ONLINEHELP

Drive Monitor FX3-MOC

Motion Control

GB

SICK
Sensor Intelligence.

Copyright

Content

1 About the Drive Monitor FX3-MOC 5
1.1 Product Features 5
1.1.1 Features 5
1.1.2 Your benefits 6
1.1.3 Target applications 6
1.1.4 Industries 6
1.2 Application Examples 7
1.2.1 Stationary applications 7
1.2.1.1 Access protection with standstill detection 7
1.2.1.2 Maintenance and service mode 8
1.2.1.3 Emergency stop SS1 9
1.2.1.4 Emergency stop SS2 10
1.2.1.5 Summary of the stationary applications 10
1.2.2 Mobile applications 11
1.2.2.1 Safe operating stop SOS 11
1.2.2.2 Safely limited Speed SLS 12
1.2.2.3 Safe direction SDI 12
1.2.2.4 Scanner field switching with Safe Speed Monitor SSM 13
1.2.2.5 Summary AGV 13

2 Encoders in the FX3-MOCx motion control module 14
2.1 A/B incremental encoders 15
2.2 Sin/Cos encoder 15
2.2.1 Overview of special parameters for sin/cos encoders 15
2.2.2 Sin/Cos analog voltage monitoring 15
2.2.3 Sin/Cos- increased resolution 16
2.3 SSI encoder 17
2.3.1 Overview of special parameters for SSI encoders 17
2.3.2 Double data transmission 17
2.3.3 Error bit evaluation 18
2.3.4 Max. data reception interval 18

3 Encoder configuration 19
3.1 A/B incremental encoders 19
3.2 Sin/Cos encoder 23
3.3 SSI master/listener 24

4 Connection of encoders 26
4.1 Encoder connection 26
4.2 Connection boxes for encoders 28

5 Data exchange CPU - MOC 33
5.1 Data exchange between main module and MOCx 33
5.2 Data types in the MOCx logic- 34
5.3 Function blocks for data conversion 36
5.3.1 Speed to laser scanner 36
5.3.2 UI8 to Boolean converter 36
5.3.3 Boolean to UI8 converter 36
5.3.4 Example 37

Content

6 Function blocks for monitoring functions 38
6.1 Speed Cross Check 39
6.2 Speed Monitor 49
6.3 Safe Stop 59

7 Configuration for an AGV 63
7.1 Hardware 63
7.2 MOC Logic 65
7.3 CPU Logic 70

8 Configuration for a Gantry 73
8.1 Hardware 73
8.2 MOC Logic 75
8.3 CPU Logic 87

9 Diagnostic 98

10 How to get the system running 100

11 Frequent error causes 103
11.1 General error causes 103
11.1.1 Mechanics errors 103
11.1 .2 Electromagnetic Compatibility (EMC) 103
11.2 Speed Cross Check 103
11.3 Faulty hardware configuration 104

12 Detailed technical data 105

13 Accessories 107

14 Further informations 108
14.1 Dimensional drawing 108
14.2 Ordering information 108

1 About the Drive Monitor FX3-MOC

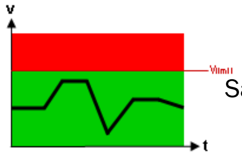
The MOCx motion control modules are modules for drive monitoring. With their help a very wide range of types of drives (electrical, pneumatic, hydraulic etc.) can be safely monitored, if suitable sensors are installed.

The function blocks which are described in this chapter are just available in conjunction with a MOCx module. They are specifically tailored to applications for drive monitoring. In general two types of function blocks are used. The first of these are the actual monitoring function blocks, which monitor speed, position or stop and brake functions.
The Others are function blocks for data conversion. These function blocks are necessary, because the MOCx modules can also process integer data types, in contrast to the rest of the Flexi Soft system.

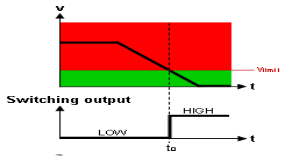
1.1 Product Features

1.1.1 Features

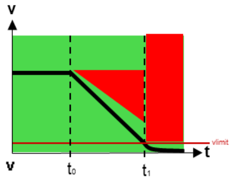
- 7 drive safety functions:



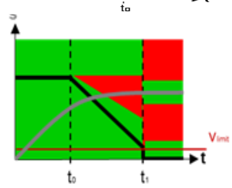
SLS
Safely-limited speed



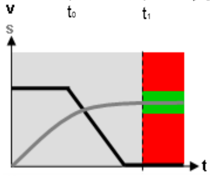
SSM
Safe speed monitor



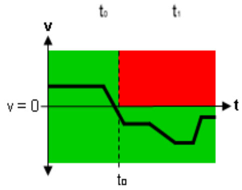
SS1
Safe stop 1



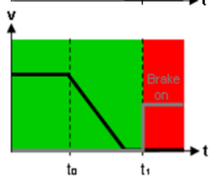
SS2
Safe stop 2



SOS
Safe operating stop



SDI
Safe direction



SBC
Safe brake control

- For all common encoder interfaces
- Programmable logic
- Monitoring of up to 10 speed levels and 4 brake ramps
- Monitoring of multiple axes is possible

1.1.2 Your benefits

- Integration into a Flexi Soft system with one software tool and one project file allows quick project planning and commissioning
- Easy logic development by using predefined, modifiable, freely configurable applications
- Optimal level of integration into higher-level controllers via all common fieldbus systems through gateways
- Documentation of the entire safety application simplifies machine acceptance and validation
- Monitoring movements instead of shutting down increases machine productivity
- Flexibility because of a wide range of drive safety functions

1.1.3 Target applications

- Handling systems
- Portal robots
- Packaging machines
- Printing machines
- Filling machines
- Rack feeders
- AGVs
- Drilling-, Milling- and CNC-machines
- Servo presses
- Turning tables
- Winding (foil, paper, metal)
- Tire production

1.1.4 Industries

- Machine building
- Automotive
- Tire manufacturing
- Press manufacturers
- Wood
- Paper
- Print
- Intra-logistics
- Transport-logistics
- Packaging

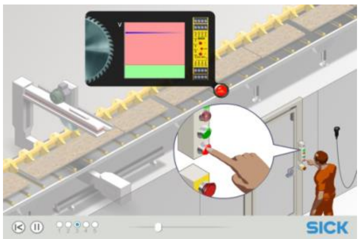
1.2 Application Examples

1.2.1 Stationary applications

Machines where the operator is separated from the hazardous point by mechanical measures like housing, cover, moveable guards

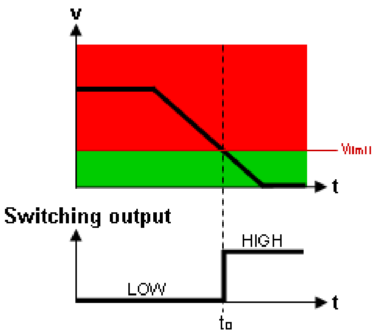
- Access protection with standstill detection
- Maintenance and service mode with limited speed
- Emergency stop request

1.2.1.1 Access protection with standstill detection



<u>**Application: The unlocking of a safety door for saw line**</u>

A safe speed monitor (SSM) signal has to be provided for a saw line. For this reason the rotation speed will be checked. When the rotation speed is located in a safety area, a signal will be sent to the safety controller. When the saw is at standstill, the safety controller gives the command to unlock the safety door whereupon the safety door can be opened.



SSM releases a safe signal when the speed is exceeding or undercutting a configured limit.

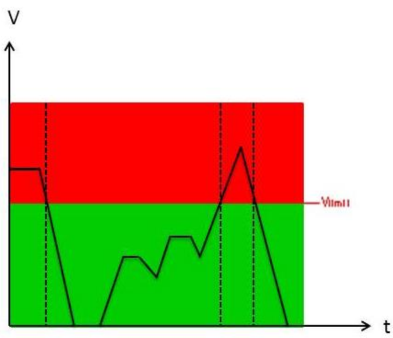
Your benefits:

- It shows the standstill of the drive
- Fast access into the machine and fast intervention
- Rise in the productivity

1.2.1.2 Maintenance and service mode

Application: Maintenance with limited speed on a saw line

Maintenance has to be done with reduced speed to minimize the risk for the operator. Therefore the Drive Monitor monitors the speed. When the speed exceeds the limit because of a malfunction, the Drive Monitor switches off the drive.

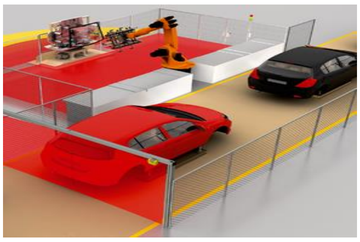


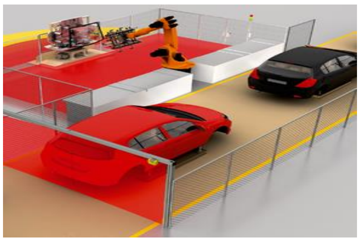
The SLS function monitors the speed of the drive. The drive will be stopped by the drive monitoring device e.g. via STO, when a configured speed limit is exceeded.

Your benefits:

- Drive is only allowed to move with a limited speed
- Higher efficiency, beacause downtime of the machine/system is minimized
- Convenient and safe maintenance

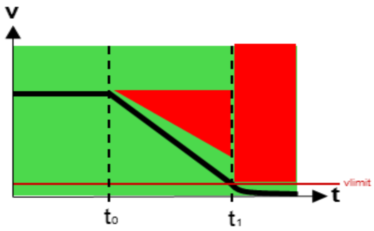
1.2.1.3 Emergency stop SS1





<u>Application: Access protection with light curtain</u>

When an operator enters a danger area, the SS1 function will be activated and a safely monitored emergency stop ramp will be activated for the directions of motion. The motion will be stopped fast so that the operator will not be injured. The STO function will be activated when the speed is below the limit.



SS1 induces the brake operation of the drive via an emergency stop ramp at the point t_0. This brake operation is monitored by the Drive Monitor. When the current speed is below the limit (standstill = t_1), then the energy supply of the drive will be disconnected (STO).

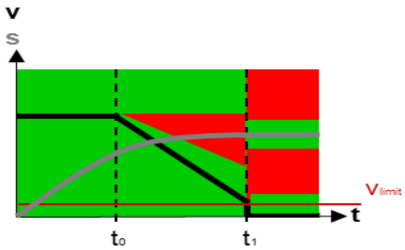
Your benefits:

- Stop via emergency stop ramp by monitoring the brake operation
- Fast intervention of the user
- Rise in the productivity

1.2.1.4 Emergency stop SS2

Application: Unwarranted access on the hazardous area

When a person enters the area which is protected with light curtain, the SS2 function will be activated. The motion of the machine will be stopped with an emergency stop deceleration and the Drive Monitor detects when the rotation speed is below the configured limit. Via SOS function the load of the cranes will be held in the position in which it has been stopped.



The SS2 function starts the brake operation via an emergency stop deceleration at point t_0. This brake operation will be monitored by the Drive Monitor. When the speed is below the configured speed limit (t_1), then the SOS function will be activated.

Your benefits:

- Controlled stop via an emergency stop ramp after a certain time
- Faster intervention possible
- The machine can start a new cycle without a readjustment

1.2.1.5 Summary of the stationary applications

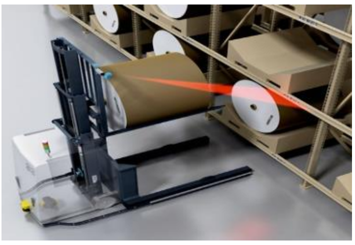
Function	Feature	Benefit
Access protection with standstill detection	Fast access into the machine and fast intervention	Rise in the productivity
Maintenance and service mode	Drive is only allowed to move with a limited speed	Better performance and machine availability, because idle times of machines can be minimized
Emergency stop SS2, after process interruption	• Faster intervention possible • The machine can start a new cycle without an readjustment	• Higher productivity • Less downtimes

1.2.2 Mobile applications

Vehicles where humans may be located in the danger area, which are secured by optical safety devices (safety laser scanner)

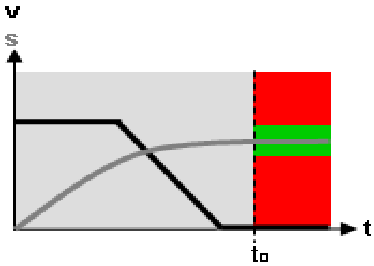
- AGVs, AGCs, cranes
- Speed monitoring and selection/activation of scanner fields depending on the speed
- Emergency stop request

1.2.2.1 Safe operating stop SOS



Application: Lifter

After the pallet has been lifted on the shelf, it should be held until it is completely out of the shelf, then it starts to drop. The drive monitoring device monitors that the position of the load is held.

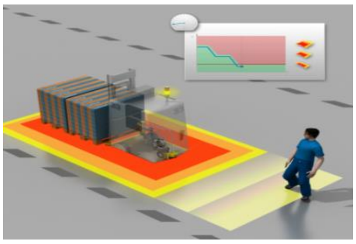


The function SOS is used to remain the drive/ rotor in the position, where the function was activated at t_0. That means that the energy supply of the drive is retained and the drive operates against the extrinsic force and torque so that the position of the rotor will be maintained. The palette remains after lifting in the rest position.

Your benefits:

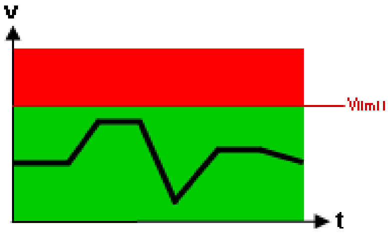
- The drive or application will be stopped into the current position and remains on energy supply
- Safe intervention is possible without a new adjustment of the machine
- Rise in the productivity

1.2.2.2 Safely limited Speed SLS



Application: Automatic guided vehicles (AGV)

When an AGV travels into a direction of an obstacle, the laser scanner sends a signal to reduce the speed. The speed monitoring of the drive monitoring device is acitvated. The AGV doesn't have to stop, but it can approach slowly and controlled to the obstacle.
The drive has to stop only when the protection field of the scanner is violated or the speed will not be reduced to the expected level.



The SLS function monitors the speed of the drive.
The drive will be stopped by the drive monitoring device e.g. via STO, when a configured speed limit is exceeded. In the application the AGV system can approach slowly to an obstacle.

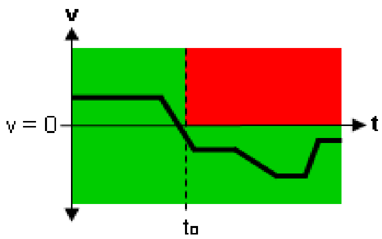
Your benefits:

- The AGV is only allowed to move with a limited speed
- Controlled approaching of the AGV to the obstacle is possible
- Better performance, because idle times of the AGV system can be minimized

1.2.2.3 Safe direction SDI

Application: Automatic guided vehicles (AGV)

In an AGV-application the scanner recognizes an obstacle. The speed has to be reduced and if the protection field is violated, the AGV will be stopped completely. The drive monitoring device activates the SDI function whereby the currently used rotating direction of the drive will be blocked i.e. now the vehicle can only move away from the obstacle.

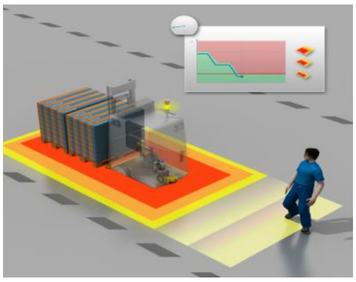


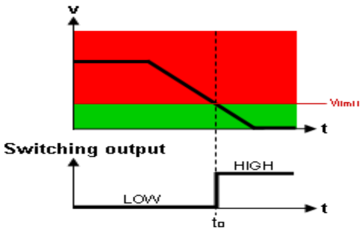
The function SDI ensures that the drive can only move in one direction. In the application it will be assured that the AVG can only move away from the obstacle

Your benefits:

- It ensures that the drive can only move in one direction
- Raising efficiency
- Flexible shape of the transport process

1.2.2.4 Scanner field switching with Safe Speed Monitor SSM





Application: Switching the safety field of a laser scanner

A safe speed monitor (SSM) signal has to be provided to switch the fields of a scanner depending on the speed of the vehicle. For this reason the rotation speed will be checked. When the rotation speed is located in a safety area, a signal will be sent to the safety controller. Then the safety controller passes the command to the scanner to switch the field.

SSM releases a safe signal when the speed is exceeding or undercutting a configured limit. In the application, the signal can be used to switch scanner fields.

Your benefits

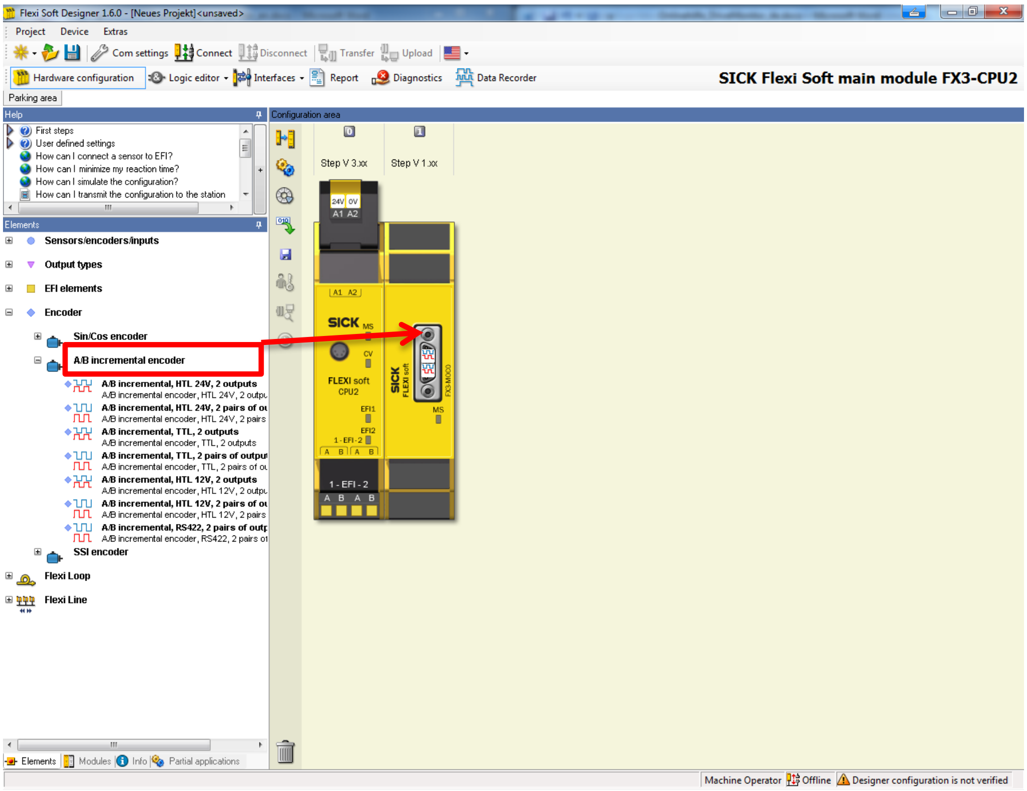
- It shows the speed value of the drive
- Engineering in one file
- Rise in the productivity

1.2.2.5 Summary AGV

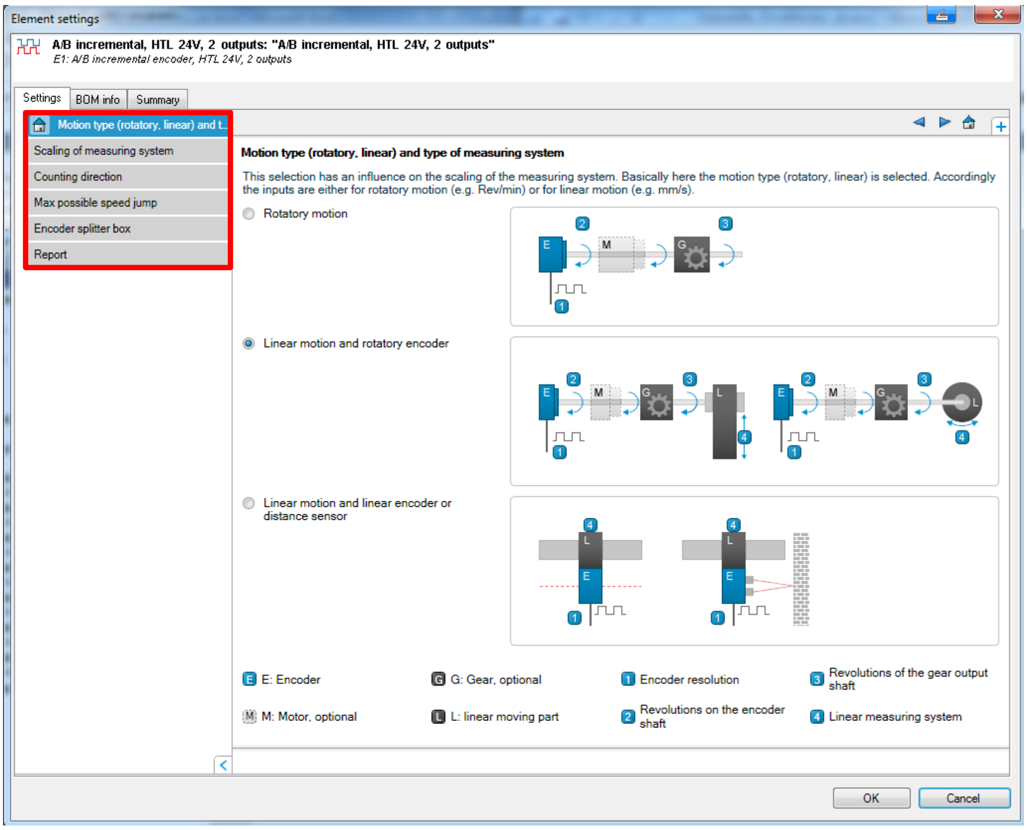
Function	Feature	Benefit
Safe Speed Monitor SSM	Detecting speed, selecting protecting field of scanner, EFI communication	• Less wiring • More flexible navigation
Emergency stop SS2, after detection of an obstacle	• Drive will be decelerated electrically • No use of mechanical brake	• Fast restart • Less abrasion of mechanical components • Less downtimes
Safe operating stop SOS and direction detection SDI	• Fast restart when obstacle is away • More flexibility for navigation	• Higher productivity • Less downtimes

2 Encoders in the FX3-MOCx motion control module

There are a total of 3 encoder groups to choose from. With drag & drop it's possible to draw or erase the different types of encoders on the Drive Monitor. To get to the encoder configuration double click on the encoder symbol.



At the beginning all encoders are treated equally.



All three encoder groups have the tabs “Motion type”, “Scaling of measuring system”, “Counting direction”, “Max possible speed jump”, “Encoder splitter box” and “Report”, which initially all be treated equally.

The group of A/B incremental encoder has no further adjustment.

The group of Sin/Cos encoder has the additionally tab “Sin/Cos analog voltage monitoring”.

The group of SSI encoder has the additional tabs “SSI Settings” and “Error evaluation”. More information contains chapter 3 “Encoder configuration”.

2.1 A/B incremental encoders

For this encoder type there are no encoder-specific settings or monitoring features.
To reach the required safety level, function blocks in the MOCx logic can be used to check the information provided by the encoder (motion data).

2.2 Sin/Cos encoder

2.2.1 Overview of special parameters for sin/cos encoders

Parameters	Description	Possible values
Sin/Cos analog voltage monitoring	See section 10.3.2 “Sin/Cos analog voltage monitoring” on page 279 (guide book).	• Inactive • Active
Increased resolution	See section 10.3.3 Sin/cos increased resolution” on page 283 (guide book).	• Normal • Inverted

2.2.2 Sin/Cos analog voltage monitoring

This function is used to detect errors in the encoder system. This can be helpful particular in applications in which an axis is to be monitored using only one sin/cos encoder. With sin/cos analog voltage monitoring it is checked whether the sine voltage and cosine voltage are in the required ratio in relation to each other.

If the sin/cos analog voltage monitoring detects invalid voltage ratios, then the status bits in the motion data for the related encoder are set to invalid.

The status bits become valid again if the following conditions are met for at least 1 second without interruption:

- The sin/cos analog voltage monitoring detects valid ratios.
- All other checks made also provide a positive result.

The ratio between the sine voltage and the cosine voltage is checked for two criteria by the MOCx module during the sin/cos analog voltage monitoring:

- vector length
- signal swing

Further informations are to find in the Operating instruction, chapter 10.3

2.2.3 Sin/Cos- increased resolution

This function is available on sin/cos encoders and is relevant for sin/cos encoder systems with a low resolution that can result in the coarse graduation of the speed measurement. With increased resolution activated, the number of counting points is increased by a factor of 4 and in this way the resolution of the speed measurement improved.
Without increased resolution (deactivated). With increased resolution (activated).

If the resolution of the speed measurement without sin/cos increased resolution is already less than or equal to the internal representation of the speed value in the data type motion (1 digit = 0.5 rev/min or 1 mm/s), then the activation of this option has no effect.

2.3 SSI encoder

2.3.1 Overview of special parameters for SSI encoders

Parameters	Description	Possible values
Baud rate	The baud rate for the clock output as SSI master.	• 0 = Listener • 100-1000 kBaud
[1] Number of bits for the entire SSI protocol frame	Number of clock cycles for a transmission	16 - 62
2] Number of leading bits	Number of leading bits that do not contain position data	0 - 46
[3] Number of position data bits	Number of bits that contain the relevant position data bits	16 - 32
Double data transmission	Selection whether the position value is transmitted once or twice in an SSI protocol frame	• Transmission of a single position value • Double transmission of the position value
[4] Number of bits between the position data bits	Only available with double transmission of the position value	0 - 30
Data code	Data code for the position data bits	• Binary • Gray
Error bit evaluation	Monitoring of error bits that are provided in the SSI protocol frame by the encoder	For each bit that is not a position data bit • 1 is error • 0 is error
Max. data reception interval	Maximum time in which valid position data are expected	4 - 100 ms

2.3.2 Double data transmission

Certain SSI encoders support multiple data transmission. During this process the same encoder data are output again, if during this process the pause between the data packages (monoflop time) is not exceeded. In this way, e.g., data corrupted by transmission interference can be detected.

The MOCx module supports double position data transmission. If double data transmission is activated, then it is checked by the MOCx module whether the two values for the position data in the SSI protocol frame are identical. If they are not identical, then the position data from this SSI protocol frame are ignored.

2.3.3 Error bit evaluation

Certain SSI encoders also transmit in the SSI protocol frame, in addition to the position data bits, error bits that reflect the result of internal encoder monitoring functions. Such error bits can be evaluated using the MOCx module.
During this process it can be defined individually for each bit whether 1 or 0 represents the error state. If the error state is detected for at least one selected error bit, then the position data in this SSI protocol frame will be ignored.

2.3.4 Max. data reception interval

With this function it is possible to temporarily tolerate invalid position data and to use the last valid position data for this time. If not all relevant checks were valid at least once for longer than the Max. data reception interval, then the status bits in the motion data for the related encoder are set to invalid.

The status bits are valid again if the Max. data reception interval was met for at least one second without interruption.

On the SSI encoder the timer for the Max. data reception interval starts when one of the following monitoring functions delivers a negative result:

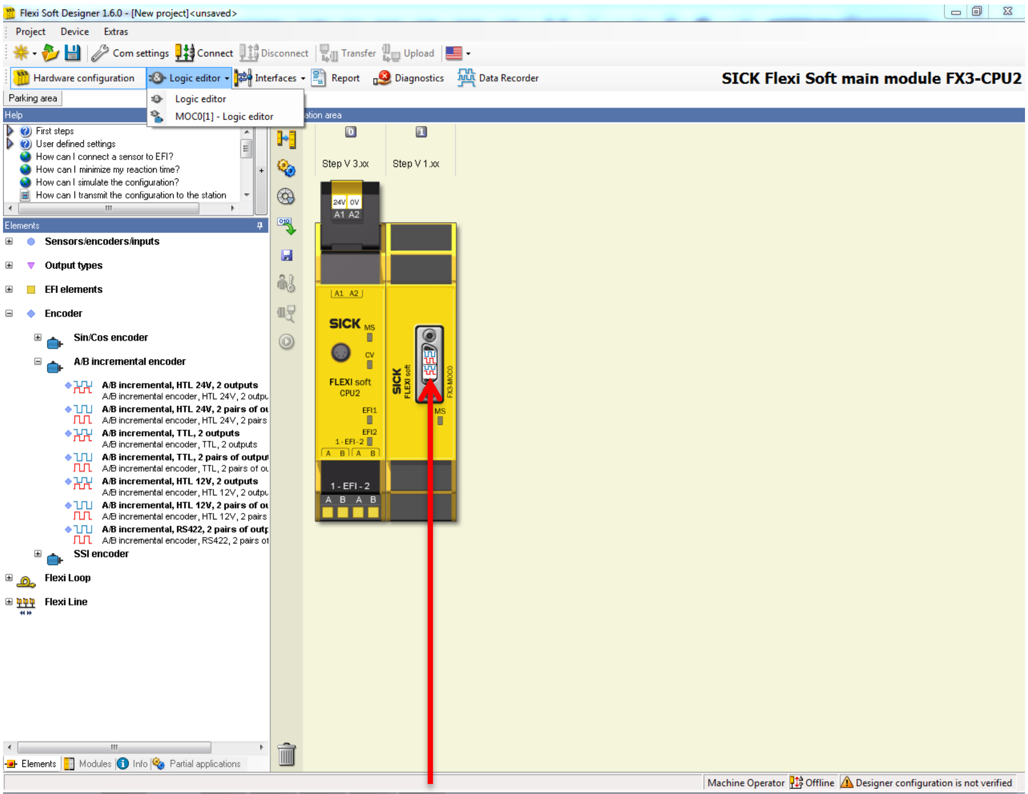
- ID monitoring
- SSI protocol frame not received or only incompletely received (only applies for SSI master)
- double data transmission
- error bit evaluation
- max. speed step

The MOCx module only ever evaluates just one SSI protocol frame in the 4 ms cycle. If the SSI listener transmits several SSI protocol frames within 4 ms, the addition SSI protocol frames are not evaluated.

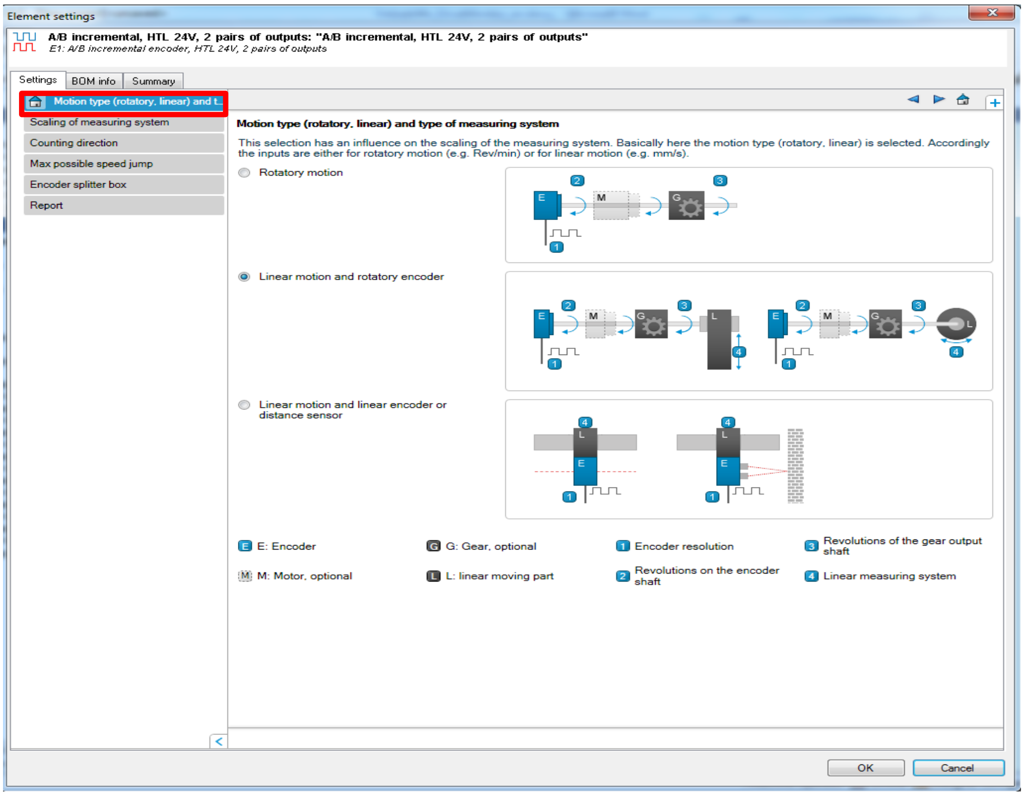
3 Encoder configuration

3.1 A/B incremental encoders

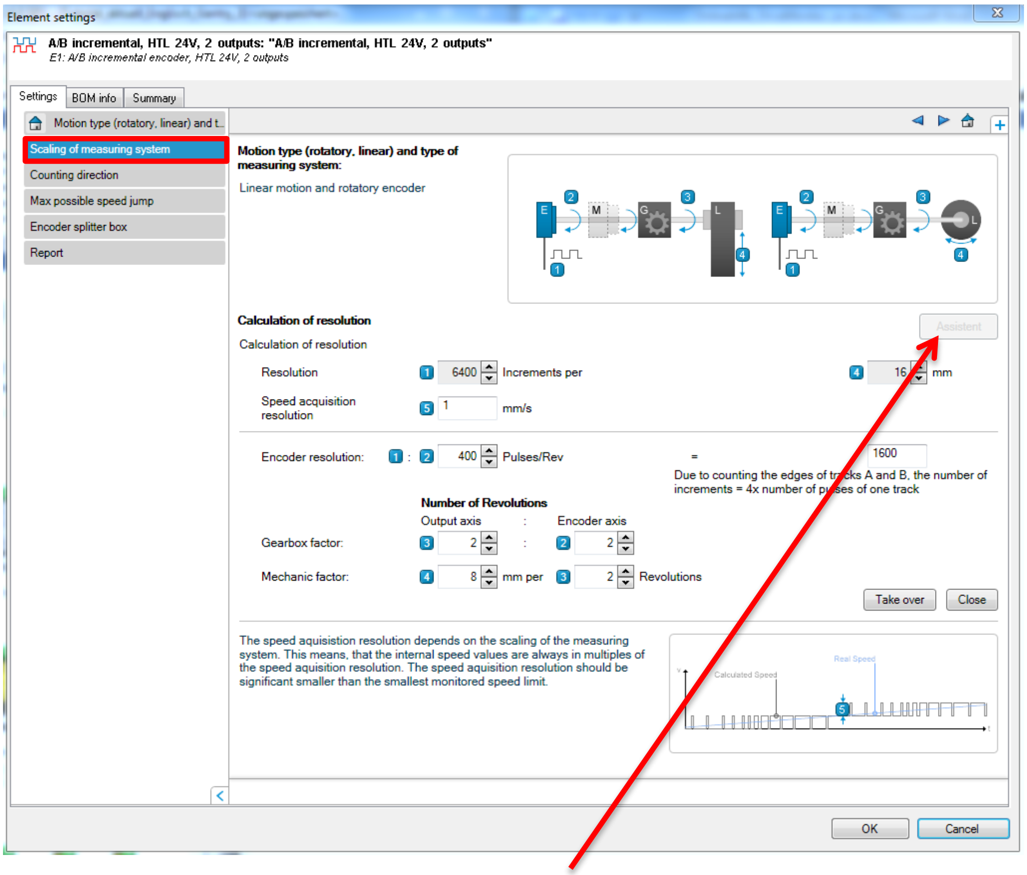
In the following is an example, which is not suitable for every application.



To get to the encoder configuration double click on the encoder symbol.



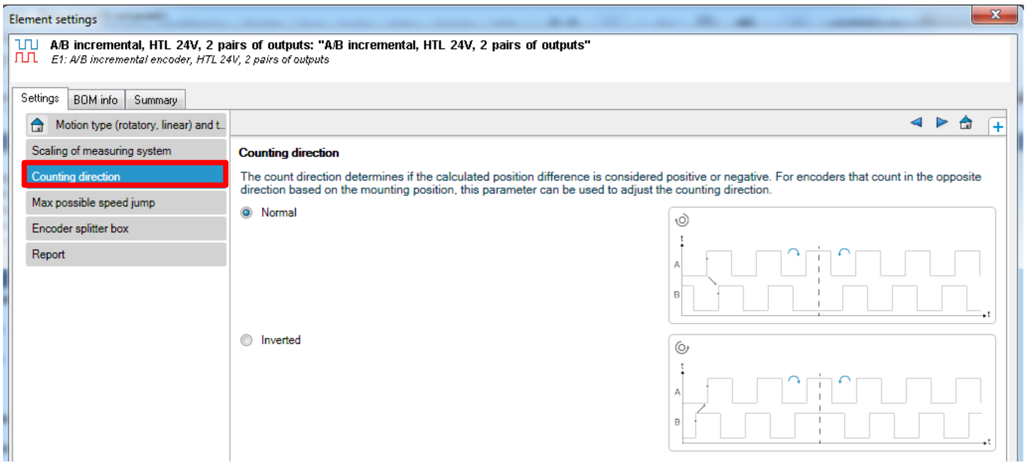
In the tab "Motion type" the used type of motion is selected.



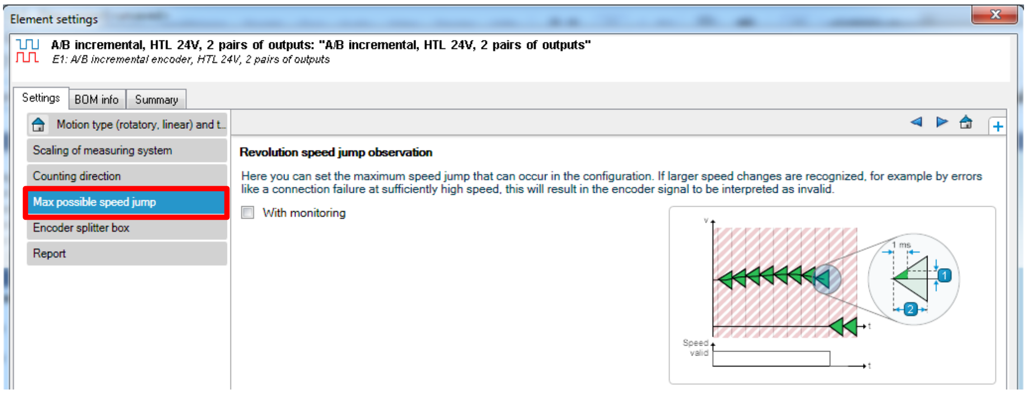
With the “Scaling of measuring system”, the ratio between the information which is provided by the encoder and the part moved mechanically is defined (number of increments per turn or per mm, depending on movement type).

The available Assistant allows it to calculate the scaling, considering a gearbox factor and a mechanical factor.

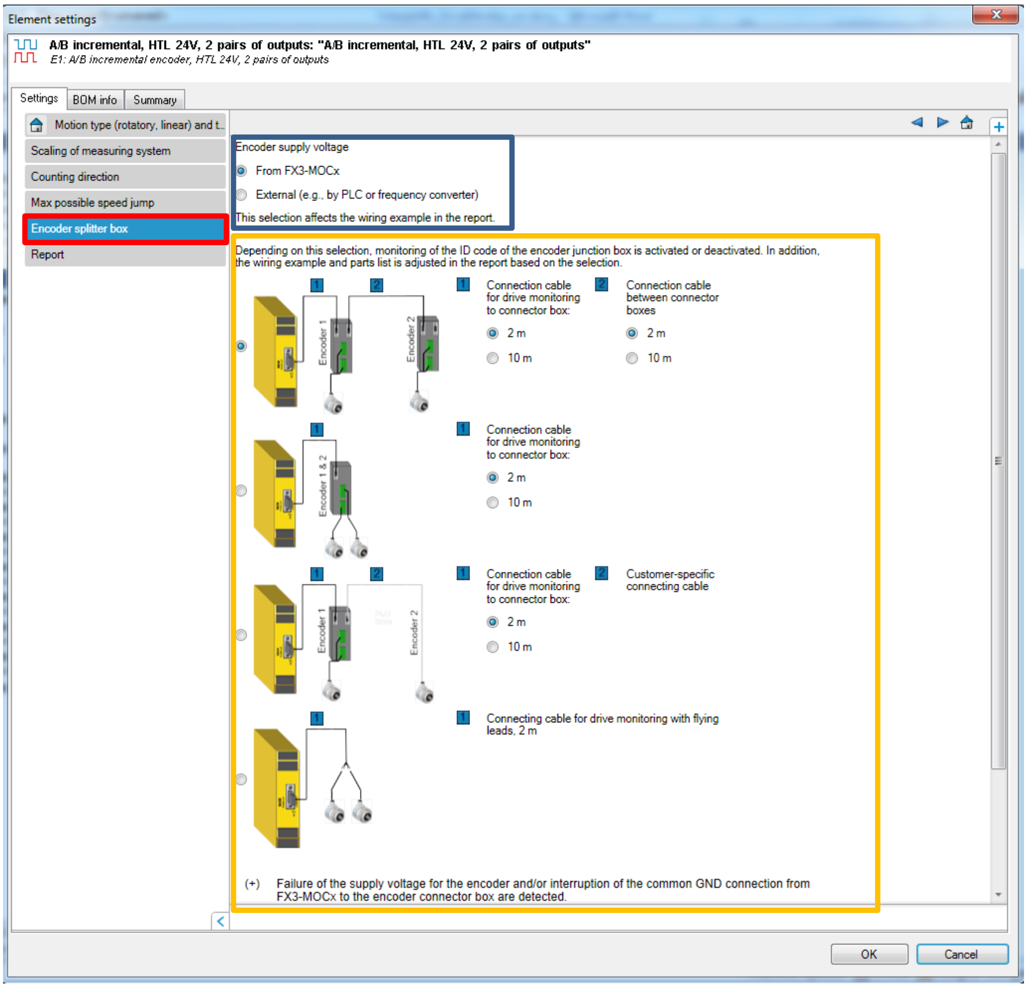
Based on this scaling, the information which is provided by the encoder is converted so that the internal motion signal always has a consistent representation and because of that, a usage in the MOCx logic is possible independent of the measuring system scaling.



The counting direction defines if the determined position change is evaluated as positive (normal) or negative (inverted). In case of encoders which count in the opposite direction because of their installation position, the counting direction can be changed using this parameter.



