SUPREME SENSORING
TWK

Document no.: **NOC** 15597 EE
Date: 16 Mar 2022

Switching cam encoder NOCIO with IO-Link interface

Data sheet NOC 15893

IO-Link

User manual

UL CERTIFIED SAFETY US·CA E517808

PLd ISO 13849 FUNCTIONAL SAFETY SENSOR

SIL2 IEC 61508 FUNCTIONAL SAFETY SENSOR

POB 10 50 63 ■ 40041 Düsseldorf ■ Germany
Tel. +49/211/96117-0 ■ Fax +49/211/63 77 05
info@twk.de ■ www.twk.de

Table of contents

1 Safety instructions

1.1 Scope

This user manual is valid exclusively for the following absolute switching cam encoder with IO-Link interface:

- NOCIO79-xxxx-xxxxRxxxxSxxxLxx

1.2 Documentation

The following documents must be observed:

- The owner's system-specific operating instructions
- This user manual
- Data sheet number 15893
- The connection assignment enclosed with the device
- Assembly instructions number 16169 available on our homepage

1.3 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.

1.4 Commissioning

- The relevant device may only be set up and operated in combination with this manual and the documentation specified under point 1.2.
- The product is intended for use in industrial machinery applications as defined in the Electrical Standard for Industrial Machinery, NFPA 79 and EN 60204-1
- 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.
- Do not update the firmware while device is in operation in the application. Customer has to ensure that the safety contacts are open in the safety chain.
- Check all electrical connections before commissioning the system.
- If the equipment is used in a manner not specified by the manufacturer, the protection provided by the equipment may be impaired.

2 General information

General functional principle

NOCIO is a backlash-free electronic switching cam encoder with up to two galvanically separated safety switching outputs. The NOCIO is a safety device and certified according to IEC 61508 (SIL2) and ISO 13849 (PLd).

A configurable multiturn absolute encoder with IO-Link interface and switching cams with separate controller are integrated in the compact housing. The supply voltage, IO-Link signal and safety switching contact are each galvanically separated from each other.

Encoder and cam parameters can be accessed and changed via the IO-Link interface.

A special shaft design appropriate to the play-compensating measurement gear ZRS is available.

NOCIO has a number of error detection procedures and self diagnosis functions as required by a safety design. Any detected error (at encoder and cams) is indicated by transmitting an error message via IO-Link and by opening the safety contact. Errors are recorded in a device error history object (0x1003) for diagnosis.

To avoid parameter changes during normal operation, the parameters can only be amended when the rotating speed of the encoder shaft v_s is approximately zero. Otherwise, the new parameters are not accepted and the valid flags 0xXXFE cannot be set to 0xA5. An error message is generated.

Rotary encoder

The rotary encoder has a standard IO-Link interface (no safety protocol). The position signal has a resolution of up to 15(16) bits per revolution with a measuring range of 4096 revolutions. The resolution can be reduced on customer request. The position reading can be referenced / preset using IO-Link protocol. The signal path (CW/CCW) can be set.

The NOCIO also provides a speed and acceleration signal via the cyclic IO-Link data. The switching outputs can be triggered by position, speed and/or acceleration.

Switching outputs (→ cams)

The safety switching outputs are potential-free and galvanically separated. They are controlled by electronic cams which can be configured according to the customers application.

Switching outputs are implemented using forcibly driven safety relays with a long service life. Each safety contact consist of two relays connected in series. These two relays switch with a brief offset (in the millisecond range). This guarantees reliable contact separation - even when switching high external voltages and/or currents. A separate controller unit monitors the function of the switching outputs. If incorrect switching is detected, an error will be output and the relays open.

The switching contacts are normally open contacts. In normal operating the contact is closed and the relay coils are powered. They open when the position / speed / acceleration limits are reached, a fault is detected, the supply voltage is too low or when the NOCIO is shut off completely.

The triggering limits (low and high limits) can be configured via IO-Link. Customer-specific switching procedures can also be implemented in the factory.

The switching information for the cams/relays is taken from the rotary encoder. The switching outputs are activated and deactivated without backlash, electronically and wear-free in contrast to a mechanical switching cam encoder. A small hysteresis can be activated to avoid contact flickering.

Direct and alternating voltage can be switched.

Alternative connector assignments can be realised on customer's request.

3 Installation instructions

3.1 General

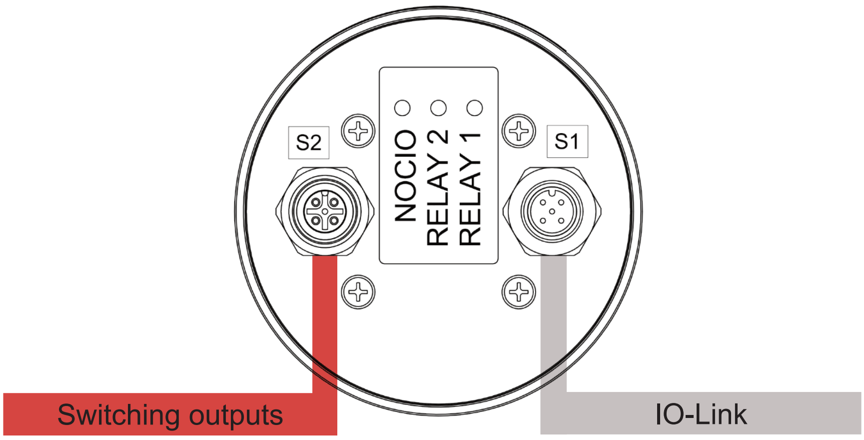
For the planning of the cabling and installation please note the "IO-Link Design Guideline, Order No. 10.912" from the IO-Link community (https://io-link.com/en/).

3.2 Electrical connection

The NOCIO79 has two connectors M12 A-coded:

Connection	name	connector
IO-Link	S1	M12x4 A-coded pins
Switching outputs	S2	M12x5 A-coded socket

The IO-Link port is a class A type. This means a three-core unshielded control line is sufficient for cabling. Four core cables are permissible because the Pin 2 of the NOCIO is not connected. The switching contacts are normally open contacts. For the detailed connection diagram please refer to the data sheet no. NOC15893.



3 Installation instructions

3.3 Status LEDs

The NOCIO is equipped with one bi-colour device status LED, and up to two LEDs for the relay status (yellow LED for safety contacts, green for non-safety contacts).

Device LED	Relay 2 LED	Relay 1 (safety) LED	Description
green / red	green or yellow	yellow	
green on			Operating voltage available
green flashing			Encoder exchanges data with master Short term: preset acknowledge (see below)
red on			Encoder error (see handbook for details)
red flashing			Hard error (flashing codes see handbook) Short term: preset acknowledge (see below)
		on	Relay 1 (safety) contact is in condition TRUE (contact is closed)
	on		Relay 2 (only if installed) contact is in condition TRUE (contact is closed)

A preset command is acknowledged by the device LED with 2s of flashing at 5 Hz. If the command has been carried out successful, the LED flashes green. If the command is not accepted (e.g. due to inappropriate cam states) the LED flashes red.

3.4 IODD file

An IODD file to integrate the switching cam encoder into a IO-Link master software is available for download on our website www.twk.de. This describes the features of the IO-Link subscriber in the standardised XML format.

Every NOCIO variant has its own IODD file for exact identification.

4 System description

4.1 Principle function diagram of cams and relays

NOCIO provides 4 cams. A cam is a software module which compares the input signal (source) with limits which are deposit as a parameter set for each cam. For each different source (e.g. position, speed and acceleration) a set of limits is valid. Therefore these limits have to be adapted to the chosen source. The other cam parameters as well.

If no limit is exceeded the cam is in the TRUE state. If a limit is exceeded, the cam changes to the FALSE state. But this is only when the related cam is enabled via *Cam enable* (13x0/05). If not enabled the cam is always in the FALSE state, even it is inverted.

When a cam is inverted via *Cam polarity* (13x0/06) the TRUE and FALSE states are swapped. The actual state (limit exceeded or not) of each cam can be read out via *Cam state* (1300). In addition the state of each cam is displayed in the status word (bits 0 to 3). Cam = TRUE means status-bit = 1.The status bits and the status in object 1300 display always the actual state of the cams without regarding the *Acknowledge* command (30FC or bits 0/1 in control byte - see below), because the acknowledge command refers to the relay status only (set TRUE).

The relay outputs can be assigned to any cam (1 to 4) and to more than one cam simultaneously. This is done via object *Relay assign* (30x0/01). Therefore the relays can react to several cam limits and to several sources. The particular relay contact is closed (TRUE) when no limit is exceeded and therefore the cam(s) is (are) TRUE as well. It changes to the FALSE state when limits are exceeded, which is indicated by the cams (FALSE). Therefore we have: *cam = TRUE* means *relay = TRUE* and vice versa. If a cam is inverted by 13x0/06 (cam polarity) the cam is FALSE when no limit is exceeded and TRUE when source value is outside the limits.

If a relay is assigned to more than one cam, the cams are combined with logical "and" (&) → relay = TRUE when cam1 & cam2 & ... = TRUE.

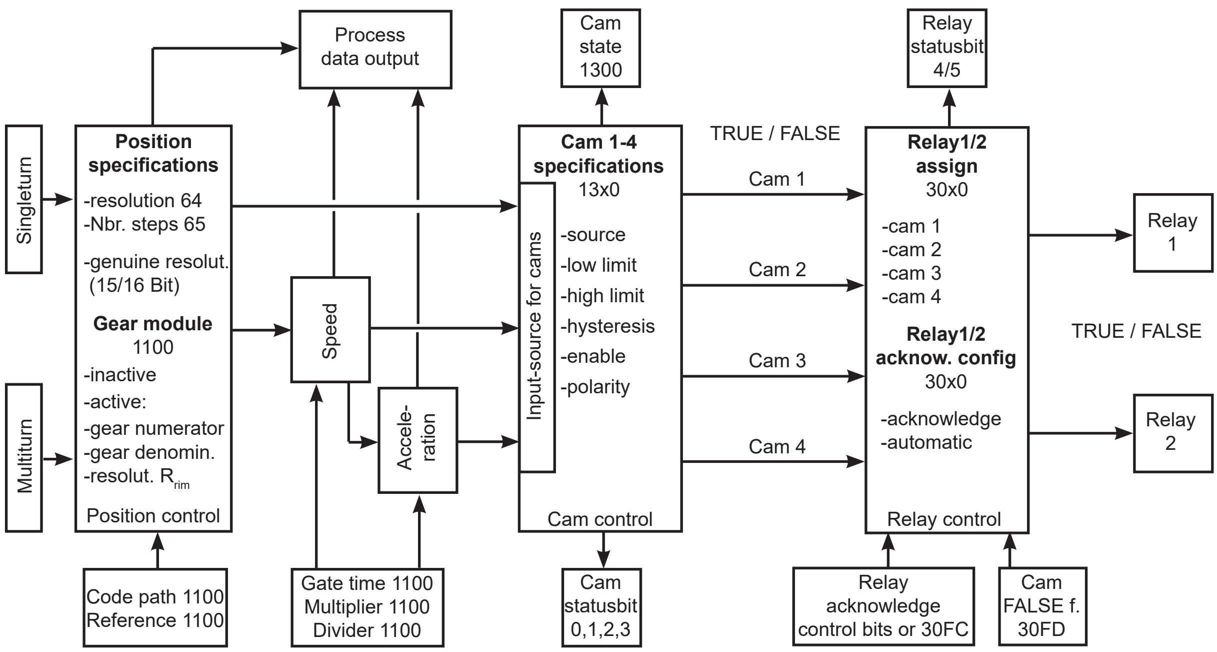
The state of each relay can be supervised by status bit 4 (relay 1) and status bit 5 (realy 2) in the status word. In addition the reset behaviour of a relay can be determined via object 30x0/02: *Automatic* or *Acknowledge*. At *automatic* the relay changes to the state TRUE as soon as no limit of the assigned cams is exceeded any more (related cam = TRUE. Be aware of cam polarity). At *acknowledge* the user has to send the *acknowledge command* (via 30FC or bit 0/1 in control byte) to set relays TRUE again. But the relay(s) change(s) to the state TRUE only when all related cams = TRUE.

If 30x0/02 is set to "1" (=acknowledge) after boot up of NOCIO the acknowledge command has to be sent to NOCIO once to set the relay to condition TRUE (NOCIO = "operational").

The relays can be tested at any time (and state of NOCIO) via object *Cam FALSE forcing* (30FD). The PLC or safety chain has to ignore the relay trip in case of test. This object has no influence on the relays directly but on the cams. To do the test, the relay(s) have to be TRUE – means the relays are assigned to one/more cams with are (all) TRUE and valid flags are set "A5". With 30FD the cam(s) are set to FALSE by force (input "1" for related cam(s)). Therefore the relay goes into condition FALSE and the user can see that the contact opens properly. After that the user inputs "0" for the related cam(s) (at any time later) for setting the cam(s) TRUE again. Relay closes again after acknowledge command (if ackn. = active) and test is done.

With the gear module (1100/09/A/B) the user has the possibility to calibrate the output signals (position/speed/acceleration) to the rim of a gearbox (rim / pinion) to which the shaft (pinion) of the NOCIO is mechanical connected to. The number of teeth of rim and pinion and the desired output resolution has to be inserted in the gear module. It can be activated / deactivated.

See handbook 15597 for the full list of parameters and the safety parameterization process (CRC checksums, etc.)



4 System description

Function

The function of the switching outputs is implemented by means of relays. Each relay (safety or non-safety) provides a normally open (NO) contact. It is open when NOCIO is not under power or in the error state (error detected due to internal diagnosis routines). The safety contact provides a "forcibly driven" safety relay. Its control contact is used for detection of the proper function of the main contact. The customer contact is internally build by two contacts (relays) connected in series due to one operational relay (the forcibly driven one) and one additional redundant relay (normal relay). So reliable contact separation is provided. The non-safety switching contact is designed with a standard relay with high reliability (but not PL d capable).

All switching outputs are galvanically separated in terms of operating voltage and IO-Link. The whole PCB electronic is separated to the housing due to the recommendation of IO-Link specifications.

The NOCIO's absolute encoder sends the shaft position data (position / speed / acceleration) to the main controller. The main controller informs the relay controller when to open / close which relay (see diagram above). The exact position of the switching flanks, i.e. calibration of the cams, can be carried out via parameterisation by IO-Link.

