SUPREME SENSORING
TWK

Document no.: **NOC** 13100 WE
Date: 08.09.2020

Specifications

Rotary encoders
Switching cam rotary encoders
Slewing ring rotary encoders
High resolution rotary encoders
Type NOCN, HBN and TBN/TRN#

with
CANopen safety / non-safety interface
Profile (S0, S1, S2), S3, S4#, C2, C3

especially *

NOCN-N20/S3 (Version 2)
NOCN-N42/S3 (Version 2)
NOCN-N43/S3 (Version 2)
NOCN-N44/S3 (identical to NOCN-N42 Version 2)
NOCN-N45/S3 (identical to NOCN-N43 Version 2)
NOCN-N46/S3 (identical to NOCN-N20 Version 2 and NOCN-N45/S3)

CANopen EN50325-5
(Formerly CiA DS304)

#: See separate handbook for SIL2 (ISO 61508) certified TBN/TRN/S4: TXN15469
*: Renewed ISO 13849 (PL d) certification in August 2020 only for these types

TWK-ELEKTRONIK GmbH
Bismarckstraße 108
40210 Düsseldorf
Tel.: +49 211 961170
info@twk.de
visit us at | twk.de

Safety instructions

Scope

This specification is valid exclusively for the following absolute encoders / switching cam encoders with CANopen (safety) interface:

- NOCN, HBN and TBN/TRN with safety architecture (SIL2)

Documentation

The following documents must be observed (depending on device):

- The owner's system-specific operating instructions
- This specification NOC13100
- Data sheet numbers NOC13099
  NOC13292
  HBN13218
  TBN/TRN14271
  (and other ones)
- The connection assignment enclosed with every device

Proper use

The TWK-ELEKTRONIK GmbH absolute encoders and linear transducers are used to register angular or linear positions and make their measured value available in the form of an electrical output signal. As part of a system, they have to be connected to the downstream electronics and must only be used for this purpose.

Commissioning

- The relevant device may only be set up and operated in combination with this and the documentation specified above
- Protect the device against mechanical damage during installation and operation.
- Device commissioning and operation may only be undertaken by a specialist electrician.
- Do not operate the device outside of the limit values specified in the data sheet.
- Check all electrical connections before commissioning the system.

First use

For the first use it may be necessary to modify node-ID and baud-rate to match the encoder to customers controller (ex works: ID 1 and 20 kbps). Please see chapter 9.10 (LMT objects) and chapter 9.5.8 (Object 1010 store parameters) for the handling of these parameters. A detailed example is shown in chapter 12.6 and 12.7. For all modifications the device has to be set ***preoperational*** first.

To set the encoder operational after power on 13FE (and perhaps 30FE - cam) has to be set A5.

Reset behaviour of NOCN with cams:

The valid flag 30FE (cams) has to be set 30FE = 0 when a reset of NOCN via 81 is required (complete NOCN reset - reset node).

A reset via 82 (only CANopen communication reset) can be done without setting 30FE = 0.

Table of contents

Table of contents

Table of contents

♦: Only valid for NOCN with cams.
♦♦: Only valid for NOCN with slewing ring functionality

1 Intended use

The rotary encoder resp. switching cam encoder is intended for use in safety-relevant systems.

The version without cams provides a safety rotary encoder in singleturn or multiturn design with a huge number of different flange an shaft designs (See datasheets NOC13099, NOC13292, HBN13218 and TBN/TRN14271, etc.).

Thanks to its robust design, the switching cam encoder can be used in applications with harsh environmental conditions. Thanks to the combination of two relays per cam and the use of corresponding monitoring functions, the device offers two safe electrical/electronic cams. In addition to the safely switching cams, the switching cam encoder can also provide position data like a normal absolute encoder via a CAN interface either securely via CANopen Safety or insecurely via a normal CANopen interface. The switching cam encoder is designed for connection to a control system which evaluates the measured values transmitted via the CAN bus and, in the event of error messages or the absence of process data, reacts according to the relevance of the error and prevents a hazardous state.

Several objects and descriptions are not valid for HBN, TBN/TRN. In case of doubt please ask our service engineers.

Devices of model TBN/TRN/S4 have a separate handbook: TXN15469.

2 Design

The switching cam encoder consists of a redundantly designed position registration module, an evaluation unit, a switching cam encoder module and an absolute encoder module which has a secure CANopen Safety interface and an insecure CANopen interface to the application.

Position registration is carried out through angle and revolution measurement. The signals from encoders used to register the position of the shaft and to scan a mechanical transmission for measuring the revolutions covered are combined to form a summarized position value. The position registration system is equipped with a separate controller. Position registration is designed in fully redundant form. The position data are transferred to the evaluation module via two channels.

The evaluation module evaluates the position data supplied by the position registration system. The measured values are filtered, the two redundant channels are compared and the measurement signals are linearized. If the measured values are recognized as safe, the module makes the position data available to the downstream processes. The evaluation module provides drivers for various position registration units so that various sensor units can operate together with the evaluation module. This module is not used solely to measure angles; other sensor units for measuring variables such as travel, inclination, vibrations, etc. can also be registered and further processed.

The switching cam encoder module provides two securely switching cams (four safety cams at NOCN79. Available on request). The secure position value supplied by the position registration system is evaluated and used to generate the switching states of the two cams. Thanks to the connection of two relays in series and relay status monitoring functions for each individual relay in hardware and software form, the cams ensure safe switching. The cam parameters can be reprogrammed via the CANopen interface. However, programming can also be undertaken in the factory if the switching cam encoder is to be operated independently of the CANopen interface.

The absolute encoder module accepts the position data provided by the evaluation module, undertakes calibration depending on the set parameters and transfers the data to the CANopen interface. Depending on setting, the CANopen interface can transfer both secure data via CANopen Safety and secure data via a normal, insecure CANopen interface to the application. The encoder module is independent of the cam function.

3 Device versions

The device designation nomenclature is defined in the NOC13099, NOC13292, HBN13218 and TBN/TRN14271 data sheet.

3.1 Switching cam encoder

Rotary encoder NOCN including switching outputs. All of the CANopen objects described in these specifications are used including all valid flags (13FE, 30FE and 61FE).

All programming examples in chapter 11 and 12 are used.

3.2 Rotary encoder

Rotary encoder NOCN without switching outputs and high resolution encoder HBN, TBN/TRN. None of the CANopen objects described in these specifications concerning the switching outputs (Cams) are used (e.g. number group 63xx). The cam_valid_flag 30FE is not valid for this version. It is set to 0. To make sure that this valid flag is not set to A5 the checksum in 30FF is set to 0 as well.

The slewing ring functionality parameter object 3100 is not valid as well (See 3.3).

The rotary encoder is available in a singleturn or a multiturn version. The CANopen objects which refer exclusively to the multiturn version are not used (i.e. 6002 total_measuring_range_in_measuring_units).

The programming examples in chapter 11 and 12 are used except the parts which refer to the adjustment of the cams.

3.3 Rotary encoder with slewing ring functionality

Rotary encoder NOCN and TRN with special software for slewing ring applications. The rotary encoder is coupled to a slewing ring gear with a toothed gear. All of the CANopen objects described in these specifications are used on implementation as a switching cam encoder. On implementation as a rotary encoder, none of the CANopen objects concerning the switching outputs are used (e.g. number group 63xx). The cam_valid_flag 30FE is not valid for this version. It is set to 0.To make sure that this valid flag is not set to A5 the checksum in 30FF is set to 0 as well. The slewing ring functionality parameter object 3100 is valid (Numbers of teeth).

As well valid are: 31FE (gear_valid_flag) and 31FF (gear_CRC_checksum). When writing a gear parameter to object 3100, sub xy 31FE is deactivated automatically. Then transmit to the new gear parameter related checksum to object 31FF(calculation with TWK program). At last 31FE has to be activated. The new parameters are set. Saving with object 1010, sub 2.

When the code-sense (CW or CCW) has to be modified, this can be done by 6100, sub 01. In this case (means: encoder with slewing ring functionality) the following valid flags have to be set A5: At first 31FE and then 61FE (if 31FE is read out you see ‘A5’. But it has to be set ‘A5’ anyway).

The programming examples in chapter 11 and 12 are valid except the parts which refer to the adjustment of the cams.

In some applications it is necessary to get directly the angle of the slewing unit.

A rotary encoder is coupled via a gear to a toothed gear or with a worm gear / gear units mounted on the worm shaft. This results in a certain ratio due to the number of teeth.

The software in the encoder can be set (by the factory or the customer) in a way that the output signal of the encoder is the angle position of the slewing unit. CANopen object 3100 is used for this. The angle resolution can be set for example to 0.1° (i.e. 3600 steps per 360° of the encoder). Meaning if the slewing unite turns 360° the output of the encoder will also only turn by 360° (3600 → 0 steps).

Ever in case the slewing unit turns constantly only in one direction the output will not be affected. Meaning the output signal will continue to give angle values between 0 and 360° even for infinite revolutions.

S makes the slewing ring functionality in the order code.

Attention: When power supply of NOCN or TRN is switched off it is not recommended to turn the shaft more than 500 turns in the same direction. Otherwise the safety slewing ring position may not be correct any more.

But when the slewing ring position is not correct any more this is detected by NOCN or TRN. Regularily an error message will be transmitted by the encoder (FF FF 81 00 80 04 02 00) when power supply is switched on again (Exception: Shaft was turned n * 4096 rev.). To delete this error message, do the following:

1. Set valid flag 31FE active (A5). 2. Set valid flag 61FE active (A5). 3. Set valid flag 13FE active (A5). 4. Set encoder operational. If this may not work please do: 1. Load default values by object 1011, sub 01. 2. Power off/ on reset again. 3. Set 13FE active (A5). 4. Set encoder operational.

In every case the output signal and the slewing ring position has to be checked and adjusted again by setting a preset value at an appropriate position.

This remark is only valid for slewing ring functionality.

Adjustable parameters (via CANopen objects)

Adjustable parameters		from	to
Code sequence	6100/01	CW	CCW
Number of teeth - slewing ring	3100/01	1	65536
Number of teeth - pinion of NOCN/TRN	3100/02	1	65536
Measuring range (Resolution position)*	3100/03	1	4096 x i
Resolution for speed *	3100/04	1	4096 x i
Speed integration time [ms]	6101/05 (for speed)	10	1000

i = Gear ratio: *Number of teeth - slewing ring to Number of teeth - pinion of NOCN/TRN*
*= read only / not realized at the time

Example:
Number of teeth - slewing ring 131
Number of teeth - pinion of NOCN: 11
Resolution position *: 36,000
Resolution for speed *: 36,000
Speed integration time: 100

*= read only / not realized at the time

Comparison of some characteristics when the encoder is coupled to the slewing ring

Characteristics	Encoder	Slewing ring
Resolution	4096 steps	Adjustable, max. 4096 x i steps
Accuracy	± 0.2 %	± 0.2 % x 1/i
Measuring range	4096 revolustions	Revolution repeatable ∞ times
Reproducibility	± 0.02 %	± 0.02 % x 1/i
Temperature drift within -40° C to +85° C	< ± 0.02°	< ± 0.02° x 1/i
Internal system position monitoring	3 %	3 % x 1/i

4 Documents

Machinery Directive	2006/42/EC
EN61508	Functional safety of electronic systems
EN13849	Safety of machinery - Safety-related parts of control systems
ISO11898	CAN Interface.
CiA DS 301	Application Layer and Communication Profile Version 4.2.0
EN50325-5	Safety-relevant Communication.
CiA DSP 305	Layer Setting Service.
CiA DSP 310	Framework for Safety-relevant Communication Version 1.01.
CiA DS 406	Device Profile for Encoders, Version 4.0.1 / Version 4.0.2
EN61326-1	Electrical equipment for measurement, control and laboratory use - EMC requirements
EN61326-3	Electrical equipment for measurement, control and laboratory use - EMC requirements - Safety functions

Available test reports for NOCN:

NOC13327	TÜV SIL2/PLd Certificate (ISO 13849 / ISO 61508) *
NOC13328	TÜV SIL2/PLd Certificate Testreport (ISO 13849 / ISO 61508) *
NOC14850	Additional TÜV PLd Certificate Testreport (conformity to ISO 13849:2015) *
NOC14959	Additional TÜV PLd Certificate Testreport #1 for NOCN-N42/S3 (con. to ISO 13849)*
NOC15109	Additional TÜV PLd Certificate Testreport #2 for NOCN-N42/S3 (con. to ISO 13849)*
NOC15806	TÜV PLd Certificate due to ISO 13849, created in August 2020 **
NOC15801	TÜV PLd Certificate Testreport due to ISO 13849, created in August 2020 **

ZE13214	EN60068-2-52	Salt mist cyclic
ZE13215	EN60068-2-52	Additional informations to salt mist cyclic
NOC13431	EN61000-4-4	Burst
NOC13432	EN60068-2-27	Shock
NOC13433	EN60529	Protection grade IP
NOC13434	EN60068-2-6	Vibration
NOC13435	-	Magnetic fields
NOC13436	EN61000-4-2	ESD
NOC13455	EN61000-6-4	EMC Emission
ZE13468	EN60068-2-31	Rough handling
ZE13469	EN60068-2-30	Damp heat cyclic
ZE13470	EN60068-2-78	Damp heat steady state
ZE13471	EN60068-2-78 and EN60068-2-30	Additional informations to damp heat
ZE13473	EN60068-2-1	Cold environment, device not supplied
ZE13474	EN60068-2-1	Cold environment, device supplied
ZE13475	EN60068-2-2	Dry heat, device supplied
ZE13476	EN60068-2-14 (61163-1)	Temperature changes
ZE13477	EN61000-4-29	Voltage dips, short interruptions
ZE13478	EN61000-4-3/-6/-8 (61000-6-2)	Declaration of compliance EMC
ZE13480	EN61000-4-5	Surge
NOC13842	-	Hardware tests in general
ZE14326	IEC60529 / ISO20653	IP grade (valid for NOCN with shaft seal = type series 3)

*: Created Nov. 2014 → Not valid any more since Nov. 2019
**: Valid for NOCN-N20-Version2 (V2) / N42-V2 / N43-V2 / N44 (=N42-V2) / N45 (=N43-V2) / N46 (=N20-V2)

5 Electrical specifications NOCN

5.1 General

Refer to the NOC13099, NOC13292, HBN13218 and/or TBN/TRN14271 data sheet and the variant specifications for the precise electrical specifications.

1.	Supply voltage	9 V ... 36 VDC
2.	Power consumption	< 2.5 W
3.	Temperature range	- 40° C to + 70° C
4.	Communication profile	Full CAN Part A (11-bit) CANopen CiA 301 V 4.2 Encoder profile CANopen CiA 406 V 4.0.1 / V 4.0.2 CANopen Safety EN 50325-5

5.2 Measuring system

1.	Measuring range	4096 revolutions
2.	Resolution	4096 steps per revolution (Option: higher resolution on request)
3.	Accuracy	± 0.25 % (Option: higher accuracy on request)
4.	Temperature drift	± 0.02° at -40 °C to 70° C
5.	Sensor unit cycle time	1 ms

5.3 Cam unit

1.	Maximum switching current	0.5 A at 30 VDC/ AC
2.	Maximum switching voltage	60 VDC / VAC
3.	Response time:	20 ms
4.	Maximum resistance ON	0.5 ohms
5.	Category of use:	AC 14 (Alternating current) DC 13 (Direct current)

5.4 Overall system

1.	Dutycycle (rise time) supply voltage	500 ms (10 % to 90 %)
2.	Message density	10,000 messages / s
3.	Memory cycle density	3 s per memory cycle
4.	Set-up time	2 s at +25° C 20 s at -30°C

5.5 Velocity signal

The encoder provides a velocity signal. It is transmitted via SRDO2. The relationship between the value v in SRDO2 and the value rotations per minute u is:

$$u\ [\text{rev./min}] = \frac{v\ [\text{digits}] \times 60000 \times S_D}{B\ [\text{digits/rev}] \times T\ [\text{ms}] \times S_M}$$

Examples for speed resolutions B:
- NOCN: 2^{12} = 4096
- NOCN79: 2^{16} = 65536
- HBN standard: 2^{16} = 65536
- HBN special versions: 2^{32} = ~4.3 Billion

It means:

u = Shaft speed in revolutions per minute

v = Digits read out from SRDO2 (Remark: The velocity signal in SRDO2 is in the format Signed 16 Bits. At increasing position values you can use v directly. At decreasing position values you have to calculate: *FFFF - v* first, before inserting in the formula. Increasing or decreasing position values depend on the setting of *safety code sequence* - object 6101, sub 01). v_{max}: 15 Bits due to sign of velocity signal.

B = Basic Resolution for velocity signal - depending on device

T = Safety speed integration time in milliseconds - see object 6101, sub 05

S_D = Safety speed divider - see object 6101, sub 07 (if existing, else = 1)

S_M = Safety speed multiplier - see object 6101, sub 06 (if existing, else = 1)

60000 = Compensation factor milliseconds ↔ minutes

5.6 Connection

Communication profile

Full CAN Part A (11-bit) CANopen 301 V 4.1
Encoder CANopen DS 406 V 4.0.1 and V 4.0.2 resp.
CANopen Safety EN 50325-5

With galvanic bus separation

CAN connection assignment: 5-pin M12 male/female, coding A

Pin number	Signal designation
1	CAN_GND
2	+Vs / GND
3	-Vs / GND (in case of non galv. separation pin 3 is connected to pin 1 and/or to shield)
4	CAN high
5	CAN low

Remark: It is recommended not to exceed 30 m of length of connecting cable/line between controller and NOCN. The cable/line is defined as a signal line.

Switching output connection assignment: 8-pin M12, male, coding A - valid for NOCN58/64

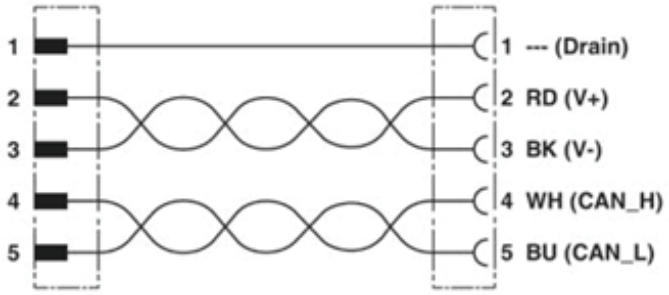
Pin number	Signal designation
1	n.c.
2	Safety contact 1 / (13)
3	n.c.
4	Safety contact 2 / (23)
5	Safety contact 1 / (14)
6	n.c.
7	Safety contact 2 / (24)
8	n.c.

Switching output connection assignment: 8-pin M12, male, coding A - valid for NOCN79

Pin number	Signal designation
1	Safety contact 1 / (13)
2	Safety contact 1 / (14)
3	Safety contact 2 / (23)
4	Safety contact 2 / (24)
5	Safety contact 3 / (33)
6	Safety contact 3 / (34)
7	Safety contact 4 / (43)
8	Safety contact 4 / (44)

See datasheet NOC13099, NOC13292, HBN13218 and TBN/TRN14271 for further possibilities of galvanic separation / non separation due to customer requirements.

Recommended type of CANopen/power supply connecting cable:



6 Environmental data

6.1 Dust, moisture

Permissible relative humidity	100 %
Protection type:	IP 67, optionally IP 69K

6.2 Vibration shock

EN 60068-2-6	Vibration sinus
EN 60068-2-27	Mechanical shock
Resistance to shock	25 g 6 ms each 100x on 3 axes (1 g = 9.81 m/s²)
Resistance to vibration	Amplitude of excursion 5 mm (peak value) or 10 g from 5 to 2000 Hz each 1h on 3 axes

6.3 EMC

EN 61000-6-2	Interference immunity for industrial environments
EN 61000-6-4	Interference emission for industrial environments

See as well chapter 4 for standard test details and related reports for NOCN etc.. The reports are available on request. See datasheets for results for the different encoders.

7 Mechanical system

The mechanical dimensions are specified in data sheet NOC13099, NOC13292, HBN13218 and/or TBN/TRN14271 or in related drawings due to special mechanical designs.

8 CANopen functionality

The baud rates are implemented according to the following table. The baud rate 125 kBit/s is defined as the default setting.

Baud rate table

Oscillator [MHz]	Baud rate [kBit/s]	Number of the units	Sample point	BRP R	SJW	PRS	PHS1	PHS2
50	1000	10	6	5	1	2	4	4
	800	10	6	10	1	2	4	4
	500	10	6	10	1	2	4	4
	250	10	6	20	1	2	4	4
	125	16	9	25	1	2	6	7
	100	16	9	25	1	2	6	7
	50	10	6	100	1	2	4	4
	20	**10**	**6**	**250**	**1**	**2**	**4**	**4**

8.1 Rotary encoder / Switching cam encoder and CANopen

The switching cam encoder is always regarded as a secure system. The CANopen interface enables both secure operation via CANopen Safety and insecure operation via CANopen Standard. Due to the fact that the position data are relevant to the switching cam encoder and the switching cam encoder parameters are set via the CANopen interface, validation of these parameters is necessary even in the case of operation without CANopen Safety. As a result of this, the position parameters are administered in the secured area of the 6100h objects and the cam parameters offer identical validation with a CRC checksum and data valid flag (cam_data_valid) in the cam parameter area.

All actions which influence the behaviour of the cams are only permitted if the system's cam function is shut off by deactivating the cam_data_valid flag 30FE.

Changes in the position and cam parameter area are only possible if the cam function is as well deactivated by the cam_data_valid flag 30FE.

The load default function can only be executed when the cam function is deactivated (30FE).

The cam_data_valid flag (securing the cam parameters) can only be set when the safety application configuration valid flag (securing the position parameters) is active (61FE).

The unsecured LMT command reset (81) can only be executed when the cams are deactivated (30FE=0).

On deactivation of the cam function, the cams are set to secure state (all relays open).

The LMT command reset for CAN communication (82) can be executed anytime.

The SRDO configuration valid flag, object 13FE, can be set irrespective of the state of the cam_data_valid flag (30FE).

8.2 Redundant position registration system

The CANopen profile definition for the encoder is executed once according to CiA 301 application layer and according to EN 50325-5 framework for safety-relevant communication.

The sensor system (position registration) is designed in redundant form. The sensor system's measured values are supplied to the self-monitoring controller on separate interfaces (SPI). This compares the position values of both sensors. If the measured values lie within a specified tolerance, the signals are transferred as validated for downstream evaluation. Otherwise, an error message is generated, the cams are shut off and the controller assumes a secure state (pre-operational). The sensor system's read-in position value is the basis for the cams' switching points and the input parameter for the encoder functions.

8.3 Error behaviour

8.3.1 General

All errors that occur are displayed and stored in a hierarchical structure. In the error register (object 1001 and 1003) is the error type of the whole system coded. In object 6503 the error type is coded. The error code gives a detailed error description depending on the error type. Sensor errors affect both the cam controller as well as the data output of the CANopen Safety system. Both systems are no longer able to work in case a sensor error occurs. Errors in the cam switch put the faulty cam in the safe state. The functional cam continues to be active.

Errors in the cam switch are reported via the CANopen interface with an alarm message to the controller. Through the error_behaviour object 1029 the CAN interface can be set up to be operated independently from the status of the cam switch.

Nevertheless the alarm messages of the cam switch will always be sent.

See Chapter 13 ***Recognition aid for error / emergency messages*** at the end of this document as well.

8.3.2 Fatal errors

Errors which place the functional capability of the controller in doubt - ROM or RAM CRC errors, oscillator drift and watchdog triggering - are not answered with a CAN message. Immediately after detecting the error and shutting off the cams, the controller assumes a secure state (endless loop without actions). This error must be detected and processed by the control system (e.g. hardware device error).

8.3.3 CANopen emergency message

If the encoder has discovered an error, an emergency message is transmitted unless the node is set to STOP state. The error code is additionally entered in the error register and in object 6503. The behaviour in the event of an error is defined in object 1029 error behaviour. In the event of an error, the sensor switches to the NMT state PRE-OPERATIONAL. If an error disappears (CAN channel error), an EMC message is transmitted again with a deleted error bit. The time interval between emergency messages is determined by object 1015 Inhibit Time EMCY. The absolute encoder's error states remain set until reset or power on occurs.

The emergency message has the following structure:

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
EMC error code		Error register	Object 6503		Extended (custom) error code		

EMC error codes:

0x FFFF	Customer-specific error; error in the sensor system or cam switch
0x8120	Passive state error
0x8140	Return from bus off state
0x8110	Overrun error, not all messages can be transmitted by the sensor

Error register codes, see object 1001 and 1003.

A distinction is made between communication errors (bit 4) and vendor-specific errors (bit 7).

Object 6503: Type of error in case of errors in the sensor system or cam switch

Error code: Detailed error code depending on the type of error in object 6503

The data are output on the bus in Intel format.
A distinction is made between two types of error:

1. Errors in the sensor system (error code 0xFFFF)
   All errors which render proper sensor operation impossible.
2. Communication errors (error code 0x81xx)
   Errors due to the bus system; these are not usually caused by the sensor but indicate a malfunction in the bus system.

All sensor errors are critical errors.

The user of the overall system must assess the errors in the bus system and define the reaction to them.

Examples:

EEPROM CRC error

0	1	2	3	4	5	6	7
Error code		Error register	Device-specific error		(Custom) error code		
0xFF	0xFF	0x81	0x00	Obj. 6503 0x20	0x00	0x00	0x00

Passive error

0	1	2	3	4	5	6	7
Error code		Error register	Device-specific error		(Custom) error code		
0x20	0x81	0x11	0x00	0x00	0x00	0x00	0x00

Return from bus off

0	1	2	3	4	5	6	7
Error code		Error register	Device-specific error		(Custom) error code		
0x40	0x81	0x11	0x00	0x00	0x00	0x00	0x00

Expiry of the inhibit time is followed by the message "correct operation":

0	1	2	3	4	5	6	7
Error code		Error register	Device-specific error		(Custom) error code		
0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00

See chapter 13 (Error recognition aid) for a better understanding of the error code structure.

8.4 (Custom) Error codes

8.4.1 Sensor error

Bit 7 is set in the alarm object 6503.
Detailed error description is given in Byte 5, 6 and 7. See below.

Error code

5	6	7
Channel number	Error type	Error code

As the sensor system is redundant, the two channels need to be distinguished in the channel number.

Byte 5: Channel number:

No error	0x00
Error channel 1	0x01
Error channel 2	0x02
Error: No channel	0x03

Byte 6: Error type:

No error	0x00
Error communication	0x01
Error sensor	0x02
Error speed overflow	0x03
Error synchronism	0x04

Byte 7: Error code:
(Error in communication)

No error	0x00
Error time out	0x01
Error CRC	0x02
Error MSG_NBR	0x03
Error position	0x04

(Error of sensor)

CRC error code range	0x06
Chip error of detection ICs	0x0A
Error in gear detection	0x12
Error of parameter	0x42
Error of calibration parameters	0x82

8.4.2 Cam error

Bit 4 is set in the alarm object 6503

5	6	7
Error type, s.b.	see below (s.b.)	always 0

Byte 5: Error type:

No error	0
Error communication	1
Error relay	2

When: Error in communication: 1 in byte 5:

Byte 6:

General error	0x00
No error	0x01
No start flag found	0x02
No end flag found	0x03
Error frame	0x04
CRC error when checked	0x05
Timeout	0x06
Error data length	0x07
Unknown address	0x08
Error message number	0x09

(SLV)

General error	0x80
No error	0x81
No start flag found	0x82
No end flag found,	0x83
Error frame,	0x84
CRC error when checked	0x85
Timeout	0x86
Error data length	0x87
Unknown address	0x88
Error message number	0x89

Byte 7 = 0

When: Relay error: 2 in byte 5:

Byte 6: Shows the real state of the relays

Byte 7: Shows the intended state of the relays

8.4.3 Supply voltage

Bit 6 in object 6503 is set to alarm.