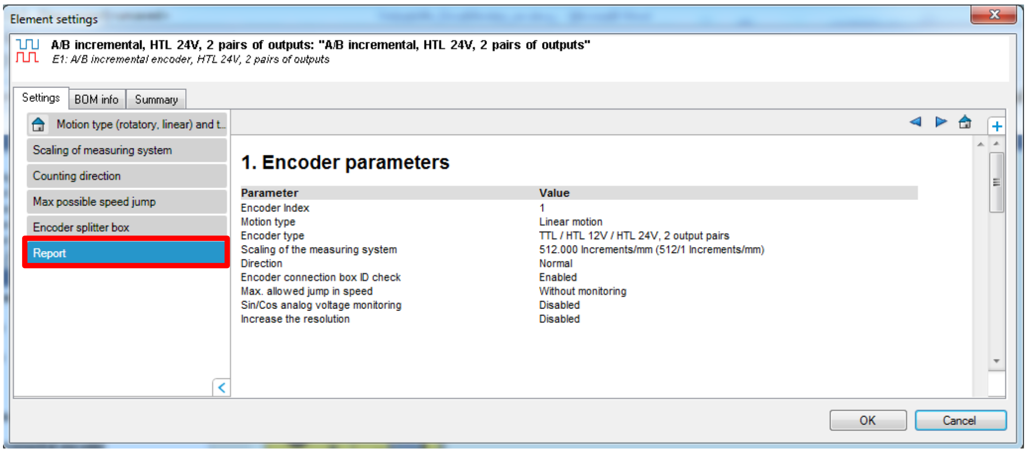
The maximum speed step which can occur in a configuration/application can be entered here. This increases the diagnostic ability of the security application because a demolition/ defect of an encoder can be determined.



The type of the encoder connection defines if an encoder connection box is used for the encoder. The monitoring of the encoder connection box ID is activated or deactivated based on this selection.

The selection of the supply voltage (“From FX3-MOCx” or “From external”) has no effect on the function of the device. Corresponding to the selection, only the wiring diagram in the Flexi Soft Designer report is adapted.

In the encoder connection box there is an ID in conjunction with the outputs for the encoder voltage supply from the MOCx module (“ENC1_24V” or “ENC2_24V”). If a type of connection with at least one encoder connection box is selected in the configuration, then the MOCx module cyclically checks this ID code.
(Further information: Operating instruction page 277)



A summary of the configuration of the function block, including all input and output links and the configured parameters is displayed on the tab “Report”.

3.2 Sin/Cos encoder

The configuration of the Sin/Cos encoder is similar to the other incremental encoder, but there is an additional parameter set “Sin/Cos analog voltage measuring”.

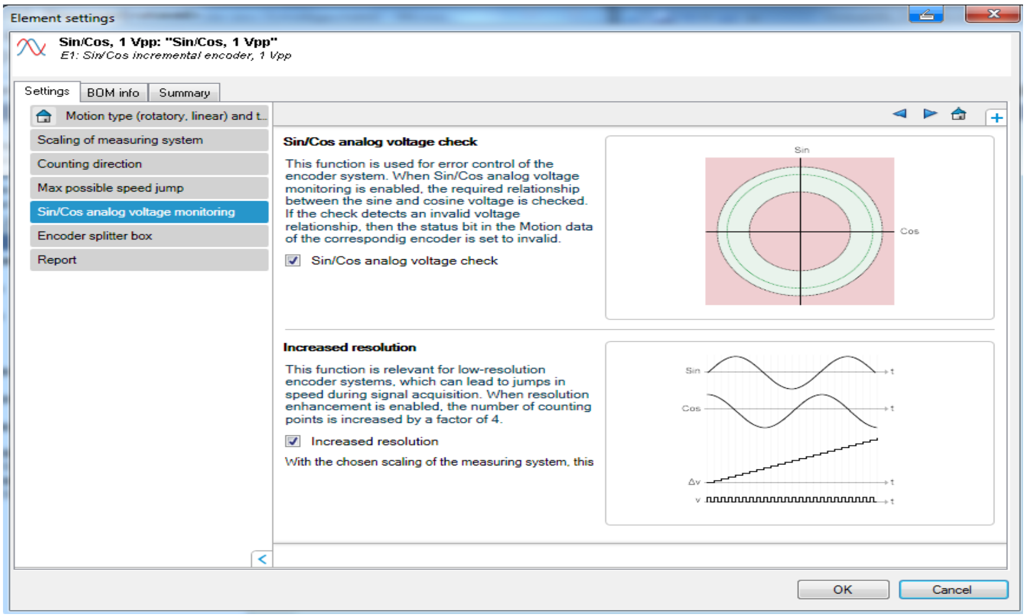
Sin/Cos analog voltage check:

When this parameter is activated the Diagnostic Coverage DC will be increased, because faults on the encoder or the connection of the encoder can be detected.

When it is deactivated PL d can not be achieved with one single Sin/Cos encoder(see chapter 2.2.2).

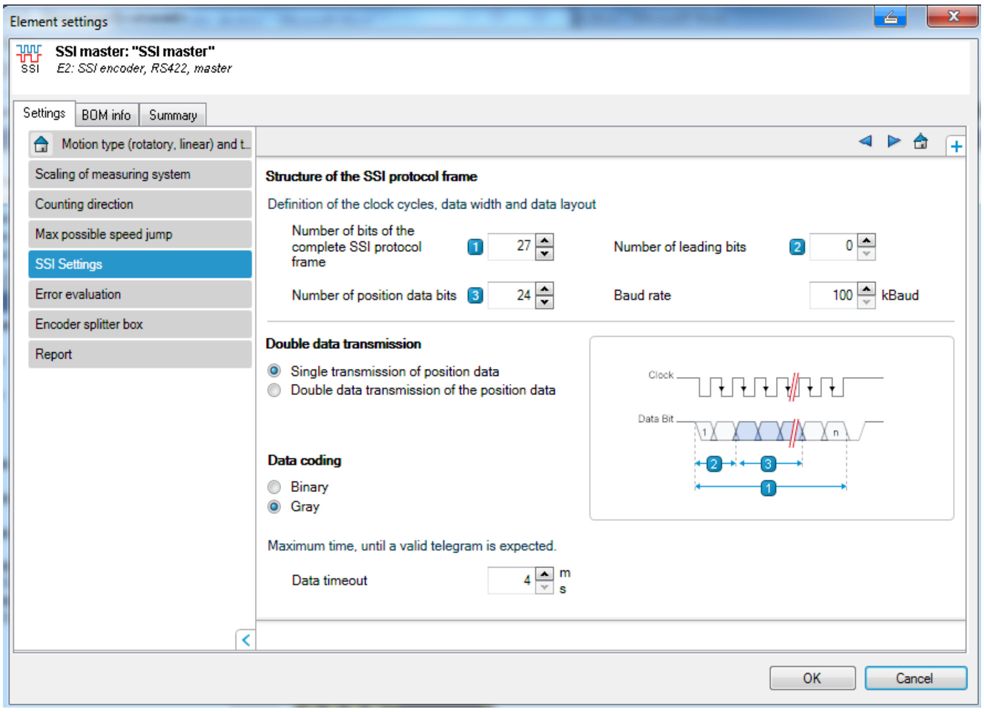
Increased resolution:

Further informations in Chapter 2.2.3.

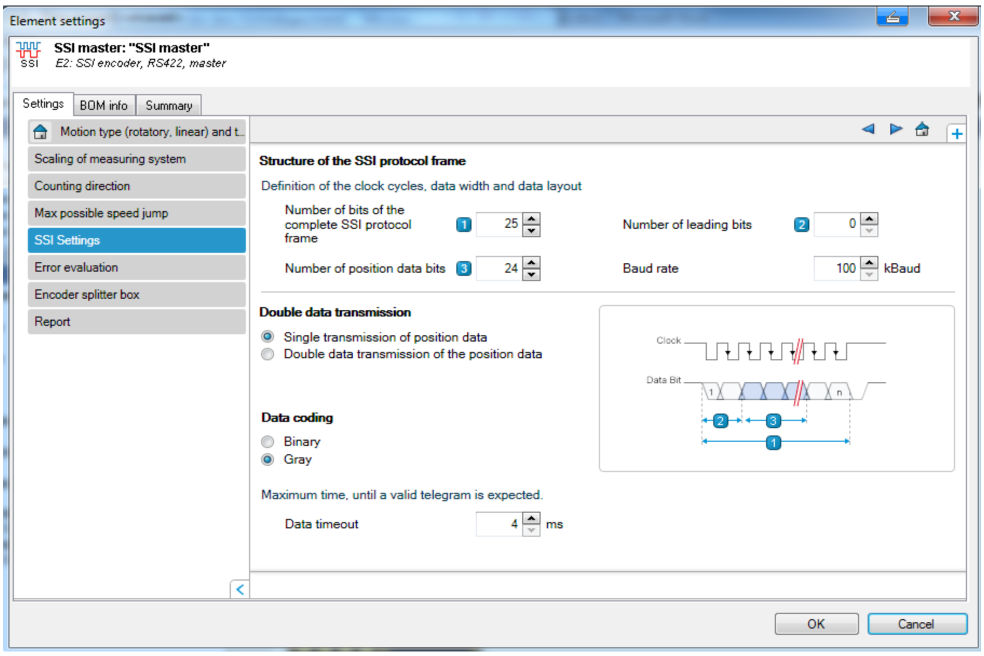


3.3 SSI master/listener

SSI encoders can often be found as linear measurement sensor. For logistics applications typical candidates are DME5000, DL100 and OLM
For machine tools applications typical candidates are TTK-70 or similar systems.
Thus there is a broad spectrum between very large distances as well a very high precision.
SSI sensors typically measure the position and the Drive Monitor calculates the speed as a difference between two positions and the delta time. This generally leads to more dynamic speed changes and a less continuous system.
Due to the measurement principle and transmission path, there may be less continuous velocity profiles with small velocity jumps (even at standstill) in these systems. This has to be considered during the configuration, to keep the reliability of the system high.



DME generally works with the following configuration:



The “Number of bits of the complete SSI protocol frame” is higher than “Number of position data bits”. For example DME provides additional information, regarding diagnosis and sensor error in the SSI protocol. The meaning and interpretation of these bits is defined in the section “Error evaluation”.

Number of leading bit:

This parameter defines if the position information (measured data) starts with the first or any other bit of the SSI frame. So with this parameter it can be configured where additional diagnostic and error informations are placed in the SSI frame.

Baud rate:

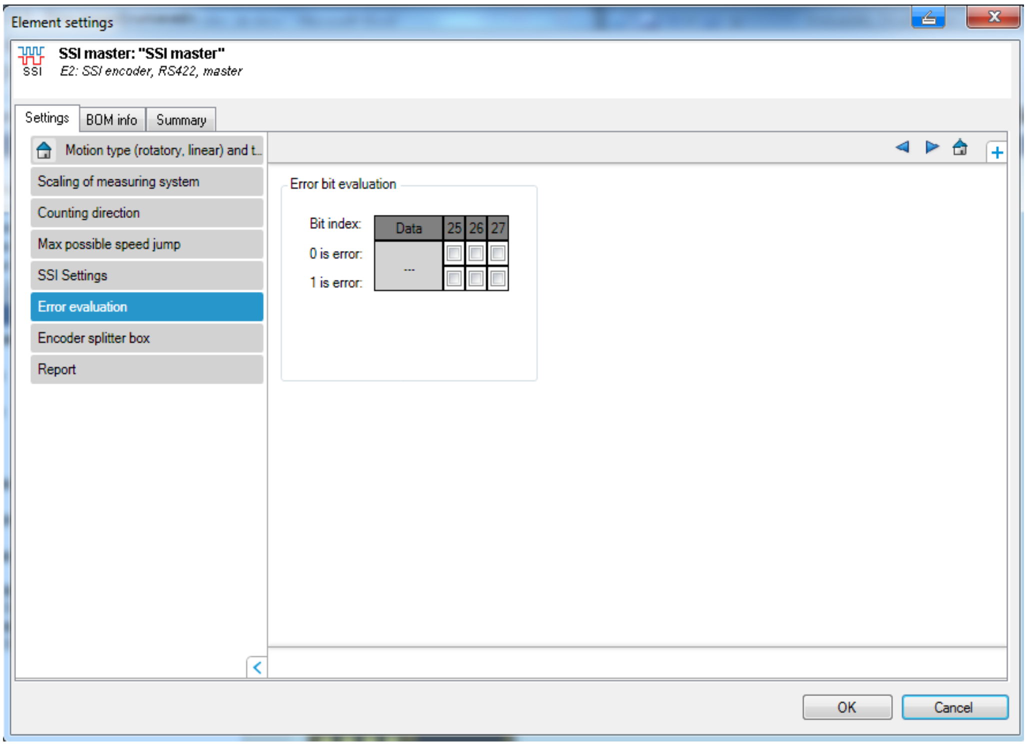
Frequency of the clock signal of the SSI interface.

Data Time out:

The maximum time in which valid position data are expected.

Error evaluation:

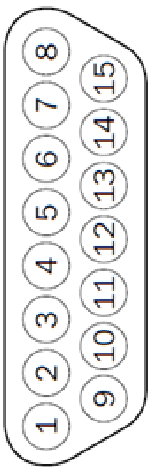
Defines where error bits and their status are placed when a fault is signaled by the sensor.



4 Connection of encoders

4.1 Encoder connection

On the front of the FX3-MOC0 there is a 15-pin micro-S-Sub socket for the connection of up to two encoders.

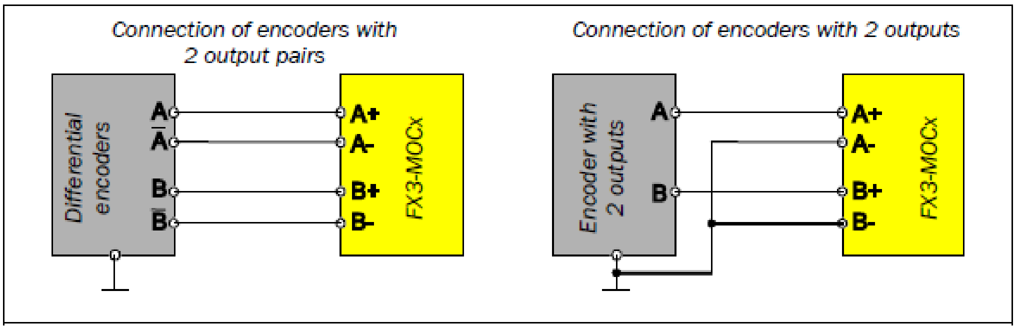
Plug	Pin	Signal	Description	Color coding
	1	ENC1_A+	Encoder 1, signal pair A, positive signal	White
	2	ENC1_B+	Encoder 1, signal pair B, positive signal	Green
	3	ENC1_C+	Encoder 1, signal pair C, positive signal	Grey
	4	ENC1_24V	24 V voltage supply for encoder 1/ hardware ID connection box encoder 2	Blue
	5	ENC2_24V	24 V voltage supply for encoder 2/ hardware ID connection box encoder 1	Red
	6	ENC2_C+	Encoder 2, signal pair C, positive signal	White/Green
	7	ENC2_B+	Encoder 2, signal pair B, positive signal	Grey/Pink
	8	ENC2_A+	Encoder 2, signal pair A, positive signal	Black
	9	ENC1_A-	Encoder 1, signal pair A, negative signal	Brown
	10	ENC1_B-	Encoder 1, signal pair B, negative signal	Yellow
	11	ENC1_C-	Encoder 1, signal pair C, negative signal	Pink
	12	ENC_0V	0 V voltage supply for encoder 1 and encoder 2	White/Yellow
	13	ENC2_C-	Encoder 2, signal pair C, negative signal	Brown/Green
	14	ENC2_B-	Encoder 2, signal pair B, negative signal	Red/Blue
	15	ENC2_A-	Encoder 2, signal pair A, negative signal	Purple

Connection of encoders

Depending on the encoder type, the signals are to be connected as per the following table:

Signal	**Incremental encoder, 2 output pairs (HTL 24 V, HTL 12 V, TTL)**	**Incremental encoder, 2 outputs (HTL 24 V, HTL 12 V, TTL)**	**Incremental encoder, 2 output pairs (RS-422)**	**Sin/Cos encoder**		**SSI encoder**
ENCx_C-	-		B-	-	-	Clock-
ENCx_C+	-		B+	-	-	Clock+
ENCx_B-	B-	0V	-	Sin_Ref	Sin-	-
ENCx_B+	B+	B	-	Sin	Sin+	-
ENCx_A-	A-	0V	A-	Cos_Ref	Cos-	Data Out-
ENCx_A+	A+	A	A+	Cos	Cos+	Data In+
ENC_0V	0 V supply for the encoders					
ENCx_24V	24 V supply for encoders, available on the FX3-MOC0 module					
Uout	On-board voltage output, can be switched between 5 V, 7 V, 12 V and 24 V, available on encoder connection boxes					

The following graphic shows the differences on the connection of incremental encoders with two output pairs and incremental encoders with two outputs.



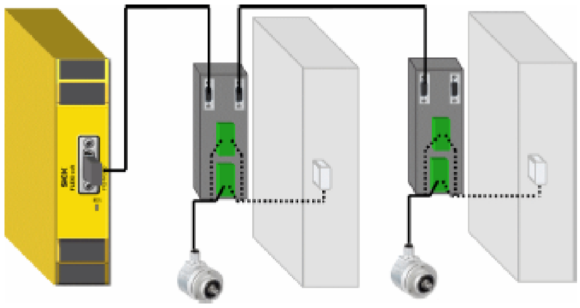
On encoders with 2 outputs the input A- and B- on the MOCx must not remain open, but must be connected to 0 V. The connection must be made as close as possible to the encoder's 0 V connection.

24 V are available on the encoder connection on the FX3-MOC0 module to supply the encoder. A selectable supply voltage is available on the connection boxes.

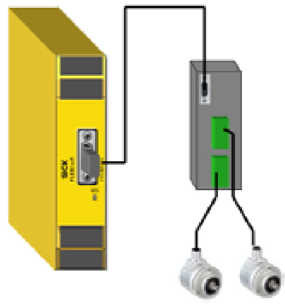
4.2 Connection boxes for encoders

The connection boxes ease the connection of encoders to the encoder interface on the FX3-MOC0 modules, in particular for encoders that are used for a FX3-MOC0 as well as for motor feedback in a drive system.

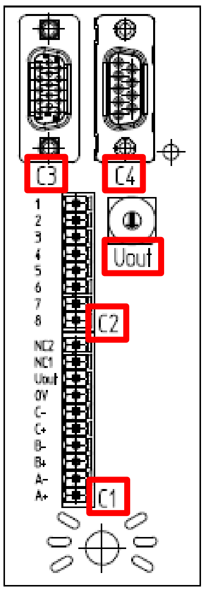
- **Motor feedback splitter box:** Facility for connecting one encoder. Is normally used in conjunction with a motor feedback encoder. Additional terminals are available for forwarding signals that are not required for the FX3-MOC0 module, but that can still be carried in the encoder cable, e.g. brake operation, temperature sensor etc.
- **Dual encoder connection box:** Facility for connecting two encoders.



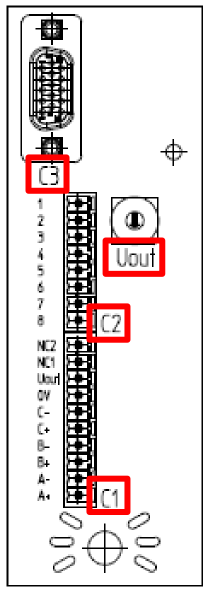
Connection principle with motor feedback splitter box



Connection principle with dual encoder connection box



Connections of the motor feedback splitter box



Connections of the dual encoder connection box

Description

<table>
<tr><th></th><th>Motor feedback splitter box</th><th>Dual encoder connection box</th></tr>
<tr><td>C1</td><td colspan="2">• Plug-in spring clamp terminals for the connection of encoder signals from an encoder
• 2 terminals for forwarding other signals</td></tr>
<tr><td>C2</td><td>• Plug-in spring clamp terminals with 8 terminals for forwarding other signals</td><td>• Plug-in spring clamp terminals for the connection of encoder signals from a further encoder</td></tr>
<tr><td>C3</td><td colspan="2">• 15-pin HD-D-Sub socket with M3 screws for the connection of the connection cable to the FX3-MOC0</td></tr>
<tr><td>C4</td><td>• 9-pin D-Sub socket with M3 screws for the connection of a second motor feedback splitter box (forwarding the ENC2_x signals from the 15-pin HD-D-Sub plug)</td><td>-</td></tr>
<tr><td>Uout</td><td colspan="2">• Selector switch for the on-board voltage supply for encoders, supplied from the FX3-MOC0</td></tr>
<tr><td></td><td colspan="2">• Screen terminals for both encoder cables (from the encoder and to the motor controller) for a low-impedance connection for both cable screens</td></tr>
<tr><td></td><td colspan="2">• ID in combination with the supply voltage for sampling by the FX3-MOC0</td></tr>
</table>

WARNING

The connection boxes for encoders must be mounted in an enviroment that corresponds to enclosure rating IP 54 (EN 60 529), e.g. in a control cabinet with the enclosure rating IP 54.

On-board voltage supply Uout

On-board voltage supply that can be used optionally for the encoder, supplied from the FX3-MOC0 (**C3**.ENC1_24V and **C3**.ENC2_24V). The output voltage Uout can be switched between 5 V, 7 V, 12 V and 24 V nominal using a rotary switch.

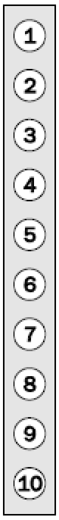
Switch position	Supply voltage Uout	Max. current	Notes
0	5 V	240 mA	Tolerance: 5 %
1	7 V	170 mA	
2	12 V	100 mA	
3	24 V nominal	75 mA	The supply voltage for the encoder can be up to 2 V below the supply voltage for the main module (terminal A1).

WARNING

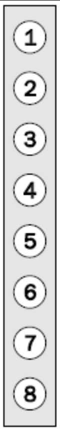
Only actuate the rotary switch for the supply voltage with the supply voltage switched off!

Actuate the rotary switch for the supply voltage on the connection box only if the supply voltage is switched off. Otherwise the encoder connected may be damaged by voltage spikes that occur on switching.

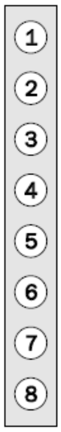
Pin assignment encoder connection C1

Terminal strip	Terminal	Signal	Description
	1	NC2	Not connected to the splitter box, is only used for forwarding a signal, e.g. for an external voltage supply (instead of using ENC_Uout)
	2	NC1	
	3	Uout	Encoder voltage supply from the on-board voltage supply in this motor feedback splitter box
	4	ENC_0V	Encoder 0 V voltage supply
	5	ENC_C-	Encoder channel C, negative signal
	6	ENC_C+	Encoder channel C, positive signal
	7	ENC_B-	Encoder channel B, negative signal
	8	ENC_B+	Encoder channel B, positive signal
	9	ENC_A-	Encoder channel A, negative signal
	10	ENC_A+	Encoder channel A, positive signal

Pin assignment of the connection terminals C2 on the motor feedback splitter box

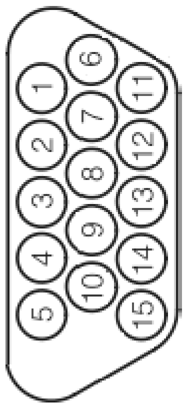
Terminals strip	Terminal	Signal	Description
	1	NC	Not connected to the splitter box, is only used for forwarding signals
	2	NC	
	3	NC	
	4	NC	
	5	NC	
	6	NC	
	7	NC	
	8	NC	

Pin assignment of the connection terminals C2 on the dual encoder connection box

Terminal strip	Terminal	Signal	Description
	1	Uout	Encoder voltage supply from the voltage regulator in this connection box
	2	ENC_0V	Encoder 0 V voltage supply
	3	ENC2_C-	Encoder 2, channel C, negative signal
	4	ENC2_C+	Encoder 2, channel C, positive signal
	5	ENC2_B-	Encoder 2, channel B, negative signal
	6	ENC2_B+	Encoder 2, channel B, positive signal
	7	ENC2_A-	Encoder 2, channel A, negative signal
	8	ENC2_A+	Encoder 2, channel A, positive signal

Pin assignment of the HD-D-Sub-15 socket C3 on the encoder connection boxes

Both connection boxes for encoders have a 15-pin HD-D-Sub socket for the connection to the FX3-MOC0.

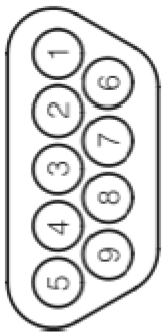
Socket	Pin	Signal	Description
	1	ENC1_A+	Encoder 1, signal pair A, positive signal
	2	ENC1_A-	Encoder 1, signal pair A, negative signal
	3	ENC1_24V	24 V voltage supply for encoder 1
	4	ENC2_A+	Encoder 2, signal pair A, positive signal
	5	ENC2_A-	Encoder 2, signal pair A, negative signal
	6	ENC1_B+	Encoder 1, signal pair B, positive signal
	7	ENC1_B-	Encoder 1, signal pair B, negative signal
	8	ENC_0V	0 V voltage supply for encoder 1 and encoder 2
	9	ENC2_B+	Encoder 2, signal pair B, positive signal
	10	ENC2_B-	Encoder 2, signal pair B, negative signal
	11	ENC1_C+	Encoder 1, signal pair C, positive signal
	12	ENC1_C-	Encoder 1, signal pair C, negative signal
	13	ENC2_24V	24 V voltage supply for encoder 2
	14	ENC2_C+	Encoder 2, signal pair C, positive signal
	15	ENC2_C-	Encoder 2, signal pair C, negative signal

Pin assignment of the D-Sub-9 socket C4 on the motor feedback splitter box

This motor feedback splitter box also has a 9-pin D-Sub socket for the connection of a second motor feedback splitter box.

Notes:

- The connection of a dual encoder connection box is not permitted here.
- A maximum of two motor feedback splitter boxes are allowed per MOCx module.

Socket	Pin	Signal	Description
	1	ENC_A+	Encoder channel A, positive signal
	2	ENC_B+	Encoder channel B, positive signal
	3	ENC_C+	Encoder channel C, positive signal
	4	ENC_24V	24 V voltage supply for encoder/ hardware ID connection box
	5	ENC_ID	Encoder splitter box ID
	6	ENC_A-	Encoder channel A, negative signal
	7	ENC_B-	Encoder channel B, negative signal
	8	ENC_C-	Encoder channel C, negative signal
	9	ENC_0V	0 V voltage supply for encoder

5 Data exchange CPU - MOC

5.1 Data exchange between main module and MOCx

As the main modules and the MOCx modules can process different data types and more complex signal pre-processing and logic can be programmed in the MOCx, the exchange of data between the modules must be organized. 18 bits can be sent to the MOCx from the main module and from the MOCx to the main module 16 bits. These bits must be routed in the logic editor. The name of the input + block + module is assigned to these bits as the tag name.

The data are exchanged via the internal FLEXBUS+ bus.

5.2 Data types in the MOCx logic-

The function blocks in the MOCx can process various data types. This feature differentiates them from the function blocks in the main module, which can only process Boolean values. Which data type is expected or output, depends on the related function block input or output used.

Boolean

Data of type Boolean are binary. They can only be either 1 or 0, low or high.

Motion

Data of type Motion contain all the information provided by an encoder. They comprise the following elements:

Element	Size	Internal value range (number of digits)	Resolution for rotary movement type	Resolution for linear movement type
Speed value	16 bit with sign	–32.767 to +32.767	1 digit = 0,5 rpm	1 digit = 1 mm/s
Speed status	1 bit	0 = invalid 1 = valid		
Relative position value	32 bit with sign	–2.147.483.647 to +2.147.483.647	1 digit = 1/30.000 rev	1 digit = 1/250 mm
Relative position status	1 bit	0 = invalid 1 = valid		
Absolute position value	32 bit with sign	–2.147.483.647 to +2.147.483.647	1 digit = 1/30.000 rev	1 digit = 1/250 mm
Absolute position status	1 bit	0 = invalid 1 = valid		

Internal resolution of the speed and position information

The smallest unit of speed information and position information acquired is defined by the internal resolution of these data. It can also be further limited by the resolution of the encoder system.

Type of measurement system	Speed information	Position information
Rotary movement type	0,5 rpm	1/30.000 rev
Linear movement type	1 mm/s	1/250 mm

UI8

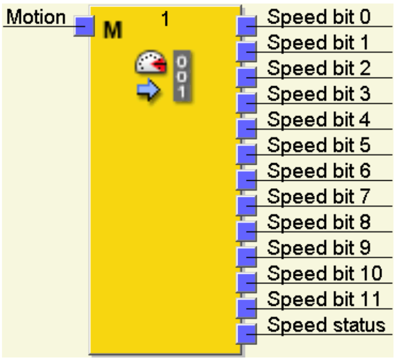
Data of type UI8 make it possible, e.g., to select or display a speed or position range.

Element	Size	Values for speed ID
UI8	8 bit	• 0 = invalid • 1-31 = range index

Inputs and outputs that expect or output data types other than Boolean are correspondingly marked in the function block symbols. Here M stands for Motion data and UI8 for Unsigned integer 8-bit.

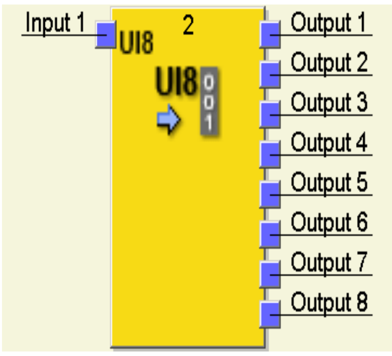
5.3 Function blocks for data conversion

5.3.1 Speed to laser scanner



The function block **"Speed to laser scanner"** converts the speed on the "Motion" input into a Boolean value with the scaling cm/s. For this purpose the 12 outputs "Speed bit 0" to "Speed bit 11" are available as well as the Speed status output. Each of the "Speed bit x" outputs contains the corresponding bit of the output value calculated for the speed.
This value can, e.g., be output to a SICK laser scanner connected via EFI and used by this scanner for speed-based monitoring case switching.

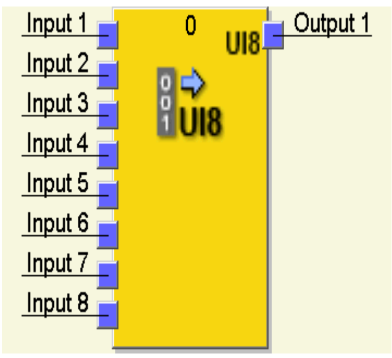
5.3.2 UI8 to Boolean converter



The function block **"UI8 to Boolean converter"** converts an 8-bit integer value (UINT8) on Input 1 to Boolean. Output 1 to Output 8 output the decoded value as Boolean. This function is a pure data type conversion so that connection to Boolean signals is possible.

Example: It is used to transmit the *Speed status ID* from the Drive Monitor to the CPU.

5.3.3 Boolean to UI8 converter



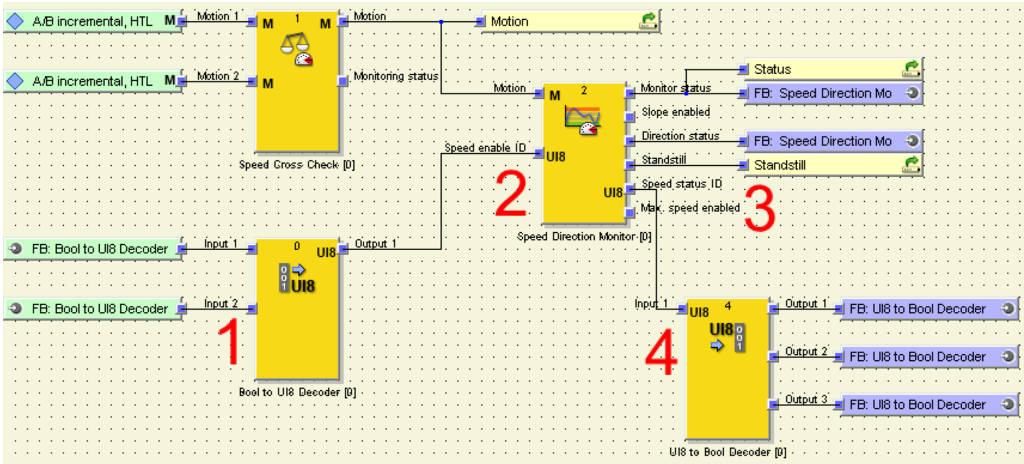
The function block **"Boolean to UI8 converter"** converts an 8-bit Boolean value on Input 1 to Input 8 to an integer value (UINT8). The decoded value is output as an integer on Output 1. This function is a pure data type conversion so that connection to Boolean signals is possible.

Example: It is used to transmit the *Speed enable ID* from the CPU to the Drive Monitor.

5.3.4 Example

As the main modules and the MOCx modules can process different data types and more complex signal pre-processing and logic can be programmed in the MOCx, the exchange of data between the modules must be organized. 18 bits can be sent to the MOCx from the main module, from the MOCx to the main module 16 bits. These bits must be routed in the logic editor. The name of the input + block + module is assigned to these bits as the tag name.

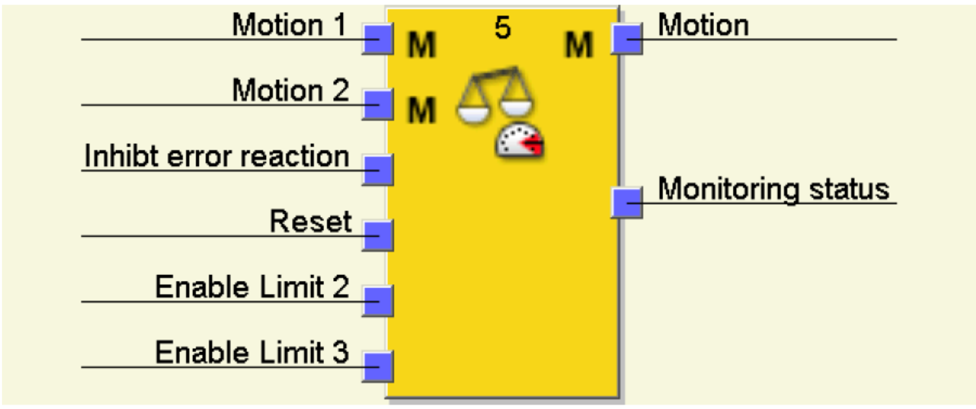
The data are exchanged via the internal FLEXBUS+ bus.



1. At first the binary data from the CPU is converted to *Speed enable ID* (UI8 data). Thus the data can be used in the MOC logic.
2. The *Speed enable ID* activates the maximum speed which is currently allowed.
3. The *Speed status ID* indicates the current speed range.
4. Finally the *Speed status ID* (UI8 data) from the MOC logic is converted to binary data. Thus the data about the speed can be used in the CPU logic.
   The information receives the CPU via the backplane bus.

6 Function blocks for monitoring functions

6.1 Speed Cross Check



General description:

The function block **"Speed cross check"** compares speed values from two different signal sources. The checks made during this process are used to achieve a higher level of safety, particularly if encoders that are not safe are used.
Due to, e.g. slip, friction, mechanical coupling behavior etc., continuous and also temporary differences between the two measured values may occur. For this reason the function block provides various parameters that can be used to compensate for such differences. In this way erroneous switching can be prevented and the availability of the machine safeguarded.

Inputs:

Motion In:
Data of type **"Motion"** contain all the information provided by an encoder.

Inhibit error reaction:
Using the optional input **"Inhibit error reaction"** it's possible to control if an error (monitoring status = Low) also simultaneously triggers an error reaction. In this case the output "Motion" is set to 0 and invalid. If safety is ensured independent of the monitored movement, e.g. by means of a closed guard, then this error reaction can be inhibited. If the input **"Inhibit error reaction"** is high, an error (Monitoring status = Low) does not cause that the "Motion" output to be set to 0.

Reset
Activation of the optional input **"Reset"** for error reset using an external signal.

Enable Limit 2 and 3:
The inputs **"Enable limit 2"** and **"Enable limit 3"** are optional. When these inputs are used and a high signal is present, the transgression of the value of Limit 1, relative speed difference is allowed. Both inputs are limited in time and can be set via the Max time limit 2 and 3.

Outputs:

Motion:

The parameter “Speed output mode” defines which value is to be output at the **“Motion”** output. The following settings are possible:

- Motion 1
- Higher speed of Motion 1 or Motion 2
- Mean speed of Motion 1 or Motion 2

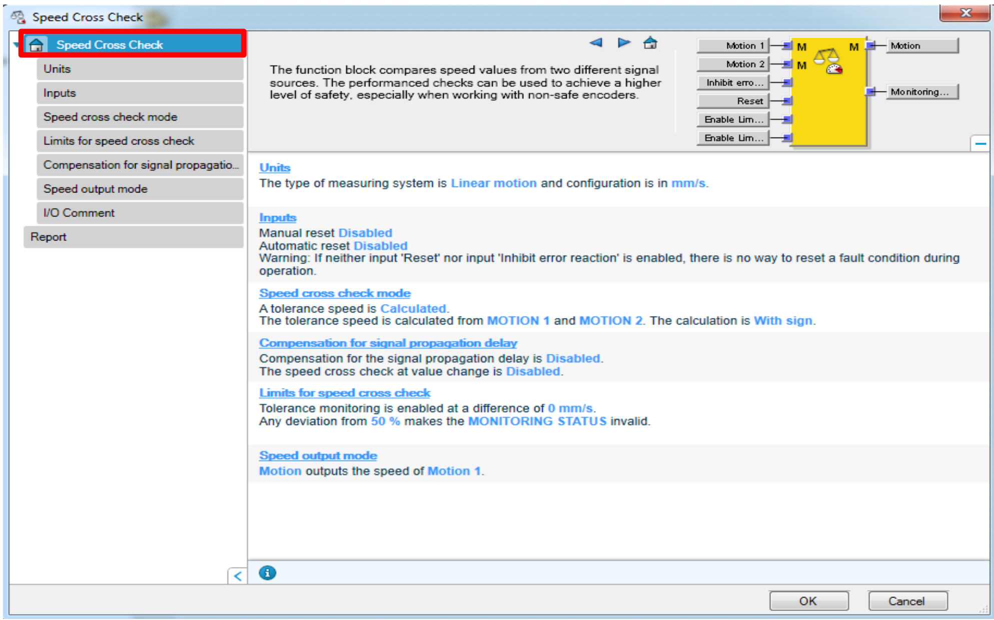
Monitoring status:

The output **“Monitoring status”** goes low if the allowed relative speed difference is exceeded. This happens independent from the state of the input “Inhibit error reaction”. The output **“Monitoring status”** goes high again when the error has been reset. There are two possible ways of resetting the error:

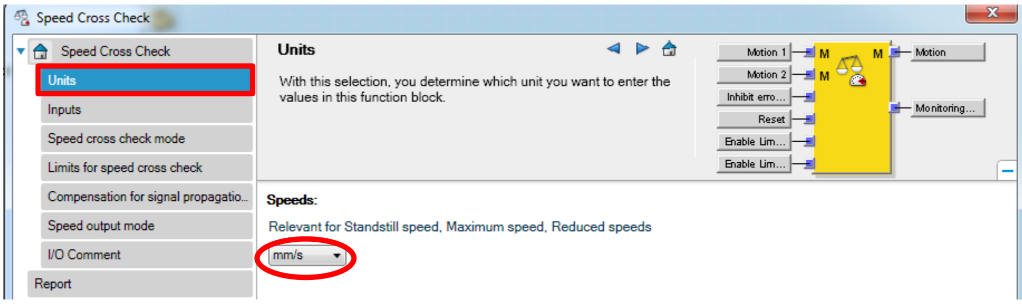
- Manual Reset: An error is reset by a rising edge on the optional input “Reset”, if the averaged speed nearly was zero for a period of about 1 s and the relative speed difference is lower than Limit 1.

- Automatic reset: An error is reset if the optional input “Inhibit error reaction” is high and the speed difference drops below the permissible relative speed difference before the input “Inhibit error reaction” changes from high to low. The option “Automatic error reset is dependent on the absolute speed” defines if the absolute speed is considered.

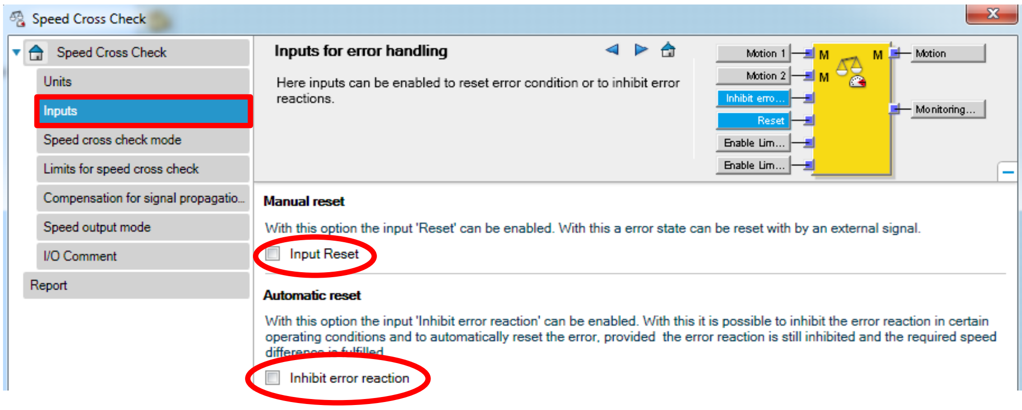
With double click on the function block, the context menu opens.



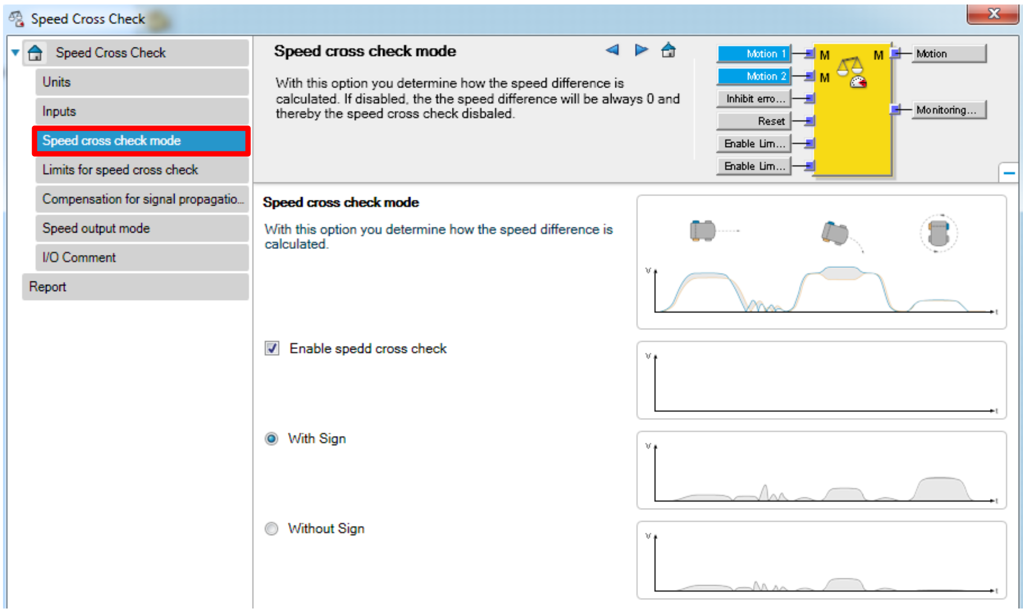
This function block compares speed values from two different signal sources.