The relay can trigger on shaft position or shaft speed and shaft acceleration (parameter "cam source"). For every trigger source there is an individual set of related limit parameters (see table above). Each relay output can be assigned to one or more cams. Note, that the cam limits are both part of the TRUE state (see below)

If the relay contact has been triggered by position or speed / acceleration, it can be reset automatically when the related cam limit is not exceeded any more. Or it has to be reset by an IO-Link command. Parameters: *Relay acknowledge* (30FC) or appropriate bit in the control byte.

The proper operation of the relay contact can be tested within normal operation of NOCIO: parameter *Cam FALSE forcing* (30FD).

The position switching flanks of the switching output are set as follows in the factory as regards the angular position of the shaft: see cam diagram below.

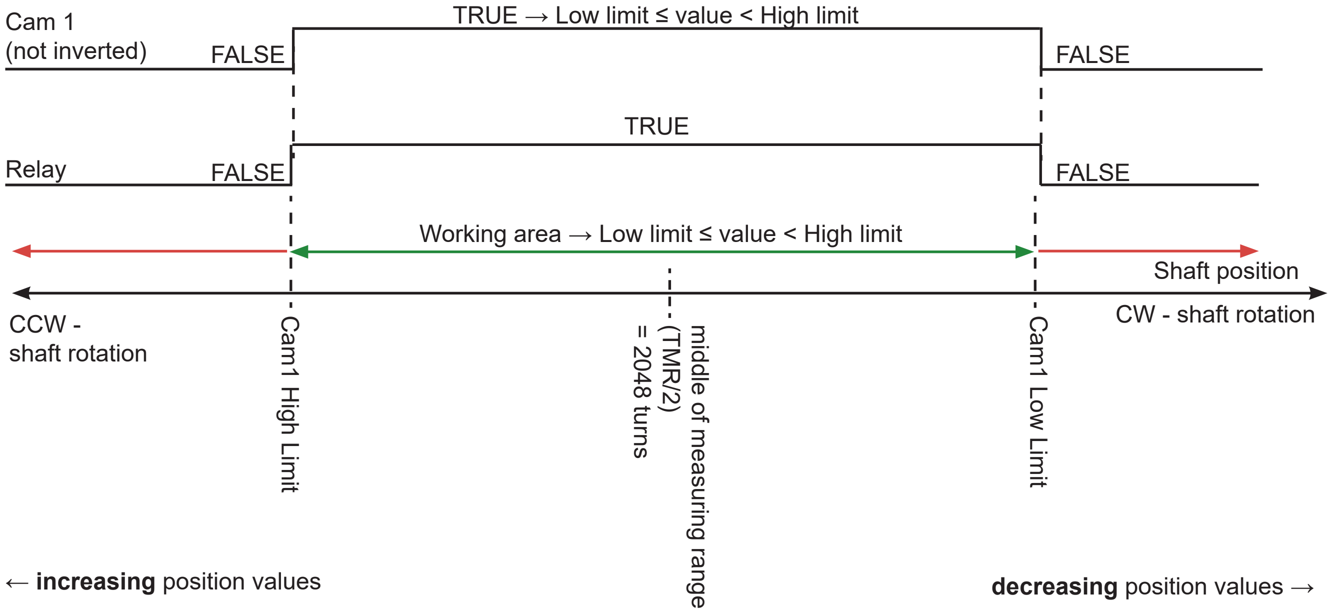
The switching output switching flanks refer to the IO-Link output signal of the absolute encoder. If the position signal's preset function is used (IO-Link output signal offset), the switching flanks are also accordingly shifted with reference to the shaft position.

To avoid undesired switching back and forth (flutter) on the part of the relays as a result of slight bit changes on the switching flank, a switching hysteresis of 10 digits is pre-programmed. This can be changed via IO-Link. Valid for function *Automatic*. Else the contact remains open until acknowledgement.

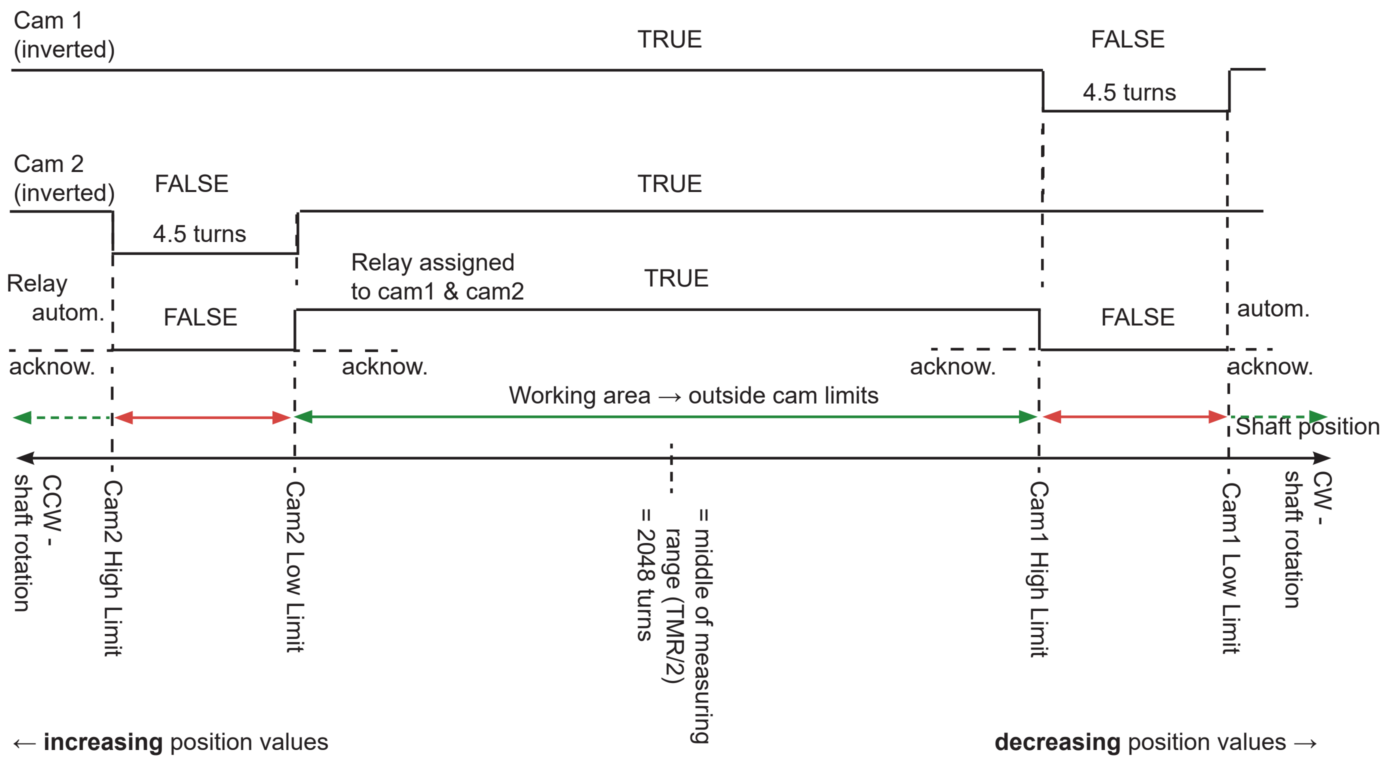
5 System description

4.2 Cam setting for position trip

Cam diagram (for position triggering)
First possibility for realising limit switch in CW and CCW direction by using **one** cam (here cam1).
In this case only statusbit 0 (cam1) is used/active. Working area is within low- and high-limit. Alarm area is outside limits



Cam diagram (for position triggering - linear presentation)
A possibility for realising limit switch with a detection possibility, whether the working are has been exceed in CW or CCW direction by using **two** cams (here cam 1 & 2) is shown below.
Note, that the circle can be extended above the maximum position value. When the cam high limit is smaller than the cam low limit, the cam is extended over the inverted pitch circle.
In this case the status bits 0 (cam1) and 1 (cam2) are used/active. Working area is outside low- and high-limits of cam1 and cam2. Therefore, cams have to be inverted. Relay = TRUE when cam1 & cam2 = TRUE.
4.5 turns for the cam length is exemplary. It can be any other length.
Cam 1 high limit has to be set to cam 2 low limit in order to close the circle (see below).



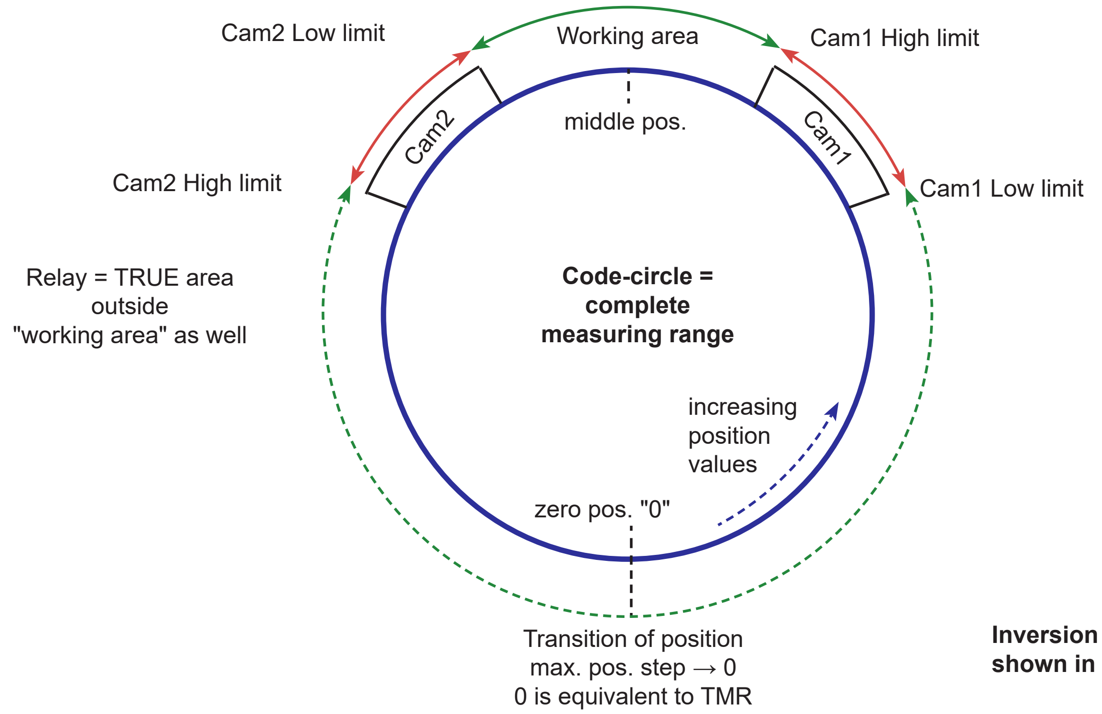
4 System description

Cam diagram (for position triggering - circular presentation)

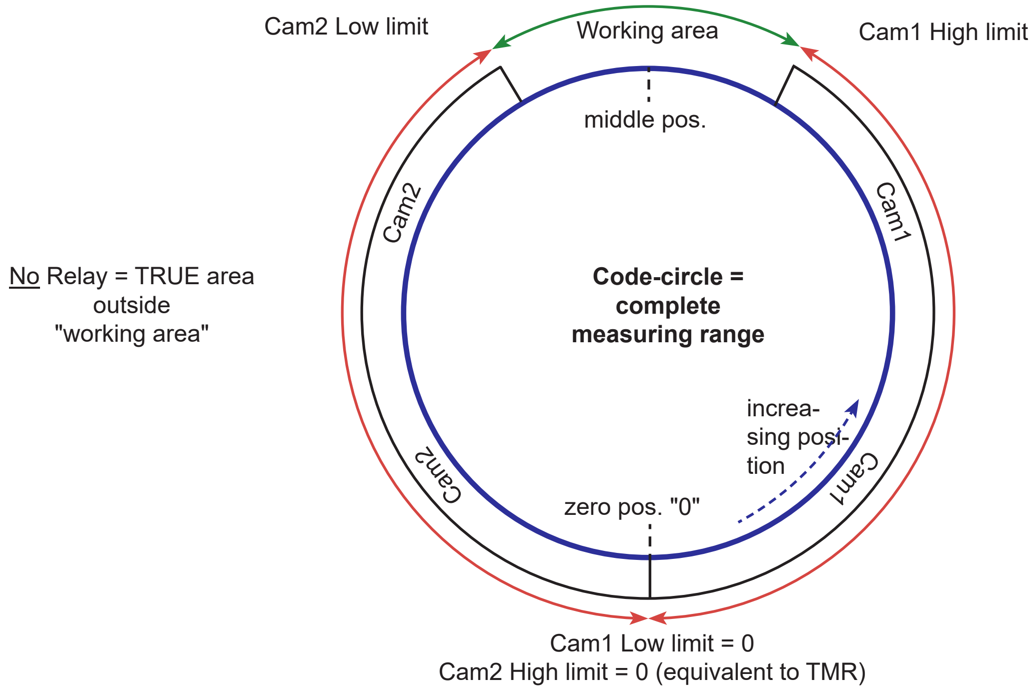
For better understanding in the following presentations the measuring range of position of a rotary encoder (here NOCIO) is displayed as a circle. Due to the fact that after the highest position value of an encoder follows "zero", the scale of position can be displayed like this.

The upper diagram shows 2 narrowed cams on the position scale like they are shown in the linear diagram one page before (inversion of cams not shown at circle diagrams). In this case a working area (green arrow) and a second "allowed" area (or working area) is present (green dashed arrow). In both areas is valid: status of cam1 = status of cam2 (e.g. = TRUE)

If this second area (green dashed arrow) is not required, the cams have to be enlarged at least to the "zero point" of the position scale (in the case that the working area is around the middle of the measuring range: Total measuring range (TMR) / 2. See diagram below. An overlap of the cams is allowed as well.



Inversion of cams not shown in these diagrams



4.3 Speed and acceleration calculation

NOCIO provides the possibility to use the switching output for shaft speed and/or shaft acceleration surveillance. A related limit can be defined for a cam triggering. This function allows to trigger the relay contact (it opens) when the limit is exceeded.
The shaft speed v_s is calculated using of the measured absolute position value. Within the speed gate time T_{gv} the difference between two position values is calculated. T_{gv} can be parameterized from 1 to 1000 milliseconds (ms). The result is the average speed v_s within this gate time period. The update time of the position value and speed value is 1 ms (position update rate = 1 kHz): Every 1 ms a new calculation is done with a new couple of position values which have a time difference of T_{gv}.
When a speed change at the encoder shaft occurs ($v_{s1} \rightarrow v_{s2}$), the speed output signal reacts after 1 ms and reaches the final value v_{s2} after T_{gv}.
The internal position value is the base of speed calculation (and therefore acceleration calculation as well). The position sensor of the encoder provides a resolution R_P of 15 bit/360°. The internal data format of position signal is always 16 bits (Remark: the resolution of the position output signal of NOCIO is independent of the res. of the internal position value). The content of this position signal depends on v_s.

