



Function Blocks for PSx-3__ with EtherNet/IP

halstrup-walcher GmbH

Stegener Straße 10
D-79199 Kirchzarten

Phone: +49 (0) 76 61/39 63-0
Fax: +49 (0) 76 61/39 63-99

E-Mail: info@halstrup-walcher.de
Internet: www.halstrup-walcher.de

Table of Contents

1	Safety precautions	4
1.1	Appropriate use	4
1.2	Symbols	4
2	Data Structure DRIVE_DATA	5
3	Error Description (Error ID)	6
4	Description of the function blocks	8
4.1	Commonalities of all function blocks	8
4.2	MC_Move	11
4.3	MC_Error	14
4.4	MC_ReadParameter	16
4.5	MC_WriteParameter	18
4.6	MC_Parametrization	20
4.7	MC_PositionParametrization	23

Purpose of instruction manual

This instruction manual describes the function blocks for the PSx-3__-EI (with EtherNet/IP interface).

Improper use of these devices or failure to follow these instructions may cause injury or equipment damage. Every person who uses the devices must therefore read the manual and understand the possible risks. The instruction manual, and in particular the safety precautions contained therein, must be followed carefully. **Contact the manufacturer if you do not understand any part of this instruction manual.**

The manufacturer reserves the right to continue developing these function blocks without documenting such development in each individual case. The manufacturer will be happy to determine whether this manual is up-to-date.

© 2015

The manufacturer owns the copyright to this instruction manual. It must not be copied either wholly or in part or made available to third parties.

1 Safety precautions

1.1 Appropriate use

The positioning systems PSx-3__-EI are especially suitable for automatically setting tools, stops or spindles for wood-processing equipment, packing lines, printing equipment, filling units and other types of special machines.

PSx3__-EI positioning systems are not stand-alone devices and may only be used if coupled to another machine.

1.2 Symbols

The symbols given below are used throughout this manual to indicate instances when improper operation could result in the following hazards:



WARNING!

This warns you of a potential hazard that could lead to bodily injury up to and including death if the corresponding instructions are not followed.



CAUTION!

This warns you of a potential hazard that could lead to significant property damage if corresponding instructions are not followed.



INFORMATION!

This indicates that the corresponding information is important for operating the function blocks properly.

2 Data Structure DRIVE_DATA

For each drive there's a data structure, in which some data of the drive is deposited. For each drive a global instance of this structure is required. This instance must be provided to each FB that operates on the corresponding drive. Hereby it shall be prevented for example, that two FBs accesses the parameter interface of a single drive.

Parameter name	Data type	Written by	Description
Name	STRING[16]	User (optional)	Name of axis
Description	STRING[32]	User (optional)	Description (e.g. function, task of this axis)
State	DINT	Function blocks	Actual state

Example:

In the project there are three drives. Each drive shall be moved with a FB MC_Move, additionally it should be possible to determine the state of each drive with MC_Error and for each drive it should be possible to perform any user-defined writing and reading accesses.

To achieve this, altogether three variables of the type DRIVE_DATA are necessary, these have to be generated in the Controller Tags section:

- Drive_1
- Drive_2
- Drive_3

For the execution of the required functions for each drive also an instance of the FB MC_Move, MC_Error, MC_ReadParameter and MC_WriteParameter is required:

- MC_Move_1, MC_Error_1, MC_Read_1, MC_Write_1:
VAR_IN_OUT "Drive" → enter "Drive_1"
- MC_Move_2, MC_Error_2, MC_Read_2, MC_Write_2:
VAR_IN_OUT "Drive" → enter "Drive_2"
- MC_Move_3, MC_Error_3, MC_Read_3, MC_Write_3:
VAR_IN_OUT "Drive" → enter "Drive_3"

3 Error Description (Error ID)

Subsequently the error codes are shown, which are displayed by the function blocks:

