enabled), 1 (PDO disabled); **standard value = 0**

Sub-Index 2h: **Standard value = FEh (254)** (transfer mode = asynchronous, controlled by (unsigned8) manufacturer specific event)

Sub-Index 3h: **Standard value = 0h** (no Inhibit time)
(unsigned16)

Instructions on Sub index 1h:
PDO is released by standard via 31 bit (0=enabled). This setting can be modified, but has to be transferred again after each activation (Reset or Power-on) of the encoder, since this bit cannot be stored in the EEPROM.

Instruction on Sub index 2h:
The standard value FEh may be modified, but must be retransferred after each activation of the encoder (Reset or Power-on), since this bit cannot be stored in the EEPROM.

Instruction on Sub index 3h:
The standard value FEh can be modified. Save to a non volatile storage [EEPROM] via Object 1010.

Table of the transfer modes:

Code (dec.)	Transfer mode				
	Cyclical	Acyclical	Synchronous	Asynchronous	RTR only
0		X	X		
1-240	X		X		
241-251	reserved				
252			X		X
253				X	X
254				X	
255				X	

<u>Meaning of the transfer mode code:</u>

0:	After SYNC, but only when modifying value since the last SYNC
1 ... 240:	Send value after 1. ... 240. SYNC
252:	SYNC leads to an internal storage of the value; value must be get via RTR
253:	After RTR the value is updated and send
254:	After modifying value (device timer = 0) or after cycle time is up (device timer ≠ 0) the value is updated and send.

Remarks on the transfer mode 0 ... 253:
If requested by RTR or SYNC respectively, the PDO will be sent after the current calculation (about 800 µs) has been finished, which leads to a "waiting time" amounting to 0.2...1 ms. The encoder disk will be read out timer-controlled every millisecond, the values will be calculated afterwards.

If the **transfer mode 254** is used for the PDO (asynchronous event-controlled):The selected cycle time (see object 6200h) has to be larger than the bus transfer period, in order to allow for an interference-free transmission of the PDOs!
Baud rate 10 kBaud: cycle time at least 14 ms
Baud rate 20 kBaud: cycle time at least 10 ms
Baud rate 50 kBaud: cycle time at least 4 ms For a cycle time=0 (i.e. . PDO in case of modification of value) the baud rate has to amount to at least 125 kBaud.

When the position transfer is controlled by asynchronous cycle time modified positions can also be transferred at once and not only after the cycle time is up.

Position transfer	Cycle time-Value (Object 6200H)	Inhibit-time (Object 1800, Sub-Index 2)	Function
Asynchronous	0	0	Max. output frequency when modifying values
Asynchronous	0	>0	Output when modifying values and closing waiting time (Inhibit time)
Cycle time	>0	0	Cycled time output
Asynchronous / Cycle time	>0	>0	Combination of Asynchronous and Cycle time

5.1.2.18 Object 1802h: PDO2 parameter (synchronous, cyclical)

The object contains the parameters for the process data object PDO2. In the standard setting, the process data of the encoder will be read out synchronously by means of this service, initiated by RTR and SYNC objects. Subindex 1h: COB ID and release Subindex 2h: Transfer mode (Table see object 1800h) Subindex 3h: Inhibit time (Minimum waiting time until this PDO can be sent again)

1802	RECORD	PDO2 parameter		rw	M

Data contents:

Subindex 1h:	Bit 0...10:	11 Bit identifier; **standard-ID = 280h + node number**
(unsigned32)	Bit 11...29:	0 (reserved for 29 Bit identifier devices)
	Bit 30:	0 = RTR allows (unchangeable)
	Bit 31:	0 (PDO enabled), 1 (PDO disabled); **standard value = 1**
Subindex 2h:	**standard value = 1h** (Transfer mode = synchronous, cyclical)	
(unsigned8)	(Table see object 1800h)	
Subindex 3h:	**standard value = 0h** no inhibit time)	
(unsigned16)		

Instruction on subindex 1h:
PDO is disabled by standard via bit 31 (1=disabled). This setting can be modified, but has to be transferred again after each activation (Reset or Power-on) of the encoder, because this value cannot be stored in the EEPROM.

Instruction on subindex 2h:
The standard value 01h can be modified, but must be retransferred after each activation of the encoder (Reset or Power-on), because this value cannot be stored in the EEPROM.

Instruction on subindex 3h:
The standard value 0h can be modified, but must be retransferred after each activation of the encoder (Reset or Power-on), because this value cannot be stored in the EEPROM.

5.1.3 Detailed Description of the Manufacturer Specific Encoder Parameters

5.1.3.1 Object 2000h: Warning positions

Up to 4 warning positions may be programmed. According to the programming, a message can be initiated via a type 1 PDO (position value + signal flags) in case the warning position has been exceeded/undershot (see object 2005h, PDO-Type).

The status of the **signal flags** can also be inquired via **object 2004h**.

The warning positions are stored in the sub indices 1, 2, 3 and 4.

Subindex 0 only „read" (delivers always the value 4).

2000	ARRAY	Warning position 1...4	Unsigned32	rw	0

Data contents:

Subindex 1...4:

Bit 0...29:	Warning position in the value range 0 ... programmed total resolution; **standard value = 0h**
Bit 30:	Activation if the warning position is undershot; 1=active, 0=inactive; **standard value = 0**
Bit 31:	Activation if the warning position is exceeded; 1=active, 0=inactive; **standard value = 0**

5.1.3.2 Object 2001h: Offset value

The offset allows to shift the scaled value range.

The offset value is added to the scaled position value within the encoder. The position value is thus shifted up or down by the recorded offset value.

2001	VAR	Offset-Wert	Signed32	rw	0

Data contents:

Standard value = 0h

5.1.3.3 Object 2002h: Speed

To measure the rotational speed of the encoder shaft, the difference between two physical (unscaled) values of position with a time period of 5 ms is calculated. The difference between the two values will be read out as a signed 16 bit value (positive value = clockwise direction of rotation if a read only access with object 2002 h has been granted).

The output of the speed by means of PDO is made possible by setting the desired PDOs to type 2 (see object 2005h).

2002	VAR	Speed	Signed16	rw	0

5.1.3.4 Object 2003h: Acceleration

To measure the acceleration of the encoder shaft, the difference between two physical (unscaled) values of position with a time period of 5 ms is calculated. The difference between the two values will be read out as a signed 16 bit value (positive value = clockwise direction of rotation if a read only access with object 2002 h has been granted).

The output of the acceleration by means of PDO is made possible by setting the desired PDOs at type 2 (see object 2005h).

2003	VAR	Acceleration	Signed16	rw	0

5.1.3.5 Object 2004h: Signal flags of the warning positions

The status of the signal flags for the warning positions (see object 200h) can be inquired in this object.

The output of the signal flags by means of PDO is made possible by setting the desired PDOs to type 1 (see object 2005h).

2004	VAR	Signal flags of the warning positions	unsigned8	ro	0

Data contents:

Bit 0: warning position 1 exceeded
Bit 1: warning position 1 exceeded
Bit 2: warning position 2 exceeded
Bit 3: warning position 2 exceeded
Bit 4: warning position 3 exceeded
Bit 5: warning position 3 exceeded
Bit 6: warning position 4 exceeded
Bit 7: warning position 4 exceeded

The respective bits will be set to 1, if the programmed warning position is exceeded or undershot.

5.1.3.6 Object 2005h: PDO Type

This object helps to determine the types for PDO1 and PDO2.

The COB ID and the transfer type for the PDOs is determined in the objects 1800h and1802h.

PDO1 is by standard released via **Bit 31 in the 1800h** object, subindex 1h.
PDO2 is by standard disabled via **Bit 31 in the 1802h** object, subindex 1h.

This setting can be modified, but has to be transferred again after each activation (Reset or Power-on) of the encoder, because this value cannot be stored in the EEPROM.

2005	VAR	PDO-Typ	unsigned8	rw	0

Data content:

	Bit 7 ... 4 = PDO2	**Bit 3 ... 0 = PDO1**
00h	Default	Default
01h	Default	Type1
02h	Default	Type2
10h	Type1	Default
11h	Type1	Type1
12h	Type1	Type2
20h	Type2	Default
21h	Type2	Type1
22h	Type2	Type2

Standard: PDO1 and PDO2 set to type Default (00h)

Meaning:

COB-ID	**Process data in binary code**							
11 Bit	Byte 0 $2^7 \ldots 2^0$	Byte 1 $2^{15} \ldots 2^8$	Byte 2 $2^{23} \ldots 2^{16}$	Byte 3 $2^{31} \ldots 2^{24}$	Byte 4	Byte 5	Byte 6	Byte 7
Default:	**Value of position**							
Type1:	**Value of position**				**Flags**[1)]			
Type2:	**Value of position**				**Speed**		**Acceleration**	

[1)] Signal flags of the warning positions, see also object directory index 2004h.If the transfer mode 254 (asynchronous event-controlled; see table at object 1800h) is used and the device timer=0 , the PDO will be sent when the flag is being modified.

5.1.4 Detailed Description of the General Encoder Parameters

5.1.4.1 Object 6000h: Operating parameters

Activation of: revert code sequence, diagnosis request, scaling function.

The preset value (see object 6003h) will be deleted, if either the code sequence is modified or the scaling function activated.

6000	VAR	Operating parameters	Unsigned16	rw	M/O

Data contents:

Bit 0: Code sequence;	0 = ascending in case of clockwise rotation (cw) 1 = ascending in case of counterclockwise rotation (ccw) **Standard: Bit = 0**

Bit 1: Hardware check; 0 = disable, 1 = enable; **Standard: Bit = 0** (see Object 6503)
Bit 2: Scaling function; 0 = disable, 1 = enable; **Standard: Bit = 0** (s. Object 6001,6002)
Bit 12: residual value compensation; 0 = disable, 1 = enable; **Standard: Bit = 0** (s. Object 6002);
A bit modification will only take effect after an encoder reset.
Bit 3...11, 13..15: not used (0)

5.1.4.2 Object 6001h: Measuring steps per revolution (resolution)

This parameter presents the desired resolution per revolution. The encoder calculates the corresponding scaling factor internally.

The total number of measuring steps is presented via object 6002h.

In case the resolution is being modified, a possibly programmed preset value (see object 6003h) will be deleted.

The resulting scaling factor SCF (which is multiplied by the physical position value) is calculated according to the following formula:

Errore. Il s

$$\mathbf{SCF} = \frac{\text{Measuring steps per revolution (6001h)}}{\text{phys. resolution single turn (6501h)}}$$

6001	VAR	Measuring steps per revolution	Unsigned32	rw	M

Data contents:

Byte 0	Byte 1	Byte 2	Byte 3
$2^7 \dots 2^0$	$2^{15} \dots 2^8$	$2^{23} \dots 2^{16}$	$2^{31} \dots 2^{24}$

Value range: 0 ... (max. phys. resolution per revolution)

Standard value = physical resolution per revolution

Example with standard value:
Encoder/1213 (resolution = 13 Bit per revolution):

Data contents = 20 00h

Byte 0	Byte 1	Byte 2	Byte 3
00	20h	00h	00h

5.1.4.3 Object 6002h: Total number of measuring steps

This parameter sets the total number of measuring steps. After having executed the given total number of measuring steps, the encoder is reset to zero.

