



## **Function Blocks for PSx-3\_\_ with POWERLINK 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](mailto:info@halstrup-walcher.de)  
Internet: [www.halstrup-walcher.de](http://www.halstrup-walcher.de)

## Table of Contents

<b>1</b>	<b>Safety precautions .....</b>	<b>4</b>
1.1	Appropriate use .....	4
1.2	Symbols .....	4
<b>2</b>	<b>Data Structure PSx_type.....</b>	<b>5</b>
<b>3</b>	<b>Error Description (Error ID) .....</b>	<b>6</b>
<b>4</b>	<b>Description and use of the function blocks .....</b>	<b>8</b>
4.1	Adding the library "h_w" .....	8
4.2	Generating variables for each positioning drive.....	10
4.3	Assignment of channel names to process variables ("I/O Mapping").....	10
4.4	Setting the variables "Device" and "NodeID" .....	11
4.5	Generating instances of the desired function blocks .....	11
4.6	Setting the input variable "pDrive" .....	12
4.7	Setting further input and output variables .....	12
4.8	Commonalities of all function blocks .....	12
4.9	Lockings between the function blocks .....	13
4.10	hwMC_Move.....	13
4.11	hwMC_Error.....	16
4.12	hwMC_ReadParameter.....	17
4.13	hwMC_Parametrization.....	21
4.14	hwMC_PositionParametrization .....	25
<b>5</b>	<b>Example project.....</b>	<b>28</b>

## Purpose of instruction manual

This instruction manual describes the function blocks for the PSx-3\_\_-PL (with POWERLINK 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.

**© 2016**

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\_\_-PL 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\_\_-PL 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 PSx\_type

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. For this purpose, the address of this instance of PSx\_type is assigned to the input variable "pDrive" of the corresp. function block.

**Example:** `hwMC_Move_1.pDrive := ADR(Motor_1);`

Hereby it shall be prevented for example, that two FBs accesses the service data channel of a single drive.

Furthermore, with the help of this data structure the addresses for the input and output data of the corresponding drive are assigned to program variables.

Data structure PSx\_type:

Parameter name	Data type	Written by	Description
NodeID	USINT	User	Node number of drive
Device	STRING[30]	User	Device name of POWERLINK interface (z.B. "IF3")
In	PSxIn_type	see below	Input data (Data from drive to PLC)
Out	PSxOut_type	see below	Output data (Data from PLC to drive)
State	USINT	Function blocks	Indicates if currently one of the function blocks reserves the service data channel to this drive = 0 → SDO channel not reserved > 0 → SDO channel reserved

Data structure PSxIn\_type:

Parameter name	Data type	Written by	Description
ModuleOK	BOOL	PLC	Connection to the drive FALSE → not ok TRUE → ok
StatusWord	UINT	Drive	Status word
ActualPosition	DINT	Drive	Actual position
ActualSpeed	INT	Drive	Actual speed in rpm

Data structure PSxOut\_type:

Parameter name	Data type	Written by	Description
ControlWord	UINT	User	Control word
TargetPosition	DINT	User	Target position

### 3 Error Description (Error ID)

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

ErrorID (hex)	Description
<b>16xF000 (mask)</b>	<b>FB</b>
16#1xxx	Error in hwMC_Move
16#2xxx	Error in hwMC_Error
16#3xxx	Error in hwMC_ReadParameter
16#4xxx	Error in hwMC_Parametrization
16#5xxx	Error in hwMC_PositionParametrization
<b>16#0F00 (mask)</b>	<b>Internal FB and PD errors</b>
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
<b>16#00F0 (mask)</b>	<b>Parameter errors</b>
16#xx1x	Parameter: SDO protocol timed out
16#xx2x	Parameter: attempt to read a write only object
16#xx3x	Parameter: attempt to write a read only object
16#xx4x	Parameter: object does not exist in the object dictionary
16#xx5x	Parameter: access failed due to an hardware error
16#xx6x	Parameter: length of service parameter does not match
16#xx7x	Parameter: Sub-index does not exist
16#xx8x	Parameter: value range of parameter exceeded
16#xx9x	Parameter: value of parameter written too high
16#xxAx	Parameter: value of parameter written too low
16#xxBx	Parameter: access failed due to the present device state
16#xxFx	Parameter: other error
<b>16#000F (mask)</b>	<b>Drive errors</b>
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:

