Function Blocks for PSx-3__ with PROFINET interface

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

Document 7100.005454B Version 1.1 4/2017

Table of Contents

Purpose of instruction manual

This instruction manual describes the function blocks for the PSx-3__-PN (with PROFINET 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.

1 Safety precautions

1.1 Appropriate use

The positioning systems PSx-3__-PN 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__-PN 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.
ℹ	**INFORMATION!** This indicates that the corresponding information is important for operating the function blocks properly.

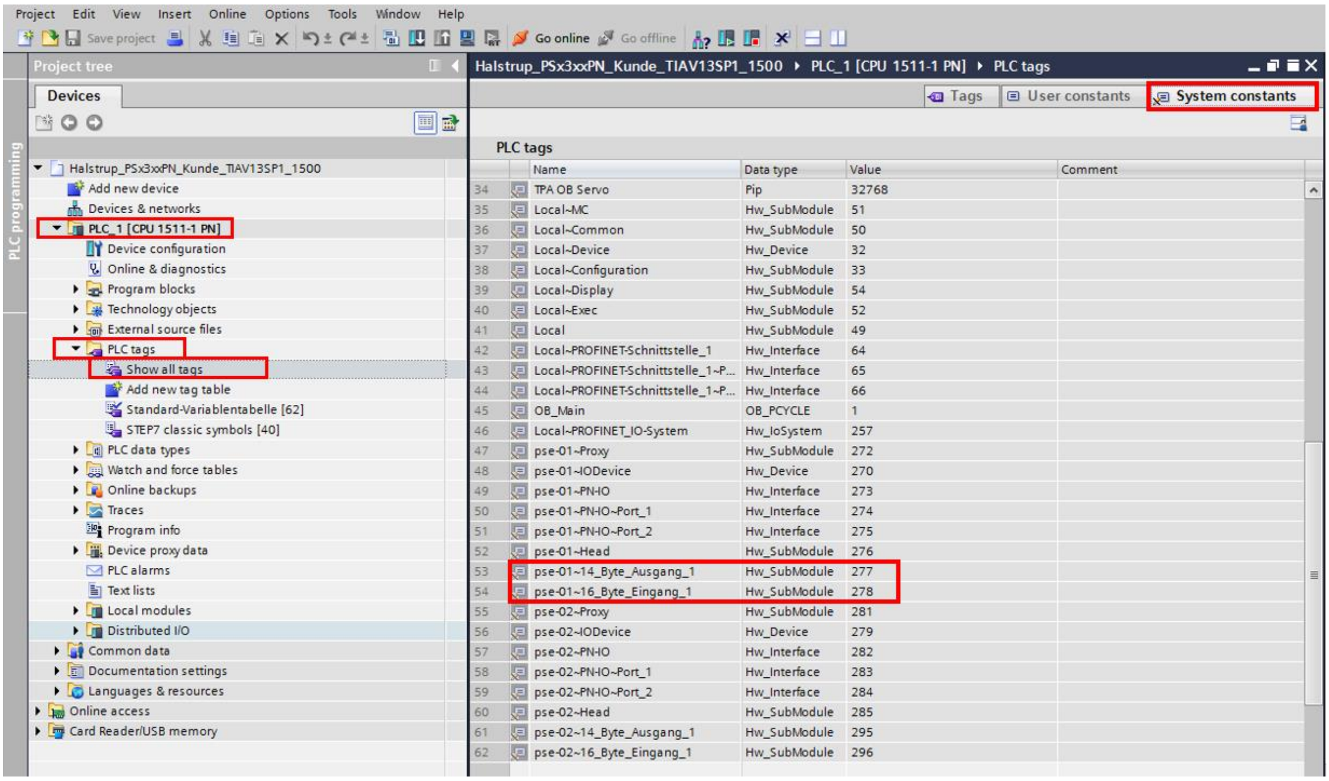
2 Data Structure DRIVE_DATA

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. Furthermore, in this data structure the addresses for the input and output data of the corresponding drive has to be deposited.

Parameter name	Data type	Written by	Description
PdAddressIn	INT	User	Address process data Slave → Master
PdAddressOut	INT	User	Address process data Master → Slave
Name	STRING[16]	User (optional)	Name der Achse
Beschreibung	STRING[32]	User (optional)	Description (e.g. function, task of this axis)
State	DINT	Function blocks	Actual state
Private	STRUCT	Function blocks	Data structure for internal use
Only in the versions for the TIA Portal:			
DB_LIB_IN/OUT	DT_LIB_IN/OUT	Function blocks	Data structure for internal use

In the versions for the TIA Portal the data structure DRIVE_DATA additionally includes a parameter of the structure “DT_LIB_IN/OUT”, therefore this data structure has to be taken over of the library into the user project.

The following diagram shows how in the TIA Portal the given process data addresses may be checked:



Halstrup_PSx3xxPN_Kunde_TIAV13SP1_1500 ▸ PLC_1 [CPU 1511-1 PN] ▸ PLC tags

PLC tags

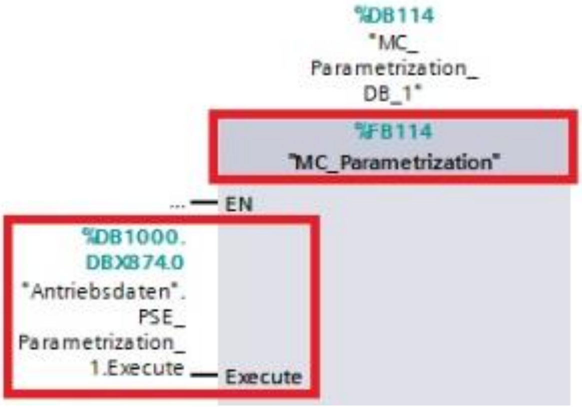
	Name	Data type	Value	Comment
34	TPA OB Servo	Pip	32768	
35	Local~MC	Hw_SubModule	51	
36	Local~Common	Hw_SubModule	50	
37	Local~Device	Hw_Device	32	
38	Local~Configuration	Hw_SubModule	33	
39	Local~Display	Hw_SubModule	54	
40	Local~Exec	Hw_SubModule	52	
41	Local	Hw_SubModule	49	
42	Local~PROFINET-Schnittstelle_1	Hw_Interface	64	
43	Local~PROFINET-Schnittstelle_1~P...	Hw_Interface	65	
44	Local~PROFINET-Schnittstelle_1~P...	Hw_Interface	66	
45	OB_Main	OB_PCYCLE	1	
46	Local~PROFINET_IO-System	Hw_IoSystem	257	
47	pse-01~Proxy	Hw_SubModule	272	
48	pse-01~IODevice	Hw_Device	270	
49	pse-01~PN-IO	Hw_Interface	273	
50	pse-01~PN-IO~Port_1	Hw_Interface	274	
51	pse-01~PN-IO~Port_2	Hw_Interface	275	
52	pse-01~Head	Hw_SubModule	276	
53	pse-01~14_Byte_Ausgang_1	Hw_SubModule	277	
54	pse-01~16_Byte_Eingang_1	Hw_SubModule	278	
55	pse-02~Proxy	Hw_SubModule	281	
56	pse-02~IODevice	Hw_Device	279	
57	pse-02~PN-IO	Hw_Interface	282	
58	pse-02~PN-IO~Port_1	Hw_Interface	283	
59	pse-02~PN-IO~Port_2	Hw_Interface	284	
60	pse-02~Head	Hw_SubModule	285	
61	pse-02~14_Byte_Ausgang_1	Hw_SubModule	295	
62	pse-02~16_Byte_Eingang_1	Hw_SubModule	296	

(Addresses of process data in Step 7 TIA)

3 Data Component “Antriebsdaten”

In the versions for the TIA Portal this data component serves as a template for connecting the inputs and outputs of the function blocks. The data component provides the variables which are necessary to connect each input and output of all available function blocks. Each variable is present two times, so two positioning systems may be controlled.