The measuring steps per revolution (resolution) are set via object 6001h.

If the total number of measuring steps is being modified, a possibly programmed preset value will be deleted.

With operation without residual value compensation (Bit 12 of Object 6000h = 0):
If the encoder is used in continuous operation, the total number of measuring steps must not exceed only 2^x • "measuring steps per revolution (6001h)" (with x = 1 ... 12). Otherwise, there will always be a skip in the output code when the code disc) cross the physical zero point (with single-turn encoders after each turn, with multi-turn after 4096 turns).

With operation with residual value compensation (Bit 12 of Object 6000h = 1):
Any value may be entered, since, in this mode, the residual value is automatically stored to the encoder's EEPROM with every physical zero crossing (see above).

Totally, there are workable 1,000,000 EEPROM write cycles!

6002	VAR	Total number of measuring steps	Unsigned32	rw	M

Data contents:

Byte 0	Byte 1	Byte 2	Byte 3
$2^7 \ldots 2^0$	$2^{15} \ldots 2^8$	$2^{23} \ldots 2^{16}$	$2^{31} \ldots 2^{24}$

Value range: 0 ... (max. total resolution)

Standard value = total resolution

Example with standard value:
Encoder/1213 (total resolution = 13 Bit per revolution • 12 Bit revolutions):

Data content = 2 00 00 00h

Byte 0	Byte 1	Byte 2	Byte 3
00h	00h	00h	02h

5.1.4.4 Object 6003h: Preset value

The position value of the encoder is set to the above-mentioned preset value. In doing so, e.g. the zero position of the encoder can be aligned with the zero point of the machine.

The preset value will be deleted if the code sequence is modified or the scaling inactivated or modified (see objects 6000h, 6001h, and 6002h). In the encoder, the preset value will be converted into a respective offset value and will be added to the position value (Offset = Preset - Position). This offset value can be read via object 6509h.

6003	VAR	Preset value	Unsigned32	rw	M

Data contents:

Byte 0	Byte 1	Byte 2	Byte 3
$2^7 \dots 2^0$	$2^{15} \dots 2^8$	$2^{23} \dots 2^{16}$	$2^{31} \dots 2^{24}$

Value range: 0 ... (programmed total resolution)

Standard value = 0

By writing the value FFFF FFFFh, the preset value will be deleted!

5.1.4.5 Object 6004h: Value of position

The current position value (settled up against the scaling factor, preset, and offset) is read out by the encoder.

6004	VAR	Value of position	Unsigned32	ro	M

Data contents:

Byte 0	Byte 1	Byte 2	Byte 3
$2^7 \dots 2^0$	$2^{15} \dots 2^8$	$2^{23} \dots 2^{16}$	$2^{31} \dots 2^{24}$

5.1.4.6 Object 6200h: Cyclic timer

Defines the cycle time by means of which the current position will be read out via PDO1 (see object 1800h). The timer controlled output will become active as soon as a cycle time > 0 is entered.

6200	VAR	Cyclic timer	Unsigned16	rw	M

Data contents:

Value range: 0 ... FFFFh (65535) is the cycle time in milliseconds

Standard value = 0h

If the transfer mode 254 is used (asynchronous event-controlled, see object 1800h):
The selected cycle time has to be longer than the bus transfer period, in order to allow for an interference-free transmission of the PDOs!
Baud rate 10 kBaud: cycle time at least 14 ms
Baud rate 20 kBaud: cycle time at least 10 ms
Baud rate 50 kBaud: cycle time at least 4 ms
For a cycle time=0 (i.e. . PDO in case of modification of value), the baud rate has to amount to at least 125 kBaud.

5.1.4.7 Object 6500h: Indicate operating status

The object shows the settings programmed via the object 6000h.

6500	VAR	Operating status	Unsigned16	ro	M

Data contents:

see Object 6000h.

5.1.4.8 Object 6501h: phys. resolution single turn

The object shows the physical resolution per revolution (number of position values on the single turn code disc). The value depends on the encoder type.

6501	VAR	physical resolution single turn	Unsigned32	ro	M

Data contents:

Byte 0	Byte 1	Byte 2	Byte 3
$2^7 \dots 2^0$	$2^{15} \dots 2^8$	$2^{23} \dots 2^{16}$	$2^{31} \dots 2^{24}$

Example:

Encoder/1213 (resolution = 13 Bit per revolution): Data contents = 20 00h

Byte 0	Byte 1	Byte 2	Byte 3
00	20h	00h	00h

5.1.4.9 Object 6502h: Number of revolutions

The object displays the number of revolutions which can be received by the multi turn encoder. The value depends on the encoder type.

6502	VAR	Number of revolutions	Unsigned16	ro	M

Data contents:

Byte 0	Byte 1
$2^7 \dots 2^0$	$2^{15} \dots 2^8$

Example:

Encoder/1213 (12 bit for revolutions = 4096): Data contents = 10 00h

Byte 0	Byte 1
00	10h

5.1.4.10 Object 6503h: Alarm message

In addition to those errors signaled by emergency messages, the object 6503h offers other error messages. The corresponding error bit is set to 1, unless the error does not exist anymore.

6503	VAR	Alarm message	Unsigned16	ro	M

Data contents:

Bit 0:	position error; 0 = valid position value, 1 = position error
Bit 1:	hardware check; 0 = no error, 1 = error
Bit 2...15:	unsigned

In both cases, an emergency message (ID=80h+node number) with the error code 1000h (generic error) is sent simultaneously when an alarm occurs.

5.1.4.11 Object 6504h: Supported alarm messages

By means of this object, the alarm messages supported by the encoder are displayed (see object 6503h).

6504	VAR	Supported alarm messages	Unsigned16	ro	M

Data contents:

Bit 0:	1 = position error is supported
Bit 1:	1 = hardware check is supported
Bit 2...15:	unsigned

5.1.4.12 Object 6505h: Warning message

Warning messages indicate that the tolerances of the internal encoder parameters have been exceeded. In contrast to an alarm or emergency message, the measured value can nevertheless be valid. The corresponding warning bit is set to 1, while the tolerance is still exceeded.

6505	VAR	Warning message	Unsigned16	ro	M

Data contents:

Bit 0:	exceeding of revolution; 0 = none, 1 = exceeding of 10.000 r/min
Bit 1:	unsigned
Bit 2:	CPU watchdog status; 0 = all right, 1 = reset executed
Bit 3:	operating time; 0 = not exceeded, 1 = 100.000 hours exceeded
Bit 4...15:	unsigned

If the 0 bit is active, an emergency message (ID=80h+node number) with the error codeFF00h (Device specific) is sent simultaneously.
If the bits 2 or 3 are active, an emergency message (ID=80h+node number) with the error code 5000h (device hardware) is sent simultaneously.

5.1.4.13 Object 6506h: Supported warning messages

By means of this object is indicated which warning messages are supported by the encoder (see Object 6505h).

6506	VAR	Supported warning messages	Unsigned16	ro	M

Data contents:

Bit 0:	1 = exceeding of rotational speed is supported
Bit 1:	unsigned
Bit 2:	1 = CPU watchdog status is supported
Bit 3:	1 = Operating time warning is supported
Bit 4...15:	unsigned

5.1.4.14 Object 6507h: Profile and software version

The version number of the encoder profile used here is stored in the first 16 bit. The second16 bit contain the number of the software version implemented in the encoder.

6507	VAR	Profile and software version	Unsigned32	ro	M

Data contents:

Profile version		Software version	
Byte 0	Byte 1	Byte 2	Byte 3
$2^7 \dots 2^0$	$2^{15} \dots 2^8$	$2^{23} \dots 2^{16}$	$2^{31} \dots 2^{24}$

Example:

Profil version 1.0 and software version 1.1:

Byte 0	Byte 1	Byte 2	Byte 3
00	01h	10h	01h

5.1.4.15 Object 6508h: Operating time counter

While the encoder supply voltage is on, the operating time counter is incremented every 6 minutes (= 0.1 hours).

Note: After an operating time of 100.000 hours, the bit no. 3 is set in the object 6505h (warning message).

6508	VAR	Operating time counter	Unsigned32	ro	M

Data contents:

Operating time in 1/10 hours as binary 32 bit value.

5.1.4.16 Object 6509h: Offset value

A preset value entered via object 6003h is converted internally into the corresponding offset value.

The object 6509h indicates the calculated offset value.

6509	VAR	Offset value	Signed32	ro	M

5.1.4.17 Object 650Ah: Manufacturer offset

Contains the offset value determined via the object 2001h

650A	VAR	Manufacturer offset	Signed32	ro	M

5.1.4.18 Object 650Bh: Serial number

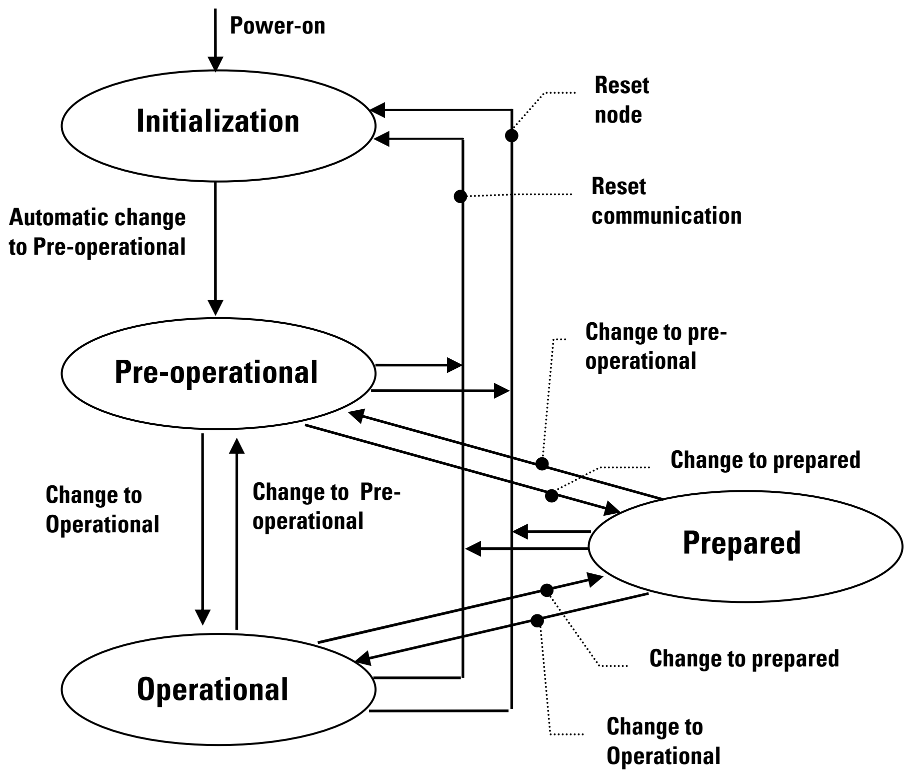
This object provides the serial number of the encoder.

