Baumer
Passion for Sensors

Description of functions and interfaces

EN580C

Absolute encoder with CANopen bus cover

EN-US

V1 | 4/26/2024

Table of contents

1 About this document

1.1 Instruction manual: purpose and scope of application

The present manual describes the functions and configurable parameters/commands of *Baumer* industrial encoders.

This manual applies to the following product families:

- *EN580C*

1.2 Applicable documents

- Available for download at www.baumer.com:
  - Data sheet
  - EU Declaration of Conformity
- Attached to product:
  - Quickstart
  - General information sheet (11042373)

1.3 Labels in this manual

Identifier	Usage	Example
Dialog element	Indicates dialog elements.	Click the ***OK*** button.
Unique name	Indicates the names of products, files, etc.	*Internet Explorer* is not supported in any version.
`Code`	Indicates entries.	Enter the following IP address: `192.168.0.250`

1.4 Warnings in this manual

Warnings draw attention to potential personal injury or material damage. The warnings in this manual indicate different hazard levels:

Symbol	Warning term	Explanation
	**DANGER**	Indicates an imminent potential danger with high risk of death or serious personal injury if not being avoided.
	WARNING	Indicates potential danger with medium risk of death or (serious) personal injury if not being avoided.
	CAUTION	Indicates a danger with low risk, which could lead to light or medium injury if not avoided.
	NOTE	Indicates a warning of material damage.
	**INFO**	Indicates practical information and tips that enable optimal use of the devices.

2 General functionality

Absolute rotary encoder with flange diameter 58 mm. Position, speed and acceleration values are transmitted via CANopen interface/protocol (EN 50325-5). The encoder is developed in accordance with CiA standards:

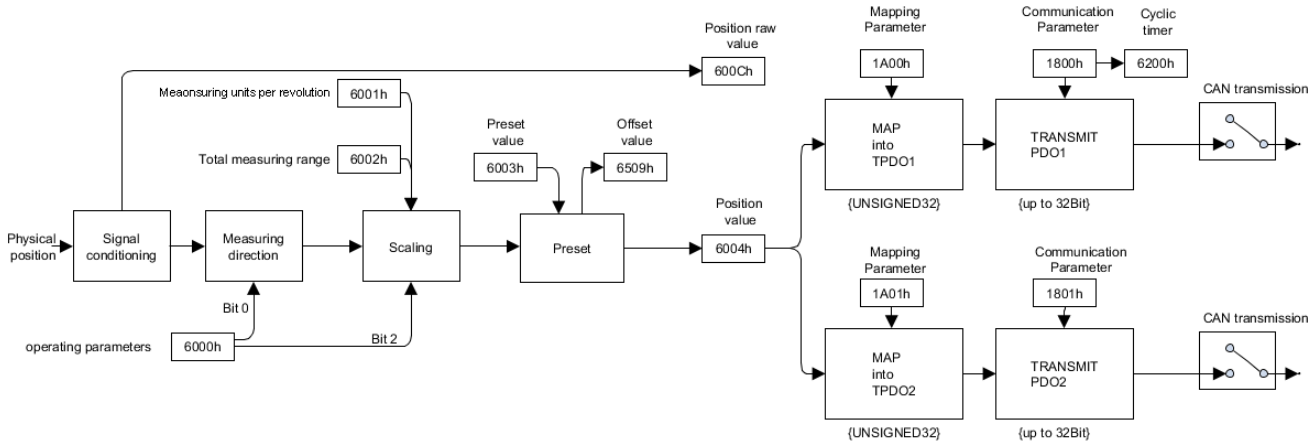
- CiA DS301 (Communication profile)
- CiA DSP305 (LSS Profile)
- CiA DS406 (Device profile encoder)

3 Operating principle

The sensor element delivers measurement signals for the absolute rotary movement of the code disc. Absolute encoders assign a unique value to each position. For doing so, the code disk is rotating over a reading head (Opto Asic), which detects the uniquely encoded position signal throughout one disc revolution. The LED positioned at the opposite of the reading head generates the required light.

In the event of power failure, the unique shaft position will be retained. This eliminates the need for any reference run relating to the start or home position after power supply has been restored.

4 Block diagram



III. 1: Operating principle, overview

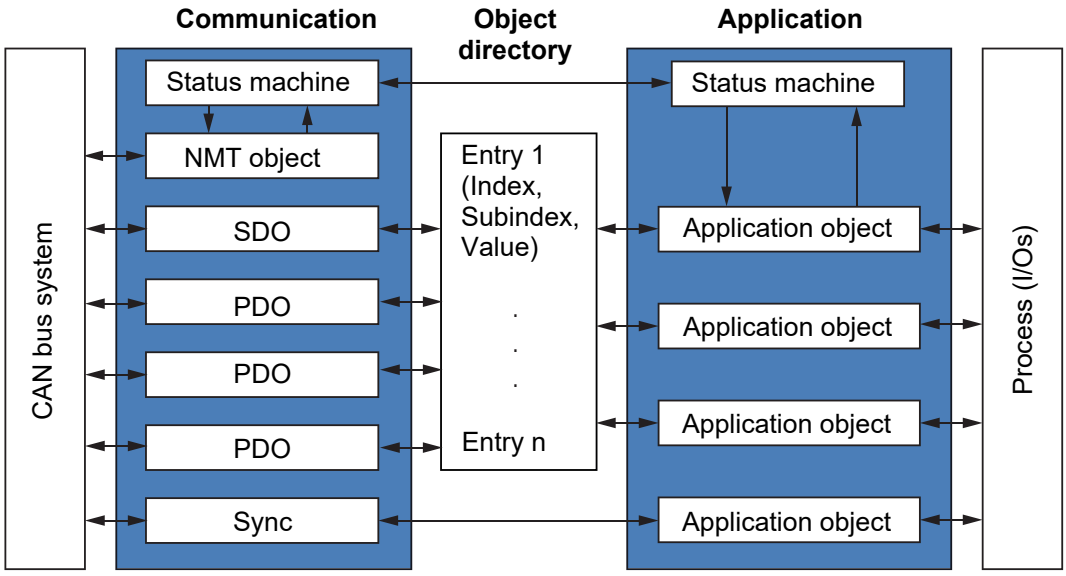
5 Interface

5.1 CANopen

CANopen is a common *CAN* application layer, optimized for fast data exchange in real-time systems. The organization CAN in Automation (CiA) is responsible for the standards applicable to the related profiles.

CANopen comprises the protocol definition (communication profile) as well as the device profiles for the respective device class. Process data objects (PDO) serve for fast communication of input and output data. The *CANopen* device parameters and process data are structured in an object directory.

Any data in this object directory are accessible via service data objects (SDO). There are more objects (e.g. telegram types) for network management (NMT), synchronization, error messages, etc.



Ill. 2: CANopen model

CANopen allows for:

- Easy access to all device and communication parameters
- Synchronization of several devices
- Automatic network configuration
- cyclic and event-driven process data traffic

CANopen consists of four communication objects (COB) with different properties:

- Process data objects for real-time data (PDO)
- Service data objects for parameter and program transfer (SDO)
- Network management (NMT, Heartbeat)
- Predefined objects (for synchronization, emergency messaging)

All device and communication parameters are organized in an object directory. An object includes object name, data type, number of sub indexes, structure of parameters and address. According to CiA, this object directory comprises three parts: Communication profile, device profile and manufacturer-specific profile.

Also see about this

CANopen object dictionary [▶ 40]

5.1.1 Supported profiles

The following CANopen profiles are supported:

- CiA 301 / Version 4.2.0 (Communication)
- CiA 305 / Version 3.0.0 (LSS)
- CiA 406 / Version 4.1.0 (encoder profile)

5.1.2 Supported CANopen services

The device supports the following CANopen services:

- 1 Network Management (according to CiA 301)
- 1 SDO server (according to CiA 301)
- 2 TPDOs (according to CiA 301/CiA 406)
- 1 Emergency Producer (according to CiA 301/CiA 406)
- 1 Heartbeat Producer (according to CiA 301)
- 1 Node guarding (according to CiA 301)
- 1 LSS client (according to CiA 305)

5.1.3 SDO Service

The sensor supports 1 SDO server (expedited read/write, segmented read).

Structure of a *SDO* telegram:

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

A *SDO-COB ID* is structured as follows:

- Master → Encoder : 600h + *Node-ID*
- Encoder ← Master : 580h + *Node-ID*

DLC (Data length code) denotes the telegram length. It is structured as follows:

1 byte command + 2 byte object + 1 byte subindex + number of data bytes (0...4).