Error code: Too high voltage

5	6	7
00	00	01

Error code: Too high voltage

5	6	7
00	00	02

8.4.4 Device error

Bit 3 in object 6503 is set to alarm.

Runtime error / safety hard beat

5	6	7
01	01	01

Hard error (fatal error)

5	6	7
08	AB	01

See end of document for further explanations concerning error behavior.

9 CANopen profile definition

9.1 Overview

Table of all objects contained in the encoder profile (Objects marked ♦: Only valid for NOCN with cams. Objects marked ♦♦: Only valid for encoders with slewing ring functionality).

Index	Data type	Designation	Data length	Memory type	M / O
Communication Profile Area					
1000	VAR	device_type	LONG	ro	M
1001	VAR	error_register	BYTE	ro	M
1003	ARRAY	pre_defined_error_field	-	ro	O
1005	VAR	COB-ID_SYNC	LONG	rw	O
1008	VAR	manufacturer_device_name	STRING	ro	O
1009	VAR	manufcturer_hardware_version	STRING	ro	O
100A	VAR	manufacturer_software_version	STRING	ro	O
1010	ARRAY	store_parameters	LONG	-	O
1011	ARRAY	restore_default_parameters	LONG	-	O
1014	VAR	COB-ID-EMCY	LONG	rw	O
1015	VAR	inhibit_time_EMCY	LONG	rw	O
1017	VAR	producer_heartbeat_time	WORD	rw	O
1018	RECORD	identity object		ro	M
1029	ARRAY	Error behavior	BYTE	rw	M
Transmit SRDO communication parameters					
1301	RECORD	SRDO1 communication parameters		rw	M
1302	RECORD	SRDO2 speed parameters		rw	M
SRDO mapping objects					
1381	RECORD	SRDO_1_mapping_parameters			
1382	RECORD	SRDO_2_mapping_parameters			
Safety CAN objects					
13FE	VAR	configuration valid	BYTE	rw	M
13FF	ARRAY	safety configuration checksum		rw	M
Transmit PDO communication parameters					
1800	RECORD	PDO1_communication_parameters	-	rw	
1801	RECORD	PDO2_communication_parameters	-	rw	
Transmit PDO mapping objects					
1A00	RECORD	PDO1_mapping_objects	-	ro	
1A01	RECORD	PDO2_mapping_objects	-	ro	
Encoder objects					
6000	VAR	operating parameters	(WORD)	ro†	M
6001	VAR	measuring units per revolution	LONG	ro†	M 2
6002	VAR	total measuring range in measuring units	LONG	ro†	M 2
6003	VAR	preset value	LONG	ro†	M 2
6004	VAR	position value	LONG	ro	M
6030	RECORD	speed_value		ro	O
6031	RECORD	speed parameters	-	ro†	O

† At encoders with CANopen standard profile: read/write rw

Index	Data type	Designation	Data length	Memory type	M / O
Encoder safety objects					
6100	RECORD	safety_position_configuration_parameters	-	rw	M
6101	RECORD	safety_speed_configuration_parameters	-	rw	M
6120	ARRAY	safety_standard_position_value	BYTE	rw	M
6121	ARRAY	safety_inverted_standard_position_value	BYTE	rw	M
6124	ARRAY	safety_speed_value	BYTE	rw	O
6125	ARRAY	safety_inverted_speed_value	BYTE	rw	O
61FE	VAR	safety_configuration_valid	BYTE	rw	M
61FF	ARRAY	safety_configuration_signature		rw	M
Objects for achieving compatibility					
6200		cyclic_timer	WORD	rw	
Encoder cams					
♦ 6300	VAR	cam state register	BYTE	ro	M
♦ 6301	VAR	cam enable register	BYTE	rw	M
♦ 6302	VAR	cam polarity register	BYTE	rw	M
♦ 6310	VAR	cam 1 low limit	LONG	rw	M
♦ 6311	VAR	cam 2 low limit	LONG	rw	M
♦ 6312	VAR	cam 3 low limit *	LONG	rw	M
♦ 6313	VAR	cam 4 low limit *	LONG	rw	M
♦ 6320	VAR	cam 1 high limit	LONG	rw	M
♦ 6321	VAR	cam 2 high limit	LONG	rw	M
♦ 6322	VAR	cam 3 high limit *	LONG	rw	M
♦ 6323	VAR	cam 4 high limit *	LONG	rw	M
♦ 6330	VAR	cam 1 hysteresis	WORD	rw	M
♦ 6331	VAR	cam 2 hysteresis	WORD	rw	M
♦ 6332	VAR	cam 3 hysteresis *	WORD	rw	M
♦ 6333	VAR	cam 4 hysteresis *	WORD	rw	M
Encoder diagnosis objects					
6500	VAR	Operating status	(WORD)	ro	M
6501	VAR	Single-turn resolution	LONG	ro	M
6502	VAR	Number of distinguishable revolutions	WORD	ro	M
6503	VAR	Alarms	WORD	ro	M
6504	VAR	Supported alarms	WORD	ro	M
6506	VAR	Supported warnings	WORD	ro	M
6507	VAR	Profile and software version	LONG	ro	M
6508	VAR	Operating time	LONG	ro	M
6509	VAR	Offset value	LONG	ro	M
650A	RECORD	Module identification		ro	M
650B	VAR	Serial number	LONG	ro	M

* Only at NOCN79 with four cams (this device is available on request)

650D	VAR	Absolute accuracy	BYTE	ro	M
650E	VAR	Device cabability	LONG	ro	M
LMT objects					
2000	VAR	node ID	BYTE	rw	O
2001	VAR	bit_rate	BYTE	rw	O
Manufacturer-specific objects					
♦ 30FE	VAR	cam_data_valid	BYTE	rw	O
♦ 30FF	RECORD	cam_CRC_checksum	WORD	rw	O
♦♦ 31FE	VAR	gear_ data_valid	BYTE	rw	O
♦♦ 31FF	RECORD	gear_CRC checksum	WORD	rw	O
♦♦ 3100	VAR	position_overflow_free_configurati-on_parameters	WORD / LONG	rw	O
3701	VAR	SRDO_control_checksum	WORD	ro	O
3702	VAR	Safety_Encoder_control_checksum	WORD	ro	O
♦ 3703	VAR	Cam_control_checksum	WORD	ro	O
♦♦ 3704	VAR	Gear_control_checksum	WORD	ro	O
♦ 30FD	VAR	cam_self_test	Byte	rw	O

rw Read/write ro Read only wp Factory programming

Table 1 Overview of CANopen objects

9.2 Safety-relevant data object SRDO1

The sensor supplies 24 or 25 significant data bits in unsigned long format. Objects 6120/6121 are output.

Byte 3								Byte 2								Byte 1								Byte 0							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	MSB								Position data															LSB

Byte 3								Byte 2								Byte 1								Byte 0							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
1	1	1	1	1	1	1	1	MSB								Position data inverted															LSB

The position values are transmitted cyclically with the refresh time (object 1301).

9.3 Safety-relevant data object SRDO2

The sensor supplies 16 significant data in unsigned word format for the speed measurement value. Objects 6124/6125 are output on the SRDO2.

Byte 1								Byte 0							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
MSB							Speed								LSB

Byte 1								Byte 0							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
MSB							Speed inverted								LSB

The speed values are transmitted cyclically with the refresh time (object 1302).

9.4 Process data objects PDOs

The measured position and speed values are output in these objects.

Byte 5								Byte 4								Byte 3							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	31	30	29	28	27	26	25	24
MSB							Speed								LSB	0	0	0	0	0	0	0	0

Byte 2								Byte 1								Byte 0							
23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MSB							Position																LSB

Special PDO format at NOCN-N42 / N42-V2 (=N44) and NOCN-N20-V2 (=N46) and NOCN-N43-V2 (=N45).
Remark:Only at NOCN-N42 and N42-V2 (=N44) the PDO is active. At other devices SRDO1/2 is active (p.20).

Byte 7								Byte 6								Byte 5								Byte 4							
23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
MSB								CRC24 over 1. position + 2. sequence number															LSB	MSB Sequence number LSB							

Byte 3								Byte 2								Byte 1								Byte 0							
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MSB								Position																							LSB

For CRC24 is valid: polynomial 0x323009, initial value FFFFFF, no final XOR, MSB first

9.5 Communication service data objects

For every modification of the encoder via SDO communication the device must be set preoperational first.

It is recommended to wait for the response message of the encoder before sending another SDO message. Then you can be sure that the SDO message (e.g. parameterization) is worked out properly, save procedure (1010) as well. Or implement a time constant between the SDO commands which is long enough (e.g. 50 ms).

Please don't make a reset of the device (power OFF/ON or NMT reset) before all values are transmitted and/or saved properly > wait for response of encoder after saving or wait at least 500 ms.

9.5.1 Object 1000 device_type

The encoder types are defined as follows:

Coding	Device type designation
1	Single-turn absolute rotary encoder
2	Multiturn absolute rotary encoder
3	Single-turn absolute rotary encoder with electronic turn count
4	Incremental rotary encoder
5	Incremental rotary encoder with electronic counting
6	Incremental linear encoder
7	Incremental linear encoder with electronic counting
8	Absolute linear encoder
9	Absolute linear encoder with cyclic coding
10	Multi-sensor encoder interface

Device_type structure:

	Byte 0	Byte 1	Byte 2	Byte 3
Device type	Device profile number		Encoder type	
TRN	0x96	0x01	0x02	0x00

device_type

Index	Sub	Description	Length		Memory		Range/value	Action	Default
			COM	MEM	Type	Location			
1000	0	Device_type	Long	Long	ro	ROM	0x0002196	-	-

9.5.2 Object 1001 error_register

Bit	M / O	Designation
0	M	generic error
1	O	current
2	O	voltage
3	O	temperature
4	O	communication error (overrun, error state)
5	O	device profile specific
6	O	Reserved (always 0)
7	O	manufacturer-specific

The error register is the global error register. It summarises all errors in bit 0.
Generic, communication and manufacturer-specific errors are supported. In the event of an error, the generic error bit is always set. The error which has occurred can be read off in object 6503 Alarms.

error_register

Index	Sub	Description	Length		Memory		Range/value	Action	Default
			COM	MEM	Type	Location			
1001	0	error_register	Byte	Byte	ro	RAM	0, 0x11, 0x81	-	-

9.5.3 Object 1003 pre_defined_error_field

All alarm messages transmitted via emergency messages are stored in this object. Sub-index 0 contains the error code of the last emergency message stored. Overwriting sub-index 0 with ZERO deletes the stored messages. This object contains 20 entries at maximum. When this number is exceeded no further error will be stored.

pre-defined error field

Index	Sub	Description	Length		Memory		Range/value	Action	Default
			COM	MEM	Type	Location			
1003	0	Last_error_entry	Byte	Byte	rw	RAM	0, ... 20	-	-
	1	one_stored_error_msg.							
	2	two_stored_error_msg.							
		...							

Structure_pre_defined_error_field

Byte 0	Byte 1	Byte 2	Byte 3
Alarm code	Custom error code		

The error codes are listed in Chapter 8.3 *Error behaviour* and in chapter 13 at the end of this document.

9.5.4 Object 1005 COB-ID-SYNC

Identifier of the sync message transmitted by the master.
No range or plausibility check takes place. 29-bit identifiers are not supported.

COB-ID-SYNC

Index	Sub	Description	Length		Memory		Range/value	Action	Default
			COM	MEM	Type	Location			
1005	0	COB-ID-SYNC	Long	Long	rw	E^2ROM	1...0x7FF	-	0x80

9.5.5 Object 1008 manufacturer_device_name

The name of the device is stored as a string and is output via SDO segment transfer.
Examples: In general: "Encoder TRN Safety"
NOCN-N42: "TRN-N42"
NOCN-N43: "TRN-N43"
NOCN-N44/N45/N46*,... : "Encoder NOCN Safety"

manufacturer_device_name

Index	Sub	Description	Length		Memory		Range/value	Action	Default
			COM	MEM	Type	Location			
1008	0	manufacturer_device_name	String	String	ro	ROM	See above	-	-

9.5.6 Object 1009 manufacturer_hardware_version

Hardware version of the device. It is stored as a string.
It is output via SDO segment transfer.
Examples: In general: "P-0698 P-0693"
NOCN-N42/N43: "P-0693V05_P-0815V01"
NOCN-N44/N45/N46*,... : "P-0827V01_P-0815V02"

manufacturer_hardware_version

Index	Sub	Description	Length		Memory		Range/value	Action	Default
			COM	MEM	Type	Location			
1009	0	manufacturer_hardware_version	String	String	ro	ROM	See above	-	-

9.5.7 Object 100A manufacturer_software_version

Software version of the device. It is stored as a string.
It is output via SDO segment transfer.
Examples: In general: "Safety standard"
NOCN-N42: "TRN V01.06"
NOCN-N43: "TRN V01.01"
NOCN-N44/N45/N46*,... : "NOCN V02.01"

manufacturer_software_version

Index	Sub	Description	Length		Memory		Range/value	Action	Default
			COM	MEM	Type	Location			
100A	0	manufacturer_software_version	String	String	ro	ROM	See above	-	-

NOCN-N44 is identical to new (year 2020) version NOCN-N42-V2. Ditto: N45 = N43-V2 and N46 = N20-V2.

9.5.8 Object 1010 store_parameters

By inputting "save" (0x65766173hex resp. 1702257011dez) as a password in the relevant sub-index, the writeable objects are saved to the E²PROM or flash memory.
The object cannot be changed on writing. Reading the object is possible.
1 (saving through command, Page 93 CiA 30) is returned.

Which parameters are stored is defined by specifying the sub-index.

Sub-index 01 Storage of all parameters except the 0x2000 to 0x2FFF range.
Sub-index 02 Storage of communication parameters 0x1000 to 0x1FFF.
Sub-index 03 Storage of parameters 0x6000 to 0x9FFF defined in the profile.
Sub-index 04 Storage of the manufacturer-specific range 0x2000 to 0x2FFF (common).
Sub-index 05 Storage of the manufacturer-specific range 0x3000 to 0x3FFF (cams).

store_parameters

Index	Sub	Description	Length		Memory		Range/value	Action	Default
			COM	MEM	Type	Location			
1010	0	largest_supported_ sub-index	-	-	ro	ROM	5	-	-
	1	save_all_parameters (except node ID and baudrate)	Long	Long	rd/(rw)	ROM	"save" =0x65766173	*	1
	2	save_communication_parameters	Long	Long			"save"	*	1
	3	save_application_ parameters	Long	Long			"save"	*	1
	4	save_LMT_parameters (only node ID, baudrate)	Long	Long			"save"	*	1
	5	Save_manufacturer_parameters	Long	Long			"save"	*	1

* Parameters are backed-up in the E²PROM on inputting the correct password (save = 65 76 61 73).

→ **Wait for response of NOCN after saving before resetting the device or wait at least 0.5 to 1 second.**

9.5.9 Object 1011 restore_default_parameters

On inputting "load" (0x64616F6Chex resp. 1684107116dez) as the password in the relevant sub-index, the default parameters are loaded into the objects of the number group selected with the sub-index in the RAM. Reading the object is possible.
The device has to be set “preoperational” at first.

Attention:

The command can only be executed if the device's cam function has been deactivated by deleting (Set 0) the cam_data_valid object 30FEh flag.

1 (device restores parameters) is returned.

Which parameters are loaded is defined by specifying the sub-index.

Sub-index 01 Loading of all parameters except the 0x2000 to 0x2FFF range.
Sub-index 02 Loading of communication parameters 0x1000 to 0x1FFF.
Sub-index 03 Loading of parameters 0x6000 to 0x9FFF defined in the profile.
Sub-index 04 Loading of the manufacturer-specific range 0x2000 to 0x2FFF.
Sub-index 05 Loading of the manufacturer-specific range 0x3000 to 0x3FFF.

restore_default_parameters

Index	Sub	Description	Length		Memory		Range/value	Action	Default
			COM	MEM	Type	Location			
1011	0	largest_supported_ sub-index	-	-	ro	ROM	5	-	-
	1	load_all_default_ parameters (except node ID and baudrate)	Long	Long	rd/(rw)	ROM	"load" =0x64616F6C	*	1
	2	load_communica-tion_parameters	Long	Long			"load"	*	1
	3	load_application_ parameters	Long	Long			"load"	*	1
	4 **	load_LMT_parame-ters (only node ID, baudrate)	Long	Long			"load"	*	1
	5	load_manufacturer_ parameters	Long	Long			"load"	*	1

* On inputting the correct password (load=6C 6F 61 64), the default parameters are loaded from the ROM. In case for a complete load default (COB IDs included): Subindex 1 and 4 have to be executed.

** When loading the default values: For all default node IDs which exceed 32 dez (33 ….) the related COB ID 1 and COB ID 2 have to be entered manually in objects 1301, sub 05 / sub 06 and 1302, sub 05 / sub 06. Otherwise the SRDOs are disabled (e.g. 0x8000 0141 / 0x8000 0142, chap. 9.6). For all node IDs up to 32 dez the COB IDs are adopted by the encoder automatically. (See as well objects 1301, 1302 and 2000 and example 12.6).

9.5.10 Object 1014 COB-ID-EMCY

Identifier for the emergency message which the encoder transmits on occurrence of an alarm.
After "Load default", the identifier is COB-ID-EMCY + node ID.
If the user changes the COB ID, the node address is no longer added.
No range or plausibility check takes place.
29-bit identifiers are not supported.

COB-ID-EMCY

Index	Sub	Description	Length		Memory		Range/ value	Action	Default
			COM	MEM	Type	Location			
1014	0	COB-ID-EMCY	Long	Long	rw	E^2PROM	-	*	0x80+node ID

* Default state evaluation, then addition of the node address.

9.5.11 Object 1015 inhibit_time_EMCY

Blocking time to limit the bus load in the event of EMCY messages in quick succession. The resolution is 100 µs per digit.

inhibit_time_EMCY

Index	Sub	Description	Length		Memory		Range/value	Action	Default
			COM	MEM	Type	Location			
1015	0	inhibit_time_EMCY	Word	Word	rw	E^2PROM	0...0xFFFF	-	1000

9.5.12 Object 1017 producer_heartbeat_time

If a value greater than zero is entered here, the heartbeat message is transmitted on the identifier guard COB ID + node ID in the producer_heartbeat_time interval in ms.

producer_heartbeat_time

Index	Sub	Description	Length		Memory		Range/value	Action	Default
			COM	MEM	Type	Location			
1017	0	producer_heartbeat_ time	Word	Word	rw	E²PROM	0...0xFFFF	-	0

The format of the heartbeat message:

Bit No.	7	6	5	4	3	2	1	0
Content	0	Subscriber status						

0: BOOT-UP
4: STOPPED
5: OPERATIONAL
127: PRE-OPERATIONAL

9.5.13 Object 1018 identity_object

This object contains data assigned to the individual encoder. The object is the address for the Layer Setting Service (LSS).
The following data must be entered:

1. Manufacturer ID — Assigned by CiA
2. Product code — TWK-internal
3. Revision number — TWK software revision number
4. Serial number

The serial number can be written via LSS in factory programming state.

identity_object

Index	Sub	Description	Length		Memory		Range/value	Action	Default
			COM	MEM	Type	Location			
1018	0	largest_supported_ sub-index	-	-	ro	ROM	4	-	-
	1	vendor ID	Long	Long	ro	ROM	0x0000 010D	-	-
	2	product_code	Long	Long	ro	ROM	0x0000 xxxx*	-	-
	3	revision_number	Long	Long	ro	ROM	0x0001 0001	-	-
	4	serial_number	Long	Long	ro(rw)	E²PROM	0	**	-

* depends on device, see TYxxxxx (e.g. 6320 for NOCN, 6302 for TRN), ** Written in factory programming state.

At newer devices (e.g. NOCN-N44/N45/N46/...) this parameter displays the TY number for explicit definition. TY number is valid for the connecting diagram and the EDS file number and unique for every device model, e.g. variant.

Examples:

NOCN58-KA-4096R4096S3S1V1N44 (= NOCN58-KA-4096R4096S3S1V1N42-V2): 0x3D42 (=TY15682)
NOCN64-KZA2-4096R256S3S2V1N45 (= NOCN64-KZA2-4096R256S3S2V1N43-V2): 0x3D43 (=TY15683)

9.5.14 Object 1029 error behaviour_object

This object defines the behaviour in the event of an error. The sub-indices are assigned to the error types as follows:

Sub-index	Error type	Bit in error register 1001	Bit in alarm object 6503
1	Communication error	4	-
2	CRC error	7	5
	Supply out of range	7	6
	Sensor error	7	7
3 **	Cam error	7	4
4 *	Redundancy error	7	3

The following settings in object 1029 are possible for the behaviour in the event of an error:

Value	Description
0x00	Switch from OPERATIONAL to PRE-OPERATIONAL
0x01	No NMT status switch
0x02	Switch to STOP state
0x03	Reserve
::::::::	
0x7F	Reserve
0x80	Manufacturer-specific
::::::::	
0xFF	Manufacturer-specific

error_behaviour

Index	Sub	Description	Length		Memory		Range/value	Action	Default
			COM	MEM	Type	Location			
1029	0	largest_supported_ sub-index	-	-	ro	ROM	3	-	-
	1	communication_error	Byte	Byte	rw	E²PROM	0, 1, 2	-	0x00
	2	internal_device_error	Byte	Byte	ro	E²PROM	0	-	-
**	3	cam_error	Byte	Byte	ro	E²PROM	1	-	-
*	4	redundancy_error	Byte	Byte	rw	E²PROM	0, 1, 2	-	0x01

* Valid for full redundant encoder systems (i.e. TBN-N97). Value = 0x01 means: no change of NMT status. This error occurs when one of the CAN nodes detects that the other system doesn't react any more. Reason can be a defective hardware or an NMT reset of one of the systems. A reset of the two nodes one after the other has to be avoided during setup. After boot up the two nodes will be started via NMT_Reset and via NMT_Start the nodes will be set operational (See trace at the end of this document).

** Only for NOCN devices with switching contacts (cams).

9.6 CANopen Safety process data objects

The process data are output via two Safety Relevant Data Objects (SRDOs).

Attention:

The SRDO is only active when the object configuration_valid (object 13FE) is set (written with the datum 0xA5). The configuration_valid object is stored in the E²PROM. The flag is deleted if one of the Safety Relevant Data Objects is changed.

COB ID structure

MSB										LSB
EN	x	x	x	x	x	x	x	0	COB ID high	COB ID low

The MSB represents the enable bit.
Bit 31 = 0 (**EN**) SRDO enabled
Bit 31 = 1 (**EN**) SRDO disabled (=0x80000”FF+2xID” resp. 0x80000”100+2xID”)

The plausibility of the other bits is not checked. 29-bit identifiers are not supported.

Only transmission type 254 is supported (Cyclic output with the defined refresh time, see object 1301 / 1302).

9.6.1 Object 1301 SRDO1_communication_parameters

The object under this index is only writeable in PRE-OPERATIONAL state.
Modifications will cause 13FE = ‘0’ (must be set ‘A5’ after modification). New CRC Checksum 13FF/01 must be calculated and transmitted to the encoder.

The configuration_valid byte is reset after each write access.

SRDO_communication_parameters

Index	Sub	Description	Length		Memory		Range/ value	Action	Default
			COM	MEM	Type	Location			
1301	0	largest_supported_ sub-index	-	-	ro	ROM	6	-	-
	1	information_direction	Byte	Byte	ro	ROM	1	-	-
	2	refresh_time (SCT)	Word	Word	rw	E²PROM	1..65,535 [ms]	-	25
	3	validation_time (SRVT)	Byte	Byte	ro	ROM	20 [ms]	-	-
	4	transmission_type	Byte	Byte	ro	ROM	254	-	-
	5 **	COB-ID_1	Long	Long	rw	E²PROM	257..384	*	0xFF + (2x node ID)
	6 **	COB-ID_2	Long	Long	rw	E²PROM	257..384	*	0x100 + (2x node ID)

* Default state evaluation, then addition of the node address. After overwriting, addition is no longer executed. The configuration_valid byte 13FE is reset. No plausibility check is undertaken for the COB IDs.

** For all node IDs which exceed 32 dez (33 ….) the related COB ID 1 and COB ID 2 have to be entered manually in objects 1301, sub 05 / sub 06 and 1302, sub 05 / sub 06. Otherwise the SRDOs are disabled (e.g. 0x8000 0141 / 0x8000 0142, chap. 9.6). For all node IDs up to 32 dez the COB IDs are adopted by the encoder automatically. This has to be considered as well when the default value of the node ID exceeds 32 dez and the default values are loaded by 1011, sub 04. (See as well objects 1302, 1011 and 2000 and example 12.6).

Only transmission type 254 is supported (Cyclic output with the defined refresh time, see object 1301, sub 02).

To shut off the SRDO, both COB IDs must be disabled. If only one of the two COB IDs is disabled, setting the configuration_valid flag is not possible.

9.6.2 Object 1302 SRDO2_speed_parameters

The object under this index is only writeable in PRE-OPERATIONAL state. Modifications will cause 13FE = '0' (must be set 'A5' after modification). New CRC Checksum 13FF/02 must be calculated and transmitted to NOCN.

The configuration_valid byte is reset after each write access.

SRDO_speed_parameters

Index	Sub	Description	Length		Memory		Range/ value	Action	Default
			COM	MEM	Type	Location			
1302	0	largest_supported_ sub-index	-	-	ro	ROM	6	-	-
	1	information_direction	Byte	Byte	ro	ROM	1	-	-
	2	refresh_time (SCT)	Word	Word	rw	E²PROM	1..65,535	-	25 [ms]
	3	validation_time (SRVT)	Byte	Byte	ro	ROM	20 [ms]	-	-
	4	transmission_type	Byte	Byte	ro	ROM	254	-	-
	5 **	COB-ID_1	Long	Long	rw	E²PROM	257..384	*	0x13F + (2x node ID)
	6 **	COB-ID_2	Long	Long	rw	E²PROM	257..384	*	0x140 + (2x node ID)

* Default state evaluation, then addition of the node address. After overwriting, addition is no longer executed. The configuration_valid byte is reset. No plausibility check is undertaken for the COB IDs.

** For all node IDs which exceed 32 dez (33 ….) the related COB ID 1 and COB ID 2 have to be entered manually in objects 1301, sub 05 / sub 06 and 1302, sub 05 / sub 06. Otherwise the SRDOs are disabled (e.g. 0x8000 0181 / 0x8000 0182, chap. 9.6). For all node IDs up to 32 dez the COB IDs are adopted by the encoder automatically. This has to be considered as well when the default value of the node ID exceeds 32 dez and the default values are loaded by 1011, sub 04. (See as well objects 1301, 1011 and 2000 and example 12.6).