ErrorID (hex)	Beschreibung
16xF000 (mask)	FB
16#1xxx	Error in MC_Move
16#2xxx	Error in MC_Error
16#3xxx	Error in MC_ReadParameter
16#4xxx	Error in MC_WriteParameter
16#5xxx	Error in MC_Parametrization
16#6xxx	Error in MC_PositionParametrization
16#0F00 (mask)	Internal FB and PD errors
16#x1xx	Error in state machine or other FB internal error
16#x6xx	Unallowed input data change
16#x7xx	Connection Faulted
16#00F0 (mask)	Parameter errors
16#xx1x	Parameter: communication timeout (1000 ms)
16#xx2x	Parameter: invalid parameter number
16#xx3x	Parameter: value is read only
16#xx4x	Parameter: lower or upper limit exceeded
16#xx5x	Parameter: faulty sub-index
16#xx6x	Parameter: not an array
16#xx7x	Parameter: incorrect data type
16#xx8x	Parameter: setting not allowed (resetting only)
16#xx9x	Parameter: request cannot be processed due to operating state
16#xxAx	Other error
16#000F (mask)	Drive errors
16#xxx1	Drag error
16#xxx2	Under- or overvoltage motor supply
16#xxx3	Positioning run aborted
16#xxx4	Temperature exceeded
16#xxx5	Absolute measuring system error
16#xxx6	Block or overcurrent error
16#xxx7	Manual displacement
16#xxx8	Incorrect target value
16#xxx9	Under- or overvoltage during run
16#xxxA	Lower position limit exceeded
16#xxxB	Upper position limit exceeded

The errors "Drive Errors" are a copy of the error bits in the status word of the PSx.

Examples:

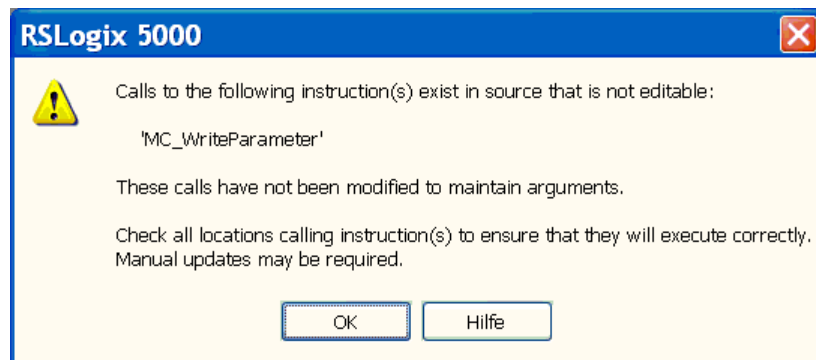
- Run command (MC_Move) with incorrect target value → ErrorID = 16#1008

- Writing a parameter (MC_WriteParameter) with invalid parameter number → ErrorID = 16#4020

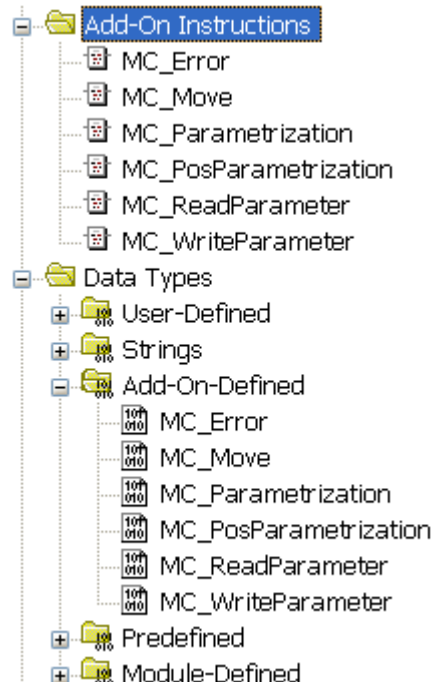
4 Description of the function blocks

Initially the function blocks have to be included in an own RSLogix project. This happens in the Controller Organizer with right click on "Add-On Instructions", then "Import Add-On Instruction" and importing the desired function blocks individually.