650B	VAR	Serial number	Unsigned32	ro	M

6 Network Management

The encoder supports the simplified network management (minimum boot up) defined in the profile for "minimum capability devices".

The following phase diagram according to DS 301 shows the different node conditions andthe respective network commands (controlled by the network master via NMT-services):



Initialization: Initial state after the supply voltage is on. The node automatically changes after the execution of the reset /initialization routines to the status pre-optional.

Pre-operational: Now the node can be operated via SDO messages under the standard identifier and can thus be parameterized by means of its object directory, e.g. programming of the encoder and communication parameters.

Operational: The node is active. Process data is read out via the PDOs

Prepared: In this state, the node is no longer active, i.e. neither a SDO nor a PDO communication is possible. The node can be set either to the state operational or pre-operational by means of a NMT command.

6.1 Description of the NMT Commands

The commands are transferred as an unconfirmed NMT object.
The NMT object is structured as follows:

	Byte 0	Byte 1
COB-ID = 0	**Command byte**	**Node number**

The COB ID of the NMT object is always 0!

The node is addressed via the node number. Node number 0 operates all nodes..

The following table gives a short overview of the commands used:

Command byte (hex)	**Description**
01h	Start_Remote_Node: Change to Operational
02h	Stop_Remote_Node: Change to Prepared
80h	Enter_Pre-Operational_State: Change to Pre-operational
81h	Reset_Node: Reset node[1]
82h	Reset_Communication: Reset communication[2]

[1] All parameters of the entire object directory are set to Power-on values.

[2] Only the parameters in the section communication profile of the object directory are set to Power-on values.

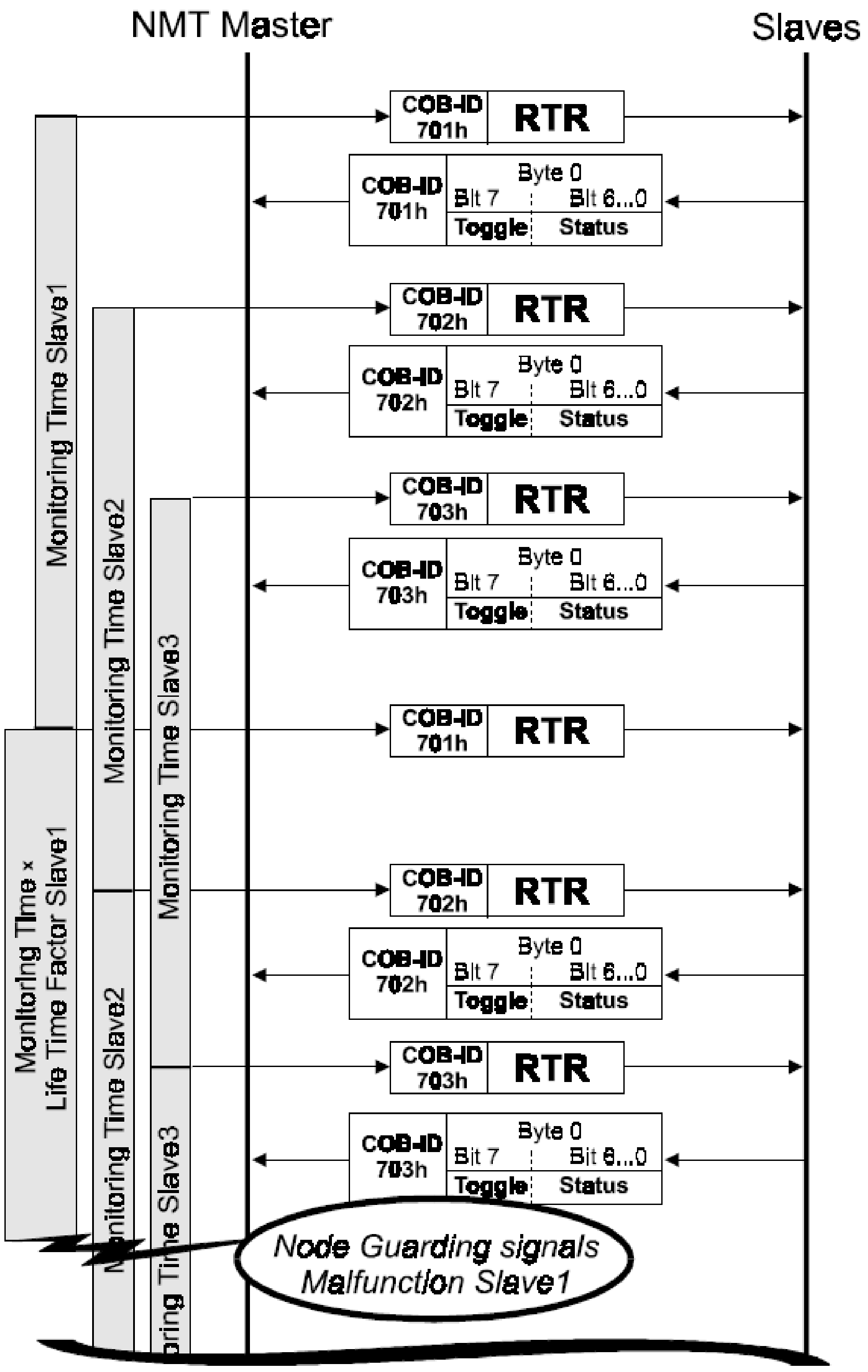
6.2 Node guarding

6.2.1 General

Node guarding is used to control the communication capability of the stations.
The node guard should always be activated if the station sends only irregularly (event -controlled) on the bus. If the station's data is regularly requested via RTRs anyway, an additional node guard is not necessary.
If an error is detected by the node guard, the incorrect node should be set to a safe state. Depending on the application this can usually be the status prepared.
In case of active node guarding , the network master regularly (monitoring time) sends an RTR message with the COB ID for node guarding at the nodes. The node has to respond by issuing a status message within its life time period agreed on with the network master.

The COB identifier for node guarding is determined by object 100Eh (**standard ID =700h+node number**). The **monitoring time (100Ch)** multiplied by the **life time factor (100Dh)** is the **life time period** for the node guarding protocol.

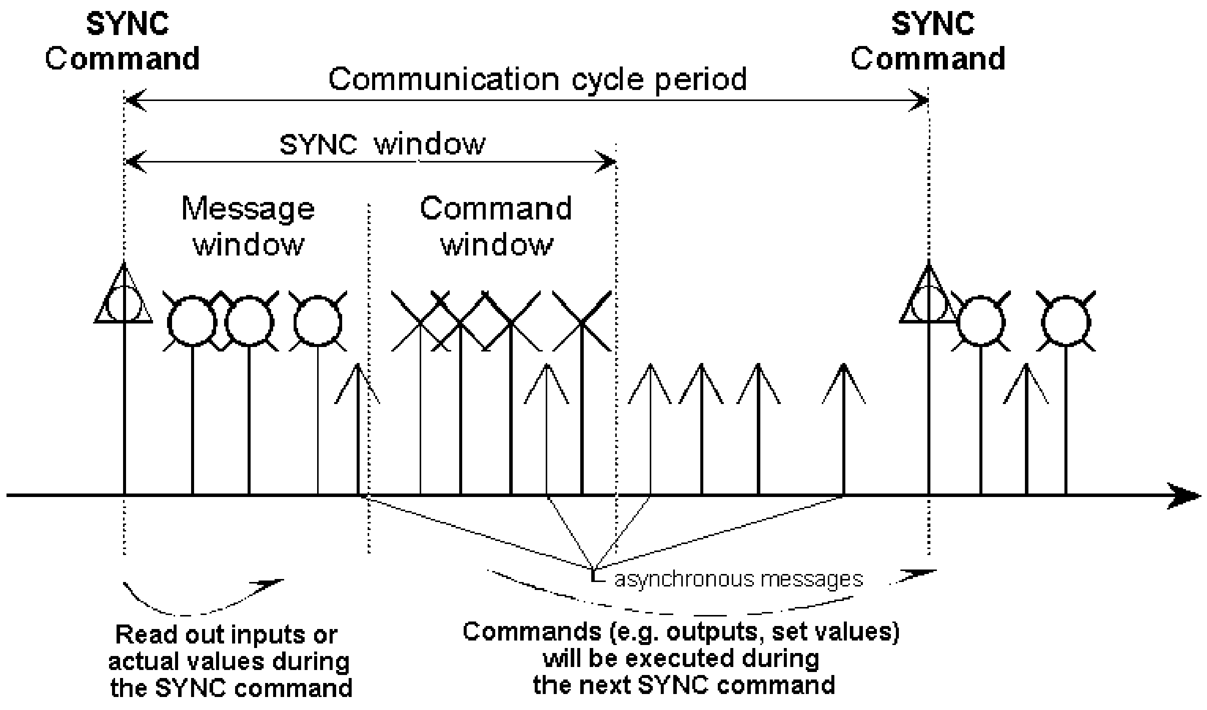
6.2.2 Example: Node guarding with the node numbers 1, 2, and 3


NMT Master
Slaves
COB-ID 701h RTR
COB-ID 701h Byte 0 Bit 7 Bit 6...0 Toggle Status
COB-ID 702h RTR
COB-ID 702h Byte 0 Bit 7 Bit 6...0 Toggle Status
COB-ID 703h RTR
COB-ID 703h Byte 0 Bit 7 Bit 6...0 Toggle Status
COB-ID 701h RTR
COB-ID 702h RTR
COB-ID 702h Byte 0 Bit 7 Bit 6...0 Toggle Status
COB-ID 703h RTR
COB-ID 703h Byte 0 Bit 7 Bit 6...0 Toggle Status
Monitoring Time Slave1
Monitoring Time Slave2
Monitoring Time Slave3
Monitoring Time x Life Time Factor Slave1
Node Guarding signals Malfunction Slave1

6.2.3 Malfunction detection of the node guarding protocol:

Failure	Reason
Acknowledge message with unchanged toggle bit	The application process of the slave is possibly no longer active or too busy at the moment
The acknowledged status of the node does not comply with the status expected by the NMT master.	e.g. short voltage drop
The slave does no longer respond.	▪ A malfunction or deactivation of the node ▪ Failure of the CAN Bus
The slave does no longer receive control RTRs from the NMT master	failure of the NMT master

6.3 Bus Synchronization


SYNC Command
Communication cycle period
SYNC Command
SYNC window
Message window
Command window
asynchronous messages
Read out inputs or actual values during the SYNC command
Commands (e.g. outputs, set values) will be executed during the next SYNC command

The identifier for the SYNC message is determined in object 1005h. Standard value = 80h

Many real-time applications require not only a cyclical data transfer but also a synchronization between several bus nodes. The axes of a kinematics have to be synchronized or I/O modules in SPS have to set outputs and read inputs at the same time. Synchronized drives expect set values and send actual values in predetermined windows of time. The CANopen communication profile meets these requirements by means of (optional) high priority synchronization messages, which divide the time axis in equal communication cycles.