Example:



In case the data component is taken over into the project, it is advisable to delete the variables which are not used (in order that PLC memory is not reserved needlessly) and to adjust the data component to the number of drives which are actually present.

The data component uses the data types “Parameter” and “ParameterEnable” by default, so these types have to be taken over of the library into the project also.

4 Error Description (Error ID)

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

ErrorID (hex)	Description
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#x2xx	Invalid PD input address
16#x3xx	Invalid PD output address
16#x4xx	Error while reading PD
16#x5xx	Error while writing PD
16#x6xx	Unallowed input data change
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

5 Description and use of the function blocks

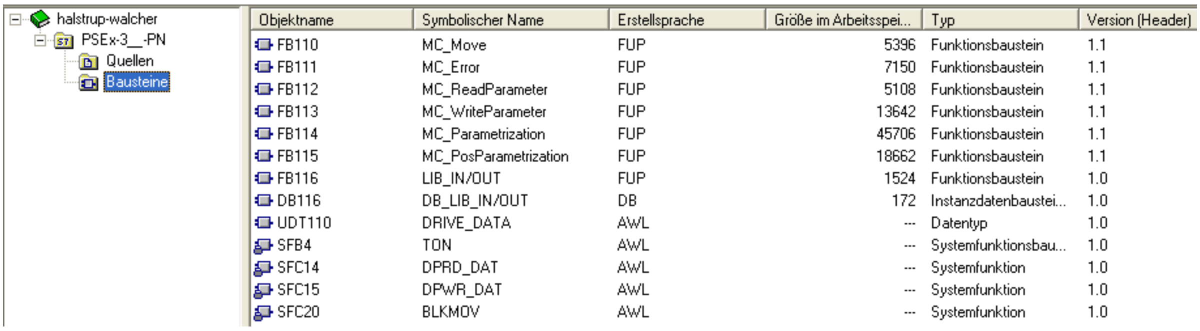
Initially the function blocks have to be included in an own user project. This happens by opening the desired library and copying the desired function blocks.

The following librarys are available:

- “halstrup-walcher”
  → for Step 7 Classic V5.5
- “library_PSx3xxPN_TIA13SP1_300”
  → for CPU S7-300 im TIA-Portal (V13 SP1)
- “library_PSx3xxPN_TIA14_300”
  → for CPU S7-300 im TIA-Portal (V14)
- “library_PSx3xxPN_TIA13SP1_1200-1500”
  → for CPU S7-1200/1500 im TIA-Portal (V13 SP1)
- “library_PSx3xxPN_TIA14_1200-1500”
  → for CPU S7-1200/1500 im TIA-Portal (V14)

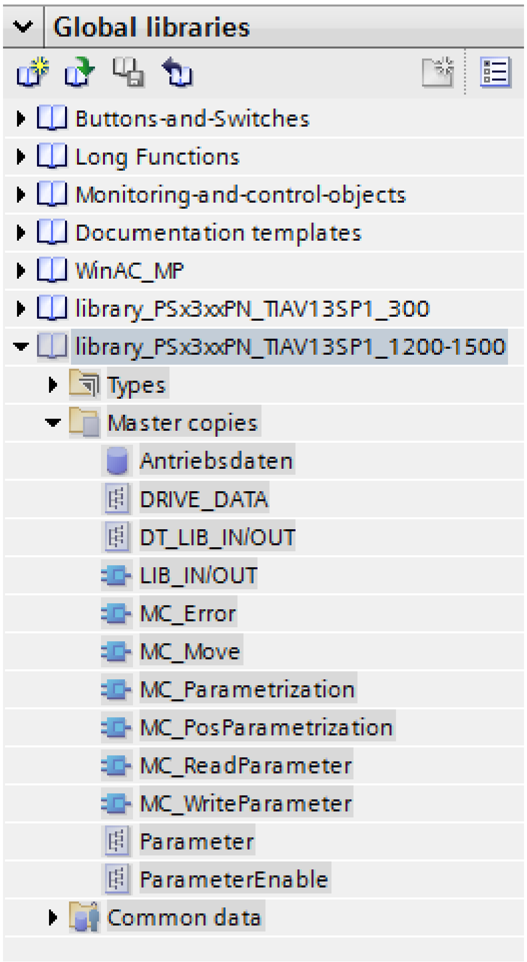
5.1 Opening the library

After opening the library the function blocks present itself in the following way:



Objektname	Symbolischer Name	Erstellsprache	Größe im Arbeitsspei...	Typ	Version (Header)
FB110	MC_Move	FUP	5396	Funktionsbaustein	1.1
FB111	MC_Error	FUP	7150	Funktionsbaustein	1.1
FB112	MC_ReadParameter	FUP	5108	Funktionsbaustein	1.1
FB113	MC_WriteParameter	FUP	13642	Funktionsbaustein	1.1
FB114	MC_Parametrization	FUP	45706	Funktionsbaustein	1.1
FB115	MC_PosParametrization	FUP	18662	Funktionsbaustein	1.1
FB116	LIB_IN/OUT	FUP	1524	Funktionsbaustein	1.0
DB116	DB_LIB_IN/OUT	DB	172	Instanzdatenbaustei...	1.0
UDT110	DRIVE_DATA	AWL	---	Datentyp	1.0
SFB4	TON	AWL	---	Systemfunktionsbau...	1.0
SFC14	DPRD_DAT	AWL	---	Systemfunktion	1.0
SFC15	DPWR_DAT	AWL	---	Systemfunktion	1.0
SFC20	BLKMOV	AWL	---	Systemfunktion	1.0

(View in Step 7 Classic V5.5)



(View in Step 7 TIA V13 + V14)

5.2 Copying of function blocks into the user project

The following elements of the library (independently of the actually used function blocks “MC_...” and the number of drives) in any case have to copied into the project:

When using the library version for Step 7 Classic V5.5:

Copy into “S7 Program → Blocks”:

- FB116 (“LIB IN/OUT”)
- DB116 (“DB_LIB_IN/OUT”)
- UDT110 (“DRIVE_DATA”)
- SFB4 (“TON”)
- SFC14 (“DPRD_DAT”)
- SFC15 (“DPWR_DAT”)
- SFC20 (“BLK_MOV”)

When using one of the library versions for Step 7 TIA V13 + V14:

Copy into “Program blocks” of the desired CPU:

- Data type DRIVE_DATA
- Data type DT_LIB_IN/OUT
- Function block LIB_IN/OUT

These elements serve for the communication to the drive. The functions and function blocks of this group must NOT be called in the program.

Besides the desired function blocks “MC_...” have to be copied into the user project, e.g. all or a choice of the following blocks:

- MC_Move
- MC_Error
- MC_ReadParameter
- MC_WriteParameter
- MC_Parametrization
- MC_ PosParametrization

In the versions for the TIA Portal additionally the data component “Antriebsdaten” may be taken over as a template for connecting the inputs and outputs of the function blocks. In ist standard implementation, the data types “Parameter” and “ParameterEnable” also have to be taken over (see chapter 3).

5.3 Generating instance data blocks

Per axis and per desired function block an instance data block has be generated.

5.4 Generating global data

Per axis one global variable of the type DRIVE_DATA has to be generated (size: 94 byte in Classic Step 7, 144 byte in TIA Portal). Additionally those variables have to be generated which are connected to the inputs and outputs of the particular blocks.