Eventually the following message appears which can ignored:



As result the function blocks present itself in the following way in the Controller Organizer:



4.1 Commonalities of all function blocks

The FBs respectively use a part of the following variables, all of them of the type VAR_IN_OUT. These inputs and outputs must be connected in either case, otherwise there will be error messages when downloading the project.

Drive

Reference to the desired drive (see also chapter 2)

- Type: DRIVE_DATA
- Nature: VAR_IN_OUT

ErrorDescription

Error description

- Type: STRING
- Default value: ""
- connect with any string variable in the Controller Tags section

INPUT_ConnectionFaulted

Signals if the input data of the corresp. drive are valid

- Type: BOOL
- connect with I.ConnectionFaulted of the corresp. drive

INPUT_Status_Word

Status word of the corresp. drive

- Type: INT
- connect with I.status_word of the corresp. drive

INPUT_Actual_Position

Actual position of the corresp. drive

- Type: DINT
- connect with I.actual_position of the corresp. drive

INPUT_Parameter_ID

Parameter number of the parameter interface of the corresp. drive

- Type: INT
- connect with I.Parameter_ID of the corresp. drive

INPUT_Parameter_value

Parameter value of the parameter interface of the corresp. drive

- Type: DINT
- connect with I.Parameter_value of the corresp. drive

INPUT_Subindex

Array subindex of the parameter interface of the corresp. drive

- Type: INT
- connect with I.Subindex of the corresp. drive

OUTPUT_Control_Word

Control word of the corresp. drive

- Type: INT
- connect with O.control_word of the corresp. drive

OUTPUT_Target_Position

Target position of the corresp. drive

- Type: DINT
- connect with O.target_position of the corresp. drive

OUTPUT_Parameter_ID

Parameter number of the parameter interface of the corresp. drive

- Type: INT
- connect with O.Parameter_ID of the corresp. drive

OUTPUT_Parameter_value

Parameter value of the parameter interface of the corresp. drive

- Type: DINT
- connect with O.Parameter_value of the corresp. drive

OUTPUT_Subindex

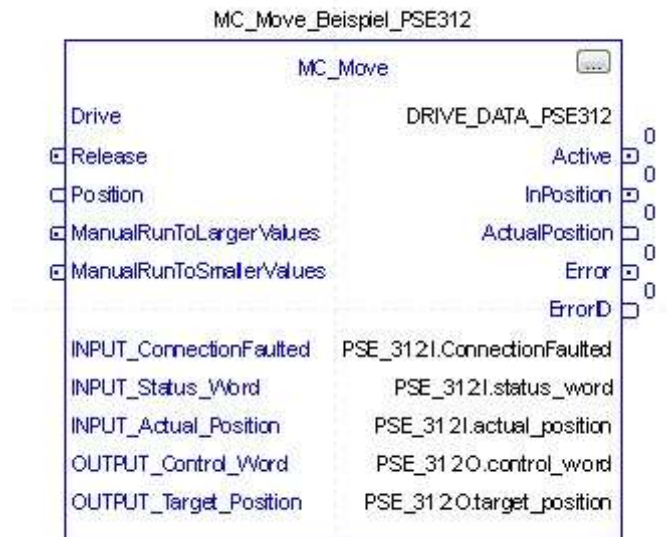
Array subindex of the parameter interface of the corresp. drive

- Type: INT
- connect with O.Subindex of the corresp. drive

In the following descriptions of the particular FBs these variables of the type VAR_IN_OUT are not listed separately.

4.2 MC_Move

This FB serves to send run commands to the drive.



Release

Release of the drive

- Type: BOOL
- Initial value: FALSE
- Nature: VAR_INPUT

Description:

- Run commands will only be executed if this bit is set.
- This input directly controls the release bit (bit 4) in the control word. If this input stays activated and e.g. the readjustment in the drive is activated, the drive readjusts automatically.
- If the input is activated and the target position is changed, the drive immediately moves to that position. An edge is not necessary.
- If the input is deasserted during the run, the drive stops.