- At v_s < ~1 rpm an averaging algorithm (over 32 position values) for position provides R_P=16 bits instead of 15 bits → speed v_s provides a resolution based on R_P=16 bits.
- At shaft speed v_s > ~1 rpm the averaging algorithm is deactivated → R_P=15 bits position resolution → the LSB of data format 16 bits is always 0 → resolution of v_s is reduced accordingly.

The resolution R_P of the position signal determines the resolution R_{vs} of the speed signal in conjunction with T_{gv}. Basis is 16 bits data format.

Example 1 (v_s < 1 rpm)**:**

Given: R_P = 16 bit/360 ° → **0.0055 °**
Required: R_{vs} = 12 bit/360 ° per s → **0.088 °/s**
→ $T_{gv} = R_P / R_{vs}$ = **0.062 s** (62 ms)
(T_{gv} = 62 ms is required to obtain R_{vs} = 0.088 °/s, means to see every R_{vs} step: 0.088 °/s, 0.176 °/s, ...)

Example 1a (v_s > 1 rpm)**:**

Given: R_P = 15 bit/360 ° → **0.011 °** (data format is 16 bits with LSB = constant 0)
Required: R_{vs} = 12 bit/360 ° per s → **0.088 °/s**
→ $T_{gv} = R_P / R_{vs}$ = **0.125 s** (125 ms)
(T_{gv} = 125 ms is required to obtain R_{vs} = 0.088 °/s, means to see every R_{vs} step: 0.088 °/s, 0.176 °/s, ...)

Example 2 (v_s > 1 rpm)**:**

Given: R_P = 15 bit/360 ° → **0.011 °** (data format is 16 bits with LSB = constant 0)
Required: R_{vs} = 15 bit/360 ° per s → **0.011 °/s**
→ $T_{gv} = R_P / R_{vs}$ = **1 s** (1000 ms)
(T_{gv} = 1 s is required to obtain R_{vs} = 0.011 °/s, means to see every R_{vs} step: 0.011 °/s, 0.022 °/s, ...)

Example 3 (v_s > 1 rpm)**:**

Given: R_P = 15 bit/360 ° → **0.011 °** (data format is 16 bits with LSB = constant 0)
Required: R_{vs} = **0.1 °/s**
→ $T_{gv} = R_P / R_{vs}$ = **0.11 s** (110 ms)
(T_{gv} = 110 ms is required to obtain R_{vs} = 0.1 °/s, means to see every R_{vs} step: 0.1 °/s, 0.2 °/s, ...)

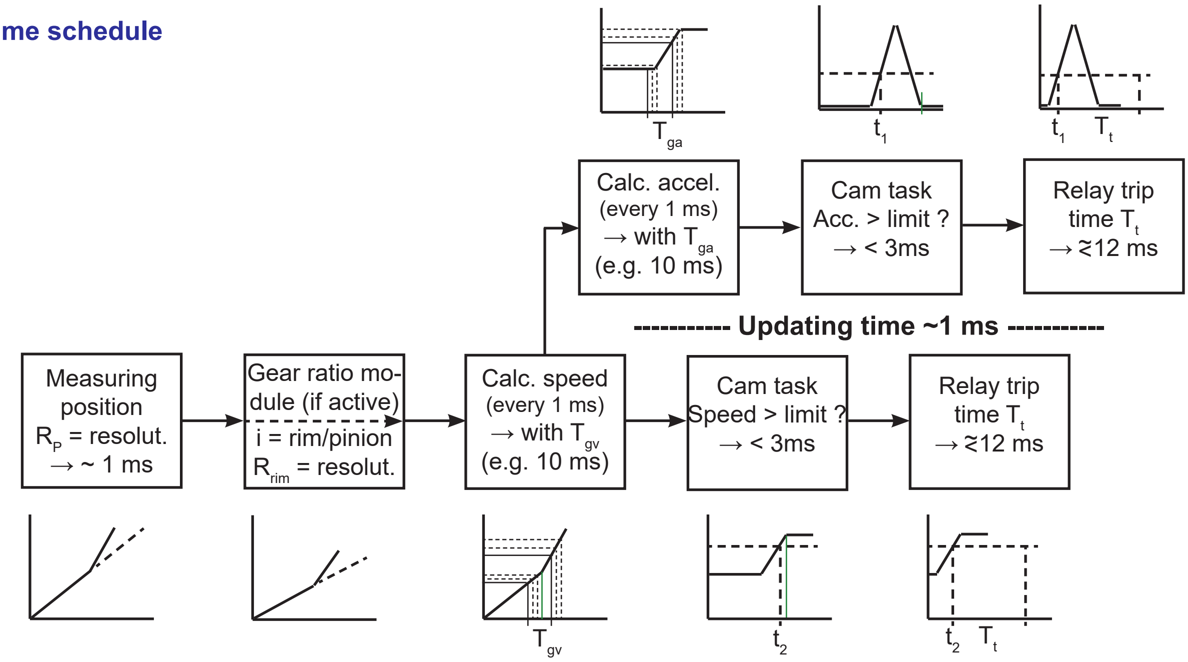
If a gear with a ratio of i (i = rim / pinion) is used in the application (e.g. i = 10) the speed of the encoder shaft (pinion gear) is i-times as high as the speed v_{ms} of the main shaft (gear rim). Therefore for the resolution of speed R_{vms} of the gear rim (= main shaft) is valid: $R_{vms} = R_{vs} / i$. Example #4: i = 10 → R_{vms} = 1 °/s / 10 = 0.1 °/s.

Acceleration is determined by calculation of the speed difference of two different speed values within an acceleration gate time T_{ga} like it is done at speed calculation by means of a position difference. The actualisation time is 1 ms as well. An acceleration limit can be defined for a cam triggering for acceleration surveillance. This function allows to trigger a relay contact (it opens) when the acceleration limit is exceeded. If the speed limit is not reached yet, the acceleration trigger can be used to stop the application due to a high acceleration value before (t_1) the speed limit is reached already (t_2).

4 System description

Block diagram for speed and acceleration calculation

Time schedule



Speed calibrating with "speed multiplier" and "speed divider"

The calibrating factors *speed multiplier* and *speed divider* are meant to set the speed signal of encoder shaft v_s to the required scaling.
Examples for different scalings $v_s x$ are:

$v_s 1$ = [steps/10 ms]
$v_s 2$ = [steps/100 ms]
$v_s 3$ = [steps/s]
$v_s 4$ = [0.1 °/s] $v_s 5$ = [1 °/s]
$v_s 6$ = [rad/s]
$v_s 7$ = [rps]
$v_s 8$ = [rpm]

Remark: The units (result of calibration) are integer. E.g. when choosing $v_s 8$ you will get **rpm** without any decimal places!

For scalings $v_s 1$ to $v_s 3$ the position difference in steps within the specified time (10ms, 100 ms and 1 s) is required. The relevant data format for the scaling is always 16 bits, independent of speed and associated actual resolution. Within the gate time T_{gv} (which needs to be chosen before calulating the scaling factors), the difference D_{Pos} between position steps P_1 and P_2 is calculated:

$D_{Pos} = P_2 - P_1$

To obtain the speed signal v_s this difference is divided by the gate time T_{gv} :

$v_s = (P_2 - P_1) / T_{gv}$ = steps / T_{gv}

Speed (acceleration) multiplier and divider can be used to adjust the output scaling for different time bases and position units (e. g. instead of "steps" ° or rotations per time unit can be chosen).

The tables on the next pages show some examples for scaling factors. In **Table 1** a gate time **$T_{gv} = T_{ac}$ = 125 ms** is chosen. With the given definitions *speed (acceleration) multiplier* and *divider* the above output scaling can be obtained (smallest possible integer values chosen by dividing by 10, 100, 1000 etc.).

In **Table 2** similar calculations have been carried out for gate times of **10 ms** and **100 ms**.

4 System description

Scaling unit S_U	Speed multiplier ≤ 65535	Speed divider ≤ 65535	Acceleration multiplier ≤ 65535	Acceleration divider ≤ 65535
[steps/10ms] *	**10** (= time base t_B of scaling unit [ms])	**125** → = T_{gv} [ms] x f_D	**10** (= time base t_B of scaling unit [ms])	**125** → = T_{ac} [ms]
[steps/100 ms] *	**100**	**125** → = T_{gv} [ms] x f_D	**100**	**125**
[steps/s] * [steps/1000 ms]	**1000**	**125** → = T_{gv} [ms] x f_D	**1000**	**125**
[0.1 °/s] [3600/1000 ms]	**1000**	**2275** → = T_{gv} [ms] x f_D	**1000**	**125**
[1 °/s] [360/1000 ms]	**100**	**2275** → = T_{gv} [ms] x f_D / 10	**1000**	**125**
[rad/s] [6.28../1000 ms]	**1**	**1304** → = T_{gv} x f_D / 1000	**1000**	**125**
[rps] [1/1000 ms]	**1**	**8192** → = T_{gv} x f_D / 1000	**1000**	**125**
[rpm] [1/60000 ms]	**60**	**8192** → = T_{gv} x f_D / 1000	**480** t_B / 125	**1** T_{ac} / 125

Table 1 (**T_{gv}** = **T_{ac}** = 125 ms), dimension factor **f_D** = **R_B / S_o**, where **R_B** is the internal resolution of the speed calculation (65536 steps/360°) and **S_o** the output (position) scaling.

Scaling (unit)	Speed multiplier ≤ 65535	Speed divider ≤ 65535	Acceleration multiplier ≤ 65535	Acceleration divider ≤ 65535
[steps/10ms] *	**1 / 1**	**1 / 10**	**1 / 1**	**1 / 10**
[steps/100 ms] *	**10 / 1**	**1 / 1**	**10 / 1**	**1 / 1**
[steps/s] *	**100 / 10**	**1 / 1**	**100 / 10**	**1 / 1**
[0.01 °/s]	**5000 / 500**	**91 / 91**	**100 / 10**	**1 / 1**
[0.1 °/s]	**500 / 50**	**91 / 91**	**100 / 10**	**1 / 1**
[1 °/s]	**50 / 5**	**91 / 91**	**100 / 10**	**1 / 1**
[rad/s]	**10 / 1**	**1043 / 1043**	**100 / 10**	**1 / 1**
[rps]	**25 / 5**	**16384 / 32768**	**100 / 10**	**1 / 1**
[rpm]	**375 / 75**	**4096 / 8192**	**6000 / 600**	**1 / 1**
[0.1 rpm]	**1875 / 375**	**2048 / 4096**	**6000 / 600**	**1 / 1**
[0.01 rpm]	**9375 / 1875**	**1024 / 2048**	**6000 / 600**	**1 / 1**

Table 2 (T_{gv} T_{av} = 10 ms / **100 ms,** R_B = 65536 steps/360°), smallest values (quotient) chosen already

Table 1 and Table 2: → every step (digit) in the speed output signal now represents the required unit.
Remark: The units (result of calibration) are integer. E.g. you will get **rpm** without positions after decimal point: 1, 2, 3, ... rpm. Use another rpm-unit from the table for a higher resolution. For example **0.1 rpm** or **0.01 rpm**. But see examples 1 to 4 on page 20 before for "real" resolutions due to gate time.

*: "steps" are meant here as steps based on the internal position resolution (16 Bits). It is not meant as steps based on the position output signal which may be scaled (e.g. reduced to 13 Bits). In that case the factor f_D has to be calculated. Example: Basis res.: R_B = R_P = 16 Bits. Output resolution R_O = 13 Bits. → f_D = 65536 / 8192 = 8 → Increase speed divider by factor 8.

4 System description

The scalings of the page before refer to the rotating speed v_s of the encoder shaft. When a gear is used with the encoder with a ratio of i = rim / pinion (usually number of teeth rim > number of teeth pinion → i > 1) the speed values can be scaled to the speed of main shaft v_{ms} by: $v_{ms} = v_s / i$.

Ratio "i" can be set by using parameters Gear ratio (numerator) = 1100/09 and Gear ratio (denominator) = 1100/0A. For example: rim = 153 and pinion = 20 → i = 153/20.
In this way the output signal of speed can be calibrated to the application and further requirements. The related speed limits for the relay triggering have to be adapted as well by customer because they refer to the scaled speed signal.

The acceleration a_s of the encoder shaft is calculated by using another gate time: acceleration gate time T_{ga}. Within this gate time the difference D_{speed} of the calculated and scaled speed is built:

$D_{speed} = v_{s2} - v_{s1}$

To obtain the acceleration a_s it is built:

$a_s = (v_{s2} - v_{s1}) / T_{ga}$

Example for the dimension (unit) of the output a_s (results) and therefore for the scaling of the acceleration limits:

With v_s 4 = [0.1 °/s] and T_{ga} = 100 ms → unit of a_s = [0.1 °/s / 100 ms]

With the calibrating factors *acceleration multiplier* (1100/07) and *acceleration divider* (1100/08) a calibration due to a required scaling can be done similar to the speed calibrations like mentioned above, especially to match the time unit/base (e.g.: [0.1 °/s/100ms] → [0.1 °/s²] → use acc. multiplier = 10, acc. divider = 1).

Calculating v_s in rpm with defined parameters

When *speed multiplier* sm and *speed divider* sd are defined, a certain position resolution R_P for the basis of speed calculation is given and a speed gate time T_{gv} is chosen, the shaft speed v_s - measured in rpm - can be calculated as:

$$v_s\,[\text{rev./min}] = \frac{n\,[\text{digits}] \times 60000 \times sd}{R_P\,[\text{digits/rev}] \times T_{gv}\,[\text{ms}] \times sm}$$

n is the number of digits of the speed output signal. Remark: n is in the format signed 16 Bits. At increasing position values (→ $v_s > 0$) you can use n directly. At decreasing position values (→ $v_s < 0$) you have to calculate: FFFF - n first, before inserting in the formula. Increasing or decreasing position values depend on the setting of safety code sequence CW/CCW. n_{max}: 15 Bits due to sign of velocity signal (16th bit).