The command byte defines whether data is read or set and the number of data bytes:

SDO command	Function	Length	Description
22h	Download Request	Max. 4 bytes	Send parameters to rotary encoder
23h	Download Request	4 bytes	Send parameters to rotary encoder
2Bh	Download Request	2 bytes	Send parameters to rotary encoder
2Fh	Download Request	1 byte	Send parameters to rotary encoder
60h	Download Response	–	Transfer confirmation to Master
40h	Upload Request	–	Parameter request to encoder
42h	Upload Request	Max. 4 bytes	Parameters to master, max. 4 bytes

SDO command	Function	Length	Description
43h	Upload Request	4 bytes	Parameter to master, 4 bytes
4Bh	Upload Request	2 bytes	Parameter to master, 2 bytes
4Fh	Upload Request	1 bytes	Parameter to master, 1 byte
80h	Abort Message		Rotary encoder reports error code to master

abort message signals an object access error. SDO command byte is 80h. Object and subindex are those of the requested object. The error code comes in bytes 8...5.

COB 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

Bytes 8...5 equal SDO *abort message* (byte 8 = MSB). The following messages are supported:

- 05030000h - Toggle bit unchanged
- 05040001h - Command not valid or unknown
- 06010001h - Read access to write only
- 06010002h - Write access to read only
- 06020000h - Object supported
- 06040041h - No object mapping to PDO
- 06040042h - would exceed PDO length
- 06040042h - Parameter incompatible
- 06060000h - Access error due to hardware error
- 06070010h - Incorrect data type
- 06090011h - Subindex not supported
- 06090030h - Value outside the limit
- 06090031h - Value too high
- 06090032h - Value too small
- 08000000h - General error
- 08000020h - Incorrect memory signature
- 08000022h - Error due to current device status
- 08000024h - No data available

SDO examples

Master requests value from slave. Typical query is a position query: Object 6004h

COB ID	DLC	Com-mand	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

Slave responds position value to master. The position value has a length of 4 bytes, detailed values can be found at object 6004h.

COB ID	DLC	Com-mand	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

Master writes value to slave. Setting the position value is in preset object 6003h.

COB ID	DLC	Com-mand	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

Response of slave to written value.

COB ID	DLC	Com-mand	Object L	Object H	Subindex	Data 0	Data 1	Data 2	Data 3
580h + Node-ID	8	60h	03h	60h	0	a	b	c	d

5.1.3.1 Store parameters

Writing the ASCII value ***save*** to 1010h-xwill save the corresponding to the non-volatile memory. The parameters are loaded from the non-volatile memory after reset or power-on.

A write command (SDO download request) to object 1010h-x will be immediately acknowledged by the encoder. Non-volatile storage runs in the background.

⚠ WARNING

Unexpected device behavior caused by incorrect settings

Interruption of power supply immediately after transmission of the save command will restore the default parameters at next power-on.

a) Make sure any power interruption or transmission of NMT reset command will be at least 1 second after transmission of the save command.

For more detailed information on the following please refer to chapter *Annex [▶ 40]*.

CANopen access: 1010h

Name	Object	Subindex	Description
Store parameters	1010h	–	
Highest subindex supported		00h	5
Save all parameters		01h	=“evas“ (65766173h) to save
Save communication parameters		02h	=“evas“ (65766173h) to save
Save application parameters		03h	=“evas“ (65766173h) to save
Save manuf. specific parameters		04h	=“evas“ (65766173h) to save

Signature	MSB			LSB	
ISO 8859	**e**	**v**	**a**	**s**	character
	0x65	**0x76**	**0x61**	**0x73**	hex
	1702257011				dez

5.1.3.2 Restore default parameters

Writing the ASCII value ***load*** to 1011h-x will immediately restore default in the corresponding objects.

NOTICE

Changes will not be adopted until reset or at next power-on.

For more detailed information on the following please refer to chapter *Annex [▶ 40]*.

CANopen Access: 1011h

Name	Object	Subindex	Description
Restore default parameters	1011h	–	
Highest subindex supported		00h	5
All parameters		01h	=“daol“ (64616F6Ch) to load
Communication parameters		02h	=“daol“ (64616F6Ch) to load
Application parameters		03h	=“daol“ (64616F6Ch) to load
Manuf. specific parameters		04h	=“daol“ (64616F6Ch) to load

Signature	MSB			LSB	
ISO 8859	**d**	**a**	**o**	**l**	character
	0x64	**0x61**	**0x6F**	**0x6C**	hex
	1684107116				dez

5.1.4 PDO Service

TPDO1 and TPDO2 are supported. PDO transmission is only in NMT operating mode **Operational**.

5.1.4.1 Communication types

CANopen supports different communication types of process data objects. The following communication types are supported (object 180xh-2):

Communication type	Description
Synchronous transmission (1-240)	In synchronous data transmission, PDO transmission is after the n-th sync frame.
Asynchronous transmission (255)	In asynchronous data transmission, PDO transmission is time-triggered. The time interval between 2 PDOs can be set in object 180xh-5 or alternatively in 6200h.
Manufacturer-specific transmission (254)	Standard setting. Corresponds to asynchronous transmission.

For detailed parameter information see chapter *Annex [▶ 40]*.

5.1.4.2 COB-ID

The COB ID for both PDOs is changed by object 180xh-1.

Standard values:

- TPDO1: 180h + *Node-ID*
- TPDO2: 280h + *Node-ID*

Changes are immediately adopted.

NOTICE

Overwriting and saving the COB ID for TPDOx will retain it even in the event of later changes to the *Node-ID*.

5.1.4.3 PDO mapping

The encoder supports dynamic mapping. Both objects *1A00h* and *1A01* are used for configuration.

The standard configuration is defined in the object directory.

Instruction:

a) Disable mapping by writing 0 to object *1A0xh-0*.

b) Write the desired mapping entry.

c) Re-enable mapping by writing the number of mapped objects to object *1A0xh-0*.

INFO

In the object directory, in column *Access rights* the mappable objects are flagged with *m*.

5.1.4.3.1 TPDO mapping parameter

For more detailed information on the following please refer to chapter *Annex [▶ 40]*.

CANopen access: 1A00h

Name	Object	Subindex	Description
Transmit PDO1 mapping	1A00h	–	
Number of mapped application objects in TPDO		00h	Maximum value is 8
1st mapping parameter		01h	Position encoder, Object 6004h

CANopen Access: 1A01h

Name	Object	Subindex	Description
Transmit PDO2 mapping	1A01h	–	
Number of mapped application objects in TPDO		00h	Maximum value is 8
1st mapping parameter		01h	Position encoder, Object 6004h

5.1.4.3.2 TPDO communication parameter

For more detailed information on the following please refer to chapter *Annex [▶ 40]*.

CANopen access: 1800h

Name	Object	Subindex	Description
TPDO 1 communication parameter	1800h	–	Transmit PDO mapping 1
Highest subindex supported		00h	5
COB-ID		01h	COB-ID for TPDO 1
PDO type		02h	Transmission type
Event timer		05h	Cycle time [in ms]

CANopen Access: 1801h

Name	Object	Subindex	Description
TPDO 2 communication parameter	1801h	–	Transmit PDO mapping 2
Highest subindex supported		00h	5
COB-ID		01h	COB-ID for TPDO 2
PDO type		02h	Transmission type
Event timer		05h	Cycle time [in ms]

5.1.4.3.3 Cycle timer PDO1

This object mirrors object 1800:05h (*Event timer*).

For more detailed information on the following please refer to chapter *Annex [▶ 40]*.

CANopen access: 6200h

Name	Object	Subindex	Description
Cycle timer PDO1	6200h	–	In milliseconds, internally linked to object 1800:05h

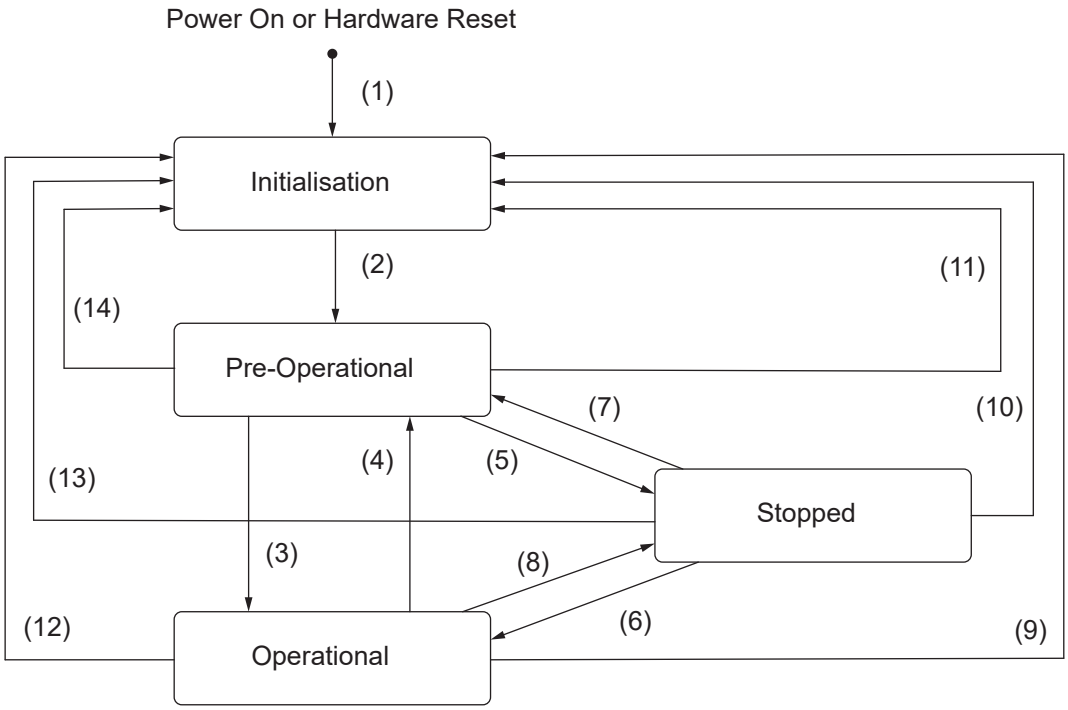
5.1.5 Network management (NMT)

Network management (NMT) defines the communication behavior of a *CANopen* node.