Only transmission type 254 is supported (Cyclic output with the defined refresh time, see 1302, sub 02).
To shut off the SRDO, both COB IDs must be disabled. If only one of the two COB IDs is disabled, setting the configuration_valid flag is not possible.

9.7 CANopen process data objects

9.7.1 Object 1800 Transmit PDO asynchronous

All asynchronous and cyclical events are processed by this object.

The cycle timer, object 6200, acts on this PDO and is logically equivalent to the event timer.
Transmission type 252 enables synchronous data acceptance.
No plausibility check is undertaken for the COB ID.

The inhibit time has a resolution of 100 µs. The event timer has a resolution of 1 ms.

Transmit_PDO_1

Index	Sub	Description	Length		Memory		Range/value	Action	Default
			COM	MEM	Type	Location			
1800	0	largest_supported_ sub-index	-	-	ro	ROM	5	-	-
	1	COB ID	Long	Long	rw	E²PROM	-	*	0x180+ID
	2	transmission type	Byte	Byte	rw	E²PROM	252, 253, 254	-	253
	3	inhibit time	Word	Word	rw	E²PROM	0..65,535 [ms]	-	0
	4	reserved	-	-	-	-	-	-	-
	5	event_timer	Word	Word	rw	E²PROM	0..65,535 [ms]	-	0

* On reading (upload), the node address is added to the selected identifier.

252: Value is picked-up with sync and transmitted with remote frame (cyclic transmission via object 6200)
253: Value is picked-up and transmitted with remote frame (cyclic transmission via object 6200)
254: Value is picked-up and transmitted with every change of value

Disable PDO1 with 0x80000180+ID as COB ID

9.7.2 Object 1801 Transmit PDO synchronous

All **synchronous** events are processed via this PDO.
The inhibit timer is not implemented for this PDO, as no bus overload can occur in the case of synchronous data output.
No plausibility check is undertaken for the COB ID.

Nor is the event_timer implemented, as it is not necessary for synchronous data output.

Transmit_PDO_2

Index	Sub	Description	Length		Memory		Range/value	Action	Default
			COM	MEM	Type	Location			
1801	0	largest_supported_ sub-index	-	-	ro	ROM	2	-	-
	1	COB ID	Long	Long	rw	E²PROM	-	*	0x280+ID
	2	Transmission type	Byte	Byte	rw	E²PROM	0....240 **	-	1

* On reading (upload), the node address is added to the selected identifier.

** 0 ≤ n ≤ 240: Every nth sync-message the PDO2 is transmitted.

Disable PDO2 with 0x80000280+ID as COB ID

9.8 Mapping objects

9.8.1 Object 1381 SRDO1_mapping_parameters

The parameter contains the following coding for each "mapping" object:

Byte 0	Byte 1	Byte 2	Byte 3
Index		Sub-index	Length

The length is specified as the number of bits in hex coded form.

SRDO_1_mapping_parameters

Index	Sub	Description	Length		Memory		Range/value	Action	Default
			COM	MEM	Type	Location			
1381	0	largest_supported_sub_index	-	-	ro	ROM	8	-	-
	1	first_SRDO_mapping_object	Long	Long	ro	ROM	0x6120 0108	-	-
	2	second_SRDO_mapping_object	Long	Long	ro	ROM	0x6121 0108	-	-
	3	third_SRDO_mapping_object	Long	Long	ro	ROM	0x6120 0208	-	-
	4	fourth_SRDO_mapping_object	Long	Long	ro	ROM	0x6121 0208	-	-
	5	fifth_SRDO_mapping_object	Long	Long	ro	ROM	0x6120 0308	-	-
	6	sixth_SRDO_mapping_object	Long	Long	ro	ROM	0x6121 0308	-	-
	7	seventh_SRDO_mapping_object	Long	Long	ro	ROM	0x6120 0408	-	-
	8	eighth_SRDO_mapping_object	Long	Long	ro	ROM	0x6121 0408	-	-

9.8.2 Object 1382 SRDO2_mapping_parameters

The parameter contains the following coding for each "mapping" object:

Byte 0	Byte 1	Byte 2	Byte 3
Index		Sub-index	Length

The length is specified as the number of bits in hex coded form.

SRDO_2_mapping_parameters

Index	Sub	Description	Length		Memory		Range/value	Action	Default
			COM	MEM	Type	Location			
1382	0	largest_supported_sub_index	-	-	ro	ROM	4	-	-
	1	first_SRDO_mapping_object	Long	Long	ro	ROM	0x6124 0108	-	-
	2	second_SRDO_mapping_object	Long	Long	ro	ROM	0x6125 0108	-	-
	3	third_SRDO_mapping_object	Long	Long	ro	ROM	0x6124 0208	-	-
	4	fourth_SRDO_mapping_object	Long	Long	ro	ROM	0x6125 0208	-	-

9.8.3 Object 1A00 transmit_PDO_1_mapping

The parameter contains the following coding for each "mapping" object:

Byte 0	Byte 1	Byte 2	Byte 3
Index		Sub-index	Length

The length is specified as the number of bits in hex coded form.

transmit_PDO_1_mapping *

Index	Sub	Description	Length		Memory		Range/value	Action	Default
			COM	MEM	Type	Location			
1A00	0	largest_supported_sub_index	-	-	ro	ROM	2	-	-
	1	PDO1_mapping_object1	Long	Long	ro	ROM	0x6004 0020	-	-
	2	PDO1_mapping_object2	Long	Long	ro	ROM	0x6030 0110	-	-

Different mapping at NOCE-N42 / N42-V2 (=N44) due to special PDO (see page 21).

9.8.4 Object 1A01 transmit_PDO_2_mapping

The parameter contains the following coding for each "mapping" object:

Byte 0	Byte 1	Byte 2	Byte 3
Index		Sub-index	Length

The length is specified as the number of bits in hex coded form.

transmit_PDO_2_mapping

Index	Sub	Description	Length		Memory		Range/value	Action	Default
			COM	MEM	Type	Location			
1A01	0	largest_supported_ sub_index	-	-	ro	ROM	2	-	-
	1	PDO2_mapping_ object1	Long	Long	ro	ROM	0x6004 0020	-	-
	2	PDO2_mapping_ object2	Long	Long	ro	ROM	0x6030 0110	-	-

Different mapping at NOCE-N42 / N42-V2 (=N44) due to special PDO (see page 21).

9.9 Safety CAN objects

9.9.1 Object 13FE configuration_valid

This parameter is reset (= 0) each time a "safety relevant parameter" is accessed. Entering 0xA5 switches the configuration to valid. In the case of an invalid value (not 0 or 0xA5) in the configuration_valid flag or incorrect setting of the F parameters, write access is rejected and the SRDOs are not transmitted in OPERATIONAL mode.

Attention:

The flag is automatically reset by writing to the safety position data in area 61xx$_h$.
The flag can only be activated if the data valid flag of the safety position parameter safety_configuration_valid, object 61FE$_h$, is activated.

The parameter is stored in the E²PROM.

configuration_valid

Index	Sub	Description	Length		Memory		Range/value	Action	Default
			COM	MEM	Type	Location			
13FE	0	configuration_valid	Byte	Byte	rw	E²PROM	0 / 0xA5	-	0x0*

* Standard setting is 0. In some special cases this valid flag is set A5 ex works.

9.9.2 Object 13FF safety_configuration_checksum

This parameter contains the checksum crosswise through the safety CAN parameters in objects 1301 and 1302 according to the table below. The checksum is written by the master. The checksum is checked on setting the configuration_valid flag. If no correspondence with the checksum stored in this object is ascertained, setting the flag is blocked (configuration_valid remains ZERO). To calculate the CRC checksum you can use the special TWK program.

To every CRC checksum object xxFF exists a second CRC control checksum object 370x, where the CRC is calculated with a different CRC16 polynomial. This second CRC therefore differs in comparison to the first CRC and is read only. It is called ..._**control**_... . The second CRC can be read by the PLC to make sure on a second path/check that the encoder accepted the new parameters. Only the main CRC xxFF has to be written when parameters were modified. The control CRC can be calculated with TWK CRC program 2.0.5 and the related xml file. This feature is available from end of year 2019 on.

Link to the program on www.twk.de: Look at device NOCN/HBN etc. → “Download” and “Software Safety CRC”. Or use the following link: www.twk.de/files/CRC-Calculator20.zip
Description file: CRC14076. Please ask our technicians for xml files for special versions of encoders or look at www.twk.de.

Attention: The parameter can only be written if the data valid flag of the safety position parameter safety_configuration_valid, object 61FE, is activated.

safety_configuration_checksum

Index	Sub	Description	Length		Memory		Range/value	Action	Default
			COM	MEM	Type	Location			
13FF	0	largest_supported_ sub-index	Byte	Byte	ro	ROM	4	-	4
	1	SRDO1_checksum (for changes in object 1301)	Word	Word	rw	E²PROM	0 ... 0xFFFF	-	0x....*
	2	SRDO2_checksum (for changes in object 1302)	Word	Word	rw	E²PROM	0 ... 0xFFFF	-	0x....*

* Depends on default values in objects 1301 and 1302. Control CRC object: 3701.

Pay attention to the different values of the mapping objects and COB IDs which have to be chosen for the CRC checksums for sub 1 and sub 2.

See examples for parameterization at the end of this document.

SRDO1 checksum calculation scheme, calculation type: MSB – first.

Object	Length	Value	Generator polynomial
information_direction	8	1301/1	
refresh_time (SCT)	16	1301/2	
validation_time (SRVT)	8	1301/3	
COB-ID_1	32	1301/5	
COB-ID_2	32	1301/6	
largest_supported_sub-index	8	1381/0	
	8	1	$2^{16} + 2^{12} + 2^5 + 1$ = 0x11021 for first (main) CRC
first_SRDO_mapping_object	32	1381/1	
	8	2	
second_SRDO_mapping_object	32	1381/2	$2^{16} + 2^{14} + 2^1 + 1$ = 0x14003 for second CRC
	8	3	
third_SRDO_mapping_object	32	1381/3	
	8	4	
fourth_SRDO_mapping_object	32	1381/4	
	8	5	
fifth_SRDO_mapping_object	32	1381/5	
	8	6	
sixth_SRDO_mapping_object	32	1381/6	

Calculation of the checksum:

Generator polynomial:	$2^{16} + 2^{12} + 2^5 + 1$ = 0x11021
2nd generator polynomial for control checksum:	$2^{16} + 2^{14} + 2^1 + 1$ = 0x14003
Initial value:	0x00 00
Final XOR:	no
Details:	See example for object 31FF

9.10 LMT objects

9.10.1 Object 2000 node ID

The node address of the encoder.

The parameter only becomes effective after saving with object 1010, sub 04 and a power on / NMT reset.

Remark: It is not possible to save this parameter with object 1010, sub 01 (save all parameters).

The parameter is not reset to the default value with load_default 1011, sub 01 but with 1011, sub 04.

For all node IDs which exceed 32 dez (33 ….) the related COB ID 1 and COB ID 2 have to be entered manually in objects 1301, sub 05 / sub 06 and 1302, sub 05 / sub 06. Otherwise the SRDOs are disabled (i.e. 0x8000 0122 / 0x8000 0123, chap. 9.6). For all node IDs up to 32 dez the COB IDs are adopted by the encoder automatically. This has to be considered as well when the default value of the node ID exceeds 32 dez and the default values are loaded by 1011, sub 04. (See as well objects 1301, 1302 and 1011 and example 12.6).

node ID

Index	Sub	Description	Length		Memory		Range/value	Action	Default
			COM	MEM	Type	Location			
2000	0	node ID	Byte	Byte	rw	E²PROM	1 ...127	-	0x01

9.10.2 Object 2001 bit_rate

Baud rate of the CAN bus.

This object can also be changed using the Layer Setting Service.

The bit rate index is set according to the following table:

Index	Baud rate [kBaud/s]
0	1000
1	800
2	500
3	250
4	125
5	100
6	50
7	**20**

The parameter only becomes effective after saving with object 1010, sub 04 and a power on reset.
Remark: It is not possible to save this parameter with object 1010, sub 1 (save all parameters).

The parameter is not reset to the default value with load_default 1011, sub 01 but with 1011, sub 04.

bit_rate

Index	Sub	Description	Length		Memory		Range/value	Action	Default
			COM	MEM	Type	Location			
2001	0	bit_rate	Byte	Byte	rw	E²PROM	0 ...7	-	7/3*

* Depending on device. New Safety products have 250 kbaud as default.

9.11 Manufacturer-specific objects

9.11.1 ♦ Object 30FE cam_valid_flag

This parameter is to be reset (= 0) with each write access to a cam parameter or on changing the position registration parameters in object 6100. Entering 0xA5 switches the configuration to valid when the following prerequisites are met:

1. The data valid flag of the position registration system, object 61FE, is set to active (0xA5).
2. The cam parameter checksum is correct.

Otherwise, write access is rejected and the SRDOs are not transmitted when requested.

The parameter is stored in the E²PROM. The standard default value is 0.

Setting the cam_valid_flag blocks write access to the safety position data, objects 61xx.

This valid flag has to be set 30FE = 0 when a reset of NOCN via 81 is required (complete NOCN reset). A reset via 82 (only CANopen communication) can be done without setting 30FE = 0.

cam_valid_flag

Index	Sub	Description	Length		Memory		Range/value	Action	Default
			COM	MEM	Type	Location			
30FE	0	cam_valid_flag	Byte	Byte	rw	E²PROM	0 / 0xA5	-	0x0

9.11.2 ♦ Object 30FF cam_CRC_checksum

The object contains the checksum across the cam parameters in objects 63xx. The checksum must be rewritten in the event of changes by the user. The checksum is checked on setting the configuration_valid flag. If no correspondence with the checksum stored in this object is ascertained, setting the flag is blocked (cam_valid_flag remains ZERO). To calculate the CRC checksum you can use the special TWK program.

To every CRC checksum object xxFF exists a second CRC control checksum object 370x, where the CRC is calculated with a different CRC16 polynomial. This second CRC therefore differs in comparison to the first CRC and is read only. It is called ..._**control**_... . The second CRC can be read by the PLC to make sure on a second path/check that the encoder accepted the new parameters. Only the main CRC xxFF has to be written when parameters were modified. The control CRC can be calculated with TWK CRC program 2.0.5 and the related xml file. This feature is available from end of year 2019 on.

Link to the program on www.twk.de: Look at device NOCN/HBN etc. → download “Software Safety CRC”.
Or use the following link: www.twk.de/files/CRC-Calculator20.zip. Description file: CRC14076
Please ask our technicians for xml files for special versions of encoders or look at www.twk.de.

cam_CRC_checksum

Index	Sub	Description	Length		Memory		Range/value	Action	Default
			COM	MEM	Type	Location			
30FF	0	largest_supported_ sub-index	Byte	Byte	ro	ROM	2	-	2
	1	cam_safety_configu-ration_signature	Word	Word	rw	E²PROM	0 … 0xFFFF	-	0x....*

* Depends on default values in various cam objects 63xx. Control CRC checksum: object 3703.

Calculation of the checksum:

Generator polynomial:	$2^{16} + 2^{12} + 2^5 + 1 = 0x11021$
2nd generator polynomial for control checksum:	$2^{16} + 2^{14} + 2^1 + 1 = 0x14003$
Initial value:	0x00 00
Final XOR:	no
Details:	See example for object 31FF

9.11.3 ♦♦ Object 31FE gear_valid_flag

This parameter is to be reset (= 0) with each write access to a gear parameter. Entering 0xA5 switches the configuration to valid when the following prerequisites are met: The gear parameter checksum is correct.

Otherwise, write access is rejected and the SRDOs are not transmitted when requested.

The parameter is stored in the E²PROM.

gear_valid_flag

Index	Sub	Description	Length		Memory		Range/value	Action	Default
			COM	MEM	Type	Location			
31FE	0	gear_valid_flag	Byte	Byte	rw	E²PROM	0 / 0xA5	-	0xA5

9.11.4 ♦♦ Object 31FF gear_CRC_checksum

The object contains the checksum across the gear parameters in object 3100. The checksum must be rewritten in the event of changes by the user. The checksum is checked on setting the configuration_valid flag 31FE. If no correspondence with the checksum stored in this object is ascertained, setting the flag is blocked (slewing_ring_valid_flag remains ZERO). To calculate the CRC checksum you can use the special TWK program.

To every CRC checksum object xxFF exists a second CRC control checksum object 370x, where the CRC is calculated with a different CRC16 polynomial. This second CRC therefore differs in comparison to the first CRC and is read only. It is called ..._**control**_... . The second CRC can be read by the PLC to make sure on a second path/check that the encoder accepted the new parameters. Only the main CRC xxFF has to be written when parameters were modified. The control CRC can be calculated with TWK CRC program 2.0.5 and the related xml file. This feature is available from end of year 2019 on.

Link to the program on www.twk.de: Look at device NOCN/HBN etc. → download “Software Safety CRC”.
Or use the following link: www.twk.de/files/CRC-Calculator20.zip. Description file: CRC14076
Please ask our technicians for xml files for special versions of encoders or look at www.twk.de.

gear_CRC_checksum

Index	Sub	Description	Length		Memory		Range/value	Action	Default
			COM	MEM	Type	Location			
31FF	0	largest_supported_sub-index	Byte	Byte	ro	ROM	2	-	2
	1	gear_safety_configuration_signature	Word	Word	rw	E²PROM	0 … 0xFFFF	-	0x....*

* Depends on default values in object 3100, sub 01 to sub 03. Control CRC checksum: object 3704.

The default values are:
3100/00 → 0x03
3100/01 → 0x0064
3100/02 → 0x000A
3100/03 → 0x8CA0

Calculation of the checksum:

Generator polynomial of grade 16: $2^{16} + 2^{12} + 2^5 + 1$ (= 1 0001 0000 0010 0001 bin = 0x(1)1021 hex)

2nd generator polynomial for control checksum: $2^{16} + 2^{14} + 2^1 + 1$ (= 1 0100 0000 0000 0011 bin = 0x(1)4003 hex)

Initial value: 0x00 00

Final XOR: No

The calculation processes the objects in ascending direction. All sub-indices which are CRC relevant have to be taken into account. It may be not every sub index of a CAN-object. The resulting value is processed according to the following sequence:

- Value of sub-index 0 0x03
- Number of first relevant sub-index 0x01
- Value of first relevant sub-index 0x0064
- Number of second relevant sub-index 0x02
- Value of second relevant sub-index 0x000A
- Number of third relevant sub-index 0x03
- Value of third relevant sub-index 0x8CA0

If the CRC contains multiple objects, the related sequences have to be stringed together:
sequence object 1, sequence object 2, ...

Example:

The following data array in this sequence has to be used for the CRC calculation.

We take object 3100, sub 00 to sub 03 with the default values (see below).

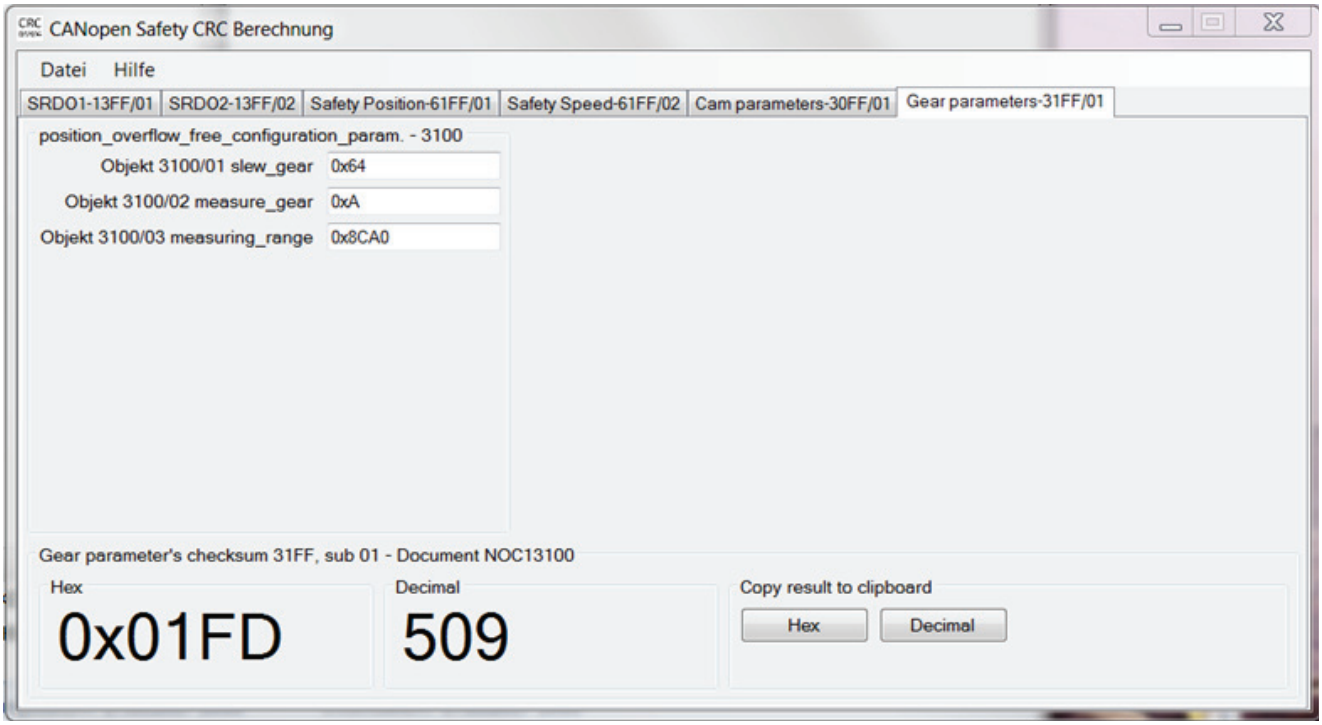
Data in hex, data length included (e.g. 'Word' 100dez = 0064hex).

Low byte before high byte (e.g. 0064 → 6400).

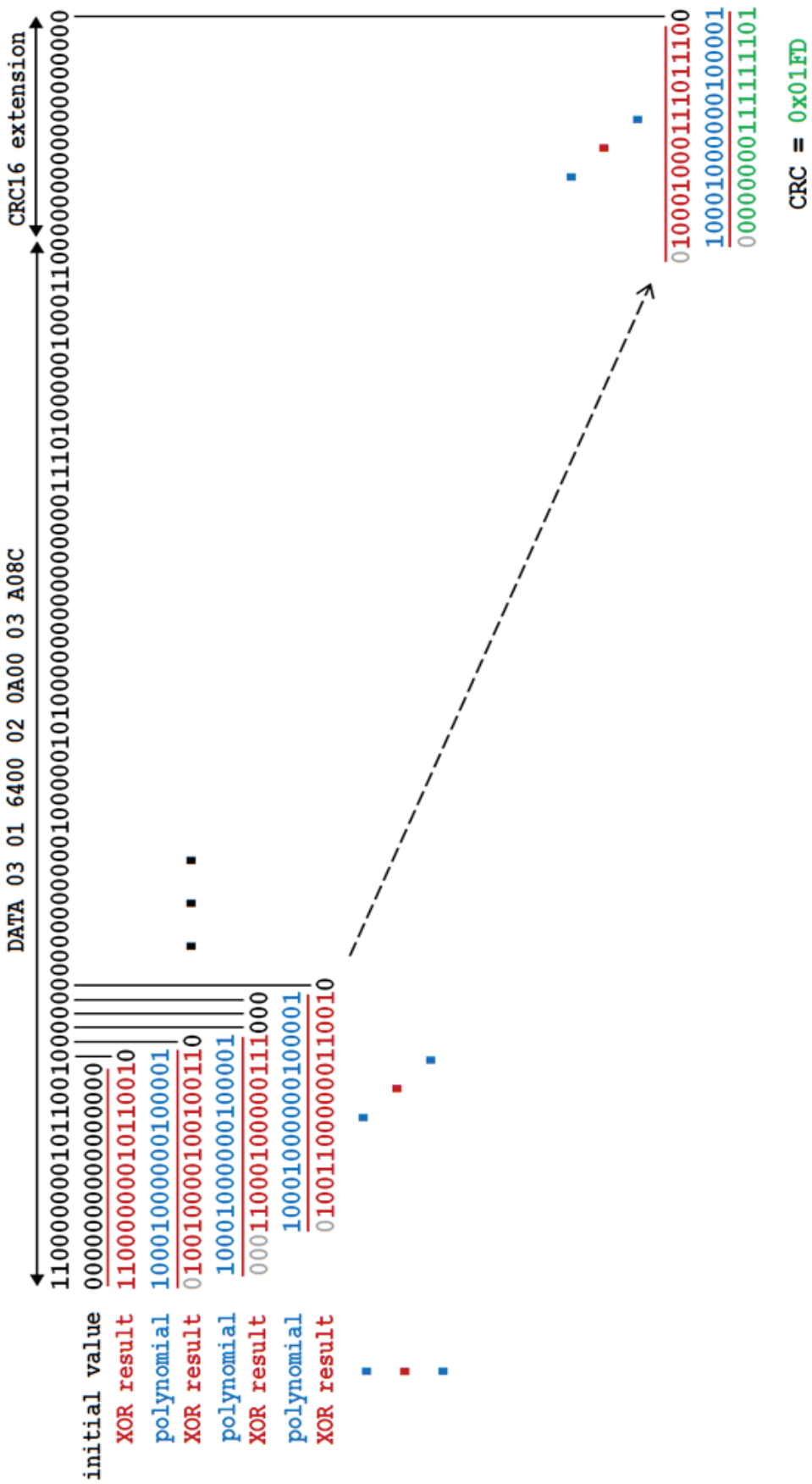
value sub 0	first sub	value	second sub	value	third sub	value
03	**01**	**6400**	**02**	**0A00**	**03**	**A08C**
Result binary↓ and hex →			MSB **03 01 6400 02 0A00 03 A08C** LSB			
0000 00**11 0000 0001 0110 0100 0000 0000 0000 0010 0000 1010 0000 0000 0000 0011 1010 0000 1000 1100**						

The result has to be taken for the CRC calculation with an appropriate program.
For a binary calculation, several XOR steps have to be done. See short description below.

With the TWK CRC calculation program the checksum can be calculated easily:



How to calculate the CRC checksum, using the binary values:



9.11.5 ♦♦ Object 3100 position_overflow_free_configuration_parameters

Only valid for NOCN version with slewing ring function (Version 'S' in order code number). Output of slewing ring position. Modifications will cause 31FE = '0' (must be set 'A5' after modification). New CRC Checksum 31FF/01 must be calculated and transmitted to NOCN. Adjustment of gear parameters.

position_overflow_free_configuration_parameters

Index	Sub	Description	Length		Memory		Range/value	Action	Default
			COM	MEM	Type	Location			
3100	0	largest_supported_ sub-index	-	-	ro	ROM	3	-	-
	1	slew_gear [teeth]	Word	Word	rw	E²PROM	1 … 65,535	-	100 (0x64)
	2	measuring_gear [teeth]	Word	Word	rw	E²PROM	1 … 65,535		10 (0xA)
	3	measuring_range	Word	Word	rw	E²PROM	1 … 65,535 **and** ≤ 8192 x i (i= slew_gear / measuring_ gear)		36,000 (0x8CA0)

* At the time read only

slew_gear Number of teeth driving gear (slewing)
measuring_gear Number of teeth of encoder tooth gear
measuring_range Number of steps output for 1 turn of the slewing ring

Maximum of gear ratio i = slew_gear / measuring_gear: i <= 32 (for example 384 / 12). Settings over 32 will cause an SDO error.