4 System description

Example for speed and acceleration calculation

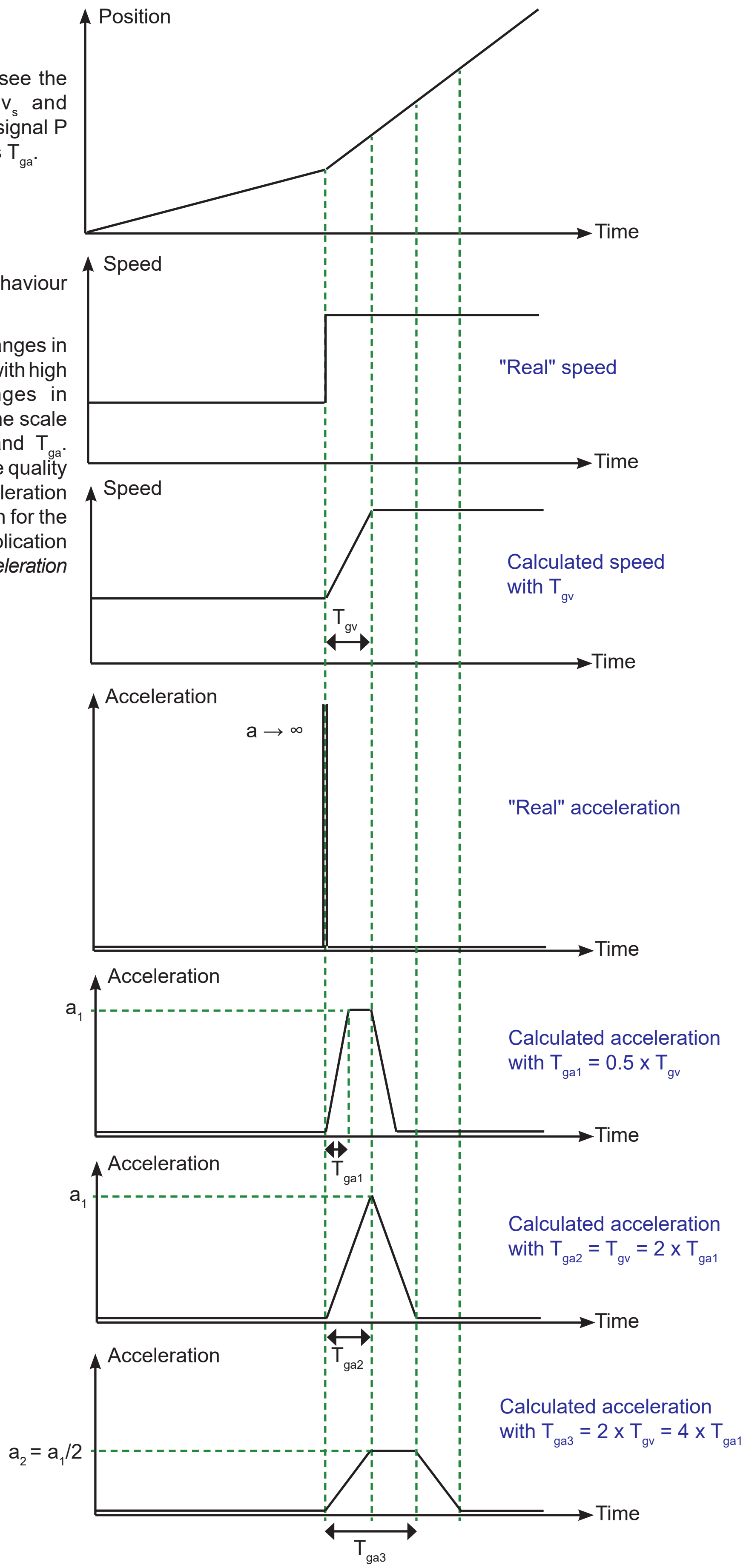
In the diagrams on the right side are to see the results of the calculation of speed v_s and acceleration a_s by means of the position signal P and gate time T_{gv} and different gate times T_{ga}.

$v_s = (P_2 - P_1) / T_{gv} \rightarrow$ steps / T_{gv}

$a_s = (v_{s2} - v_{s1}) / T_{ga} \rightarrow$ steps / T_{ga}

Taken is a *jump function* to show the behaviour of the calculation processes.

Real applications do not have sudden changes in the behaviour of movement. Applications with high inertia and a high mass have changes in characteristics of movement on a long time scale in comparison to the gate times T_{gv} and T_{ga}. Therefore the reaction time (in ms) and the quality of the output signal of speed v_s and acceleration a_s is sufficient, i.e. fast and precise enough for the appraisal of the moving behaviour of an application (see as well *Block diagram for speed and acceleration calculation*).

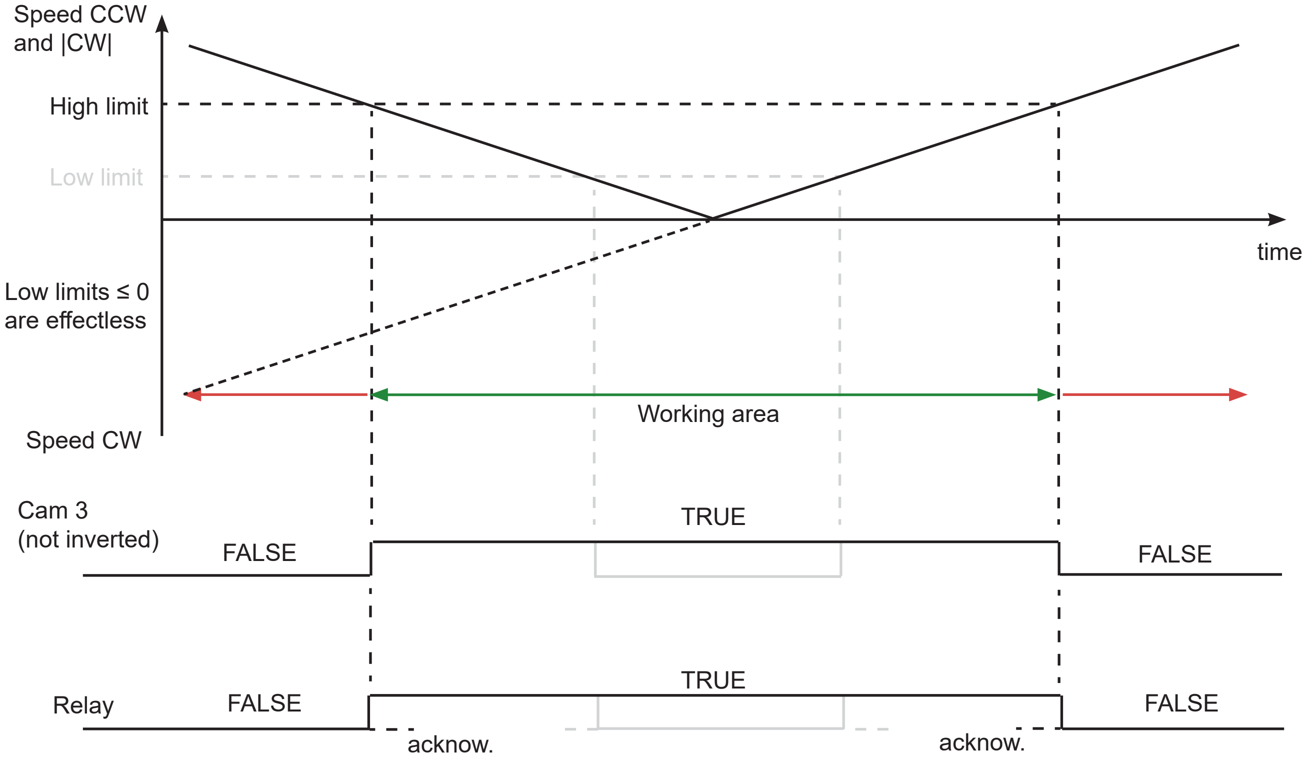


4 System description

4.4 Cam setting for speed and acceleration trip

Cam diagram (for speed triggering, code path CCW)

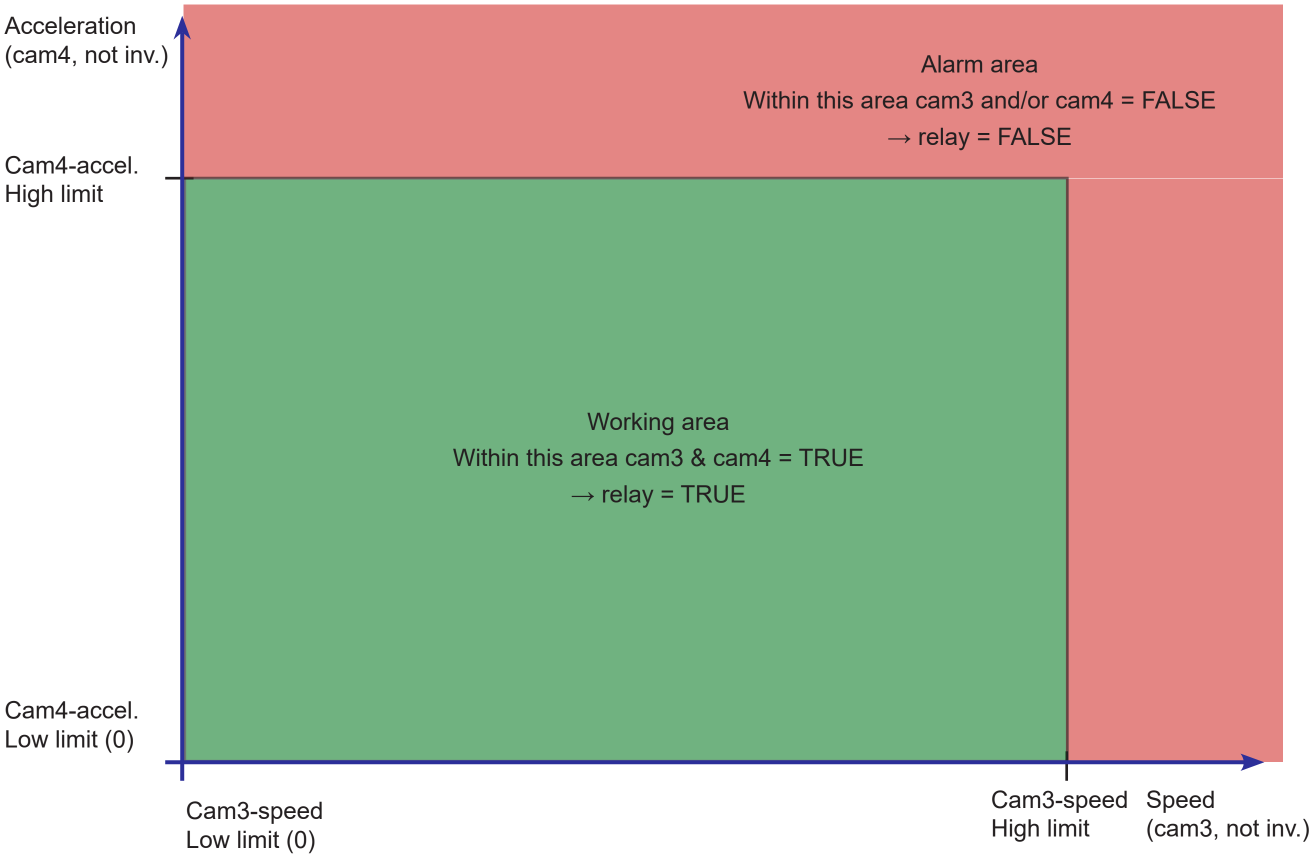
Absolute speed value for source of cam3 (*Cam3 source* (1330/01) = 3). Direction of shaft turning is not distinguished.
In this case the status bits 2 (Cam3) is used/active. Working area is within low- and high-limit. Alarm area is outside limits.



4 System description

The settings for acceleration trip are the same as for speed trip. Working area is within low- and high-limit of a cam. Alarm area is outside limits (the other way around is possible as well like position trip).
When speed and acceleration trip is required simultaneously, a diagram can look like shown below. Speed trip and acceleration trip are independent from each other.

Cam diagram (2-dimensional. Cam source setting: absolute values of speed and acceleration)



4.5 Slewing ring functionality

In some applications it is necessary to get directly the angle of the slewing unit (gear rim).
A rotary encoder is coupled via a gear to a toothed gear (pinion) or with a worm gear (rim) / gear units mounted on the worm shaft.

This results in a certain ratio due to the number of teeth of rim and pinion.

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. 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).

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.

Use the parameters 1100/09 (gear numerator) and 1100/0A (gear denominator) for the input of the gear parameters and for activation of the slewing ring functionality. If "zero" is input at one or both of these parameters, the slewing ring functionality is deactivated and NOCIO output of position is standard.

Use object 1100/0B for selection of the required resolution for the slewing ring functionality (e.g. 3,600 steps/360° or 36,000 steps / 360°).

The activation of slewing ring functionality with its resolution has impact on position and therefore on calculation of speed and acceleration. Therefore it has impact on the cam and relay behaviour as well because the source of a cam refers to the values calculated due to slewing ring functionality (See block diagram on page 21).

When slew ring functionality is activated, *position preset* (control byte bit 3) only works when *Reference value* is set between *0* and *Resolution gear rim* (R_{rim}) - 1. For *Reference value* ≥ R_{rim} executing *position preset* has no effect.