There are two categories of network management: The NMT services for device control can be used to initialize, start and stop the bus users. There are also NMT services for connection monitoring.

There are the following status:

- *Init (Initialisation)*
- *Pre-Operational*
- *Operational*
- *Stopped*



Ill. 3: Status of a *CANopen* node

(1)	At Power On the NMT state initialisation is entered autonomously
(2)	NMT state initialisation finished - enter NMT state Pre-Operational automatically
(3)	NMT service start remote node indication or by local control
(4), (7)	NMT service enter Pre-Operational indication
(5), (8)	NMT service stop remote node indication
(6)	NMT service start remote node indication
(9), (10), (11)	NMT service reset node indication
(12), (13), (14)	NMT service reset communication indication

Status	Description
Init (Initialisation)	After power on, a *CANopen* node is automatically in status *Init*. Having completed *Init*, the node is automatically in status *Pre-Operational*.
Pre-Operational	The service data objects (SDO) are active and the node can be configured. Process data objects (PDO) are still locked.

Status	Description
Operational	Process data objects (PDO) are active. If reading or communication is no longer feasible due to a problem (e.g. CAN error), the encoder will try to transmit a corresponding emergency message. This way, the *CANopen* master will immediately recognize any fatal error.
Stopped	Communication with node is not possible. Only NMT messages are received. The outputs go into error state.

5.1.5.1 NMT Reset Communication

This function will trigger CAN controller restart.

Internal initialization time is <1s. Next, the boot-up message is transmitted.

NOTICE

Any configuration parameters which had not been saved will be lost.

CANopen: NMT Reset Communication

COB-ID	Byte 0	Byte 1
0	82h (NMT Communication Reset)	*Node-ID* (0=Broadcast)

Tab. 1: NMT-Frame

Once having successfully completed the function, the encoder transmits a *Boot-up Message*.

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

5.1.5.2 NMT Reset Node

Command *NMT Reset Node* will completely reset the encoder.

Internal initialization time is <1s. Next, the boot-up message is transmitted.

NOTICE

Any configuration parameters which had not been saved will be lost.

CANopen: NMT Reset Node

COB-ID	Byte 0	Byte 1
0	81h (NMT Reset)	*Node-ID* (0=Broadcast)

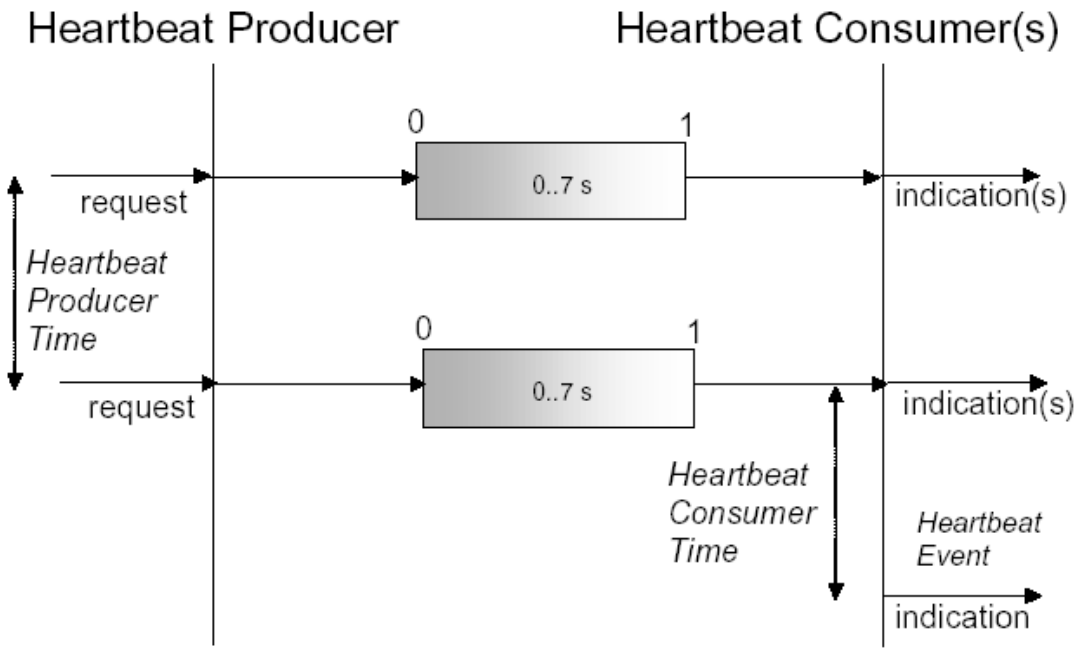
Tab. 2: NMT-Frame

Once having successfully completed the function, the encoder transmits a *Boot-up Message*.

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

5.1.6 Heartbeat

The sensor supports heartbeat producer functionality. Configuration is in done via object 1017h.



A *Heartbeat Producer* is cyclically transmitting the heartbeat message at the frequency specified in object *Producer heartbeat time*. One or more *Heartbeat Consumer* can receive the message. The relationship between producer and consumer is configured via object directory entries. The *Heartbeat Consumer* monitors heartbeat reception within the *Heartbeat Consumer Time*. Not receiving the heartbeat within this time will generate a heartbeat event.

Example of a heartbeat protocol

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

Heartbeat messages comprise the *COB-ID* and one byte. The NMT status is delivered in the byte.

- 0: Boot up event
- 4: Stopped
- 5: Operational
- 127: Pre-Operational

In other words, in the example, the sensor is in state Pre-Operational (7Fh = 127).

5.1.6.1 Consumer heartbeat time

Object consumer heartbeat time must provide the expected heartbeat cycle times. Heartbeat producer monitoring starts once the first heartbeat has been received. If heartbeat time is 0 or node ID is 0 respectively higher than 127, the corresponding object entry will not be used.

The heartbeat time must be given in multiples of 1ms.

NOTICE

Consumer heartbeat time should be higher than the *Producer heartbeat time*. Prior to receiving the first heartbeat, the heartbeat status is unknown.

For more detailed information on the following please refer to chapter *Annex [▶ 40]*.

CANopen Access: 1016h

Name	Object	Subindex	Description
Consumer heartbeat time	1016h	–	Consumer heartbeat time [ms].

5.1.6.2 Producer heartbeat time

Function *Producer heartbeat time* is used to read/write the producer heartbeat time in [ms].

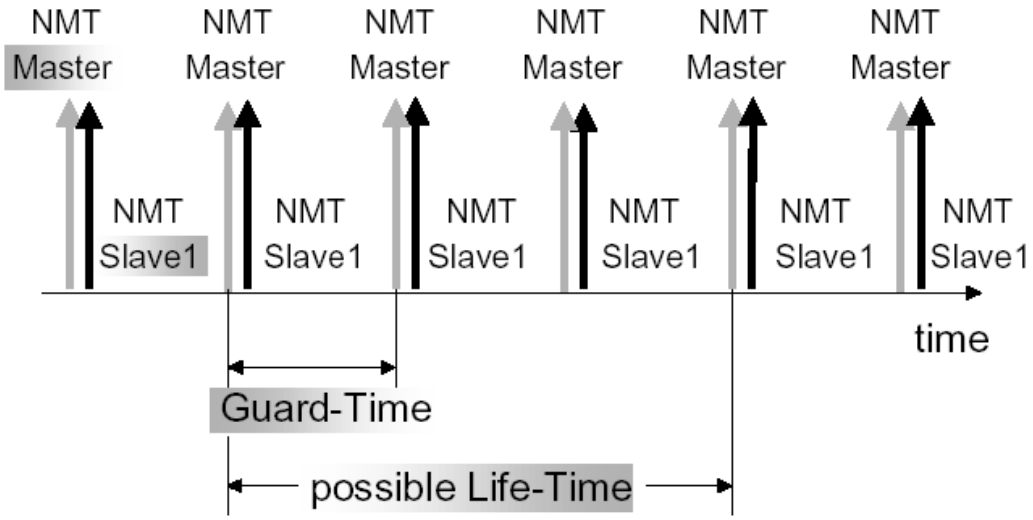
For more detailed information on the following please refer to chapter *Annex [▶ 40]*.