Example for 3100/sub3: If I = 3 (for example 30 / 10) the maximum of the measuring range is ≤ 24,576. If I = 1 → measuring range is ≤ 8192. Depending on ratio i, 3100/sub3 has to be matched. This is reasonable for providing every single step of resolution.
Setting of the Preset value is possible via object 6100, sub 02. This parameter is verified by checksum of the profile specific parameters.
Remark: The **gear parameters** in object 3100 can be set easily after disabling of 31FE and transmission the related checksum via 31FF and then enabling of 31FE.

9.11.6 Object 3701 SRDO_control_checksum

This is a control CRC checksum related to 13FF, calculated with a different polynomial ($2^{16} + 2^{14} + 2^1 + 1$ = 0x14003). See statements at CRC object 13FF. Not valid at all devices. Only at special NOCN variants. The control CRC can be calculated with TWK CRC program 2.0.5 and the related xml file.

SRDO_control_checksum

Index	Sub	Description	Length		Memory		Range/value	Action	Default
			COM	MEM	Type	Location			
3701	0	largest_supported_ sub-index	Byte	Byte	ro	ROM	2	-	2
	1	SRDO1_control_ checksum	Word	Word	ro	E²PROM	0 … 0xFFFF	-	0x...
	2	SRDO2_control_ checksum	Word	Word	ro	E²PROM	0 … 0xFFFF	-	0x...

9.11.7 Object 3702 Safety_Encoder_control_checksum

This is a control CRC checksum related to 61FF, calculated with a different polynomial ($2^{16} + 2^{14} + 2^1 + 1$ = 0x14003). See statements at CRC object 61FF. Not valid at all devices. Only at special NOCN variants. The control CRC can be calculated with TWK CRC program 2.0.5 and the related xml file.

Safety_Encoder_control_checksum

Index	Sub	Description	Length		Memory		Range/value	Action	Default
			COM	MEM	Type	Location			
3702	0	largest_supported_ sub-index	Byte	Byte	ro	ROM	2	-	2
	1	SRDO1_control_signature	Word	Word	ro	E²PROM	0 … 0xFFFF	-	0x...
	2	SRDO2_control_signature	Word	Word	ro	E²PROM	0 … 0xFFFF	-	0x...

9.11.8 ♦ Object 3703 Cam_control_checksum

This is a control CRC checksum related to 30FF, calculated with a different polynomial ($2^{16} + 2^{14} + 2^1 + 1$ = 0x14003). See statements at CRC object 30FF. Not valid at all devices. Only at special NOCN variants. The control CRC can be calculated with TWK CRC program 2.0.5 and the related xml file.

Cam_control_checksum

Index	Sub	Description	Length		Memory		Range/value	Action	Default
			COM	MEM	Type	Location			
3703	0	largest_supported_ sub-index	Byte	Byte	ro	ROM	1	-	1
	1	cam_control_safety_configuration_signature	Word	Word	ro	E²PROM	0 … 0xFFFF	-	0x...

9.11.9 ♦♦ Object 3704 Gear_control_checksum

This is a control CRC checksum related to 31FF, calculated with a different polynomial ($2^{16} + 2^{14} + 2^1 + 1$ = 0x14003). See statements at CRC object 31FF. Not valid at all devices. Only at special NOCN variants. The control CRC can be calculated with TWK CRC program 2.0.5 and the related xml file.

Gear_control_checksum

Index	Sub	Description	Length		Memory		Range/value	Action	Default
			COM	MEM	Type	Location			
3704	0	largest_supported_ sub-index	Byte	Byte	ro	ROM	1	-	1
	1	gear_control_safety_configuration_signature	Word	Word	ro	E²PROM	0 … 0xFFFF	-	0x...

9.11.10 ♦ Object 30FD cam_self_test

The correct function of the switching contacts (cams) can be tested by using this object. Correct function means that the contacts open properly to be sure that the contacts will open in case of a safety issue (reaching cam-limits or device failure). The test procedure can take place in the operational or non operational status of the NOCN to ensure that the CANopen position signal is still available. The cams are in the test status during the test and will (not) react on limits which are presumably exceeded. Object 30FD is only realized at certain NOCN cam switches.

During this test the PLC of the application has to recognize that a test switching procedure takes place.

The active cam (contact closed) can be 'overruled' by using this object 30FD, means to set the cam inactive. An inactive cam is always open. It is not possible to close an open contact via 30FD due to safety issues. 'Contact open' is the safe state.

This feature is available from end of year 2019 on. It is not valid for every device.

Requirements:

- State: operational or preoperational
- The cams to test have to be closed (contact closed):
  - 30FE = 0xA5
  - The tested cam has to be set active by object 6300 (it usually is)
  - The cams have to be closed due to position (see cam high and cam low limits)
- No CRC neccessary

Recommended sequence:

- Transmit the required number at an appropriate point in time (no critical situation) to 30FD (e.g. 0x2 for cam 2)
- The relevant cam opens
- Transmit 0x00 to 30FD (or another number for another cam to test if required)
- The tested cam (contact) will close again (and/or another cam will open for test)

cam_self_test

Index	Sub	Description	Length		Memory		Range/value	Action	Default
			COM	MEM	Type	Location			
30FD	0	Largest sub-index	-	-	ro	ROM	1	-	1
	1	cam_self_test	Byte	Byte	rw	ROM	0...3 *	-	0x0

* 0 ... 15 at NOCN79 with four cams

cam_self_test format

Cam 1	Cam 2	Cam 3 *	Cam 4 *	N.u.	N.u.	N.u.	N.u.
1 inactive 0 active	1 inactive 0 active	1 inactive 0 active	1 inactive 0 active	0	0	0	0

* Valid for NOCN79 with four cams. N.u.: Not used.

9.12 Objects according to profile definition

9.12.1 Object 6000 operating_parameters

Operating mode byte for the sensor.
The parameter is mirrored from the safety area.

The state table:

Bit	M / O	Designation	0	1
0	M	Code direction	CW	CCW
1	O	Set firmly to low		
2	O	scaling_function_control	Inactive	Active
3-7	O	Set firmly to zero		

operating_parameters

Index	Sub	Description	Length		Memory		Range/value	Action	Default
			COM	MEM	Type	Location			
6000	0	operating_parameters	Word	Word	ro†	E²PROM	0x0, 0x01, 0x04, 0x05 *	Sen	0x0,4

* 0x0=CW and inactive, 0x01=CCW and inactive, 0x04=CW and active (standard), 0x05= CCW and active
† At encoders with CANopen standard profile: read/write rw

9.12.2 Object 6001 measuring_units_per_revolution

Number of steps per revolution.
The parameter is firmly set. No changes are possible.

measuring_units_per_revolution

Index	Sub	Description	Length		Memory		Range/value	Action	Default
			COM	MEM	Type	Location			
6001	0	measuring_units_per_ revolution	Long	Data	ro†	E²PROM	depending on model and perfomance e.g.: 4096	Sen	-

† At encoders with CANopen standard profile: read/write rw

9.12.3 Object 6002 total_measuring_range_in_measuring_units

Total measuring range in measuring units
This parameter cannot be changed.

total_measuring_range_in_measuring_units

Index	Sub	Description	Length		Memory		Range/value	Action	Default
			COM	MEM	Type	Location			
6002	0	total_measuring_ran-ge_in_measuring_units	Long	Data	ro†	ROM	depending on model and performance e.g.: 16777216	Sen	-

† At encoders with CANopen standard profile: read/write rw

9.12.4 Object 6003 preset_value

Sets the encoder to the value specified in object 6003 or 6100.
The parameter is mirrored from the safety area.

preset_value

Index	Sub	Description	Length		Memory		Range/value	Action	Default
			COM	MEM	Type	Location			
6003	0	preset_value	Long	Long	ro†	E²PROM	0…(obj. 6002)-1	Sen	-

† At encoders with CANopen standard profile: read/write rw

9.12.5 Object 6004 position_value

Position value. This value is the measured value.
The parameter is provided as a position value by the sensor. This object is updated cyclically. The parameter is taken over from the safety area and contains the secure measured value.

position_value

Index	Sub	Description	Length		Memory		Range/value	Action	Default
			COM	MEM	Type	Location			
6004	0	position_value	Long	Long	ro	RAM	0..(obj 6002) -1	-	-

Remark: Object 600C contains the position raw data with a resolution depending on device (e.g. 16 Bits at TRN-N70 or TBN-N106). 600C can be used for calculating the speed signal with max. resolution (see object 6101).

9.12.6 Object 6030 speed_value

Speed value. The dimension is digits per object 6031 sub-index 2 in ms with a resolution of 12 bits. This object is updated cyclically.
This parameter is equivalent to safety object 6124.

speed_value

Index	Sub	Description	Length		Memory		Range/value	Action	Default
			COM	MEM	Type	Location			
6030	0	largest_supported_ sub_index	Byte	Byte	ro	ROM	1	-	-
	1	speed_value_channel1	Word	Word	ro	RAM	0..(obj 6002) -1	-	-

9.12.7 Object 6031 speed_parameter

The object is mirrored from the safety area (object 6101).

speed_parameter

Index	Sub	Description	Length		Memory		Range/value	Action	Default
			COM	MEM	Type	Location			
6031	0	largest_supported_ sub_index	Byte	Byte	ro	ROM	4	-	-
	1	speed_source_selec-tor	Byte	Byte	ro†	ROM	1, 2	-	2*
	2	speed_ integration _time	Word	Word	ro†	RAM	1 to 1000 [ms]	-	100*
	3	speed_multiplier	Word	Word	ro†	RAM	1 to 65535	-	100*
	4	speed_divider	Word	Word	ro†	RAM	1 to 65535		10*

* Mirrored from object 6101.

† At encoders with CANopen standard profile: read/write rw

9.13 Safety objects according to profile definition

9.13.1 Object 6100 safety_position_configuration_parameters

The object defines the behaviour of the position registration system in the safety area.
Modifications will cause 61FE = ‘0’ (must be set ‘A5’ after modification). New CRC Checksum 61FF/01 must be calculated and transmitted to NOCN.

The parameter can only be changed in PRE-OPERATIONAL state.

Attention:

The object can only be written if the device's cam function has been deactivated by deleting the cam_ data_valid object 30FE flag (set ‘0’).

safety_position_configuration_parameters

Index	Sub	Description	Length		Memory		Range/value	Action	Default
			COM	MEM	Type	Location			
6100	0	largest_supported_ sub_index	Byte	Byte	ro	ROM	3	-	-
	1	safety_code_sequence *	Word	Word	rw	E²PROM	0 to 1 (0=cw, 1=ccw)	-	0
	2	safety_preset_value	Long	Long	rw	E²PROM	0..(obj 6002)-1		0
	3	Safety_HR_preset_ value			ro		0x7FFFFFFFFF FFFFFF		

* In case of code ‘S’ (slewing ring functionality): After modification of 6100, sub01 the valid flag 31FE must be written to ‘A5’ (It shows ‘A5’ when read out but it has to be written to ‘A5’ again). Then set 61FE to ‘A5’. Otherwise the error FF FF 81 00 80 04 02 00 will be given out. This error can only be reset by setting 31FE = ‘A5’.

9.13.2 Object 6101 safety_speed_configuration_parameters

The object defines the behaviour of speed measurement in the safety area.
Modifications will cause 61FE = ‘0’ (must be set ‘A5’ after modification). New CRC Checksum 61FF/02 must be calculated and transmitted to NOCN.
The parameter can only be changed in PRE-OPERATIONAL state.

If **6100,** sub01, sub02 and sub03 is modified, 6101, sub01, sub02 and sub03 is modified automatically and vice versa.

Attention: The object can only be written if the device's cam function has been deactivated by deleting the cam_data_valid object 30FE flag (set ‘0’).

safety_speed_configuration_parameters

Index	Sub	Description	Length		Memory		Range/value	Action	Default
			COM	MEM	Type	Location			
6101	0	largest_supported_ sub_index	Byte	Byte	ro	ROM	7	-	-
	1	safety_code_sequence	Word	Word	rw	E²PROM	0, 1 (0=cw, 1=ccw)	-	0
	2	safety_preset_value	Long	Long	rw	E²PROM	0..(obj 6002)-1	-	0
	3	safety_preset_HR_value	Long64	Long64	ro	ROM	0x7FFFFFFFFF FFFFFF	-	-
	4	safety_speed_source *	Byte	Byte	rw/ro	ROM	1, 2	-	2
	5	safety_speed_ integration _time	Word	Word	rw	E²PROM	1…1000 [ms]	-	100
	6	safety_speed_multiplier	Word	Word	rw	E²PROM	1 … 65535	-	100**
	7	safety_speed_divider	Word	Word	rw	E²PROM	1 … 65535	-	10**

* 1 = scaled position from object 6004. 2 = Raw data of position from object 600C, no scaling factor enabled. The related resolution of setting 1 or 2 for the speed signal depends on the device (e.g. TRN-N70: 2=16 Bits)

** These subs are not realized at all devices. See TY sheet at delivered device or variant specification (e.g.N107). They can be used for calibration purposes (speed resolution and possible max. speed). The default values depend on the default values of slew_gear and measuring_gear (object 3100 position_overflow_free_configuration_parameters) if the encoder has slewing ring functionality. If not, the default values will be 1 and 1. At NOCN79 with v-source B = 16 Bits sub 6 and 7 are chosen to achieve resolution displayed in article number (e.g. 1 and 8 to achieve 13 Bits at NOCN79-…-8192R…). Remark for 6101/sub6 and sub7: To get a safety speed signal which is matched to customer requirements (calibration to an application. For example to a gear ratio i = slew_gear / measuring_gear), the desired values have to be entered in 6101/sub6 and sub7 (not realized in every device. Then sub 06 and 07 do not exist). As well in case of slewing ring functionality every modification of slew_gear and measuring_gear has to be entered here as well if a matched speed signal is required.

9.13.3 Object 6120 safety_standard_position_value

The object contains the current position. This object is used in the mapping structure for data output within the SRDO. In the event of access to individual objects, it must be noted that the consistency of the measured value is not ensured.

safety_standard_position_value

Index	Sub	Description	Length		Memory		Range/value	Action	Default
			COM	MEM	Type	Location			
6120	0	largest_supported_ sub_index	Byte	Byte	ro	ROM	4	-	-
	1	position_value_1	Byte	Byte	ro	RAM	0..(obj 6002)-1	-	-
	2	position_value_2	Byte	Byte	ro	RAM	0..(obj 6002)-1	-	-
	3	position_value_3	Byte	Byte	ro	RAM	0..(obj 6002)-1	-	-
	4	position_value_4	Byte	Byte	ro	RAM	0..(obj 6002)-1	-	-

9.13.4 Object 6121 safety_inverted_position_value

The object contains the current bit-inverted position. This object is used in the mapping structure for data output within the SRDO. In the event of access to individual objects, it must be noted that the consistency of the measured value is not ensured.

safety_inverted_standard_position_value

Index	Sub	Description	Length		Memory		Range/value	Action	Default
			COM	MEM	Type	Location			
6121	0	largest_supported_ sub_index	Byte	Byte	ro	ROM	4	-	-
	1	inverted_position_ value1	Byte	Byte	ro	RAM	0..(obj 6002)-1	-	-
	2	inverted_position_ value2	Byte	Byte	ro	RAM	0..(obj 6002)-1	-	-
	3	inverted_position_ value3	Byte	Byte	ro	RAM	0..(obj 6002)-1	-	-
	4	inverted_position_ value4	Byte	Byte	ro	RAM	0..(obj 6002)-1	-	-

9.13.5 Object 6124 safety_speed_value

The object contains the current measured speed value. This object is used in the mapping structure for data output within the SRDO. In the event of access to individual objects, it must be noted that the consistency of the measured value is not ensured.

safety_speed_value

Index	Sub	Description	Length		Memory		Range/value	Action	Default
			COM	MEM	Type	Location			
6124	0	largest_supported_ sub_index	Byte	Byte	ro	ROM	2	-	-
	1	speed_value1	Byte	Byte	ro	RAM	0 ..0xFFFF	-	-
	2	speed_value2	Byte	Byte	ro	RAM	0 ..0xFFFF	-	-

9.13.6 Object 6125 safety_inverted_speed_value

The object contains the current bit-inverted measured speed value. This object is used in the mapping structure for data output within the SRDO. In the event of access to individual objects, it must be noted that the consistency of the measured value is not ensured.

safety_inverted_speed_value

Index	Sub	Description	Length		Memory		Range/value	Action	Default
			COM	MEM	Type	Location			
6125	0	largest_supported_ sub_index	Byte	Byte	ro	ROM	2	-	-
	1	inverted_speed_value1	Byte	Byte	ro	RAM	0 ..0xFFFF	-	-
	2	inverted_speed_ value2	Byte	Byte	ro	RAM	0 ..0xFFFF	-	-

9.13.7 Object 61FE safety_configuration_valid

The object is the confirmation flag stating that the configuration is valid. The other processes such as switching cam encoder and CANopen Safety can only be activated after setting this flag. On setting the flag, the consistency of the CRC checksum and the set parameters is checked.

This parameter can only be changed in PRE-OPERATIONAL state.

When the CRC checksum is not correct (i.e. in case of changes of parameters without changing the CRC checksum or a wrong checksum is transmitted to the NOCN) 61FE can't be activated. An error message comes:

(80 fe 61 00 22 00 00 08).

Attention:

The object can only be written if the device's cam function has been deactivated by deleting the cam_ data_valid object 30FE flag (set '0').

safety_configuration_valid

Index	Sub	Description	Length		Memory		Range/value	Action	Default
			COM	MEM	Type	Location			
61FE	0	safety_configurati- on_valid	Byte	Byte	rw	E²PROM	0 / 0xA5	-	0xA5

9.13.8 Object 61FF safety_configuration_signature

The object contains the checksum across the safety encoder parameters in objects 6100 and 6101. The checksum must be rewritten in the event of changes by the user. The checksum is checked on setting the configuration_ valid flag. If no correspondence with the checksum stored in this object is ascertained, setting the flag is blocked (safety_configuration_signature remains ZERO). To calculate the CRC checksum you can use the special TWK program.

To every CRC checksum object xxFF exists a second CRC control checksum object 370x, where the CRC is calculated with a different CRC16 polynomial. This second CRC therefore differs in comparison to the first CRC and is read only. It is called ..._**control**_... . The second CRC can be read by the PLC to make sure on a second path/check that the encoder accepted the new parameters. Only the main CRC xxFF has to be written when parameters were modified. The control CRC can be calculated with TWK CRC program 2.0.5 and the related xml file. This feature is available from end of year 2019 on.

Link to the program on www.twk.de: Look at device NOCN/HBN etc. → download "Software Safety CRC".
Or use the following link: www.twk.de/files/CRC-Calculator20.zip. Description file: CRC14076
Please ask our technicians for xml files for special versions of encoders or look at www.twk.de.

The parameter can only be changed in PRE-OPERATIONAL state.

Attention:

The object can only be written if the device's cam function has been deactivated by deleting the cam_ data_valid object 30FE flag (set '0').

safety_configuration_signature

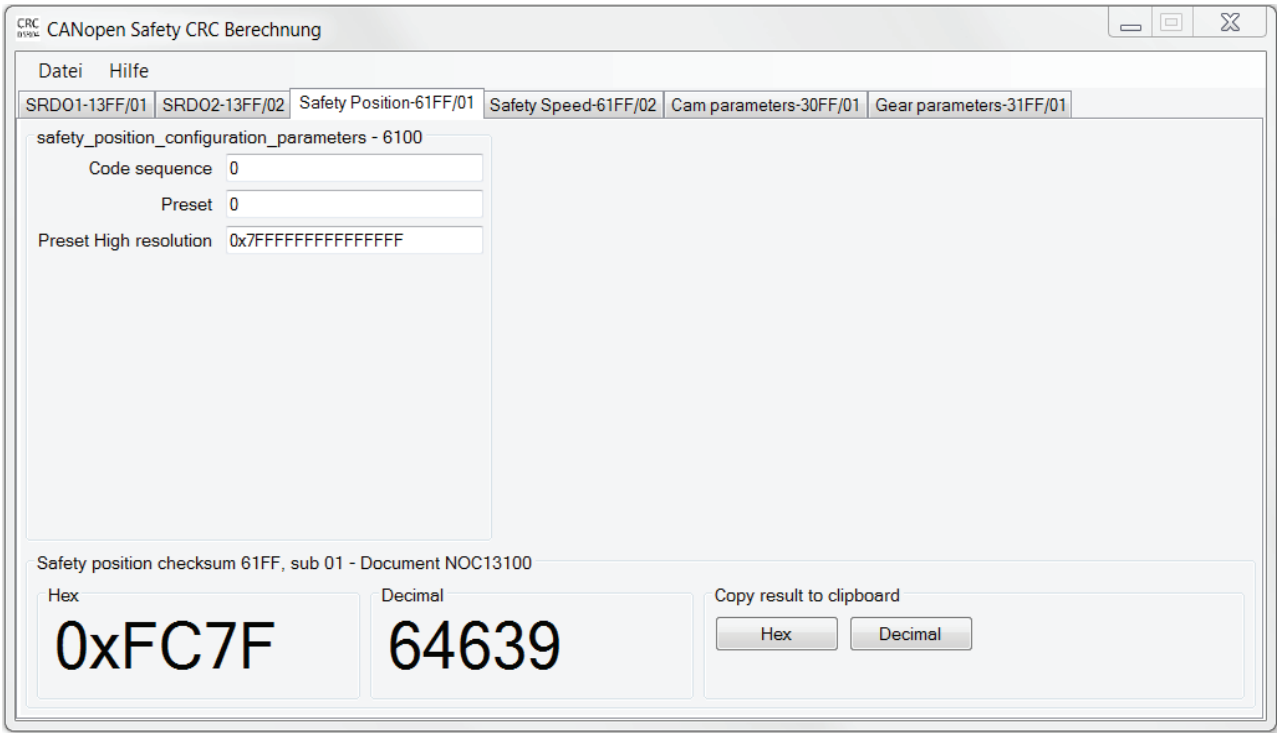
Index	Sub	Description	Length		Memory		Range/value	Action	Default
			COM	MEM	Type	Location			
61FF	0	largest_supported_ sub_index	Byte	Byte	ro	ROM	4	-	4
	1	SRDO1_signature (for changes in object 6100)	Word	Word	rw	RAM	0 ..0xFFFF	-	0x....*
	2	SRDO2_signature (for changes in object 6101)	Word	Word	rw	RAM	0 ..0xFFFF	-	0x....*

* Depends on default values in objects 6100 and 6101. Control CRC checksum: object 3702.

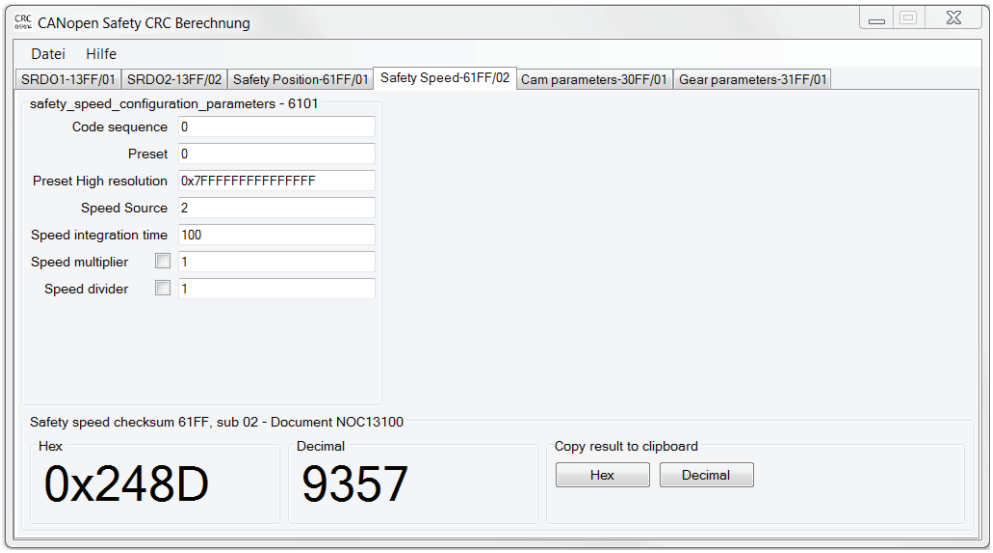
Pay attention to the different values which have to be tagged in the TWK program for the CRC checksums for sub 1 and sub 2:

Possible versions:
- NOCN with code 'S': 'Speed multiplier' and 'Speed divider' always exist.
- Standard NOCN with code 'R': 'Speed multiplier' and 'Speed divider' do not exist
- Special versions of NOCN with code 'R' (For example NOCN-N17): 'Speed multiplier' and 'Speed divider' do exist

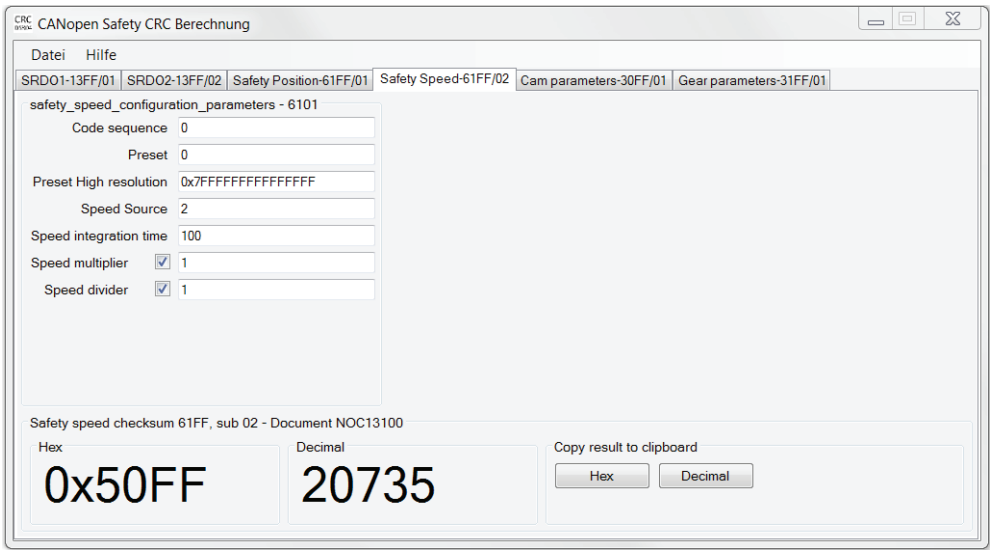
At each kind of encoder NOCN you have to tag for 61FF, **sub 01:**



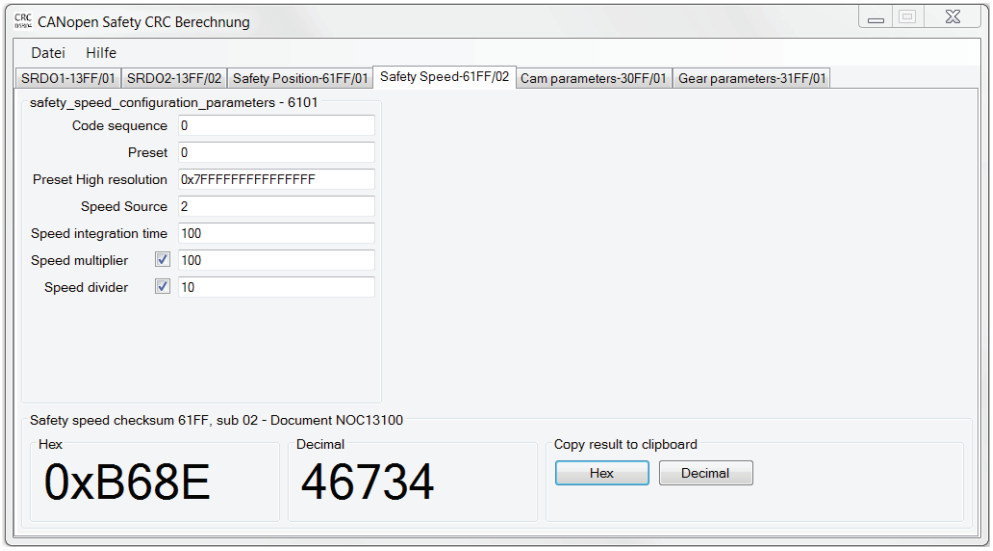
At standard version of NOCN you don't have to tag 'Speed multiplier' and 'Speed divider' for 61FF, **sub 02:**



At special versions of NOCN (for example NOCN-N17) you have to tag 'Speed multiplier' and 'Speed divider'. They are set to '1':



In case of code ‘S’ (Slewing ring functionality) you always have to tag ‘Speed multiplier’ and ‘Speed divider’. The standard values for ‘Speed multiplier’ and ‘Speed divider’ are 100 and 10. In this case the related window for 61FF, **sub 02** looks like:



To recognize which version is yours (i.e if 6101, sub 06 and sub 07 exist) please see EDS file and/or connection diagram TYxxxxx which is delivered with each device. Or you can try to read out 6101, sub 06 and sub 07 if these documents are not handy.