If applicable, it’s reasonable to generate an additional data block of the type “global” for this data (e.g. DB1000), in case of the TIA Portal also the template “Antriebsdaten” may be used (see chapter 3).

It’s not necessary to connect all inputs and outputs. If parts of a block are not used, the associated inputs may stay unconnected, then the respective initial value for this input is valid. Outputs not used also may stay open.

5.5 Commonalities of all function blocks

The function blocks are inserted in a part of the program that is called cyclically (e.g. in the OB1) and immediately be linked with their respective instance data blocks.

To the input “Drive” (type IN_OUT) the variable of the type “DRIVE_DATA” has to be connected which is provided for this axis.

The input EN and the output ENO of each function block may stay unconnected.

The inputs and outputs “Drive”, “EN” and “ENO” are not listed any more separately in the following descriptions of the particular function blocks.

5.6 Lockings between the function blocks

The function blocks partly are locked against each other. Thereby it’s ensured e.g. that not two accesses out of different function blocks can be executed simultaneously on the parameter channel of an axis.

The following rules apply:
- In case the input “Release” of MC_Move is set, the function blocks MC_Parametrization and MC_ PosParametrization cannot be activated (setting of “Execute” results in error 16#x100).
- In contrast it is possible to call MC_ReadParameter or MC_WriteParameter during a movement (e.g. in order to read out the actual value of the torque or to change the target speed during a run).
- In case the input “Execute” of MC_Parametrization or MC_ PosParametrization is set , the function block MC_Move cannot be activated (setting of “Release” results in error 16#1100).
- Always only one of the function blocks MC_ReadParameter, MC_WriteParameter, MC_Parametrization or MC_ PosParametrization can be active. If the input “Execute” of one of these function blocks is already set and the input “Execute” of a further function block is set, this one will report error 16#x100.
- The function block MC_Error can always be activated, independently of the other function blocks.

5.7 Example

The following example shall explane the procedure of involving the function blocks:

In the project three drives are located. Each drive shall be controlled by a function block MC_Move, additionally it shall be possible to determine the state of each drive with MC_Error and it shall be possible for each drive to execute any read and write access.

According chapter 5.2 for this purpose we copy the following blocks out of the library into the project of the user:

When using the library version for Step 7 Classic V5.5:

Copy into “S7 Program → Blocks”:

- FB116 (“LIB IN/OUT”)
- DB116 (“DB_LIB_IN/OUT”)
- UDT110 (“DRIVE_DATA”)
- SFB4 (“TON”)

- SFC14 (“DPRD_DAT”)
- SFC15 (“DPWR_DAT”)
- SFC20 (“BLK_MOV”)
- FB110 (“MC_Move”)
- FB111 (“MC_Error”)
- FB112 (“MC_ReadParameter”)
- FB113 (“MC_WriteParameter”)

When using one of the library versions for Step 7 TIA V13 + V14:

Copy into “Program blocks” of the desired CPU:

- Data type DRIVE_DATA
- Data type DT_LIB_IN/OUT
- Function block LIB_IN/OUT
- FB110 (“MC_Move”)
- FB111 (“MC_Error”)
- FB112 (“MC_ReadParameter”)
- FB113 (“MC_WriteParameter”)

In the next step according chapter 5.3 we generate the following instance data blocks:

- three instance data blocks of the type FB110 (“MC_Move”), e.g.
  - DB1101, symbolic name = “PSE_1_FB110”
  - DB1102, symbolic name = “PSE_2_FB110”
  - DB1103, symbolic name = “PSE_3_FB110”
- three instance data blocks of the type FB111 (“MC_Error”), e.g.
  - DB1111, symbolic name = “PSE_1_FB111”
  - DB1112, symbolic name = “PSE_2_FB111”
  - DB1113, symbolic name = “PSE_3_FB111”
- three instance data blocks of the type FB112 (“MC_ReadParameter”), e.g.
  - DB1121, symbolic name = “PSE_1_FB112”
  - DB1122, symbolic name = “PSE_2_FB112”
  - DB1123, symbolic name = “PSE_3_FB112”
- three instance data blocks of the type FB113 (“MC_WriteParameter”), e.g.
  - DB1131, symbolic name = “PSE_1_FB113”
  - DB1132, symbolic name = “PSE_2_FB113”
  - DB1133, symbolic name = “PSE_3_FB113”

After that according chapter 5.4 we generate another data block of the type “global” with the name DB1000. In that data block we generate three variables of the type “DRIVE_DATA”, e.g. with the following identifiers:

- “PSE_1”
- “PSE_2”
- “PSE_3”

Additionally there we generate those variables which control the particular blocks, e.g.:

- “PSE_1_Move_DINT_TargetPosition”,
- “PSE_2_Move_DINT_TargetPosition”,
- “PSE_1_Read_BOOL_Execute”,
- “PSE_2_Read_BOOL_Execute”,

- …

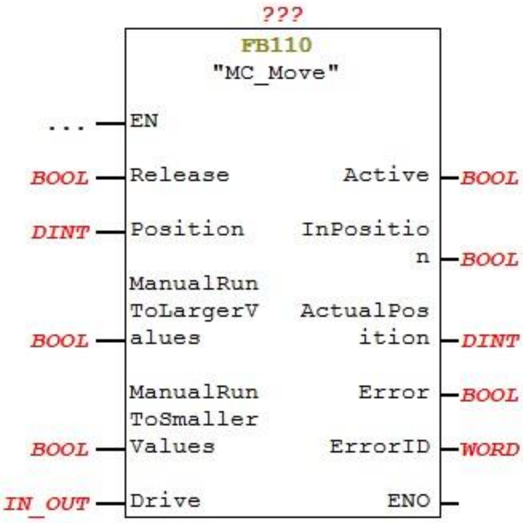
Now all preparations are done in order to use the function blocks. In the editor “Program blocks” the function blocks “MC_...” appear in the section “FB blocks”. So in our example the function blocks MC_Move, MC_Error, MC_ReadParameter and MC_WriteParameter are inserted each three times in a part of the program that is called cyclically (e.g. in the OB1) and immediately be linked with their respective instance data blocks.

To the input “Drive” we connect the variable of the type “DRIVE_DATA” which is provided for this axis (thus “PSE_1”, “PSE_2” and “PSE_3”).

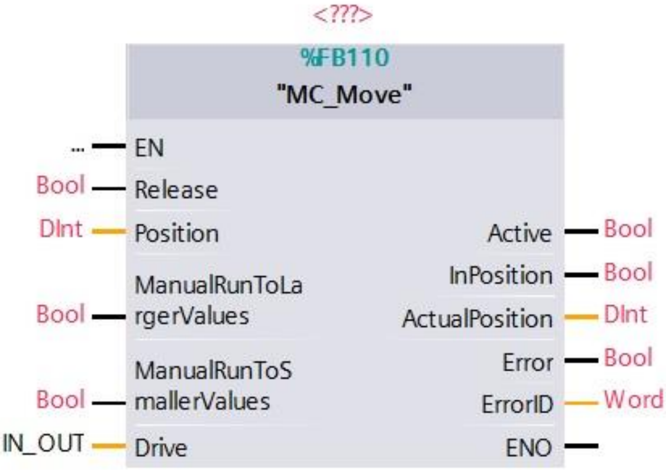
After that we connect the rest of the required inputs and outputs with those variables which are provided for this purpose (thus “PSE_1_Move_DINT_TargetPosition”, …).

5.8 MC_Move (FB110)

This FB serves to send run commands to the drive.