On the tab “Units” can be set, which units to be used e.g. for calculation of speeds (mm/s, km/h, rpm, etc.).



Additional inputs can be enabled to reset a error condition or to inhibit a error reaction.



With the “speed cross check mode" can be defined, how the speed difference has to be calculated.
The configuration mainly depends on the application.
In the following part are some recommendations for the configuration.

1. Stationary machines with fixed mechanical coupling
   - With sign

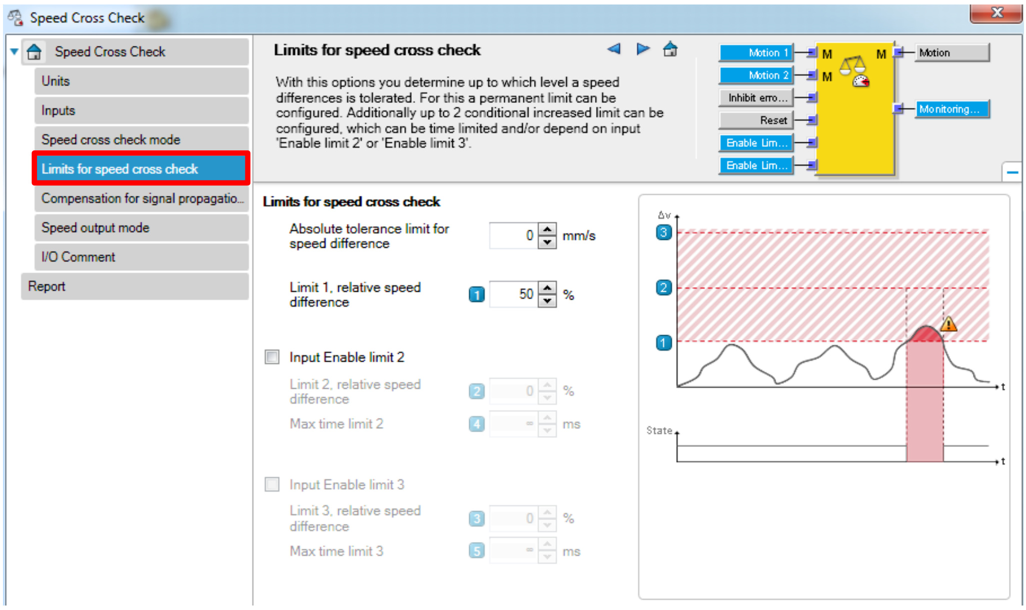
2. Stationary machines with elastic mechanical coupling
   - Without sign

3. Stationary machines with fixed mechanical coupling and SSI linear measurement sensors

   Notice: Typically linear measurement systems with SSI interface have high resolution but low repeatability precision, which means the position is jumping from cycle to cycle. This causes discontinuities in speed and large speed deltas even, when the position jump is only a little absolute value. To compensate that it is better to compare without sign.

   - Without sign

4. AGV – Enable/Turning the position
   - Without sign

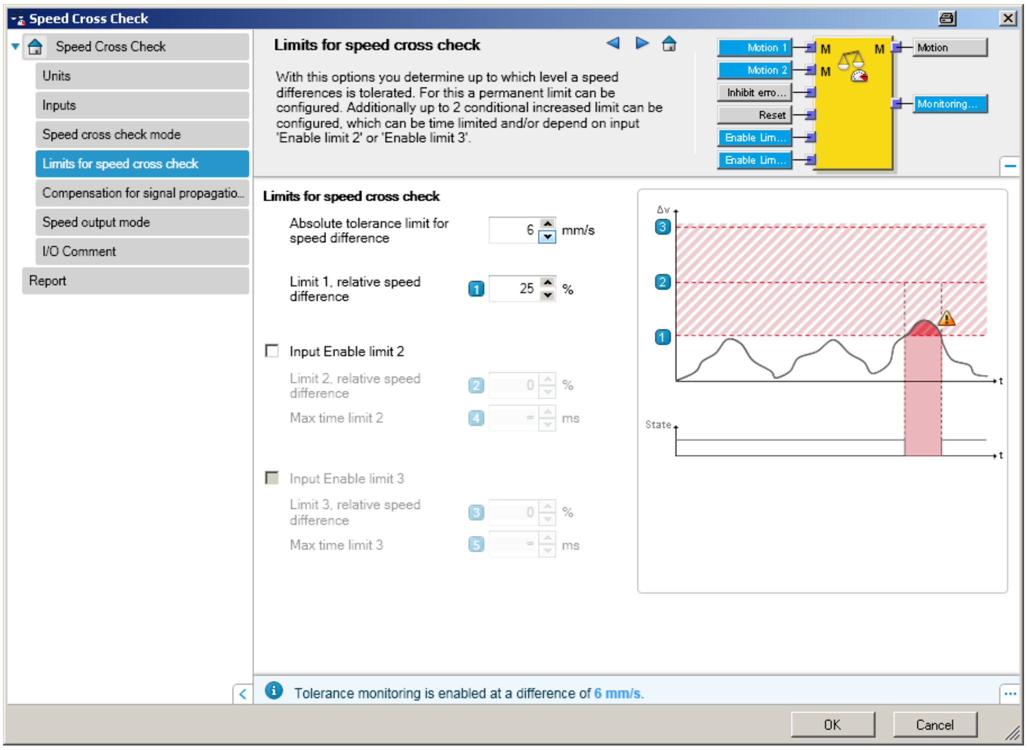


In which height a speed difference should be tolerated, has to be set in the tab "Limits for speed cross check".

Different applications, with a possible selection, are described in the following part.

1. Stationary machines with fixed mechanical coupling

Typically the coupling is good and there are nearly no differences between the encoders.

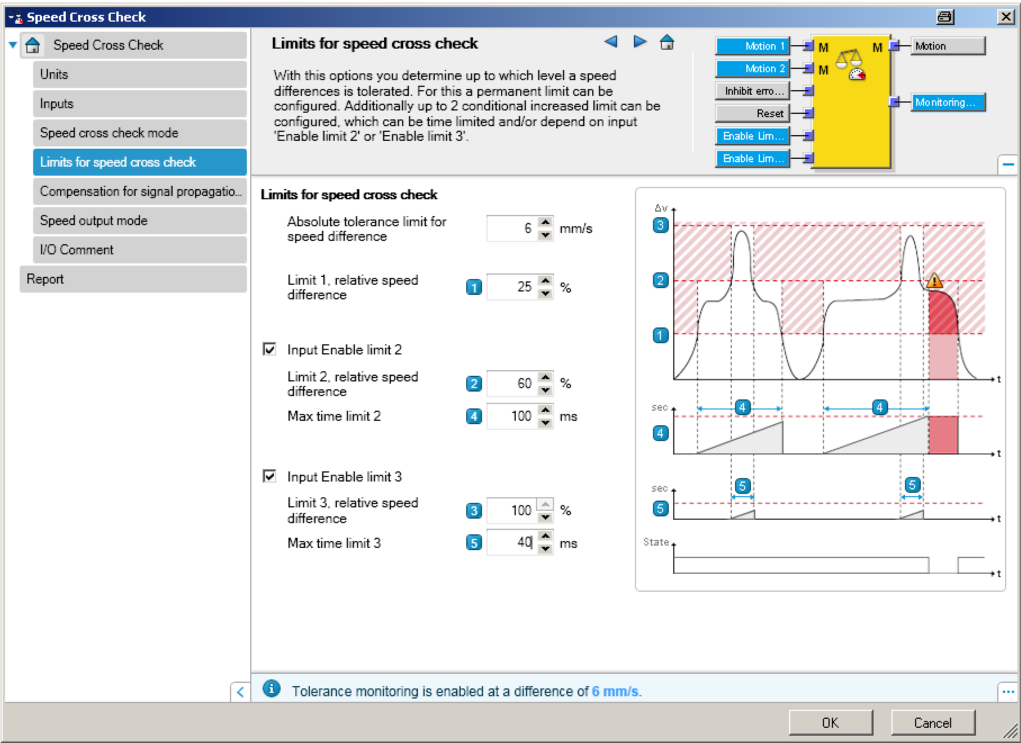


2. Stationary machines with elastic mechanical coupling

Typically in this type of machines, one sensor will react faster than the other. In worst case one sensor accelerates in positive direction while the other accelerates in negative because of bad coupling or mechanical distortion. During the acceleration and direction change of the drive, the sensors need time to accelerate both in the same direction. The configuration below can be a start point to optimize the system.

WARNING

It must be ensured by a risk analysis, that with activation of the limits 2 and 3 possible errors at the sensors are detected sufficiently rapidly and do not cause hazards.



The inputs “Input Enable limit x” can be connected to a static 1.

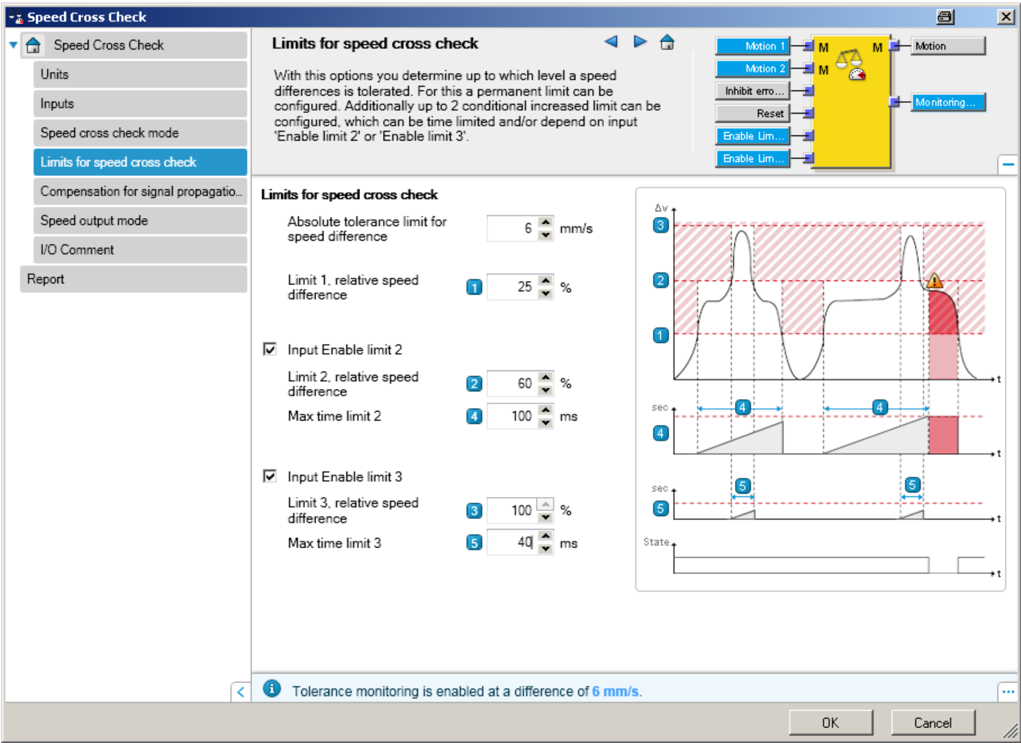
3. Stationary machines with fixed mechanical coupling and SSI linear measurement sensors

Typically linear measurement systems with SSI interface have high resolution but low repeatability precision, which means that the position is jumping from cycle to cycle. This leads to disruption of the speed and even to large differences in speed, when the jump position is only a small absolute value. To compensate this it is better if no sign is used.

During the acceleration and direction change of the drive, the sensors need time to accelerate both in the same direction.
The configuration below can be a start point to optimize the system.

WARNING

It must be ensured by a risk analysis, that with activation of the limits 2 and 3 possible errors at the sensors are detected sufficiently rapidly and do not cause hazards.



The inputs “Input Enable limit x” can be connected to a static 1.

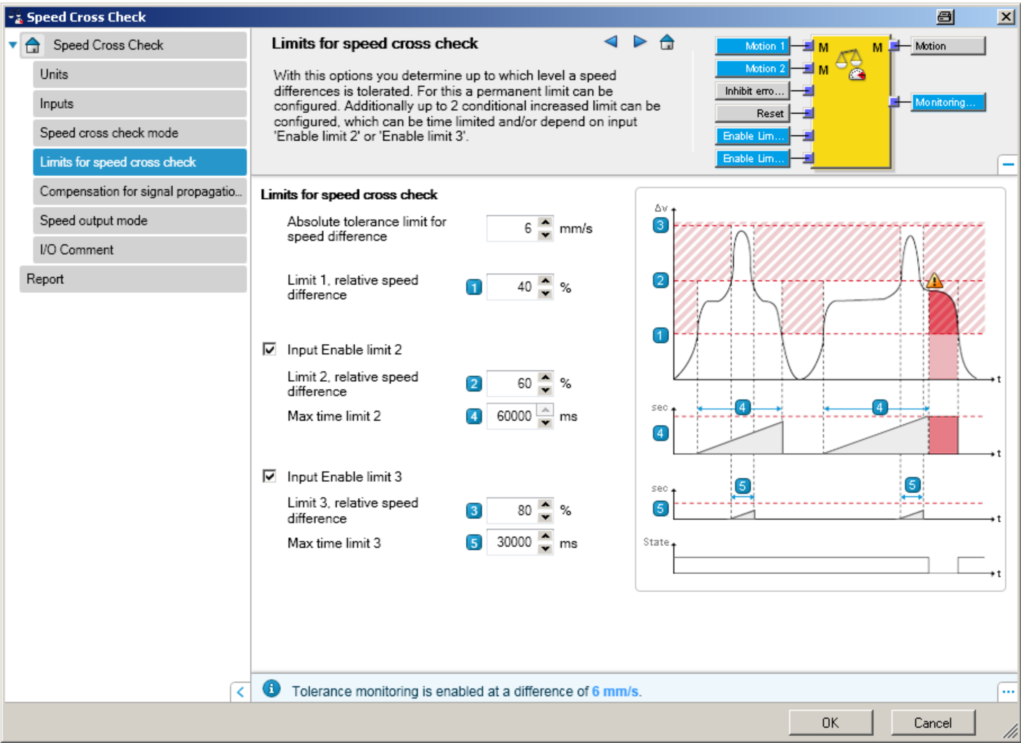
4. AGV – Enable/Turning the position

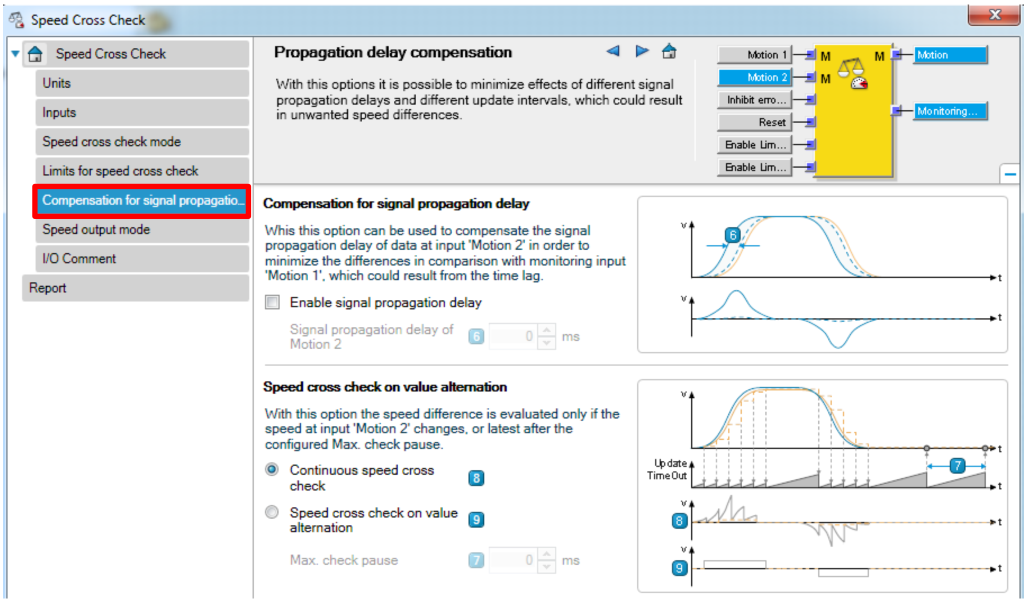
When AGVs travel curves the speed information from the encoders can be different for a longer period of time. This can be configured with “Limit 2, relative speed difference” and “Limit 3, relative speed difference”.
Limit 2 is the parameter for curves with large radius and Limit 3 for small radius.

The inputs “Input Enable limit x” supposed to be activated by a PLC vehicle via the Flexi Soft CPU. In addition, they can also be connected to a static 1.

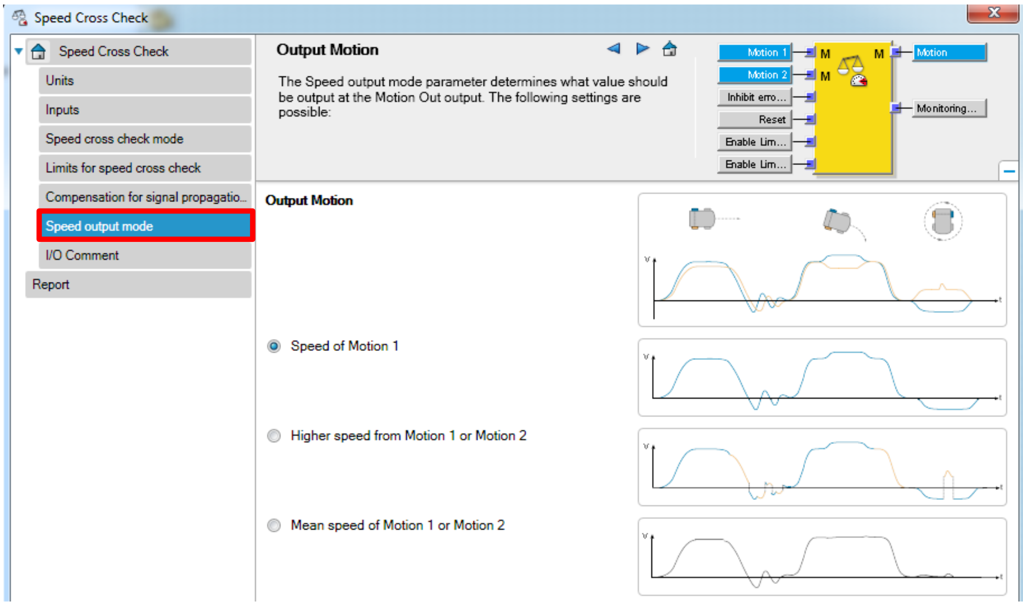
WARNING

It must be ensured by a risk analysis, that with activation of the limits 2 and 3 possible occuring errors at the sensors are detected sufficiently rapidly and do not cause hazards.

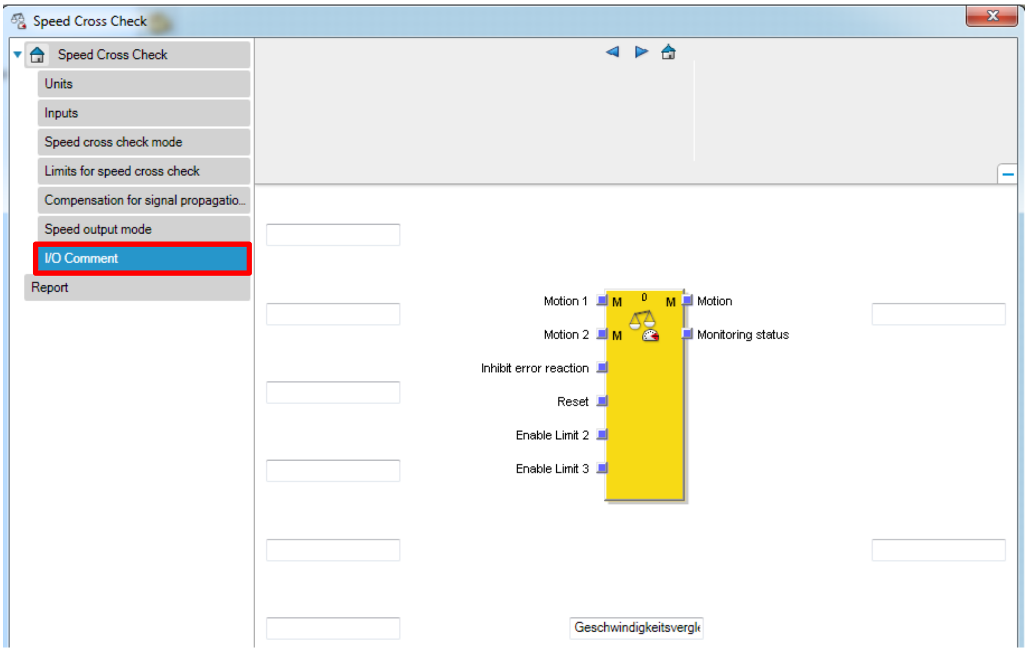




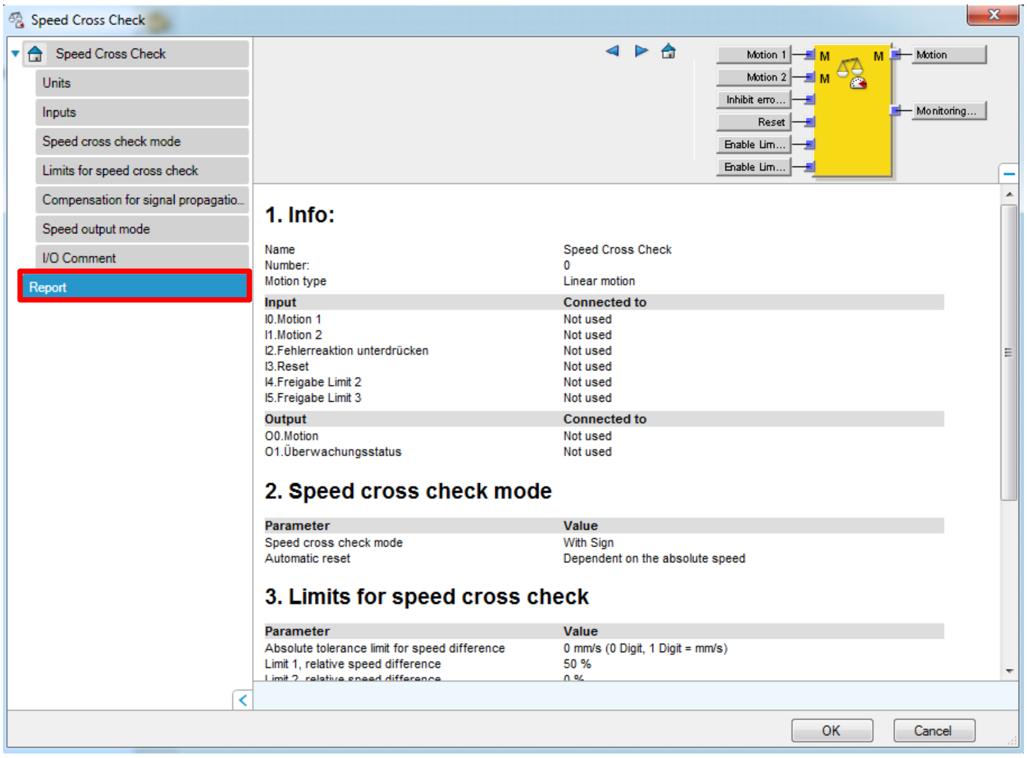
In the tab "Compensation for signal propagation ", effects can be minimized, which may occur because of different signal propagation delays and different update rates.



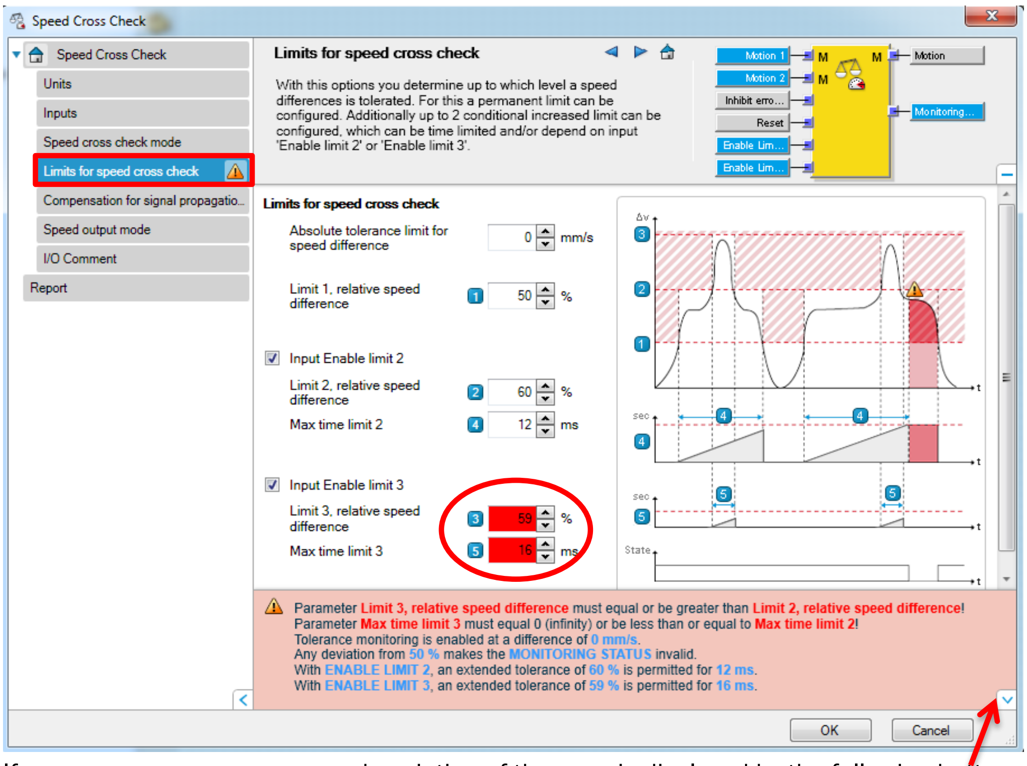
Among "Speed output mode" it is possible to determine what value should be output at the output “Motion”.



The tab "I/O comment" allows it to replace the pre-defined names for the function block inputs and outputs by own identifiers and to add a name or descriptive text to the function block. This name or text is displayed in the logic editor under the function block.

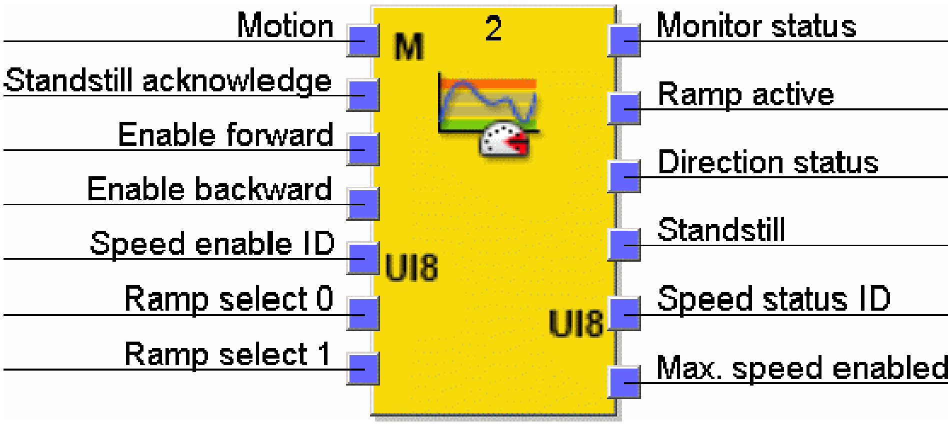


The tab "report" shows a summary of the configuration of the function block, including all input and output links and the configured parameter.



If an error message appears, a description of the error is displayed by the following button, which helps to fix the error.

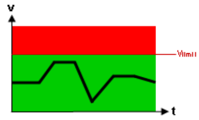
6.2 Speed Monitor



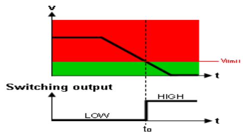
General description:

The function block **"Speed monitor"** is the central building block for all speed and direction monitoring in an application. In principle this function block can undertake the following functions:

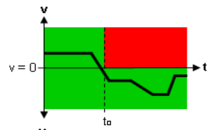
- Safely limited speed SLS



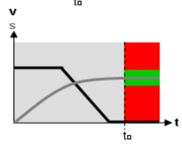
- Safe speed monitoring SSM



- Safe Direction SDI



- Safe operating stop SOS



In addition the function block can monitor the ramps on the transition from one monitored speed to the next. For this purpose up to four different ramps can be activated via the inputs "Ramp selection 0" and "Ramp selection 1".

Inputs:

Motion In:
Data of type **"Motion"** contains all the information provided by an encoder.

Standstill acknowledge:
Using the optional input **"Standstill acknowledge",** the internal standstill detection can be deactivated.

Enable forward & backward:
With the aid of the optional **"Enable forward"** and **"Enable backward"** inputs the permitted movement direction can be enabled.

Speed enable ID:
The permissible speed range is selected using the optional input **“Speed enable ID”**.
If the current speed on the input “Motion” is higher than the selected speed range, the output “Monitor status” goes low.

Ramp select:
It’s possible to define up to four different step widths for a speed ramp and select between these widths with the aid of the inputs **“Ramp selection 1”** and **“Ramp selection 0”**.

Input values		Selected step width
Ramp selection 1	**Ramp selection 0**	
0	**0**	Ramp speed transition 1 (fastest ramp)
0	**1**	Ramp speed transition 2
1	**0**	Ramp speed transition 3
1	**1**	Ramp speed transition 4 (slowest ramp)

Outputs:

Monitor status:
The output **“Monitor status”** is high in the normal case. It goes low if one of the following monitoring functions has the result 0:

- monitoring of the reduced speed selected via the input “Speed enable ID”
- monitoring of the maximum speed
- direction monitoring

Ramp active:
While a speed ramp is active, the output **“Ramp active”** is high.

Direction status:
The output **“Direction status”** indicates the direction:
0 = forward (positive velocity)
1 = reverse (negative velocity)

Standstill:
Standstill condition met:

- The Motion Speed is lower than the standstill speed for at least the duration of the standstill acceptance time
  or
- The standstill position window is not exceeded.

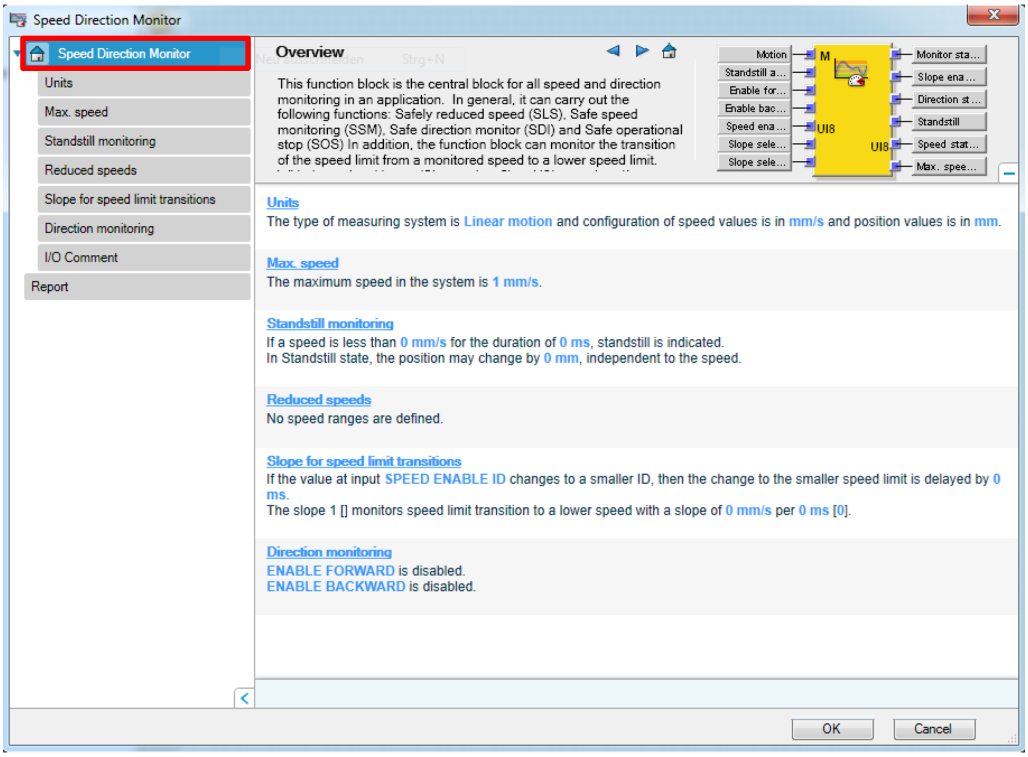
Speed status ID:
Independent of which speed range is currently active for monitoring, the current speed range with which the drive is currently operating is output via the output **“Speed status ID”.** The data are output as a UI8 value.

<u>Max speed enabled:</u>

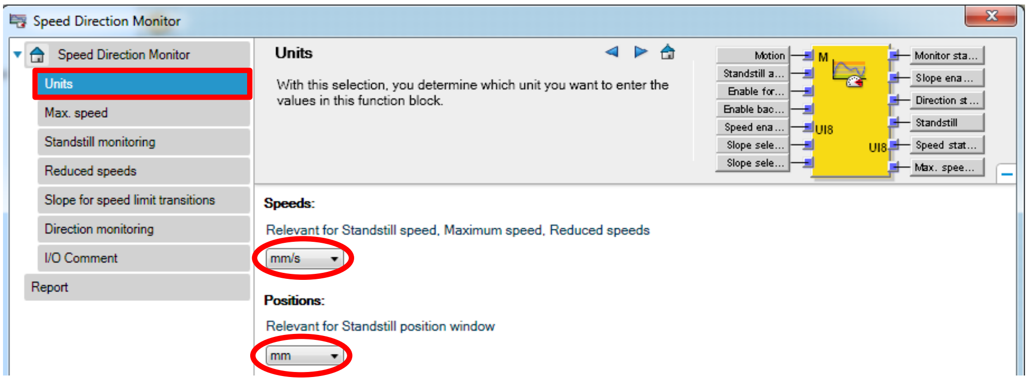
The maximum speed which may occur in the system is set with this value.

The output **"Max. speed enabled"** is high if the highest speed range used is activated by the "Speed enable ID" input. This output can be used as a reset condition for a subsequent MOCx function block "Safe Stop" on the "Stop 2 reset" input.

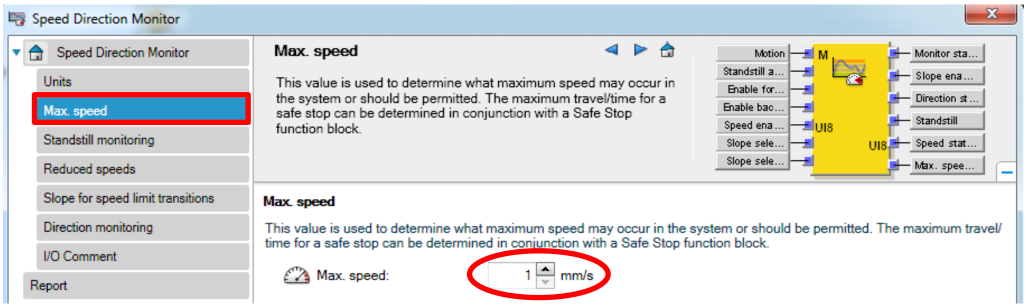
With double click on the function block, the context menu opens.



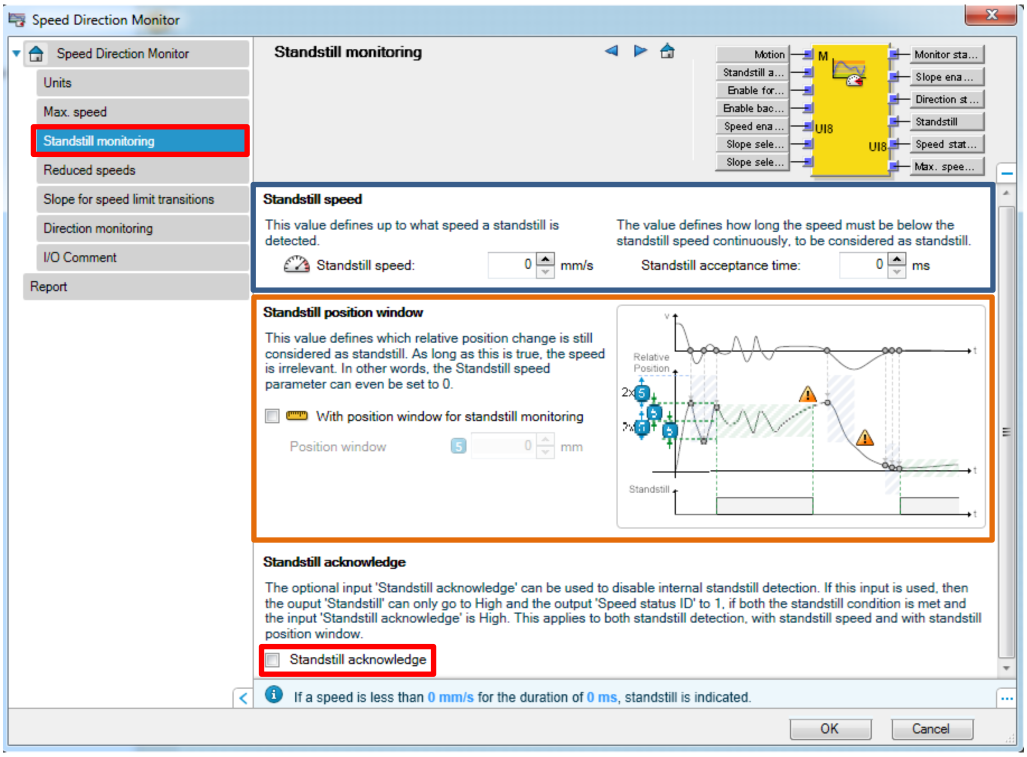
This is the central block for all speed and direction monitoring function in an application.



On the tab "Units" can be set, which units to be used e.g. for calculation of speeds (mm/s, km/h, rpm, etc.).



With "max speed" can be set the maximum speed, which is permitted.
This function is crucial for the stopping distance, if the safety functions SS1 and SS2 are used in the application.



The Standstill speed, the Standstill position window and the Standstill acknowledge can be set across "Standstill monitoring".

Standstill detection with Standstill speed:

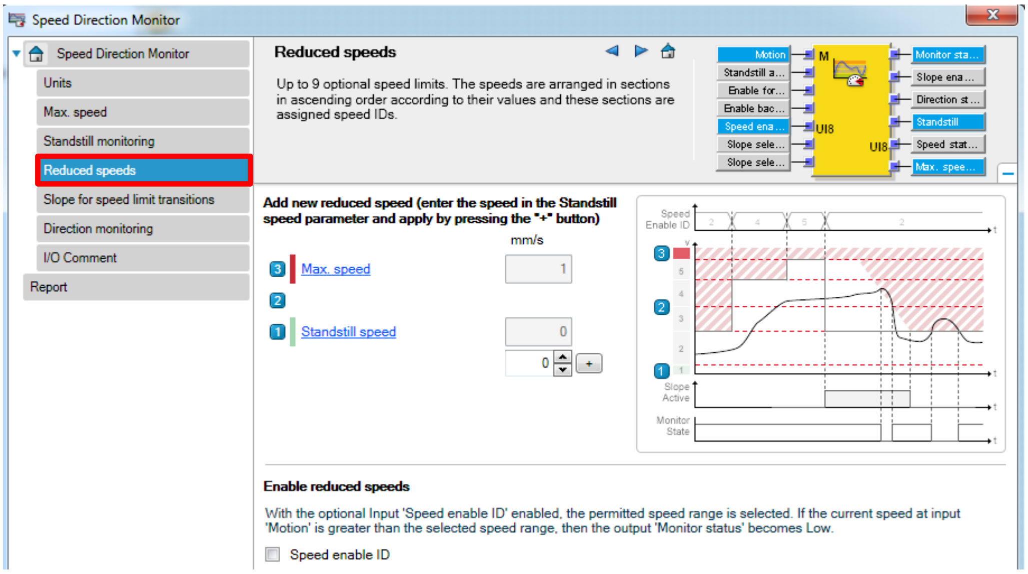
- The Motion Speed is lower than the Standstill speed for at least the duration of the Standstill acceptance time.