Position

Target position to be approached

- Type: DINT
- Initial value: 0
- Nature: VAR_INPUT

Description:

- If during a run a new target position is sent, this target position is approached immediately.
- If the release bit is still set after the end of a run and the target position is changed, the drive immediately approaches to that position.



INFORMATION!

In order to move to the same target position e.g. after a blocking condition, the release has to be deasserted and asserted again.

ManualRunToLargerValues

Manual run to larger values

- Type: BOOL
- Initial value: FALSE
- Nature: VAR_INPUT

Description:

- Manual run to larger values, finishing at the positive range limit.
- Additionally the input "Release" has to be on resp. set.



CAUTION!

When deasserting the input "ManualRunToLargerValues", additionally the release input has to be deasserted. Otherwise the drive will move to the target position (FB input "Position").

ManualRunToSmallerValues

Manual run to smaller values

- Type: BOOL
- Initial value: FALSE
- Nature: VAR_INPUT

Description:

- Manual run to larger values, finishing at the negative range limit.
- Additionally the input "Release" has to be on resp. set.



CAUTION!

When deasserting the input "ManualRunToSmallerValues", additionally the release input has to be deasserted. Otherwise the drive will move to the target position (FB input "Position").

Active

Run command or run is active

- Type: BOOL
- Nature: VAR_OUTPUT

This output is asserted, if:

- the release bit is set from 0 to 1 (rising edge)
- the release is already present and the target position is changing
- the bit "drive is running" in the status word of the drive is set (e.g. when the drive is readjusting its position)

This output is deasserted, if:

- at the end of a run the bit "drive is running" in the status word of the drive is no longer set
- a communication error occurs

InPosition

Target position reached

- Type: BOOL
- Nature: VAR_OUTPUT

This output is a copy of the status bit "target position reached". If a communication error occurs, it will be deasserted.

Actual position

Actual value of the position

- Type: DINT
- Nature: VAR_OUTPUT

This value is a copy of the actual position. If a communication error occurs, the value will be set to 0.

Error

Error while executing the FB or error in drive

- Type: BOOL
- Nature: VAR_OUTPUT

The error bit also might be set during a move of the drive (e.g. drag error).

ErrorID

Error code

- Type: INT
- Nature: VAR_OUTPUT

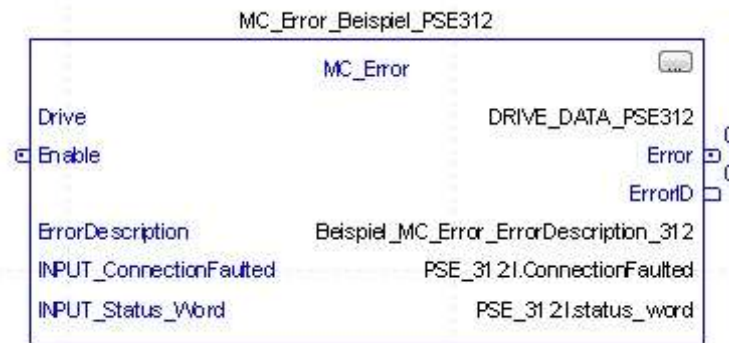
The error bit also might be set during a move of the drive (e.g. drag error). In case of no error, the value is set to 0.

If the drive reports multiple errors, the ErrorID with the highest priority is shown. This priority corresponds to the order in the following table (highest priority has 16#x1xx):

ErrorID	Description
16#x1xx	FB internal error
16#x2xx	Invalid PD input address
16#x3xx	Invalid PD output address
16#x4xx	Error while reading PD
16#x5xx	Error while writing PD
16#xxx2	Under- or overvoltage motor supply
16#xxx4	Temperature exceeded
16#xxx5	Absolute measuring system error
16#xxx8	Incorrect target value
16#xxx9	Under- or overvoltage during run
16#xxx6	Block or overcurrent error
16#xxx7	Manual displacement
16#xxxA	Lower position limit exceeded
16#xxxB	Upper position limit exceeded
16#xxx3	Positioning run aborted
16#xxx1	Drag error

4.3 MC_Error

This FB reports the state of the drive and the FB as error bit, error code ("ErrorID") and as text.