(View in Step 7 Classic V5.5)



(View in Step 7 TIA V13 + V14)

Release

Release of the drive

- Type: BOOL
- Initial value: FALSE
- Direction: 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
- Direction: 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
- Direction: 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
- Direction: 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
- Direction: 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
- Direction: 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
- Direction: 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
- Direction: OUTPUT

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

ErrorID

Error code

- Type: WORD
- Direction: 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.

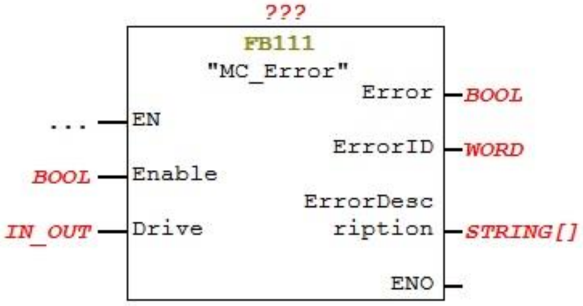
The outputs „Error“ and „ErrorID“ of MC_Move are always updated – also if the input „Release“ is not set.

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):

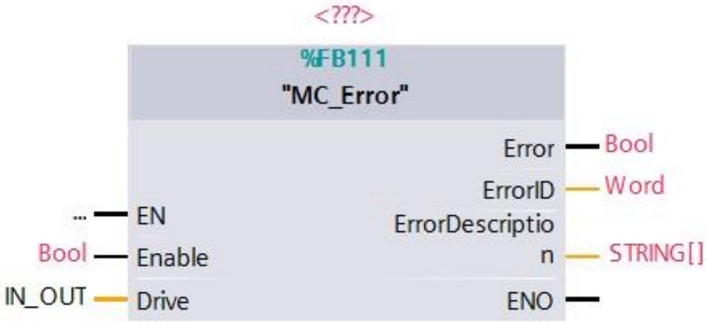
ErrorID	Description
16#x1xx	FB internal error
16#x2xx	Invalid PD input address
16#x3xx	Invalid PD output address
16#x4xx	Error while reading PD
16#x5xx	Error while writing PD
16#xxx2	Under- or overvoltage motor supply
16#xxx4	Temperature exceeded
16#xxx5	Absolute measuring system error
16#xxx8	Incorrect 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

5.9 MC_Error (FB111)

This FB reports the state of the drive and the FB as error bit, error code (“ErrorID”) and as text. In the case that MC_Error is activated, the error code is always the same as the error code of the function block MC_Move.



(View in Step 7 Classic V5.5)



(View in Step 7 TIA V13 + V14)

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
- Direction: INPUT

Error
Error while executing the FB or error in drive

- Type: BOOL
- Default value: FALSE
- Direction: OUTPUT

ErrorID
Error code (see following table “ErrorID”)

- Type: WORD
- Default value: 0
- Direction: OUTPUT

ErrorDescription

Error Description as text

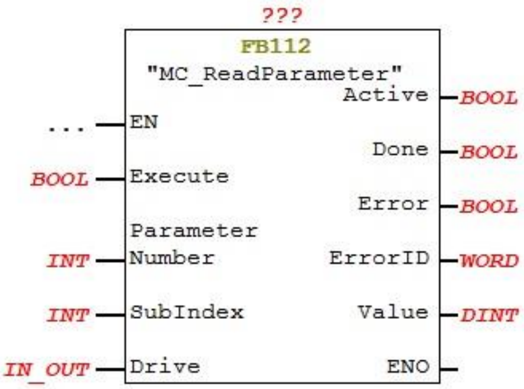
- Type: STRING
- Default value: “”
- Direction: OUTPUT

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

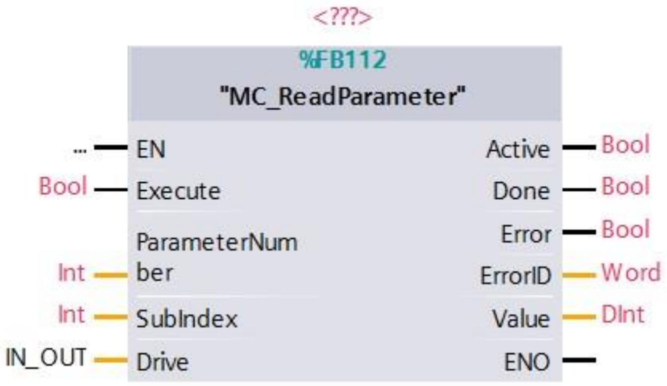
ErrorID	ErrorDescription
16#x1xx	FB internal error
16#x2xx	Invalid PD input address
16#x4xx	Error while reading PD
16#xxx2	Under- or overvoltage motor supply
16#xxx4	Temperature exceeded
16#xxx5	Absolute measuring system error
16#xxx8	Incorrect target value
16#xxx9	Under- or overvoltage during run
16#xxx6	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

5.10 MC_ReadParameter (FB112)

With this FB values of parameters can be read by the drive. Each parameter can be read except par. 13 (“device model as string”).



(View in Step 7 Classic V5.5)



(View in Step 7 TIA V13 + V14)

Execute
Start of a reading process
- Type: BOOL
- Initial value: FALSE
- Direction: 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
- Direction: INPUT

SubIndex
Array subindex of the parameter
- Type: INT
- Initial value: 0
- Direction: INPUT

Active
Bit is set as long as the reading process is active
- Type: BOOL
- Default value: FALSE
- Direction: 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
- Direction: 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
- Direction: OUTPUT

ErrorID

Error code (see table “ErrorID” in chapter 4)

- Type: WORD
- Default value: 0
- Direction: OUTPUT

Drive errors are not considered when reading a parameter.

Value

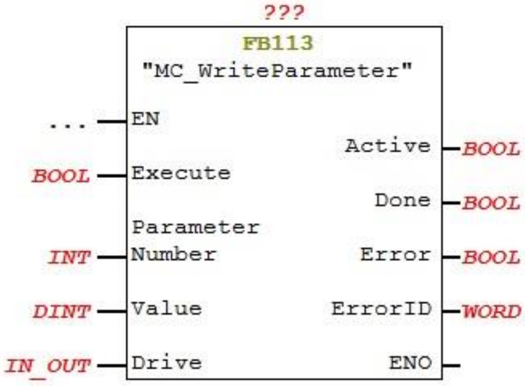
Actual value of the parameter which has been read

- Type: DINT
- Default value: 0
- Direction: OUTPUT

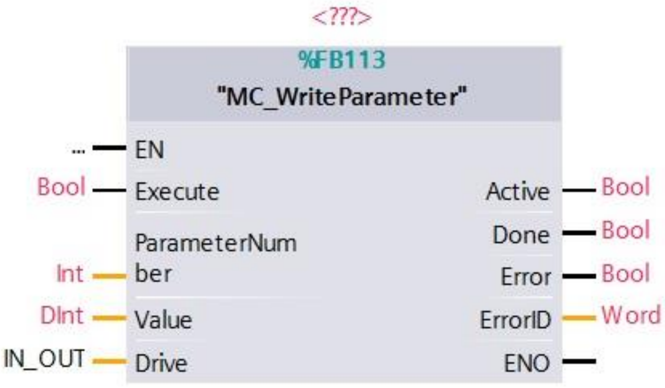
When an error occurred, the value will be 0.