In addition to the Standstill speed a Standstill position window can be defined.
This value defines which relative position change still may be considered as standstill.
The position monitoring at standstill can be started by two different conditions:

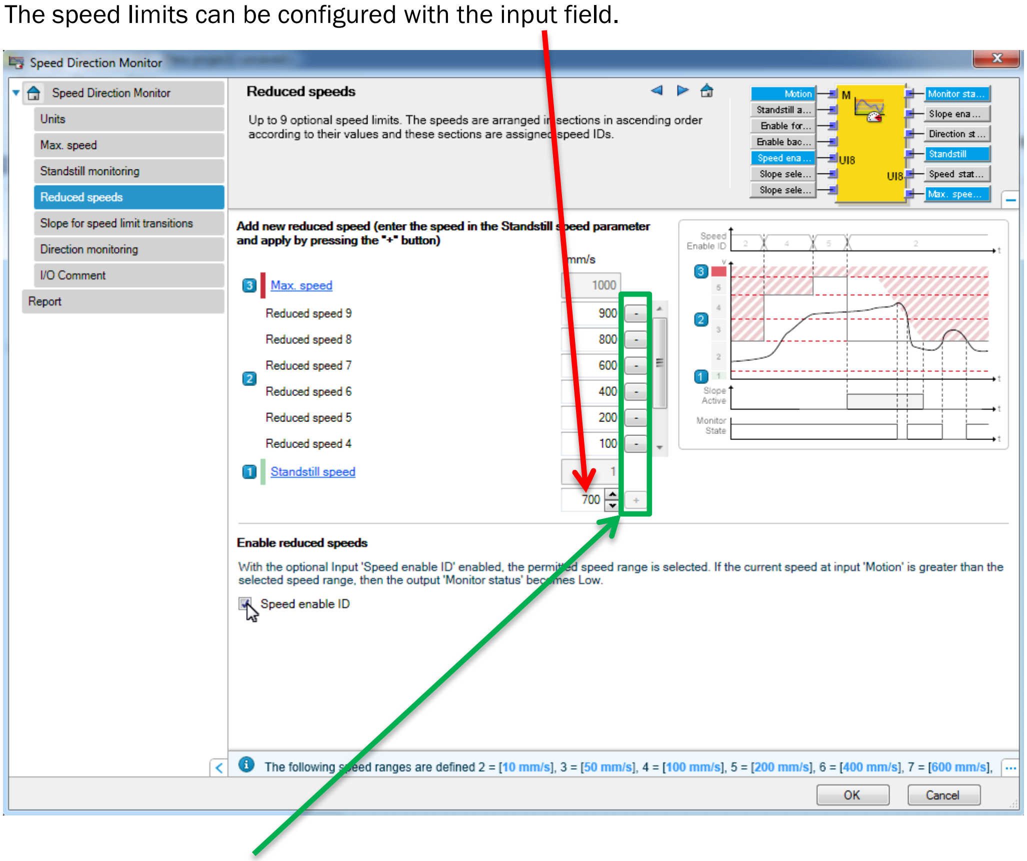
- The speed was zero for three consecutive logic execution cycles,

or

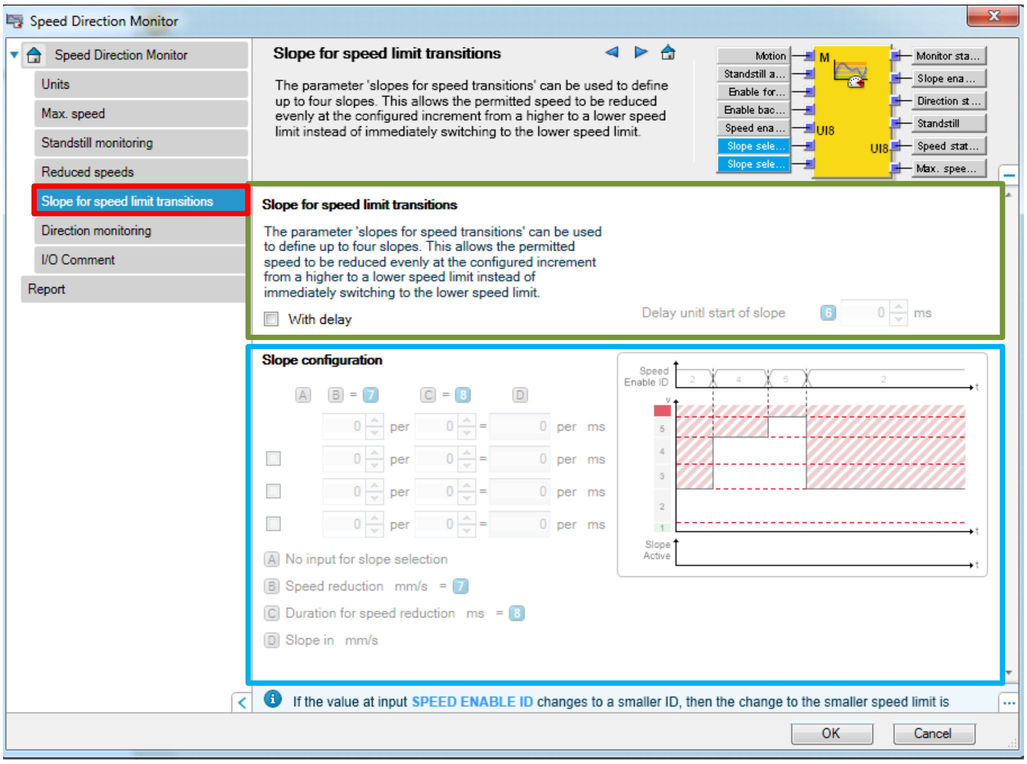
- The speed has reached the value zero three times and the relative position difference during this process was less than 2 × the value for the Standstill position window.



Across the tab "Reduced speed" can be set up to nine optional speed limits.
The speed limits can be configured with the input field.



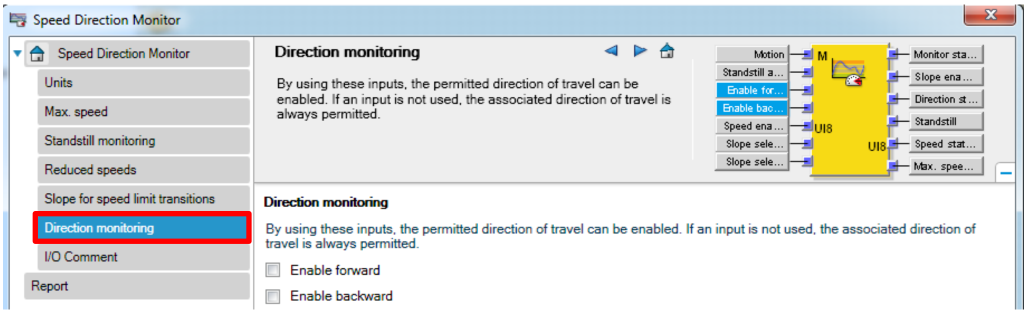
With the plus or minus symbol it is possible to configure new speed limits or to delete old ones.
The configured speeds are assigned to speed-IDs. These speed-IDs are arranged ascending like the speed values, i.e. the highest speed always has the highest ID.
The speed IDs can be used to activate a maximum speed via the input “Speed enable ID” (from the CPU) or to transfer the current speed range over the output “Speed status ID” to the CPU.



At “Slope for speed limit transitions” may be defined up to four slopes, which are reduced from a higher to a lower speed limit.

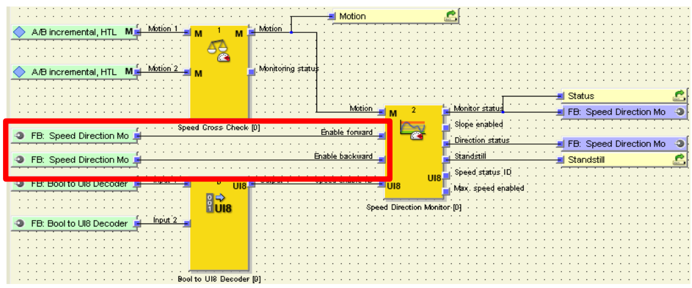
Time, in which the function block still does not expect a reaction from the system, i.e. no brake ramp.

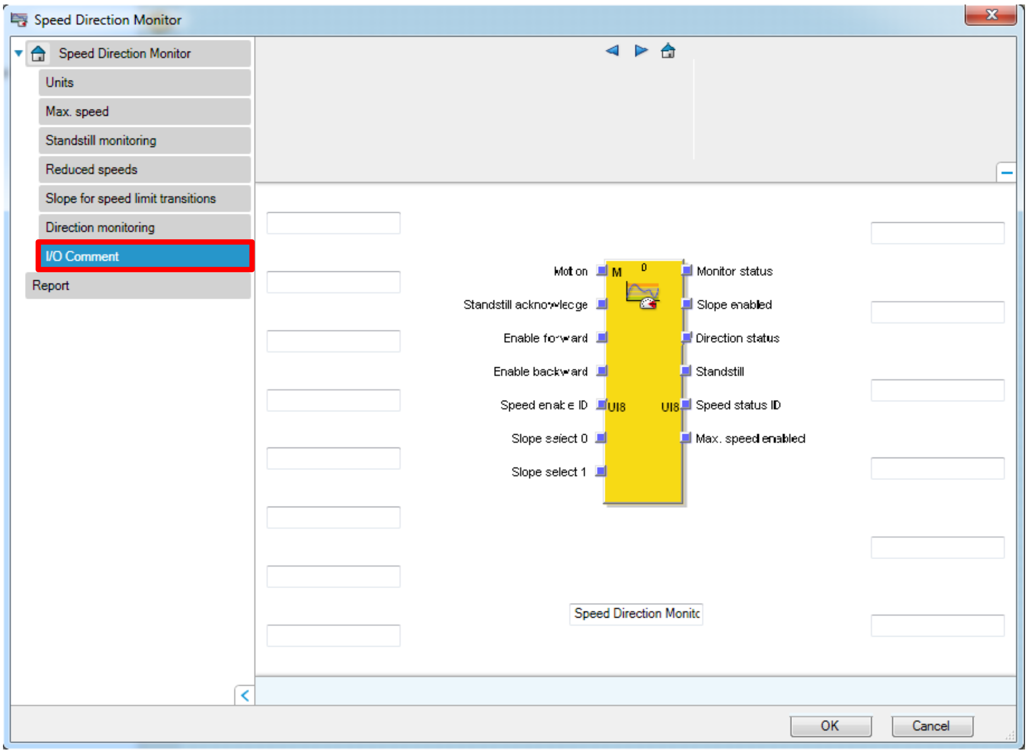
This value defines which step width to be used for speed reduction on changing from a higher active speed enable ID to a lower ID.



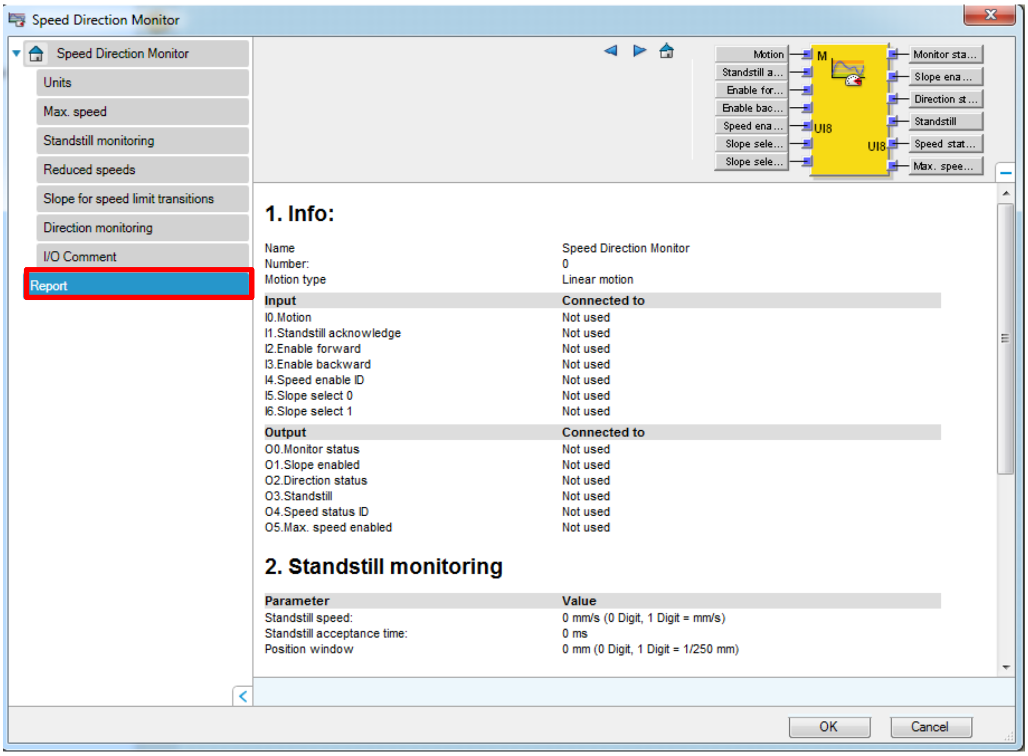
“Enable forward” and “Enable backward” are optional inputs for the explicit enable of the movement direction. If an input is not used, the associated direction of travel is always permitted. The high signal releases the movement direction.

Just as in the following example, the two inputs could be controlled by the CPU

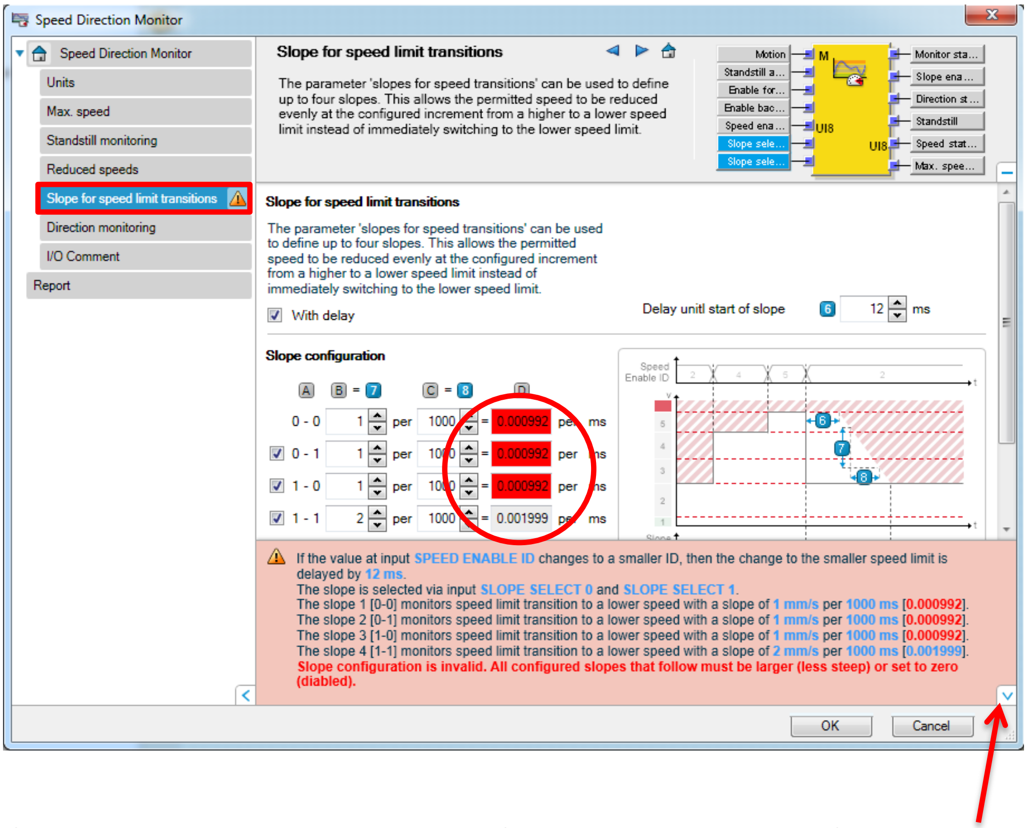




The tab "I/O comment" allows it to replace the pre-defined names for the function block inputs and outputs by own identifiers and to add a name or descriptive text to the function block. This name or text is displayed in the logic editor under the function block.

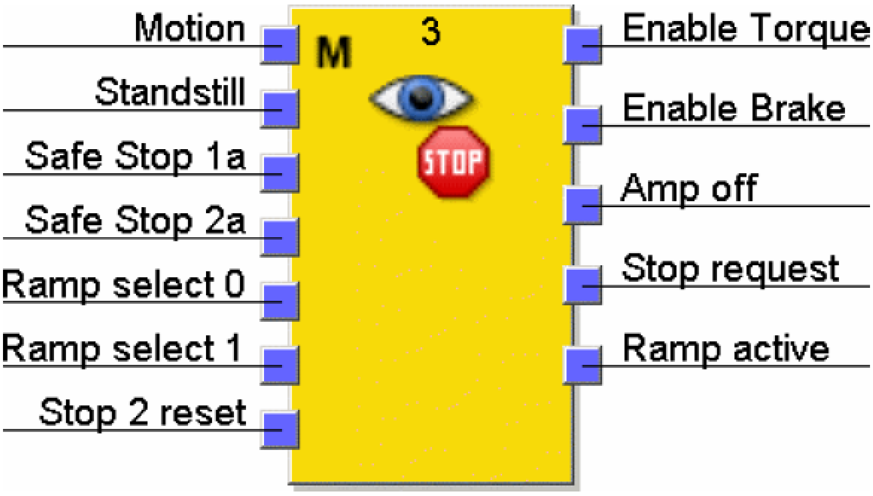


The tab "report" shows a summary of the configuration of the function block, including all input and output links and the configured parameter.



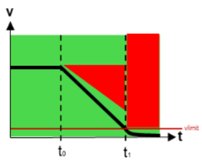
If an error message appears, a description of the error is displayed by the following button, which helps to fix the error.

6.3 Safe Stop

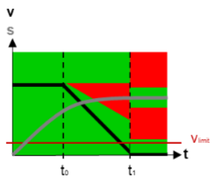


General description:

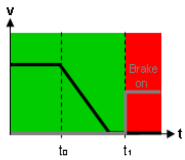
The MOCx function block **"Safe Stop"** is used to trigger and monitor the safe stop of a drive system. During this process the drive must be shut down in a controlled manner. In this way the brake torque of the drive can be used to bring the drive to a standstill in a shorter time that would be possible with an uncontrolled stop.
As the stop ramp for a drive system is normally not safe, the MOCx function block **"Safe Stop"** monitors the actual reduction in the speed down to standstill.



SS1
Safe stop 1



SS2
Safe stop2



SBC
Safe brake control

Inputs:

Motion In:
Data of type **"Motion"** contain all the information provided by an encoder.

Standstill:
If, during the standstill monitoring after a Safe Stop 1 or after a Safe Stop 2 the input **"Standstill"** goes low sometime (i.e. the standstill condition is not met or no longer met), then the output "Amp off request" and the outputs "Enable brake" and "Enable torque" are switched off with the delay configured.

Safe Stop 1a:
SS1 induces the brake operation of the drive via an emergency stop ramp at the point t_0. This break operation is monitored from the drive monitoring device. When the current speed is below the standstill limit, then the energy supply of the drive will be disconnected (STO).

Safe Stop 2a:
The **SS2** function starts the brake operation via an emergency stop declaration at t_0. This brake operation will be monitored by the drive monitoring device. When the speed is below the configured standstill limit, then the SOS function will be used to remain the drive/ rotor in the position where the function was activated.

Ramp select:
Up to four several stop ramps are configurable. Between these can be switched via the optional inputs **“Ramp selection 1”** and **“Ramp selection 2”**.

Input values		Selected step width
Ramp selection 1	**Ramp selection 0**	
0	**0**	Ramp slope 1 (fastest ramp)
0	**1**	Ramp slope 2
1	**0**	Ramp slope 3
1	**1**	Ramp slope 4 (slowest ramp)

Stop 2 reset:
Safe Stop 2 can be reset by a rising edge on the optional input **“Stop 2 reset”** after standstill has been detected.

Outputs:

The drive system can be controlled by the outputs of the function block.

Enable Torque:
Safety signal, which switches off the drive system’s torque (STO).

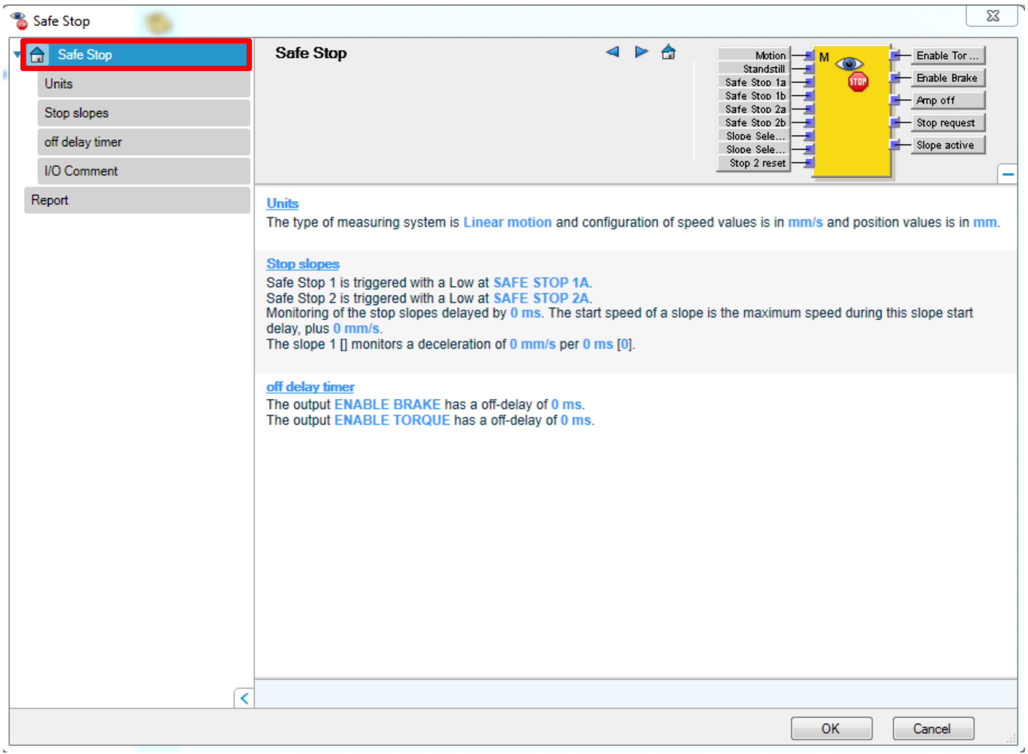
Enable Break:
Safety signal, that switches off the power supply to the mechanical brake if it’s necessary.

Amp off:
Not safe signal, which triggers the shut down of the amplifier, the drive’s torque and also, engages the brake, if it’s necessary. Control via an output from STIO or XTIO.

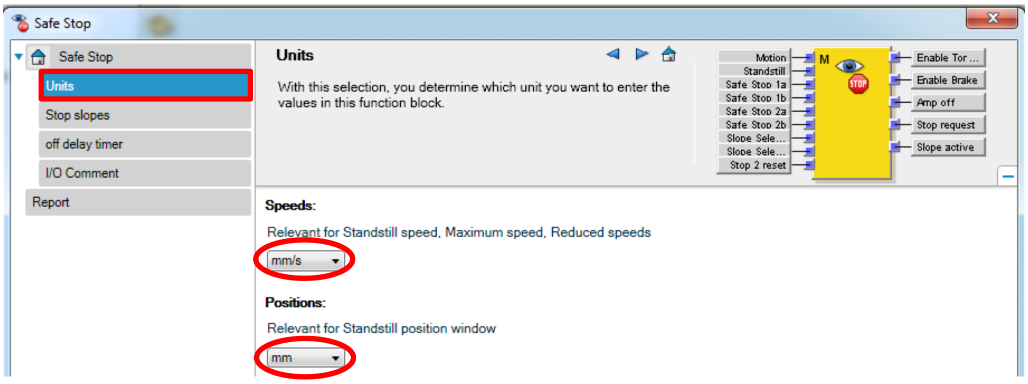
Stop request:
Not safe signal, that triggers the stop ramp of the drive. Control via an output from STIO or XTIO.

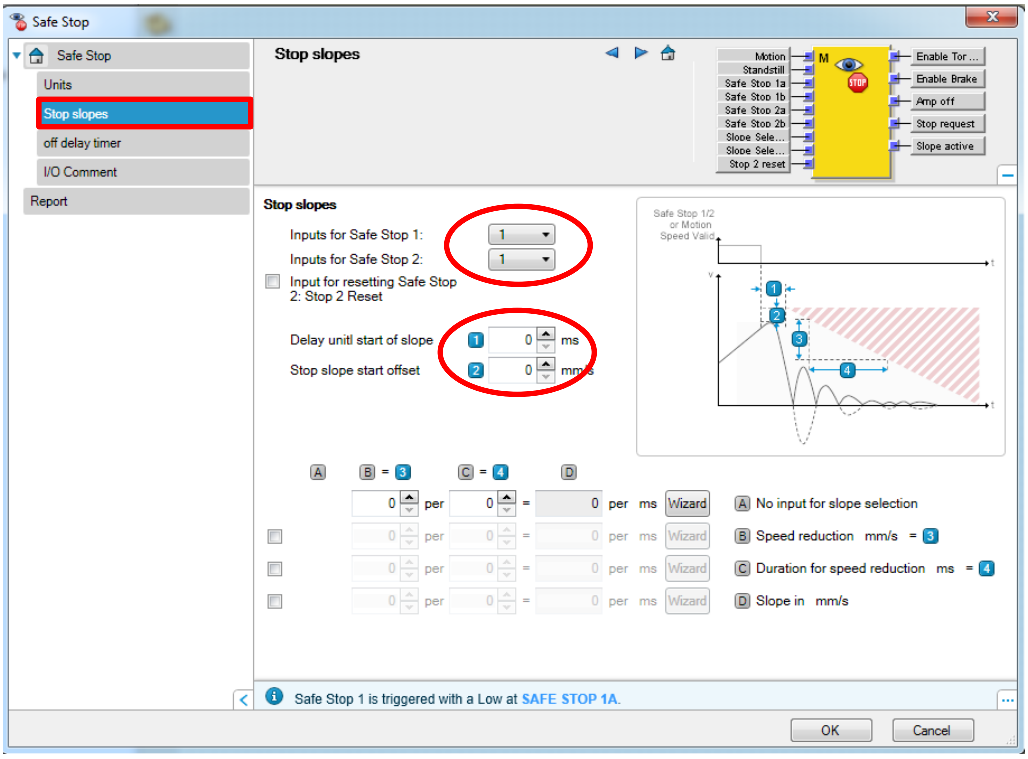
Ramp active:
Not safe signal, which schows if a stop ramp is active.

With double click on the function block, the context menu opens.

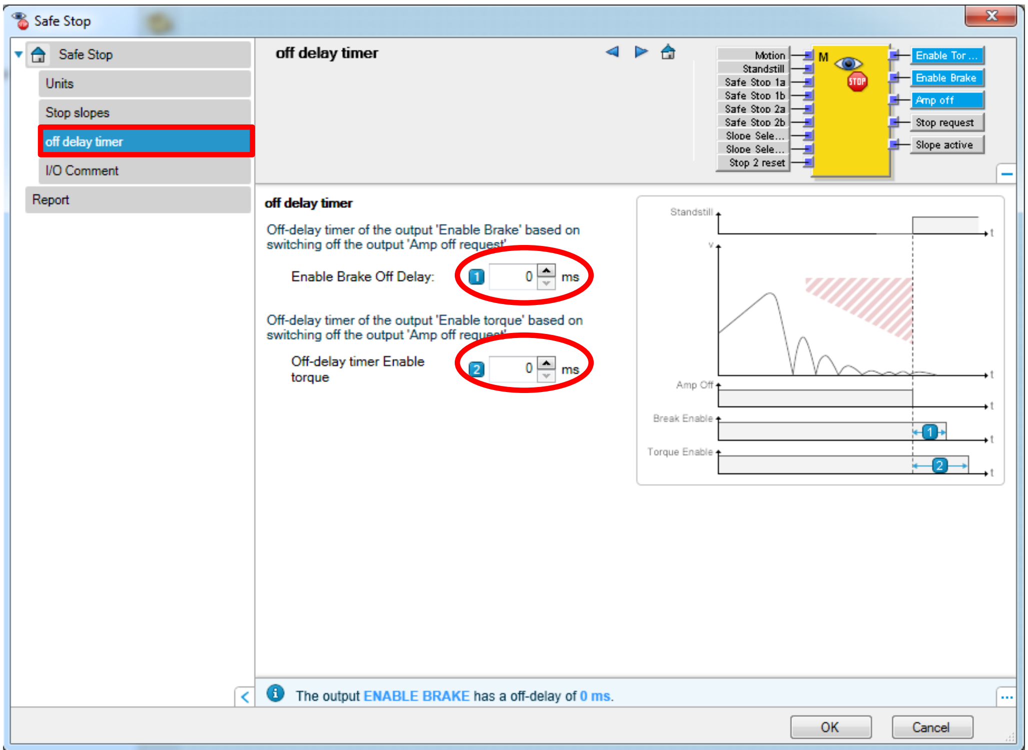


On the tab “Units” can be set, which units to be used e.g. for calculation of speeds (mm/s, km/h, rpm, etc.).

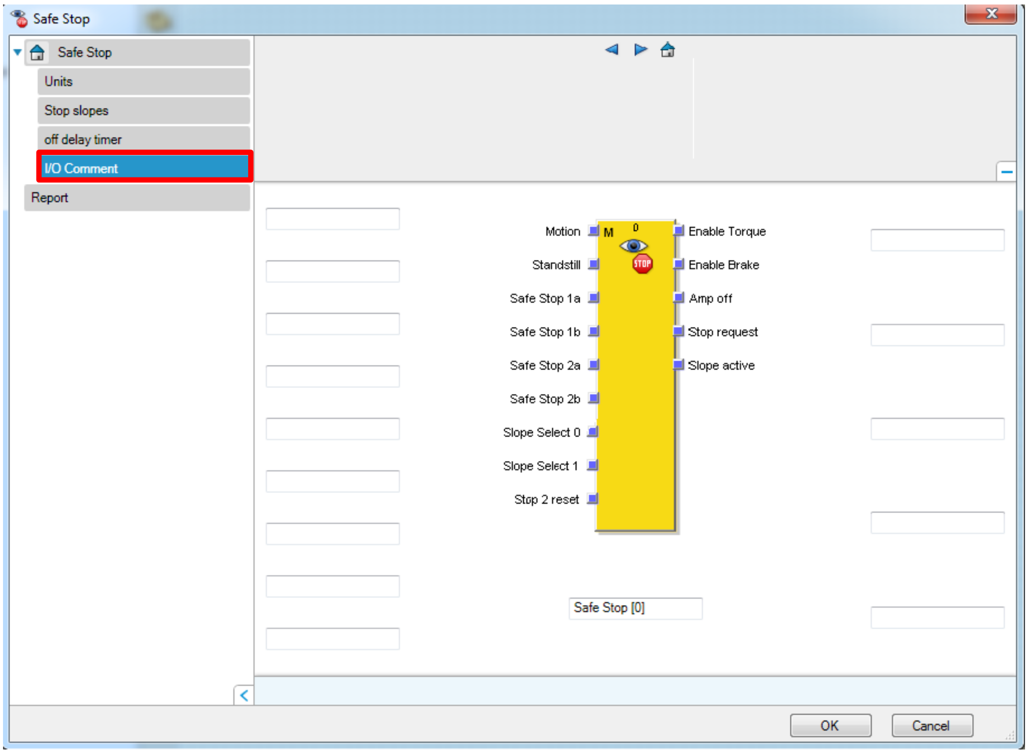




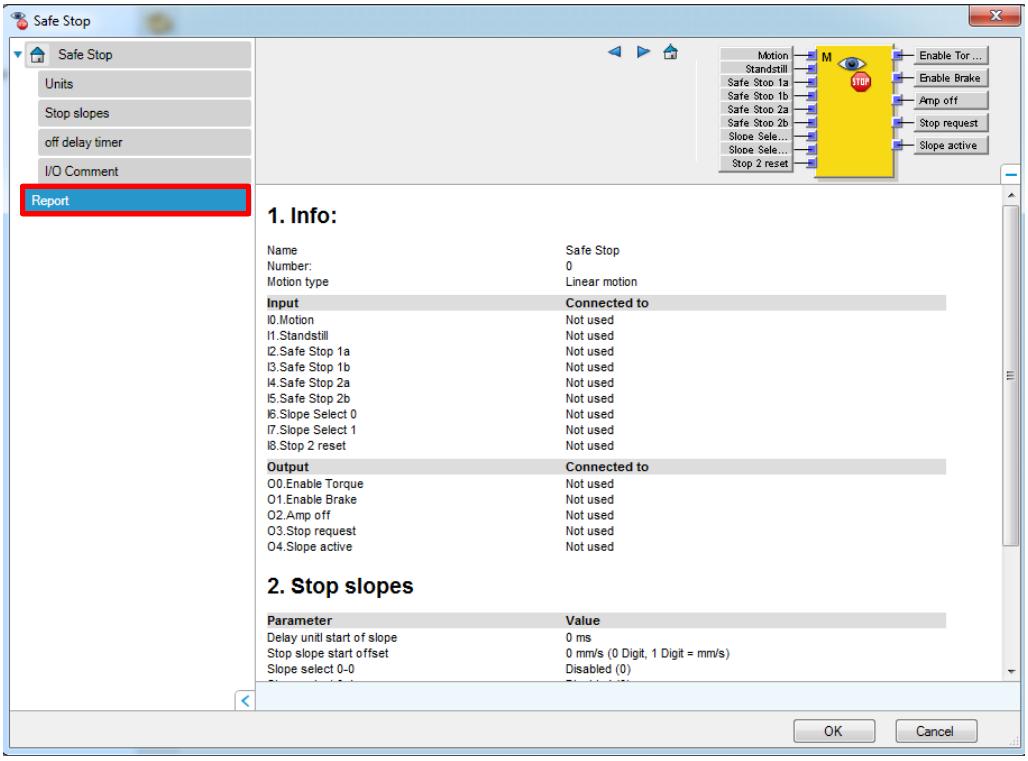
In "Stop slopes" the number of inputs, the delay time and the speed reduction is defined.



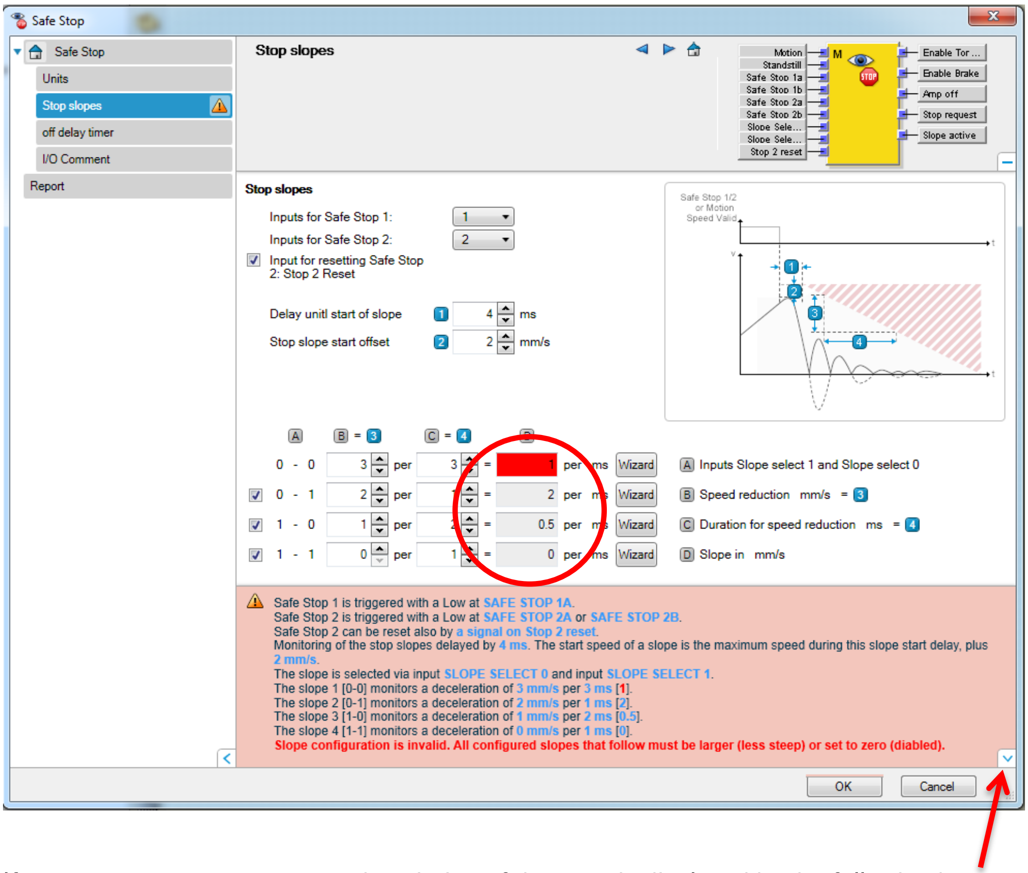
The off delay times for the outputs “Enable Brake” and “Enable Torque” are set in the tab "off delay timer".



The tab "I/O comment" allows it to replace the pre-defined names for the function block inputs and outputs by own identifiers and to add a name or descriptive text to the function block. This name or text is displayed in the logic editor under the function block.



The tab "report" shows a summary of the configuration of the function block, including all input and output links and the configured parameter.



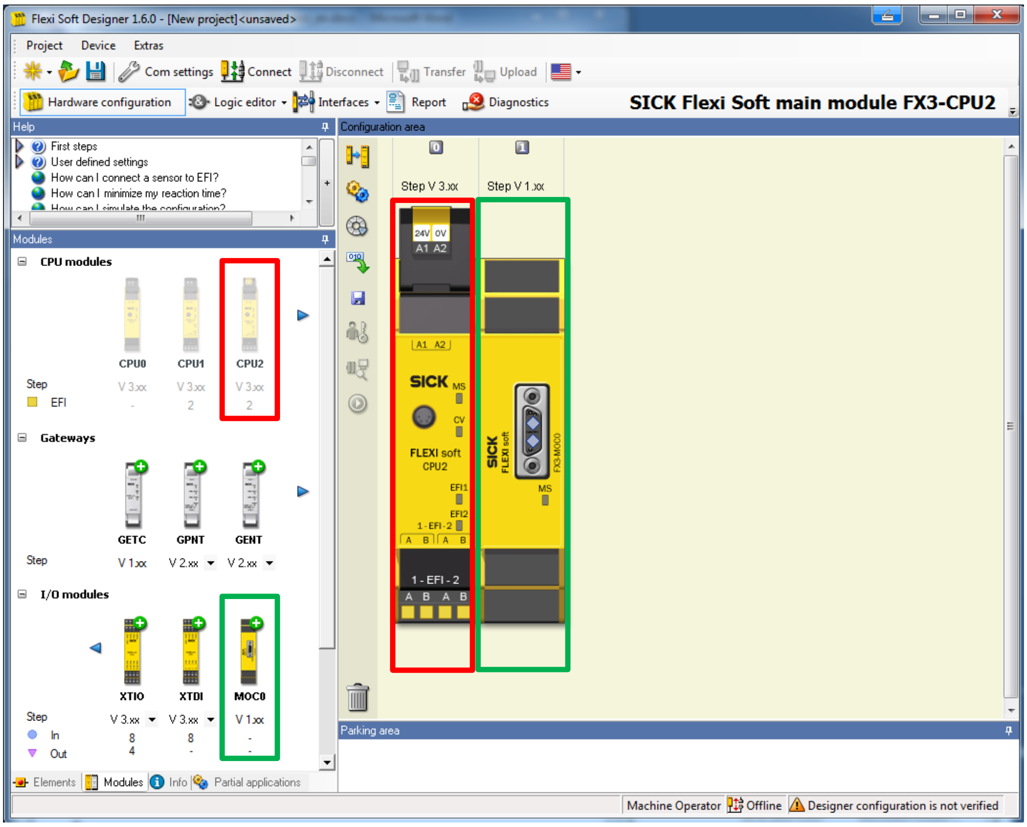
If an error message appears, a description of the error is displayed by the following button, which helps to fix the error.

7 Configuration for an AGV

7.1 Hardware

This is an example and it is not suitable for every AGV.

In this example, the speed of the AGV is determined by using the Drive Monitor FX3-MOC0. The speed is transmitted by the CPU and EFI to the laser scanner. Thereupon the laser scanner activates his field set.

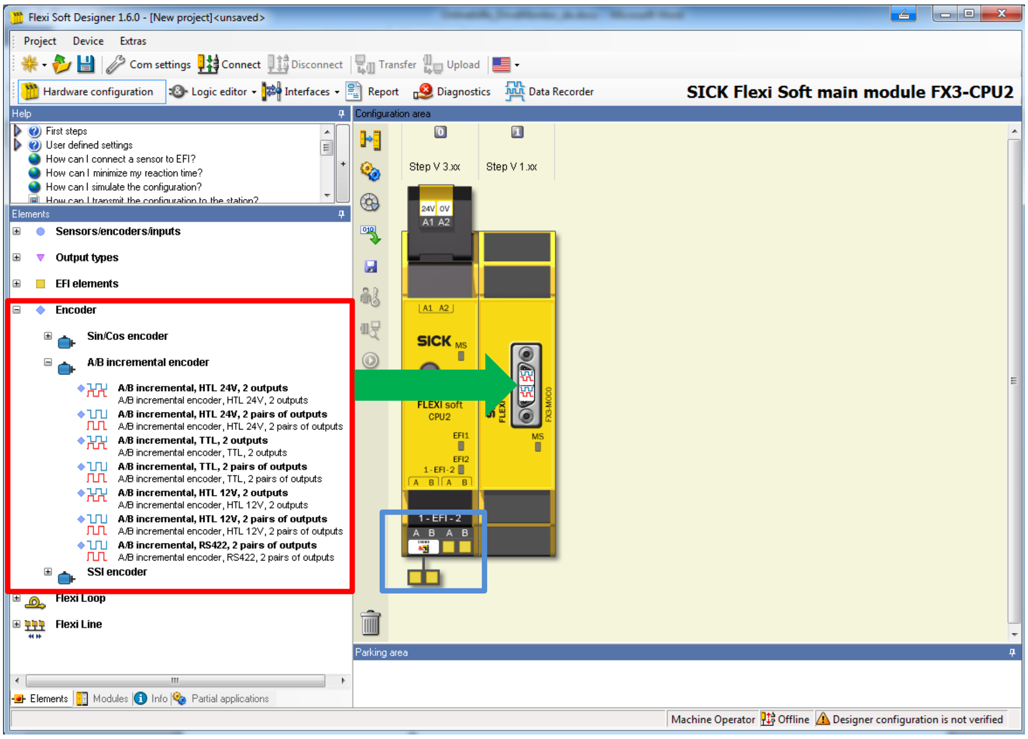


After the start of the FSD, the required modules must be placed in the configuration area.

At first the CPU module should be selected.
In this example the CPU 2 is selected with a simple click. After that the module is indicated in the configuration area.

The Drive Monitor is located at the I/O modules. With a simple click on the picture, the Drive Monitor is also placed in the configuration area.

The station is configured and the encoders can be connected with the Drive Monitor.



In addition to the modules the convenient Encoder must be placed in the hardware.

The encoder can be found in the tab "Elements". By clicking on the encoder symbol there are more choices.

With "drag and drop", the matching encoders can be dragged on the Drive Monitor.

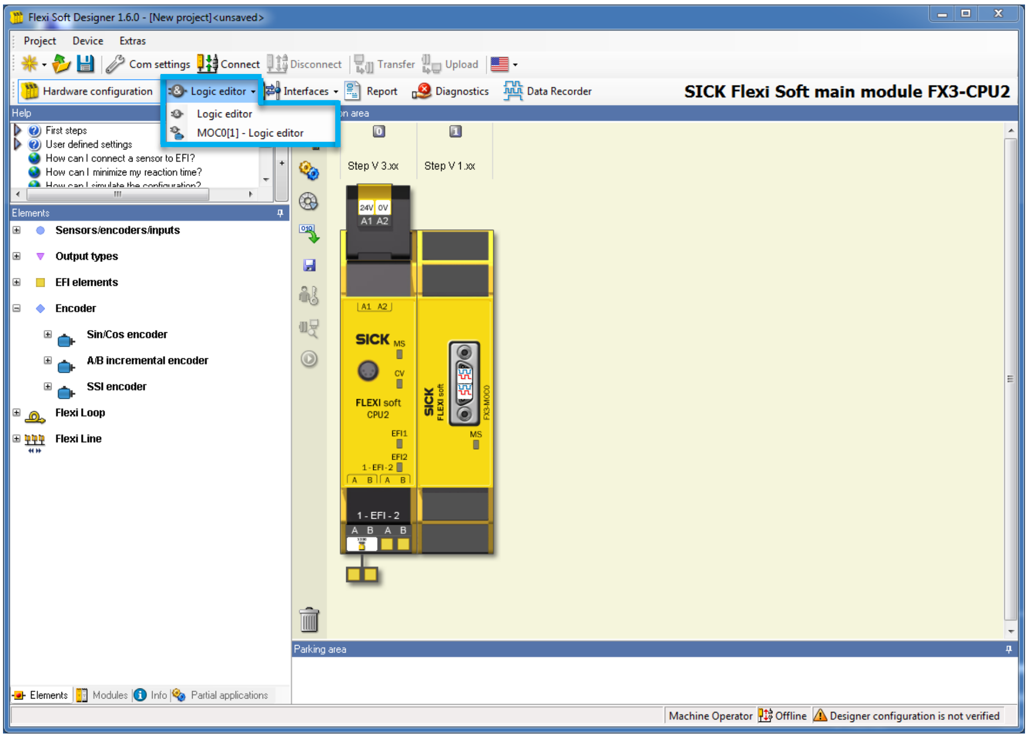
By double clicking on the encoder (in the configuration area), the configuration menu opens (see Chapter 3).