Adjustable parameters	from	to
Number of teeth - slewing ring 1100/09	1 (0)	65535
Number of teeth - pinion of NOCIO 1100/0A	1 (0)	65535
Resolution position R_{rim} 1100/0B	1	65535 (R_{rim} ≤ encoder resolution x i)

i = Gear ratio *Number of teeth - slewing ring* to *Number of teeth - pinion of NOCIO*

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

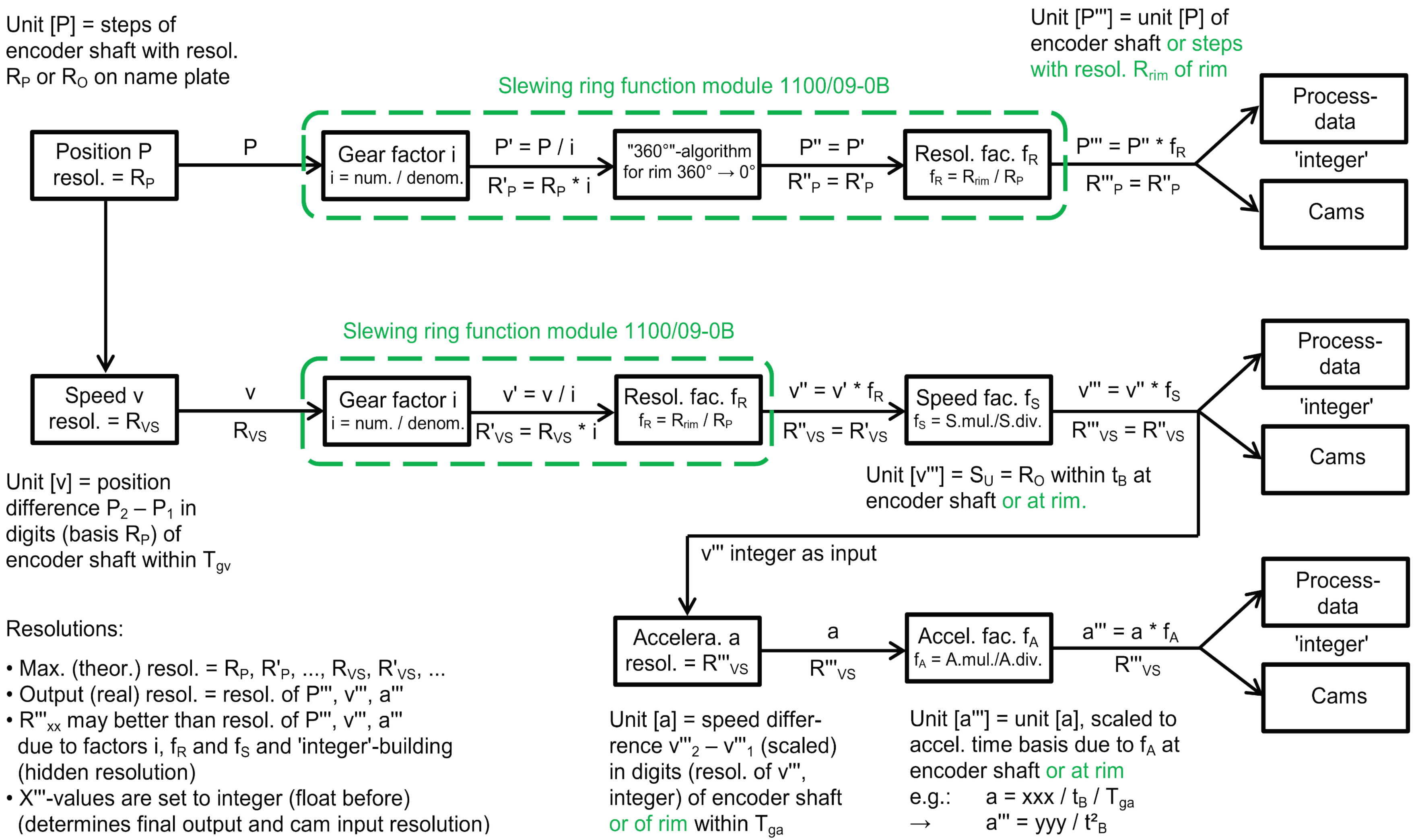
Characteristic	Encoder	Slewing ring
Resolution	8192 steps	Adjustable, max. 8192 x i steps
Accuracy	± 0.2 %	± 0.2 % / i
Measuring range	4096 revolutions	Revolution repeatable ∞ times
Reproducibility	± 0.05 %	± 0.05 % / i
Temperature drift	< 0.02°	< 0.02° / i

All of the values specified in this table (except the measuring range) refer to one revolution of NOCIO shaft or one revolution of slewing ring resp.

4 System description

4.6 Block diagram for scaling factors

Influence of several scaling factors on position / speed / acceleration signal and their resolutions in process data and in cam handling



Green: property valid when slewing ring function module is active (→ gear num. & gear denom. ≠ 0)

For better understanding it is written: P, v, a, R_{VS}, …. and not v_s (shaft), v_{ms} (main shaft = rim), R_{VMS} (main shaft = rim) etc. It's used here: X', X'', X''', ..

5 Process data exchange

5.1 Overview

Input data: Encoder → Master

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8	Byte 9	Byte 10	Byte 11	Byte 12
Position				Speed		Acceleration		Temperature		Status	

Output data: Master → Encoder

Byte 1
Control

5.2 Input data

5.2.1 Position data format

Byte 1								Byte 2								Byte 3								Byte 4							
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
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
0	0	0	0	0	Position value *																										

* At 15 bit resolution. Number of bits used varies with resolution.

The position value is output as a 32-bit unsigned integer value in Motorola format (Big Endian). The absolute encoder‘s counting direction and the preset value can be changed via the Parameter 1100. (see chapter 8).

5.2.2 Speed data format

The speed value is determined via the cyclically read-in of the position data. The measurement unit is steps per gating time. The gating time (time interval for determining the change of position) is adjustable in the range of 1 - 1000 ms (see see chapter 8). The default value is 10 ms.

Byte 5								Byte 6							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
16 bit speed value															

The speed value is output as a 16-bit signed integer value in Motorola format (Big Endian). The following applies to the prefix:

positive for increasing position
negative for decreasing position

The refresh rate of the speed signal is independent from the selected gating time always 1 ms. The speed measurement resolution is independent of the resolution of the encoder. It is always based on the maximum resolution.

5.2.3 Acceleration data format

The acceleration value is the speed difference within the acceleration gate time. The measurement unit is steps per gating time. The gating time (time interval for determining the change of position) is adjustable in the range of 1 - 1000 ms (see chapter 8). The default value is 125 ms.

Byte 7								Byte 8							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
16 bit acceleration value															

The acceleration value is output as a 16-bit signed integer value in Motorola format (Big Endian). The following applies to the prefix:

positive for -increasing speed at increasing position values
-decreasing speed at decreasing position values

negative for -decreasing speed at increasing position values
-increasing speed at decreasing position values

The refresh rate of the acceleration signal is independent from the selected gating time always 1 ms.

5.2.4 Temperature data format

Byte 9								Byte 10							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
16 bit value															

This is the temperature value measured inside the main controller. The measurement unit is °C with a resolution of 0.1 °C. It is output as a 16-bit signed integer value in Motorola format (Big Endian).

5.2.5 Status data format

Byte 11								Byte 12							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
16 bit status															

5 Process data exchange

	Bit	Description	IO-Link def.	
Byte 1	0	Cam1 limit status. Bit = 1 (TRUE): Low limit ≤ value < high limit. Bit = 0 else. (If cam is not inverted)	15	Octet 0
	1	Cam2 limit status. Bit = 1 (TRUE): Low limit ≤ value < high limit. Bit = 0 else. (If cam is not inverted)	14	
	2	Cam3 limit status. Bit = 1 (TRUE): Low limit ≤ value < high limit. Bit = 0 else. (If cam is not inverted)	13	
	3	Cam4 limit status. Bit = 1 (TRUE): Low limit ≤ value < high limit. Bit = 0 else. (If cam is not inverted)	12	
	4	Safety relay (1) status. Bit = 1 (TRUE): Relay = TRUE (=contact closed) All cams with "AND" combination: cam1 & cam2 & ... = TRUE → Relay = TRUE	11	
	5	Relay (2) status. Bit = 1 (TRUE): Relay = TRUE (=contact closed) All cams with "AND" combination: cam1 & cam2 & ... = TRUE → Relay = TRUE	10	
	6 & 7	reserved	9-8	
Byte 2	8	Encoder fault	7	Octet 1
	9	Bit = TRUE: Valid flag 11FE is set to A5 (standard parameter configuration is valid)	6	
	10	Bit = TRUE: Cam valid flag 30FE is set to A5 (cam parameter configuration is valid)	5	
	11 - 15	reserved	4-0	

5.3 Output data

5.3.1 Control byte data format

Byte 1							
7	6	5	4	3	2	1	0
7	6	5	4	3	2	1	0
8 bit control							

Bit	Description
0	Acknowledge limit trip "Safety relay" (1). Relay contact closes (TRUE), if all assigned cams are TRUE. Command will be accepted only at shaft speed v_s = ~0. If both conditions fulfilled, this command is executed immediately on rising edge.
1	Acknowledge limit trip second (safety or non-safety) relay (2). Relay contact closes (TRUE), if all assigned cams are TRUE. Command will be accepted only at shaft speed v_s = ~0. If both conditions fulfilled, this command is executed immediately on rising edge.
2	Application reset. All parameters are restored to factory default values (except I&M data). No IO-Link communication reset. Command will be accepted only at shaft speed v_s = ~0 and 0x11FE = 0x30FE = 0. If conditions are fulfilled, this command is executed immediately on rising edge.
3	Position preset. Sets position value to "reference value" (parameter 0x1100/02) Command will be accepted only if shaft speed v_s = ~0, 0x11FE = 0xA5 and 30FE = 0. If all condition are fulfilled, this command is executed immediately on rising edge.
4 - 7	reserved

5.3.2 Position preset

To adapt the position of the encoder to the machine position, the encoder can be preset to any value within the measuring range. The preset value can be specified via the *Reference position* object 0x1100/2 (see chapter 8). The preset is executed by setting bit 3 in the control byte or by using the button *Position preset* (write 0xA1 to object 0x02/0). The preset value can only be set when the encoder shaft is stationary, the safety configuration valid flag 0x11FE is set to 0xA5 and the cam valid flag 0x30FE is set to 0.

The preset value is taken over with the rising edge of bit 3 of the control byte. An offset value is calculated (from the current actual position and the preset value) by the encoder and stored in the encoders memory, where it is protected against zero voltage.

The internally saved position preset offset value is reset during a factory or application reset.

A successful preset is acknowledged by a green flashing of the status LED for 2s, a rejected preset (e.g. due to the wrong safety flag configuration) is indicated by a red flashing.

Remark: Executing the position preset uses the function *Save all parameters* (Button 0xA0) to save the new offset value. Therefore, all other parameters that have been changed are saved as well!

5.3.3 Reset options

There are three different reset types implemented, which are defined in the IO-Link standard and which can be accessed via IO-Link system buttons. Resets are only accepted if shaft speed v_s = ~0 and 11FE = 30FE = 0. If condition is fulfilled, command is executed immediately on rising edge.

Button ID	Name	Description
128	Device reset	Main controller restart. Equivalent to power cycle (off - on)
129	Application reset	All parameters are restored to factory default values (except I&M data). No IO-Link communication reset. Can also be executed by control byte bit 2 (see above).
130	Restore factory settings	All parameters are restored to factory default values, including I&M data. No IO-Link communication reset.

5.3.4 Firmware update

Firmware updates via the IO-Link Control tool require a firmware password. The password is **0x54574b10**. After a firmware update an application or factory reset must be carried out.

Device-specific parameters are not affected by the update.

6 Setting-up operation

This documentation is based on the Lumberg LioN-P PROFINET IO-Link Master, with IO-Link Control Tool V5.1 and TIA Portal V16.

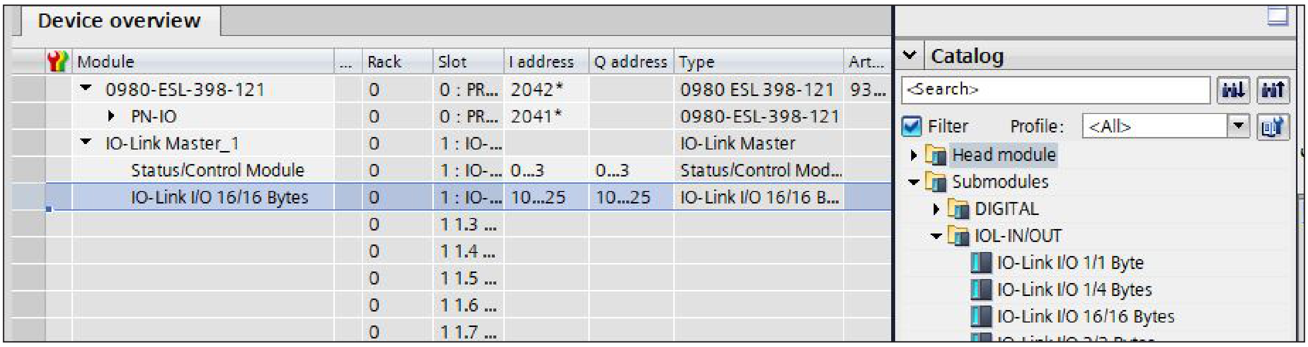
6.1 Prerequisites

The PLC project is set up and the master is installed via its GSD file in the Profinet of the PLC.

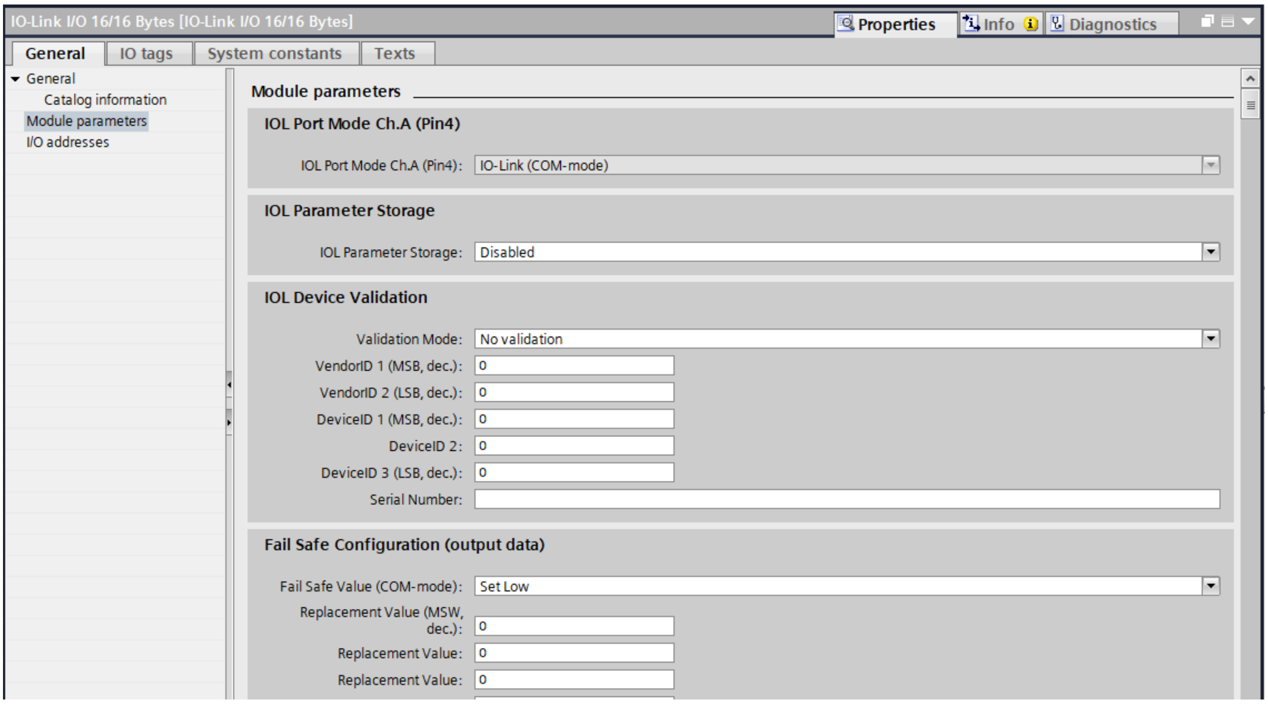
6.2 Inserting the NOCIO

Open the master module in the device view and open the tree of the available submoduls in the hardware catalogue.