The SYNC messages contain no data at all and can be used by I/O modules as interrupt in order to read out inputs or set outputs respectively. Intelligent devices such as drives are consequently able to synchronize themselves by means of the PLL procedure.

In the **message window** following the SYNC message, the nodes transmit their actual values/input values; in the **command window**, set and other output values are being transferred, which will become valid for the next SYNC message. The above-mentioned windows are marked by different message priorities. Since the message window follows the SYNC command, it is even observed by simple components without timer. Low priority SDO messages can use the band width within the windows which is not being used for other purposes. Otherwise, they can use the period between command message and SYNC message.

7 Bus Connection

7.1 General requirements for bus connection of the encoder

The continuous CAN bus has to be terminated at both ends with a bus termination resistance of 120 Ohm between CAN+ and CAN-.

The screen has to be connected on both side with ground. The power supply lines should also be entirely screened. If this is impossible, appropriate filter methods should be applied.

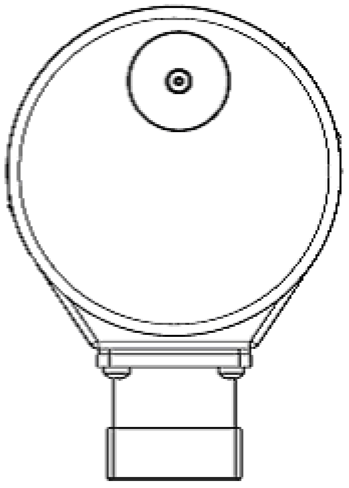
For bus resp. connecting cables, only use signal cables with twisted wire pairs (CAN+ with CAN-, +UB with 0 V) with screen.

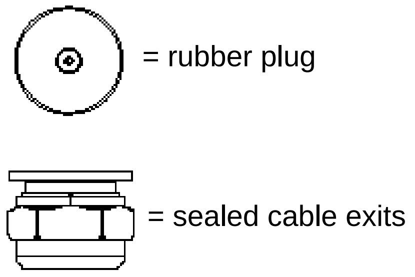
7.2 Connection types

There are four connection types available for CANopen encoder. The following chapters are describing how to connect each connection type.

Connection type	Page
Cable	46/47
Conin	48/49
Bus cover with double conin	50/51
Bus cover with 3 sealed cable exits	52/53

7.2.1 Cable





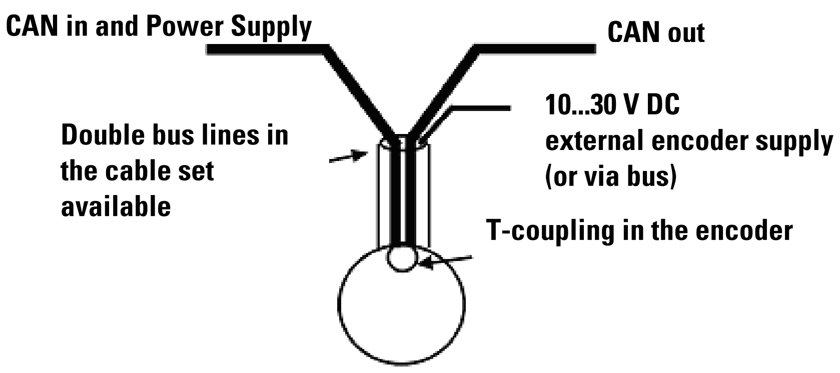
⚠ **Do not use a spur line!**

⇒ Use a double bus line

⇒ Connect the bus line with cable pair 1 (CAN in + and CAN in -), blue (CAN GND in) as well as with pair 3 (UB in, 0 V in) of the TPE cable (pin assignment, next page).

<u>In case there are other devices following within the same bus phase:</u>

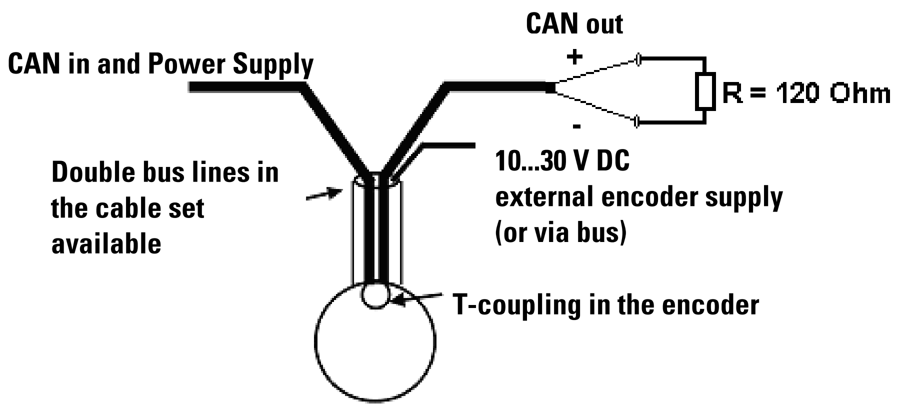
⇒ Connect the continuing cable with pair 2 (CAN out +, CAN out -) as well as brown (CAN GND out) (pin assignment, next page) .



In case there are no other devices following within the same bus phase:

⚠ **External bus termination resistor necessary!**

⇒ Connect a bus termination resistor of **120 Ohm** between CAN out + and CAN out - at the end of the bus phase!

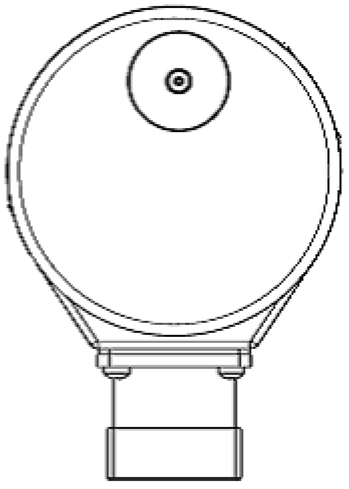


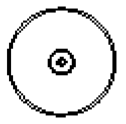
Pin assignment:

Cable pair	TPE cable	Signal	Description
Pair 1	Yellow	CAN in+	CAN in+ bus line (dominant H)
	Green	CAN in -	CAN in- bus line (dominant L)
Pair 2	Pink	CAN out +	CAN in+ bus line (dominant H)
	Grey	CAN out -	CAN in- bus line (dominant L)
	Blue	CAN GND in	Ground (0V potential for CAN Bus)
	Brown	CAN GND out	Ground (0V potential for CAN Bus)
Pair 3	White	UB in	Encoder power supply
	Brown	0V in	Ground (0V for encoder power supply)
	Screen	Screen	Screen connected with encoder housing

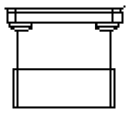
Note: Use paired signal cables with screen, place screen on both sides!

7.2.2 Conin, axial





= Rubber plug



= Conin, 12 pol.

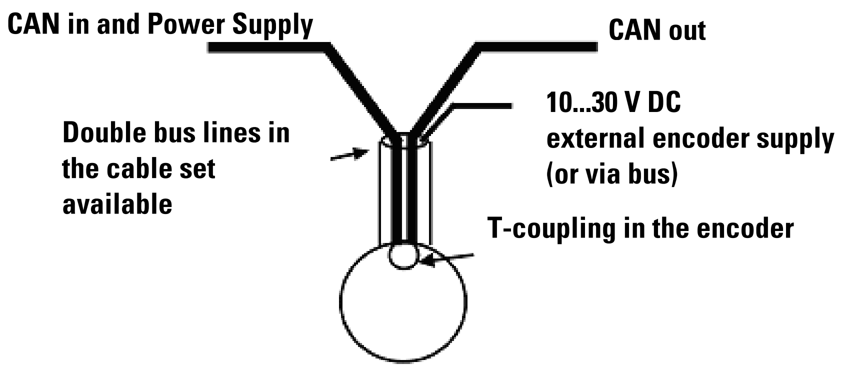
Do not use a spur line!

⇒ Use a double bus line

⇒ Connect the bus line with PIN 7 (CAN in +), PIN 2 (CAN in -), PIN 3 (CAN GND in), PIN 12 (UB in) and PIN 10 (0 V in) (pin assignment, next page).

<u>In case there are other devices following within the same bus phase:</u>

⇒ Connect the continuing bus line with PIN 4 (CAN out +), PIN 5 (CAN out -) and with PIN 11 (CAN GND out) (pin assignment, next page) .

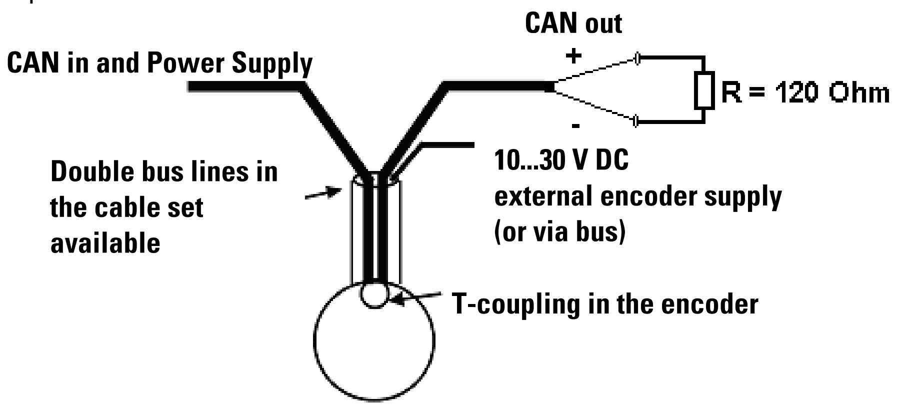


In case there are no other devices following within the same bus phase:

Possibility A: Bus line is carried out of the encoder!

⚠ **External bus termination resistor necessary!**

⇒ Connect a bus termination resistor of **120 Ohm** between CAN out + and CAN out - at the end of the bus phase!


CAN out
CAN in and Power Supply
+
-
R = 120 Ohm
Double bus lines in the cable set available
10...30 V DC external encoder supply (or via bus)
T-coupling in the encoder

Possibility B: Bus line ends in the encoder!

⇒ Draw off the rubber plug from the encoder.
Now you have a free view on the DIP switch and LED display (see 8.3).
⇒ Activate the bus termination resistor (DIP switch 9 and10 to „ON“)!