See as well examples for parameterization at the end of this document.

Calculation of the checksum:

Generator polynomial:	$2^{16} + 2^{12} + 2^5 + 1 = 0x11021$
2nd generator polynomial for control checksum:	$2^{16} + 2^{14} + 2^1 + 1 = 0x14003$
Initial value:	0x00 00
Final XOR:	no
Details:	See example for object 31FF

9.14 Objects for achieving compatibility

9.14.1 Object 6200 Cyclic Timer

In the case of values > 0, the object position value 6004 is transmitted cyclically with the value of the cyclic timer in ms on PDO 1. This object is logically set equal to event_timer of PDO1.

Cyclic Timer

Index	Sub	Description	Length		Memory		Range/value	Action	Default
			COM	MEM	Type	Location			
6200	0	cyclic_timer	Word	Word	rw	XRAM	0...0xFFFF	-	0

9.15 ♦ Cam setting

9.15.1 ♦ General information

All cam functionality parameters are only writeable if:

1. The CANopen communication is in PRE-OPERATIONAL state.
2) The device's cam function has been deactivated by deleting the cam_data_valid object 30FEh flag.
3) Modifications at objects 63xx will cause 30FE = '0' (must be set 'A5' after modification). New CRC Checksum 30FF/01 must be calculated and transmitted to NOCN. NOCN79 with four cams has a separate CRC calculation file for the cam CRC.

9.15.2 ♦ Object 6300 cam_state_register

Cam state. Indicates whether the cams are in the active range.

cam_state_register

Index	Sub	Description	Length		Memory		Range/value	Action	Default
			COM	MEM	Type	Location			
6300	0	Biggest sub-index	-	-	ro	ROM	1	-	-
	1	cam_state_register	Byte	Byte	ro	RAM	0 … 3 *	-	-

* 0 ... 15 at NOCN79 with four cams

cam_state_register format

Cam 1	Cam 2	Cam 3 *	Cam 4 *	N.u.	N.u.	N.u.	N.u.
1 active 0 inactive	1 active 0 inactive	1 active 0 inactive	1 active 0 inactive	0	0	0	0

N.u.: Not used.
* Valid for NOCN79 with four cams

9.15.3 ♦ Object 6301 cam_enable_register

The cams can be switched on and off. In inactive state (cam disabled), the cam is no longer updated during position evaluation.

cam_enable_register

Index	Sub	Description	Length		Memory		Range/value	Action	Default
			COM	MEM	Type	Location			
6301	0	Biggest sub-index	-	-	ro	ROM	1	-	-
	1	cam_enable_register	Byte	Byte	rw	RAM	0 … 3 *	-	0x03

* 0 ... 15 at NOCN79 with four cams

cam_enable_register format

Cam 1	Cam 2	Cam 3 *	Cam 4 *	N.u.	N.u.	N.u.	N.u.
1 enabled 0 disabled	1 enabled 0 disabled	1 enabled 0 disabled	1 enabled 0 disabled	0	0	0	0

N.u.: Not used.
* Valid for NOCN79 with four cams

9.15.4 ♦ Object 6302 cam_polarity_register

The polarity of the cams can be changed. If the polarity flag belonging to the cam is active (equal to 1), the relevant relay is dropped off when the cam state is active (see object 6300). If the flag is equal to zero, the relay is picked up when the cam state is active.

cam_polarity_register

Index	Sub	Description	Length		Memory		Range/value	Action	Default
			COM	MEM	Type	Location			
6302	0	Biggest sub-index	-	-	ro	ROM	1	-	-
	1	cam_polarity_register	Byte	Byte	rw	RAM	0 … 3 *	-	0

* 0 ... 15 at NOCN79 with four cams

cam_polarity_register format

Cam 1	Cam 2	Cam 3 *	Cam 4 *	N.u.	N.u.	N.u.	N.u.
1 enabled 0 disabled	1 enabled 0 disabled	1 enabled 0 disabled	1 enabled 0 disabled	0	0	0	0

N.u.: Not used.
* Valid for NOCN79 with four cams

9.15.5 ♦ Object 6310 cam_1_low_limit

If this position is reached, the cam state "active" is assumed.

cam_1_low_limit

Index	Sub	Description	Length		Memory		Range/value	Action	Default
			COM	MEM	Type	Location			
6310	0	Biggest sub-index	-	-	ro	ROM	1	-	-
	1	cam_1_low_limit	Long	Long	rw	RAM	0… object 6002	-	0

9.15.6 ♦ Object 6311 cam_2_low_limit

If this position is reached, the cam state "active" is assumed.

cam_2_low_limit

Index	Sub	Description	Length		Memory		Range/value	Action	Default
			COM	MEM	Type	Location			
6311	0	Biggest sub-index	-	-	ro	ROM	1	-	-
	1	cam_2_low_limit	Long	Long	rw	RAM	0..object 6002	-	2048

Object 6312 is valid for **cam 3 Low Limit** with same boundary conditions
Object 6313 is valid for **cam 4 Low Limit** with same boundary conditions

9.15.7 ♦ Object 6320 cam_1_high_limit

If this position is exceeded, the cam state "inactive" is assumed.

cam_1_high_limit

Index	Sub	Description	Length		Memory		Range/value	Action	Default
			COM	MEM	Type	Location			
6320	0	Biggest sub-index	-	-	ro	ROM	1	-	-
	1	cam_1_high_limit	Long	Long	rw	RAM	0… object 6002	-	4096

9.15.8 ♦ Object 6321 cam_2_high_limit

If this position is exceeded, the cam state "inactive" is assumed.

cam_2_high_limit

Index	Sub	Description	Length		Memory		Range/value	Action	Default
			COM	MEM	Type	Location			
6321	0	Biggest sub-index	-	-	ro	ROM	1	-	-
	1	cam_2_high_limit	Long	Long	rw	RAM	0… object 6002	-	6144

Object 6322 is valid for **cam 3 High Limit** with same boundary conditions
Object 6323 is valid for **cam 4 High Limit** with same boundary conditions

9.15.9 ♦ Object 6330 cam_1_hysteresis

Cam hysteresis for high and low limit. The hysteresis is intended to prevent the cam from constantly changing its state in the event of minor position changes.

cam_1_hysteresis

Index	Sub	Description	Length		Memory		Range/value	Action	Default
			COM	MEM	Type	Location			
6330	0	Biggest sub-index	-	-	ro	ROM	1	-	-
	1	cam_1_hysteresis	Word	Word	rw	RAM	0…(object 6002) / 2 - 1	-	10

9.15.10 ♦ Object 6331 cam_2_hysteresis

Cam hysteresis for high and low limit. The hysteresis is intended to prevent the cam from constantly changing its state in the event of minor position changes.

cam_2_hysteresis

Index	Sub	Description	Length		Memory		Range/value	Action	Default
			COM	MEM	Type	Location			
6331	0	Biggest sub-index	-	-	ro	ROM	1	-	-
	1	cam_2_hysteresis	Word	Word	rw	RAM	0…(object 6002) / 2 - 1	-	10

Object 6332 is valid for **cam 3 hysteresis** with same boundary conditions
Object 6333 is valid for c**am 4 hysteresis** with same boundary conditions

9.16 Diagnosis objects

9.16.1 Object 6500 operating_status

Current sensor status. This is a representation of object 6000.
The parameter is provided by the sensor.

operating_status

Index	Sub	Description	Length	Memory	Range/value	Action	Default
6500	0	operating_status	Word	ro	Object 6000*	-	-

* 0x00=CW and inactive, 0x01=CCW and inactive, 0x04=CW and active (standard), 0x05= CCW and active

9.16.2 Object 6501 singleturn_resolution

Maximum single turn resolution

singleturn_resolution

Index	Sub	Description	Length	Memory	Range/value	Action	Default
6501	0	singleturn_resolution	Long	ro	4096	-	-

9.16.3 Object 6502 number_of_distinguishable_revolutions

Maximum number of distinguishable revolutions.
This parameter cannot be changed.

number_of_distinguishable_revolutions

Index	Sub	Description	Length	Memory	Range/value	Action	Default
6502	0	number_of_distinguis-hable_revolutions	Long	ro	4096	-	-

9.16.4 Object 6503 alarms

Internally, there is only one error byte. If an alarm occurs, an emergency message is transmitted. During SDO upload, the error byte is loaded into the object's MSB.

The following errors are evaluated:

Bit	Error type
0	Not used
1	Not used
2	Not used
3	Device error
4	Cam error
5	CRC parameter error
6	Supply out of range
7	Sensor error

Extended error coding is available for certain errors in this byte in the emergency message and in object 1003 pre_defined_error_field (see point 8.4 error codes).

CRC error: The alignment parameters and the CAN interface parameters are monitored by CRC. In the case of an error in one of the areas, this flag is set.

Sensor error: Position sensor error or the measured values of the sensors read-in in redundant form lie too far apart. Or the position controller discovers a scanning error.

Cam error: Cam controller or relay controller have detected an error.

Device error: In case of full redundant systems each systems controls the other system. If one system recognizes that the other system doesn't work anymore this error will be generated and transmitted via CAN. A hardware error (look at fatal errors) is not transmitted via CAN.

Alarms

Index	Sub	Description	Length		Memory		Range/value	Action	Default
			COM	MEM	Type	Location			
6503	0	alarms	Word	Word	ro	RAM	-	See above	-

9.16.5 Object 6504 supported_alarms

Supported alarm messages.
This is a representation of the error displays possible in the case of index 6503.

supported_alarms

Index	Sub	Description	Length		Memory		Range/value	Action	Default
			COM	MEM	Type	Location			
6504	0	supported_alarms	Word	Word	ro	ROM	0xF8	-	-

9.16.6 Object 6506 supported_warnings

Supported warning messages.
No warnings are supported. Object 6505 can therefore be omitted.

supported_warnings

Index	Sub	Description	Length		Memory		Range/value	Action	Default
			COM	MEM	Type	Location			
6506	0	supported_warnings	Word	Word	ro	ROM	0	-	-

9.16.7 Object 6507 profile_and_software_version

The profile and software version of the encoder.
The versions are each BCD-coded in bytes.
Version 2.5.12 results in 0x2512.
The current version of the encoder profile and the software version are entered.

Profile version		Software version	
Byte 0	Byte 1	Byte 2	Byte 3
Bits 7 - 0	Bits 15 - 8	Bits 7 - 0	Bits 15 - 8

profile_and_software_version

Index	Sub	Description	Length		Memory		Range/value	Action	Default
			COM	MEM	Type	Location			
6507	0	profile_and_software_ version	Long	Long	ro	ROM	0x400100102	-	-

9.16.8 Object 6508 operating_time

Not supported.

operating_time

Index	Sub	Description	Length		Memory		Range/value	Action	Default
			COM	MEM	Type	Location			
6508	0	operating_time	Long	Long	ro	ROM	0xFFFF FFFF	-	-

9.16.9 Object 6509 offset_value

Output offset. In our encoders, this is the zero point cell.
The parameter is evaluated by the sensor (only used internally).

offset_value

Index	Sub	Description	Length		Memory		Range/value	Action	Default
			COM	MEM	Type	Location			
6509	0	offset_value	Long	Long	ro	E²PROM	0..(obj 6002) -1	SEN	-

9.16.10 Object 650A module_identification

The manufacturer offset is used as the zero point parameter for synchronising the two nodes' position data. This parameter is written via the factory programming during system alignment.

module_identification

Index	Sub	Description	Length		Memory		Range/value	Action	Default
			COM	MEM	Type	Location			
650A	0	largest_supported_ sub-index	-	-	ro	ROM	1	-	-
	1	manufacturer_off- set_value	Long	Word	ro	ROM	0..(obj 6001) -1	-	*

* Written in factory programming state.

9.16.11 Object 650B serial_number

The serial number is written with the factory programming.

serial_number

Index	Sub	Description	Length		Memory		Range/value	Action	Default
			COM	MEM	Type	Location			
650B	0	serial_number	Long	Long	wp	XRAM	0....	*	-

* Written in factory programming state.

9.16.12 Object 650D absolute_accuracy

Displays the accuracy of the measuring value.

absolute_accuracy

Index	Sub	Description	Length		Memory		Range/value	Action	Default
			COM	MEM	Type	Location			
650D	0	absolute_accuracy	Byte	Byte	ro	E²PROM	.....	SEN	10 *

* Depending on device

9.16.13 Object 650E device_capability

Defines the operating capability of the encoder. The entry is carried out according to the following table:

Name	Bit	Value	Definition
ec: Encoder class	2:0	0b000 (b=binary)	Reserved
		0b001	Class 1
		0b010	Class 2
		0b011	Class 3
		0b011 to 0b111	Reserved
		rsl: Resolution	3
rsl: Resolution	3	0	Low
		1	High
r: Reserved	4	Reserved (always 0)	
saf: Safety	5	0	Safety not supported
		1	Safety supported
st: Safety type	7:6	0b00	CANopen Safety
		0b01 to 0b11	Reserved
r: Reserved	11:8	Reserved (always 0)	
msc1 to 4: Manufacturer-specific capability 1 to 4	15:12	0	Manufacturer-specific capability 1 to 4 disabled
		1	Manufacturer-specific capability 1 to 4 enabled

Bit assignment parameter definition

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
msc4	msc3	msc2	msc1	r				st		saf	r	rsl	ec		

device_capability

Index	Sub	Description	Length		Memory		Range/value	Action	Default
			COM	MEM	Type	Location			
650E	0	device_capability	Long	Long	ROM	ROM	0x23		-

Class 3 encoder with CANopen Safety, no manufacturer-specific definitions.

10 Object listing

Complete listing for NOCN with cams. For encoders without cams the objects which refer to the cams (marked blue and ♦) are not valid. The object 3100 is only valid for the encoder version without cams and with slewing ring functionality (marked blue and ♦♦). Default values may differ due to the version which is provided.

Name	Index	Sub	Len	Value dez	Value hex	Remark
device_type	1000		4	131478	0x00020196	
error_register	1001		1	0	0x00	
PredefinedErrorField	1003					
NumberofErrors		0	1	0	0xX	
StandardErrorField		1	4			
StandardErrorField_2		2	4			
StandardErrorField_3		3	4			
StandardErrorField_4		4	4			
StandardErrorField_5		5	4			
StandardErrorField_6		6	4			
StandardErrorField_7		7	4			
StandardErrorField_8		8	4			
StandardErrorField_9		9	4			
StandardErrorField_a		0A	4			
StandardErrorField_b		0B	4			
StandardErrorField_c			7			
StandardErrorField_d			7			
StandardErrorField_e			7			
StandardErrorField_f			7			
StandardErrorField_10			7			
StandardErrorField_11			7			
StandardErrorField_12			7			
StandardErrorField_13			7			
COB-ID_SYNC	1005		4	128	0x00000080	
manufacturer_device_name	1008		18	-	-	Encoder TRN Safety
manufacturer_hardware_version	1009		13	-	-	P-0698 P-0693
manufacturer_software_version	100a		15	-	-	Safety Standard
store_parameters	1010					
largest_supported_sub-index		0	1	5	0x05	
save_all_parameters		1	4	1	0x00000001	
SaveCommunicationParameters		2	4	1	0x00000001	
SaveApplicationParameters		3	4	1	0x00000001	
SaveLMTDefinedParameters		4	4	1	0x00000001	
SaveManufacturerDefinedParameters		5	4	1	0x00000001	

Name	Index	Sub	Len	Value dez	Value hex	Remark
restore_default_parameters	1011					
largest_supported_sub-index		0	1	5	0x05	
restore_all_default_parameters		1	4	1	0x00000001	
RestoreCommunicationDefaultParameters		2	4	1	0x00000001	
RestoreApplicationDefaultParameters		3	4	1	0x00000001	
RestoreLMTDefinedDefaultParameters		4	4	1	0x00000001	
RestoreManufacturerDefinedDefaultParameters		5	4	1	0x00000001	
COB-ID_EMCY	1014		4	141	0x0000008D	
EMCY_inhibit_time	1015		2	1000	0x03E8	
producer_heartbeat_time	1017		2	0	0x0	
identity_object	1018					
largest_supported_sub-index		0	1	4	0x04	
vendor-ID		1	4	269	0x0000010D	
product_code		2	4	25376	0x0000xxxx	
revision_number		3	4	65537	0x00020002	
serial_number		4	4	x	0xxxxxxxxx	
error_behaviour	1029					
NrofErrorClasses		0	1	3	0x03	
CommunicationError		1	1	0	0x00	
InternalDeviceError		2	1	0	0x00	
CAMError		3	1	1	0x01	
RedundancyError (only for full redundant encoders)		4	1	1	0x01	
SRDO1_communication_parameter	1301					
largest_supported_sub-index		0	1	6	0x06	
information_direction		1	1	1	0x01	
SRDO_compar_refresh_or_sct		2	2	25	0x0019	
validation_time		3	1	20	0x14	
transmission_type		4	1	254	0xFE	
SRDO_compar_COB_ID_1		5	4	281	0x00000119	0: active 8: inactive
SRDO_compar_COB_ID_2		6	4	282	0x0000011A	0: active 8: inactive
SRDO2_communication_parameter	1302					
largest_supported_sub-index		0	1	6	0x06	
information_direction		1	1	1	0x01	
SRDO_compar_refresh_or_sct		2	2	25	0x0019	
validation_time		3	1	20	0x14	
transmission_type		4	1	254	0xFE	
SRDO_compar_COB_ID_1		5	4	345	0x00000159	0: active 8: inactive
SRDO_compar_COB_ID_2		6	4	346	0x0000015A	0: active 8: inactive

Name	Index	Sub	Len	Value dez	Value hex	Remark
SRDO1_mapping_parameter	1381					
largest_supported_sub-index		0	1	8	0x08	
first_SRDO_mapping_object		1	4	1629487368	0x61200108	
second_SRDO_mapping_object		2	4	1629552904	0x61210108	
third_SRDO_mapping_object		3	4	1629487624	0x61200208	
fourth_SRDO_mapping_object		4	4	1629553160	0x61210208	
fifth_SRDO_mapping_object		5	4	1629487880	0x61200308	
sixth_SRDO_mapping_object		6	4	1629553416	0x61210308	
seventh_SRDO_mapping_object		7	4	1629488136	0x61200408	
eighth_SRDO_mapping_object		8	4	1629553672	0x61210408	
SRDO2_mapping_parameter	1382					
largest_supported_sub-index		0	1	4	0x04	
first_SRDO_mapping_object		1	4	1629749512	0x61240108	
second_SRDO_mapping_object		2	4	1629815048	0x61250108	
third_SRDO_mapping_object		3	4	1629749768	0x61240208	
fourth_SRDO_mapping_object		4	4	1629815304	0x61250208	
Configuration_valid	13fe		1	0	0x00	
safety_configuration_checksum	13ff					
Safety_checksum_Number_of_entries		0	1	2	0x02	
SRDO1_checksum		1	2	..... *	0x.... *	
SRDO2_checksum		2	2	..... *	0x.... *	
first_transmit_PDO_parameter	1800					
largest_supported_sub-index		0	1	5	0x05	
COB-ID_used_by_PDO		1	4	385	0x8000018D	0: active 8: inactive
transmission_type		2	1	253	0xFD	
inhibit_time		3	2	0	0x0	
reserved		-	-	-	-	
EventTimer		5	2	0	0x0	
second_transmit_PDO_parameter	1801					
largest_supported_sub-index		0	1	2	0x02	
COB-ID_used_by_PDO		1	4	641	0x8000028D	0: active 8: inactive
transmission_type		2	1	1	0x01	
first_transmit_PDO_mapping	1a00					
largest_supported_sub-index		0	1	2	0x02	
PDO_mapping_for_the_first_object		1	4	1610874912	0x60040020	
PDOMappingEntry_2		2	4	1613758736	0x60300110	
second_transmit_PDO_mapping	1a01					
largest_supported_sub-index		0	1	2	0x02	
PDO_mapping_for_the_first_object		1	4	1610874912	0x60040020	
PDOMappingEntry_2		2	4	1613758736	0x60300110	

* Depends on settings ex work.

Name	Index	Sub	Len	Value dez	Value hex	Remark
operating_parameters	6000		2	4	0x04	
measuring_units_per_revolution	6001		4	4096	0x00001000	
total_measuring_range_in_measuring_units	6002		4	16777216	0x01000000	
preset_value	6003		4	0	0x0	
position_value	6004		4	x	0xX	
raw_position_value	600c		4	x	0xX	
Speed value	6030					
NrOfObjects		0	1	1	0x1	
Speed value channel 1		1	2	x	0xX	
speed_parameter	6031					
NrOfObjects		0	1	2	0x04	
speed_source_selector		1	1	2	0x02	
speed_integration_time		2	2	100	0x0064	
speed_multiplier		3	2	100	0x0064	
speed_divider		4	2	10	0x000A	
safety_position_configuration_parameters	6100					
NrOfObjects		0	1	3	0x03	
safety_code_sequence		1	2	0	0x0	
safety_preset_value		2	4	0	0x0	
safety_preset_value_high_resolution		3	8		0x7FFFFFFFFF FFFFFF	
safety_speed_configuration_parameters	6101					
NrOfObjects		0	1	5	0x05	
safety_code_sequence		1	2	0	0x0	
safety_preset_value		2	4	0	0x0	
safety_preset_value_high_resolution		3	8	0	0x7FFFFFFFFF FFFFFF	
safety_speed_source_selector		4	1	2	0x02	
safety_speed_integration_time		5	2	100	0x0064	
safety_speed_multiplier		6	2	100	0x0064	
safety_speed_divider		7	2	10	0x00A	
safety_position_value	6120					
NrOfObjects		0	1	4	0x04	
safety_position_value_1		1	1	x	0xX	
safety_position_value_2		2	1	x	0xX	
safety_position_value_3		3	1	x	0xX	
safety_position_value_4		4	1	x	0xX	
safety_inverted_position_value	6121					
NrOfObjects		0	1	4	0x04	
safety_inverted_position_value_1		1	1	x	0xX	
safety_inverted_position_value_2		2	1	x	0xX	
safety_inverted_position_value_3		3	1	x	0xX	
safety_inverted_position_value_4		4	1	x	0xX	

Name	Index	Sub	Len	Value dez	Value hex	Remark
safety_speed_value	6124					
NrOfObjects		0	1	2	0x02	
safety_speed_value_1		1	1	x	0xX	
safety_speed_value_1		2	1	x	0xX	
safety_inverted_speed_value	6125					
NrOfObjects		0	1	2	0x02	
safety_inverted_speed_value_1		1	1	x	0xX	
safety_inverted_speed_value_2		2	1	x	0xX	
safety_application_configuration_valid	61fe		1	165	0xA5	
safety_application_configuration_signature	61ff					
NrOfObjects		0	1	2	0x02	
SRDO1_signature		1	2	..... *	0x.... *	
SRDO2_signature		2	2	..... *	0x.... *	
cyclic_timer	6200		2	0	0x0	
♦ Cam_state_register	6300					
NrOfObjects		0	1	1	0x1	
Cam_state_register		1	1	X	0xX	
♦ Cam_enable	6301					
Cam_enable_Number_of_available_channels		0	1	1	0x01	
Cam_enable_Cam_enable_channel_1		1	1	3	0x03	
♦ Cam_polarity	6302					
Cam_polarity_Number_of_available_channels		0	1	1	0x01	
Cam_polarity_Cam_polarity_channel_1		1	1	0	0x0	
♦ Cam1_low_limit	6310					
Cam1_low_limit_Number_of_available_channels		0	1	1	0x01	
Cam1_low_limit_Cam1_low_limit_channel_1		1	4	16505856	0x00FBDC00	
♦ Cam2_low_limit	6311					
Cam2_low_limit_Number_of_available_channels		0	1	1	0x01	
Cam2_low_limit_Cam2_low_limit_channel_1		1	4	8388608	0x00800000	
♦ Cam3_low_limit	6312					
Cam3_low_limit_Number_of_available_channels		0	1	1	0x01	
Cam3_low_limit_Cam3_low_limit_channel_1		1	4	tbd	tbd	
♦ Cam4_low_limit	6313					
Cam4_low_limit_Number_of_available_channels		0	1	1	0x01	
Cam4_low_limit_Cam4_low_limit_channel_1		1	4	tbd	tbd	
♦ Cam1_high_limit	6320					
Cam1_high_limit_Number_of_available_channels		0	1	1	0x01	
Cam1_high_limit_Cam1_high_limit_channel_1		1	4	8388608	0x00800000	
♦ Cam2_high_limit	6321					
Cam2_high_limit_Number_of_available_channels		0	1	1	0x01	
Cam2_high_limit_Cam2_high_limit_channel_1		1	4	271360	0x00042400	