5.11 MC_WriteParameter (FB113)

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



(View in Step 7 Classic V5.5)



(View in Step 7 TIA V13 + V14)

Execute

Start of a writing process

- Type: BOOL
- Initial value: FALSE
- Direction: INPUT

Description:

B 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
- Direction: INPUT

Value

Value to be written to the parameter

- Type: DINT
- Initial value: 0
- Direction: INPUT

Active

Bit is set as long as the writing process is active

- Type: BOOL
- Default value: FALSE
- Direction: 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
- Direction: 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
- Direction: OUTPUT

ErrorID

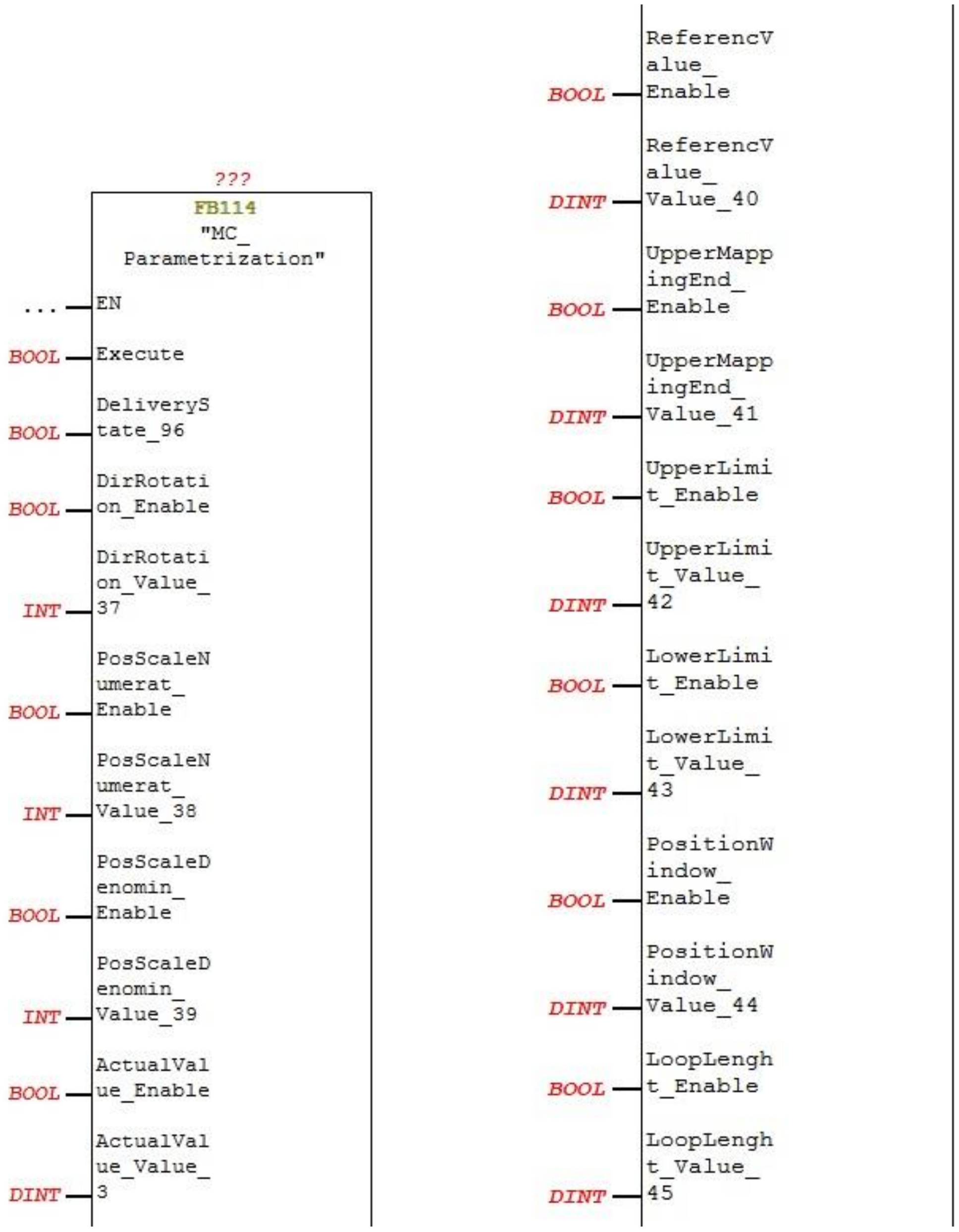
Error code (see table “ErrorID” in chapter 4)

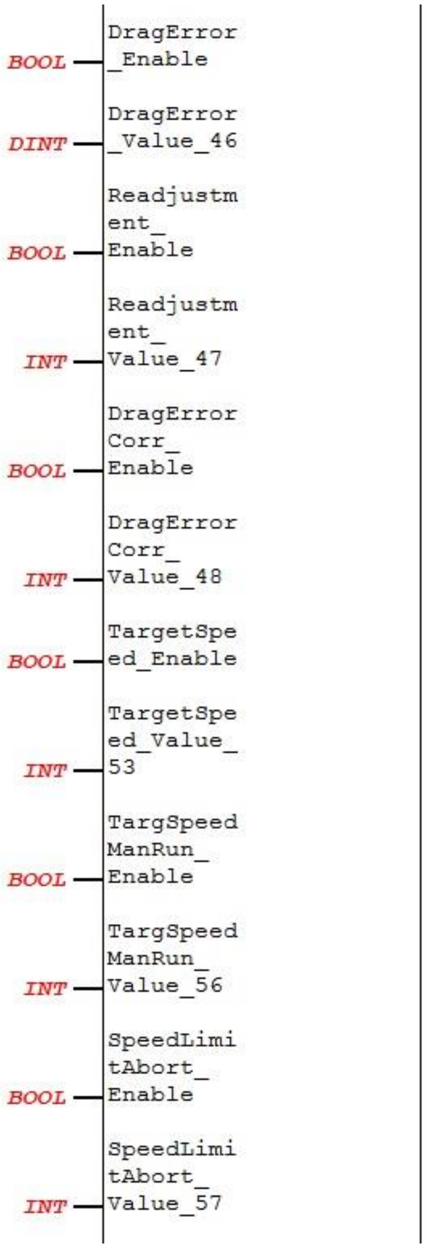
- Type: WORD
- Default value: 0
- Direction: OUTPUT

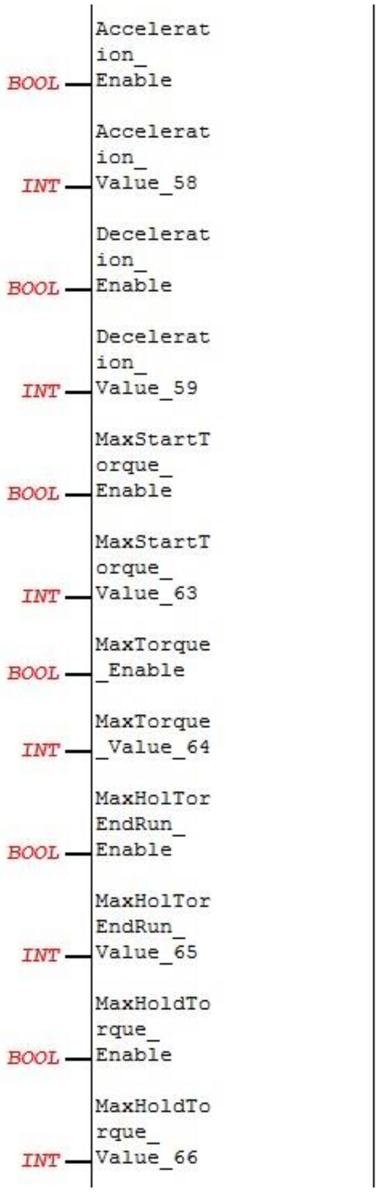
Drive errors are not considered when writing a parameter.