⚠ **Warning!**

Encoder can become leak, if rubber plug is inserted wrong or is damaged

- IP Protection is not guaranteed!
- A total failure of the encoder is possible!

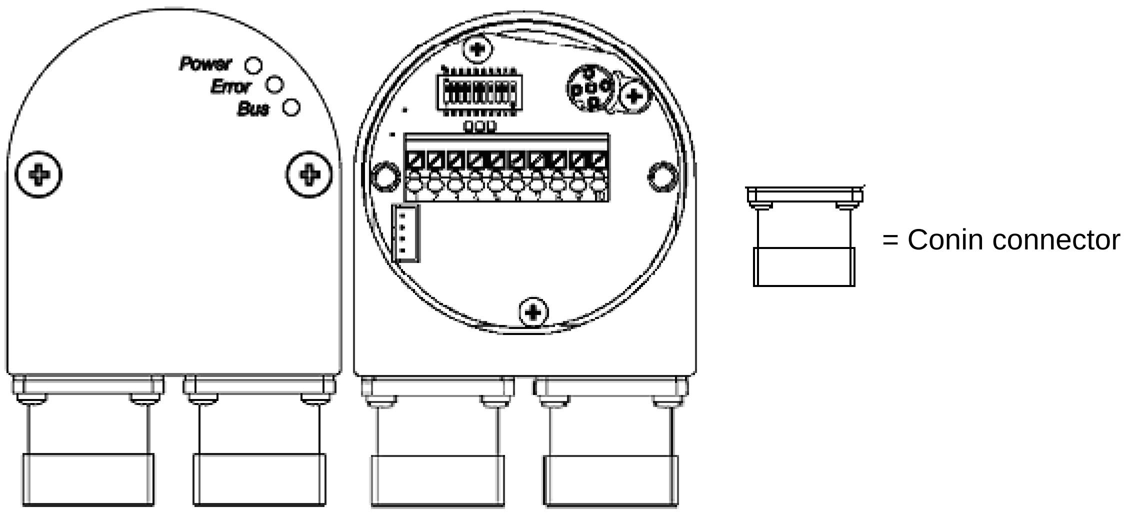
⇒ Pay attention when removing rubber plug!
⇒ push the rubber plug several times for a tight fit in the bore
⇒ damaged rubber plugs have to be replaced (Art. no. 2565007)

PIN assignment:

Conin-PIN	Signal	Description
7	CAN in+	CAN + bus line (dominant H)
2	CAN in -	CAN - bus line (dominant L)
4	CAN out +	CAN + bus line (dominant H)
5	CAN out -	CAN - bus line (dominant L)
3	CAN GND in	Ground (0V potential for CAN Bus)
11	CAN GND out	Ground (0V potential for CAN Bus)
12	UB in	Encoder power supply
10	0V in	Ground (0V for encoder power supply)
Screen	Screen	Screen connected with encoder housing

Note: Use paired signal cables with screen, place screen on both sides!

7.2.3 Bus cover with double conin



⇒ Connect the bus line with the encoder input (pins).

<u>In case there are other devices following within the same bus phase:</u>
⇒ Connect the leading bus line with the encoder output (liner).

⚠ **Maximum current of 2 A via Pin 7 and 8 !**
Recommended external fuse for the entire bus supply voltage: T 2 A

<u>In case there are no other devices following within the same bus phase:</u>
⇒ Loosen the screws and draw off the bus cover from the encoder.
⇒ **View: into the open bus cover**
⇒ Activate the bus termination resistor in the bus cover (DIP switch 9 and 10 to „ON")!
⇒ Attach the bus cover to the encoder and tighten the screws.

Connection assignment:

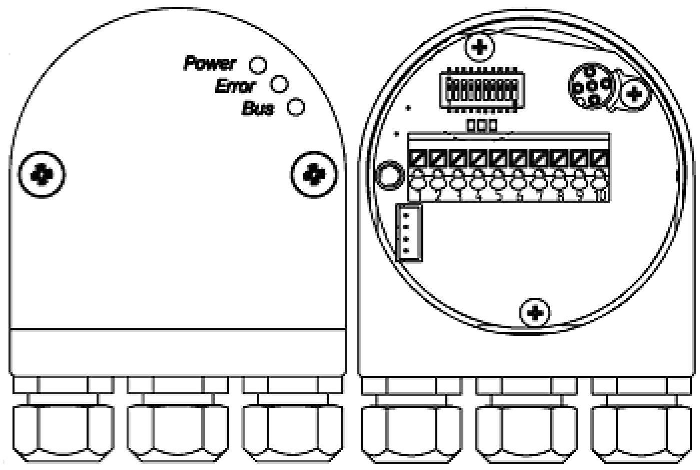
Conin-PIN	Bus cover with		Description
	Pin inserted (IN)	Socket inserted (OUT)	
1	CAN in +	CAN out +	CAN+ Bus line (dominant H)
2	CAN in –	CAN out -	CAN – Bus line (dominant L)
3	CAN GND in	CAN GND out	Ground (0V potential for CAN Bus)
4	N.C.	N.C.	Not connected
5	N.C.	N.C.	Not connected
6	N.C.	N.C.	Not connected
7	UB in	UB out	Encoder power supply
8	0V in	0V out	Ground (0V for encoder power supply)
9	N.C.	N.C.	Not connected
Screen	Screen	Screen	Screen connected to encoder housing

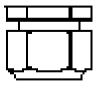
Note: Use paired signal cables with screen, place screen on both sides!

7.2.4 Bus cover with 3 sealed cable exits

There are two possibilities to connect the encoder

- **Possibility A: Connection with power supply in data cable**
- **Possibility B: Connection with power supply in a separate cable**



 = Sealed cable exits

<u>Possibility A: Connection with power supply in data cable</u>

⇒ Loosen the screws and draw off the bus cover from the encoder

⇒ **View: into the open bus cover**

⇒ ***Replace middle sealed cable exit by screw plug to ensure the encoder is sealed.***

⇒ Lead the encoder voltage supply and data cable through the left screw connection and connect it to terminal 1 (UB in), terminal 2 (0V in), terminal 3 (CAN in -), terminal 4 (CAN in +) and terminal 5 (CAN GND in) (see connection diagram, page 53. Put the cable screen on the sealed cable exits (see cable connection diagram, page 53).

<u>In case there are no other devices following within the same bus phase:</u>

⇒ Activate the bus termination resistor in the bus cover (DIP switch 9 and 10 from S1 to „ON“)!

⇒ ***Replace right cable connection by screw plug to ensure the encoder is sealed.***

<u>In case there are other devices following within the same bus phase:</u>

⇒ Lead the continuing cable through the right screw connection and connect it to the terminal 6 (CAN GND out), terminal 7 (CAN out +), terminal 8 (CAN out-), terminal 9 (0V out) and terminal 10 (UB out) according to connection diagram (page 53). Put the cable screen on the cable screw connection (see cable connection diagram, page 53).

⇒ Attach the bus cover to the encoder and tighten the screws.

Possibility B: Connection with power supply in a separate cable

⇒ Loosen the screws and draw off the bus cover from the encoder

⇒ **View: into the open bus cover**

⇒ Lead the encoder voltage supply through the middle screw connection and connect it to terminal 1 (UB in) and terminal 2 (0V in) (see connection diagram, page 53). Put the cable screen on the sealed cable exits (see cable connection diagram, page 53).

⇒ Lead the bus cable through the left screw connection and connect it to terminal 3 (CAN in -), terminal 4 (CAN in +) and terminal 5 (CAN GND in) (see connection diagram, page 53). Put the cable screen on the sealed cable exits (see cable connection diagram, page 53).

In case there are no other devices following within the same bus phase:

⇒ Activate the bus termination resistor in the bus cover (DIP switch 9 and 10 to „ON“)!

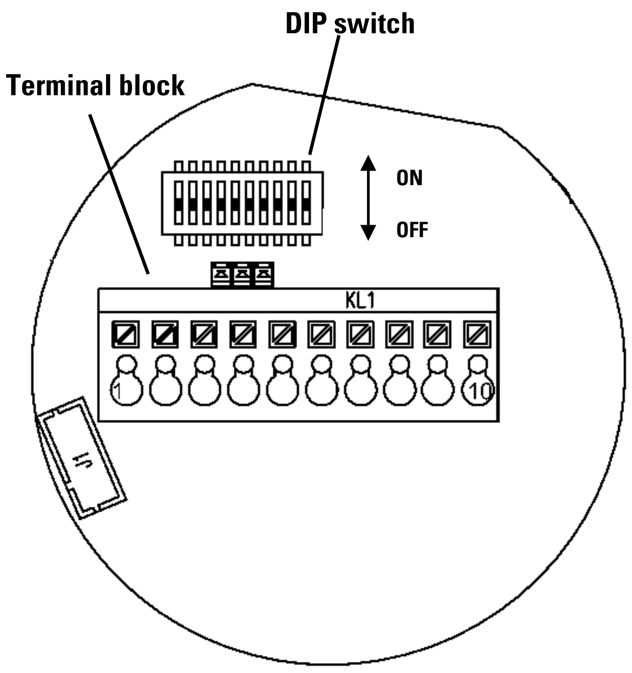
⇒ ***Replace right cable connection by screw plug to ensure the encoder is sealed.***

In case there are other devices following within the same bus phase:

⇒ Lead the continuing cable through the right screw connection and connect it to the terminal 6 (CAN GND out) , terminal 7 (CAN out +) and terminal 8 (CAN out-) (see connection diagram, page 53). Put the cable screen on the sealed cable exits (see cable connection diagram, page 53).

⇒ Attach the bus cover to the encoder and tighten the screws.