Enable

The outputs Error, ErrorID and ErrorDescription permanently are updated by the drive, as long as Enable is set. If Enable is deasserted, these outputs switch to their default values.

- Type: BOOL
- Initial value: FALSE
- Nature: VAR_INPUT

Error

Error while executing the FB or error in drive

- Type: BOOL
- Default value: FALSE
- Nature: VAR_OUTPUT

ErrorID

Error code (see following table "ErrorID")

- Type: INT
- Default value: 0
- Nature: VAR_OUTPUT

ErrorDescription

Error Description as text

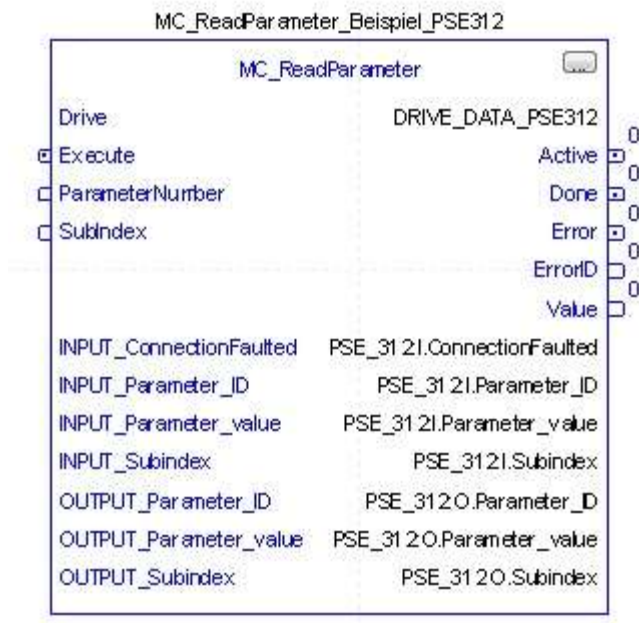
- Type: STRING
- Default value: ""
- Nature: VAR_IN_OUT

The priority corresponds to the order in the following table (highest priority has 16#x1xx).

ErrorID	ErrorDescription
16#x1xx	FB internal error
16#x2xx	Invalid PD input address
16#x4xx	Error while reading PD
16#xxx2	Under- or overvoltage motor supply
16#xxx4	Temperature exceeded
16#xxx5	Absolute measuring system error
16#xxx8	Incorrect target value
16#xxx9	Under- or overvoltage during run
16#xxx6	Block or overcurrent error
16#xxx7	Manual displacement
16#xxxA	Lower position limit exceeded
16#xxxB	Upper position limit exceeded
16#xxx3	Positioning run aborted
16#xxx1	Drag error

4.4 MC_ReadParameter

With this FB values of parameters can be read by the drive. Each parameter can be read except par. 23 ("device model as string").



Execute

Start of a reading process

- Type: BOOL
- Initial value: FALSE
- Nature: VAR_INPUT

Description:

When issuing a rising edge, a reading process of the parameter which is specified by "ParameterNumber" and "Subindex" is started. For a new reading process, a new rising edge has to be generated. When deasserting the bit, the outputs fall back to their specified default value.

ParameterNumber

Parameter number of the parameter to be read

- Type: INT
- Initial value: 0
- Nature: VAR_INPUT

SubIndex

Array subindex of the parameter

- Type: INT
- Initial value: 0
- Nature: VAR_INPUT

Active

Bit is set as long as the reading process is active

- Type: BOOL

- Default value: FALSE
- Nature: VAR_OUTPUT

The bit is being deasserted as soon as the value has been read or an error occurred.

Done

Bit is set as soon as the parameter has been read successfully and is available in "Value"

- Type: BOOL
- Default value: FALSE
- Nature: VAR_OUTPUT

The bit is being deasserted when starting a reading process.