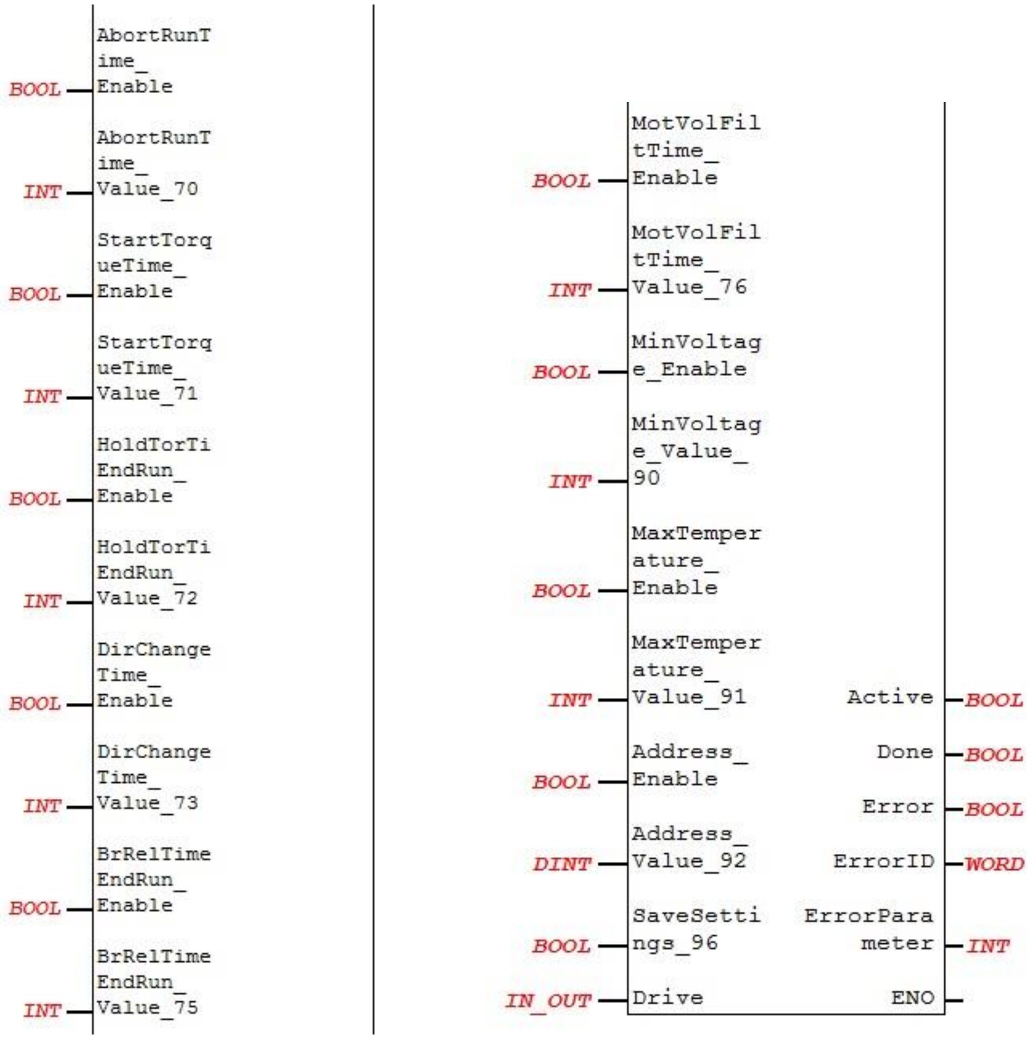
5.12 MC_Parametrization (FB114)

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

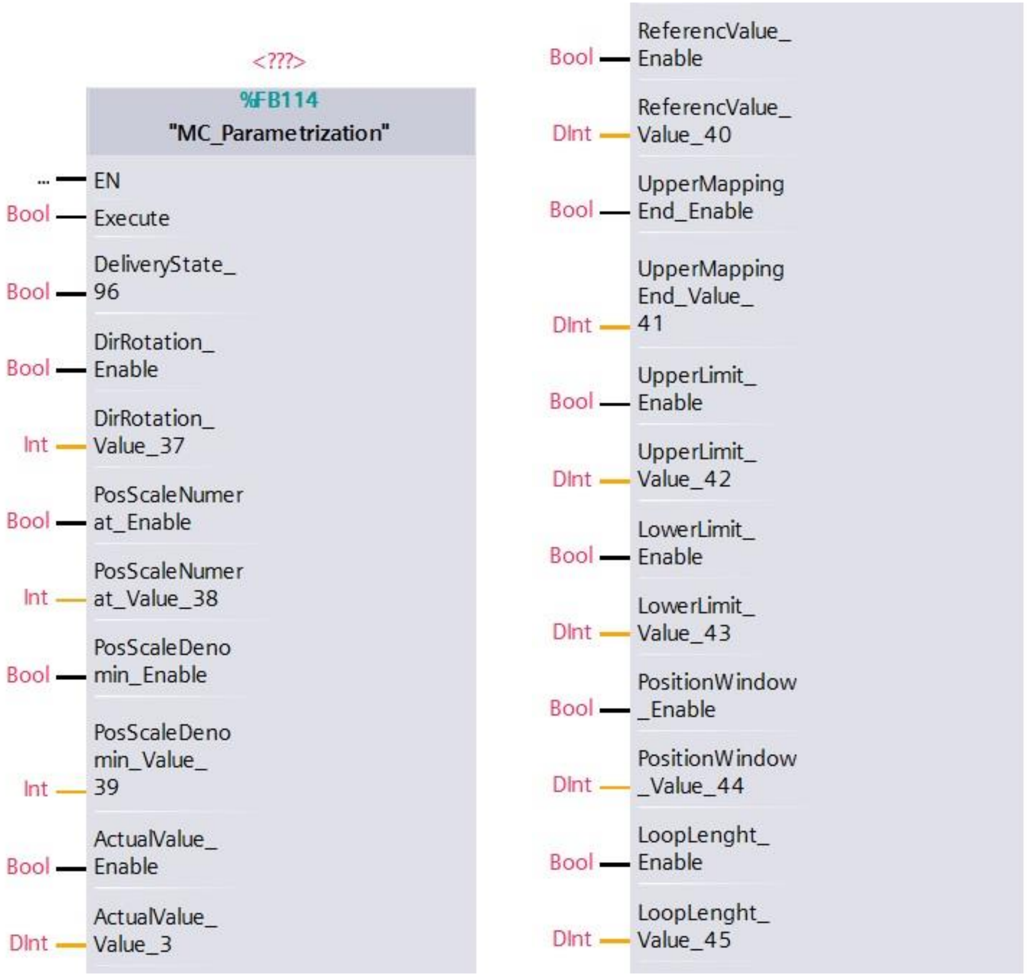


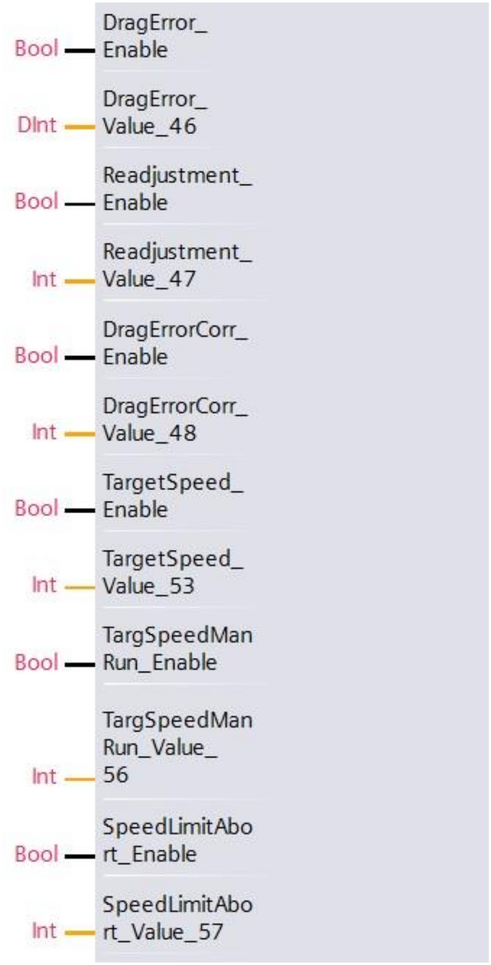

BOOL — DragError_Enable
DINT — DragError_Value_46
BOOL — Readjustment_Enable
INT — Readjustment_Value_47
BOOL — DragErrorCorr_Enable
INT — DragErrorCorr_Value_48
BOOL — TargetSpeed_Enable
INT — TargetSpeed_Value_53
BOOL — TargSpeedManRun_Enable
INT — TargSpeedManRun_Value_56
BOOL — SpeedLimitAbort_Enable
INT — SpeedLimitAbort_Value_57