* Depends on settings ex work. ♦ and ♦♦: See head of table

Name	Index	Sub	Len	Value dez	Value hex	Remark
♦ **Cam3_high_limit**	6322					
Cam3_high_limit_Number_of_available_channels		0	1	1	0x01	
Cam3_high_limit_Cam3_high_limit_channel_1		1	4	tbd	tbd	
♦ **Cam4_high_limit**	6323					
Cam4_high_limit_Number_of_available_channels		0	1	1	0x01	
Cam4_high_limit_Cam4_high_limit_channel_1		1	4	tbd	tbd	
♦ **Cam1_hysteresis**	6330					
Cam1_hysteresis_Number_of_available_channels		0	1	1	0x01	
Cam1_hysteresis_Cam1_hysteresis_channel_1		1	2	10	0x000A	
♦ **Cam2_hysteresis**	6331					
Cam2_hysteresis_Number_of_available_channels		0	1	1	0x01	
Cam2_hysteresis_Cam2_hysteresis_channel_1		1	2	10	0x000A	
♦ **Cam3_hysteresis**	6332					
Cam3_hysteresis_Number_of_available_channels		0	1	1	0x01	
Cam3_hysteresis_Cam3_hysteresis_channel_1		1	2	10	0x000A	
♦ **Cam4_hysteresis**	6333					
Cam4_hysteresis_Number_of_available_channels		0	1	1	0x01	
Cam4_hysteresis_Cam4_hysteresis_channel_1		1	2	10	0x000A	
operating_status	6500		2			
singleturn_resolution	6501		4	4096	0x1000	
number_of_distinguishable_revolutions	6502		2	4096	0x1000	
alarms	6503		2	0	0x0	
supported_alarms	6504		2	63488	0xF800	
supported_warnings	6506		2	0	0x0	
profile_and_software_version	6507		4	xxxxxxx	0xXXXXXXX	
operating_time	6508		4	-1	0xFFFFFFFF	
offset_value	6509		4	0	0x0	
module_identification	650a					
largest_supported_sub-index		0	1	1	0x01	
manufacturer_offset_value		1	4	0	0x0	
serial_number	650b		4	x	0xX	
absolute_accuracy	650d		1	10	0xA	
Device_capability	650e		4	35	0x23	
node-ID	2000		1	13	0x0D	
bit_rate	2001		1	2	0x02	
♦ **cam_safety_configuration_valid**	30fe		1	0	0x0	
♦ **cam_safety_configuration_signature**	30ff					
NrOfObjects		0	1	1	0x01	
cam_safety_configuration_signature_sub1		1	2	..... *	0x.... *	
♦♦ **gear_safety_configuration_valid**	31fe		1	165	0xA5	

* Depends on settings ex work. ♦ and ♦♦: See head of table

Name	Index	Sub	Len	Value dez	Value hex	Remark
♦♦ gear_safety_configuration_signature	31ff					
NrOfObjects		0	1	1	0x01	
gear_safety_configuration_signature_sub1		1	2	..... *	0x.... *	
♦♦ position_overflow_free_configuration_parameters	3100					
largest_supported_sub-index		0	1	3	0x03	
slew_gear		1	2	100	0x64	
measuring_gear		2	2	10	0xA	
measuring_range		3	4	36000	0x8CA0	
SRDO_control_checksum	3701					
largest_supported_sub-index		0	1	2	0x02	
SRDO1_control_checksum		1	2	..... *	0x.... *	
SRDO2_control_checksum		2	2	..... *	0x.... *	
Safety_Encoder_control_checksum	3702					
largest_supported_sub-index		0	1	2	0x02	
SRDO1_control_signature		1	2	..... *	0x.... *	
SRDO2_control_signature		2	2	..... *	0x.... *	
♦ Cam_control_checksum	3703					
largest_supported_sub-index		0	1	1	0x01	
cam_control_safety_configuration_signature		1	2	..... *	0x.... *	
♦♦ Gear_control_checksum	3704					
largest_supported_sub-index		0	1	1	0x01	
gear_control_safety_configuration_signature		1	2	..... *	0x.... *	
♦ Cam_self_test	30fd					
largest_supported_sub-index		0	1	1	0x01	
cam_self_test		1	1	0	0x00	

* Depends on settings ex work. ♦ and ♦♦: See head of table

11 Recommendation for parameterization

Recommended steps for parameterization of safety encoders by example NOCN

(See trace sequences for further details, chapter 12)

NOCN with and without cams **Set position parameter: Code sense (CW-CCW)** object 6100, sub 01	NOCN with and without cams **Set position parameter: Preset value** object 6100, sub 02	NOCN with and without cams **Set position parameter: Speed integration time** object 6101, sub 05	Only NOCN with cams **Set position parameter: cams low and high limits** object 631x and 632x sub 01 each
Set NOCN preoperational			
Deactivate data valid flag "cam" 30FE (set 0) (Only in case of NOCN with cams)	**Deactivate data valid flag "cam" 30FE (set 0)** (Only in case of NOCN with cams)	**Deactivate data valid flag "cam" 30FE (set 0)** (Only in case of NOCN with cams)	**Deactivate data valid flag "cam" 30FE (set 0)**
Set object 6100, sub 01 to the desired value (CW=0 or CCW=1) Remark: Data valid flag "position" 61FE is deactivated automatically (set 0) by writing to this object	**Set object 6100, sub 02 to the desired value** Remark: Data valid flag "position" 61FE is deactivated automatically (set 0) by writing to this object	**Set object 6101, sub 05 to the desired value** Remark: Data valid flag "position" 61FE is deactivated automatically (set 0) by writing to this object	**Set object 631x, 632x to the desired value**
Calculate the two related CRC checksums * www.twk.de/files/CRC-Calculator20.zip	**Calculate the two related CRC checksums *** www.twk.de/files/CRC-Calculator20.zip	**Calculate the two related CRC checksums *** www.twk.de/files/CRC-Calculator20.zip	**Calculate the two related CRC checksums *** www.twk.de/files/CRC-Calculator20.zip
Write these two checksums to the related object: 61FF, sub 01 and sub 02	**Write these two checksums to the related object: 61FF, sub 01 and sub 02**	**Write this checksum to the related object: 61FF, sub 02**	**Write this checksum to the related object: 30FF, sub 01**
Activate data valid flags: 31FE (set A5, at NOCN with code 'S') 61FE (set A5) Remark: When setting this flag active the preset value will be activated automatically → Position value	**Active data valid flag "position": 61FE (set A5)** Remark: When setting this flag active the preset value will be activated automatically → Position value	**Active data valid flag "position": 61FE (set A5)** Remark: When setting this flag active the preset value will be activated automatically → Position value	
Active data valid flag "cam": 30FE (set A5) (in case of NOCN with cams)	**Active data valid flag "cam": 30FE (set A5)** (in case of NOCN with cams)	**Active data valid flag "cam": 30FE (set A5)** (in case of NOCN with cams)	**Active data valid flag "cam": 30FE (set A5)**
Active data valid flag "SRDO": 13FE (set A5)	**Active data valid flag "SRDO": 13FE (set A5)**	**Active data valid flag "SRDO": 13FE (set A5)**	
Save all parameters via object 1010, sub 01 ****	**Save all parameters via object 1010, sub 01 ******	**Save all parameters via object 1010, sub 01 ******	**Save all parameters via object 1010, sub 01 ******
Set NOCN operational			
Same procedure like 6101, sub 05 for changing the parameters "Speed multiplier" and "Speed divider" in 6101, sub 06 and sub 07. See page 44. ***: You have to pay attention to the calculation of the checksum (→ Tagging of lines in the TWK program). Please see chapter 9.13.8 on page 46 for a detailed description. See datasheet 14076 for using TWK CRC programme.** ****: See object 1010 for detailed information concerning sub-indices ! Save = 0x65766173**			

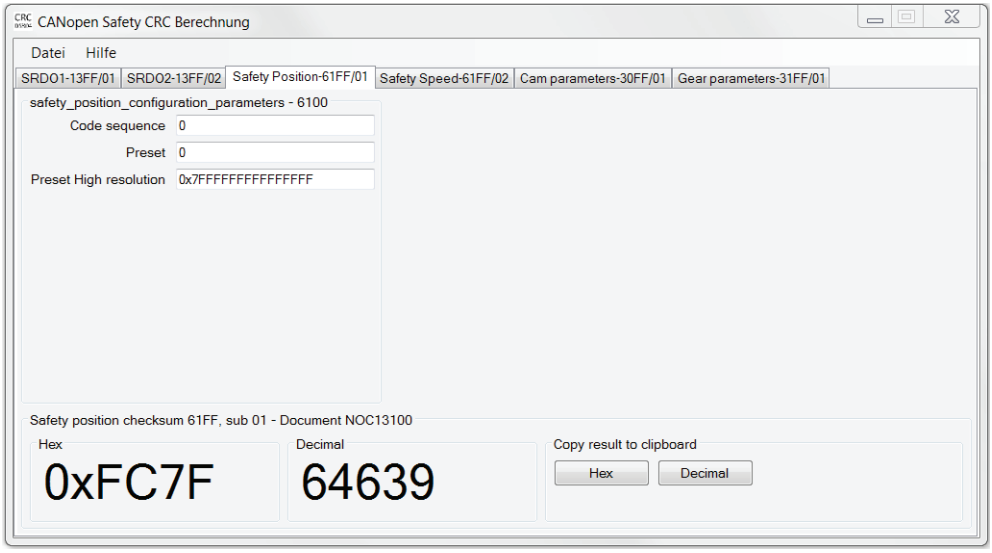
Only NOCN with cams **Set cam parameter: cam - hysteresis** object 633x, sub 01	Only NOCN with cams **Set cam parameter: cam enable / cam polarity** object 6301, sub 01 / 6302, sub 01	NOCN with and without cams **Set SRDO parameter: COB ID 1 / COB ID 2** object 1301 / 1302, sub 05 / 06 Remark: For setting Node ID use object 2000. See chap. 9.10.1 and 12.6 for details.	NOCN with and without cams **Set SRDO parameter: Refresh time** object 1301 / 1302, sub 02
Set NOCN preoperational			
Deactivate data valid flag "cam" 30FE (set 0)	**Deactivate data valid flag "cam" 30FE (set 0)**	**Deactivate data valid flag "SRDO" 13FE (set 0)**	**Deactivate data valid flag "SRDO" 13FE (set 0)**
Set objects 633x to the desired values	**Set objects 6301 and 6302 to the desired values**	**Set objects 1301 / 1302, sub 05 and 06 to the desired values:** 1301/05: COB_ID_1=FF+2*Node ID 1301/06: COB_ID_2=100+2*Node ID 1302/05: COB_ID_1=13F+2*Node ID 1302/06: COB_ID_2=140+2*Node ID	**Set object 1301, sub 2 / 1302, sub 02 to the desired values**
Calculate the related CRC checksum (TWK porgram) www.twk.de/files/CRC-Calculator20.zip	**Calculate the related CRC checksum (TWK porgram)** www.twk.de/files/CRC-Calculator20.zip	**Calculate the two related CRC checksums (TWK porgram)** www.twk.de/files/CRC-Calculator20.zip **Remark:** 13FF, sub 01: use COB_ID_1 and COB_ID_2 of object 1301, sub 05 and sub 06 and mapping parameter values of object 1381. 13FF, sub 02: use COB_ID_1 and COB_ID_2 of object 1302, sub 05 and sub 06 and mapping parameter values of object 1382.	**Calculate the two related CRC checksums (TWK program)** www.twk.de/files/CRC-Calculator20.zip **Remark:** 13FF, sub 01: use COB_ID_1 and COB_ID_2 of object 1301, sub 05 and sub 06 and mapping parameter values of object 1381. 13FF, sub 02: use COB_ID_1 and COB_ID_2 of object 1302, sub 05 and sub 06 and mapping parameter values of object 1382.
Write this checksum to the related object: 30FF, sub 01	**Write this checksum to the related object: 30FF, sub 01**	**Write these checksums to the related object: 13FF, sub 01 and sub 02**	**Write these checksums to the related object: 13FF, sub 01 and sub 02**
Activate data valid flag "cam": 30FE (set A5)	**Activate data valid flag "cam": 30FE (set A5)**	**Activate data valid flag "SRDO": 13FE (set A5)**	**Activate data valid flag "SRDO": 13FE (set A5)**
Save all parameters via object 1010, sub 01 See object 1010 for detailed information concerning sub-indices! Save=0x65766173	**Save all parameters via object 1010, sub 01** See object 1010 for detailed information concerning sub-indices! Save=0x65766173	**Save all parameters via object 1010, sub 01** See object 1010 for detailed information concerning sub-indices! Save=0x65766173	**Save all parameters via object 1010, sub 01** See object 1010 for detailed information concerning sub-indices! Save=0x65766173
Set NOCN operational			

Only NOCN with slewing ring functionality **Set gear parameter:** **Number of teeth - slewing ring** and/or **Number of teeth – NOCN pinion** and/or **Measuring range** object 3100, sub 01 to sub 03
Set NOCN preoperational
Set object 3100, sub 01 to sub 03 to the desired values * Remark: Data valid flag “gear” 31FE is deactivated automatically (set 0) by writing to this object
Calculate the related CRC checksum (TWK program: ‘gear parameter’) www.twk.de/files/CRC-Calculator20.zip
Write this checksum to the related object: 31FF, sub 01
Activate data valid flag "gear": **31FE (set A5)**
Activate data valid flag "position": **61FE (set A5)**
Activate data valid flag "SRDO": **13FE (set A5)**
Save all parameters via object **1010, sub 01** See object 1010 for detailed information concerning sub-indices ! Save = 0x65766173
Set NOCN operational

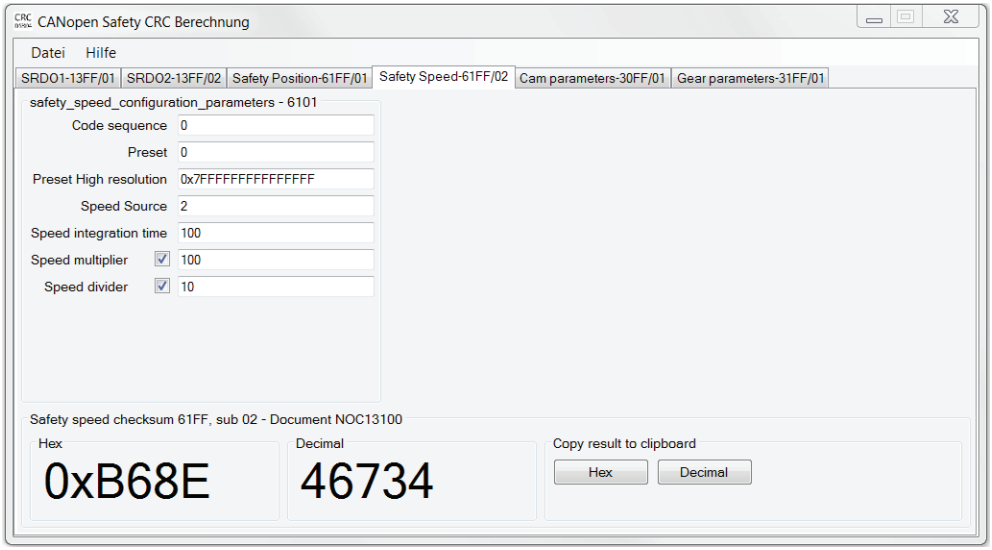
* Depending on the ratio i (slew_gear / measuring_gear) it may be necessary to match the measuring range due to the condition: measuring range ≤ 8192 x i. When speed value has to be matched to ratio i, in object 6101 / sub 6 and sub 7 this ratio has to be entered as well.

11.1 Screenshots: CRC checksum calculation program

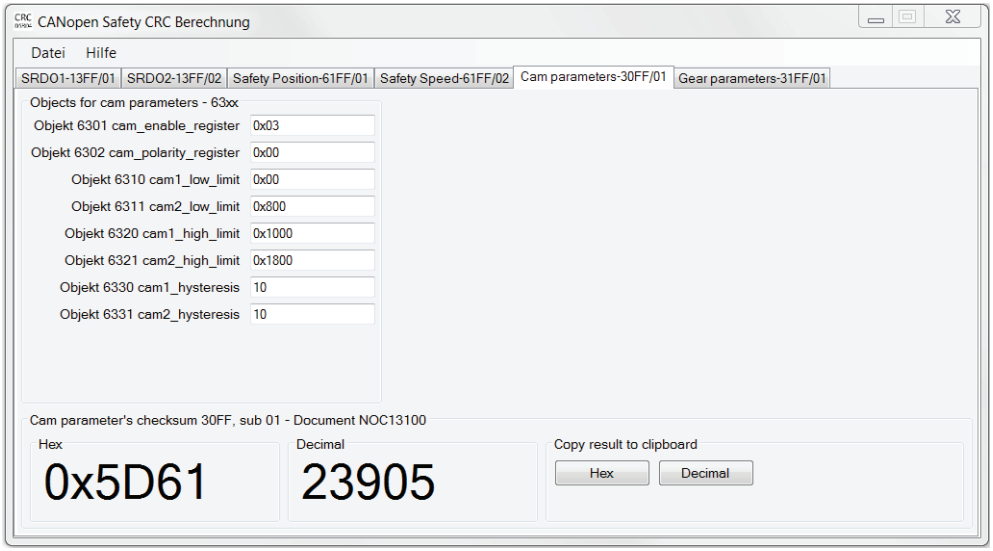
Screenshot for 6100 / 6101 default, Checksum for 61FF, sub 01



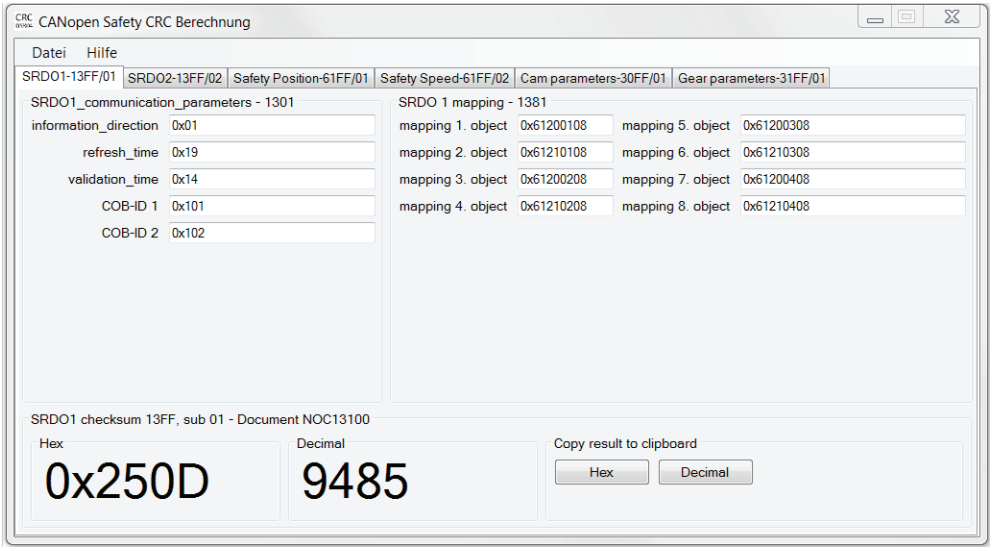
Screenshot for 6100 / 6101 default, Checksum for 61FF, sub 02
(In this case: encoder with slewing ring functionality)



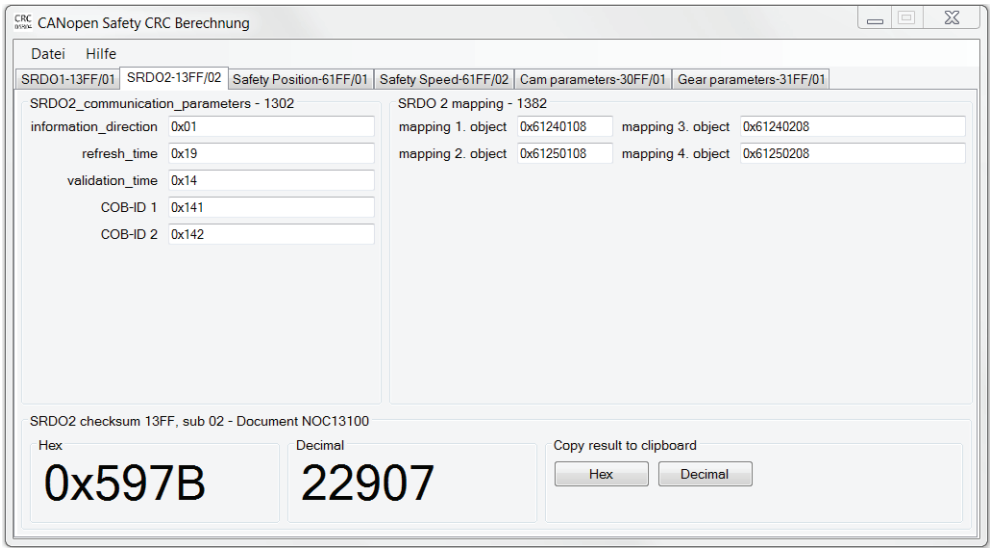
Screenshot for 63xx default, Checksum for 30FF, sub 01



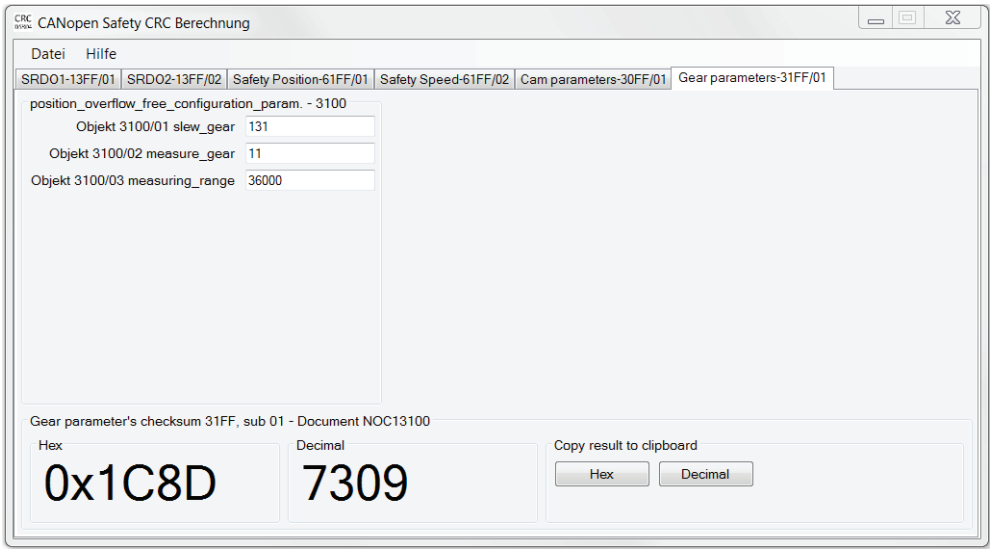
Screenshot for 1301 / 1302 default, Checksum for 13FF, sub 01



Screenshot for 1301 / 1302 default, Checksum for 13FF, sub 02



Screenshot for 3100, sub 01 to sub 03 default, Checksum for 31FF, sub 01



12 Examples for parameterization

12.1 Object 6100 safety_position_configuration_parameters. Set 6100/01 safety_code_sequence: 1=CCW

Step	Description	ID	DLC	Byte								Comment
				1	2	3	4	5	6	7	8	
				cs	Index		Sub-Index	Value				
1	Set preoperational	0	2	80	0d							
2	Deactivating 30fe/00:cam_valid_flag	60d†	8	2f	fe	30	00	00	00	00	00	30fe/00: 00
		58d†	8	60	fe	30	00	00	00	00	00	
3	Deactivating 61fe/00: safety_configuration_valid	60d	8	2f	fe	61	00	00	00	00	00	61fe/00: 00
		58d	8	60	fe	61	00	00	00	00	00	
4	Set 6100/01 and 6101/01: safety_code_sequence	60d	8	2b	00	61	01	01	00	00	00	6100/01: 0x1 set automatically 6101/01: 0x1
		58d	8	60	00	61	01	00	00	00	00	
5	Calculate CRC checksums (TWK program)											
6	Set 61ff/01: Safety_configuration_signature CRC1	60d	8	2b	ff	61	01	5b	54	00	00	61ff/01: 0x545b
		58d	8	60	ff	61	01	00	00	00	00	
7	Set 61ff/02: Safety_configuration_signature CRC2	60d	8	2b	ff	61	02	e8	a3	00	00	61ff/02: 0xa3e8 *
		58d	8	60	ff	61	02	00	00	00	00	
7a	Activating 31fe/00: gear_valid_flag (**only slewing ring 'S'**)	60d	8	2f	fe	31	00	a5	00	00	00	31fe/00: 0xa5
		58d	8	60	fe	31	00	00	00	00	00	
8	Activating 61fe/00: safety_configuration_valid	60d	8	2f	fe	61	00	a5	00	00	00	61fe/00: 0xa5
		58d	8	60	fe	61	00	00	00	00	00	
9	Activating 30fe/00: cam_valid_flag	60d	8	2f	fe	30	00	a5	00	00	00	30fe/00: 0xa5
		58d	8	60	fe	30	00	00	00	00	00	
10	Activating 13fe/00: configuration_valid	60d	8	2f	fe	13	00	a5	00	00	00	13fe/00: 0xa5
		58d	8	60	fe	13	00	00	00	00	00	
11	Save _all_parameters 1010/01	60d	8	23	10	10	01	73	61	76	65	1010/01: 0x65766173 (save) See object for details
		58d	8	60	10	10	01	00	00	00	00	
12	Deactivating 13fe/00: configuration_valid (If no SRDO is necessary. Just PDO)	60d	8	2f	fe	13	00	00	00	00	00	13fe/00: 0x0
		58d	8	60	fe	13	00	00	00	00	00	
13	Set operational	0	2	01	00							

† This identifiers refer to device node ID: 13_{dez} (= d_{hex}) → Master: 600 + node ID = 600 + d = 60d, slave: 580 + node ID = 58d.

* In this case: encoder with slewing ring functionality (See chapter 9.13.8): Speed integration time = 100, Speed multiplier = 100, speed divider = 10

Legend:
- cs Command specifier
- 2f Byte
- 2b Word
- 23 Long

Link to the program on www.twk.de: Look at device NOCN/HBN etc. → download "Software Safety CRC".
Or use the following link: www.twk.de/files/CRC-Calculator20.zip. Description file: CRC14076
Please ask our technicians for xml files for special versions of encoders or look at www.twk.de.