Moreover, a safety laser scanner S300 Mini is connected to the CPU2.

7.2 MOC Logic

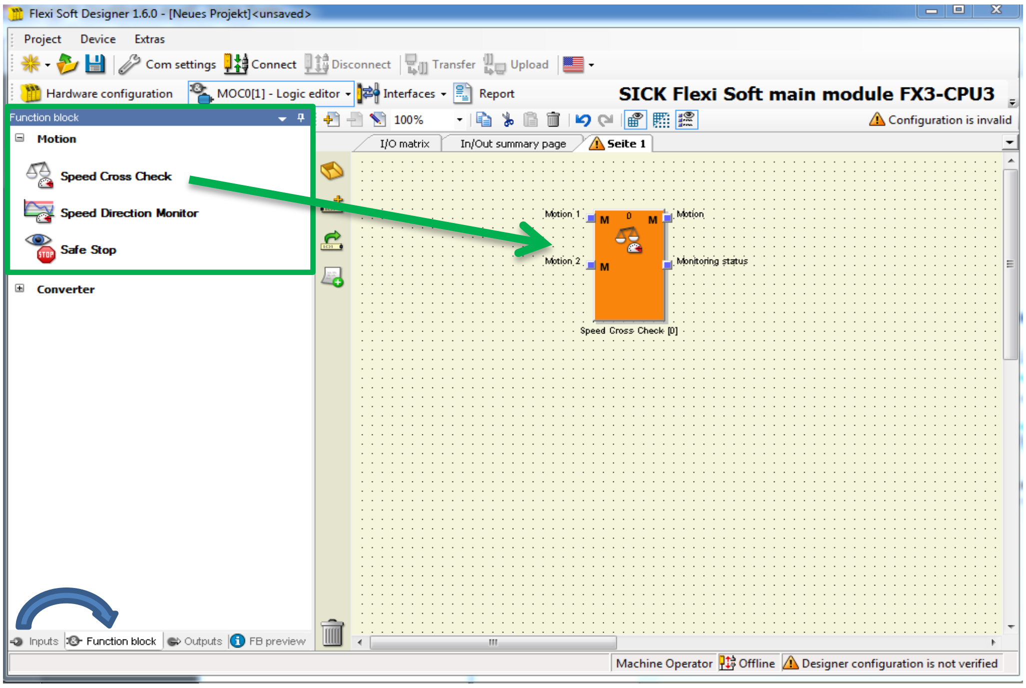
Step 1: *Calling up the MOC Editor*

After the hardware is configured, the MOC logic must be configured.



On the tab "Editor Logic", it is possible to get to the CPU or the MOC logic. By clicking on "MOC0 [1]-logic editor" the user reaches the logic of the Drive Monitor (MOC logic).

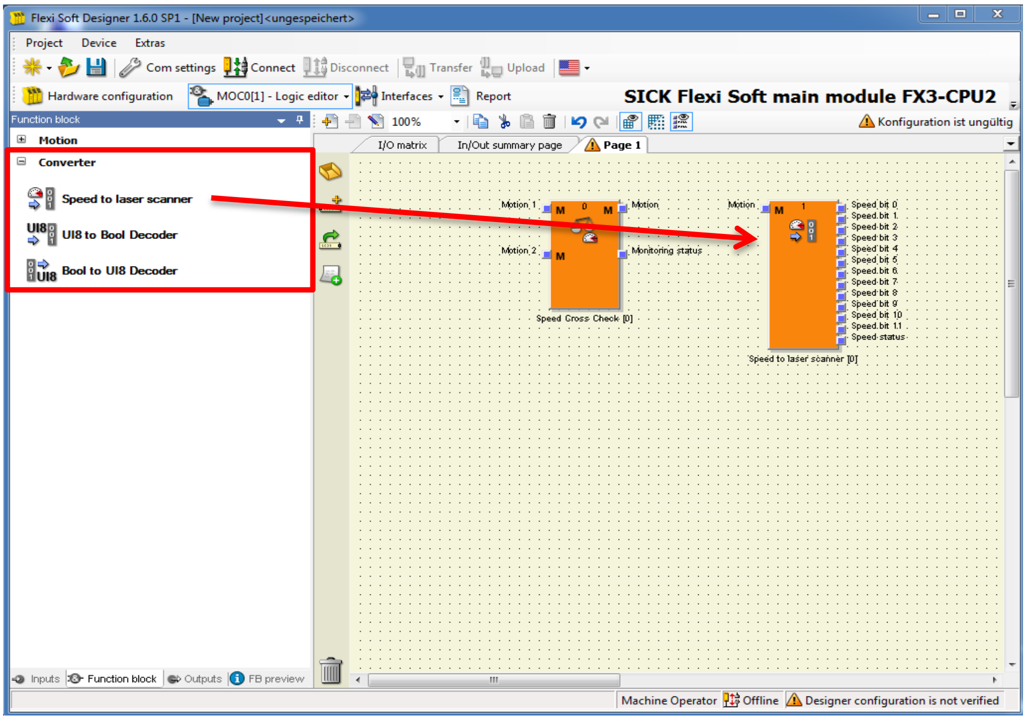
Step 2: *Placing of the function blocks*



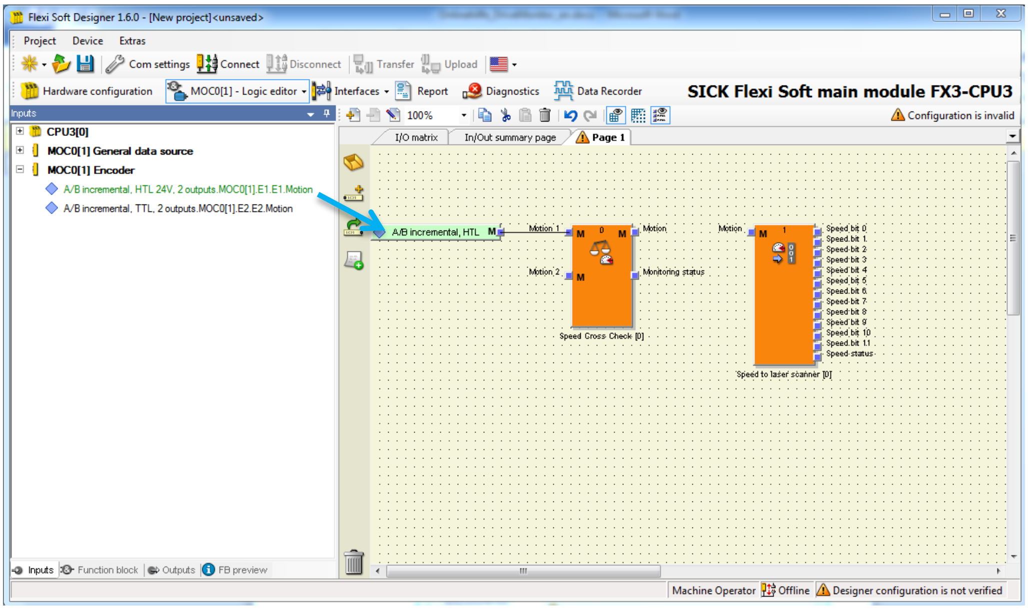
At first it is necessary to switch the tab from “Inputs” to “Function block”.
At the development of the logic, three tabs are especially important: “Inputs”, “Function block”, “Outputs”.

With "drag & drop" function blocks can be inserted or deleted in the logic .
There are function blocks for monitoring functions (motion) as well function blocks for data conversion (converter).

For this application a “Speed Cross Check” (Motion) and a “Speed to laser scanner” (Converter) are required.



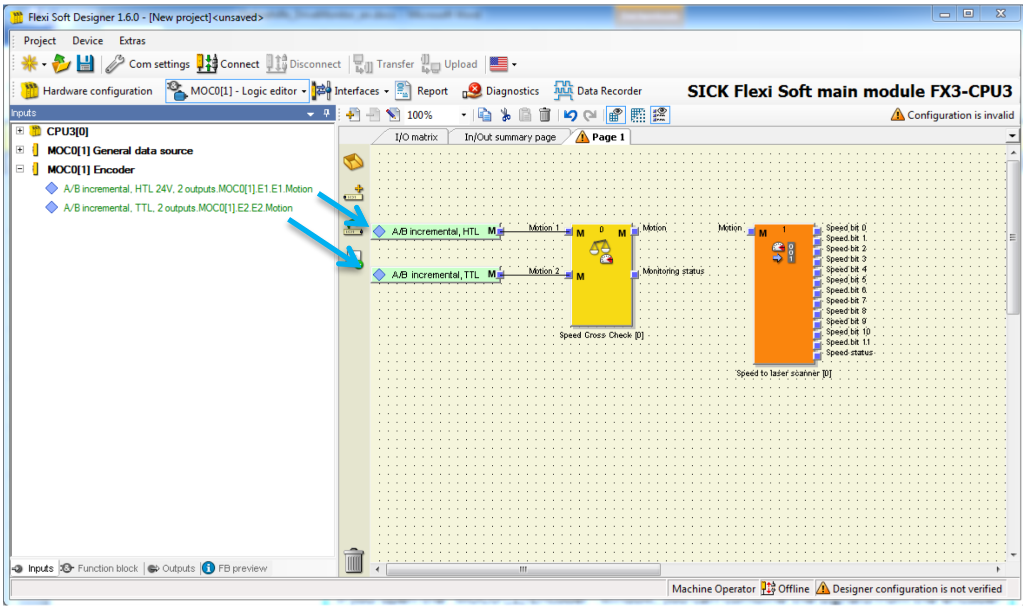
<u>Step 3:</u> *Integrating of the encoder*



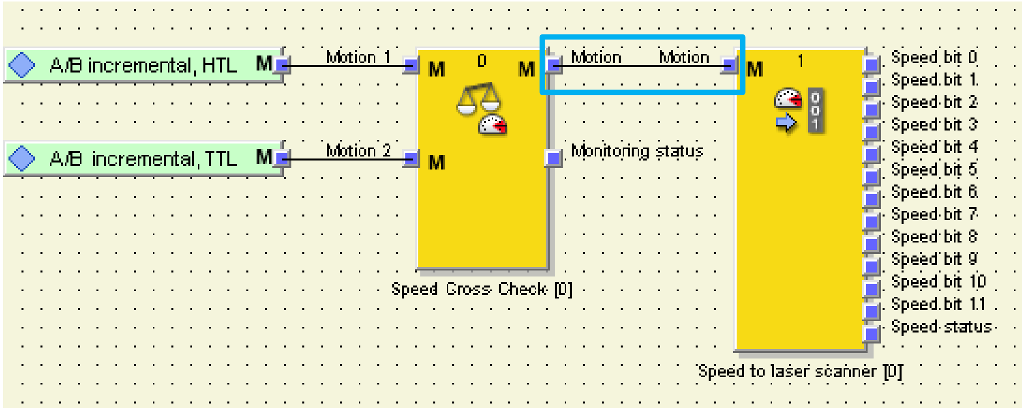
In the tab "Inputs", the information from the CPU or from the encoders is pasted into the MOC logic.

When opening the window "MOC0 [1] encoder", the signals of the encoder can be connected to the function block "Speed Cross Check". In this connection, the signals from the encoders have to be connected with the inputs "Motion 1" and "Motion 2".

If a signal is connected with the function block, the description on the left side will be green. If a signal is not used, the description will be black.
Moreover the function block changes the color from orange to yellow, because it has got two valid signals.

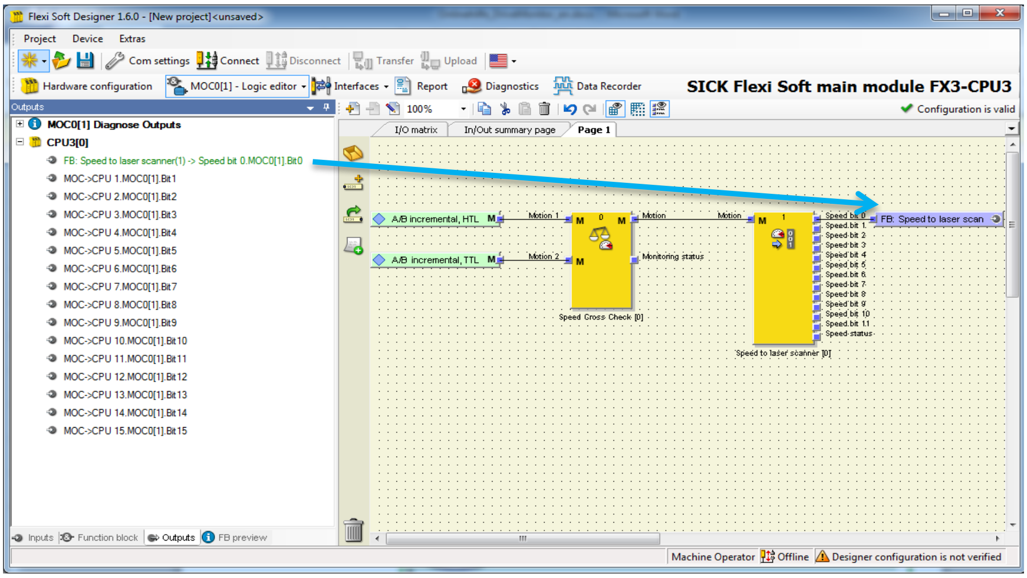


Step 4: *Integrating of the Speed to laser scanner*



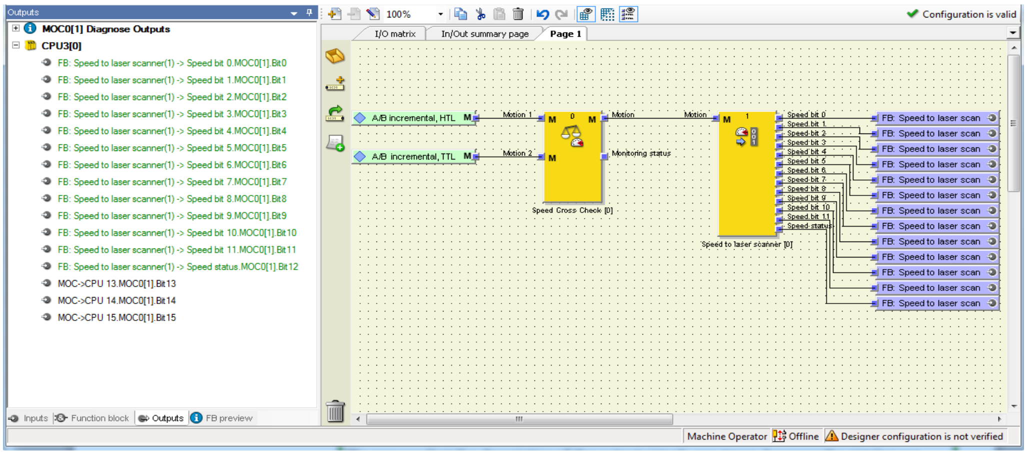
Moreover the two blocks have to be connected via the IOs “Motion”.
The connection can be made via drag & drop.
After that, the converter becomes yellow too, because it receives a valid signal from the Speed Cross Check.

<u>Step 5:</u> *Routing of the speed to the CPU*



Finally the speed bits have to be routed via the CPU from the Flexi Soft to the laser scanner. In order to that, the speed bits must be connected with the output bits. Over the tab "Outputs", the necessary elements can be found.

If an output is connected with the function block, the description on the left side will be green. If a signal is not used, the description will be black.



These output data will be used as input data in the CPU logic.
Over the tab "Inputs" in the CPU logic and the opening of the “MOCO [1]” window, the signals of the MOC can be used in the CPU (next chapter).

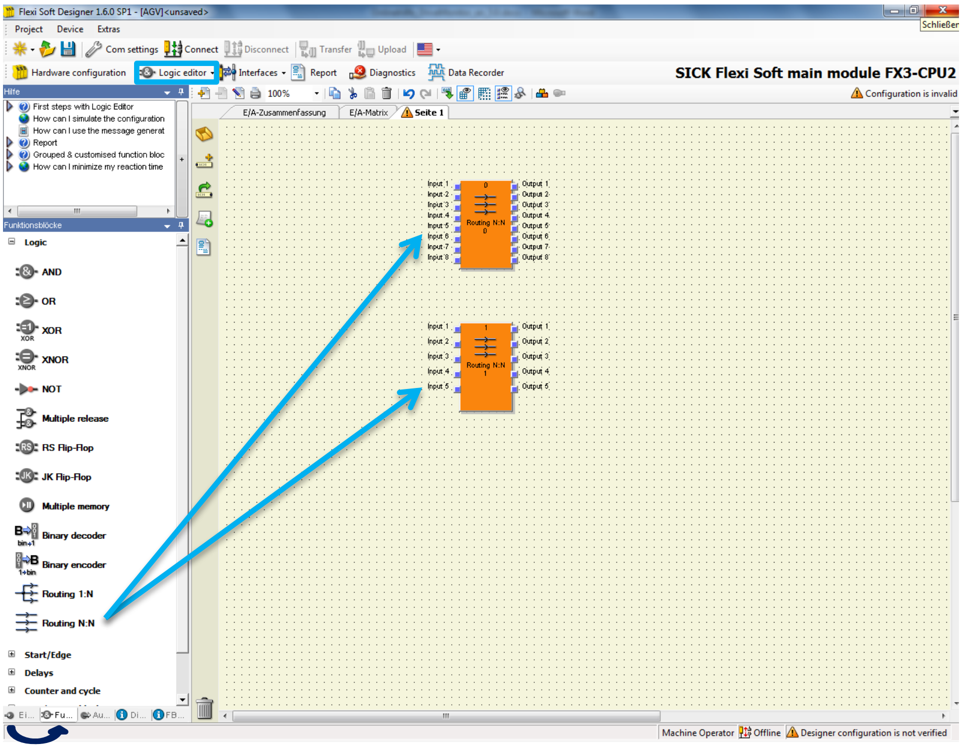
Now the encoder signals are connected to the Flexi Soft logic and they can be used in the logic of the Flexi Soft CPU.

7.3 CPU Logic

<u>Step 1:</u> *Placing of the Routing N:N function blocks*

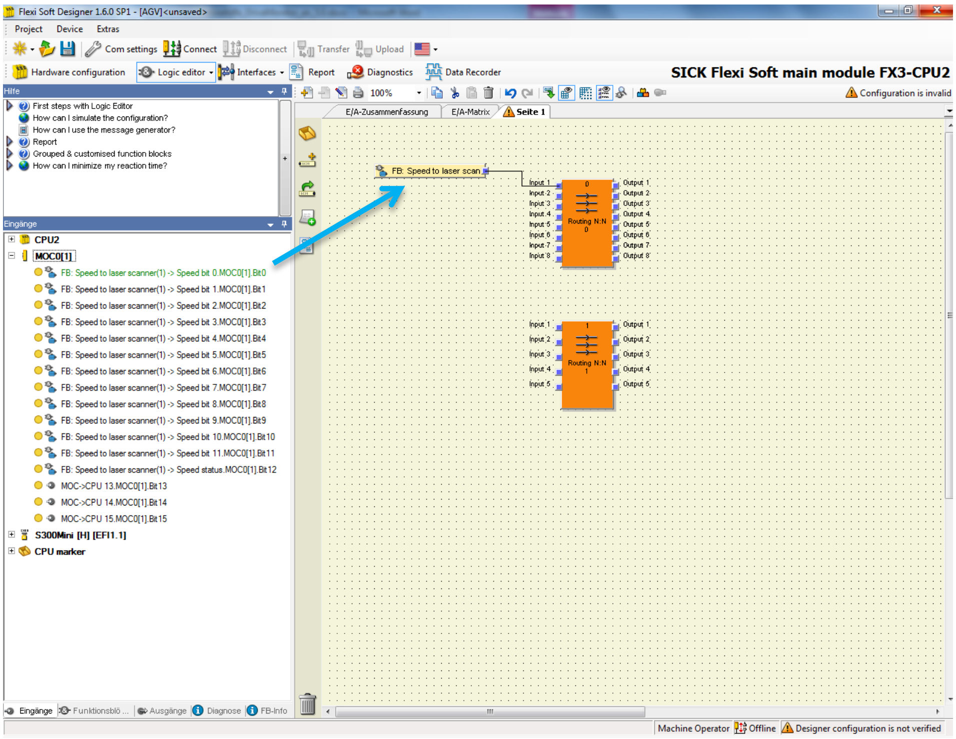
Finally, the CPU Logic must be configured.

By clicking on "Logic editor" the user reaches the logic of the CPU. The blue frame indicates how to get to the logic editor.

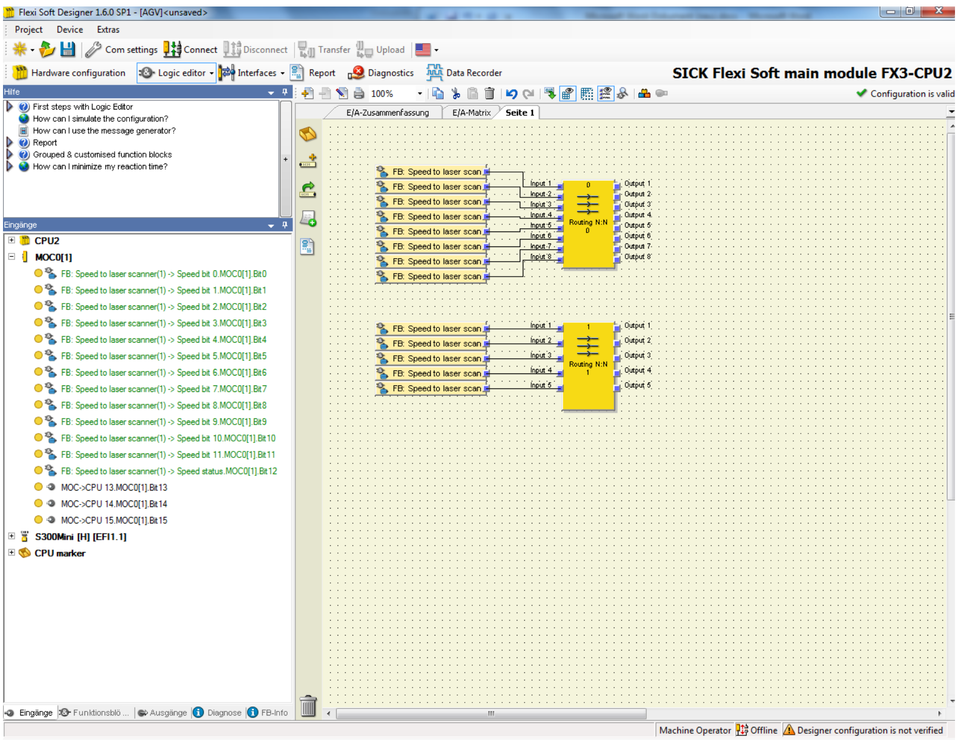


The function block "routing N:N" is found by changing the tab from "inputs" to "function blocks".
Via Drag & Drop it is possible to place the two function blocks in the logic.
With this function block the signals are getting routed from the CPU via EFI to the laser scanner. With a double click on the function block it is possible to change the number of inputs and outputs. In this case, thirteen inputs are required

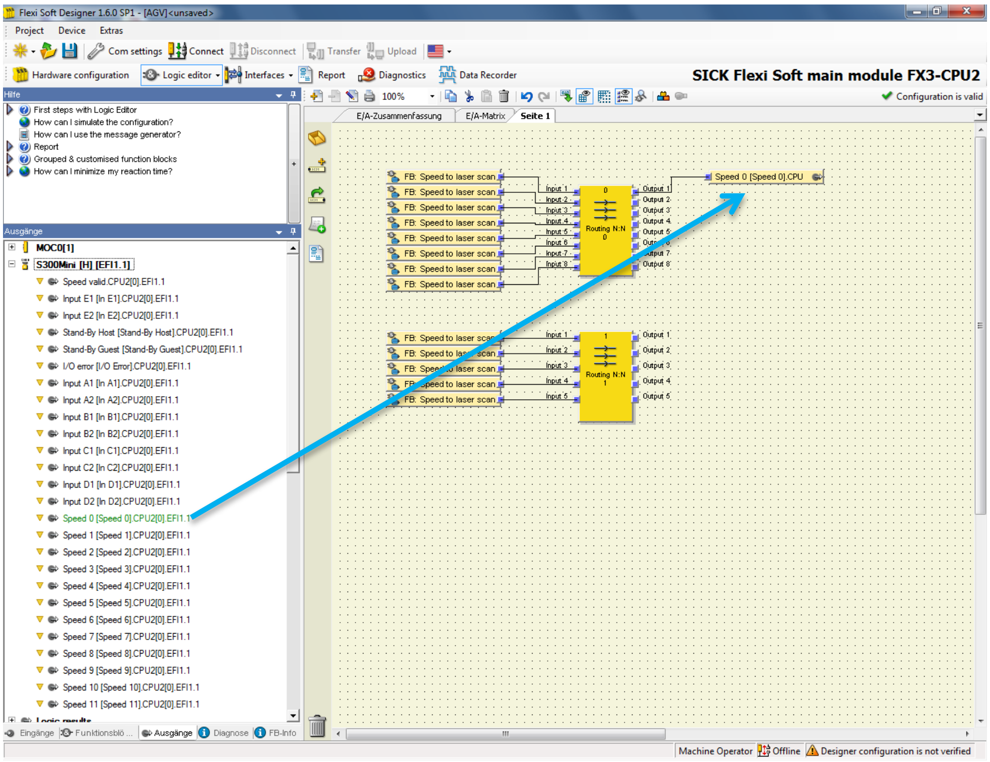
<u>Step 2:</u> *Routing of the issued speed from the MOC0 to the function blocks of the CPU*



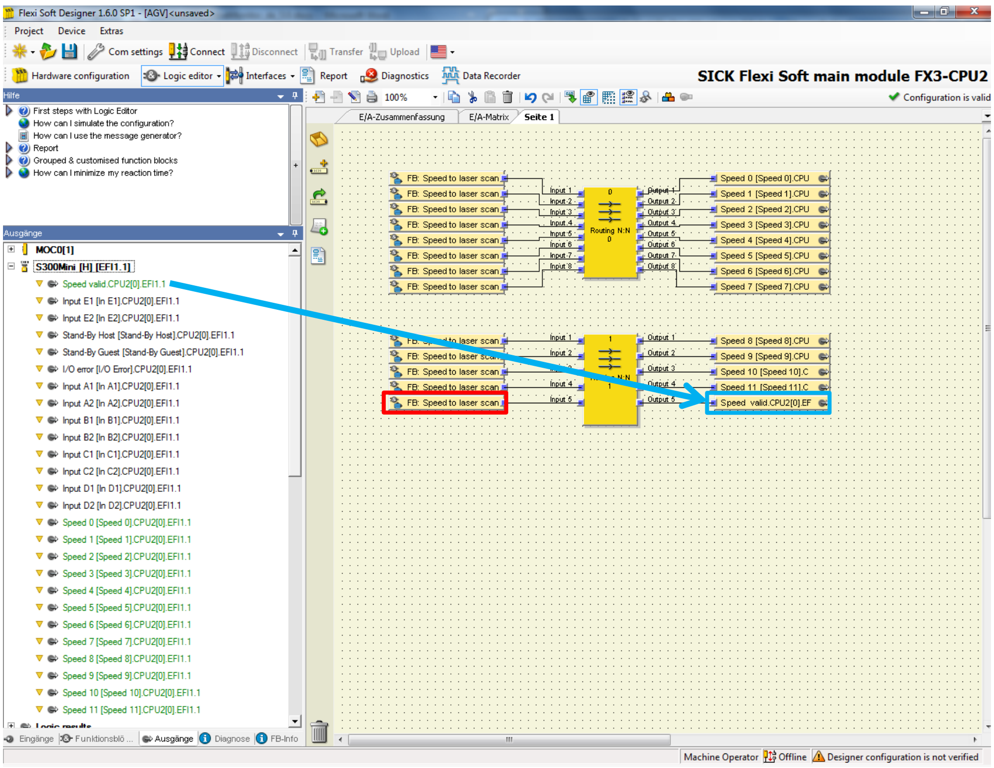
Now the information from the MOC logic is needed and therefore the tab is switched to “Inputs”.
In the Flexi Soft logic there is now a modul named “MOC[x]”, where the IOs can be found.
These are the bits which are declared as output signals in the MOC logic.
The information from the MOC logic is now being used.
The information is available via the backplane bus
Now the signals from the Drive Monitor have to be connected with the Inputs (Drag & Drop). Now it can be seen that the description of the input signals is green, because these are connected with the function block. Moreover the function block changes the color from orange to yellow, because it is connected to thirteen valid signals.



<u>Step 3:</u> *Routing the speed via EFI to the laser scanner.*



The outputs are available via “S300Mini [H] [EFI1.1]”.
With Drag & Drop, the “Speed 0” can be connected to the “Output 1” of the “Routing N:N” function block. Therefore the descriptionof the output becomes green.
Now it is necessary to do the same steps for the Speed 1 to 11. Because Input 13 is the "Speed status", there musst be selected "Speed valid" as output.



Now the logic for the AGV is configured and can be saved and transferred to the Flexi Soft station.

8 Configuration for a Gantry

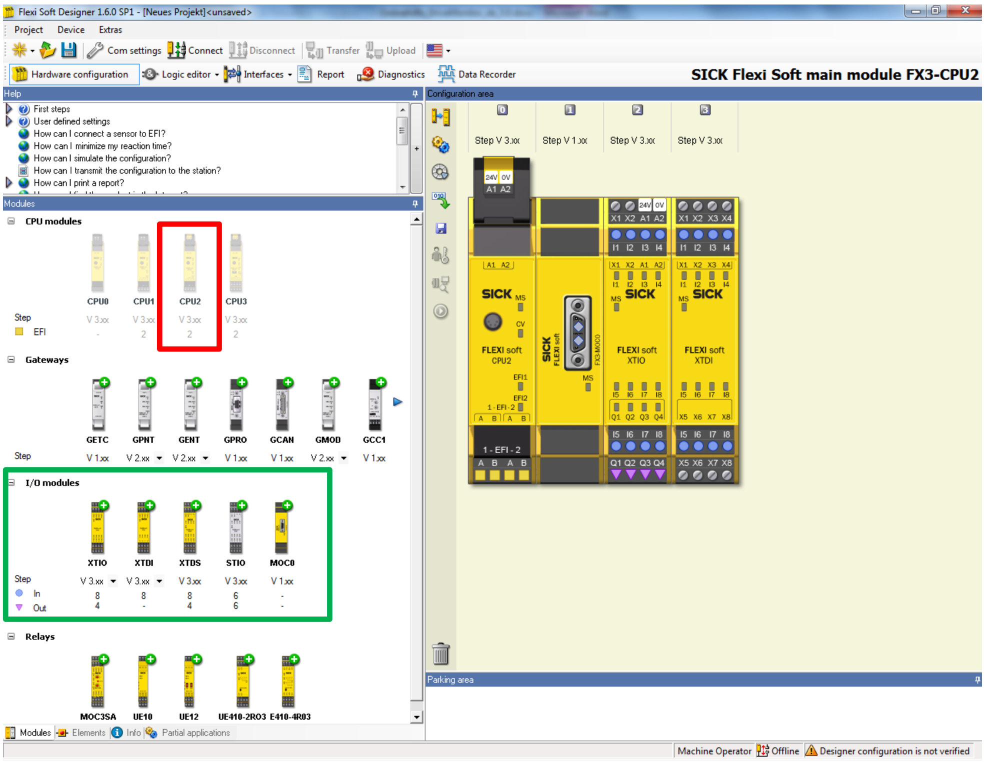
8.1 Hardware

<u>This is an example and it is not suitable for every Gantry.</u>

In this example, the following safety functions of a machine can be realized via the Drive Monitor:

- Access protection with standstill detection
- Maintenance mode with safe reduced speed
- Safe stop functions

These functions can be found in diverse machines, such as CNC-, winding machines or servo presses.

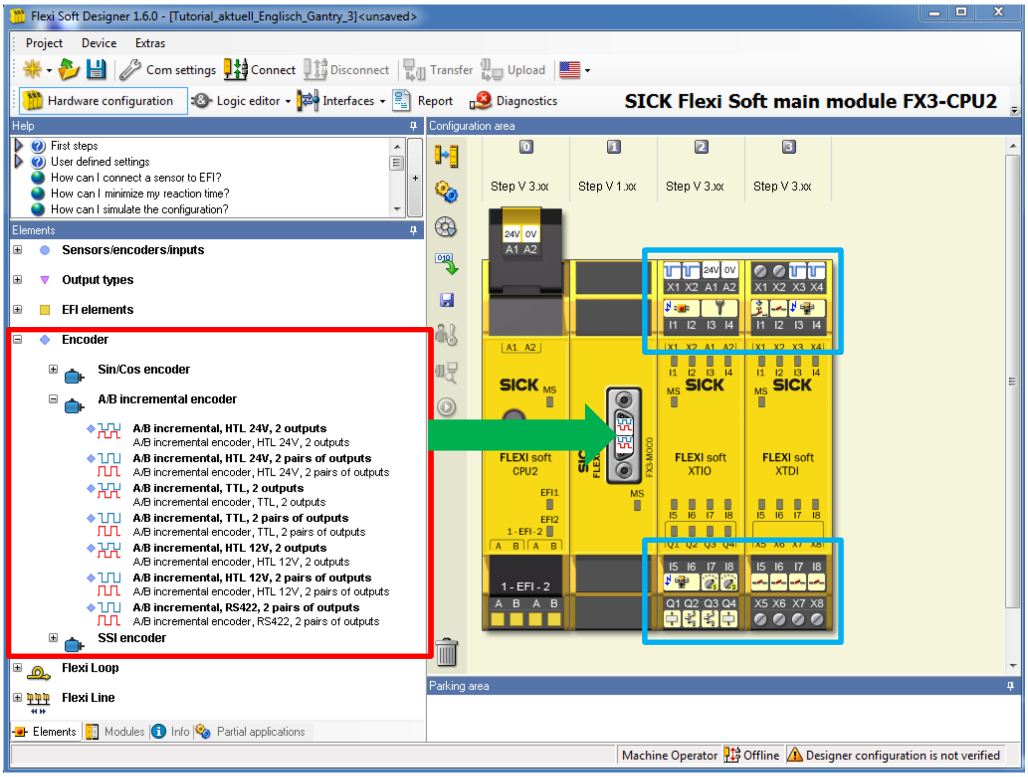


After the start of the FSD, the required modules must be placed in the configuration area.

At first the CPU module should be selected. In this example the CPU 2 is selected with a simple click. Then the module is indicated in the configuration area.

The modules XTIO, XTDI and MOC0 are located at the I/O modules. With a simple click on the picture, these modules are also placed in the configuration area.

The station is configured and the encoders can be connected with the Drive Monitor.



In addition to the modules the convenient Encoder must be placed in the hardware.

The encoder can be found in the tab "Elements". By clicking on the encoder symbol there are more choices.

With "drag and drop", the matching encoders can be dragged on the Drive Monitor.

By double clicking on the encoder (in the configuration area), the configuration menu opens (see Chapter 3).

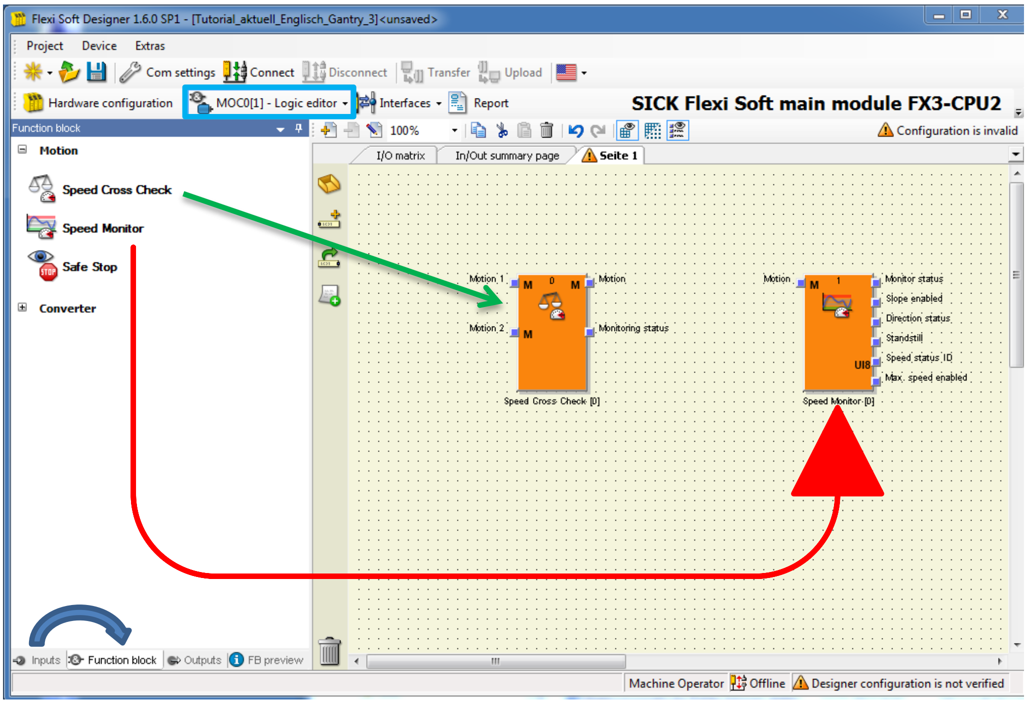
Moreover various elements like E-Stop, Enabling switch, single channel NO etc. are connected to the inputs and outputs of the modules "XTIO" and "XTDI". These elements can be found under the same-named Tab "Elements".

8.2 MOC Logic

Step 1: *Placing of the function blocks*

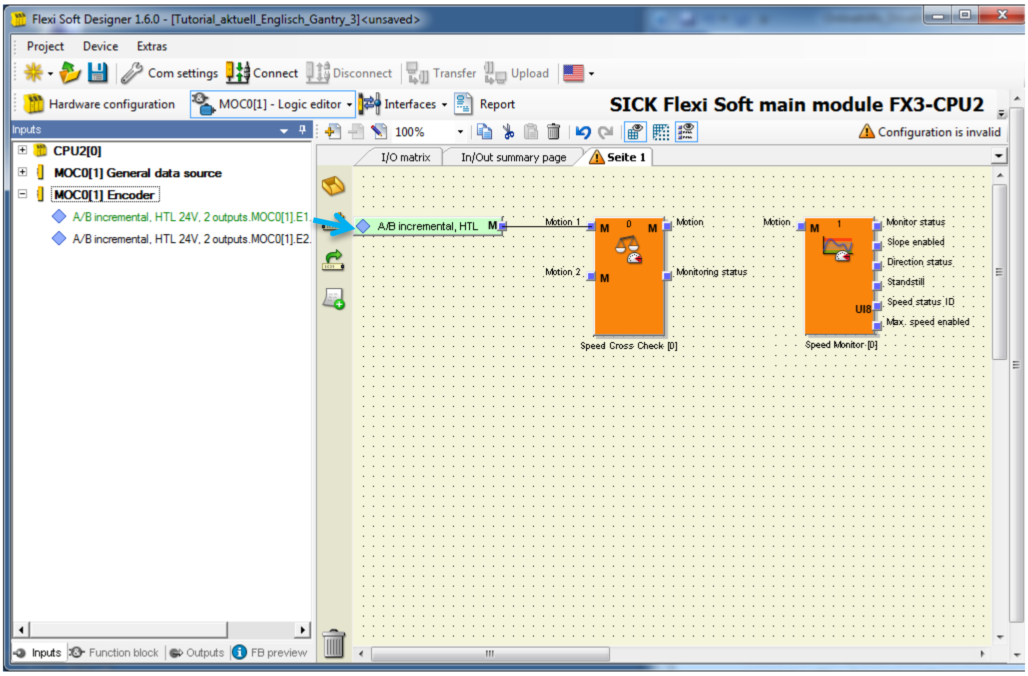
After the hardware is configured, the MOC logic must be configured.

On the "Editor Logic" tab, it is possible to get to the CPU or the MOC logic. The blue frame indicates how to get to the MOC editor. By clicking on "MOC0 [1] logic editor" the user reaches the logic of the Drive Monitor (MOC logic).



At first it is necessary to switch the tab from “Inputs” to “Function block”.
With drag & drop, the function blocks can be inserted or deleted in the logic .
There are function blocks for monitoring functions as well function blocks for data conversion.
For this application the “Speed Cross Check” and the “Speed Monitor” is required.
Both function blocks belong to the monitoring functions.