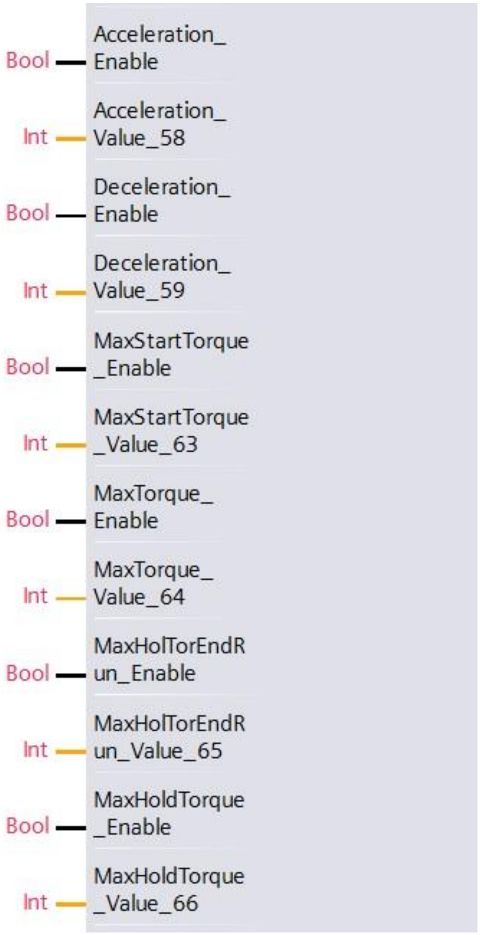

BOOL — Acceleration_Enable
INT — Acceleration_Value_58
BOOL — Deceleration_Enable
INT — Deceleration_Value_59
BOOL — MaxStartTorque_Enable
INT — MaxStartTorque_Value_63
BOOL — MaxTorque_Enable
INT — MaxTorque_Value_64
BOOL — MaxHolTorEndRun_Enable
INT — MaxHolTorEndRun_Value_65
BOOL — MaxHoldTorque_Enable
INT — MaxHoldTorque_Value_66

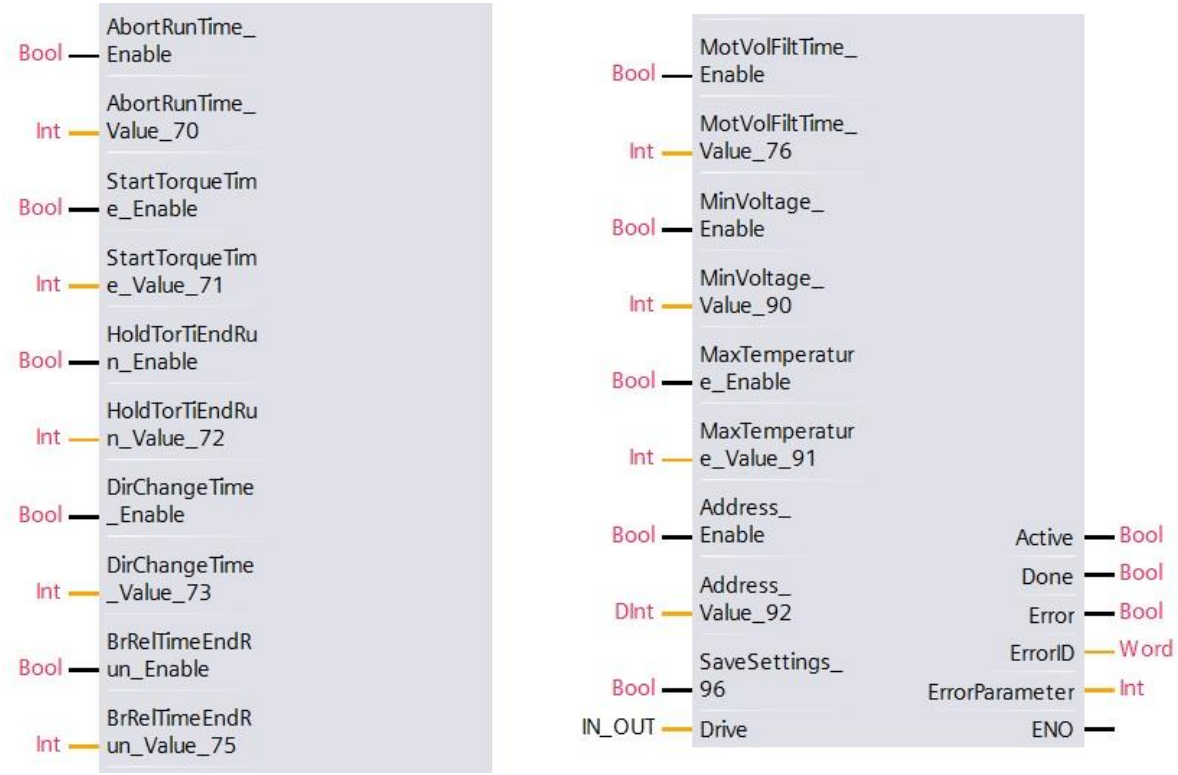

BOOL AbortRunTime_Enable
INT AbortRunTime_Value_70
BOOL StartTorqueTime_Enable
INT StartTorqueTime_Value_71
BOOL HoldTorTiEndRun_Enable
INT HoldTorTiEndRun_Value_72
BOOL DirChangeTime_Enable
INT DirChangeTime_Value_73
BOOL BrRelTimeEndRun_Enable
INT BrRelTimeEndRun_Value_75
BOOL MotVolFiltTime_Enable
INT MotVolFiltTime_Value_76
BOOL MinVoltage_Enable
INT MinVoltage_Value_90
BOOL MaxTemperature_Enable
INT MaxTemperature_Value_91
BOOL Address_Enable
DINT Address_Value_92
BOOL SaveSettings_96
IN_OUT Drive
Active BOOL
Done BOOL
Error BOOL
ErrorID WORD
ErrorParameter INT
ENO

(View in Step 7 Classic V5.5)


<???>
%FB114
"MC_Parametrization"
... EN
Bool Execute
Bool DeliveryState_96
Bool DirRotation_Enable
Int DirRotation_Value_37
Bool PosScaleNumerat_Enable
Int PosScaleNumerat_Value_38
Bool PosScaleDenomin_Enable
Int PosScaleDenomin_Value_39
Bool ActualValue_Enable
DInt ActualValue_Value_3
Bool ReferencValue_Enable
DInt ReferencValue_Value_40
Bool UpperMappingEnd_Enable
DInt UpperMappingEnd_Value_41
Bool UpperLimit_Enable
DInt UpperLimit_Value_42
Bool LowerLimit_Enable
DInt LowerLimit_Value_43
Bool PositionWindow_Enable
DInt PositionWindow_Value_44
Bool LoopLenght_Enable
DInt LoopLenght_Value_45