The NOCIO has 9 byte input and 1 byte output process data. Choose the submodule with the next higher number of I/O bytes under IOL-IN/OUT and move it on the desired subslot of the master module (here subslot 2).



Now set the I/O addresses which you want to use in your PLC program (please refer to chapter 6 for the data format). With a right click on the module/properties you can open the port configuration. Here you can change the general IO-Link settings for this port.

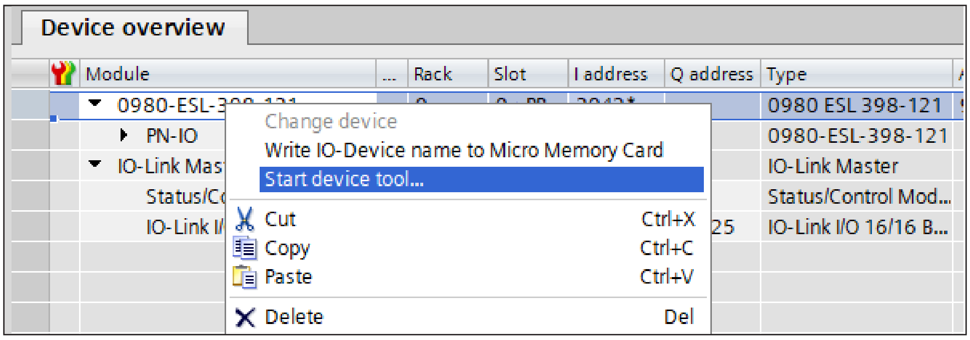


If you want to proceed with the default encoder or cam switch parameter you can now compile your configuration and download it to your PLC.

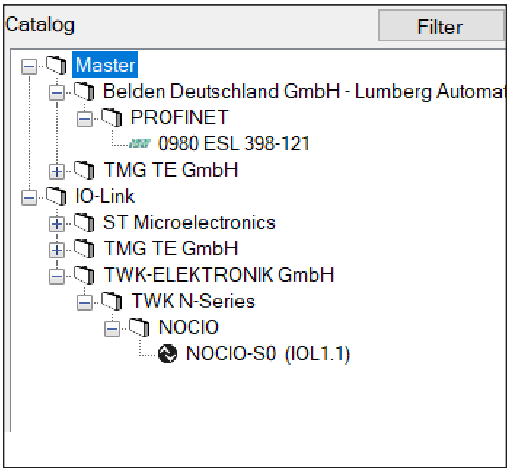
6 Setting-up operation

6.3 Changing parameters

To setup the encoder and cam switch parameter you have to use a IO-Link Device Tool. This tool allows you to install the IO-Link Device Description file (IODD) for every device connected to the master and to setup the device specific parameter. The IO-Link Device Tool from TMG can be started as stand alone program or via the TIA Portal by right clicking on the master module:

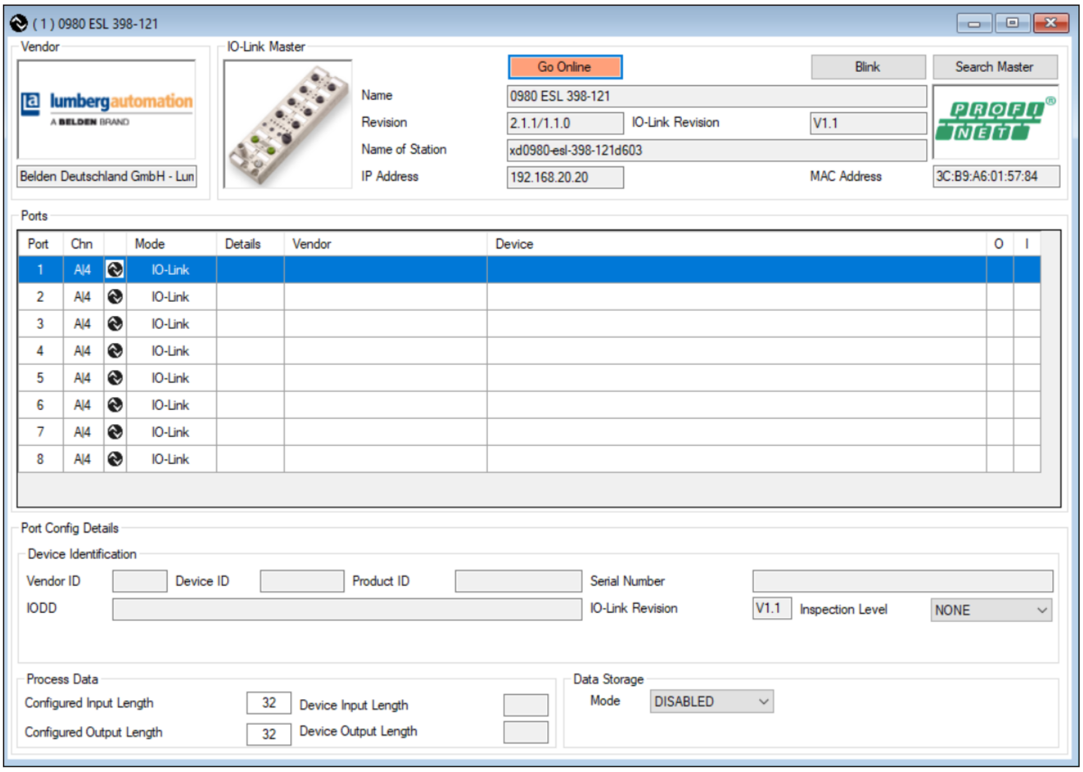


In the device tool you have to import the master description (available via the button "TMG web") and the IODD of the NOCIO. Afterwards both device are listed in the hardware catalogue.

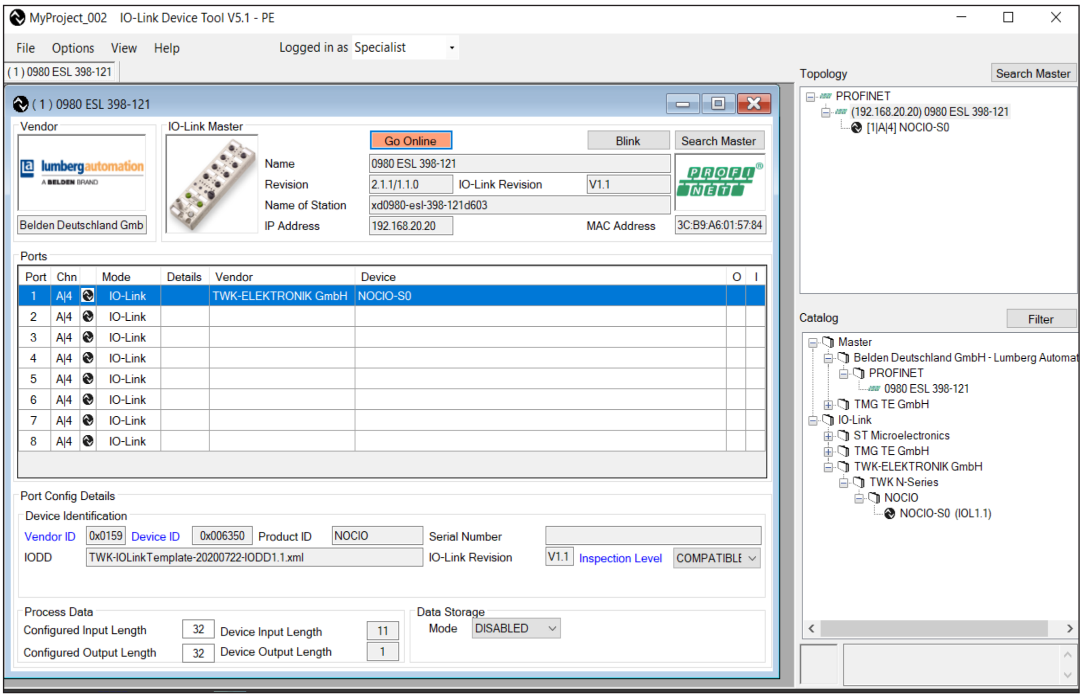


6 Setting-up operation

If you are connected to the Profinet you can now search the master. When the master is found, the masters configuration window will appear:

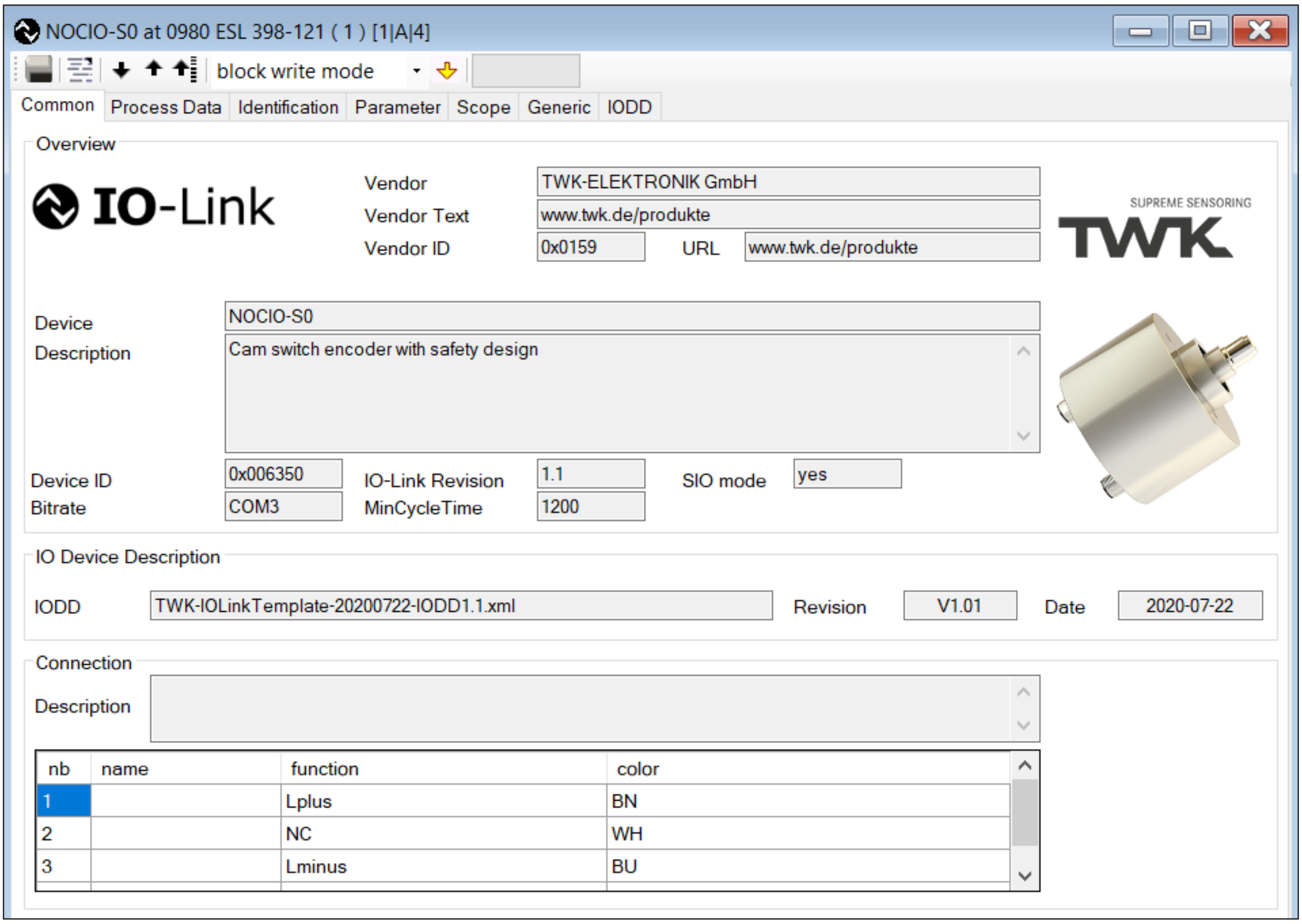


Now every device which is in the hardware catalogue can be inserted via drag and drop into the ports



6 Setting-up operation

Via a double click on the module you have access to the NOCIO information and parameter:



After changing one of the parameter you have to download the new configuration via the download button on the top. Please don't forget to set the parameter storage behaviour in your port configuration (see chapter 7.2)

7 Encoder and cam parameters

This chapter describes how the parametes of the NOCIO can be adjusted.

Before changing parameters, ensure that the overall system is in a safe state and deactivation and adjustment of NOCIO functions can be done without danger for personnel and equipment .

The related valid flag (0x11FE and/or 0x30FE) has to be set "zero" (instead of "0xA5"). Then NOCIO will accept new parameters. See table with available parameters and/or table with reduced parameters available. The valid flags can be set to zero (and to 0xA5 again later) only when the NOCIO shaft is not moving (v~0 → v approx. zero). Otherwise the valid flags remain "0xA5" and NOCIO goes on working in a normal way. Status bits 9 and 10 indicate the status of the valid flags. When the valid flags are set to "0", NOCIO is not in the normal operational status (OP) but in the status of parameterisation (PA). The the process data output depends on which valid flag is deactivated (see figure on next page).
Since the CAM operation depends on the settings of the standard parameter, the CAM valid flag 0x30FE has to be set to 0 first when going into the status of parameterisation and hast to be set to “0xA5” last when going back to normal operation.

The related encoder parameters of 0x11FE determine the calculation and output of position / speed / acceleration and therefore the behaviour of the cams and relays. 0x30FE can be set "0" without setting 0x11FE = "0". The encoder data (position / speed / acceleration) are still valid in this case. The switching outputs can be parameterized (cams and relays). After parametrization, both valid flags have to be set "A5" for normal operation of NOCIO with all functions (set 0x11FE first if it is "0"). If no cam functions are needed, 0x30FE can remain "0".
When setting valid flags to "0xA5", NOCIO checks all related parameters if they match to the CRC (described below). If they do, NOCIO accepts setting valid flag(s) "A5" (see status bits: = TRUE). If not, related valid flag (and status bit) remain "0" and a warning (IO-Link event) is transmitted to the IO-Link master.

