Baumer

Sensor Solutions • Motion Control • Vision Technologies • Process Instrumentation

Operation Guide

Overview and principles
CANopen

Baumer Electric AG
Hummelstrasse 17
CH-8501 Frauenfeld
Phone +41 (0)52 728 11 22
Fax +41 (0)52 728 11 44
sales.ch@baumer.com
www.baumer.com

11.10
Subject to modification in technic and design.
Errors and ommissions excepted

Contents

1 Safety and operating instructions 5
1.1 Supplementary information 5
1.2 Intended purpose of the equipment 5
1.3 Commissioning 5
1.4 Safety remarks 5
1.5 Transport and storage 5
1.6 Assembly 5
1.7 Electrical commissioning 5

2 CAN bus and CANopen communication 6
2.1 CAN bus 6
2.1.1 CAN bus characteristics 6
2.1.2 Examples of network users for CAN bus devices are: 6
2.2 CANopen 7

3 Communication profile 8

4 CANopen message structure 9

5 Service data communication (SDO) 10
5.1 SDO examples 11

6 Process data communication (PDO) 12
6.1 Synchronous 12
6.2 Cyclical (asynchronous) 12

7 Emergency service 13

8 Network management services 14
8.1 Description of the NMT command 14
8.2 Command byte 14
8.3 NMT state event 14
8.4 The various NMT states 15
8.4.1 Init 15
8.4.2 Pre-operational mode 15
8.4.3 Operational mode 15
8.4.4 Stopped or prepared mode 15
8.5 Status change 15
8.5.1 Start remote node (1) 15
8.5.2 Stop remote node (2) 15
8.5.3 Enter pre-operational mode (3) 15
8.5.4 Reset remote node (4) 15
8.5.5 Reset communication (5) 16

9 Heartbeat protocol 17

10 Layer Setting Services 18
10.1 Message structure 18
10.2 Commands 18
10.2.1 Switch Mode Global 18
10.2.2 Selective switch mode 18
10.2.3 Setting the node ID 19
10.2.4 Setting the bit timing 19
10.2.5 Saving the configuration protocol 19
10.2.6 Activate bit timing parameters 20

10.2.7 Request vendor ID 20
10.2.8 Request product code 20
10.2.9 Request revision number 20
10.2.10 Request serial number 20
10.2.11 Range request 21

11 Diagnosis and useful information 22
11.1 Error diagnosis field bus communication 22
11.2 Error diagnosis via field bus 22
11.2.1 Object 1001h error register 22
11.2.2 Object 1003h predefined error field 22
11.2.3 Object emergency 22
11.3 SDO abort message 22

Appendix

A. Applications 23
A.1 Setting and reading objects 23
A.1.1 Object setting 23
A.1.2 Read object 23
A.2 Commissioning 23
A.3 Changing the node ID (node no.) 23
A.4 Changing the baud rate 24
A.5 Saving the setting in the EEPROM 24

B. Operation 24
B.1 NMT statuses 24
B.2 Reading the position 24
B.3 Defining heartbeat time 25

C. Terminal assignment and commissioning 25
C.1 Mechanical mounting 25
C.1.1 Shaft encoder 25
C.1.2 Hollow shaft / end shaft encoder 25
C.2 Electrical connection 26

D. Connection and start-up with integrated fieldbus interface 26
D.1.1 Setting of node-ID and baud rate 26

E. Terminating resistor 26

1 Safety and operating instructions

1.1 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.

1.2 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.

1.3 Commissioning

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

1.4 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.

1.5 Transport and storage

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

1.6 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.

1.7 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.

2 CAN bus and CANopen communication

2.1 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.

2.1.1 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.

2.1.2 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

2.2 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).

3 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 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.

5 Service data communication (SDO)

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

5.1 SDO examples

Request of a value by the master from the slave
A frequent request will be a request for position. (Encoders: 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

6 Process data communication (PDO)

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.

6.1 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). In synchronous operation, the PDO is requested by the master via the Sync telegram.

Byte 0	Byte 1
COB ID = 80	0

6.2 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.

7 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		Manufacture Specific /Defined in Manuals				

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
Bits of the Error register are defined in the device Profiles.

8 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.

8.1 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.

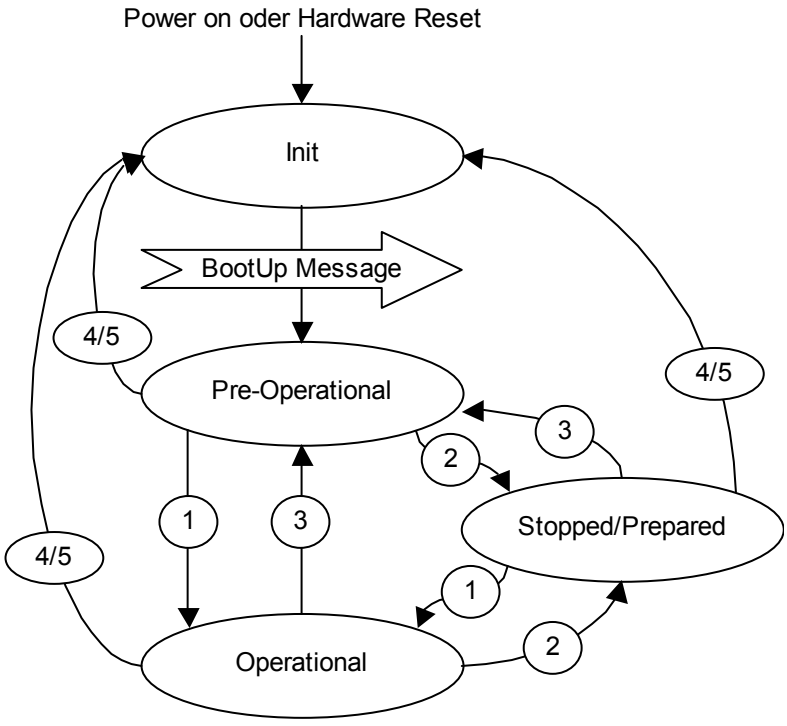
8.2 Command byte

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.