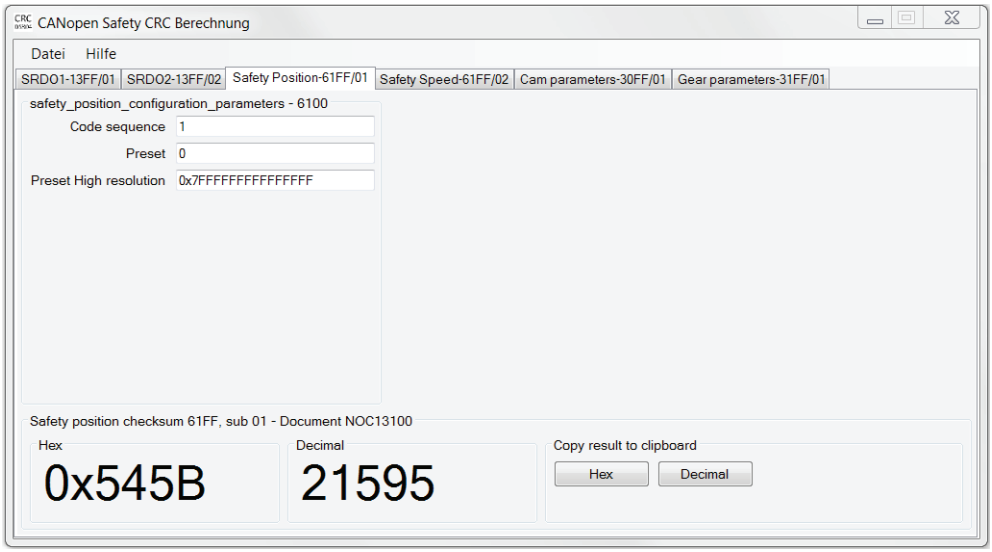


Fig 1, for 61FF, sub 01, every version of NOCN

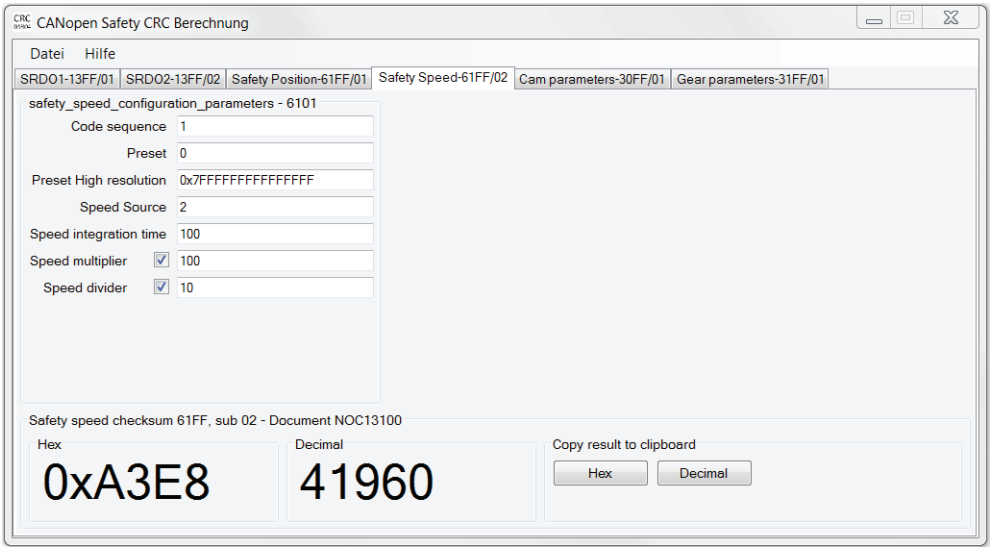


Fig 2a, for 61FF, sub 02, NOCN with code ‚S‘ (With ‘Speed multiplier’ and ‘Speed divider’)

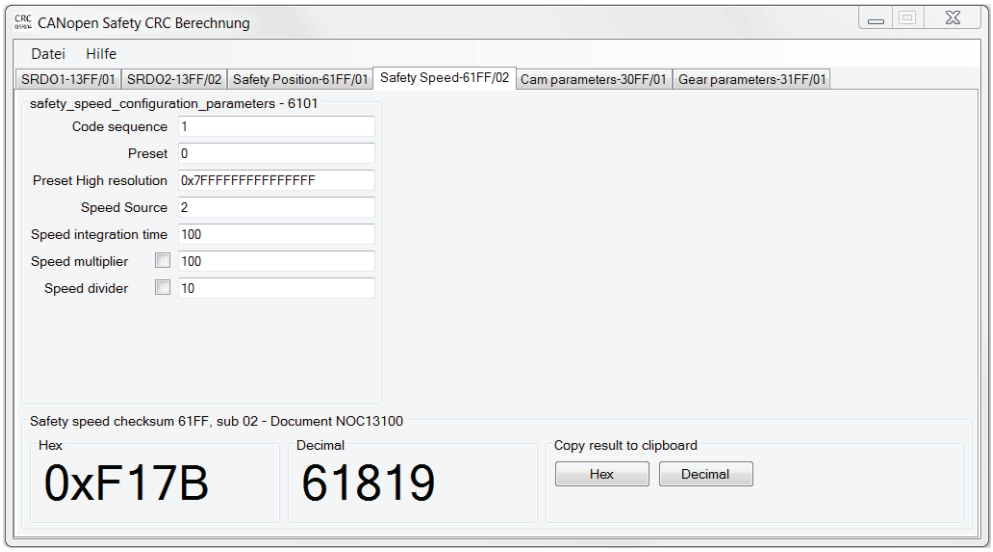


Fig 2b, for 61FF, sub 02, Standard NOCN with code ‚R' (Without 'Speed multiplier' and 'Speed divider')

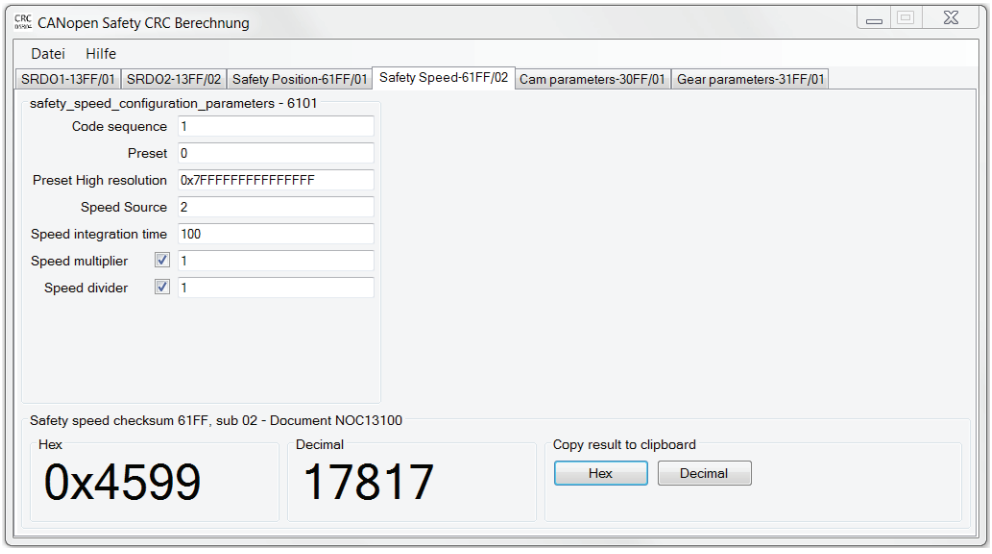


Fig 2c, for 61FF, sub 02, code ‚R', special versions (With 'Speed multiplier' and 'Speed divider')

12.2 Object 6100 safety_position_configuration_parameters. Set 6100/02 safety_preset_value: 0x10a

Step	Description	ID	DLC	Byte								Comment
				1	2	3	4	5	6	7	8	
				cs	Index		Sub-Index	Value				
1	Set preoperational	60d†	8	2f	fe	30	00	00	00	00	00	
2	Deactivating 30fe/00:cam_valid_flag	58d†	8	60	fe	30	00	00	00	00	00	30fe/00: 00
		60d	8	2f	fe	61	00	00	00	00	00	
3	Deactivating 61fe/00: safety_configuration_valid	58d	8	60	fe	61	00	00	00	00	00	61fe/00: 00
		60d	8	2b	00	61	02	0a	01	00	00	
4	Set 6100/02 and 6101/02: safety_code_sequence	58d	8	60	00	61	02	00	00	00	00	6100/02: 0x1 set automatically 6101/02: 0x10a
		58d	8	60	00	61	01	00	00	00	00	
5	Calculate CRC checksums (TWK program)											
6	Set 61ff/01: Safety_configuration_signature CRC1	60d	8	2b	ff	61	01	96	7e	00	00	61ff/01: 0x7e96
		58d	8	60	ff	61	01	00	00	00	00	
7	Set 61ff/02: Safety_configuration_signature CRC2	60d	8	2b	ff	61	02	ba	15	00	00	61ff/02: 0x15ba *
		58d	8	60	ff	61	02	00	00	00	00	
8	Activating 61fe/00: safety_configuration_valid	60d	8	2f	fe	30	00	a5	00	00	00	61fe/00: 0xa5
		58d	8	60	fe	30	00	00	00	00	00	
9	Activating 30fe/00: cam_valid_flag	60d	8	2f	fe	13	00	a5	00	00	00	30fe/00: 0xa5
		58d	8	60	fe	13	00	00	00	00	00	
10	Activating 13fe/00: configuration_valid	60d	8	23	10	10	01	73	61	76	65	13fe/00: 0xa5
		58d	8	60	10	10	01	00	00	00	00	
11	Save _all_parameters 1010/01	60d	8	2f	fe	13	00	00	00	00	00	1010/01: 0x65766173 (save) See object for details
		58d	8	60	fe	13	00	00	00	00	00	
12	Deactivating 13fe/00: configuration_valid (If no SRDO is necessary. Just PDO)	0	2	01	00							13fe/00: 0x0
		58d	8	60	fe	13	00	00	00	00	00	
13	Set operational	0	2	01	00							

† This identifiers refer to device node ID: 13_{dez} (= d_{hex}) → Master: 600 + node ID = 600 + d = 60d, slave: 580 + node ID = 58d.

* In this case: encoder with slewing ring functionality (See chapter 9.13.8). Speed integration time = 100, Speed multiplier = 100, speed divider = 10

Legend:
- cs Command specifier
- 2f Byte
- 2b Word
- 23 Long

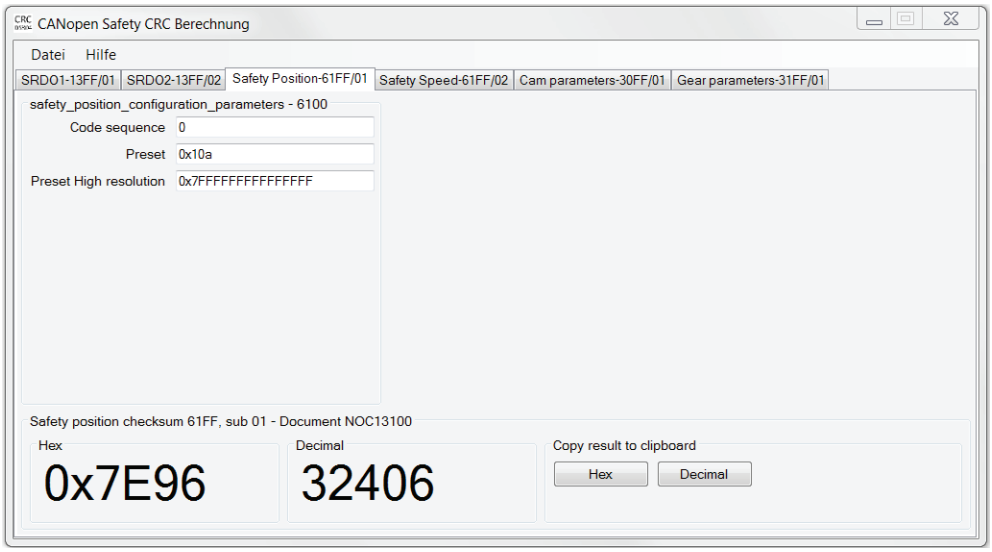


Fig 3, for 61FF, sub 01

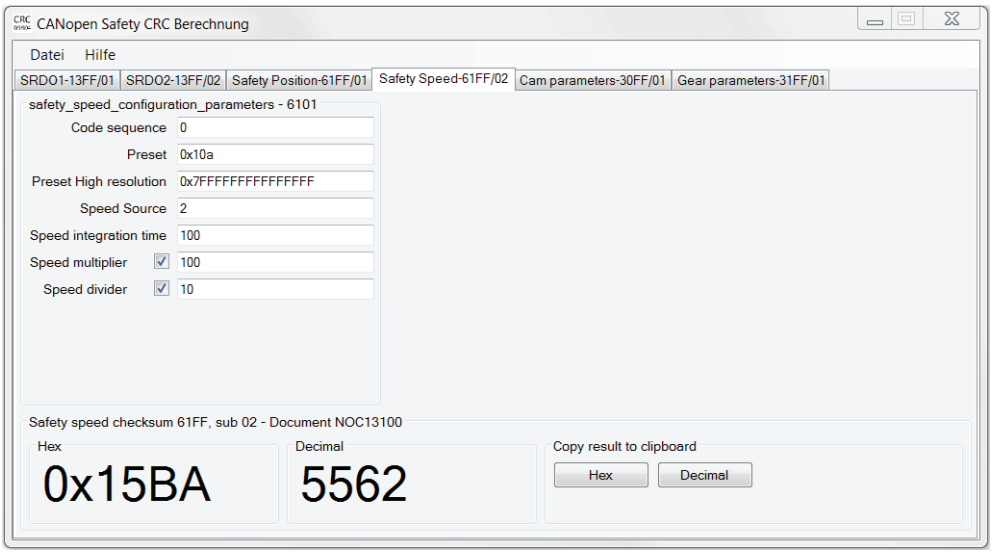


Fig 4, for 61FF, sub 02 (See chapter 12.1 for details concerning ‘Speed multiplier’ and ‘Speed divider’)

12.3 Object 6100 safety_speed_configuration_parameters. Set 6101/05 safety_speed_integration_time: 0xc8

Step	Description	ID	DLC	Byte								Comment
				1	2	3	4	5	6	7	8	
				cs	Index		Sub-Index	Value				
1	Set preoperational	0	2	80	0d							
2	Deactivating 30fe/00:cam_valid_flag	60d†	8	2f	fe	30	00	00	00	00	00	30fe/00: 00
		58d†	8	60	fe	30	00	00	00	00	00	
3	Deactivating 61fe/00: safety_configuration_valid	60d	8	2f	fe	61	00	00	00	00	00	61fe/00: 00
		58d	8	60	fe	61	00	00	00	00	00	
4	Set 6101/05: safety_ speed_integration_time	60d	8	2b	01	61	05	c8	00	00	00	6101/05: 0xc8
		58d	8	60	01	61	05	00	00	00	00	
5	Calculate CRC checksums (TWK program)											
6	Set 61ff/02: Safety_configuration_signature CRC1	60d	8	2b	ff	61	02	d0	21	00	00	61ff/02: 0x21d0 *
		58d	8	60	ff	61	02	00	00	00	00	
7	Activating 61fe/00: safety_configuration_valid	60d	8	2f	fe	61	00	a5	00	00	00	61fe/00: 0xa5
		58d	8	60	fe	61	00	00	00	00	00	
8	Activating 30fe/00: cam_ valid_flag	60d	8	2f	fe	30	00	a5	00	00	00	30fe/00: 0xa5
		58d	8	60	fe	30	00	00	00	00	00	
9	Activating 13fe/00: configuration_valid	60d	8	2f	fe	13	00	a5	00	00	00	13fe/00: 0xa5
		58d	8	60	fe	13	00	00	00	00	00	
10	Save _all_parameters 1010/01	60d	8	23	10	10	01	73	61	76	65	1010/01: 0x65766173 (save) See object for details
		58d	8	60	10	10	01	00	00	00	00	
11	Deactivating 13fe/00: configuration_valid (If no SRDO is necessary. Just PDO)	60d	8	2f	fe	13	00	00	00	00	00	13fe/00: 0x0
		58d	8	60	fe	13	00	00	00	00	00	
13	Set operational	0	2	01	00							

† This identifiers refer to device node ID: 13_{dez} (= d_{hex}) → Master: 600 + node ID = 600 + d = 60d, slave: 580 + node ID = 58d.

* In this case: encoder with slewing ring functionality (See chapter 9.13.8). Speed integration time = 100, Speed multiplier = 100, speed divider = 10

Legend:
- cs Command specifier
- 2f Byte
- 2b Word
- 23 Long

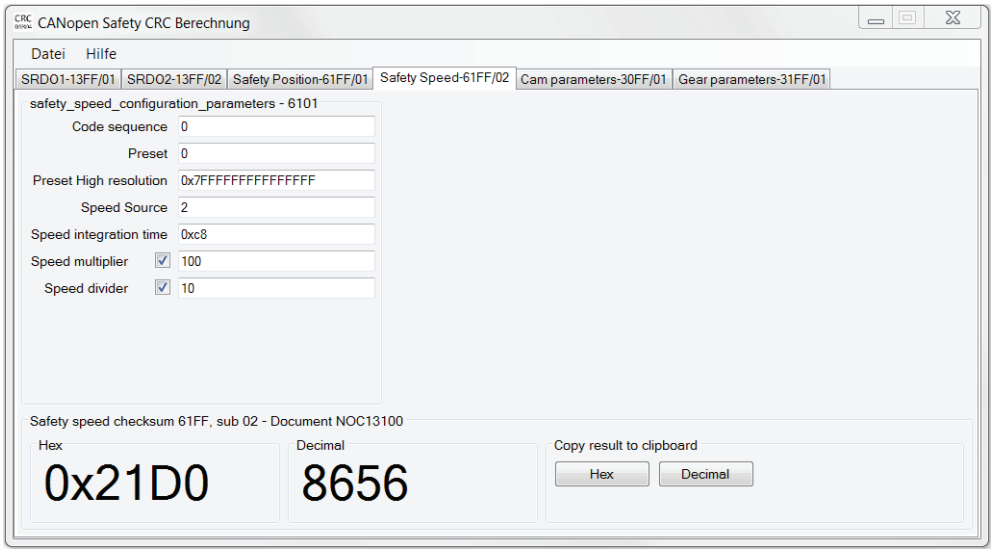


Fig 5, for 61FF, sub 02 (See chapter 12.1 for details concerning ‘Speed multiplier’ and ‘Speed divider’)

12.4 Cam setting

12.4.1 Set 6310/01 (6311/01) cam_1(2)_low_limit and 6320/01(6321/01) cam_1(2)_high_limit: 0x1000 / 0x2000 / 0x3000

Step	Description	ID	DLC	Byte								Comment
				1	2	3	4	5	6	7	8	
				cs	Index		Sub-Index	Value				
1	Set preoperational	0	2	80	0d							
2	Deactivating 30fe/00:cam_valid_flag	60d†	8	2f	fe	30	00	00	00	00	00	30fe/00: 00
		58d†	8	60	fe	30	00	00	00	00	00	
3	Set 6310/01: cam_1_low_ limit	60d	8	23	10	63	01	00	10	00	00	6310/01: 0x1000
		58d	8	60	10	63	00	00	00	00	00	
4	Set 6311/01: cam_2_low_ limit	60d	8	23	11	63	01	00	20	00	00	6311/01: 0x2000
		58d	8	60	11	63	00	00	00	00	00	
5	Set 6320/01: cam_1_ high_limit	60d	8	23	20	63	01	00	20	00	00	6320/01: 0x2000
		58d	8	60	20	63	00	00	00	00	00	
6	Set 6321/01: cam_2_ high_limit	60d	8	23	21	63	01	00	30	00	00	6321/01: 0x3000
		58d	8	60	21	63	00	00	00	00	00	
7	Calculate CRC checksum (TWK program)											
8	Set 30ff/01: cam_CRC_ checksum CRC	60d	8	2b	ff	30	01	ce	f0	00	00	30ff/01: 0xf0ce
		58d	8	60	ff	30	00	00	00	00	00	
9	Activating 30fe/00:cam_ valid_flag	60d	8	2f	fe	30	00	a5	00	00	00	30fe/00: 0xa5
		58d	8	60	fe	30	00	00	00	00	00	
10	Save _all_parameters 1010/01	60d	8	23	10	10	01	73	61	76	65	1010/01: 0x65766173 (save) See object for details
		58d	8	60	10	10	01	00	00	00	00	
11	Set operational	0	2	01	00							

† This identifiers refer to device node ID: 13_{dez} (= d_{hex}) → Master: 600 + node ID = 600 + d = 60d, slave: 580 + node ID = 58d.

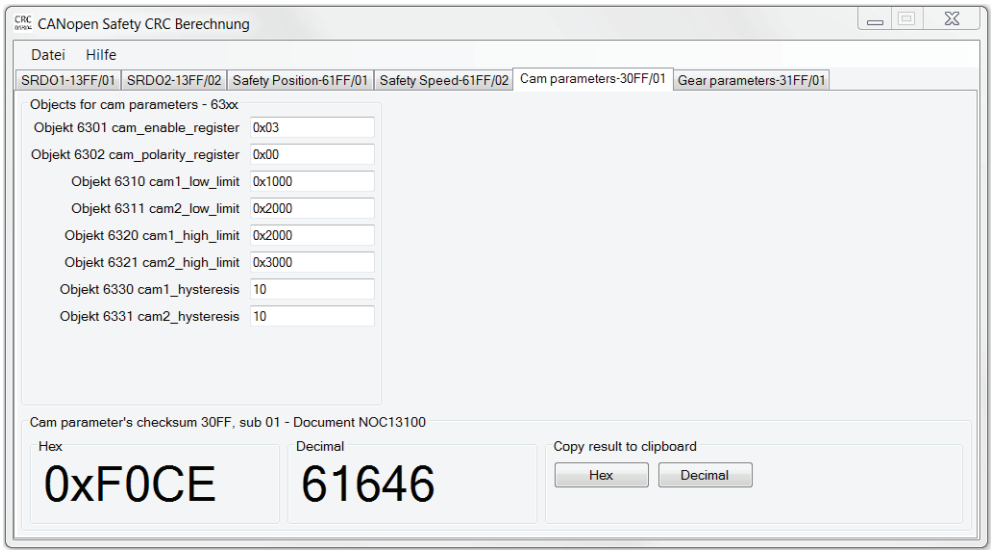


Fig 6, for 30FF, sub 01

12.4.2 Set 6330/01 cam_1_hysteresis and 6331/01 cam_2_hysteresis: 20 digits

Step	Description	ID	DLC	Byte								Comment
				1	2	3	4	5	6	7	8	
				cs	Index		Sub-Index	Value				
1	Set preoperational	0	2	80	0d							
2	Deactivating 30fe/00:cam_valid_flag	60d†	8	2f	fe	30	00	00	00	00	00	30fe/00: 00
		58d†	8	60	fe	30	00	00	00	00	00	
3	Set 6330/01: cam_1_ hysteresis	60d	8	2b	30	63	01	20	00	00	00	6330/01: 0x20
		58d	8	60	30	63	01	00	00	00	00	
4	Set 6331/01: cam_2_hy-steresis	60d	8	2b	31	63	01	20	00	00	00	6331/01: 0x20
		58d	8	60	31	63	01	00	00	00	00	
5	Calculate CRC checksum (TWK program)											
6	Set 30ff/01: cam_CRC_ checksum CRC	60d	8	2b	ff	30	01	95	4a	00	00	30ff/01: 0x4a95
		58d	8	60	ff	30	00	00	00	00	00	
7	Activating 30fe/00:cam_ valid_flag	60d	8	2f	fe	30	00	a5	00	00	00	30fe/00: 0xa5
		58d	8	60	fe	30	00	00	00	00	00	
8	Save _all_parameters 1010/01	60d	8	23	10	10	01	73	61	76	65	1010/01: 0x65766173 (save) See object for details
		58d	8	60	10	10	01	00	00	00	00	
9	Set operational	0	2	01	00							

† This identifiers refer to device node ID: 13_{dez} (= d_{hex}) → Master: 600 + node ID = 600 + d = 60d, slave: 580 + node ID = 58d.

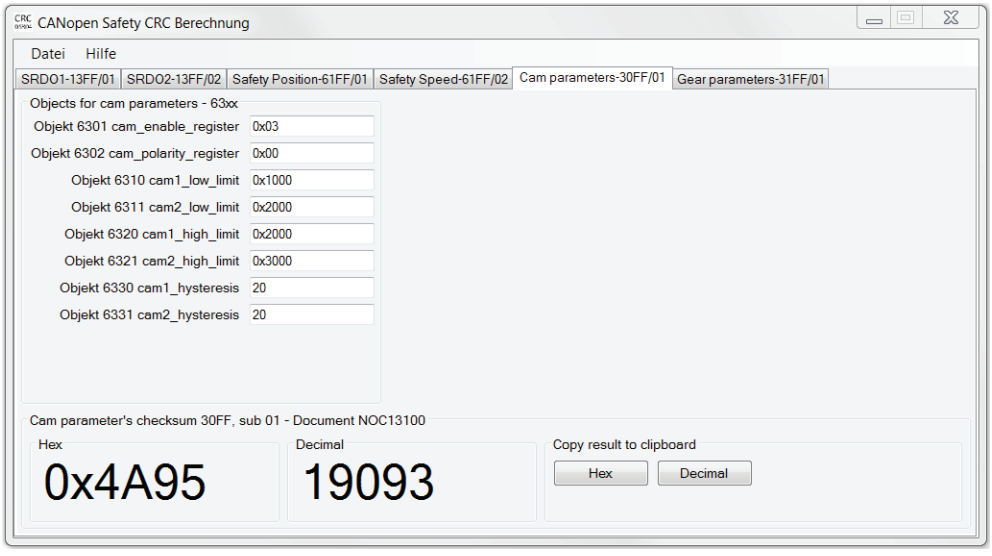


Fig 7, for 30FF, sub 01

12.4.3 Set 6301/01 cam_enable_register and 6302/01 cam_polarity_register: Only cam 1 active and polarity changed

Step	Description	ID	DLC	Byte								Comment
				1	2	3	4	5	6	7	8	
				cs	Index		Sub-Index	Value				
1	Set preoperational	0	2	80	0d							
2	Deactivating 30fe/00:cam_valid_flag	60d†	8	2f	fe	30	00	00	00	00	00	30fe/00: 00
		58d†	8	60	fe	30	00	00	00	00	00	
3	Set 6301/01: cam_enable_register	60d	8	2f	01	63	01	01	00	00	00	6301/01: 0x1 (Just cam 1 active)
		58d	8	60	01	63	01	00	00	00	00	
4	Set 6302/01: cam_polarity_register	60d	8	2f	02	63	01	01	00	00	00	6302/01: 0x1 Cam 1: polarity changed
		58d	8	60	02	63	01	00	00	00	00	
5	Calculate CRC checksum (TWK program)											
6	Set 30ff/01: cam_CRC_checksum CRC	60d	8	2b	ff	30	01	cb	95	00	00	30ff/01: 0x95cb
		58d	8	60	ff	30	00	00	00	00	00	
7	Activating 30fe/00:cam_valid_flag	60d	8	2f	fe	30	00	a5	00	00	00	30fe/00: 0xa5
		58d	8	60	fe	30	00	00	00	00	00	
8	Save _all_parameters 1010/01	60d	8	23	10	10	01	73	61	76	65	1010/01: 0x65766173 (save) See object for detailed information concerning sub-indices!
		58d	8	60	10	10	01	00	00	00	00	
9	Set operational	0	2	01	00							

† This identifiers refer to device node ID: 13_{dez} (= d_{hex}) → Master: 600 + node ID = 600 + d = 60d, slave: 580 + node ID = 58d.