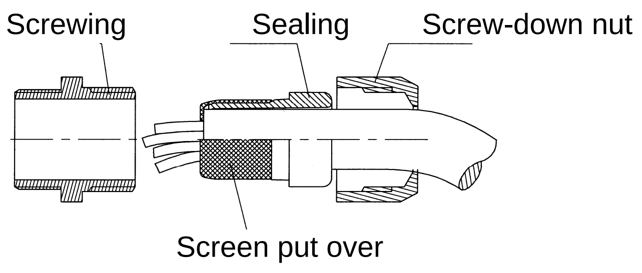
7.3 Connection diagram

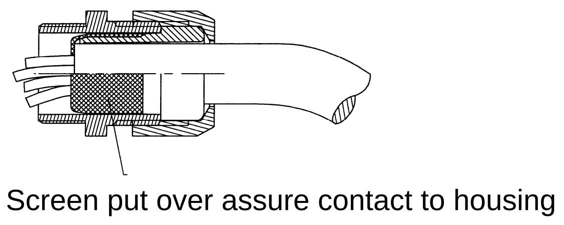


Pin assignment:

Terminal block KL 1 (10-polig)	
No.	Signal name
1	UB in (10-30V)
2	0V in
3	CAN in - (dominant L)
4	CAN in + (dominant H)
5	CAN GND in
6	CAN GND out
7	CAN out + (dominant H)
8	CAN out - (dominant L)
9	0V out
10	UB out (10-30V)

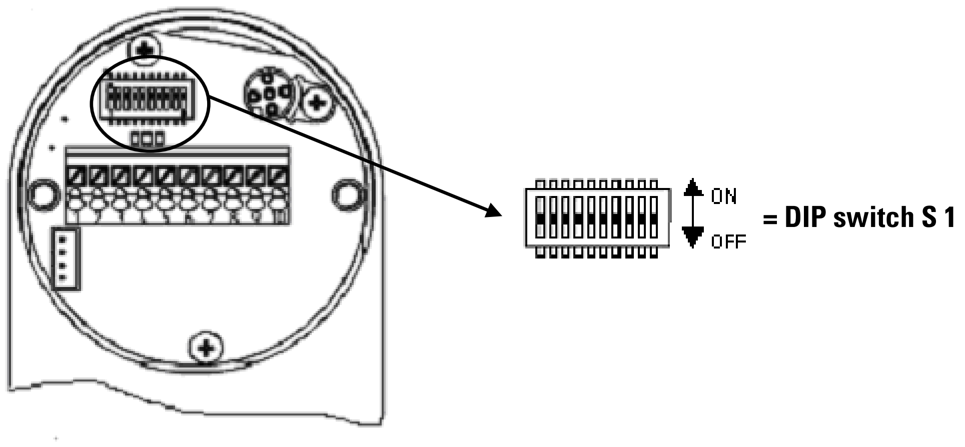
7.4 Cable connection diagram





8 Control- and display elements

8.1 DIP switch (S1)



⇒ **The Versions with cable or with conin DIP switches will be accessible when drawing off the rubber plug from the encoder (see 8.4)**

Warning!

Encoder can become leak, if rubber plug is inserted wrong or is damaged

- IP Protection is not guaranteed!
- A total failure of the encoder is possible!

⇒ Pay attention when removing rubber plug!
⇒ push the rubber plug several times for a tight fit in the bore
⇒ damaged rubber plugs have to be replaced (Art. no. 2565007)

⇒ **The Version with bus cover the DIP switches will be accessible when drawing off the bus cover**

The DIP switches will only be evaluated when running up the encoder (after Reset or Power-on). A modification of the switch position has no effect until the next Reset/ power-on.
The modification will only be accepted if the position of the DIP switches has been modified since the previous Reset. Otherwise the identifiers which have been stored in the EEPROM and which have been possibly programmed via the bus will be preserved.

With the DIP switches:

- 1 to 5 the node number is set
- 6 to 8 the baud rate is set
- 9 and10 the bus termination resistor is activated.

8.1.1 Setting the node number

The 7-bit node number is set by means of the hardware via the DIP switches 1 to 5.
The 5 DIP switches are determine the bits 0 to 4, the remaining bits 5 and 6 are irrevocably set to 0.

DIP switch (ON = 1, OFF = 0)							
DIP switches	**1**	**2**	**3**	**4**	**5**	**-**	**-**
	LSB				MSB		
Bit no.	0	1	2	3	4	5	6
Value	1	2	4	8	16	0	0

The node number is reserved and can not be used by any other node. Therefore the node number 0 is internally changed to 1.
The permissible node numbers is in the range from 1 to 31.
Each node number can only be used once within the same network.

Standard setting ex works: DIP switch in "**OFF**" position (=0), i.e. the resulting node number is **1**!

A new node number is not accepted unless the encoder is run up again (Reset/ power on). All identifiers are reset to their standard values.

Example:

DIP switch (ON = 1, OFF = 0)							
DIP switches	**1=ON**	**2=OFF**	**3=OFF**	**4=ON**	**5=OFF**	**-**	**-**
	LSB				MSB		
Bit no.	0	1	2	3	4	5	6
Value	1	2	4	8	16	0	0

➔ The node number is 9, because DIP switch 1=ON and DIP switch 4=ON.

8.1.2 Setting of the Baud rate

The baud rate is set via the DIP switches 6,7 and 8.

DIP switch (ON = 1, OFF = 0)			Baud rate in Kbit/s
DIP6	**DIP7**	**DIP8**	
OFF	OFF	OFF	1000
ON	**OFF**	**OFF**	**800**
OFF	ON	OFF	500
ON	ON	OFF	250
OFF	OFF	ON	125
ON	OFF	ON	50
OFF	ON	ON	20
ON	ON	ON	10

The standard setting for the baud rate is 800 Kbit/s ex works.

If the **transfer mode 254** is used for the PDO (asynchronous event-controlled, see object1800h): The selected cycle time (see object 6200h) has to be longer than the bus transfer period, in order to allow for an interference-free transmission of the PDOs!
Baud rate 10 kBaud: cycle time at least 14 ms
Baud rate 20 kBaud: cycle time at least 10 ms
Baud rate 50 kBaud: cycle time at least 4 ms

For a cycle time=0 (i.e. PDO in case of modification of value), the baud rate has to amount to at least 125 kBaud.

8.1.3 Activate the bus terminating resistor

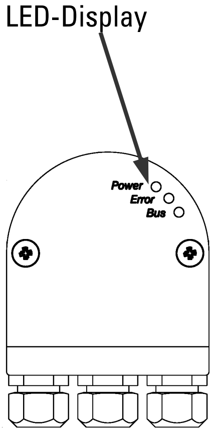
⇒ DIP switches 9 and 10 to „ON“, if encoder is the last device in the bus phase.

If an external bus termination resistor is used the termination resistor has to have minimum 120 Ohm.

Using a terminator plug, the internal termination has to be turned off (DIP 9 and DIP10 = OFF)

8.2 LED Display for the version bus cover

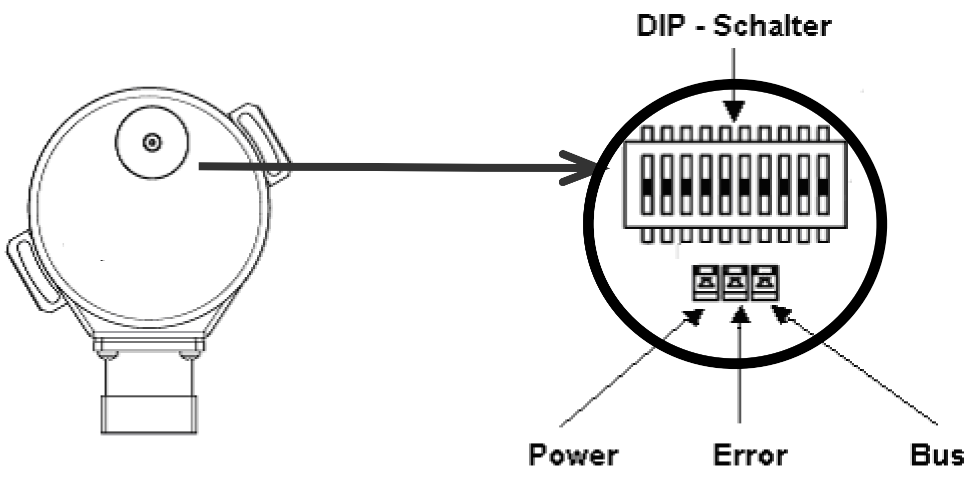
LED-Display



	LED	Meaning	Potential reason
Power (green)	OFF	no Voltage supply	-under-voltage -polarity false
	ON	Voltage supply OK	
Error (red)	OFF	Encoder provides correct position data	
	Blinking	Encoder provides wrong position data	- Bus to Encoder connection interrupted - Valid LED-current regulation range exceeded
Bus (green)	OFF	Connection to master interrupted	- Data line interrupted - wrong baud rate set - Data line misconnected
	Blinking	Connection to master not configured pre-operational state	
	ON	Connection to master, Operational state	

8.3 LED display for the version cable or conin

⇒ Remove the rubber plug. Then you will get a free view on the LED display



Warning!

Encoder can become leak, if rubber plug is inserted wrong or is damaged

- IP Protection is not guaranteed!
- A total failure of the encoder is possible!

⇒ Pay attention when removing rubber plug!
⇒ push the rubber plug several times for a tight fit in the bore
⇒ damaged rubber plugs have to be replaced (Art. no. 2565007)

Meaning of LED Display see 8.2

9 Initial Operation Procedure

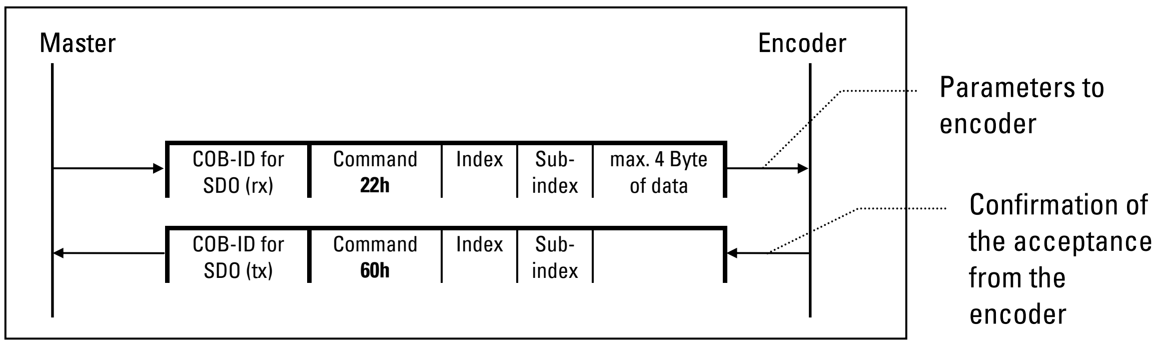
Requirements: The encoder has to be properly connected, the baud rate and the node number have been correctly set.

9.1 Switching on the Supply Voltage

After having switched on the supply voltage, the shaft encoder will be initialized and will be set to the status pre-operational. The parameters of the object directory will be loaded from the EEPROM into the encoder's main memory (RAM) by the initializing routine. If the encoder has not yet been programmed, all parameters are set to standard values. If the encoder has already been programmed, the encoder uses the most recently stored values in the EEPROM.

9.2 Programming the Encoder

If the encoder is to be re-programmed, it is recommended to do so in the pre-operational status. In this status, the bus activity is low and the encoder is not yet overloaded with the output of the process data. The communication and operating parameters in the object directory (see Chapter 5) of the encoder can be modified via SDO access:



The so-called EDS-file (Electronic Data Sheet Specification) serves as an aid for the application of standard CANopen tools and is available as download from our homepage. This EDS file contains the available object directory of the encoder.

9.2.1 Overview Standard Values

In order to administer the identifiers more easily, CANopen uses the "Predefined Master/Slave Connection Set". All identifiers with their standard values are defined in the object directory, so that the encoder can be started up without further programming.