For each new parameter the related CRC checksum has to be calculated (objects 0x..FF). The parameters are separated in groups (encoder: 0x1100/subs, CAM1: 0x1310/subs, CAM2: 0x1320/subs, CAM3: 0x1330/subs, CAM4: 0x1340/subs, RELAYx: 0x30x0/subs). For each group an own CRC is defined (see table of parameters on the next pages) and has to be calculated (e.g. by TWK program or with customer algorithm) and transmitted to objects 0x11FF and 0x30FF/subs.

When valid flags are successfully set "A5" all related new parameters become active and NOCIO works with them. They have to be saved by IO-Link command when customer wants to have them active as well after power OFF/ON or a reset of NOCIO.

- If a parameter has to be modified, set the related valid flag = "0" (see below in the table)
- Transmit parameter(s) to NOCIO
- Calculate related CRC (see below in the table). TWK program can be used
- Transmit CRC to NOCIO
- Set valid flag(s) = "A5"
- If the valid flag(s) cannot be set "A5" check consistency of paramters and CRCs and try again.
- Valid flags only can set "0" or "A5" when encoder shaft has speed v~0 (v approx. zero).

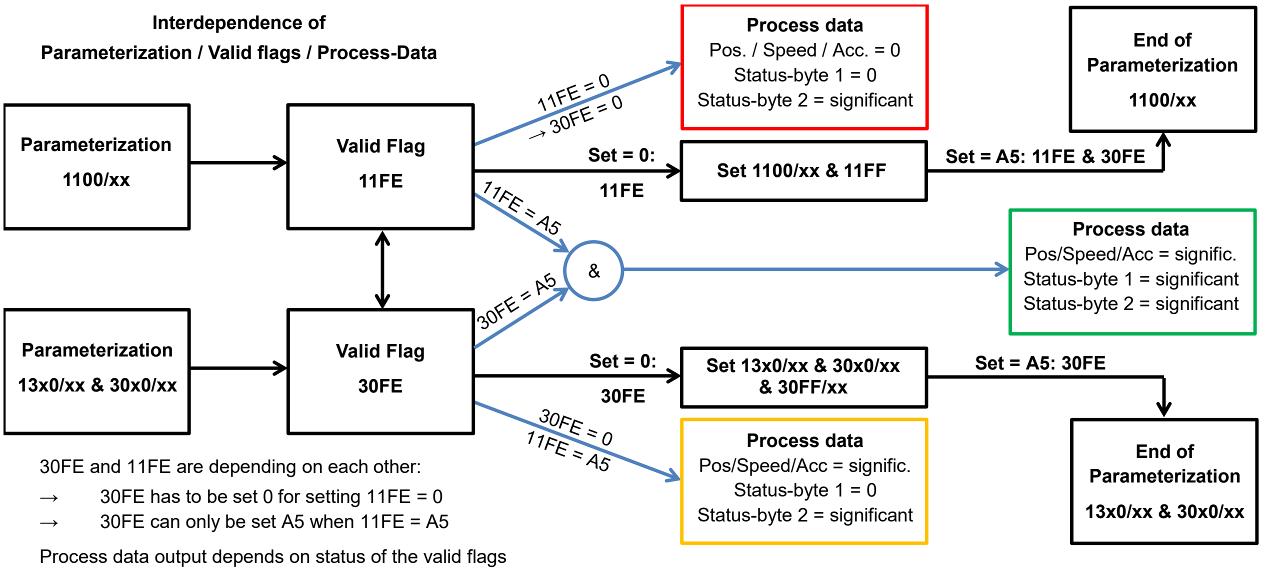
Difference between parameters 0x1100/XX (with valid flag 0x11FE) and 0x13X0/XX & 0x30X0/XX (with valid flag 0x30FE):

0x1100/XX (0x11FE): These parameters are relevant for the process data: Position / Speed / Acceleration. If the valid flag 0x11FE is set to "0" these process data will output "zero". The status byte 1 is set to zero. Only status byte 2 remains significant. As soon as valid flag 0x11FE is set to "0xA5" (with new or old parameters) the process data are not set to "zero" any more. When setting 0x11FE = "0" the valid flag 0x30FE has to be set = 0 first.

0x13X0/XX & 0x30X0/XX (0x30FE): These parameters don't have an impact on Position / Speed / Acceleration. If 0x30FE is set to "0" these process data will proceed and 0x11FE can remain "A5". The status byte 1 is set to zero → all status bits (0 to 5) are set to "0". Status byte 2 remains significant. But the cams are deactivated (FALSE) and the relays will open (FALSE). Cams and relays will be ready for operation as soon as 0x30FE is set to "0xA5" again. All relevant status bits are set due to new parameters.

Remark: In case of a encoder fault the process data *Position / Speed / Acceleration* and status byte 1 are set to "zero".

7 Encoder and cam parameter



IO-Link Interface - Parameter listing

Parameter *(object/sub) - access*	**Value range** *(data length: octet / default)*	**Parameter description**
Resolution [steps/360°] *(0x64)* - ro	1...65,536 steps *(4 / 32,768)*	Resolution per revolution (360°) [steps] Example: 15 Bits = 32768 steps. 3600 for 0.1° / 360°.
Total number of steps [steps] *(0x65)* - ro	0 ... 268,435,456 (only 2^n values) *(4 / 134,217,728)*	Total measuring range (= TMR) [steps] Example: 27 Bit = 134,217,728 = 32,768 [res] x 4096 [turns]
Device error history *(0x1003)* - rw	Per error: 0x00 - 0xFFFF FFFF *(4 / 0)*	History of up to 20 device errors. Manual reset required when full. See page 33 for details.
Code path *(0x1100/01)* - rw	0 ... 1 *(1 / 1)*	CW (clockwise): ascending values on rotation clockwise CCW (counter clockwise): descending values on rotation clockwise (viewed looking at the shaft). 0=CW / 1=CCW
Reference value *(0x1100/02)* - rw	0 ... total number of steps -1 *(4 / 67,108,864 = TMR/2)*	For adaptation to the application, the position value can be set to any value within the measuring range. By setting this parameter the cams shift as well at "position trip"-function
Speed gate time (T_{gv}) *(0x1100/03)* - rw	1 ... 1000 [ms] *(2 / 125)*	Time basis for speed registration (independent of the gate time, the position and speed and acceleration value is refreshed every 1 ms). This parameter determines the resolution of the speed signal. See below in this document.

7 Encoder and cam parameter

Parameter *(object/sub) - access*	**Value range** *(data length / default)*	**Parameter description**
Speed multiplier (sm) *(0x1100/04)* - rw	1 ... 65,535 *(2 / 100)*	Provides adjustment / calibration of speed value output (e.g. [steps/10 ms], [0,1 °/s], [1 °/s], [rad/s]) disregarding the resolution (see tables/remarks below and chapter "Speed / acceleration calculation"). Definition: f_S = speed multiplier / speed divider It determines the scaling of the input values for the speed low and high limits. The scaling is determined as well by "Gear ratio (numerator)" and "Gear ratio (denominator)". Remark: The maximum resolution of the speed and acceleration signal is determined by "speed gate time". (Default 100/2275: to obtain [1 °/s] at speed gate time = 125 ms)
Speed divider (sd) *(0x1100/05)* - rw	1 ... 65,535 *(2 / 2275)*	
Acceleration gate time (T_{ga}) *(0x1100/06)* - rw	1 ... 1000 [ms] *(2 / 125)*	Time basis for acceleration registration (independent of the gate time, the position, speed and acc. value is refreshed every 1 ms). This parameter determines the resolution of the acceleration signal together with "speed gate time" (see below).
Acceleration multiplier (am) *(0x1100/07)* - rw	1 ... 65,535 *(2 / 8)*	Provides adjustment / calibration of acceleration value output especially to provide the same base of time in the denominator: x / time². Else the acceleration time base will always be the *acceleration gate time* T_{ga}. It determines the scaling of the input values for the acceleration low and high limits. The scaling is determined as well by "Gear ratio (numerator)" and "Gear ratio (denominator)" and "speed multiplier" and "speed divider". Definition: f_A = acc. multiplier / acc. divider Remark: The maximum resolution of the acceleration signal is determined by "speed gate time" and "acc. gate time". (Default 8/1: to obtain [1 °/s²] at acc. gate time = 125 ms, because "Speed" is set to 1 °/s)
Acceleration divider (ad) *(0x1100/08)* - rw	1 ... 65,535 *(2 / 1)*	
Gear ratio numerator (number of teeth: rim) *(0x1100/09) - ro* (rw for non-safety devices)	0 ... 65,535 *(2 / 0)*	**"Slewing ring functionality" at input ≠ 0.** Provides NOCIO adaptations when a gear is coupled to the NOCIO. The output of position and speed/acceleration values can be calibrated to the gear rim movement instead of encoder shaft movement (all limit values refer to this calibration as well !). Gear ratio (i) = numerator / denominator = rim / pinion ≥ 1. Example: i = 153 / 20. Position output represents 360° of gear rim with resolution R_{rim} which can be defined by customer (e.g. 0 to 36,000 digits at R_{rim} = 36,000). After every full turn of gear rim (360°) the output signal of NOCIO goes to 0 again. Remark: The gear ratio has influence on the speed and acceleration signal. It is calibrated to the rim movement. Input values of speed multiplier and speed divider determine the speed and acc. output and their limits as well. Input values of acceleration multiplier and acceleration divider determine the acceleration output and its limits as well. At gear ratio numerator or denominator = 0, slewing ring functionality is deactivated → total measuring range is output.
Gear ratio denominator (number of teeth: pinion) *(0x1100/0A) - ro* (rw for non-safety devices)	0 ... 65,535 *(2 / 0)*	
Resolution gear rim (R_{rim}) *(0x1100/0B) - ro* (rw for non-safety devices)	1 ... 65,535 (R_{rim} ≤ encoder resolution × i) *(2 / 36,000)*	Resolution required for rim position output. This value determines the resolution of 1 turn of the rim. Only valid when gear ratio input is ≠ 0 (resolution of encoder (see above) is valid otherwise). This resolution is the basis resolution for cam input source 'position'. Premise: R_{rim} ≤ encoder resolution × i. Else an error occurs and slew ring function will not be accepted.

7 Encoder and cam parameter

Parameter *(object/sub) - access*	**Value range** *(data length / default)*	**Parameter description**
Safety config. valid flag *(11FE/00)* - rw	0 / 0xA5 *(1 / A5)*	When parameters of objects 1100/xx shall be modified, cam valid flag has to be set to 0 first. After parameterization (CRC included) set this flag to 0xA5 again for operation.
Standard CRC check-sum *(11FF/01)* - rw	0 ... 65,535 *(2 / -)*	When parameters of objects 1100/xx shall be modified, this CRC has to be calculated and transmitted to NOCIO due to safety aspects. General polynomial: $2^{16} + 2^{12} + 2^5 + 1$ = 0x(1)1021 hex), initial value = 0x00 00, no final XOR. Special polynomial and inital value on request.
Cam state *(0x1300)* - ro	0 ... F *(1 / -)*	Provides current condition of the cams: FALSE/TRUE. TRUE = relay contact closed if a relay is assigned to related cam. FALSE = limit exceeded = relay contact open. FALSE/TRUE are swapped when cams are inverted (see cam polarity). **Bit** **Value** 0 0=FALSE / 1=TRUE → cam1 1 0=FALSE / 1=TRUE → cam2 2 0=FALSE / 1=TRUE → cam3 3 0=FALSE / 1=TRUE → cam4 A disabled cam is always FALSE even it is inverted.
Cam1 source *(0x1310/01)* - rw	0 ... 5 *(1 / 1)*	Selects the input value for limit detection and therefore trip function for the relay(s). 0 = no input (cam inactive) 1 = position 2 = speed (on request / in preparation) 3 = speed absolute value (CW/CCW not distinguished, currently not available for safety devices) 4 = acceleration (on request / in preparation) 5 = acceleration absolute value (CW/CCW and speed increase/decrease not distinguished, currently not available for safety devices)) Source data format: signed 32 bit. In case of position no sign used. Scaling factors are considered (e.g. speed multiplier/divider)
Cam1 low limit *(0x1310/02)* - rw	0...0xFFFF FFFF (signed 32 bit) *(4 / …)*	Cam1 = TRUE when: Low limit ≤ actual value < high limit. This limit value is valid for any source with its characteristics. The appropriate limit has to be insert to ensure proper trigger function. At source = 1/3/5: Only values ≥ 0 can be effective. Set this parameter to =0 for absolute overspeed/overacceleration detection in combination with cam1 high limit and source = 3 or 5.
Cam1 high limit *(0x1310/03)* - rw	0...0xFFFF FFFF (signed 32 bit) *(4 / …)*	Cam1 = TRUE when: Low limit ≤ actual value < high limit. This limit value is valid for any source with its characteristics. The appropriate limit has to be insert to ensure proper trigger function. At source = 1/3/5: Only values ≥ 0 can be effective. Use this parameter for absolute overspeed/-acceleration detection at source = 3/5 (cam1 low limit inactive for =0)
Cam1 hysteresis *(0x1310/04)* - rw	0...0xFFFF *(2 / 10 digits)*	When rotation direction changes in this interval after a TRUE / FALSE transition, the switching point is moved by this value in order to prevent constant state changes of the cam. Used to prevent relay fluttering.