Error

Bit is set if an error occurred during the execution of the FB

- Type: BOOL
- Default value: FALSE
- Nature: VAR_OUTPUT

ErrorID

Error code (see table "ErrorID" in chapter 3)

- Type: INT
- Default value: 0
- Nature: VAR_OUTPUT

Drive errors are not considered when reading a parameter.

Value

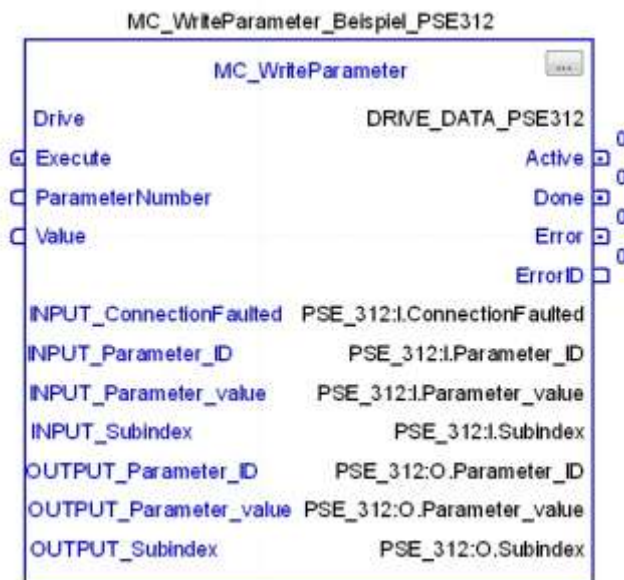
Actual value of the parameter which has been read

- Type: DINT
- Default value: 0
- Nature: VAR_OUTPUT

When an error occurred, the value will be 0.

4.5 MC_WriteParameter

With this FB values of parameters can be written into the drive.



Execute

Start of a writing process

- Type: BOOL
- Initial value: FALSE
- Nature: VAR_INPUT

Description:

When issuing a rising edge, a writing process of the parameter which is specified by "ParameterNumber" and "Subindex" with the value which is specified by the input "Value" is started. For a new writing process, a new rising edge has to be generated. When deasserting the bit, the outputs fall back to their specified default value.

ParameterNumber

Parameter number of the parameter to be written

- Type: INT
- Initial value: 0
- Nature: VAR_INPUT

Value

Value to be written to the parameter

- Type: DINT
- Initial value: 0
- Nature: VAR_INPUT

Active

Bit is set as long as the writing process is active

- Type: BOOL
- Default value: FALSE
- Nature: VAR_OUTPUT

The bit is being deasserted as soon as the value has been written or an error occurred.

Done

Bit is set as soon as the parameter has been written successfully

- Type: BOOL
- Default value: FALSE
- Nature: VAR_OUTPUT

The bit is being deasserted when starting a writing process.

Error

Bit is set if an error occurred during the execution of the FB

- Type: BOOL
- Default value: FALSE
- Nature: VAR_OUTPUT

ErrorID

Error code (see table "ErrorID" in chapter 3)

- Type: INT
- Default value: 0
- Nature: VAR_OUTPUT

Drive errors are not considered when writing a parameter.

4.6 MC_Parametrization

With this FB all parameters of the drive can be written.



The following items have to be considered when using this FB:

- For each parameter value there's additionally an enable tag in order to determine whether the parameter shall be written or not.
Example: DirRotation_Enable = 1 → DirRotation_Value is written
- The order of the write accesses is like represented in the FB diagram ("DeliveryState" → "DirRotation" → ...).
- Optionally a delivery state might be commanded before setting a certain number of parameters. To do this, the input "DeliveryState_113" has to be set to TRUE before the execution of the FB. Thus the values of each parameter are set to the delivery state (initially without saving).
- Optionally at the end additionally the written values might be saved permanently. To do this, the input "SaveSettings_113" has to be set to TRUE before the execution of the FB.
- In case of an error while writing a parameter, the subsequent parameters are not written any more. Also no saving of the values is carried out, if the input "SaveSettings" is set.