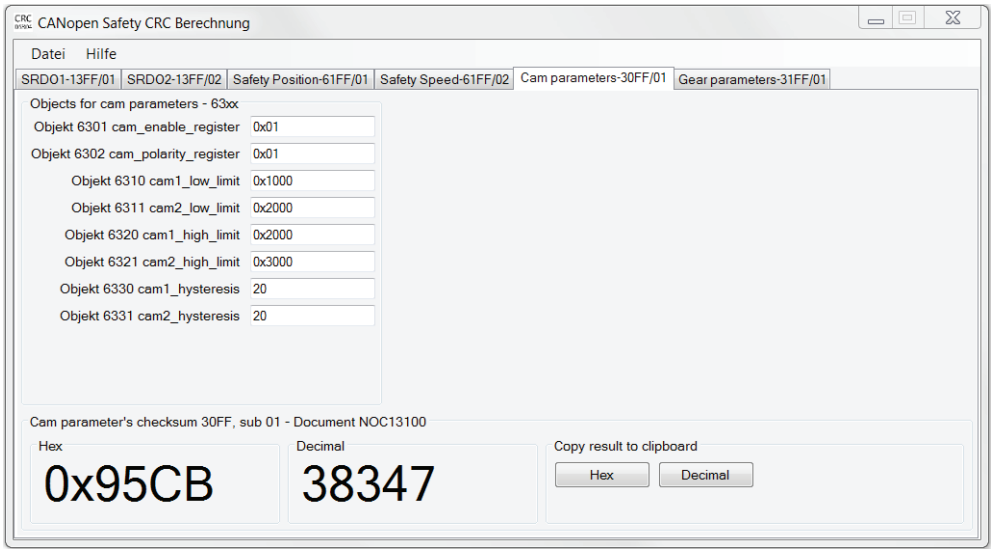


Fig 8, 30FF, sub 01

12.5 Object 1301 SRDO_communication_parameters. Set 1301/02 refresh_time (SCT): 0x200 and Object 1302 SRDO_speed_parameters. Set 1302/02 refresh_time (SCT): 0x200

Step	Description	ID	DLC	Byte								Comment
				1	2	3	4	5	6	7	8	
				cs	Index		Sub-Index	Value				
1	Set preoperational	0	2	80	0d							
2	Deactivating 13fe/00: configuration_valid	60d†	8	2f	fe	13	00	00	00	00	00	13fe/00: 00
		58d†	8	60	fe	13	00	00	00	00	00	
3	Set 1301/02: SRDO_communication_parameters	60d	8	2b	01	13	02	00	02	00	00	1301/02: 0x200
		58d	8	60	01	13	02	00	00	00	00	
4	Set 1302/02: SRDO_ speed_parameters	60d	8	2b	02	13	02	00	02	00	00	1302/02: 0x200
		58d	8	60	02	13	02	00	00	00	00	
5	Calculate CRC checksum (TWK program)											
6	Set 13ff/01: safety_configuration_checksum SRDO1	60d	8	2b	ff	13	01	c2	76	00	00	13ff/01: 0x76c2 * COB-ID1: 0x119 COB-ID2: 0x11a
		58d	8	60	ff	13	01	00	00	00	00	
7	Set 13ff/02: safety_configuration_checksum SRDO2	60d	8	2b	ff	13	02	87	3c	00	00	13ff/02:0x3c87 * COB-ID1: 0x159 COB-ID2: 0x15a
		58d	8	60	ff	13	02	00	00	00	00	
8	Activating 13fe/00: configuration_valid	60d	8	2f	fe	13	00	a5	00	00	00	13fe/00: 0xa5
		58d	8	60	fe	13	00	00	00	00	00	
9	Save _all_parameters 1010/01	60d	8	23	10	10	01	73	61	76	65	1010/0: 0x65766173 (save) See object 1010 for detailed information concerning sub-indices !
		58d	8	60	10	10	01	00	00	00	00	
10	Deactivating 13fe/00: configuration_valid (If no SRDO is necessary. Just PDO)	60d	8	2f	fe	13	00	00	00	00	00	13fe/00: 0x0
		58d	8	60	fe	13	00	00	00	00	00	
11	Set operational	0	2	01	00							

† This identifiers refer to device node ID: 13_{dez} (= d_{hex}) → Master: 600 + node ID = 600 + d = 60d, slave: 580 + node ID = 58d.

* Pay attention to the different mapping parameters and the different COB-Ids for these two CRC checksum calculations.

Legend:

	cs	Command specifier
	2f	Byte
	2b	Word
	23	Long

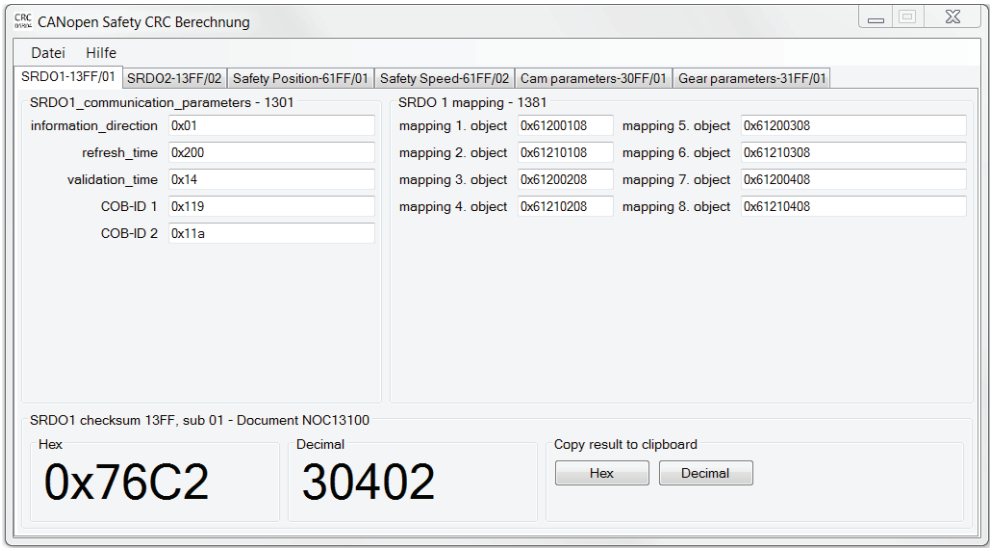


Fig 9, for 13FF, sub 01

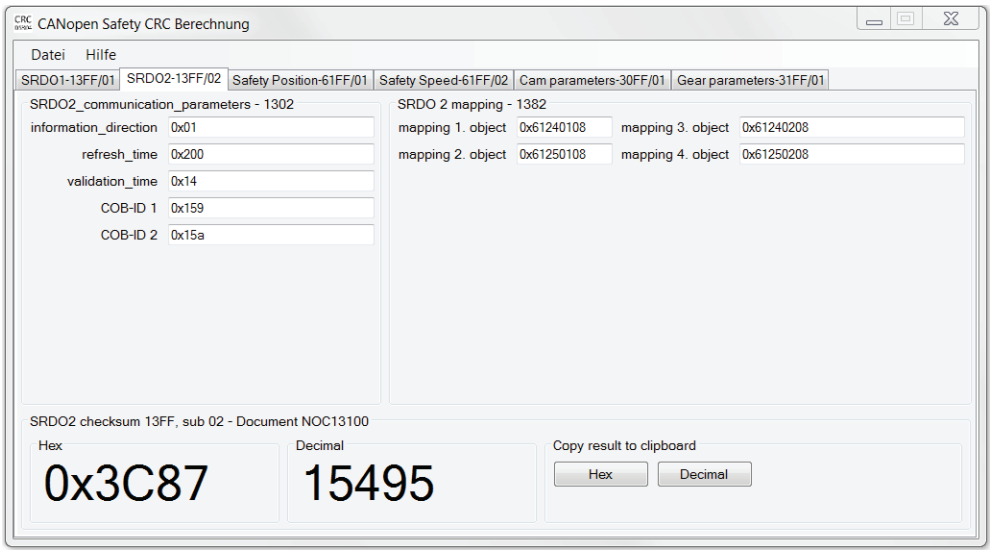


Fig 10, for 13FF, sub 02

12.6 LMT objects. Set 2000/00 node ID: 0x11 (= 17dez)

See as well chapter 9.10 and 9.5.8 (object 2001 description and saving procedure)

Step	Description	ID	DLC	Byte								Comment
				1	2	3	4	5	6	7	8	
				cs	Index		Sub-Index	Value				
1	Set preoperational	0	2	80	0d							
2	Deactivating 13fe/00: configuration_valid	60d†	8	2f	fe	13	00	00	00	00	00	13fe/00: 00
		58d†	8	60	fe	13	00	00	00	00	00	
3	Set 2000/00 node ID: 0x11	60d	8	2f	00	20	00	**11**	00	00	00	2000/00: 0x11
		58d	8	60	00	20	00	00	00	00	00	
4	Save_LMT_parameters 1010/04	60d	8	23	10	10	04	73	61	76	65	1010/04: 0x65766173 (save)
		58d	8	60	10	10	04	00	00	00	00	
5	Set 1301/05: COB-ID1 **	60d	8	23	01	13	05	**21**	**01**	00	00	1301 / 05: 121 COB_ID_1 = FF + 2 * Node ID
		58d	8	60	01	13	05	00	00	00	00	
6	Set 1301/06: COB-ID2 **	60d	8	23	01	13	06	**22**	**01**	00	00	1301 / 06: 122 COB_ID_2 = 100 + 2 * Node ID
		58d	8	60	01	13	06	00	00	00	00	
7	Set 1302/05: COB-ID1 **	60d	8	23	02	13	05	**61**	**01**	00	00	1302 / 05: 161 COB_ID_1 = 13F + 2 * Node ID
		58d	8	60	02	13	05	00	00	00	00	
8	Set 1302/06: COB-ID2 **	60d	8	23	02	13	06	**62**	**01**	00	00	1302 / 06: 162 COB_ID_2 = 140 + 2 * Node ID
		58d	8	60	02	13	06	00	00	00	00	
9	Calculate CRC checksum (TWK program)											
10	Set 13ff/01: safety_configuration_checksum SRDO1	60d	8	2b	ff	13	01	**40**	**DC**	00	00	13ff/01: 0xDC40 * COB-ID1: 0x121 COB-ID2: 0x122
		58d	8	60	ff	13	01	00	00	00	00	
11	Set 13ff/02: safety_configuration_checksum SRDO2	60d	8	2b	ff	13	02	**CC**	**81**	00	00	13ff/02: 0x81CC * COB-ID1: 0x161 COB-ID2: 0x162
		58d	8	60	ff	13	02	00	00	00	00	
12	Activating 13fe/00: configuration_valid	60d	8	2f	fe	13	00	a5	00	00	00	13fe/00: 0xa5
		58d	8	60	fe	13	00	00	00	00	00	
13	Save _all_parameters 1010/01	60d	8	23	10	10	01	73	61	76	65	1010/01: 0x65766173 (save) See object 1010 for detailed information concerning sub-indices !
		58d	8	60	10	10	01	00	00	00	00	
14	Power off / Power on											
15	Set operational	0	2	01	00							

† This identifiers refer to device node ID: 13_{dez} (= d_{hex}) → Master: 600 + node ID = 600 + d = 60d, slave: 580 + node ID = 58d.

* Pay attention to the different mapping parameters and the different COB-Ids for these two CRC checksum calculations.

** These steps are only necessary when a node ID over 32 dez (33 ….) is entered. For node ID ≤ 32 the COB IDs are adopted automatically and with MSB = 0 (=SRDO active). For node ID ≥ 33 the COB IDs are adopted with MSB = 1 (SRDO inactive). It has to be transmitted by customer this COB ID with MSB = 0.

Legend:

cs	Command specifier
2f	Byte
2b	Word
23	Long

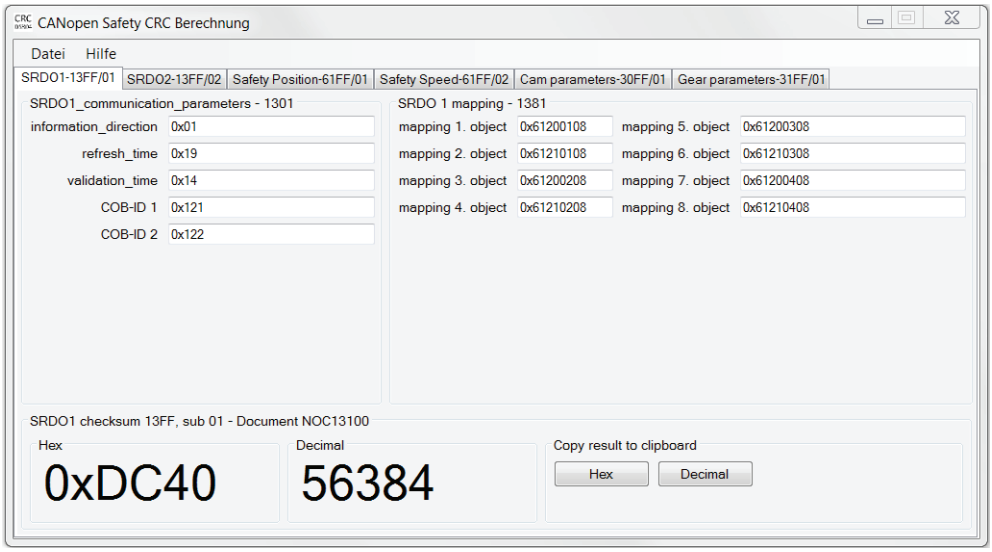


Fig 11, for 13FF, sub 01

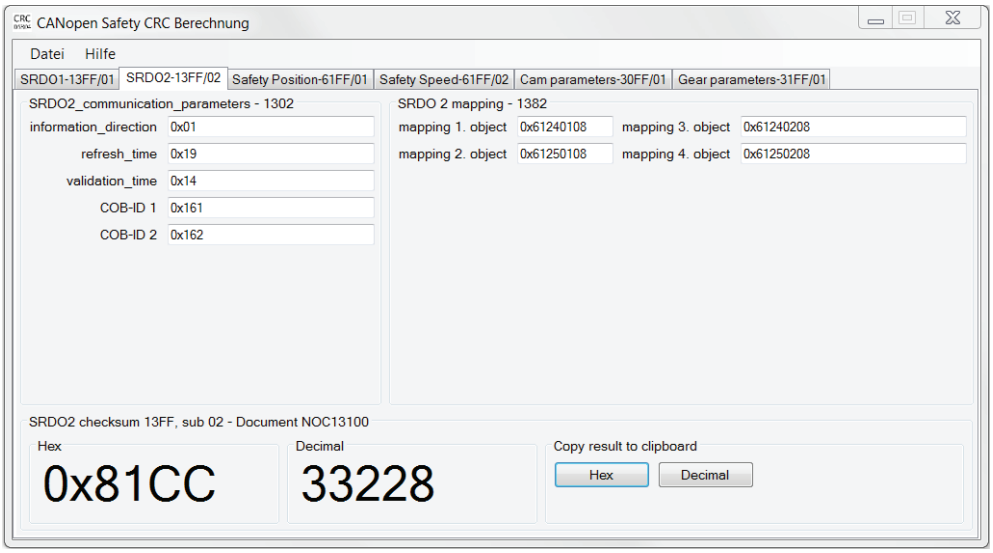


Fig 12, for 13FF, sub 02

12.7 LMT objects. Set 2001/00 bit rate: 0x2 (=500 kbps)

See as well chapter 9.10 and 9.5.8 (object 2001 description and saving procedure)

Step	Description	ID	DLC	Byte								Comment
				1	2	3	4	5	6	7	8	
				cs	Index		Sub-Index	Value				
1	Set preoperational	0	2	80	0d							
2	Set bit_rate 2001/00	60d†	8	2f	01	20	00	02	00	00	00	2001/00: 0x2 (500 kbps)
		58d†	8	60	01	20	00	00	00	00	00	
3	Save_LMT_parameters 1010/04	60d	8	23	10	10	04	73	61	76	65	1010/04: 0x65766173 (save) See object 1010 for detailed information concerning sub-indices !
		58d	8	60	10	10	04	00	00	00	00	
4	Set operational	0	2	01	00							

† This identifiers refer to device node ID: 13_{dez} (= d_{hex}) → Master: 600 + node ID = 600 + d = 60d, slave: 580 + node ID = 58d.

Legend:
- cs Command specifier
- 2f Byte
- 2b Word
- 23 Long

12.8 Behavior of full redundant encoder in case of error (i.e. TBN-N97 / TBN-N100)

(Entry in object 1029: 0x01 - no change of NMT status)

After boot up the two nodes will be started via NMT_Reset and via NMT_Start the nodes will be set operational

Step	Description	ID	DLC	Byte								Comment
				1	2	3	4	5	6	7	8	
1	Bootup message node 1	0701	1	00								
2	Bootup message node 2	0702	1	00								
3	Reset node 1	0000	2	81	01							
4	Error message of node 2	0082	8	FF	FF	81	00	08	01	01	01	node 1 failed
5	Bootup message node 1	0701	1	00								
6	Error reset of node 2	0082	8	00	00	00	00	00	00	00	00	
7	NMT start node 1	0000	2	01	01							
8	PDO1 node 1	0181	6	51	13	00	00	00	00			
9	PDO1 node 1	0181	6	51	13	00	00	00	00			
10	Reset node 2	0000	2	81	02							
11	PDO1 node 1	0181	6	51	13	00	00	00	00			
12	Error message of node 1	0081	8	FF	FF	81	00	08	01	01	01	node 2 failed
13	PDO1 node 1	0181	6	51	13	00	00	00	00			
14	PDO1 node 1	0181	6	51	13	00	00	00	00			
15	Bootup message node 2	0702	1	00								
16	PDO1 node 1	0181	6	51	13	00	00	00	00			
17	Error reset of node 1	0081	8	00	00	00	00	00	00			
18	NMT start node 2	0000	2	01	02							
19	PDO1 node 2	0182	6	52	13	00	00	00	00			
20	PDO1 node 1	0181	6	51	13	00	00	00	00			
21	PDO1 node 2	0182	6	52	13	00	00	00	00			
22	PDO1 node 1	0181	6	51	13	00	00	00	00			
23	PDO1 node 2	0182	6	52	13	00	00	00	00			
24	PDO1 node 1	0181	6	51	13	00	00	00	00			
25	PDO1 node 2	0182	6	52	13	00	00	00	00			
26	PDO1 node 1	0181	6	51	13	00	00	00	00			
27	PDO1 node 2	0182	6	52	13	00	00	00	00			

13 Recognition aid for error / emergency messages

Appearance of error / emergency messages in a trace and in object 1003 resp.
(In this chapter other SIL2 safety components are mentioned as well - inclinometer NBN and vibration sensor NVA115)

If the encoder sends an error message it has 8 Bytes in the trace and looks as follows:

FF FF 81 00 10 02 0D 0F → Example for a certain message (in this case a “cam error” at NOCN. See below).

The emergency message is sent on identifier **80 + node ID.**

To identify this error message and to read its content you have to look at the different bytes. The first two bytes (B0 and B1, B’s counted from ‘0’) in the error message make the following distinguishing: “Error in sensor” (FF FF) or “Error in CAN communication” (81 10 or 81 20 or 81 40) due to bus problems etc.. In most cases the content in byte 2 (B2) correlates to byte 0 and 1:

FF FF → 81 in byte 2, 81 = Manufacturer specific error
81 . . → 11 in byte 2, 11 = CAN communication error

Byte 3 (B3) and byte 4 (B4) display the content of object 6503 which shows the error category (alarm specification). B3 displays the low byte of 6503 and it is always 00. B4 displays the high byte of 6503. The possible categories describe malfunctions of the device and no CAN communication errors.

The next three bytes (B5, B6, B7) display detailed information which error has occurred depending on the error category B4 (= content of 6503). These errors are only errors which are caused by

- device internal CRC parameter error (internal check of parameter set)
- failure in the power supply
- encoder malfunctions

The bytes B4, B5, B6 and B7 are shown in object 1003: “pre_defined_error_field” as well. In this object all by the sensor recognized errors are stored (20 errors at maximum - then errors will be overwritten). With 1003 the customer can look if an error has occurred in the sensor and he can see this error at a later date. In 1003 you can see malfunctions of the device but no CAN communication errors. The error history object 1003 is not implemented in every device.

The sequence of B4 to B7 is reverse in 1003 in comparison to the emergency message:

Trace: FF FF 81 00 **10 02 0D 0F** → object 1003: **0 x 0F 0D 02 10**

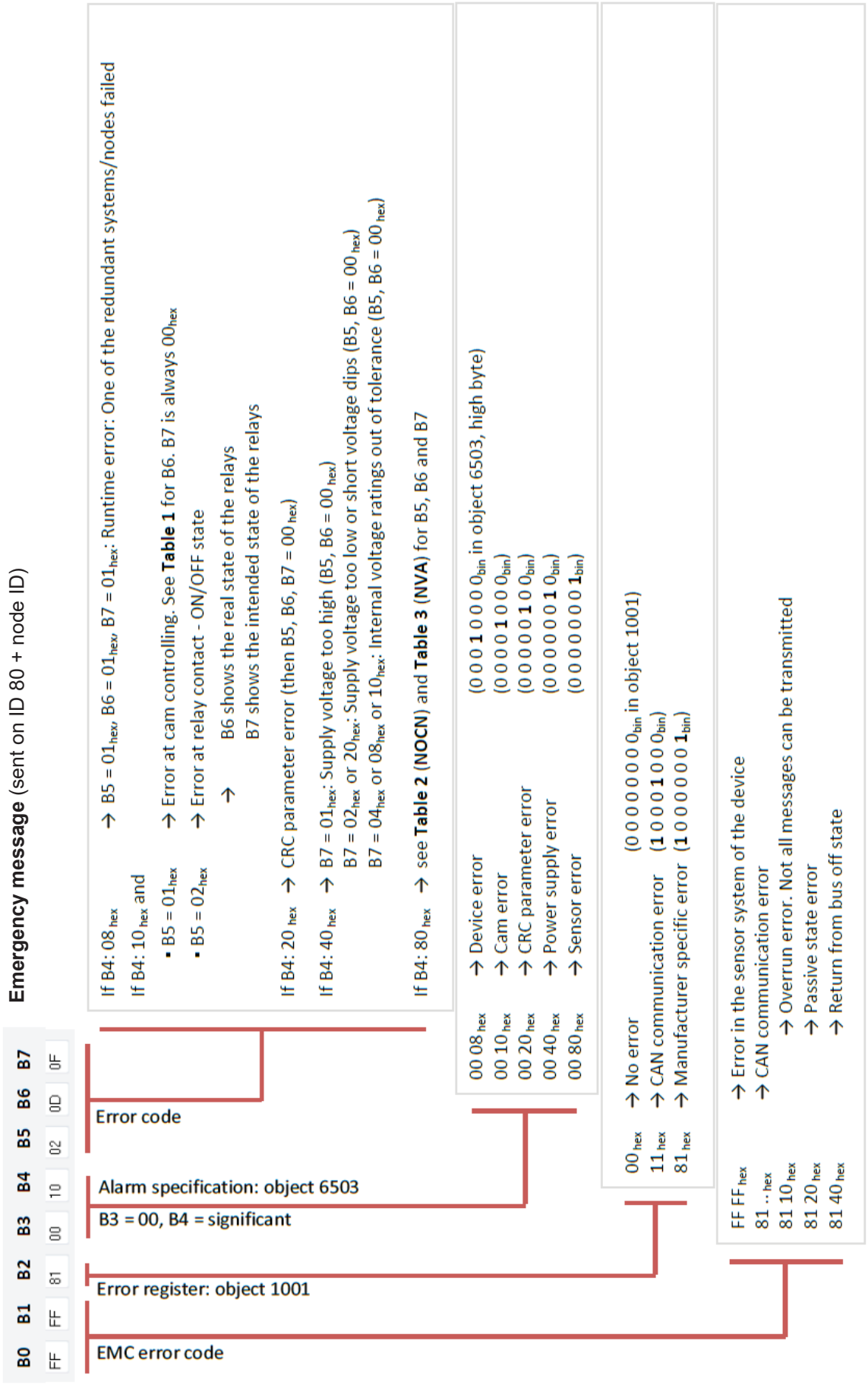
B4 (= 10 in the above trace example) displays the error category. B5 to B7 show more details which can be read in the tables and remarks on the following pages and in further documents for different encoders which are available on request for some devices.

The definition of B4 to B7 depends on the device (e.g. NOCN, NVA, NBN, HBN and so on) due to the fact that different devices have different possibilities of errors (e.g. NOCN with cams can have a cam-malfunction but NBN not).

The example on the next page shows an overview and an aid for the recognition of an error message at device NOCN with 2 switching contacts (cams):

B0 & B1:	FF FF	Error in the sensor system (device)
B2:	81	Manufacturer specific error
B3 & B4:	00 10	Cam error
B5:	02	Error at output contact ON/OFF
B6:	0D	Real state of the relays (→ R4 R3 R2 R1 = 1 1 0 1bin = Dhex)
B7:	0F	Intended state of the relays (1 1 1 1bin = Fhex)

Means: → Only 3 of the required 4 relays are closed (i.e. 1 + 3 + 4). One relay failed (i.e. 2)
→ Error of the related relay / electronic
→ Safety switching contact S1 doesn’t close because S1 consists of R1 + R2 connected in series



Remark: Not every possible error is displayed, but the most important ones.

Table 1 (NOCN with cams and NVA115 → cam error - error at cam controlling, B4 = 10, B5 = 01, B7 = 00)

Byte B6:

General error	0x00
No error	0x01
No start flag found	0x02
No end flag found	0x03
Error frame	0x04
CRC error when checked	0x05
Timeout	0x06
Error data length	0x07
Unknown address	0x08
Error message number	0x09
(SLV)	
General error	0x80
No error	0x81
No start flag found	0x82
No end flag found	0x83
Error frame	0x84
CRC error when checked	0x85
Timeout	0x86
Error data length	0x87
Unknown address	0x88
Error message number	0x89

Table 2 (NOCN → sensor error, B4 = 80)

Byte B5 (channel number):

No error	0x00
Error channel 1	0x01
Error channel 2	0x02
Error: No channel	0x03
Error invers calculation	0x04 (only slewing ring code ‘S’, when 31FE is not set A5 again)

Byte B6 (error type):

No error	0x00
Error communication	0x01
Error sensor	0x02
Error speed overflow	0x03
Error synchronism	0x04

Byte B7 (error code):

(Error in communication)	
No error	0x00
Error time out	0x01
Error CRC	0x02
Error MSG_NBR	0x03
Error position	0x04
(Error of sensor)	
CRC error code range	0x06
Chip error of detection ICs	0x0A
Error in gear detection	0x12
Error of parameter	0x42
Error of calibration parameters	0x82

Table 3 (NVA115 → sensor error, B4 = 80)

Byte B5 → always 00
Byte B6 → always 00

Byte B7	No error	0x00
	Time out error of filter	0x01
	Position monitoring	0x04
	Zero position	0x05
	Communication sampling	0x06
	No valid configuration (32FE)	0x07
	Error calibration data	0x08
	Error sensor static value	0x09
	Error sensor ID 1	0x10
	Error sensor ID 2	0x11

Modell	B0	B1	B2	B3	B4	B5	B6	B7	Objekt 1003 (B4-B7)
NOCN	FF	FF	81	00	80	02	02	12	0x12020280

Error in the Sensor system of the device
Manufactur specific error
Sensor error
Error Ch 2
Error Sensor
Error in gear detection

NOCN	FF	FF	81	00	40	00	00	02	0x02000040

Error in the Sensor system of the device
Manufactur specific error
Power supply error
always 00
always 00
too less voltage

NOCN cam switch	FF	FF	81	00	10	02	0d	0F	0x0F0D0210

Error in the Sensor system of the device
Manufactur specific error
Cam error
Error at relay contact- ON/OFF state
"is" =0d=1101
"should"=0F=1111

NOCN cam switch	FF	FF	81	00	10	01	00	86	0x86000110

Error in the Sensor system of the device
Manufactur specific error
Cam error
Error cam controlling
General error
Time out

The pre_defined_error_field object 1003 in which several errors are stored during lifetime contains 4 bytes: B7 B6 B5 B4. E.g.: Read out value of 1003: 0 x 02 00 00 40

→
B7 = 02 > supply voltage too low
B6 = 00
B5 = 00
B4 = 0x40 > power supply error

Subject to completion.