7 Encoder and cam parameter

<table>
<tr><th>Parameter
(object/sub) - access</th><th>Value range
(data length / default)</th><th>Parameter description</th></tr>
<tr><td>Cam1 enable
(0x1310/05) - rw</td><td>0 ... 1
(1 / 1)</td><td>Enables or disables cam1 condition change due to exceeding limits (0 = cam disabled, 1 = cam enabled). A disabled cam is always FALSE even it is inverted.</td></tr>
<tr><td>Cam1 polarity
(0x1310/06) - rw</td><td>0 ... 1
(1 / 1)</td><td>Provides cam1 function "normal" or "inverted".
Normal: Cam1 = TRUE: Low limit ≤ actual value ≤ high limit.
Inverted: Cam1 = FALSE: Low limit ≤ actual value ≤ high limit.
(0 = cam not inverted, 1 = cam inverted)</td></tr>
<tr><td>Cam2 specifications
(0x1320/01) - rw
- (0x1320/06) - rw</td><td>see cam1 specifications</td><td>see cam1 specifications</td></tr>
<tr><td>Cam3 specifications
(0x1330/01) - rw
- (0x1330/06) - rw</td><td>see cam1 specifications</td><td>see cam1 specifications</td></tr>
<tr><td>Cam4 specifications
(0x1340/01) - rw
- (0x1340/06) - rw</td><td>see cam1 specifications</td><td>see cam1 specifications</td></tr>
<tr><td>Relay1 assign
(0x3010/01) - rw
Safety relay</td><td>0 ... F
(1 / 3)</td><td rowspan="2">Determines the trip function of the relay output:
Relay reacts on cam status (TRUE/FALSE) specified in the cam-specifications (see objects 13x0/xx). Camx = TRUE means relay = TRUE. When camx changes to condition FALSE the related relay changes to condition FALSE. Perhaps the relay has to be acknowledged to become TRUE again, when all assigned cams are TRUE again (Relay acknowledge function). At more than one cam assigned to the relay, the cams are combined with a logical "AND" (&).
Bit table:
Function | Bit | Value
Cam1 | 0 (LSB) | 1
Cam2 | 1 | 1
Cam3 | 2 | 0
Cam4 | 3 (MSB) | 0
 | 4 to 7 | not used
Shown: $00000011_{bin} = 3_{dez} = 3_{hex}$ = "cam1&2"
→ Means for example: position1 and position2 limit detection.
→ Relay = TRUE when cam1 & cam2 = TRUE</td></tr>
<tr><td>Relay2 assign
(0x3020/01) - rw
Non-safety relay</td><td>0 ... F
(1 / 4)</td></tr>
<tr><td>Relay1 acknow. config
(0x3010/02) - rw
Safety relay</td><td>0 ... 1
(1 / 1)</td><td rowspan="2">Determines if relay has to be reset by IO-Link command (reset = contact closes again (=TRUE) after tripping. See object 30FC: relay acknowledge or control byte, bit 0) or resets automatically when related limits are not exceeded any more (related cams change to condition TRUE).
Bit | Value
0 | 0=automatic reset / 1=acknowledge reset
If set to "1" (=acknowledge) after boot up of NOCIO the acknowledge command has to be sent to NOCIO once to set the relay to condition TRUE (related cams have to be TRUE as well and valid flags 11FE and 30FE have to be set "A5").</td></tr>
<tr><td>Relay2 acknow. config
(0x3020/02) - rw
Non-safety relay</td><td>0 ... 1
(1 / 0)</td></tr>
</table>

7 Encoder and cam parameter

Parameter *(object/sub) - access*	**Value range** *(data length / default)*	**Parameter description**
Relay1 acknowledge *(0x30FC/01)* - rw Safety relay	0 ... 1 *(1 / 0)*	Resets relay to condition TRUE when acknowledgement function is activated via "relay acknow. config" and all related cams are in condition TRUE. Else relay contact remains open (FALSE). Not CRC and valid flag obligatory. This object has same function like control bits 0/1. Parameter is set to "0" automatically after 2 seconds. **Bit** **Value** 0 0=not acknowledged / 1=acknowledged
Relay2 acknowledge *(0x30FC/02)* - rw Non-safety relay	0 ... 1 *(1 / 0)*	
Cam FALSE forcing *(30FD/00)* - wo (cam FALSE simulation by force)	0 ... F *(1 / 0)*	Provides relay test function to ensure that contact opens properly. Relay can be tested by setting one or more **cams** to condition FALSE by force. Precondition: Relay 1 and/or 2 is in condition TRUE. Therefore the related relay has to be assigned to one or more cams which is/are in cond. TRUE. Valid flags have to be set to "A5" (else relay is not TRUE). Cam is set to FALSE by force with this object (30FD). Therefore relay opens (FALSE) by force due to this test function. Chose an appropriate cam and set related bit to 1. Cam and relay condition is FALSE now. All cams can be set to FALSE by input 0xF. Then set bit to 0 again (no time limit). Cam will change in condition TRUE (when other conditions for TRUE are still given). Relay condition becomes TRUE after acknowledgement via related control bit or 30FC if relays are set to "ackn." via 30x0/02. Test can be executed at any operational state. PLC has to ignore the test procedure. Not CRC and valid flag obligatory. 0=no FALSE enforcement / 1= FALSE enforcement at: **Bit** **Value** 0 → cam1 1 → cam2 2 → cam3 3 → cam4 (1 1 1 1 = F_{hex} = all cams FALSE)
Cam switch valid flag *(30FE/00)* - rw	0 / 0xA5 *(1 / A5)*	When parameters of objects 13x0/xx and 30x0/xx shall be modified, cam valid flag has to be set to 0 first. After parameterization (CRC included) set this flag to 0xA5 again for operation
Cam1 CRC checksum *(30FF/01)* - rw	0 ... 65,535 *(2 / ...)*	When parameters of objects 1310/xx (cam1) shall be modified, this CRC has to be calculated and transmitted to NOCIO due to safety aspects. General polynomial: $2^{16} + 2^{12} + 2^{5} + 1$ = 0x(1)1021 hex), initial value = 0x00 00, no final XOR. Special polynomial and inital value on request.
Cam2 CRC checksum *(30FF/02)* - rw	0 ... 65,535 *(2 / ...)*	CRC for parameters of cam2 (1320/xx) See explanation at CRC object 30FF/01
Cam3 CRC checksum *(30FF/03)* - rw	0 ... 65,535 *(2 / ...)*	CRC for parameters of cam3 (1330/xx) See explanation at CRC object 30FF/01
Cam4 CRC checksum *(30FF/04)* - rw	0 ... 65,535 *(2 / ...)*	CRC for parameters of cam4 (1340/xx) See explanation at CRC object 30FF/01
Relay CRC checksum *(30FF/05)* - rw	0 ... 65,535 *(2 / ...)*	CRC for parameters of relays (30x0/xx) See explanation at CRC object 30FF/01

7 Encoder and cam parameter

Parameter *(object/sub) - access*	**Value range** *(data length: octet)*	**Parameter description (default)**
Vendor ID *(0x00)* - read only (ro)	*2 octets*	0x0159
Device ID *(0x00)* - read only	*3 octets*	0x00 xxxx (00 = Serial device, 01 = Prototype, $xxxx_{hex}$ → TYxxxxx) The connection assignment sheet number TYxxxxx is uniquely assigned to a product name.
Vendor Name *(0x10)* - read only	*max. 64 octets*	TWK-ELEKTRONIK GmbH
Vendor Text *(0x11)* - read only	*max. 64 octets*	www.twk.de/produkte
Product Name *(0x12)* - read only	*max. 64 octets*	NOCIO79-xxx2-xxxxR4096SxT2Lxx (xx = actual variant)
Product ID *(0x13)* - read only	*max. 64 octets*	NOCIO/S0 or NOCIO/S3
Product Text *(0x14)* - read only	*max. 64 octets*	Digital switching cam encoder
Serial Number *(0x15)* - read only	*max. 64 octets*	device specific
Hardware Version *(0x16)* - read only	*max. 64 octets*	P-0835-x (x = actual version)
Firmware Version *(0x17)* - read only	*max. 64 octets*	V1.2.0 or higher
ApplicationSpecific Tag (0x18) - read / write(rw)	*max. 32 octets*	device specific (customer input, not affected by appl. reset)
Function- Tag (0x19) - read / write	*max. 32 octets*	device specific (customer input, not affected by appl. reset)
Location- Tag (0x1A) - read / write	*max. 32 octets*	device specific (customer input, not affected by appl. reset)
Order Number *(0x40)* - read only	*max. 32 octets*	NOCIO79-xxx2-xxxxR4096SxT2Lxx (xx = actual variant)
Customer Part Number *(0x41)* - read only	*max. 32 octets*	xxxx123yyyy456zzzz
Manufacturing Date *(0x42)* - read only	*max. 32 octets*	yyyy/mm/dd (individual)
Installation Date *(0x43)* - read / write	*max. 32 octets*	device specific (customer input, not affected by appl. reset)

8 Diagnosis overview

8.1 Error types

Encoder errors are indicated by a red device LED (see data sheet 15893 for status LEDs description). There are two different error modes with different behaviour of the device.

8.1.1 Device error

Examples: relay error, supply voltage error, internal sensor system error):

- Device set into a safe state (relays open)
- I/O-Link communication still possible
- Device LED constantly red
- Position / Speed / Acceleration data set to zero
- Encoder error status bit set to 1
- Error type written to *Device error history* register (0x1003)

A reset of the device is required to exit the error state. Reset can be executed by a power cycle or by executing a device reset via IO-Link (button 128). Note, that both safety flags need to be set to 0 before a device reset can be executed.

8.1.2 Critical (hard) error:

- Device set into a safe state (relays open)
- I/O-Link communication stopped
- Device LED uses flashing codes to indicate error type (see below)
- Error type written to *Device error history* register (0x1003)

A reset (power on/off) of the device is required to exit the error state.

8.2 Error register 0x1003

All errors are recorded in the error register 0x1003, which is listed in the Specialist Role menu set. The newest error is given at the first position of the register.

The register can record up to 20 errors. Once the error register is full, it cannot record new errors. It needs to be reset with button 0xFF (write 0xFF to object 0x02/0) before it can record new errors.

To avoid filling the register with identical, periodically send errors, each error is only recorded once between two resets/power ups of the device. Up to five different errors can be recorded in a power cycle, which are subsequently written into the error log.

Error codes are recorded in four bytes. The two error types listed above are recorded in the following way:

Registered error code	Byte 0	Byte 1	Byte 2	Byte 3
Normal error	Error source bit	Error Code Byte 0	Error Code Byte 1	Error Code Byte 2
Hard error	0x08	0x00	Error Code	0x08

8 Diagnosis overview

The error bit recorded in case of an encoder fault is associated with the following sources:

Error source	Error source bit
Sensor chip error	0x80
Supply voltage/current error	0x40
CRC Error (EEPROM, FLASH)	0x20
Cam error	0x10

The supply voltage/current error will return the following Error Code Byte 0:

Voltage/current error	Error Code Byte 0
Voltage exceeds maximum	0x01
Voltage exceeds minimum	0x02
Maximum power exceeded	0x10
Voltage dropout	0x20

8.3 LED flashing code

Errors which lead to encoder system standstill (hard errors) are indicated by a flashing code on the part of the red Device LED. Following introductory flickering by the red LED, a specific number of flashing cycles are output for the cause of the error.

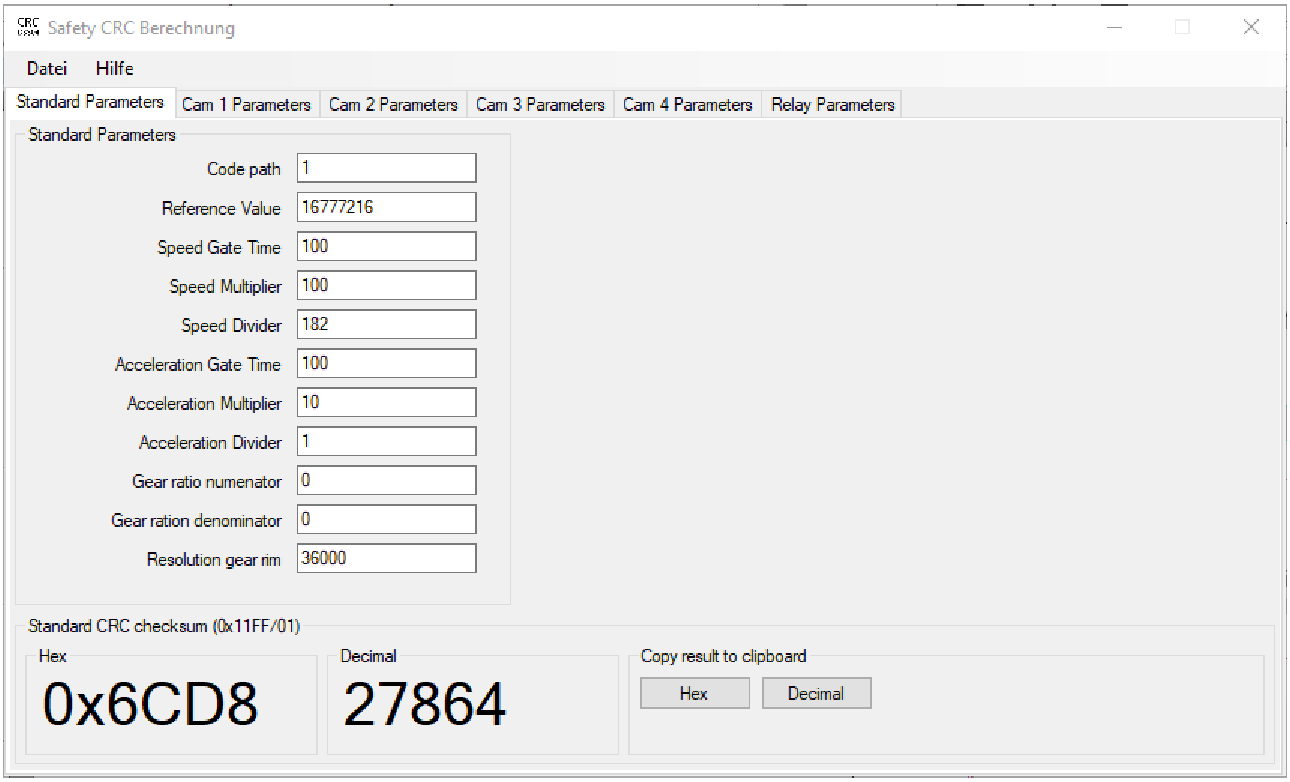
	Number of flashing cycles (Duration approx. 1 s)	Error cause
Flashing code 2	2	CRC error ROM
Flashing code 3	3	RAM/XRAM error
Flashing code 4	4	Sensor / relay / cam error
Flashing code 5	5	Programme sequence error
Flashing code 6	6	Power consumption too high

Appendix

A.1 Calculation of the cam and encoder checksums

The program SafetyCRC is a universal CRC calculation program for different types of devices. Use Version 2.0.7 or higher for calculating NOCIO CRC checksums. The latest program version can be obtained from the TWK webpage via **www.twk.de/files/CRC-Calculator20.zip**.

To set it up for the NOCIO, start the program and load the XML file "NOCIO" using "Datei → Datei laden". You should then see the following window:



The register "Standard Parameters" is used to calculate the checksum 11FF over the encoder parameter. Registers "CAM 1-4 Parameters" is used to calculate the checksum 30FF/01-04 over the cam parameter. Register "Relay Parameters" is used to calculate the checksum 30FF/05.

Customer specific XML files with different CRC polynomials and/or start values are available on request.