However, the following parameters can be reprogrammed via the SDO access according to the customers' needs (see also Chapter 5.1.2 Detailed Description of the Communication Parameters).

Index (hex)	Name	Standard value (hex)
Communication parameters		
1005	COB-ID for SYNC	80
100C	Monitoring time	0
100D	Life time factor	0
100E	COB ID for control protocols	700 + Node number
1014	COB ID for emergency objects	80 + Node number
Unchangeable	SDOI-Parameter	
	Identifier for SDO (rx) (Master → Encoder)	600 + Node number
	Identifier for SDO (tx) (Encoder→ Master)	580 + Node number
1800	PDO1 Parameter (asynchronous)	
Sub-Index 1	Identifier	180 + Node number
Sub-Index 2	Transfer mode	FE
Sub-Index 3	Inhibit time	0
1802	PDO2 Parameter (synchronous, cyclical)	
Sub-Index 1	Identifier	280 + Node number
Sub-Index 2	Transfer mode	1
Sub-Index 3	Inhibit time	0
Encoder parameters:		
2000	Warning positions	0
2001	Offset value	0
2005	PDO Typ	0 (only position)
6000	Operating parameters	0 (cw, scaling off)
6001	Measuring steps per revolution	max. resolution[1]
6002	Total number of measuring steps	max. total resolution[1]
6003	Preset value	0
6200	Cyclic timer	0

1 value depends on the encoder type

In order to save the modified parameters with no-volt protection, they have to be transferred to the EEPROM via the object 1010h (Save parameters).The original standard values (Default values on delivery) can be reloaded by means of the object 1011h.
Attention: The data previously stored in the encoder RAM will be overwritten by this procedure!

If, by mistake, the same identifier for two different CAN message objects was programmed via the bus and stored to the EEPROM, then, after the next activation, the encoder will no longer be addressable (encoder continues sending only Emergency messages).

The following measures will repair this error:

⇒ Change the encoder's DIP switches (all identifiers will be set to their standard values)

⇒ Possibly, set new identifiers by the bus

⇒ Store parameters via Object 1010h to the encoder's EEPROM.

9.2.2 Example for the Programming of an Encoder:

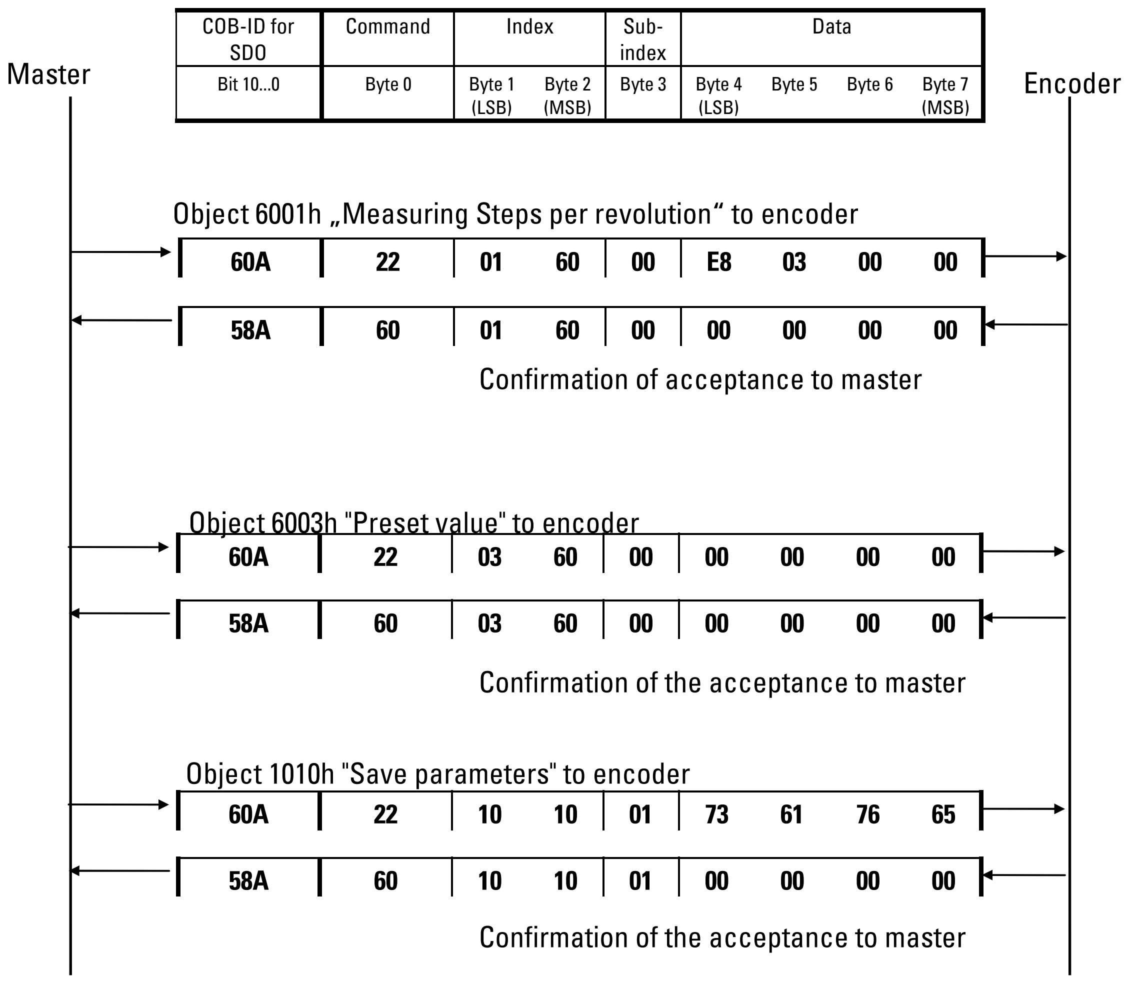
Requirements:

1. Node number = Ah (10 decimal)
2. Resolution is to be set to 1000 steps per revolution
3. Value of position is to be set to 0
4. The new parameters are to be stored in the EEPROM
5. All the other values are to remain at their standard values

Solution:

1. COB ID for SDO (rx) (Master to shaft encoder) = 600h + Node number = 60Ah
   COB ID for SDO (tx) (Shaft encoder to master) = 580h + Node number = 58Ah
2. Send object 6001h (Measuring steps per revolution) with data contents 3E8h (=1000)
3. Send object 6003h (Preset value) with data contents 0
4. Send object 1010 (Save parameters) in subindex 1 with data contents 65 76 61 73h as code word

Operation of the data transmission:



COB-ID for SDO	Command	Index		Sub-index	Data			
Bit 10...0	Byte 0	Byte 1 (LSB)	Byte 2 (MSB)	Byte 3	Byte 4 (LSB)	Byte 5	Byte 6	Byte 7 (MSB)

Object 6001h „Measuring Steps per revolution“ to encoder

60A	**22**	**01**	**60**	**00**	**E8**	**03**	**00**	**00**
58A	**60**	**01**	**60**	**00**	**00**	**00**	**00**	**00**

Confirmation of acceptance to master

Object 6003h "Preset value" to encoder

60A	**22**	**03**	**60**	**00**	**00**	**00**	**00**	**00**
58A	**60**	**03**	**60**	**00**	**00**	**00**	**00**	**00**

Confirmation of the acceptance to master

Object 1010h "Save parameters" to encoder

60A	**22**	**10**	**10**	**01**	**73**	**61**	**76**	**65**
58A	**60**	**10**	**10**	**01**	**00**	**00**	**00**	**00**

Confirmation of the acceptance to master

9.3 Generating the Operating Status

By selecting the NMT command 1: Change to Operational. Either only the encoder can be addressed by using its node number, or all nodes are addressed by node number 00h:

	Byte 0	Byte 1
COB-ID = 0	**01h**	**Node number**

10 Technical Data

10.1 Mechanical

Housing diameter	58 mm
Protection class shaft input	IP 64 or IP 67
Protection class housing	Bus cover IP 67 Cable or Conin IP 64 (IP 67 optional)
Flange	Synchro flange, clamping flange, hubshaft with tether, square flange
Shaft diameter	Solid shaft 6 mm, 10 mm; Hubshaft 10 mm, 12mm
Max. speed	12000 min^{-1} (short term), 10000 min^{-1} (continuous)
Starting torque	$\leq$ 0,5 Ncm
Moment of inertia	3,8 10^{-6} kgm^2
Spring tether (hollow shaft) Tolerance axial Tolerance radial	 $\pm$1,5 mm $\pm$0,2 mm
Max. shaft load	axial 40 N, radial 60 N
Vibration resistance (IEC 68-2-6)	100 m/s^2 (10 - 500 Hz)
Shock resistance (IEC 68-2-27)	1000 m/s^2 (6 ms)
Operating temperature	-40...+85 °C
Storage temperature	-40...+85 °C
Material shaft	Stainless steel
Material housing	Aluminum
Weight approx. ST/ MT	350g/ 400g