CANopen access: 1017h

Name	Object	Subindex	Description
Producer heartbeat time	1017h	–	Producer heartbeat time [ms]. 0=deaktiviert

5.1.7 Node and life guarding

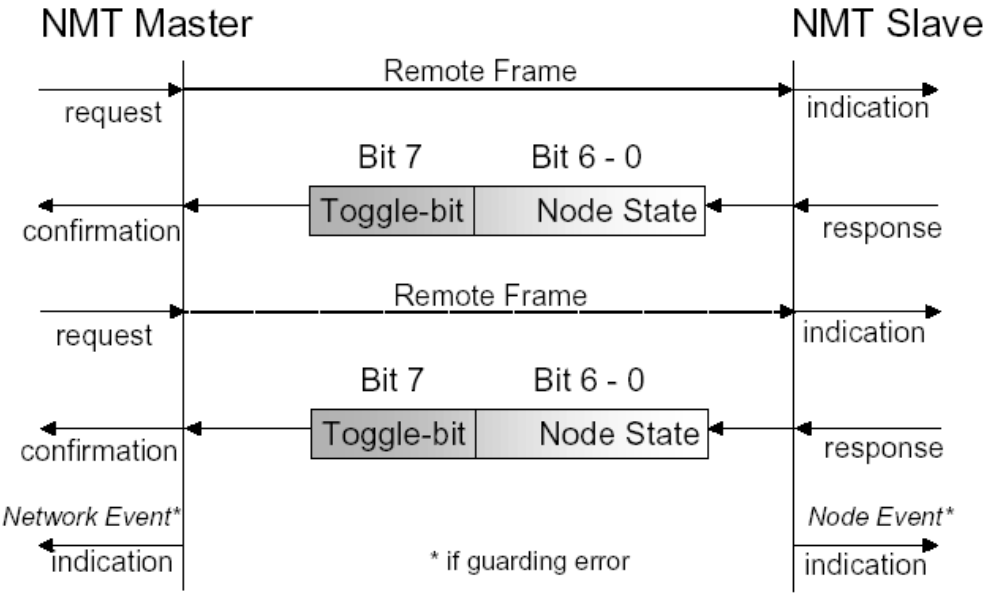
The sensor supports the node and life guarding functionality. Configuration is via CANopen objects 100Ch and 100Dh.



The NMT master can create a database with the respective NMT status of each node. This log can be used to check whether a node has withdrawn from the bus connection. Furthermore, each node can also monitor whether the control unit is still active.

The NMT master starts monitoring by a remote frame to the desired node. Each remote frame resets the life time at the station. In addition, the station returns its NMT status. This allows the NMT master to check whether the node is in correct NMT state and to react in the event of error.

Lifetime having expired will trigger a "node event". The behavior in the event of error is defined in object 1029h-1h.



5.1.7.1 Guard time

This function is for reading/writing of Guard time. Guard time defines the sensor monitoring interval (node guarding). 0 means no monitoring.

Multipliying the values of Guard time and Life Time equals the watchdog length for mutual monitoring (Life Guarding/Node Guarding).

For more detailed information on the following please refer to chapter *Annex [▶ 40]*.

CANopen access: 100Ch

Name	Object	Subindex	Description
Guard time	100Ch	–	Guard time (actual guard time is Object 100Ch*100Dh [ms])

Also see about this

- Node and life guarding [▶ 20]
- Life time factor [▶ 21]

5.1.7.2 Life time factor

This function is for reading/writing the Life time factor.

Multiplying the values of Guard time and Life time equals the watchdog length for mutual monitoring (life guarding/node guarding).

For more detailed information on the following please refer to chapter *Annex [▶ 40]*.

CANopen access: 100D

Name	Object	Subindex	Description
Life time factor	100D	–	

5.1.8 Layer Setting Service (LSS)

Baud rate and *Node-ID* can be configured via LSS (compliant to CiA 305). Alternatively, baud rate and *Node-ID* can be changed by accessing objects 2100h and 2101h.

INFO

The values required for LSS addressing, such as *Vendor ID*, revision number, product code and serial number, are printed on the label provided at the encoder housing.

5.1.8.1 Supported functions

- Switch state global
- Switch state selective
- Enable bit timing parameter
- Configure bit timing parameters
- *Node-ID*-Configure protocol
- Save configuration
- Polling the LSS address

- Polling the Identity *vendor ID*
- Polling the product code identity
- Polling the identity revision number
- Polling the identity serial number

- Polling the *Node-ID*
- Identify LSS slave
- LSS Fastscan

5.1.8.2 Message structure

COB-ID

Consumer → Producer : 2021 = 7E5h

Consumer ← Producer : 2020 = 7E4h

The COB-ID is followed by a LSS comand specifier (cs) . In the next step, up to seven data bytes will be transmitted.

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

Switch state global

7E5h →	04h	State	Reserved

State: 0: Operational state
1: Configuration state

Switch state selective

7E5h →	40h	Vendor ID	Reserved
7E5h →	41h	Product code	Reserved
7E5h →	42h	Revision number	Reserved Reserved
7E5h →	43h	Serial number	Reserved
7E4h ←	44h	Status	Reserved

Vendor ID:	ECh
Product code:	Internal product code for the respective sensor
Revision number	Current sensor revision number
Serial number	Unique, consecutive serial number
Status	The sensor returns the new status (0=operating status; 1=configuration status)

Set *Node-ID*

7E5h →	11h	*Node-ID*	Reserved	
7E4h ←	11h	*Error-Code*	*Specific Error*	Reserved

Node-ID	New sensor *Node-ID*
Error-Code	▪ 0 = OK ▪ 1 = *Node-ID* outside range ▪ 2...254 = reserved ▪ 255 = specific error
Specific Error	*Error-Code* = 255 will output the application-specific error code here.

Setting the bit timing (baud rate)

7E5h →	13h	*Table-Sel*	*TableInd*	Re-served
7E4h ←	13h	*Error-Code*	*Specific Error*	Re-served

TableSel	Selects the bit timing table; 0 = standard CiA bit timing table
TableInd	Bit timing entry in the selected table.
Error-Code	▪ 0 = OK ▪ 1 = *Node-ID* outside range ▪ 2...254 = reserved ▪ 255 = specific error
Specific Error	*Error-Code* = 255 will output the application-specific error code here.

Notes:

- To set the bit timing via LSS, use the values according to the standard CIA bit timing table.
- For non-volatile storage, *Save configuration* must be executed.
- The changed baud rate or *Node-ID* become effective after device restart.

5.1.9 Baudrate

This function sets the encoder to a specific baud rate.

For more detailed information on the following please refer to chapter *Annex [▶ 40]*.

CANopen access: 2100h

Name	Object	Subindex	Description
Baudrate	2100h	–	▪ 0: 10 kBit/s (not supported) ▪ 1: 20 kBit/s (not supported) ▪ 2: 50 kBit/s ▪ 3: 100 kBit/s ▪ 4: 125 kBit/s ▪ 5: 250 kBit/s ▪ 6: 500 kBit/s ▪ 7: 800 kBit/s ▪ 8: 1000 kBit/s

NOTICE

Table is different from the CiA standard bit timing LSS table.

- Save the new baud to the non-volatile memory using object 1010h.
- The new baud rate will become effective after device restart or NMT Reset.
- Alternatively LSS can be used for baud rate / bit timing configuration.
- Selecting a hardware switch position other than 00 means readout of baud rate setting will use three hardware dip switches.

5.1.10 Node-ID

This function is for reading and writing the *Node-ID*.

CANopen Access: 2101h

The new *Node-ID* becomes effective after NMT Reset or power on (provided the parameters were saved to the non-volatile memory).

Name	Object	Subindex	Description
Node-ID	2101h	–	Node-ID 1...127 possible

- LSS can also be used for setting the node ID.
- The hardware rotary switches can also be used for setting the node ID.

NOTICE

Setting the node ID hardware rotary switch will dominate over setting via LSS and setting via object 2101h t.

5.1.11 Identification

5.1.11.1 Device Name

This function is for readout the sensor's device name (manufacturer's device name).

For more detailed information on the following please refer to chapter *Annex [▶ 40]*.