Bool DragError_Enable
DInt DragError_Value_46
Bool Readjustment_Enable
Int Readjustment_Value_47
Bool DragErrorCorr_Enable
Int DragErrorCorr_Value_48
Bool TargetSpeed_Enable
Int TargetSpeed_Value_53
Bool TargSpeedManRun_Enable
Int TargSpeedManRun_Value_56
Bool SpeedLimitAbort_Enable
Int SpeedLimitAbort_Value_57


Bool Acceleration_Enable
Int Acceleration_Value_58
Bool Deceleration_Enable
Int Deceleration_Value_59
Bool MaxStartTorque_Enable
Int MaxStartTorque_Value_63
Bool MaxTorque_Enable
Int MaxTorque_Value_64
Bool MaxHolTorEndRun_Enable
Int MaxHolTorEndRun_Value_65
Bool MaxHoldTorque_Enable
Int MaxHoldTorque_Value_66


Bool AbortRunTime_Enable
Int AbortRunTime_Value_70
Bool StartTorqueTime_Enable
Int StartTorqueTime_Value_71
Bool HoldTorTiEndRun_Enable
Int HoldTorTiEndRun_Value_72
Bool DirChangeTime_Enable
Int DirChangeTime_Value_73
Bool BrRelTimeEndRun_Enable
Int BrRelTimeEndRun_Value_75
Bool MotVolFiltTime_Enable
Int MotVolFiltTime_Value_76
Bool MinVoltage_Enable
Int MinVoltage_Value_90
Bool MaxTemperature_Enable
Int MaxTemperature_Value_91
Bool Address_Enable
DInt Address_Value_92
Bool SaveSettings_96
IN_OUT Drive
Active Bool
Done Bool
Error Bool
ErrorID Word
ErrorParameter Int
ENO

(View in Step 7 TIA V13 + V14)

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_96" 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_96" 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
- Direction: 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
- Direction: INPUT

However, device name and IP address stay unaffected.

x_Enable
If set, the corresp. parameter is written

- Type: BOOL
- Initial value: FALSE
- Direction: INPUT

x_Value
Desired value of the parameter

- Initial value: 0
- Direction: 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__-PN.

SaveSettings
Saving the settings permanently

- Type: BOOL
- Initial value: FALSE
- Direction: INPUT

Active
Bit is set as long as the parametrization process is active

- Type: BOOL
- Default value: FALSE
- Direction: 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
- Direction: 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
- Direction: OUTPUT

ErrorID
Error code (see table “ErrorID” in chapter 4)

- Type: WORD
- Default value: 0
- Direction: OUTPUT

Drive errors are not considered during a parametrization.

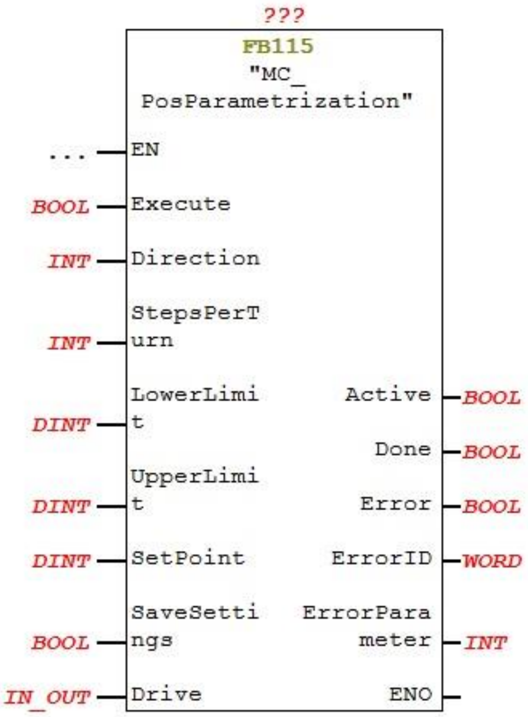
ErrorParameter
Parameter number that caused the error (in case of an error)

- Type: INT
- Default value: 0
- Direction: OUTPUT

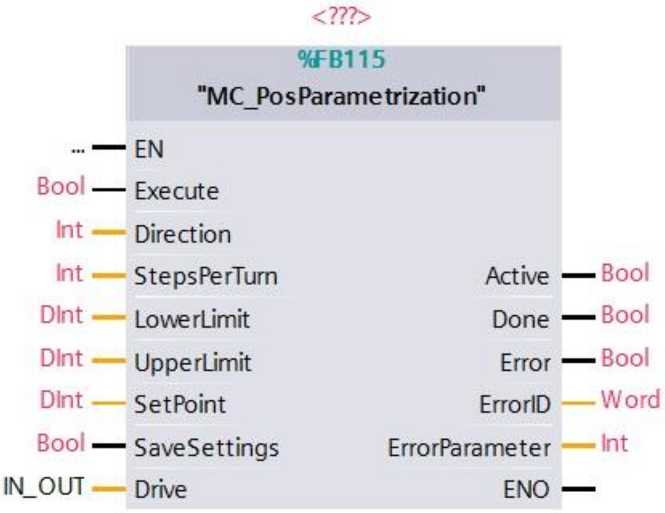
If no error occurred, this value is 0.

5.13 MC_PositionParametrization (FB115)

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).



(View in Step 7 Classic V5.5)



(View in Step 7 TIA V13 + V14)

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. 37) = Direction
  - Position scaling, numerator (Par. 38) = 400
  - Position scaling, denominator (Par. 39) = StepsPerTurn
  - Actual value (Par. 3) = SetPoint
  - If (SetPoint > UpperLimit):
    Upper mapping end (Par. 41) = SetPoint + (3 x StepsPerTurn)
    sonst:
    Upper mapping end (Par. 41) = UpperLimit + (3 x StepsPerTurn)
  - Upper limit (Par. 42) = UpperLimit
  - Lower limit (Par. 43) = LowerLimit
- The number of steps per revolution “StepsPerTurn” directly results in the value of the parameter “Position scaling, denominator” (Par. 39). Thereby it is assumed that the value of “Position scaling, numerator” (Par. 38) 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
StepsPerTurn ≥ 1	16#6140	39
StepsPerTurn ≤ 10000	16#6140	39
LowerLimit ≤ UpperLimit	16#6140	42
(UpperLimit - LowerLimit) / StepsPerTurn ≤ 250	16#6140	43
Falls SetPoint < LowerLimit: (UpperLimit - SetPoint) / StepsPerTurn ≤ 250	16#6140	3
Falls SetPoint > UpperLimit: (SetPoint - LowerLimit) / StepsPerTurn ≤ 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
- Direction: 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
- Direction: INPUT

StepsPerTurn
Number of steps per revolution at the output shaft (resolution)
- Type: INT
- Initial value: 0
- Direction: INPUT

LowerLimit
Lower limit
- Type: DINT
- Initial value: 0
- Direction: INPUT

UpperLimit
Upper limit
- Type: DINT
- Initial value: 0
- Direction: INPUT

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

- Type: DINT
- Initial value: 0
- Direction: INPUT

SaveSettings
Saving the settings permanently

- Type: BOOL
- Initial value: FALSE
- Direction: INPUT

Active
Bit is set as long as the parametrization process is active

- Type: BOOL
- Default value: FALSE
- Direction: 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
- Direction: 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
- Direction: OUTPUT

ErrorID
Error code (see table “ErrorID” in chapter 4)

- Type: WORD
- Default value: 0
- Direction: 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
- Direction: OUTPUT

If no error occurred, this value is 0.