Execute

Start of a parametrization process

- Type: BOOL
- Initial value: FALSE
- Nature: VAR_INPUT

Description:

When issuing a rising edge, a parametrization process with the given values is started. For a new parametrization process, a new rising edge has to be generated. When deasserting the bit, the outputs fall back to their specified default value.

DeliveryState

Loading of the delivery state (initially without saving)

- Type: BOOL
- Initial value: FALSE
- Nature: VAR_INPUT

However, IP address and address assigning method stay unaffected.

x_Enable

If set, the corresp. parameter is written

- Type: BOOL
- Initial value: FALSE
- Nature: VAR_INPUT

x_Value

Desired value of the parameter

- Initial value: 0
- Nature: VAR_INPUT

The parameter number is given after the parameter name. The data type, a description as well as the value range can be extracted of the instruction manual of the PSx-3__-EI.

SaveSettings

Saving the settings permanently

- Type: BOOL
- Initial value: FALSE
- Nature: VAR_INPUT

Active

Bit is set as long as the parametrization process is active

- Type: BOOL
- Default value: FALSE
- Nature: VAR_OUTPUT

The bit is being deasserted as soon as the parametrization has been finished successfully or an error occurred.

Done

Bit is set as soon as the parametrization has been finished successfully

- Type: BOOL
- Default value: FALSE
- Nature: VAR_OUTPUT

The bit is being deasserted when starting a parametrization process.

Error

Bit is set if an error occurred during the execution of the FB

- Type: BOOL
- Default value: FALSE
- Nature: VAR_OUTPUT

ErrorID

Error code (see table "ErrorID" in chapter 3)

- Type: INT
- Default value: 0
- Nature: VAR_OUTPUT

Drive errors are not considered during a parametrization.

ErrorParameter

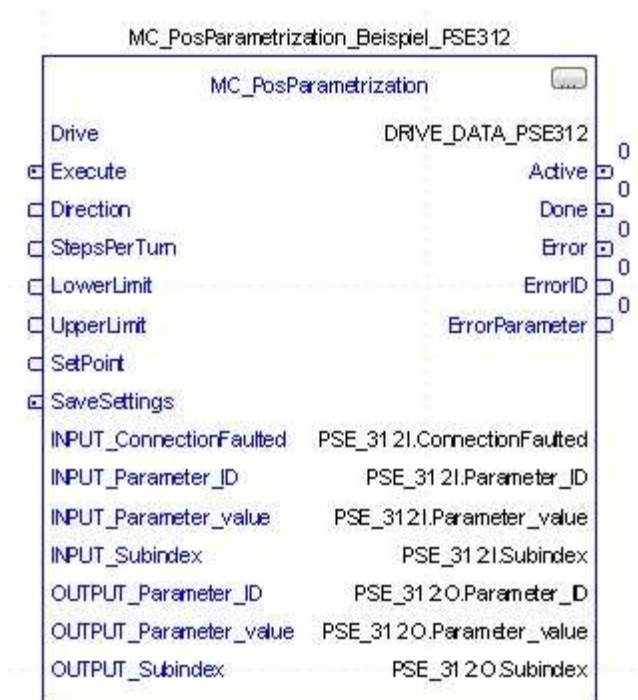
Parameter number that caused the error (in case of an error)

- Type: INT
- Default value: 0
- Nature: VAR_OUTPUT

If no error occurred, this value is 0.

4.7 MC_PositionParametrization

With this FB the parametrization of the position data can be carried out (parameters having an influence on the value of the displayed actual position).