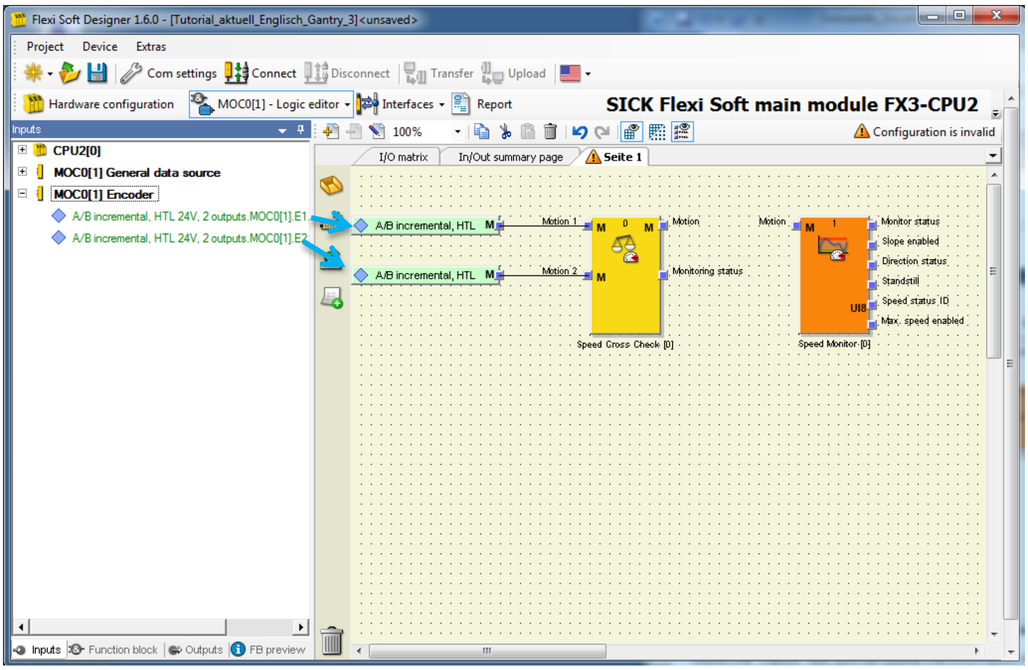
Step 2: *Integrating of the encoder*



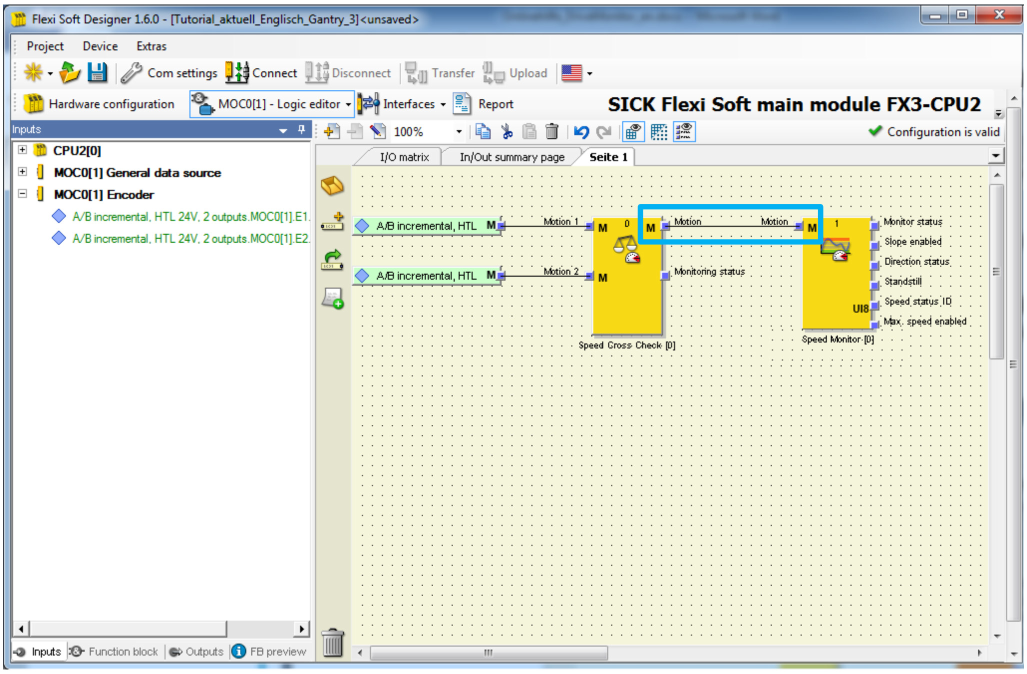
In the tab "Inputs", the information from the CPU or from the encoders is pasted into the MOC logic .
When opening the window "MOCO [1] encoder", the signals of the encoder can be connected to the function block "Speed Cross Check". In this connection, the signals from the encoders have to be connected with the inputs "Motion 1" and "Motion 2".

If a signal is connected with the function block, the description on the left side will be green. If a signal is not used, the description will be black.

Moreover the Speed Cross check changes the color from orange to yellow, because it has got two valid signals.

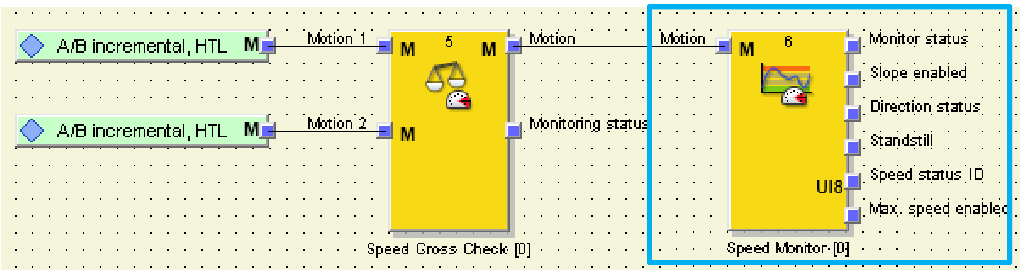


Step 3: *Integrating of the function block Speed Monitor*



Moreover the two blocks have to be connected via the IOs “Motion”.
The connection can be made via drag & drop. After that, the function block “Speed Monitor” becomes yellow too, because it receives a valid signal from the “Speed Cross Check”.

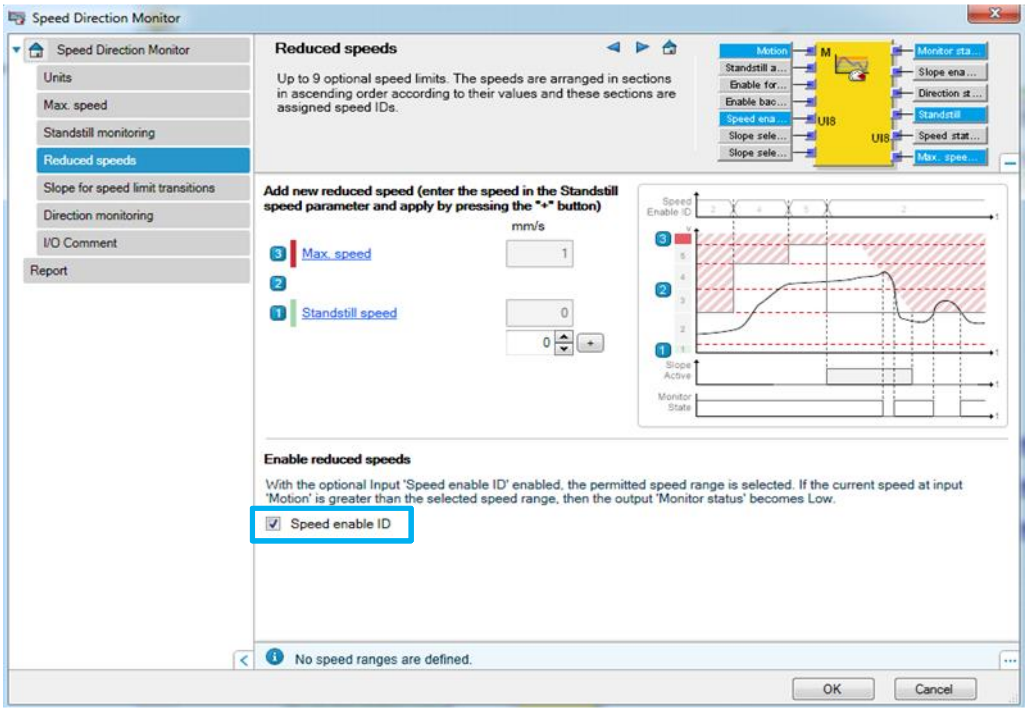
<u>Step 4:</u> *Creating the optional inputs "Speed enable ID"*



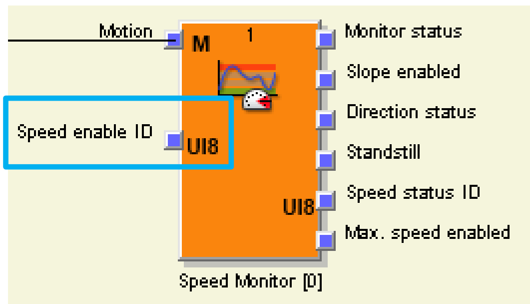
For the configuration of a Gantry, it is necessary to activate the optional input “Speed enable ID”. This input enables a defined speed.

The "Speed Enable" must be enabled:

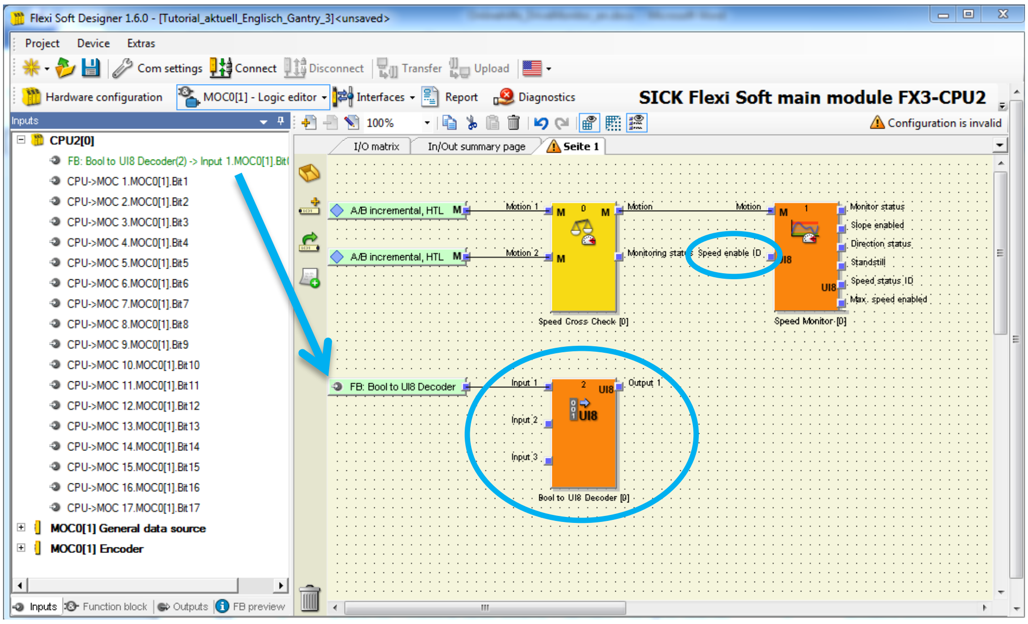
- Double click on the function block Speed Monitor
- Change the tab into “Reduced Speed”
- Speed enable ID ✓



Now the optional input “Speed enable ID” is available. Because the input still has no signal, the functional block becomes orange again.

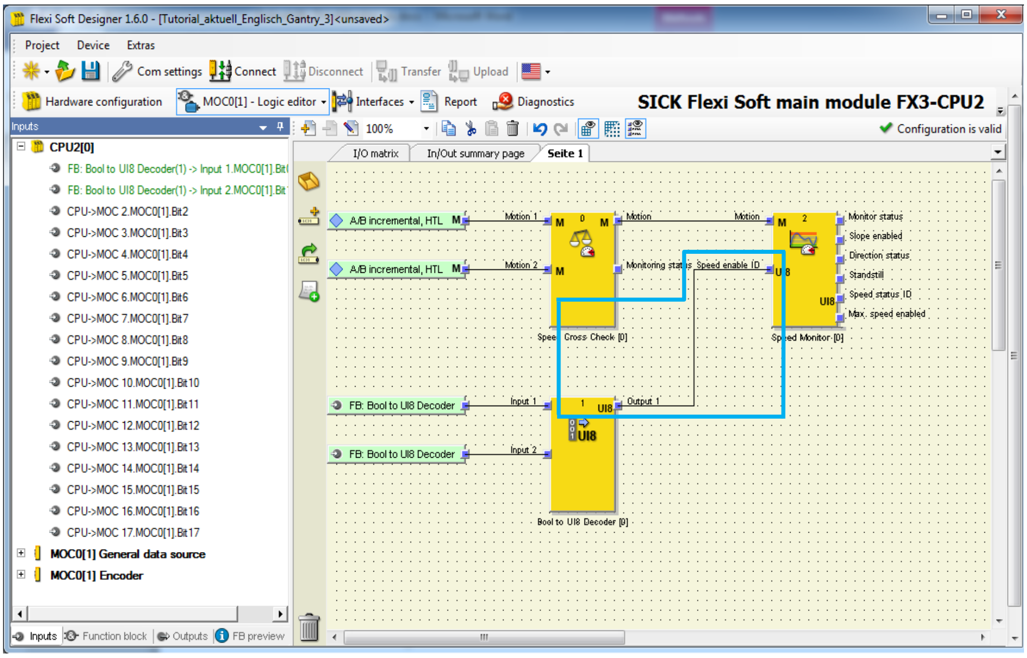


<u>Step 5:</u> *Placing of the converter and integrating of the CPU signals*



The “Bool to UI8 Decoder” is necessary to convert the data types of the signals. This is important to exchange information between the CPU (binary) and the Drive Monitor (integer).
At first, the binary data from the CPU is converted to a Speed enable ID (UI8 data).
Thus, the data can be used in the MOC logic.
Furthermore it’s necessary to change the number of inputs to two.
With a double click on the function block, it is possible to change the number of inputs.

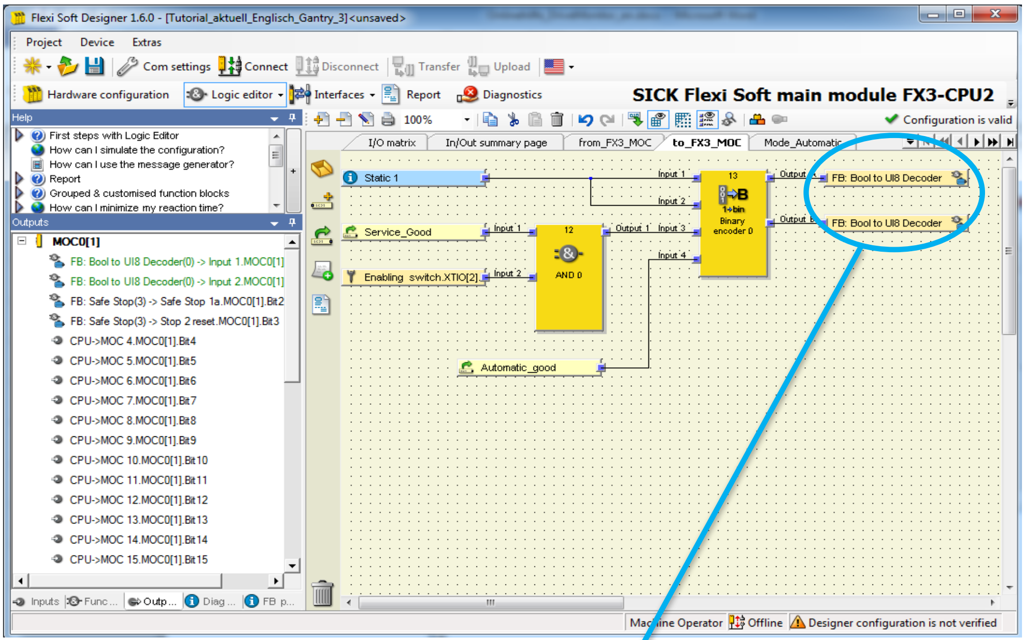
By opening the window "CPU2 [0] ", the signals from the CPU (Bit 0 and 1) can be connected to the function block "Speed Cross Check".



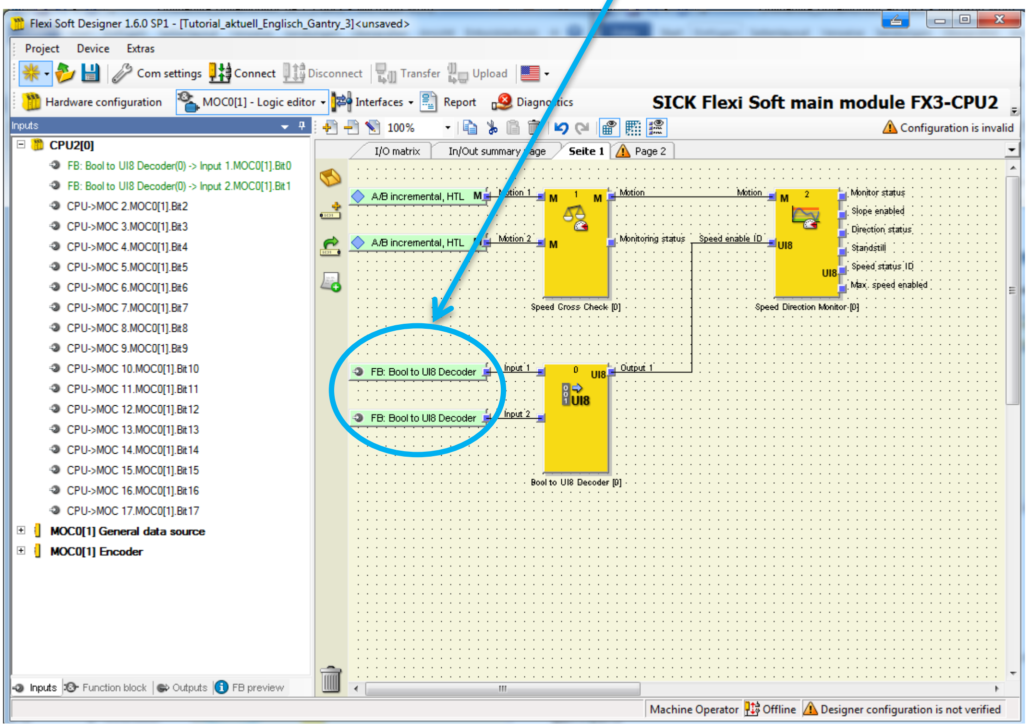
Now the output of the “Bool to UI8 Decoder” must be connected with the input “Speed enable ID” of the Speed Monitor.
The input “Speed enable ID” activates the enable speed.

The pictures show that the data will be sent from the CPU to the MOC logic (more information about the structure of the CPU logic: chapter 7.3).

<u>CPU logic:</u>

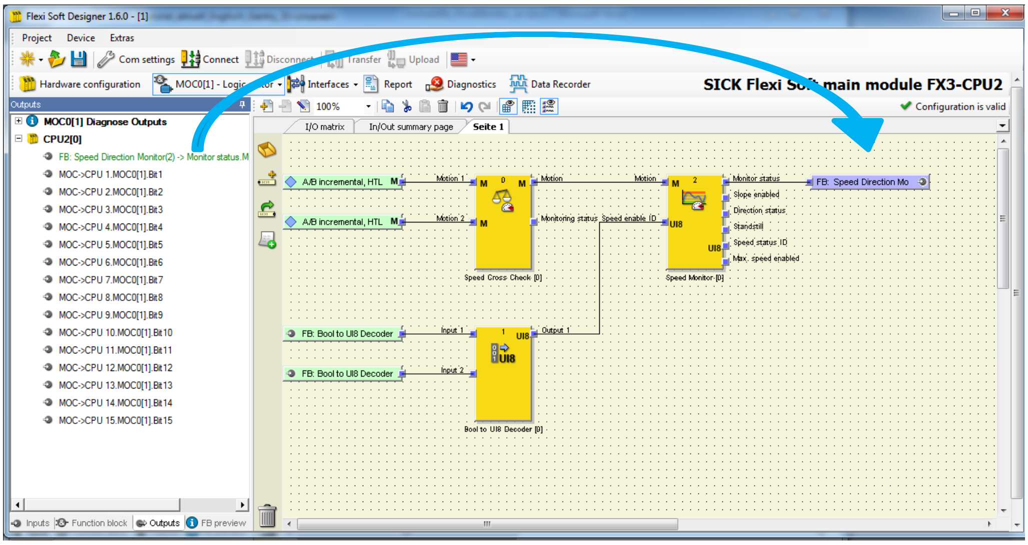


<u>MOC logic:</u>

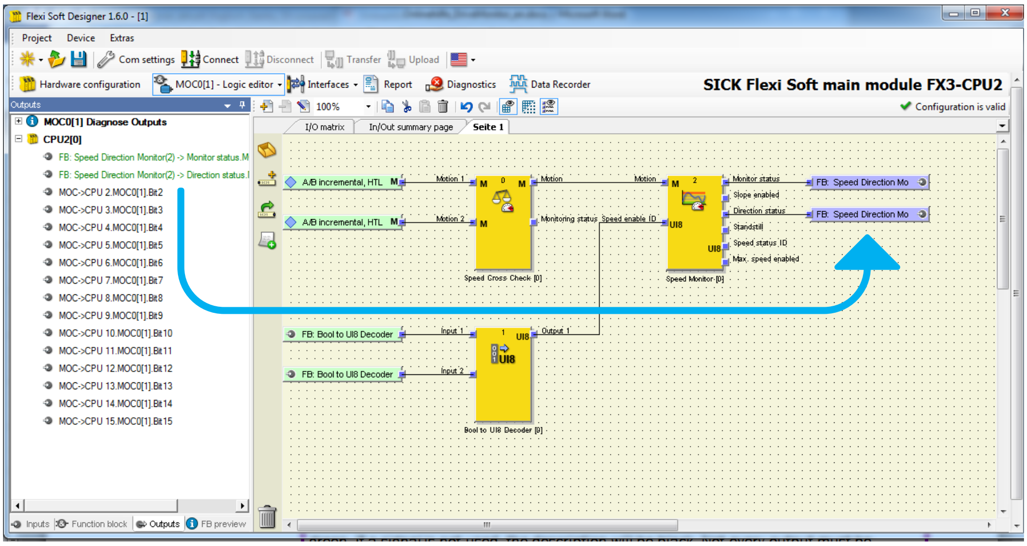


Finally the Speed Monitor receives the information about the input “Speed enable ID”.

<u>Step 6</u>: *Routing the Monitor status and Direction status to the CPU*



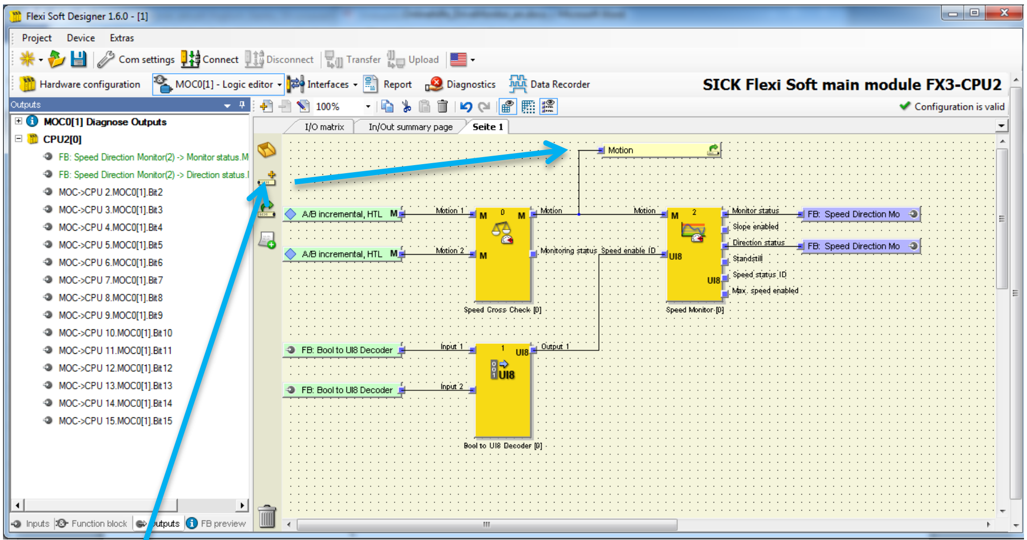
The next step is that the output signals must be connected to the output bits.
If an output is connected with the function block, the description on the left side will be green. If a signal is not used, the description will be black.



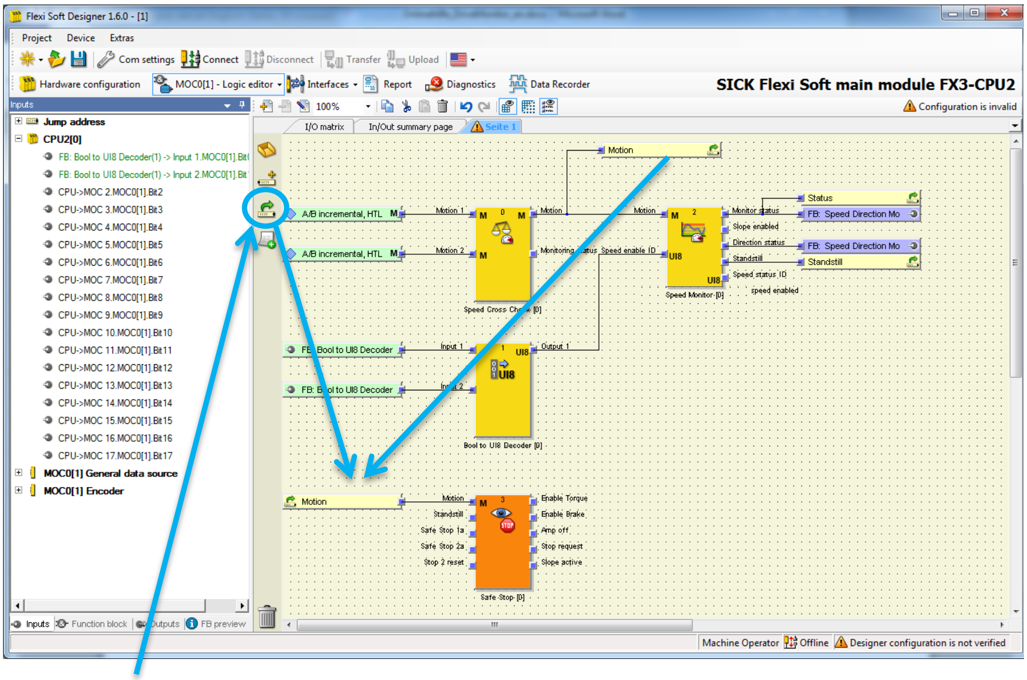
Now the signals “Monitor Status (Bit 0)” and “Direction status (Bit1)” are routed to the CPU.

<u>Step 7:</u> *Creating jump markers and allocate the inputs of the safe stop*

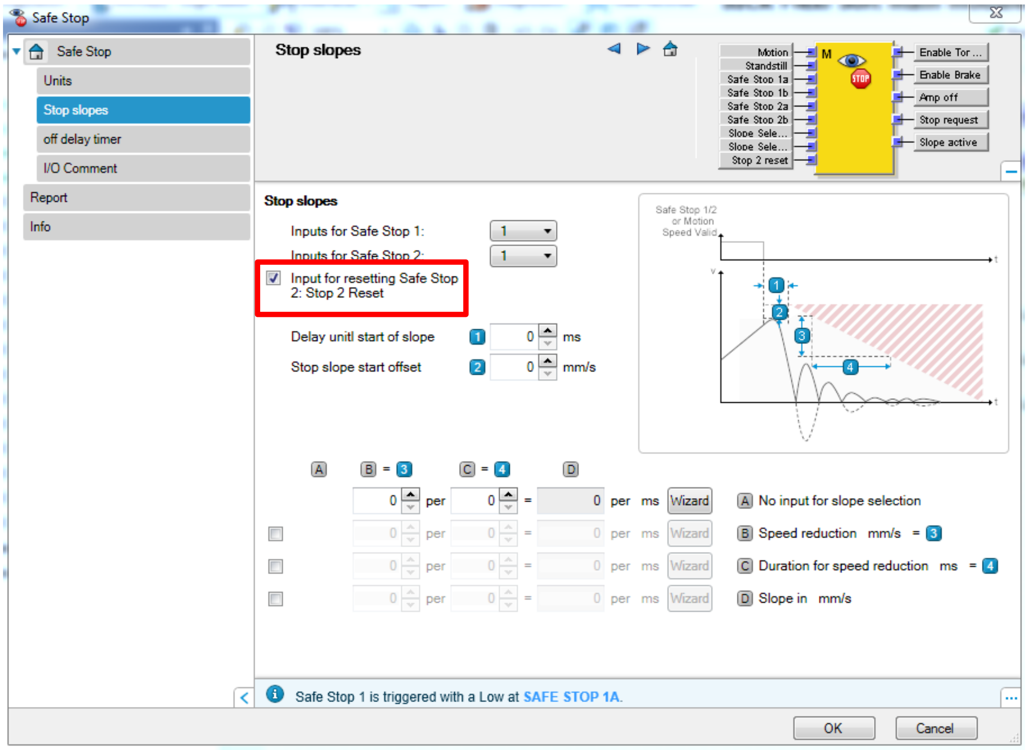
By using jump markers it´s possible to use a signal as input, as well as output for several function blocks in one logic. With Drag & Drop it is possible to insert or delete the jump markers.

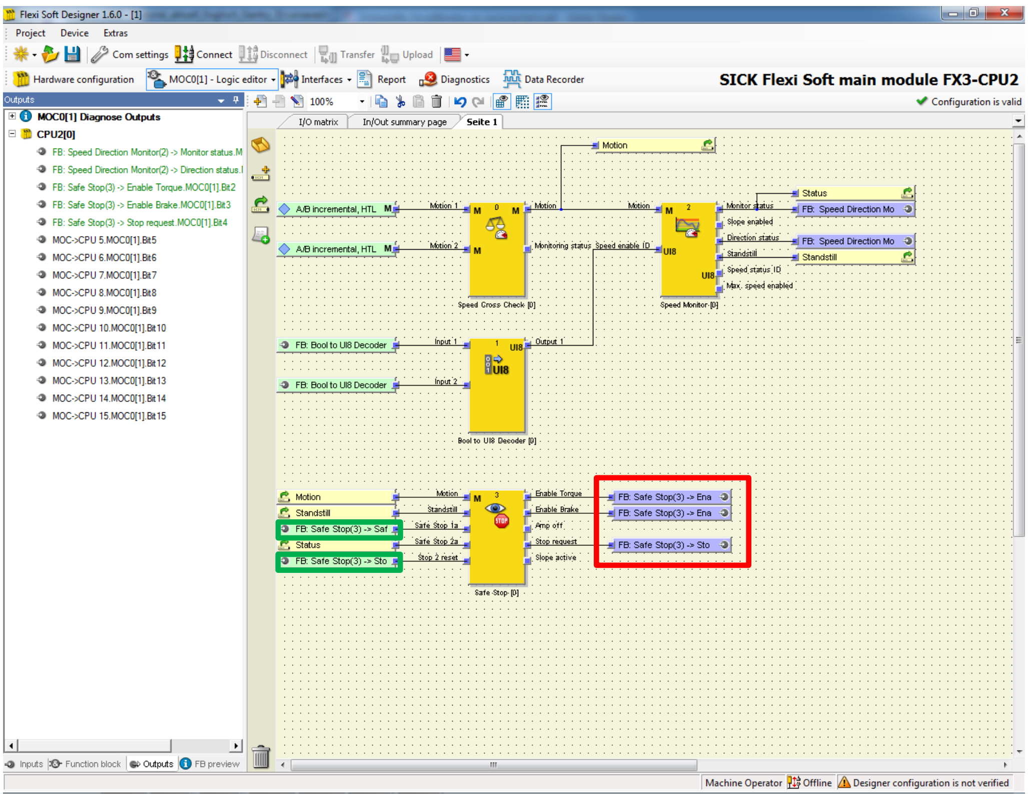


With this Button, the jump marker can be connected to every output signal in the logic.



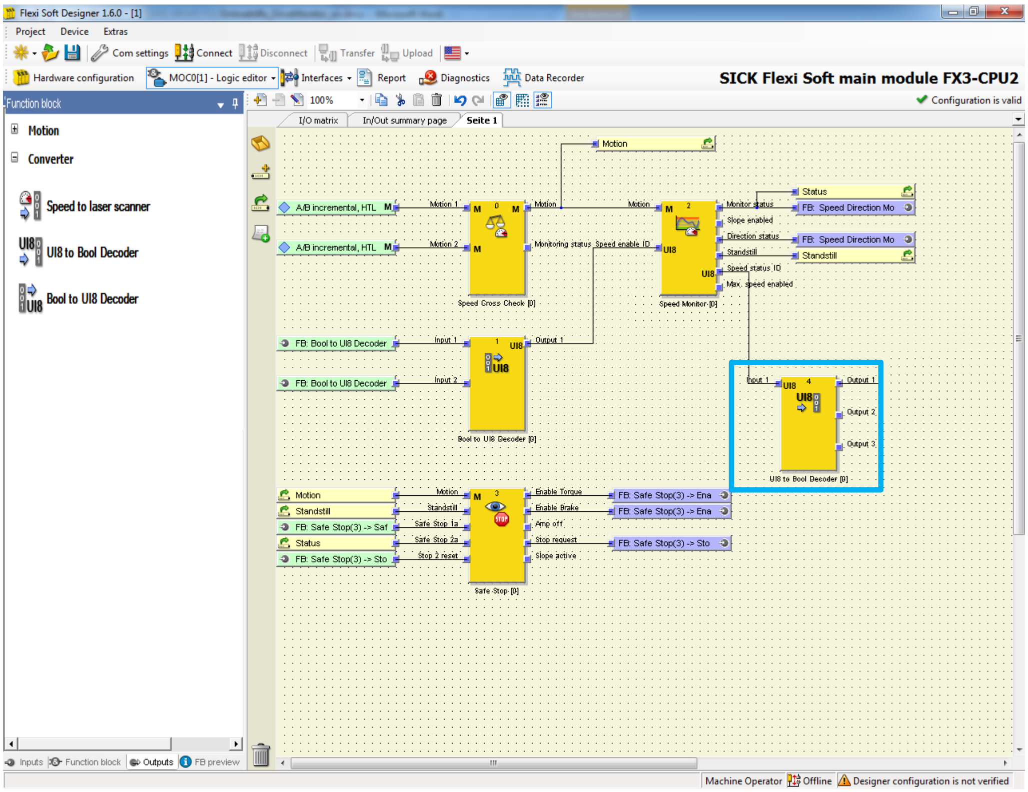
With this button each jump marker can be resumed, which was previously created or used. In addition, the jump markers “Status” and “Standstill” as well as the function block “Safe Stop” was implemented via Drag & Drop and the optional input “Stop 2 reset” was activated (see next image).
By double clicking on the function block and the switch to the tab “Stop slopes” it is possible to find the optional input “Stop 2 reset”.
After that it is necessary to activate the field “Input for resetting Safe Stop 2: Stop 2 Reset”.



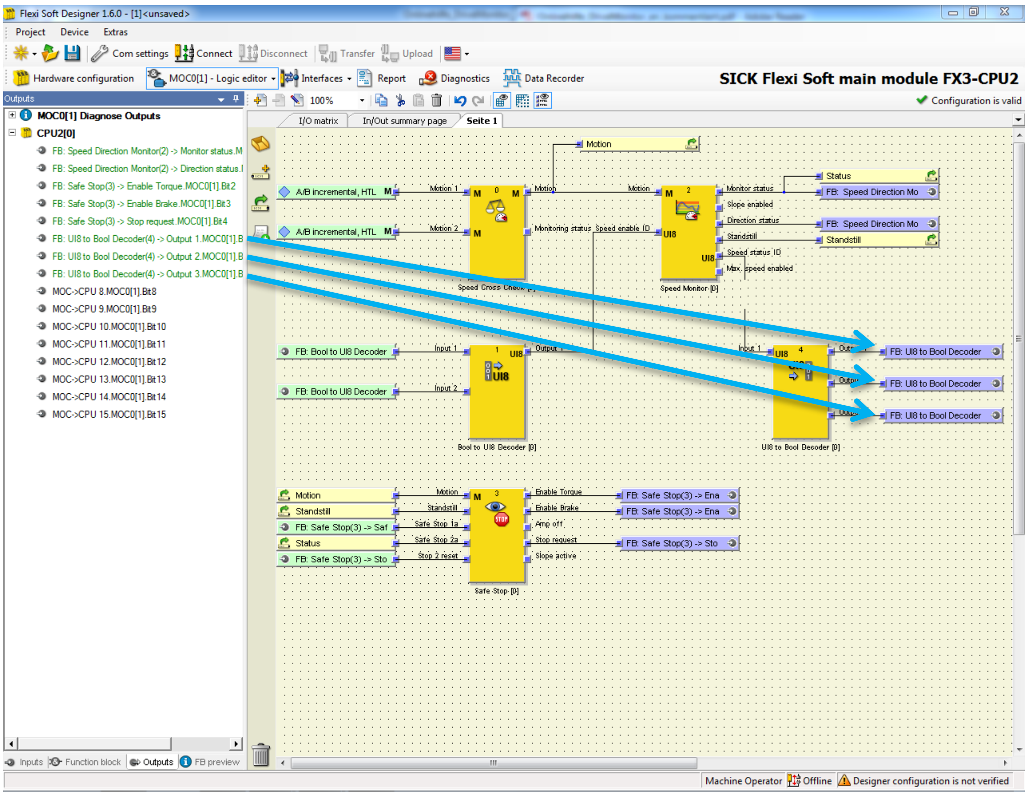


The jump markers "Motion", "Standstill" and "Status" are implemented here.
These are now input signals for the Safe Stop.
Additionally there are two more input signals (Bit 2 and 3), which are output signals in the CPU Logic.
Moreover there are three output signals (bit 2, 3 and 4), which are input signals in the CPU logic.

Step 8: *Placing of the UI8 to Bool Decoder and routing the output signals to the CPU*



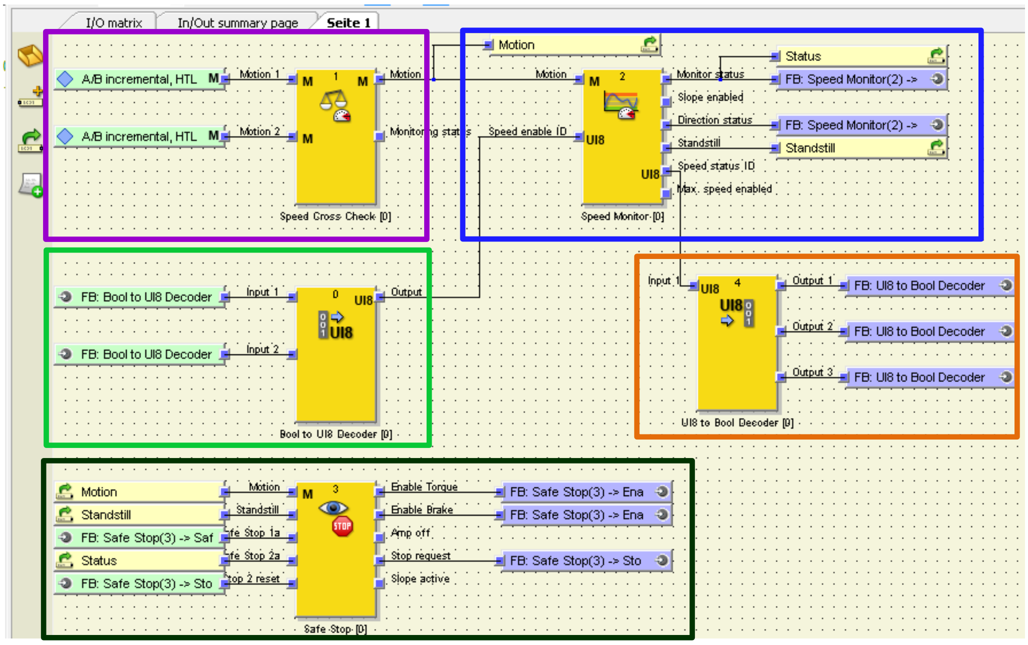
The "UI8 to Bool Decoder" converts the "Speed status ID" into a bit pattern. In this way, the actual speed level can be routed to the CPU. This output data is used as input data in the CPU.



Finally all output signals of the decoder must be connected with the output bits (Bit 5, 6 and 7).

<u>*Summary of the MOC logic*</u>

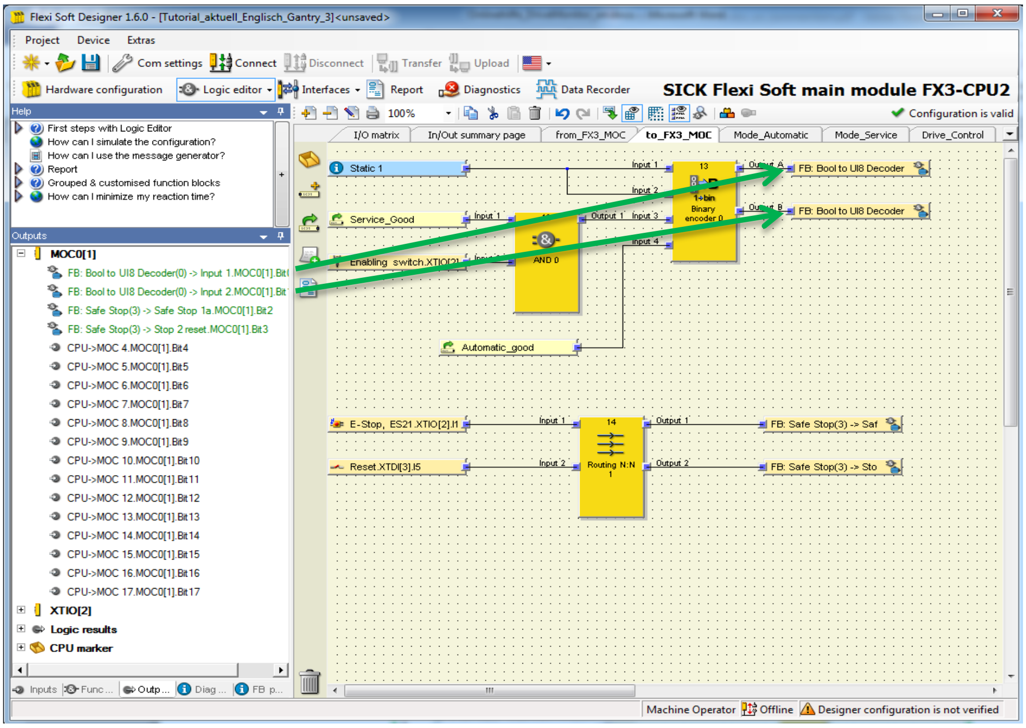
In the following part, the complete configuration and a description of all function blocks can be seen.



Because of the “Speed Cross Check” a higher safety category is given.
The function block compares speed values from two different signal sources.

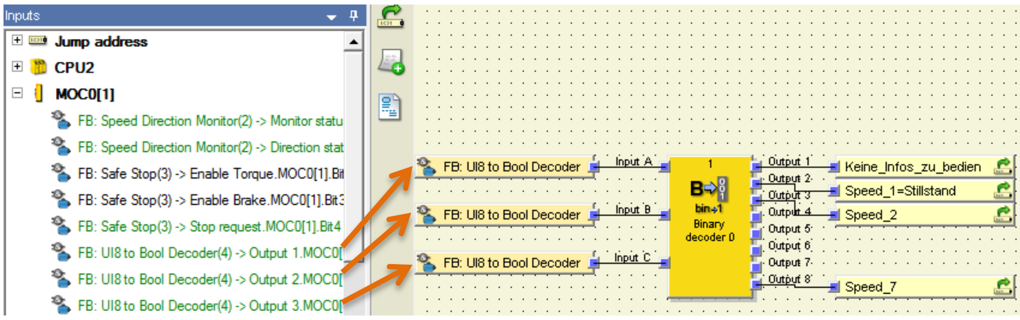
The “Speed Monitor” monitors the speed and routes the actual speed ID to the CPU. Moreover the function block sets the speed limit with the “Speed enable ID” and readout the actual speed via “Speed status ID” as well the actual direction via “Direction status”.