10.2 Electrical

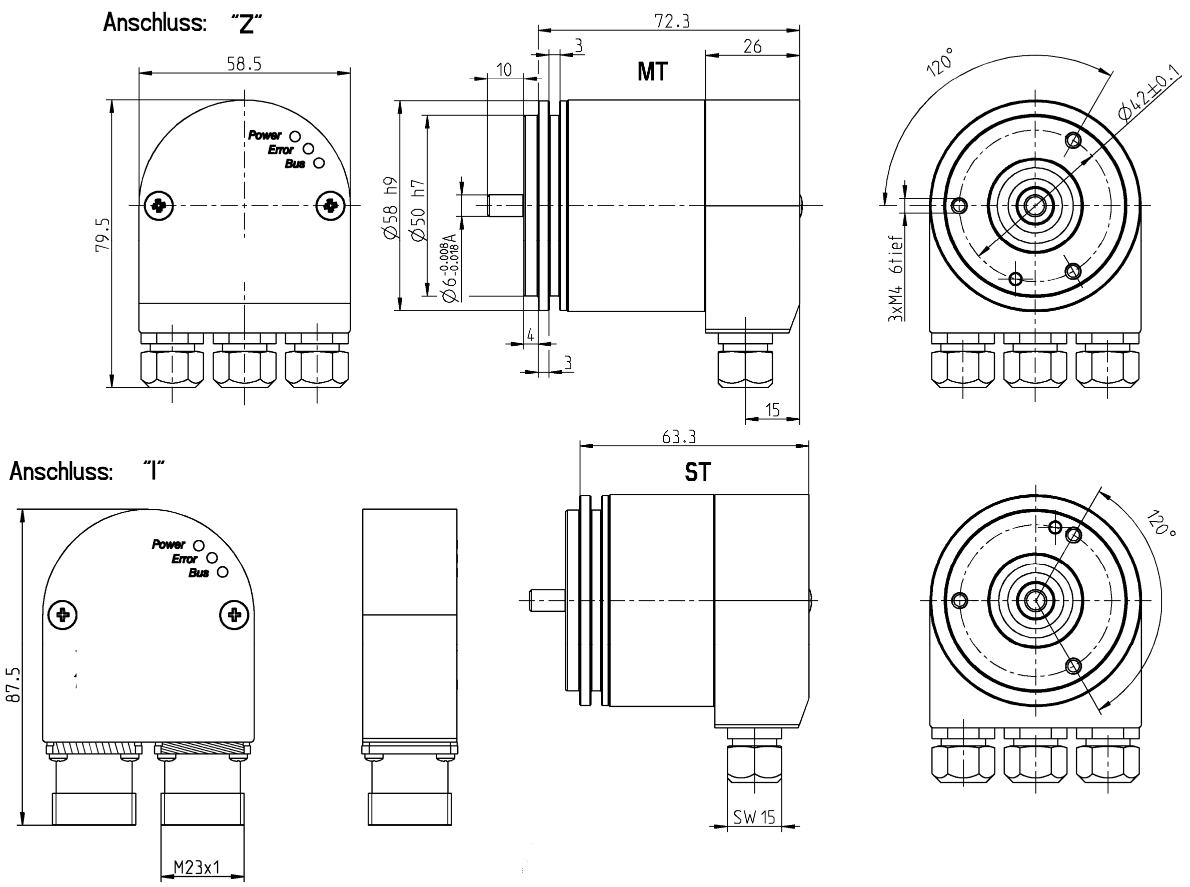
Supply voltage	DC 10 - 30 V
Max. current w/o load ST/MT	220 mA/ 250 mA
EMC	Noise emission according to EN 50081-2 Immunity to interference according to EN 50082-2
Interface	CAN High-Speed according to ISO/ DIS 11898
Protocol	CANopen according to DS 301 with profile DSP406, Programmable encoder according class C2
General design	As per EN 601010-Teil 1, protection class III, Contamination level 2, over voltage class II
Programmable	Resolution, Preset, Offset, Direction
Resolution Singleturn	10 to 14 Bit
Resolution Multiturn	12 Bit
Linearity	± ½ LSB (± 1 LSB for resolution 13, 14, 25, 26 Bit)
Output code	Binary
Integrated special functions	Speed, acceleration, round axis, limit values
Updating of values	Every millisecond (adjustable), on request
Node number	Set via DIP switches
Baud rate	Set via DIP switches
Bus termination resistor	Set via DIP switches
Connection	Cable radial or axial Conin radial or axial, cw or ccw Bus cover with • 3 sealed cable exits • double conin, 9 pole, cw, radial

11 Dimensioned drawings

11.1 Synchro flange

11.1.1 Connection for bus cover

I Bus cover with double conin, 9 pole, cw, radial
Z Bus cover with 3 sealed cable exits



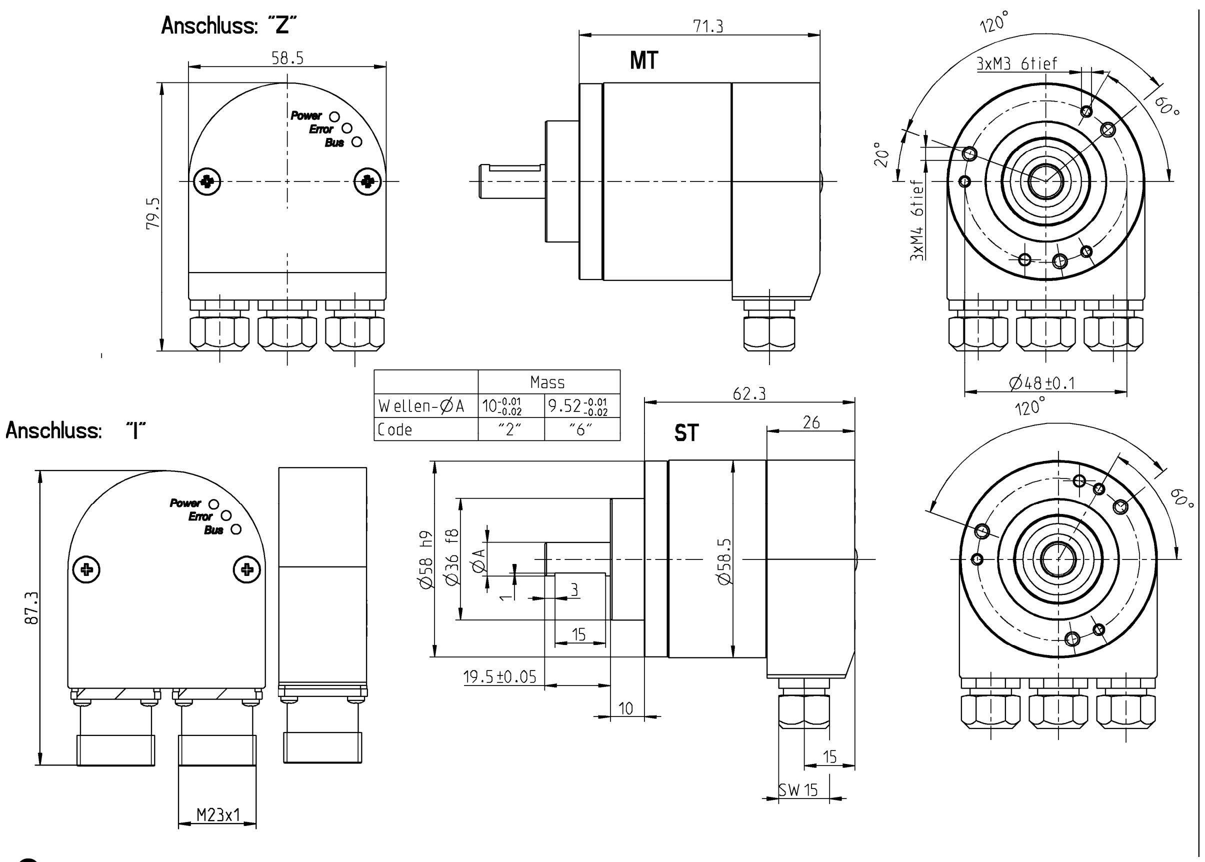
The sealed cable exits require cable diameters in a range from:
7.0 ... 7.4 mm;

Bus cables with a diameter of 7.9 ... 8.7 mm can be used, if the standard sealing are replaced by the enclosed sealing, which have a larger diameter.

11.2 Clamping flange

11.2.1 Connection for bus cover

I Bus cover with double conin, 9 pole, cw, radial
Z Bus cover with 3 sealed cable exits



The sealed cable exits require cable diameters in a range from:
7.0 ... 7.4 mm;

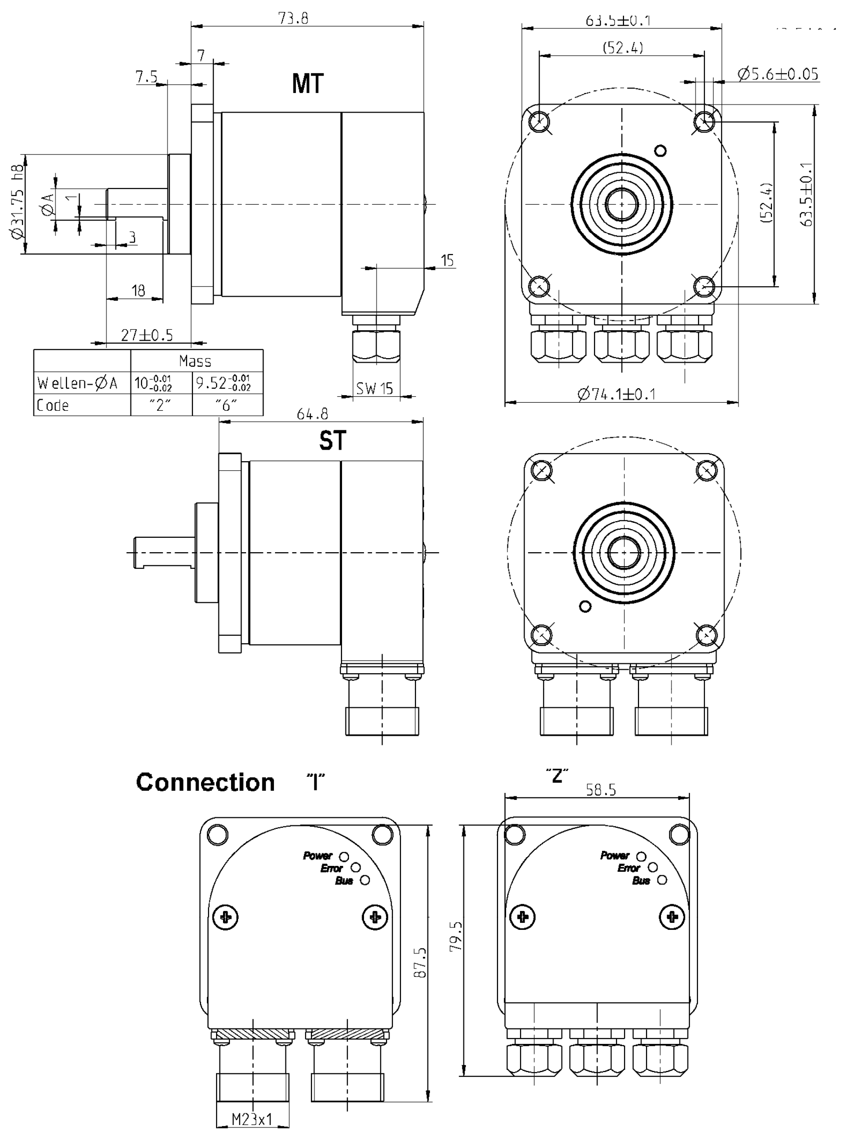
Bus cables with a diameter of 7.9 ... 8.7 mm can be used, if the standard sealing are replaced by the enclosed sealing, which have a larger diameter.

H Conin, 12 pole, radial, ccw

11.3 Quadrat flange

11.3.1 Connection for bus cover

I Bus cover with double conin, 9 pole, cw, radial
Z Bus cover with 3 sealed cable exits



	Mass	
Wellen-ØA	$10^{-0.01}_{-0.02}$	$9.52^{-0.01}_{-0.02}$
Code	"2"	"6"

The sealed cable exits require cable diameters in a range from:
7.0 ... 7.4 mm;

Bus cables with a diameter of 7.9 ... 8.7 mm can be used, if the standard sealing are replaced by the enclosed sealing, which have a larger diameter.

Hohner Automazione srl
P.le Cocchi 10, 21040
Vedano Olona (VA) Italy
Tel. +39 0332 866109
Fax +39 0332 866109
E-Mail: hohner.info@hohner.it
www.hohner.it