CANopen Access: 1008h

Name	Object	Subindex	Description
DeviceName	1008h	–	DeviceName: `EN580C_M`

5.1.11.2 Device Type

Function *Device type* is for readout the device type.

For more detailed information on the following please refer to chapter *Annex [▶ 40]*.

CANopen Access: 1000h

Name	Object	Subindex	Description
Device Type	1000h	–	▪ `00020196h`: Multiturn encoder

5.1.11.3 Identity object

Function *Identify Object* is for readout of product information. This includes

- *Vendor ID*
- Product code
- Revision number
- Serial number

For more detailed information on the following please refer to chapter *Annex [▶ 40]*.

CANopen Access: 1018h

Name	Object	Subindex	Description
Identity object	1018h		
Highest subindex supported		00h	
Vendor ID		01h	Vendor ID
Product code		02h	▪ `118`: EN580C_M Multiturn Encoder
Revision number		03h	Product revision No.
Serial number		04h	Serial No.

5.1.11.4 Module identification

This function reads out the manufacturer-specific offset.

For more detailed information on the following please refer to chapter *Annex [▶ 40]*.

CANopen access: 650Ah

Name	Object	Subindex	Description
Module identification	650Ah		
Highest subindex supported		00h	
Manufacturer offset		01h	

5.1.11.5 Profile & software version

This function reads out software version and profile as a hex value.

For more detailed information on the following please refer to chapter *Annex [▶ 40]*.

CANopen access: 6507h

Name	Object	Subindex	Description
Profile & software version	6507h	–	Contains the implemented encoder device profile version and the manufacturer specific software version.

5.1.11.6 Serial number

Function *serial number* reads out the sensor's serial.

For more detailed information on the following please refer to chapter *Annex [▶ 40]*.

CANopen access: 650Bh

Name	Object	Subindex	Description
Serial number	650Bh	–	Internally linked to object 1018h-4h

5.1.11.7 Software version

This function reads out the sensor's firmware version.

For more detailed information on the following please refer to chapter *Annex [▶ 40]*.

CANopen access: 100Ah

Name	Object	Subindex	Description
Software version	100Ah	–	Manufacturer software version

5.1.12 Diagnostic functions

5.1.12.1 Operating Status

Function *Operating Status* reads out the current operating status of the sensor.

For more detailed information on the following please refer to chapter *Annex [▶ 40]*.

CANopen access: 6500h

Name	Object	Subindex	Description
Operating Status	6500h	–	Bit 0: ▪ `0`: Position CW ▪ `1`: Position CCW Bit 2: ▪ `0`: Scaling function disabled ▪ `1`: Scaling function enabled

5.1.12.2 Operation Time

Function *Operation Time* reads out the operating time of the sensor.

For more detailed information on the following please refer to chapter *Annex [▶ 40]*.

CANopen access: 2A00h

Name	Object	Subindex	Description
Operation Time	2A00h	–	–
Highest subindex supported		00h	–
Current		01h	Currrent operation time since boot up [s].
Total		02h	Total operation time [s].

CANopen access: 6508h

Name	Object	Subindex	Description
Operating Time	6508h	–	Operating time in 0.1 hours

5.1.12.3 Operation Cycle Counter

Function *Operation Cycle Counter* reads out the number of operating cycles.

For more detailed information on the following please refer to chapter *Annex [▶ 40]*.

CANopen access: 2A01h

Name	Object	Subindex	Description
Operation Cycle Counter	2A01h	–	Number of operating cycles. Incremented at Power On.

5.2 Emergency Service

In the event of error, the device transmits an emergency message while setting the corresponding bits in the error register (object 1001h).

Error code access is via object 1003h-x. The error register saves a history of max. 8 error codes.

5.2.1 COB-ID

The COB ID for the Emergency Message can be changed (via object 1014h).

Default value: 80h + Node-ID

Changes are immediately adopted.

NOTICE

Manual editing and saving the COB ID will not change the COB-ID in susequent changes of the Node-ID.

5.2.2 Emergency COB-ID

This function can be used to read/write the sensor's *Emergency COB-ID*.

For more detailed information on the following please refer to chapter *Annex [▶ 40]*.

CANopen access: 1014h

Name	Object	Subindex	Description
Emergency COB-ID	1014h	–	COB-ID of the emergency object

5.2.3 Error Register

Function *Error register* reads out the sensor's error register.

For more detailed information on the following please refer to chapter *Annex [▶ 40]*.

CANopen Access: 1001h

Name	Object	Subindex	Description
Error Register	1001h	–	▪ `Bit0`: Generic error ▪ `Bit4`: Communication error ▪ `Bit7`: Manufacturer-specific error

5.2.4 Error behaviour

Function *error behavior* defines the sensor behavior of in the event of error.

For more detailed information on the following please refer to chapter *Annex [▶ 40]*.

CANopen Access: 1029h

Name	Object	Subindex	Description
Error behaviour	1029h	–	
Highest subindex supported		00h	
Communication error		01h	▪ `0h`: Change to pre-operational mode ▪ `1h`: No state change ▪ `2h`: Change to stopped mode

5.2.5 Alarms

Function *Alarms* outputs the alarms currently present at the sensor.

For more detailed information on the following please refer to chapter *Annex [▶ 40]*.

CANopen Access: 6503h

Name	Object	Subindex	Description
Alarms	6503h	–	Object 6503h provides alarm information according the following table.

The following alarms are supported:

Bit	Description	Value=0	Value=1
0	Position error	Not occurred	Occurred

5.2.6 Supported alarms

This function outputs the currently sensor-supported alarms.

For more detailed information on the following please refer to chapter *Annex [▶ 40]*.

CANopen access: 6504h

Name	Object	Subindex	Description
Supported alarms	6504h	–	Contains the information on supported alarms by the encoder.

5.2.7 Warnings

Function *Warnings* function outputs the warnings currently present at the sensor.

For more detailed information on the following please refer to chapter *Annex [▶ 40]*.

CANopen Access: 6505h

Name	Object	Subindex	Description
Warnings	6505h	–	Object 6505h provides warning information according the following table

Bit	Description	Value=0	Value=1
4	Battery charge	OK	Too low

5.2.8 Supported warnings

This function outputs the warnings currently supported by the sensor.

For more detailed information on the following please refer to chapter *Annex [▶ 40]*.

CANopen access: 6506h

Name	Object	Subindex	Description
Supported warnings	6506h	–	Contains the information on supported warnings by the encoder.

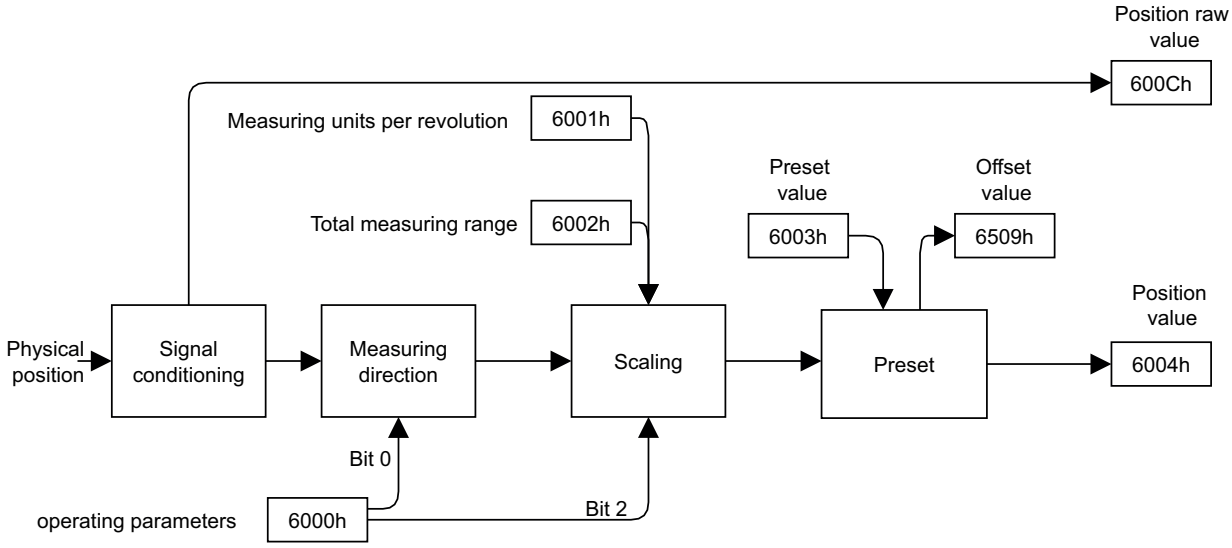
6 Operating functions

6.1 Position encoder value

This function reads out the encoder position.

The position is transmitted as part of the cyclic communication (process data). In addition, the position information is available via acyclic communication.

For more detailed information on the following please refer to chapter *Annex [▶ 40]*.