The following items have to be considered when using this FB:

- Each value has to be written and the values have to have a reasonable relation to each other. Each value is processed, after that the following parameters are written in the order stated below:
 - Direction of rotation (Par. 26) = Direction
 - Position scaling, numerator (Par. 28) = 400
 - Position scaling, denominator (Par. 30) = StepsPerTurn
 - Actual value (Par. 10) = SetPoint
 - If (SetPoint > UpperLimit):
 - Upper mapping end (Par. 34) = SetPoint + (3 x StepsPerTurn)
 - sonst:
 - Upper mapping end (Par. 34) = UpperLimit + (3 x StepsPerTurn)
 - Upper limit (Par. 36) = UpperLimit
 - Lower limit (Par. 38) = LowerLimit
- The number of steps per revolution "StepsPerTurn" directly results in the value of the parameter "Position scaling, denominator" (Par. 30). Thereby it is assumed that the value of "Position scaling, numerator" (Par. 28) is in delivery state, thus 400.
- Before writing the parameters, the entered values are checked for validity.

Subsequently the conditions and error codes which are displayed if a condition is not satisfied.

Condition	ErrorID	ErrorParameter
$\text{StepsPerTurn} \geq 1$	16#6140	39
$\text{StepsPerTurn} \leq 10000$	16#6140	39
$\text{LowerLimit} \leq \text{UpperLimit}$	16#6140	42
$(\text{UpperLimit} - \text{LowerLimit}) / \text{StepsPerTurn} \leq 250$	16#6140	43
If $\text{SetPoint} < \text{LowerLimit}$: $(\text{UpperLimit} - \text{SetPoint}) / \text{StepsPerTurn} \leq 250$	16#6140	3
If $\text{SetPoint} > \text{UpperLimit}$: $(\text{SetPoint} - \text{LowerLimit}) / \text{StepsPerTurn} \leq 250$	16#6140	3

- Optionally at the end additionally the written values might be saved permanently. To do this, the input "SaveSettings" has to be set to TRUE before the execution of the FB.
- In case of an error while writing a parameter, the subsequent parameters are not written any more. Also no saving of the values is carried out, if the input "SaveSettings" is set.

Execute

Start of a parametrization process

- Type: BOOL
- Initial value: FALSE
- Nature: VAR_INPUT

Description:

When issuing a rising edge, a parametrization process with the given values is started. For a new parametrization process, a new rising edge has to be generated. When deasserting the bit, the outputs fall back to their specified default value.

Direction

Direction in which the drive shall turn with larger values (if looking at the output shaft):

0 → CW, 1 → CCW

- Type: INT
- Initial value: 0
- Nature: VAR_INPUT

StepsPerTurn

Number of steps per revolution at the output shaft (resolution)

- Type: INT
- Initial value: 0
- Nature: VAR_INPUT

LowerLimit

Lower limit

- Type: DINT
- Initial value: 0
- Nature: VAR_INPUT

UpperLimit

Upper limit

- Type: DINT
- Initial value: 0
- Nature: VAR_INPUT

SetPoint

Value on which the measuring system is referenced (new actual value at the actual position)

- Type: DINT
- Initial value: 0
- Nature: VAR_INPUT

SaveSettings

Saving the settings permanently

- Type: BOOL
- Initial value: FALSE
- Nature: VAR_INPUT

Active

Bit is set as long as the parametrization process is active

- Type: BOOL
- Default value: FALSE
- Nature: VAR_OUTPUT

The bit is being deasserted as soon as the parametrization has been finished successfully or an error occurred.

Done

Bit is set as soon as the parametrization has been finished successfully

- Type: BOOL
- Default value: FALSE
- Nature: VAR_OUTPUT

The bit is being deasserted when starting a parametrization process.

Error

Bit is set if an error occurred during the execution of the FB

- Type: BOOL
- Default value: FALSE
- Nature: VAR_OUTPUT

ErrorID

Error code (see table "ErrorID" in chapter 3)

- Type: INT
- Default value: 0
- Nature: VAR_OUTPUT

Drive errors are not considered during a parametrization.

ErrorParameter

Parameter number that caused the error (in case of an error)

- Type: INT
- Default value: 0
- Nature: VAR_OUTPUT

If no error occurred, this value is 0.