The binary data from the CPU is converted to “Speed enable ID” (UI8 data) by the “Bool to UI8 Decoder”. This enables the speed limit (see image).



Converts the “Speed status ID” into a bit pattern.
Thus the data about the speed can be used in the CPU logic.

The image shows that the output signals from the MOC logic will be used as input signals in the CPU logic. (see chapter 7.3)



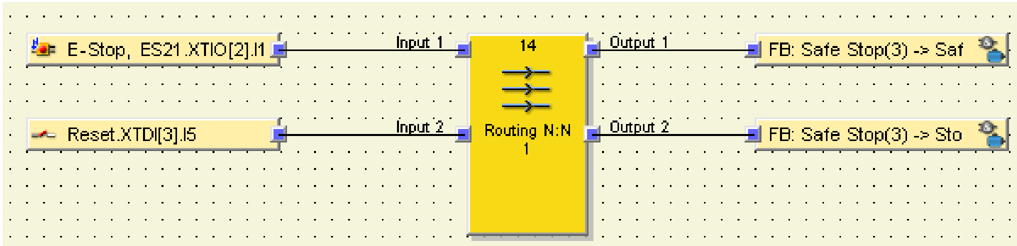
Here the “Safe Stop” function can be realized.

SS1 is activated by a signal from the CPU, if it changes to low (e.g. E-Stop)

SS2 is activated when the “Monitor status” of the “Speed Monitor” changes to low.
The connection between the “Speed Monitor” and the “Safe Stop” is realized through the jump marker “Status”.

The jump marker “Status” can be low because of the reason of an encoder error, cross check error or the speed is over the limit.

The input signals “Safe Stop 1” and “Safe Stop 2 reset” come from the CPU (see image).
The output signals can be used later in the CPU logic. (see chapter 7.3)

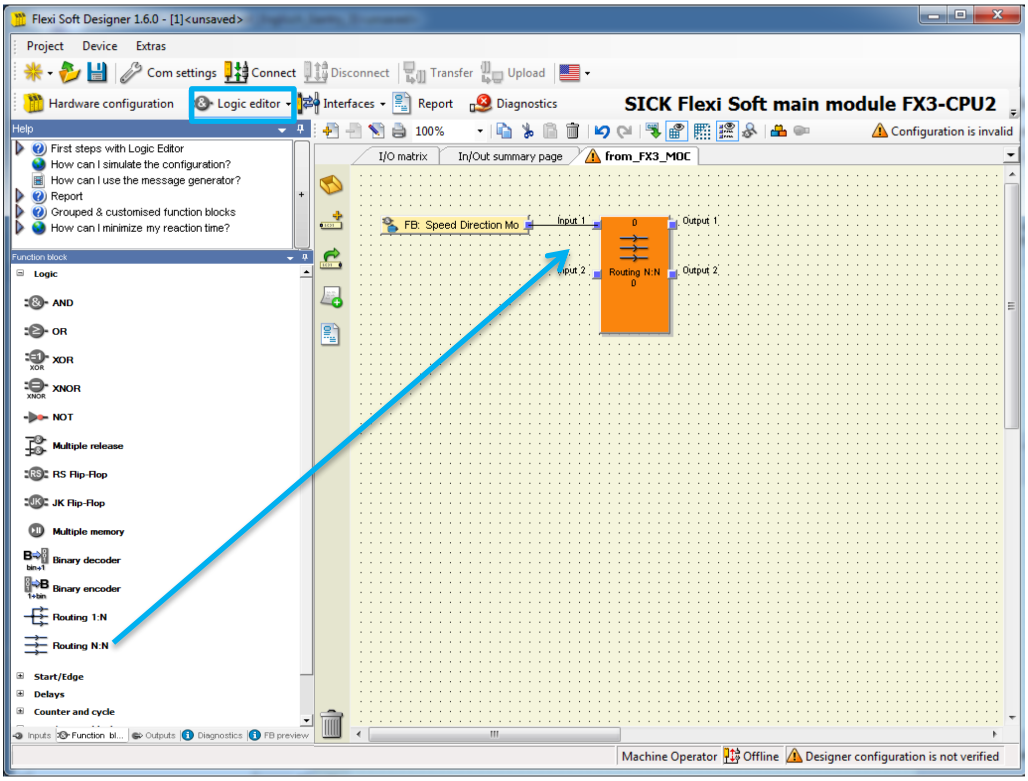


8.3 CPU Logic

Step 1: *Placing of the Routing N:N function block*

Finally, the CPU logic must be configured.
By clicking on "Logic editor" the user reaches the logic of the CPU (see blue border).

At the beginning a "Routing N:N" function clock is required, which can be inserted by Drag & Drop.

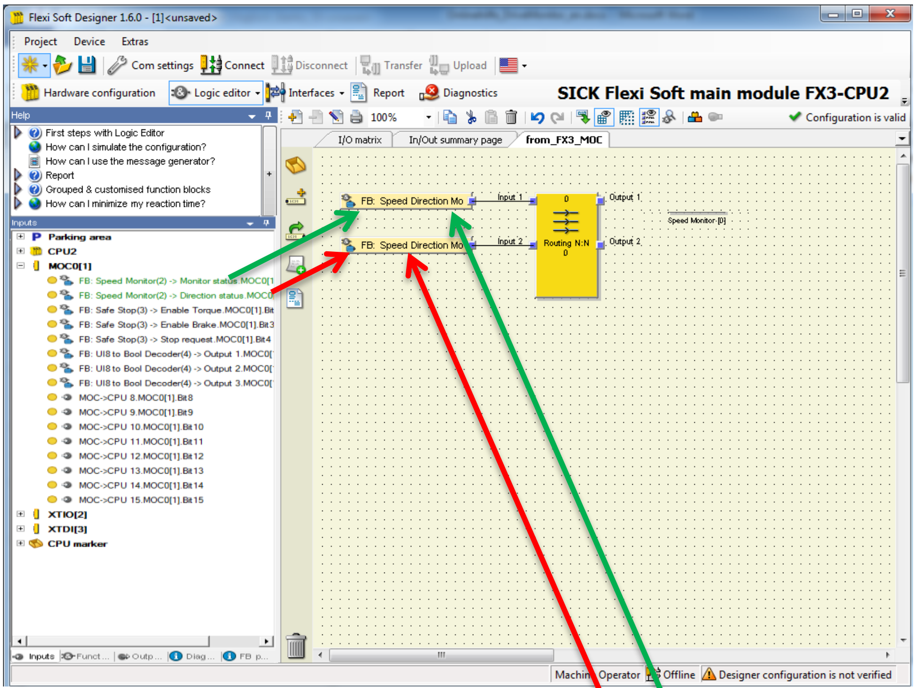


The information from the Drive Monitor receives the "Routing N:N" function block via the backplane bus and routes the infromation internally to jump markers.
The “Routing N:N” function block passes up to eight input signals to up to eight outputs in parallel. This function block allows it to connect input elements (e.g. inputs from a XTDI or XTIO module) one-to-one with output elements.

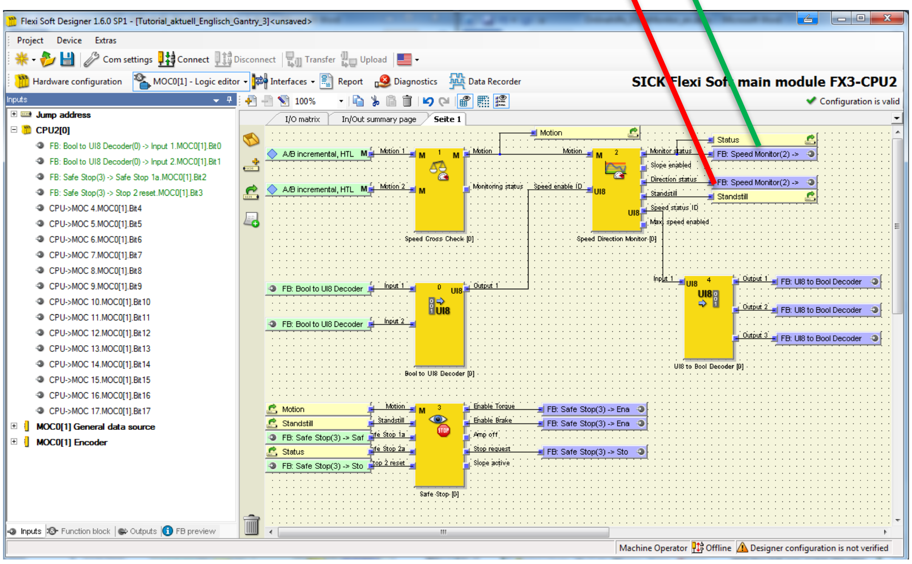
Step 2: *Routing of the Monitor status and Direction status from the MOC0 to the CPU function block*

Now, the information must be involved from the MOC logic in the CPU logic. At first it is necessary to switch the tab from “Function block” to “Inputs”.
In the Flexi Soft logic there is now a modul named MOC[x] where you can find IOs. These are the bits which are declared as output signals in the MOC logic.
The information from the MOC logic is now being used and is available via the backplane bus. The signals from the Drive Monitor (Bit 0 and 1) have to be connected with the Inputs (Drag & Drop). Now it can be seen that the descriptions of the input signals are green, because the signals are connected to the function block. Moreover the function block changes the color from orange to yellow, because it is connected to valid signals of the Drive Monitor.

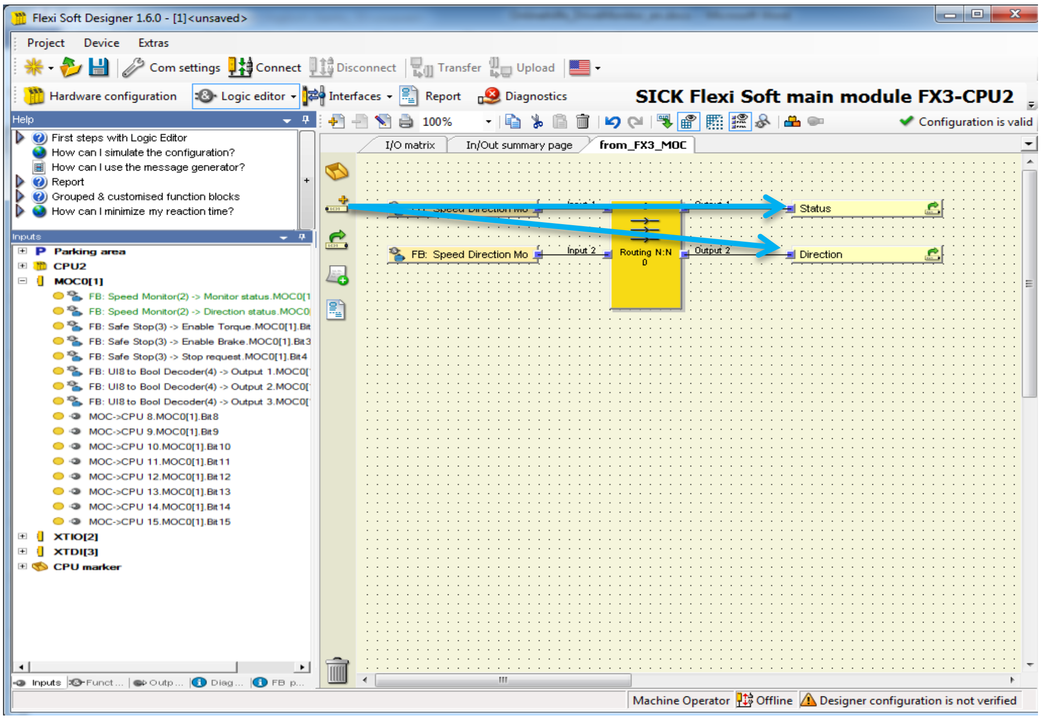
CPU logic:



MOC logic:

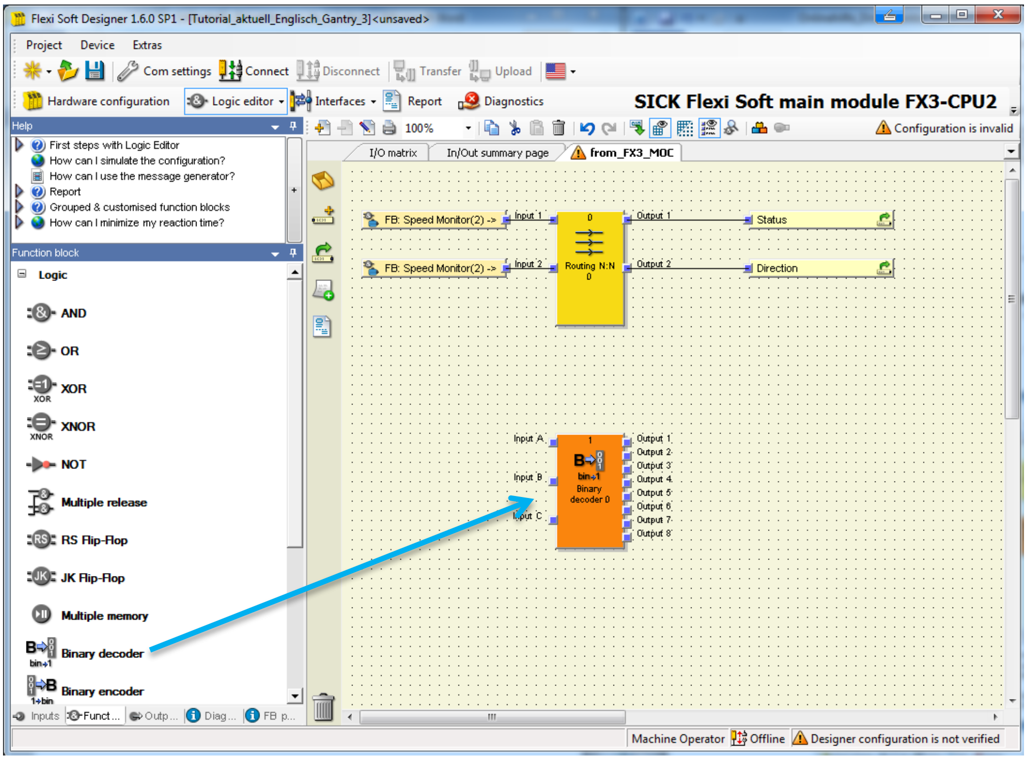


Step 3: *Allocating the outputs with internal jump markers*



Now, the information from the Drive Monitor is routed to internal jump markers. The jump markers can be connected to the function block via Drag &Drop.

<u>Step 4:</u> *Placing the Binary decoder*

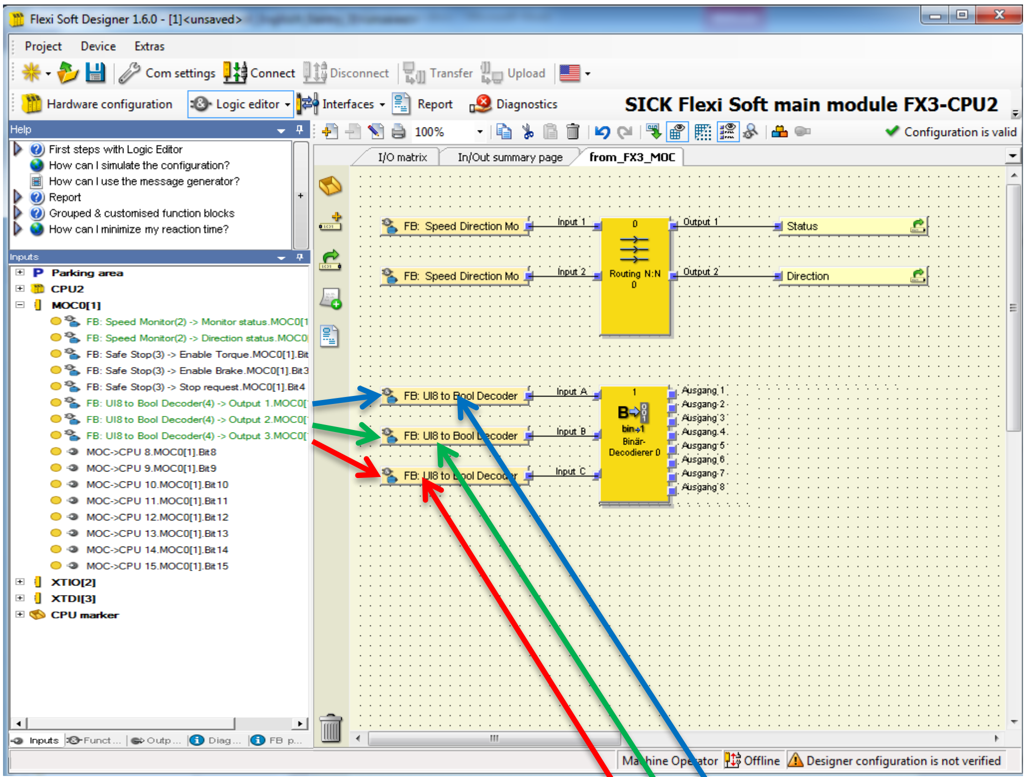


The function block “Binary decoder” decodes dependent on the current configuration a binary code to an one-out-of-N or to a priority code. The "speed status ID" is output as a threshold in the logic. Furthermore, the number of the inputs must be changed to three (double click on the function block → In/Out Settings).

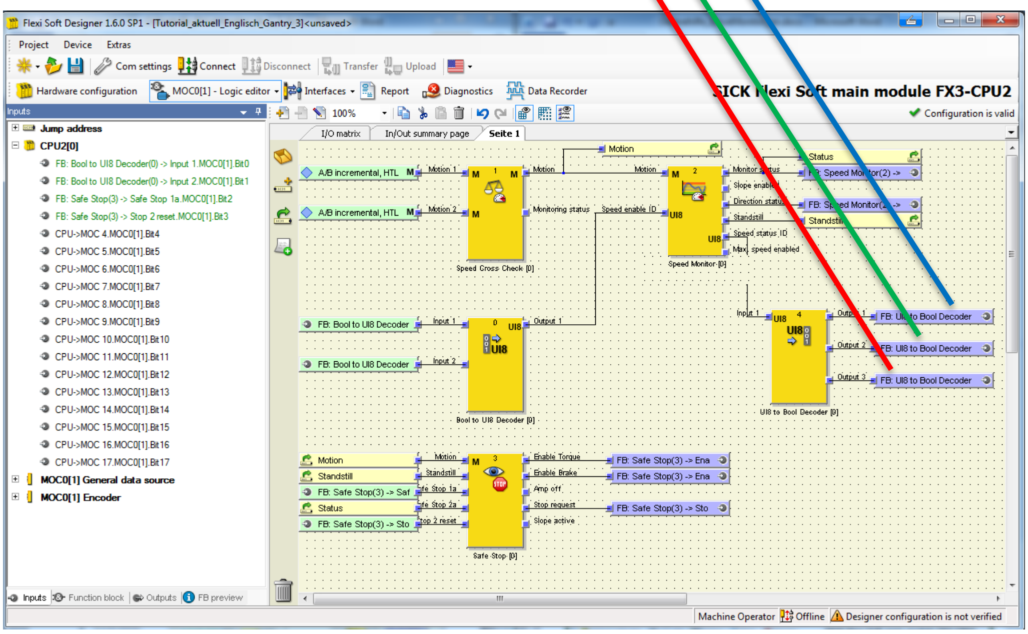
<u>Step 5:</u> *Routing of the issued bool signals from the MOC0*

Now the bits 5, 6 and 7 of the Drive Monitor a required, which are available via the backplane bus. This is necessary to get the converted information from the “Speed status ID” (MOC Logic).

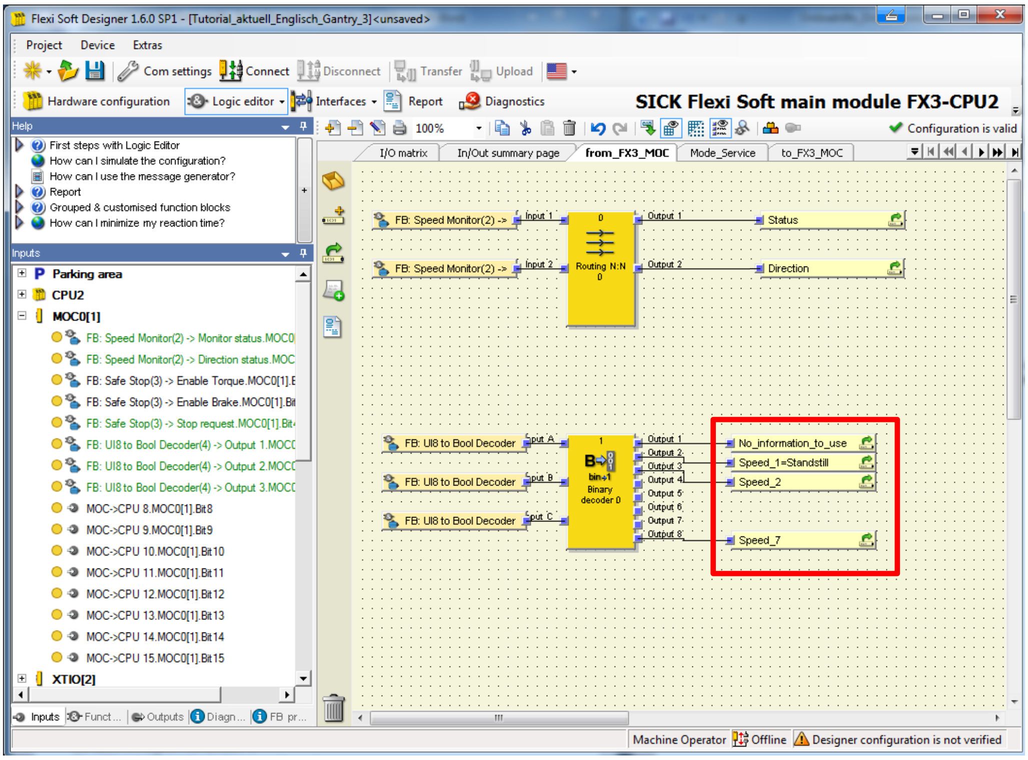
<u>CPU logic:</u>



<u>MOC logic:</u>



<u>Step 6:</u> *Allocating the outputs with internal jump markers*



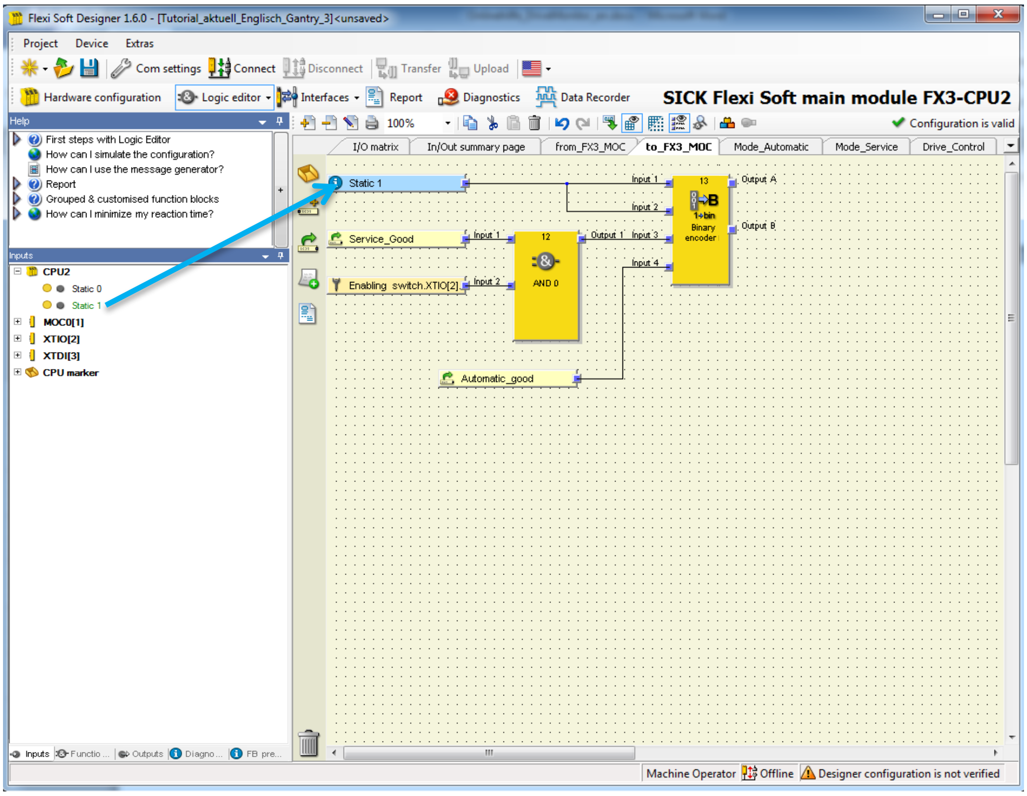
The outputs are occupied with jump markers. Because of that, the information about the speed is available.

Over the defined jump adresses it is possible anytime to use the decoded information in other parts of the program. This applies especially for the jump marker “Speed_1=Standstill”. This information is provided by the MOC.

Step 7: *Creating a logic, which will send signals to the MOC0*

In the following part an incomplete configuration can be seen, which sends the information to the Drive Monitor.

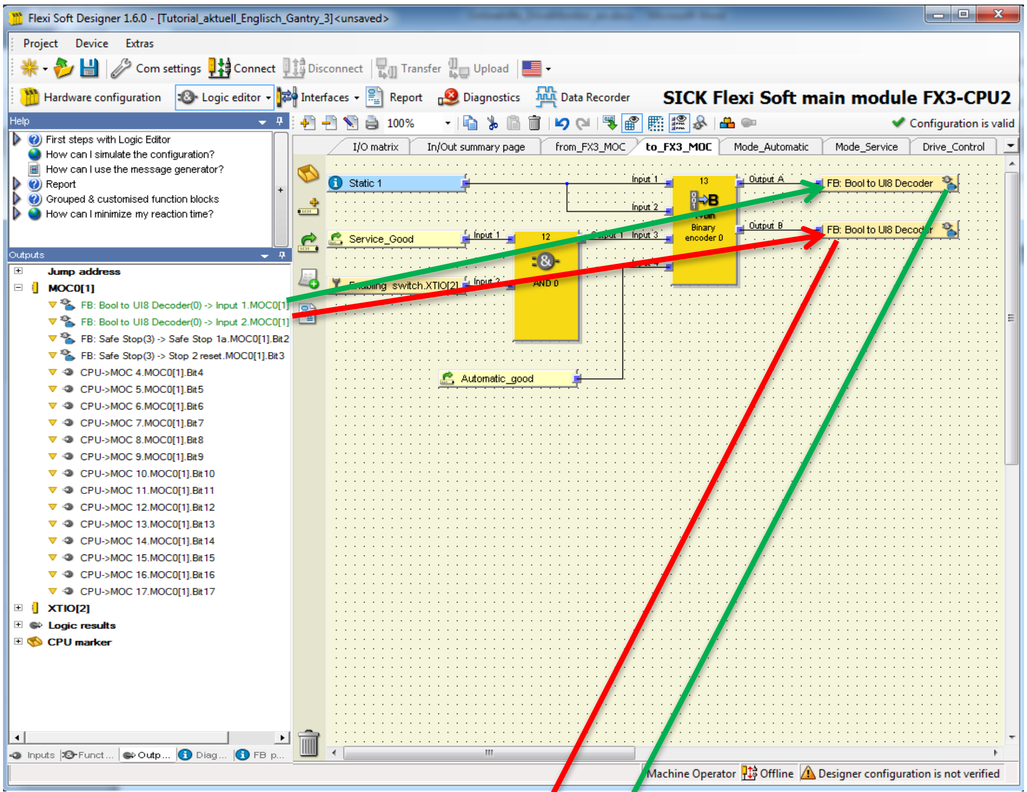
The input "Static 1" can be used to set a function block input permanently to 1 (High). This may be necessary to get a valid logic configuration, if the function block contains inputs which are not used but also can not be disabled.
With Drag & Drop the signal can be connected to the input of the binary encoder.
But it is necessary to pay attention at the parameter settings (double click on the function block). In this case, it works with the mode "Priority".



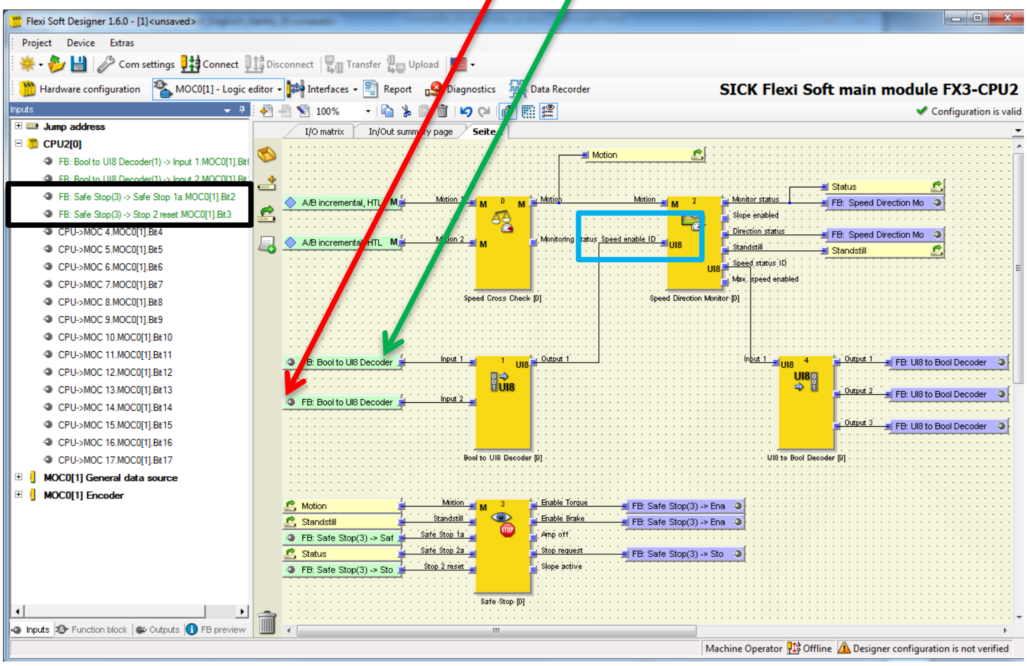
Step 8: *Routing the output singals to the MOC0*

Now the outputs are occupied with the Bit 0 and Bit 1. These signals will be sent to the Drive Monitor to activate a defined speed.

CPU logic:



MOC logic:

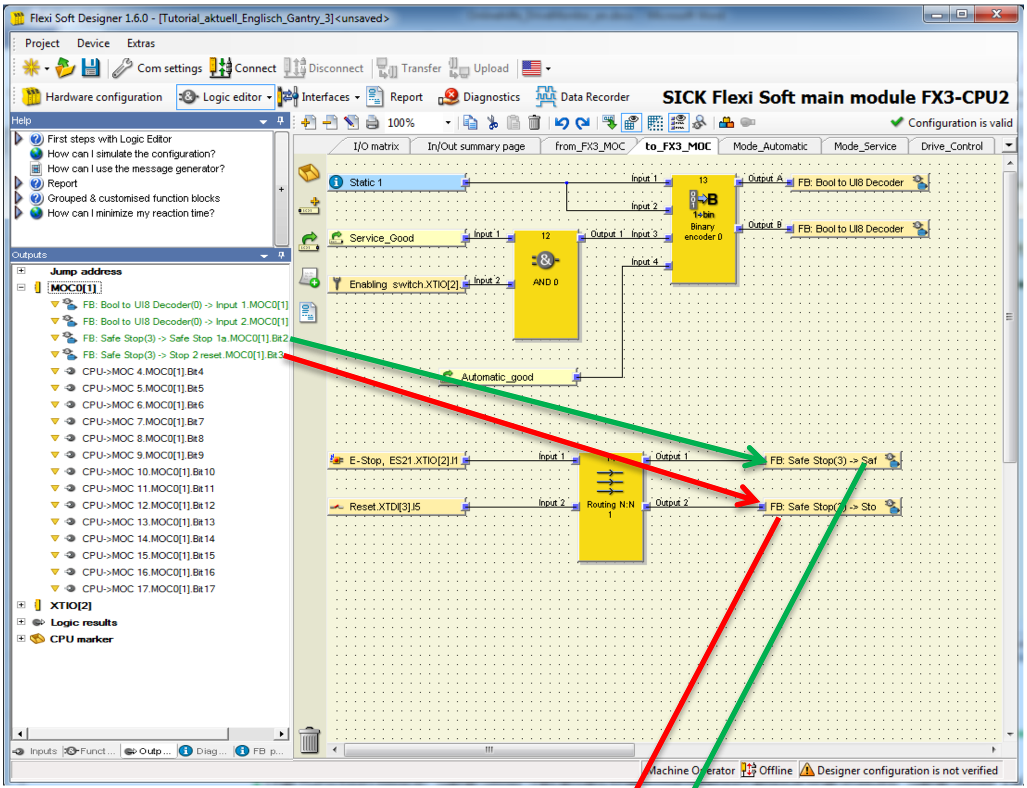


Finally the Speed Monitor receives the information about the input “Speed enable ID”.

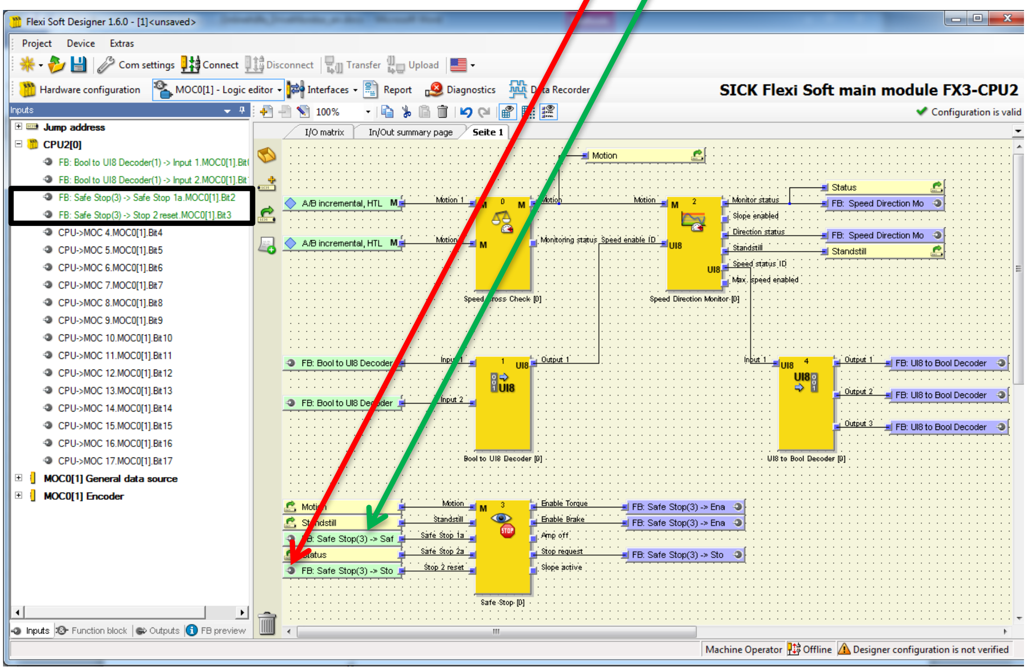
Step 9: *Creating the logic for the Safe Stop and routing the output signals to the MOC0*

At the end, a "Routing N:N" function block is still required. The inputs get the signals from the moduls “XTIO” (E-Stop) and “XTDI” (Reset). This function block is necessary for the “Safe Stop” in the MOC logic. The outputs are connected with the Bit 2 and Bit 3.

CPU logic:



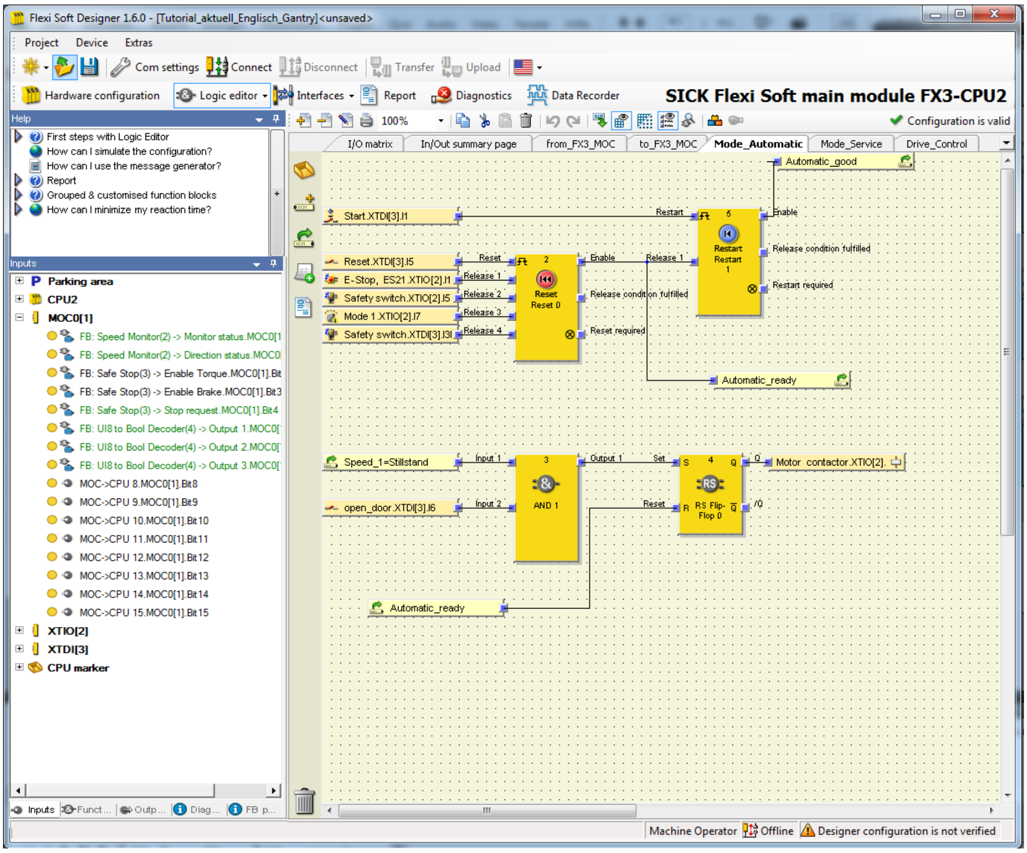
MOC logic:

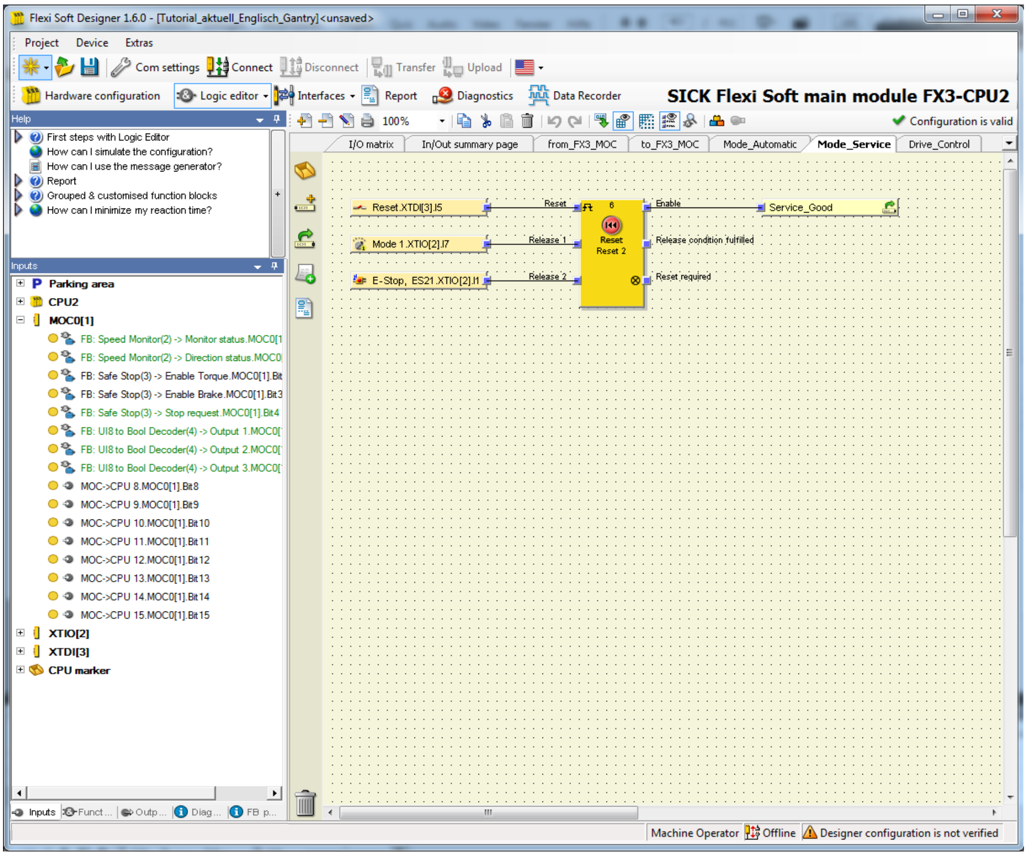


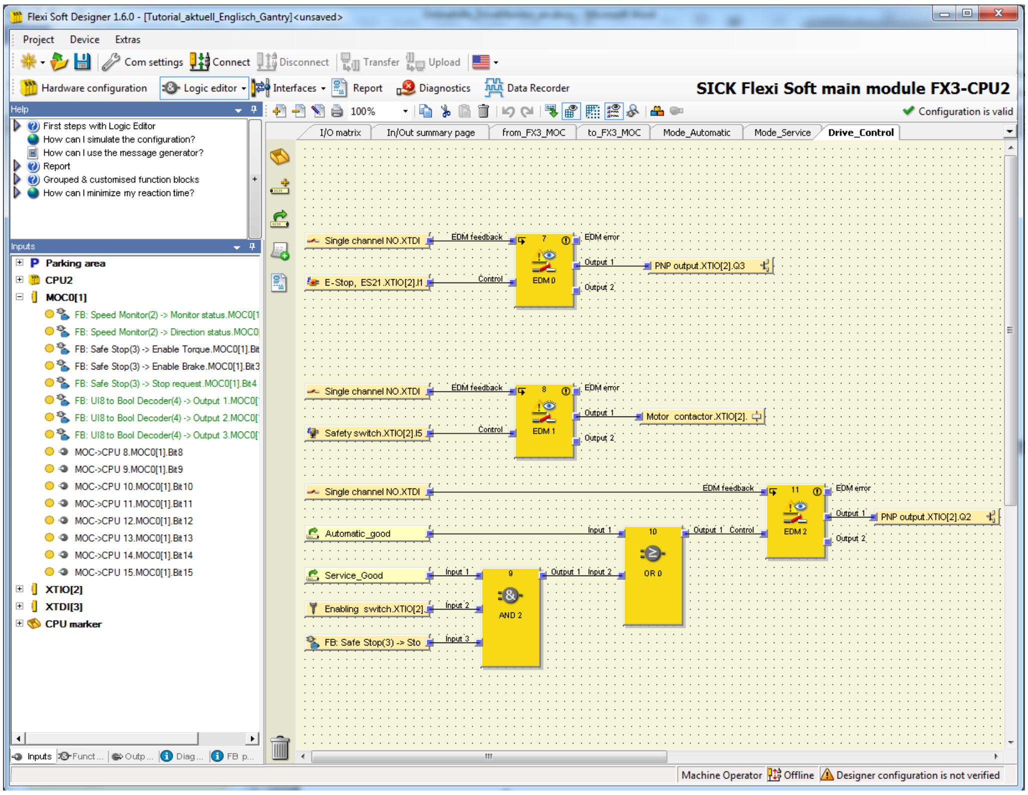
These signals will be send to the Drive Monitor. The function block “Safe Stop” receives the information over the inputs “Safe Stop 1a” and “Stop 2 reset”.