The position range depends on the settings in objects *6001h* and *6002h*.

CANopen Access: 6004h

Name	Object	Subindex	Description
Position value	6004h	–	Position in steps, scaled value

CANopen access: 600Ch

Name	Object	Subindex	Description
Position raw value	600Ch	–	Position in steps, raw value

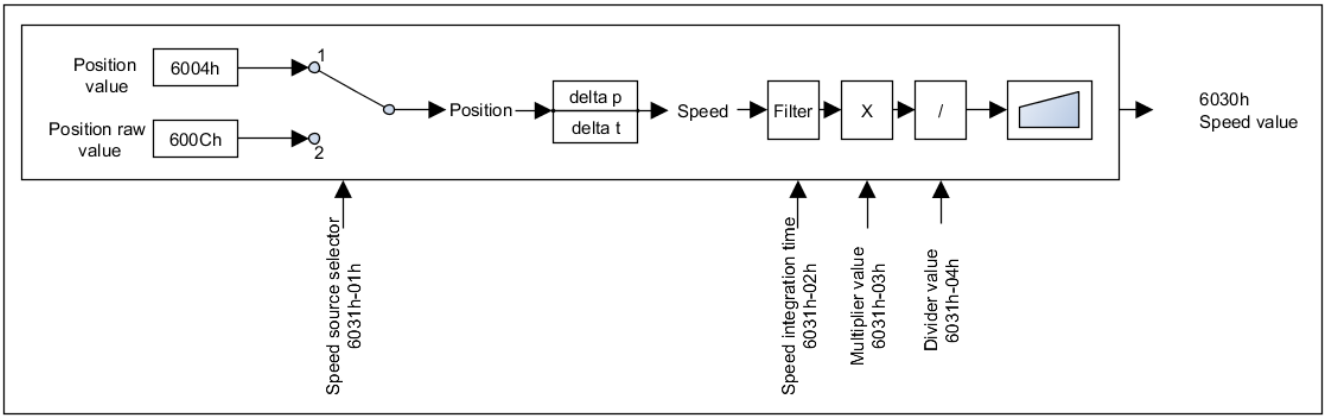
6.2 Speed Value

Function *Speed* provides 16-bit speed information together with the speed unit [steps/sec].

For more detailed information on the following please refer to chapter *Annex [▶ 40]*.

CANopen access: 6030h

Name	Object	Subindex	Description
Speed Value	6030h		
Highest sub-index supported		00h	
Speed Value		01h	Speed value in steps/second



INFO

During *Speed integration time (6031h-02)*, the value determined with *Speed Value (6030h)* is not valid.

6.3 Speed parameter

Function *speed parameter* is for editing various parameters for speed determination.

CANopen access: 6031h

Name	Object	Subindex	Description
Speed parameter	6031h		
Highest sub-index supported		00h	
Speed source selector		01h	▪ 1: 6004h Position value ▪ 2: 600Ch Position raw value
Speed integration time		02h	in ms
Multiplier value		03h	Output value multiplier
Divider value		04h	Output value divider

6.4 Acceleration Value

This function provides 16-bit acceleration information in unit [steps/sec²].

As the acceleration value is highly dynamic, the user should adapt scaling and filter to his application. As the output value is a 16-bit value, the user must pay attention to the limit values.

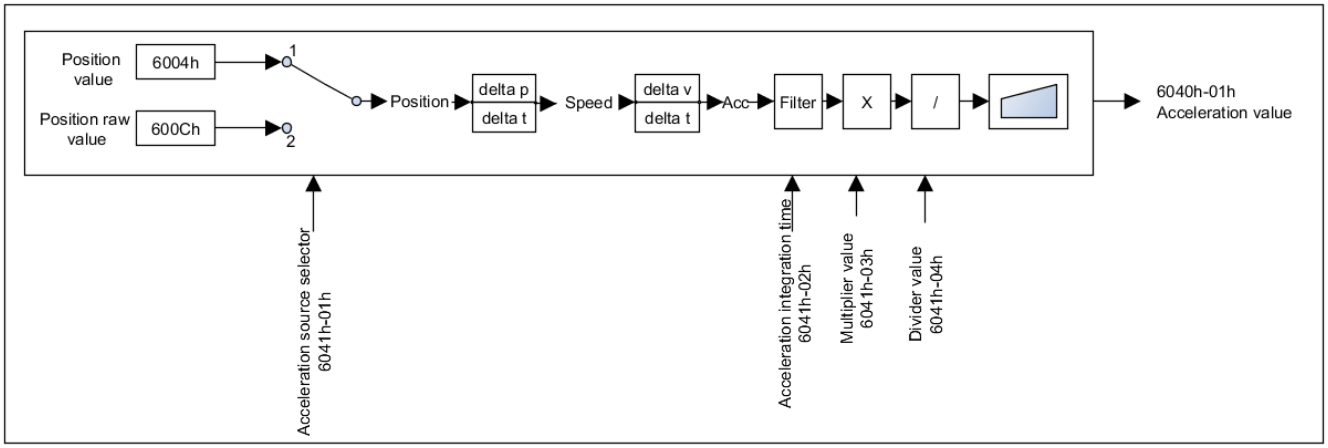
Unit of acceleration value

The acceleration value derives from the position value. In the following is a calculation example where acceleration is calculated by position. In the example, there is a change in speed to 6000rpm within one second.
6000 = rpm/s (revolutions per second)
100 = r/s² (revolutions per second²)
$100*2^{16}$= steps/s² (steps per second²)

For more detailed information on the following please refer to chapter *Annex [▶ 40]*.

CANopen access: 6040h

Name	Object	Subindex	Description
Acceleration Value	6040h		
Highest subindex supported		00h	
Acceleration value		01h	Acceleration value [steps/s²]



INFO

During *Acceleration integration time (6041h-02)*, the value determined with *Acceleration Value (6040h)* is not valid.

6.5 Acceleration parameter

This function is for editing the parameters for determining acceleration.

For more detailed information on the following please refer to chapter *Annex [▶ 40]*.

CANopen access: 6041h

Name	Object	Subindex	Description
Acceleration parameter	6041h		
Highest sub-index supported		00h	
Acceleration source selector		01h	▪ 1: 6004h Position value ▪ 2: 600Ch Position raw value
Acceleration integration time		02h	
Multiplier value		03h	Output value multiplier
Divider value		04h	Output value divider

6.6 Gear Factor

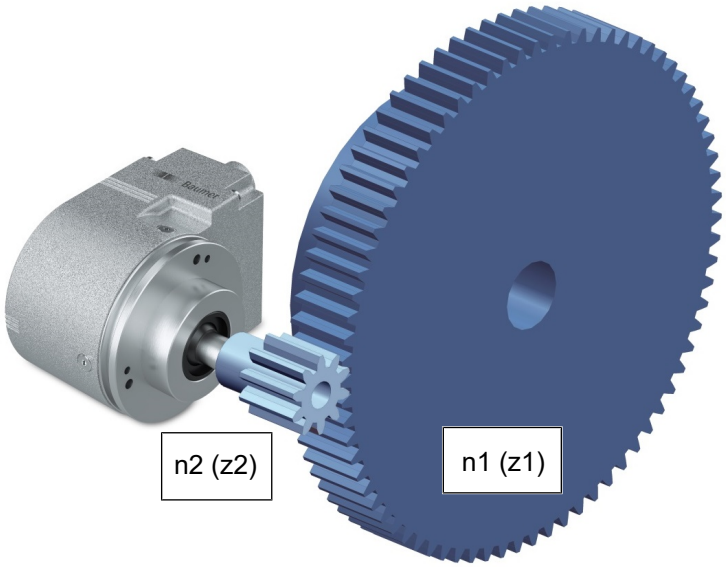
This function is used to configure the electronic gear function.

NOTICE

This function is also known as *Zähler/Nenner-Skalierung* or *Rundachsenfunktion*.

Enabled gear factor () means the encoder mounted to the primary side (gearbox input) will output position data as if mounted to the secondary side (gearbox output).

Parameter *total measuring range* always defines the number of steps required for one revolution at the gearbox output (secondary side).



Primary side (drive side)
Denominator

Secondary side (driven side)
Nominator

$$\text{Gear factor } i = \frac{\text{Numerator}}{\text{Denominator}} = \frac{\text{Speed at drive side (n2)}}{\text{Speed at drive side (n1)}} = \frac{\text{Number of teeth at driven side (z1)}}{\text{Number of teeth at drive side (z2)}}$$

The values for gear factor numerator and denominator result directly from the number of teeth. In the above example, the number of teeth at the secondary side is 75 and 10 on the primary side.

Parameter *Measuring units per revolution* is not set in the gear factor function, but results from the total measuring range, numerator and denominator.

$$\text{Measuring units per revolution} = \text{Total measuring range} * \frac{\text{Denominator}}{\text{Numerator}}$$

Example

Transmission factor to be 75:10 (i.e. 7.5). Required resolution on the secondary side of the gearbox to be "1 revolution = 10000 steps".