8.3 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.



8.4 The various NMT states

8.4.1 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

8.4.2 Pre-operational mode

In the pre-operational mode, SDOs can be read and written.

8.4.3 Operational mode

In the operational mode, the encoder transmits the requested PDOs. In addition, SDOs can be read and written.

8.4.4 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.

8.5 Status change

8.5.1 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

8.5.2 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

8.5.3 Enter pre-operational mode (3)

Change to the pre-operational mode status.

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

8.5.4 Reset remote node (4)

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

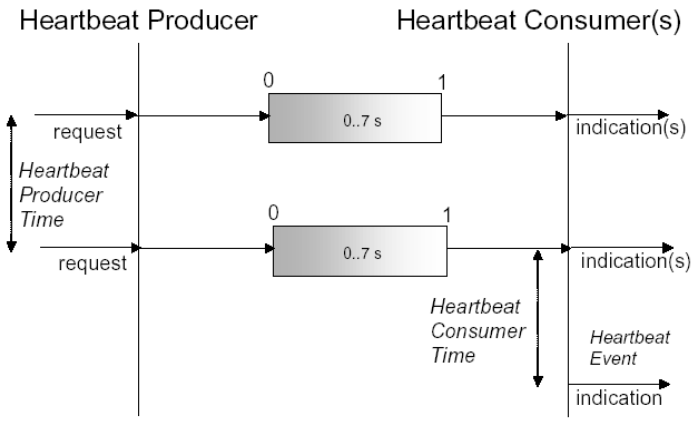
COB ID	Command byte	Node number
0	81h	0..127

8.5.5 Reset communication (5)

With the reset command, the encoder is re-initialized.
Reset Communication (5):

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

9 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).

10 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 20 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.

10.1 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	Dir	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
COB ID	<->	Cmd							

10.2 Commands

10.2.1 Switch Mode Global

COB ID	Dir	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
7E5h	->	04h	Mode	Reserved					

Mode : 0 Operation mode
1 Configuration mode

10.2.2 Selective switch mode

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

COB ID	Dir	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
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)

10.2.3 Setting the node ID

COB ID	Dir	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
7E5h	->	11h	Node ID	reserved					

7E4h	<-	11h	ErrCode	Spec error	

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.

10.2.4 Setting the bit timing

COB ID	Dir	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
7E5h	->	13h	TableSel	TableInd	reserved				

COB ID	Dir	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
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

10.2.5 Saving the configuration protocol

This protocol saves the configuration parameters in the EEPROM.

COB ID	Dir	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
7E5h	->	17h	reserved						

COB ID	Dir	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
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.

10.2.6 Activate bit timing parameters

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

COB ID	Dir	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
7E5h	->	15h	Switch delay		reserved				

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

10.2.7 Request vendor ID

Requesting the vendor ID of a selected encoder

COB ID	Dir	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
7E5h	->	5Ah	reserved						
7E4h	<-	5Ah	Vendor ID				reserved		

Vendor ID : = 5Fh

10.2.8 Request product code

Request product code of a selected encoder

COB ID	Dir	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
7E5h	->	5Bh	reserved						
7E4h	<-	5Bh	Product code				reserved		

Product code : Manufacturer-dependent product code

10.2.9 Request revision number

Request revision number of a selected encoder

COB ID	Dir	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
7E5h	->	5Ch	reserved						
7E4h	<-	5Ch	Revision number				reserved		

Revision number : Current revision

10.2.10 Request serial number

Request serial number of a selected encoder

COB ID	Dir	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
7E5h	->	5Dh	reserved						
7E4h	<-	5Dh	Serial number				reserved		

Serial number : Unique consecutive serial number of the encoder

10.2.11 Range request

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

COB ID	Dir	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
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

11 Diagnosis and useful information

11.1 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.

11.2 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.

11.2.1 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.

11.2.2 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.

11.2.3 Object emergency

Error message of a user.

11.3 SDO abort message

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

Code	Description
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

Appendix

A. Applications

A.1 Setting and reading objects

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

A.1.1 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

A.1.2 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

A.2 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.

A.3 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.

A.4 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.

Entry	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.

A.5 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

B. Operation

B.1 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.

B.2 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

B.3 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).

C. Terminal assignment and commissioning

C.1 Mechanical mounting

C.1.1 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.

C.1.2 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.

C.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

D. Connection and start-up with integrated fieldbus interface

D.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:

COBID	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:

COBID	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. For further details, see object description.

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

COBID	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:

COBID	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.

COBID	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:

COBID	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.

E. 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 Ohm, ¼ W).