In the following section is an individual configuration to see:

<u>CPU logic:</u>







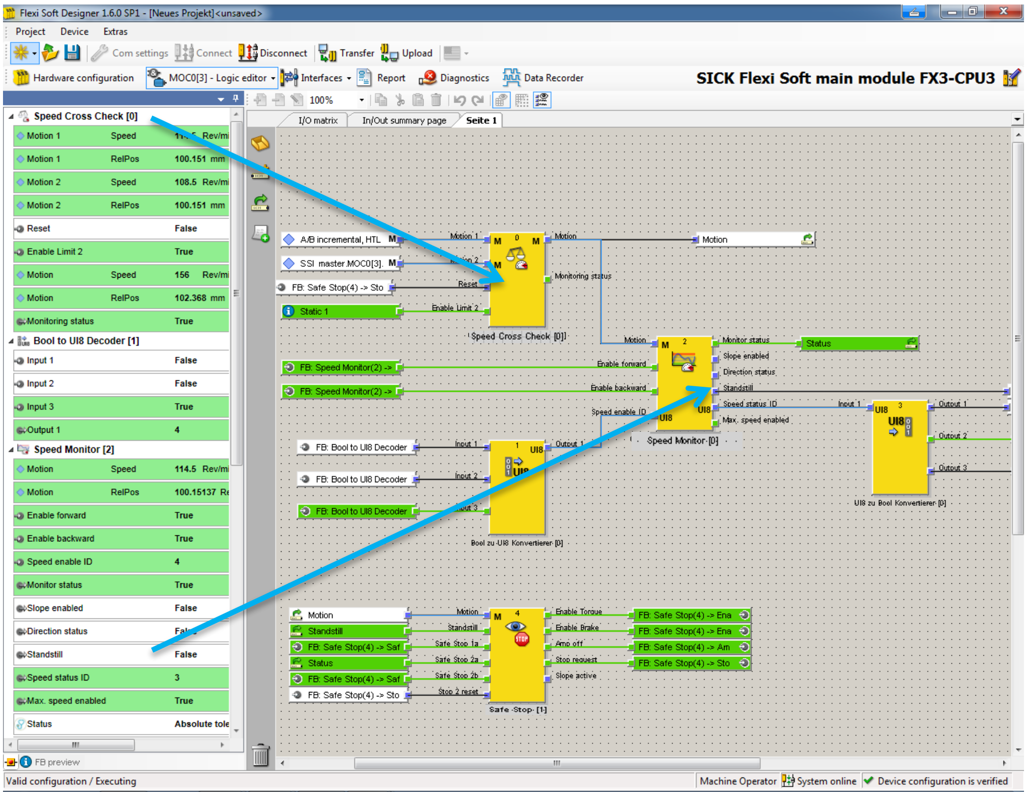
Now the logic for the Gantry is configured and can be saved and transferred to the Flexi Soft station.

This is an individual logic configuration.
It is not transferable to every situation.

9 Diagnostic

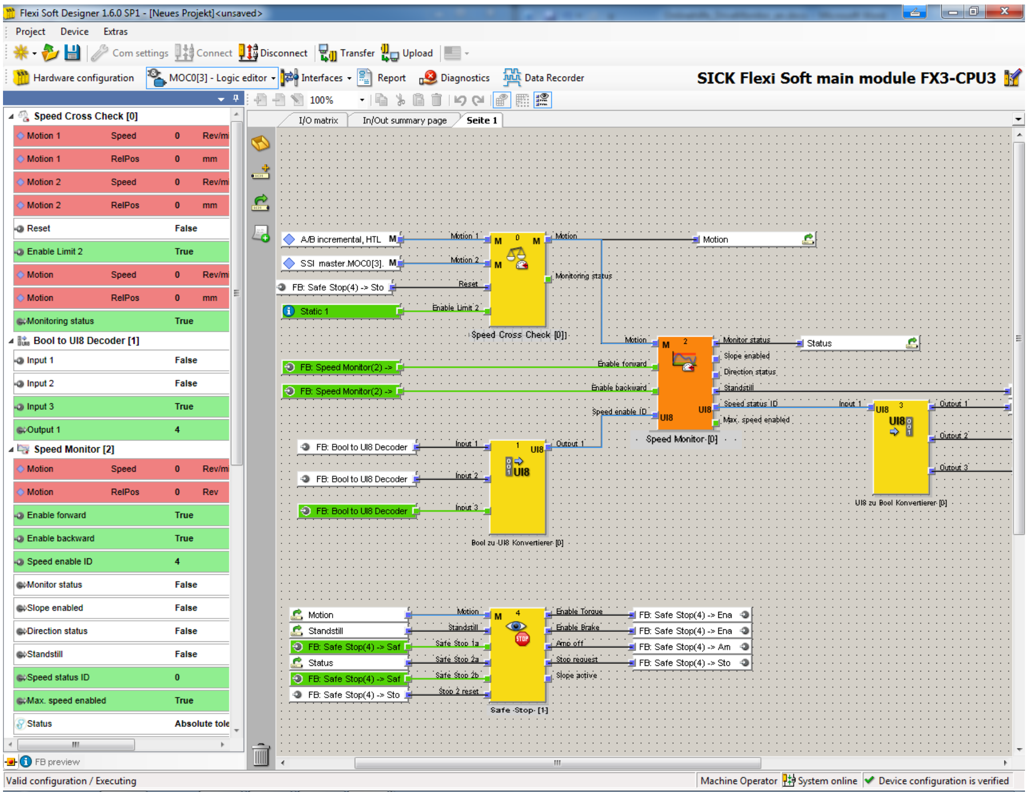
After the configuration of the two logics, the System can be connected to the Hardware. For the diagnostic it is necessary to be online and not to switch into the online edit mode.

Now, the screen below should be seen. But the content is not transferable to every application.

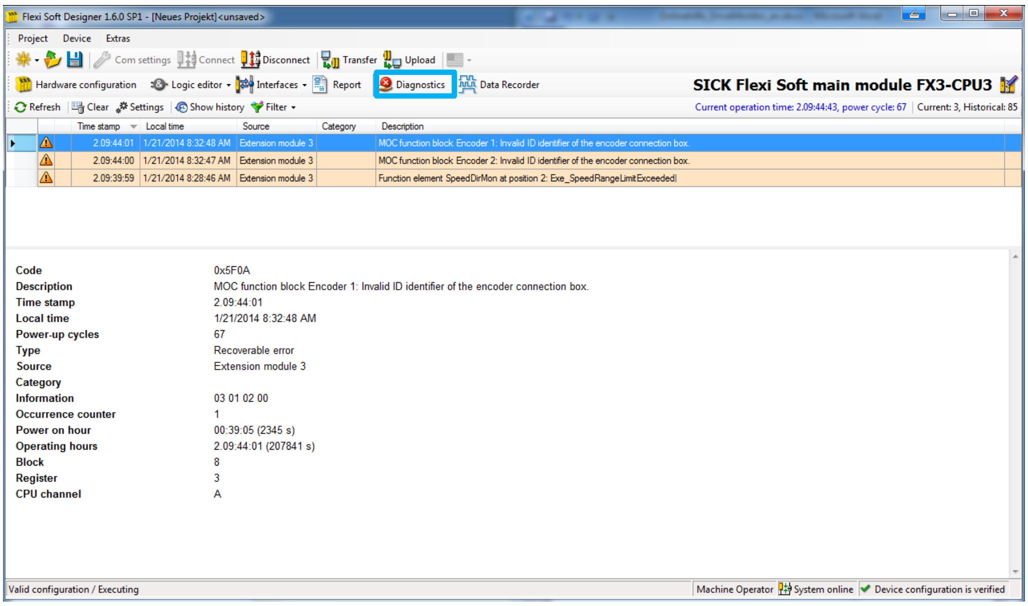


On the right side of the screen is the individually programmed MOC logic continuously to see. There is a distinction between green and white signals.
Binary data are colored green, if the signals are HIGH. If the signals are white, the outputs or inputs are LOW. This coloring is valid only for binary data.
Because the encoder signals do not provide binary data, they are permanently white.

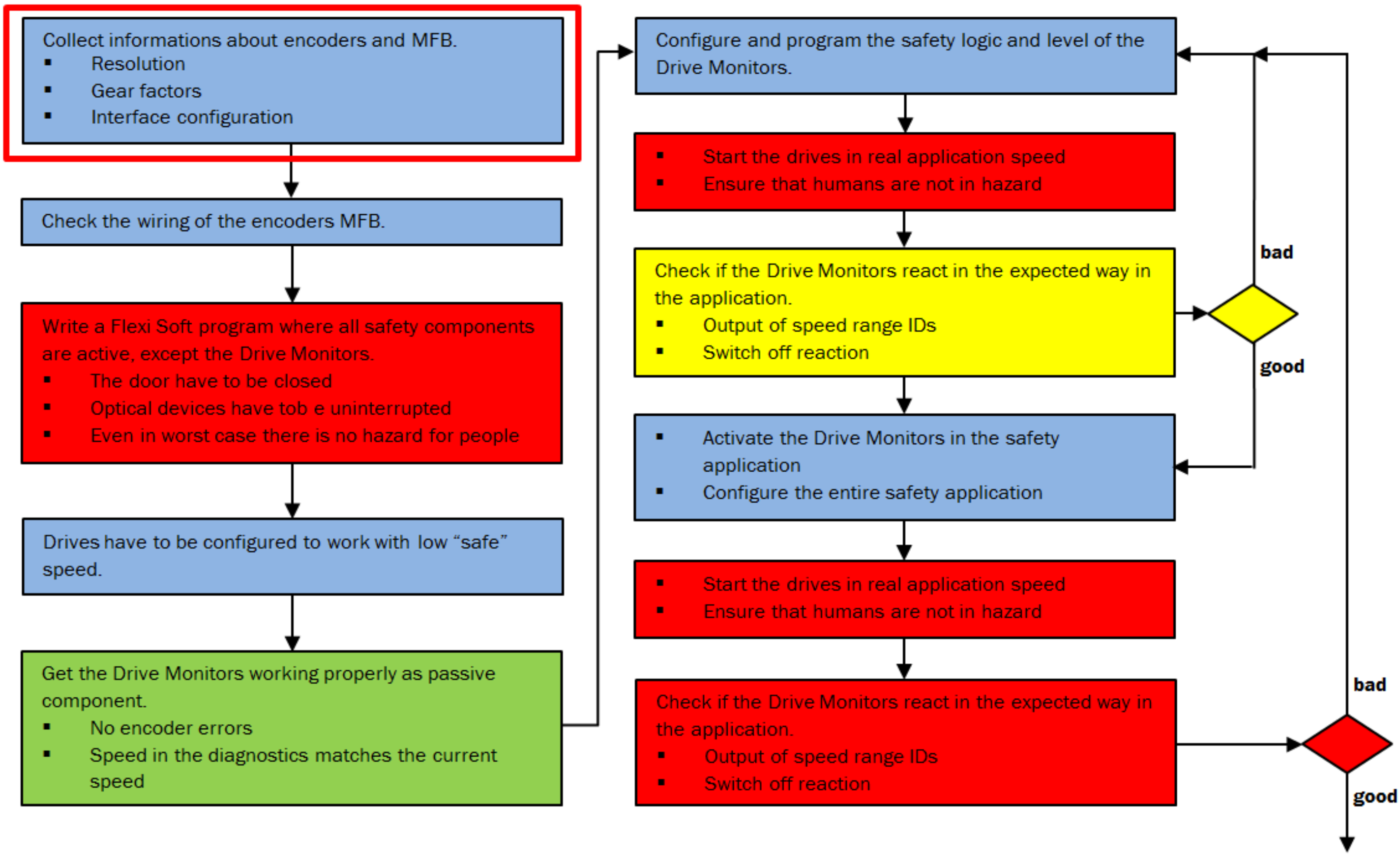
On the left side can be seen a diagnosis window.
Each functional block is marked with the name and a number.
Below the function blocks, all inputs and outputs are listed. Because of that, it is possible to see which values the signals have got.
If a signal is green, then it will be "TRUE" and it works.
Should a signal be white, then it is "False" and is not operated at the moment.
A red coloring is a fault and should be corrected.

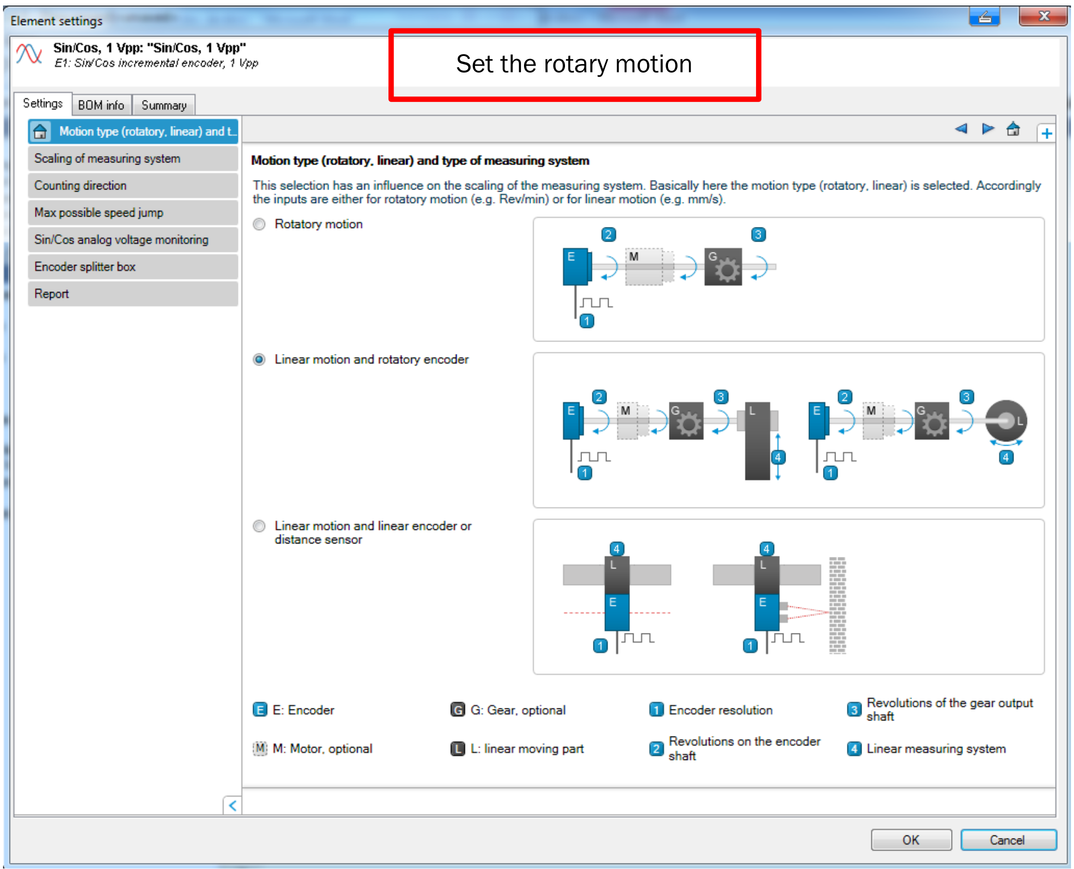


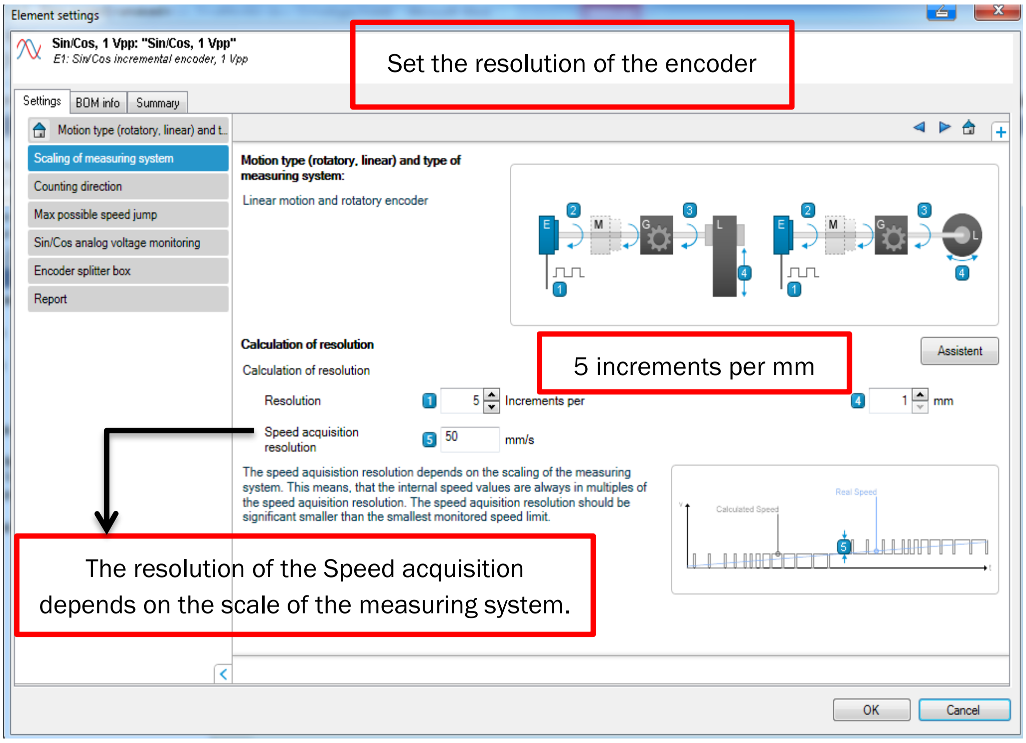
A further assistance could provide the view “Diagnostics” (blue box below), where a complete history of all messages, informations, warnings and error messages of a connected Flexi Soft safety controller is available in the upper part of the window.
By clicking on an entry in the list, details of the selected message are displayed in the lower part of the window.

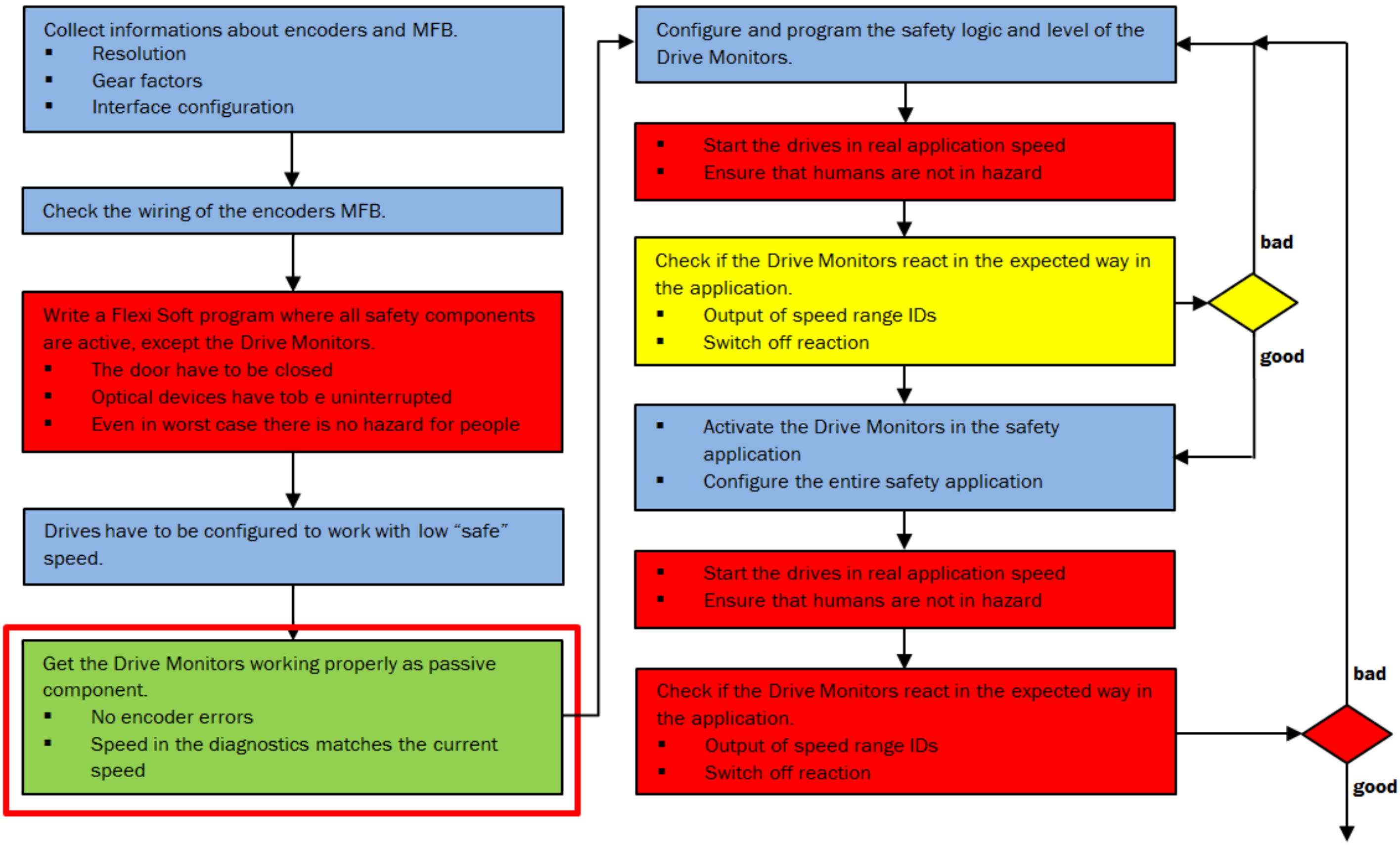


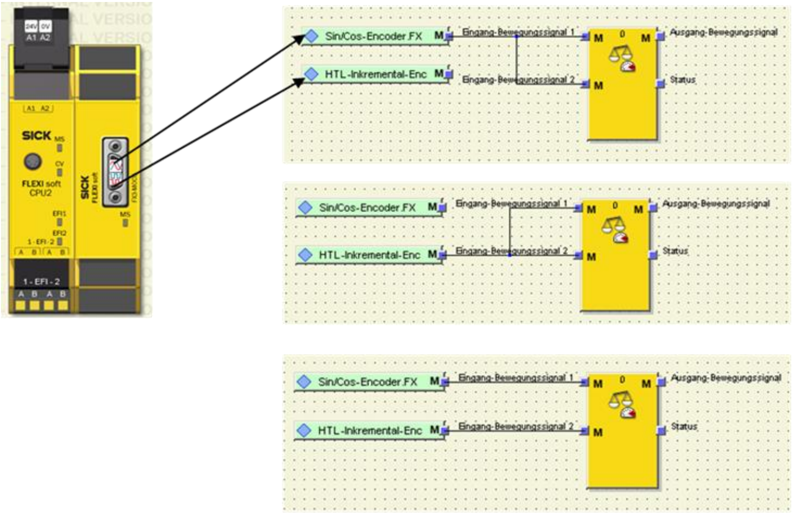
10 How to get the system running






Element settings
Sin/Cos, 1 Vpp: "Sin/Cos, 1 Vpp"
E1: Sin/Cos incremental encoder, 1 Vpp
Set the resolution of the encoder
Settings
BOM info
Summary
Motion type (rotatory, linear) and t...
Scaling of measuring system
Counting direction
Max possible speed jump
Sin/Cos analog voltage monitoring
Encoder splitter box
Report
Motion type (rotatory, linear) and type of measuring system:
Linear motion and rotatory encoder
Calculation of resolution
Calculation of resolution
5 increments per mm
Assistent
Resolution
5 Increments per
1 mm
Speed acquisition resolution
50 mm/s
The speed aquisition resolution depends on the scaling of the measuring system. This means, that the internal speed values are always in multiples of the speed aquisition resolution. The speed aquisition resolution should be significant smaller than the smallest monitored speed limit.
Real Speed
Calculated Speed
The resolution of the Speed acquisition depends on the scale of the measuring system.
OK
Cancel

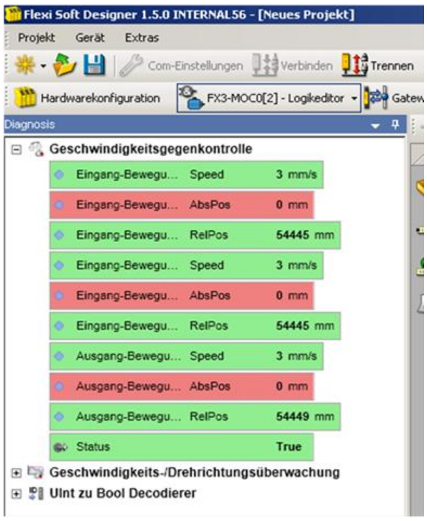

Collect informations about encoders and MFB.
▪ Resolution
▪ Gear factors
▪ Interface configuration
Check the wiring of the encoders MFB.
Write a Flexi Soft program where all safety components are active, except the Drive Monitors.
▪ The door have to be closed
▪ Optical devices have tob e uninterrupted
▪ Even in worst case there is no hazard for people
Drives have to be configured to work with low "safe" speed.
Get the Drive Monitors working properly as passive component.
▪ No encoder errors
▪ Speed in the diagnostics matches the current speed
Configure and program the safety logic and level of the Drive Monitors.
▪ Start the drives in real application speed
▪ Ensure that humans are not in hazard
Check if the Drive Monitors react in the expected way in the application.
▪ Output of speed range IDs
▪ Switch off reaction
bad
good
▪ Activate the Drive Monitors in the safety application
▪ Configure the entire safety application
▪ Start the drives in real application speed
▪ Ensure that humans are not in hazard
Check if the Drive Monitors react in the expected way in the application.
▪ Output of speed range IDs
▪ Switch off reaction
bad
good

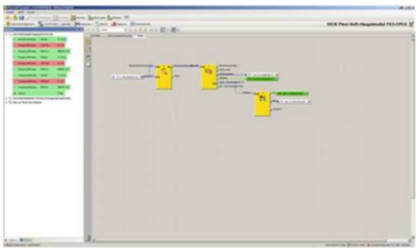


Encoder 1

Encoder 2

Encoder 1
+
Encoder 2





Application

11 Frequent error causes

11.1 General error causes

11.1.1 Mechanics errors

A typical error is an incorrectly connected encoder. The more accurate the centering for the encoder, the lower are the angular offset and shaft offset after the installation and the smaller is the stress applied to the torque arms and bearing of the Encoder.
In order not to burden the torque arm during the installation, always attach at first the encoder and then the clamping ring of the hollow shaft clamping.

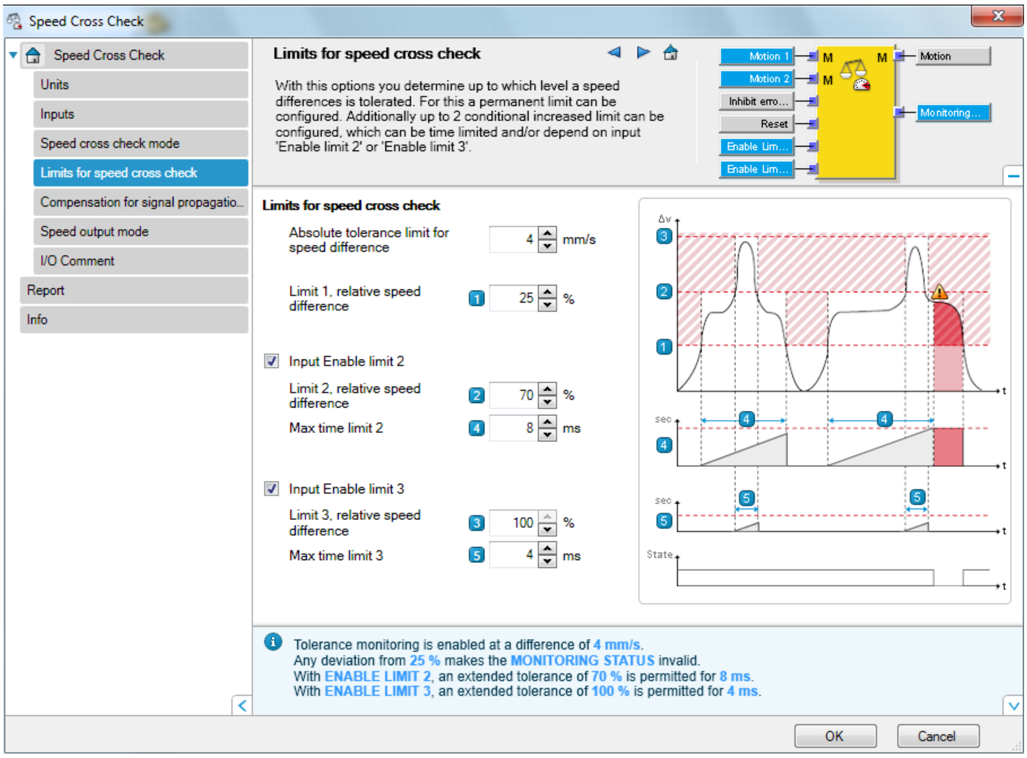
11.1.2 Electromagnetic Compatibility (EMC)

Another error cause could be the shielding. This would entail that electromagnetic fields distort the result. With regard to the EMC guidelines it is important that the cover or the cable shield is connected to the earth or the ground.
More information is available in the EMC guidelines.

11.2 Speed Cross Check

If there are any problems with the function block „Speed Cross Check“, it may be due to an incorrect setting of the "limits of speed cross check". On the basis of e.g. slip, friction, mechanical coupling behavior etc., continuous and also temporary differences between the two measured values may occur. For this reason the function block provides various parameters that can be used to compensate such differences.
In this way, error shutdowns can be avoided and the machine availability can be ensured.

In the following is a possible solution for the problem to see:



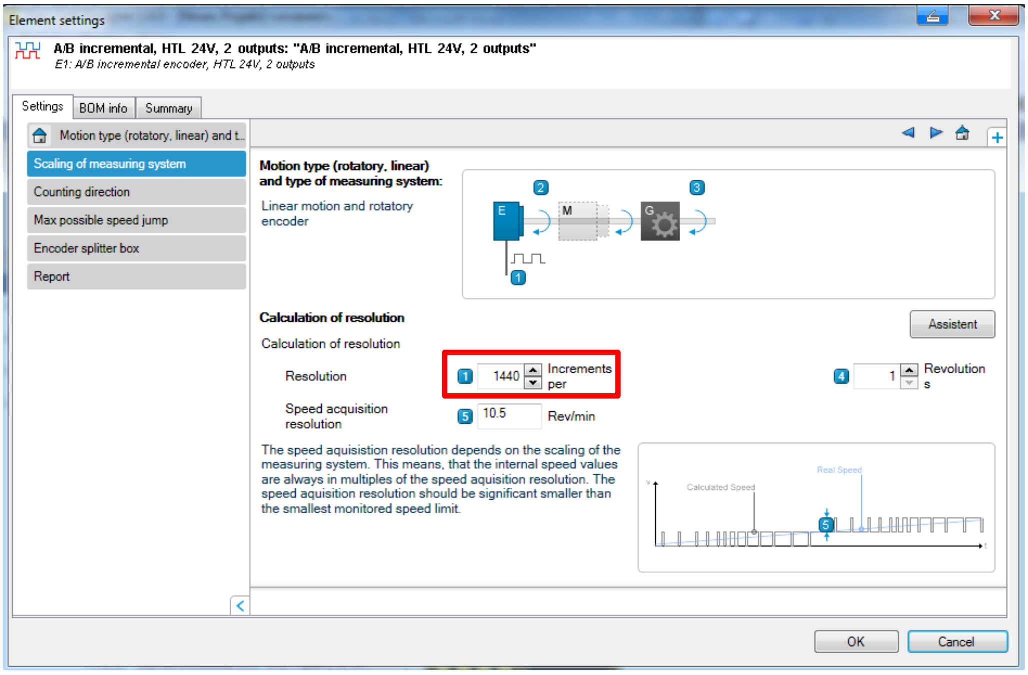
By changing the parameters, the configuration also must be adjusted. The problem can also be solved by an improved construction, if there is an alternative.

WARNING

Finally, it is necessary to create a risk analysis for the machine. The risk assessment must confirm that no hazard condition is caused by the selected configuration, i.e. that sensor errors are detected sufficiently fast.

11.3 Faulty hardware configuration

The hardware configuration may be a further error cause. Often, the encoders are configured incorrectly. First of all, the resolution should be checked because there may happen elementary errors for the system.



12 Detailed technical data

Safety-related parameters

For axes with two encoders (TTL, HTL, RS-422, Sin/Cos, SSI)

Safety integrity level	SIL3 (IEC 61508) SILCL3 (EN 62061)
Category	Category 4 (EN ISO 13849-1)
Performance level	PL e (EN ISO 13849-1)
PFHd (mean probability of a dangerous failure per hour)	5.0 * 10^{-9} (EN ISO 13849)
Minimum movement for error detection	≥ selected tolerance limit for the function block used for the cross check, e.g. speed cross check, at least 1 × within 24 h
T_M (mission time)	20 years (EN ISO 13849)

Activated for axes with one Sin/Cos encoder and Sin/Cos voltage monitoring

Safety integrity level	SIL2 (IEC 61508) SILCL2 (EN 62061)
Category	Category 3 (EN ISO 13849-1)
Performance level	PL d (EN ISO 13849-1)
PFHd (mean probability of a dangerous failure per hour)	6.0 * 10^{-9} (EN ISO 13849)
Minimum movement for error detection	≥ 1 Sin/Cos period, at least 1 × within 24 h
T_M (mission time)	20 years (EN ISO 13849)

Functions

Drive safety functions	
Safe stop 1 (SS1)	✔
Safe stop 2 (SS2)	✔
Safe operating stop (SOS)	✔
Safe speed monitoring (SSM)	✔
Safe limited speed (SLS)	✔
Safe direction of motion (SDI)	✔
Safe brake control (SBC)	✔

Interfaces

Connection type	Male connector, Micro D-Sub, 15-pin
Encoder interface	A/B incremental encoder, TTL A/B incremental encoder, HTL A/B incremental encoder, RS-422 Sin/Cos encoder SSI encoder
Data interface	Internal bus (FLEX BUS+)

Electrical data

Operating data

Protection class	III (EN 61140)
Power consumption	≤ 2.5 W [1]
Output voltage for encoder	24 V

[1] Via FLEXBUS+, without encoder voltage supply.

A/B incremental encoder, TTL, 2 outputs

Input resistance	≥ 35 kΩ
Input voltage HIGH	5 V (2 V ... 5.3 V)
Input voltage LOW	0 V (–0.3 V ... 0.8 V)
Input frequency	≤ 300 kHz

A/B incremental encoder, TTL, 2 output pairs

Input resistance	≥ 35 kΩ
Input voltage HIGH	5 V (1.2 V ... 5.6 V)
Input voltage LOW	–5 V (–5.6 V ... –1.2 V)
Input voltage common mode	–10 V ... 10 V
Input frequency	≤ 300 kHz

A/B incremental encoder, HTL 12 V, 2 outputs

Input resistance	≥ 35 kΩ
Input voltage HIGH	12 V (6.5 V ... 15 V)
Input voltage LOW	0 V (–1 V ... 2.5 V)
Input frequency	≤ 300 kHz

A/B incremental encoder, HTL 12 V, 2 output pairs

Input resistance	≥ 35 kΩ
Input voltage HIGH	12 V (4 V ... 15 V)
Input voltage LOW	–12 V (–15 V ... –4 V)
Input voltage common mode	–10 V ... 10 V
Input frequency	≤ 300 kHz

A/B incremental encoder, HTL 24 V, 2 output pairs

Input resistance	≥ 35 kΩ
Input voltage HIGH	24 V (8 V ... 30 V)
Input voltage LOW	–24 V (–30 V ... –8 V)
Input voltage common mode	–10 V ... 10 V
Input frequency	≤ 300 kHz

A/B incremental encoder, RS-422

Differential resistance	≥ 35 kΩ
Input voltage HIGH	0.2 V ... 5 V
Input voltage LOW	–5 V ... –0.2 V
Input voltage common mode	–7 V ... 7 V
Input frequency	≤ 1,000 kHz

Sin/Cos encoder

Input resistance	1 kΩ (0.9 kΩ ... 1.1 kΩ)
Differential input voltage	1 V (0.8 V ... 1.2 V)
Input voltage common mode	–10 V ... 10 V
Input frequency	≤ 120 kHz
Sin/Cos voltage monitoring, lower limit for vector length monitoring	0.5 V (0.45 V ... 0.6 V)
Sin/Cos voltage monitoring, upper limit for vector length monitoring	1.25 V (1.2 V ... 1.4 V)

SSI encoder

Differential resistance	120 Ω (100 Ω ... 150 Ω)
Clock frequency	100 kHz ... 1,000 kHz
Cycle gaps between the data packages (monoflop time)	≥ 100 µs
Data bits per frame	16 ... 62

Mechanical data

Dimensions (W x H x D)	22.5 mm x 96.5 mm x 120.8 mm
Weight	120 g

Ambient data

Enclosure rating	
Clamps	IP 20 (EN 60529)
Housing	IP 40 (EN 60529)
Ambient operating temperature	–25 °C ... +55 °C
Storage temperature	–25 °C ... +70 °C
Air humidity	10 % ... 95 %, non-condensing
Electromagnetic compatibility (EMC)	Class A (EN 61000-6-2, EN 55011)
Vibration resistance	1 g, 5 Hz ... 150 Hz (EN 60068-2-6) 3g RMS, 10 Hz ... 500 Hz (EN 60068-2-64)
Shock resistance	
Continuous shock	10 g, 16 ms (EN 60068-2-27)
Single shock	30 g, 11 ms (EN 60068-2-27)

13 Accessories

Modules

Connection modules

Description	Model name	Part no.
Facility for connecting one encoder (normally used in conjunction with a motor feedback encoder). Connection to Drive Monitor FX3-MOC: Female connector, D-Sub HD, 15-pin. Connection to a second motor feedback splitter box: Female connector, D-Sub, 9-pin.	Motor feedback splitter box	2068728
Facility for connecting two encoders. Connection to Drive Monitor FX3-MOC: Female connector, D-Sub HD, 15-pin.	Dual encoder connection box	2068729

Plug connectors and cables

Connecting cable (female connector-open)

Connection type head A	Connection type head B	Cable length	Usage	Model name	Part no.
Female connector, Micro D-Sub, 15-pin, straight	Open conductor heads	2 m	For direct encoder connection	Connecting cable	2067893

Connection cable (male-female connector)

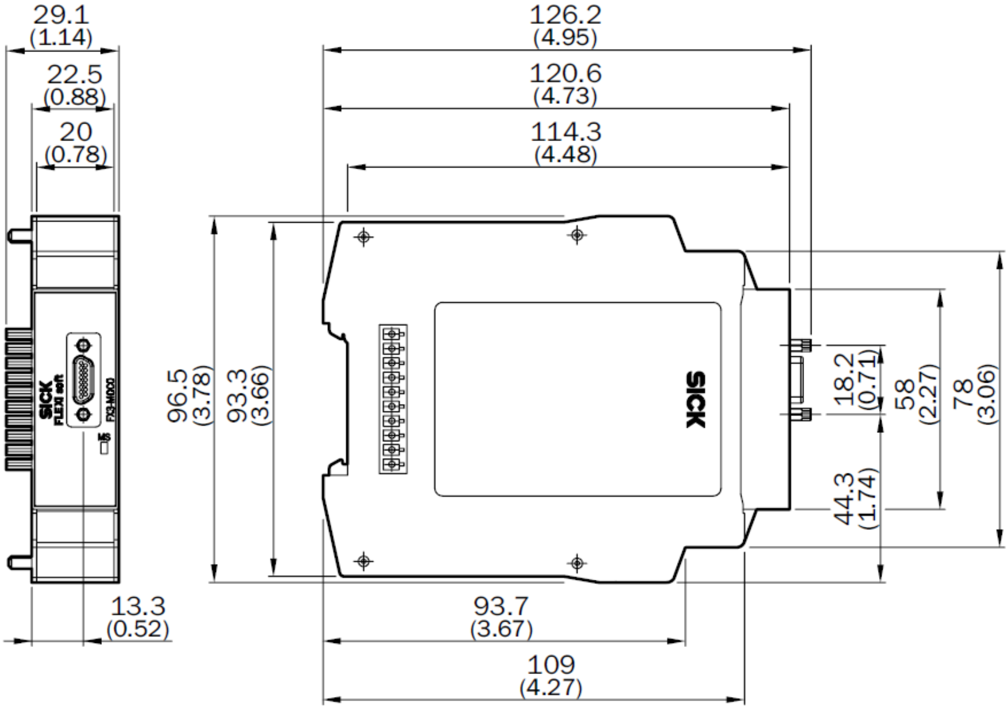
Connection type head A	Connection type head B	Cable length	Usage	Model name	Part no.
Male connector, D-Sub HD, 15-pin, straight	Female connector, Micro D-Sub, 15-pin, straight	2 m	To connect Drive Monitor FX3-MOC with motor feedback splitter box or dual encoder connection box	Connection cable	2067798
		10 m		Connection cable	2067799

Connection cable (male connector-male connector)

Connection type head A	Connection type head B	Cable length	Usage	Model name	Part no.
Male connector, D-Sub HD, 15-pin, straight	Male connector, D-Sub, 9-pin, straight	2 m	To connect two motor feedback splitter boxes with each other	Connection cable	2067800
		10 m		Connection cable	2067801

14 Further informations

14.1 Dimensional drawing



14.2 Ordering information

Description	Type	Part no.
Drive Monitor FX3-MOC	FX3-MOC00000	1062344