Numerator 75, denominator 10. Both numerator and denominator must be integer values. Total measuring range is 10000.

Encoder completes 7.5 revolutions within one revolution on the gearbox secondary side. The encoder value resulting from *Measuring units per revolution* is 10000 / 7.5 = 1333.3333.

For more detailed information on the following please refer to chapter *Annex [▶ 40]*.

CANopen Access: 2001h

Name	Object	Subindex	Description
Gear Factor Configuration	2001h	–	Configuration of electronic gear function
Highest subindex supported		00h	
Mode Control		01h	▪ `0`: electronic gear function disabled ▪ `1`: electronic gear function enabled
Numerator		02h	Numerator of the gear factor
Denominator		03h	Denominator of the gear factor

The formula below equals valid combinations of numerator, denominator and total measuring range.

$$\text{Measuring units per revolution} = \text{Total measuring range} * \frac{\text{Denominator}}{\text{Numerator}}$$

Parameter *Measuring units per revolution* must not exceed the maximum permitted encoder limits.

NOTICE

Parameter *Measuring units per revolution* is calculated by the encoder itself and does not need to be configured.

In this mode, please only configure the following parameters:

- Total measuring range **6002h**
- Gear factor numerator *2001h-02h*

- Gear factor denominator *2001h-03h*

Gear factor: Numerator 2001h-02h

This parameter is only considered with enabled gear factor functionality.

When using gear reduction (n2 < n1), the gear factor numerator is bigger than the denominator.

NOTICE

Term *numerator* is used as a synonym for *counter*.

Gear factor: Denominator 2001h-03h

This parameter is only considered with enabled gear factor functionality.

When using transmission (n2 > n1), the denominator is bigger than the numerator.

NOTICE

Term *denominator* is used as a synonym for *denominator*.

6.7 Number of distinguishable revolutions

This function outputs the maximum number of revolutions.

For more detailed information on the following please refer to chapter *Annex [▶ 40]*.

CANopen access: 6502h

Name	Object	Subindex	Description
Number of distinguishable revolutions	6502h	–	max. multiturn revolutions

6.8 Single turn resolution

This function outputs the current resolution for one revolution [steps/revolution].

For more detailed information on the following please refer to chapter *Annex [▶ 40]*.

CANopen access: 6501h

Name	Object	Subindex	Description
Used single turn resolution [step/rev]	6501h	–	max. Measuring units per revolution

6.9 Operating parameter

This function is used for editing the sensor's operating parameters.

For more detailed information on the following please refer to chapter *Annex [▶ 40]*.

CANopen access: 6000h

Name	Object	Subindex	Description
Operating parameter	6000h	–	Bit 0: ▪ `0`: Position CW ▪ `1`: Position CCW Bit 2: ▪ `0`: Scaling function disabled ▪ `1`: Scaling function enabled

6.10 is

This function will have an influence on the measuring range [Measuring range in steps].

For more detailed information on the following please refer to chapter *Annex [▶ 40]*.

CANopen Access: 6002h

Name	Object	Subindex	Description
Total measuring range	6002h	–	Total measuring range in Steps.

6.11 Measuring units per revolution

This function sets the required resolution for a single revolution [steps/revolution].

For more detailed information on the following please refer to chapter *Annex [▶ 40]*.

CANopen Access: 6001h

Name	Object	Subindex	Description
Measuring units per revolution [Step/rev]	6001h	–	Measuring units per revolution.

NOTICE

The quotient of *Total measuring range (6002h)* and *Measuring units per revolution (6001h)* must not exceed the maximum number of revolutions from object 6502h.

If the quotient of *Total measuring range (6002h)* and *Measuring units per revolution (6001h)* do not correspond to a multiple value 2^n(1,2,4,8,16..), encoder operation is in endless mode.

Example

Measuring units per revolution (6001h)	*Total measuring range (6002h)*	Quotient (no. Revolutions)	Endless mode
8192	536870912	$65536^{=229=2n}$ -multiple	Off
5000	15000	3 $\neq^{2n}$ -multiple	Power on

Limitations

If the encoder is in endless mode and not live, do not turn the encoder shaft by more than 15 Bit (32768) revolutions. Otherwise, the position value is invalid and the encoder requires another referencing using preset (6003h).

6.12 Offset value

This function reads out the sensor offset.

CANopen Access: 6509h

Name	Object	Subindex	Description
Offset value	6509h	–	Internal offset calculated during the preset process.

6.13 Preset value

This function has an influence on the preset value.

For more detailed information on the following please refer to chapter *Annex [▶ 40]*.

CANopen Access: 6003h

Name	Object	Subindex	Description
Preset value	6003h	–	Preset value in steps

7 Annex

7.1 CANopen object dictionary

The tables below show a summary of all SDO objects supported by the encoder.

Object	Object number in Hex
Subindex	
Name	Object name
Data type	U/I = Unsigned/Integer , No. = no of bits, ARR = Array, REC = Record, STR = String
Access rights	ro = read only, wo = write only, rw = read write, m = mappable
Default	Factory settings
Save	X = can be saved in EEPROM
Description	additional explanation

7.1.1 Communication profile

Object	Subindex	Name	Data type	Access rights	Default	Save	Description
1000h		Device type	U32	ro			▪ Multiturn : 00020196h
1001h		Error Register	U8	ro	0h		▪ `Bit0`: Generic error ▪ `Bit4`: Communication error ▪ `Bit7`: Manufacturer-specific error
1002h		Manufacturer status register	U32	ro	0h		Dummy object for internal use only.
1003h		Predefined error field	Array				
	00h	Number of errors	U8	rw	0h		Number of stored messages (0 - 8)
	01h	Last entry	U32	ro			Newest Error Code
	...	...	...	...			...
	08h	Oldest entry	U32	ro			Oldest Error Code
1005h		Sync COB-ID	U32	rw	80h	X	COB-ID of the sync object
1008h		Device name	STR	ro			▪ Multiturn : "EN580C_M"
100Ah		Software version	STR	ro			Software version in ASCII

Object	Subindex	Name	Data type	Access rights	Default	Save	Description
100Ch		Guard time	U16	rw	0h	X	Guard time (actual guard time is Object 100Ch*100Dh [ms])
100D		Life time factor	U8	rw	0h	X	Life time factor
1010h		Store parameters	Array				
	00h	Highest subindex supported	U8	ro	4h		No. of save possibilities = 4
	01h	Save all parameters	U32	rw	1h		=“evas“ (65766173h) to save
	02h	Save communication parameters	U32	rw	1h		=“evas“ (65766173h) to save
	03h	Save application parameters	U32	rw	1h		=“evas“ (65766173h) to save
	04h	Save manuf. specific parameters	U32	rw	1h		=“evas“ (65766173h) to save
1011h		Restore default parameters	Array				
	00h	Highest subindex supported	U8	ro	4h		No. of reset possibilities = 4
	01h	Restore all default parameters	U32	rw	1h		=“daol“ (64616F6Ch) to load
	02h	Restore communication default parameters	U32	rw	1h		=“daol“ (64616F6Ch) to load
	03h	Restore application default parameters	U32	rw	1h		=“daol“ (64616F6Ch) to load
	04h	Restore manuf. specific default parameters	U32	rw	1h		=“daol“ (64616F6Ch) to load
1014h		COB-ID emergency message	U32	rw	80h + Node-ID	X	COB-ID of the emergency object
1016h		Consumer heartbeat time	Array				
	00h	Highest subindex supported	U32	ro	1h		
	01h	Consumer heartbeat time	U32	rw	0h		Consumer heartbeat time in [ms] ▪ Bit 0..15 Consumer heartbeat time in ms ▪ Bit 16..23 Node ID
1017h		Producer heartbeat time	U16	rw	0h	X	Producer heartbeat time in ms (0 = disabled)

Object	Subindex	Name	Data type	Access rights	Default	Save	Description
1018h		Identity object	REC	ro			
	00h	Highest subindex supported	U8	ro	4h		
	01h	Vendor ID	U32	ro	5Fh	–	Vendor ID
	02h	Product code	U32	ro			▪ `118`: EN580C_M Multiturn encoder
	03h	Revision number	U32	ro			Product revision No.
	04h	Serial number	U32	ro			Serial No.
1029h		Error behaviour	Array				
	00h	Highest subindex supported	U8	ro	3h		
	01h	Communication error	U8	rw	1h	X	▪ `0h`: Change to pre-operational mode ▪ `1h`: No state change ▪ `2h`: Change to stopped mode
1800h		TPDO1 communication parameter	REC			X	
	00h	Highest subindex supported	U8	ro	5h	X	
	01h	COB-ID	U32	rw	40000180h + Node-ID	X	COB-ID for TPDO 1
	02h	Transmission type	U8	rw	FEh	X	Transmission type
	05h	Event timer	U16	rw	515	X	Cycle time [in ms]
1801h		TPDO2 communication parameter	REC			X	
	00h	Highest subindex supported	U8	ro	5h	X	
	01h	COB-ID	U32	rw	40000280h + Node-ID	X	COB-ID for TPDO 2
	02h	Transmission type	U8	rw	2h	X	Transmission type
	05h	Event timer	U16	rw	256	X	Cycle time [in ms]

Object	Subindex	Name	Data type	Access rights	Default	Save	Description
1A00h		TPDO 1 mapping parameter	Array				
	00h	Number of mapped application objects in TPDO	U8	rw	1	X	Maximum value is 8
	01h	PDO mapping entry 1	U32	rw	6004’0020h	X	Position encoder
	02h	PDO mapping entry 2	U32	rw	0h	X	
	03h	PDO mapping entry 3	U32	rw	0h	X	
	04h	PDO mapping entry 4	U32	rw	0h	X	
	05h	PDO mapping entry 5	U32	rw	0h	X	
	06h	PDO mapping entry 6	U32	rw	0h	X	
	07h	PDO mapping entry 7	U32	rw	0h	X	
	08h	PDO mapping entry 8	U32	rw	0h	X	
1A01h		TPDO 2 mapping parameter	Array				
	00h	Number of mapped application objects in TPDO	U8	rw	1	X	Maximum value is 8
	01h	PDO mapping entry 1	U32	rw	6004’0020h	X	Position encoder
	02h	PDO mapping entry 2	U32	rw	0h	X	
	03h	PDO mapping entry 3	U32	rw	0h	X	
	04h	PDO mapping entry 4	U32	rw	0h	X	
	05h	PDO mapping entry 5	U32	rw	0h	X	
	06h	PDO mapping entry 6	U32	rw	0h	X	
	07h	PDO mapping entry 7	U32	rw	0h	X	
	08h	PDO mapping entry 8	U32	rw	0h	X	
1F80h		NMT startup	U32	rw	0	X	0h = NMT producer needs to be started by NMT consumer 8h = NMT producer autonomously enters NMT state operational (self-starting)

7.1.2 Manufacturer-specific objects

Object	Subindex	Name	Data type	Access rights	Default	Save	Description
2001h		Gear Factor Configuration	ARR				Configuration of gear function
	00h	Highest subindex supported	U8	ro	3	X	
	01h	Mode Control	U8	rw	1	X	▪ 0: electronic gear function disabled ▪ 1: electronic gear function enabled
	02h	Numerator	U32	rw	1	X	Numerator of the gear factor
	03h	Denominator	U32	rw	1	X	Denominator of the gear factor
2100h		Baud rate	U8	rw	5	X	▪ 0: 10 kBit/s (not supported) ▪ 1: 20 kBit/s (not supported) ▪ 2: 50 kBit/s ▪ 3: 100 kBit/s ▪ 4: 125 kBit/s ▪ 5: 250 kBit/s ▪ 6: 500 kBit/s ▪ 7: 800 kBit/s ▪ 8: 1000 kBit/s The baud rate is activated after a reset or power-on (if parameter is saved to non volatile memory).
2101h		Node-ID	U8	rw	1	X	Node-ID 1...127 possible The new Node-ID is activated after a reset or power-on (if parameter is saved to non volatile memory).
2110h		Manufacturer options	U32	rw	8h		Bit3 = 0 BusOFF not removed Bit3 = 1 reinitate bus after BusOFF

Object	Subindex	Name	Data type	Access rights	Default	Save	Description
2300h		Customer EEPROM					Object to save optional data
	00h	Highest subindex supported	U8	ro	7h		
	01h	Data 0	U16	rw	0h	X	Sub-index can store 16 bit data (save non volatile via Object 1010h)
	02h	Data 1	U16	rw	0h	X	Sub-index can store 16 bit data (save non volatile via Object 1010h)
	03h	Data 2	U16	rw	0h	X	Sub-index can store 16 bit data (save non volatile via Object 1010h)
	04h	Data 3	U16	rw	0h	X	Sub-index can store 16 bit data (save non volatile via Object 1010h)
	05h	Data 4	U16	rw	0h	X	Sub-index can store 16 bit data (save non volatile via Object 1010h)
	06h	Data 5	U16	rw	0h	X	Sub-index can store 16 bit data (save non volatile via Object 1010h)
	07h	Data 6	U16	rw	0h	X	Sub-index can store 16 bit data (save non volatile via Object 1010h)
2A00h		Operation Time	ARR				
	00h	Highest subindex supported	U8	ro	2		
	01h	Current	U32	ro,m	0		Currrent operation time since boot up [s].
	02h	Total	U32	ro,m	0		Total operation time [s].
2A01h		Operation Cycle Counter	U32	ro,m	0		

7.1.3 Standardized device profile

Object	Subindex	Name	Data type	Access rights	Default	Save	Description
6000h		Operating parameter	U16	rw	4h	X	Configuration of encoder operating parameters Bit0: Code sequence ▪ 0: Rising values on CW Rotation ▪ 1: Rising values on CCW Rotation Bit2: Scaling function control ▪ 0: Scaling disabled ▪ 1: Scaling enabled
6001h		Measuring units per revolution [Step/rev]	U32	rw	2000h	X	Measuring units per revolution Allowed range: 1 to 2^{21} steps (200000h) @ Scaling disabled: 200000h
6002h		Total measuring range in measuring units	U32	rw	20000000h	X	Total measuring range in Steps. Number of distinguishable steps over total measuring range in [steps]. Allowed range: 2 to 2^{31} steps (80000000h) @ scaling disabled: 100000000h
6003h		Preset value	U32	rw	0h	X	Preset value in steps
6004h		Position value	U32	ro,m			Position in steps , scaled value
600Ch		Position raw value	U32	ro,m			Position in steps, raw value
6030h		Speed Value	Array	–			
	00h	Highest sub-index supported	U8	ro	1		
	01h	Speed value channel 1	I16	ro,m			Speed value in steps/second

Object	Subindex	Name	Data type	Access rights	Default	Save	Description
6031h		Speed parameter	REC	–			
	00h	Highest sub-index supported	U8	ro	4		
	01h	Speed source selector	U8	rw	1	X	▪ 1: 6004h Position value ▪ 2: 600Ch Position raw value
	02h	Speed integration time	U16	rw	100	X	in ms
	03h	Multiplier value	U16	rw	1	X	Output value multiplier
	04h	Divider value	U16	rw	1	X	Output value divider
6040h		Acceleration Value	Array				
	00h	Highest subindex supported	U8	ro	1		
	01h	Acceleration value channel 1	I16	ro,m			Acceleration value [steps/s²]
6041h		Acceleration parameter	REC				
	00h	Highest sub-index supported	U8	ro	4		
	01h	Acceleration source selector	U8	rw	1	X	▪ 1: 6004h Position value ▪ 2: 600Ch Position raw value
	02h	Acceleration integration time	U16	rw	100	X	in ms
	03h	Multiplier value	U16	rw	1	X	Output value multiplier
	04h	Divider value	U16	rw	1	X	Output value divider
6200h		Cycle timer PDO1	U16	rw	203h		In milliseconds, internally linked to object 1800h-5
6500h		Operating Status	U16	ro	4h		Bit 0: ▪ 0: Position CW ▪ 1: Position CCW Bit 2: ▪ 0: Scaling function disabled ▪ 1: Scaling function enabled
6501h		Single turn resolution [step/rev]	U32	ro	16777216		Maximum measuring units per revolution only 21 bit supported

Object	Subindex	Name	Data type	Access rights	Default	Save	Description
6502h		Number of distinguishable revolutions	U32	ro	65536		Maximum multi-turn revolutions
6503h		Alarms	U16	ro,m	0h		Bit 0: Position step width error
6504h		Supported alarms	U16	ro	1h		Bit 0: Alarm supported
6505h		Warnings	U16	ro,m	0h		Bit 4: Battery Voltage warning
6506h		Supported warnings	U16	ro	10h		Bit 4: Warning supported
6507h		Profile and software version	U32	ro	2010401h		
6508h		Operating time	U32	ro			Operating time in 0.1 hours
6509h		Offset value	I32	ro	0h		Internal offset calculated during the preset process.
650Ah		Module identification	Array				
	00h	Highest sub-index supported	U8	ro	1		
	01h	Manufacturer offset value	I32	ro	0h		
650Bh		Serial number	U32	ro			Internally linked to object 1018h-4h

List of illustrations

623968011

Baumer
Passion for Sensors

Baumer Germany GmbH & Co. KG
Bodenseeallee 7
DE-78333 Stockach
www.baumer.com