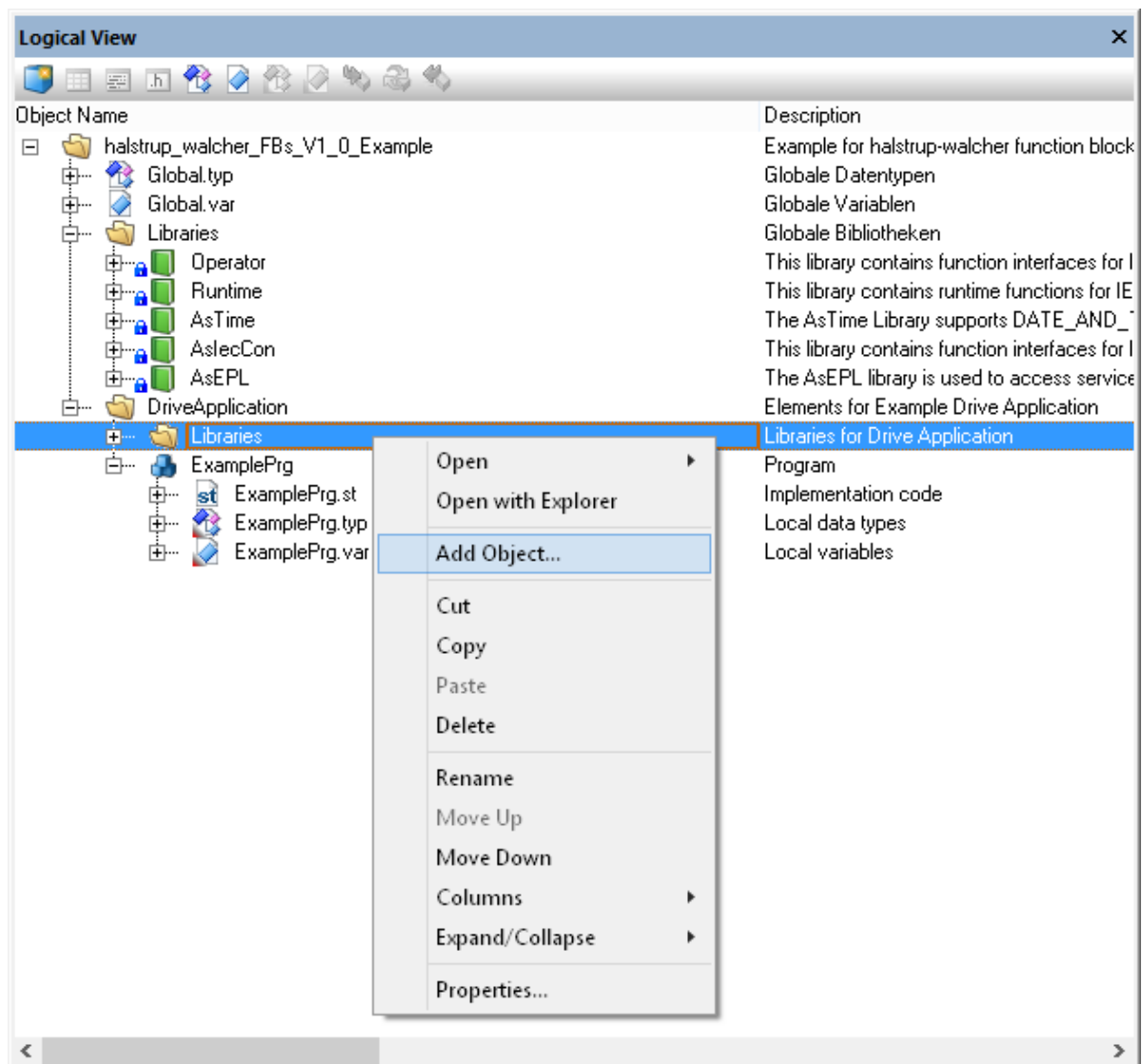
- Run command (hwMC\_Move) with incorrect target value → ErrorID = 16#1008
- Writing a parameter (hwMC\_WriteParameter) with too high value → ErrorID = 16#4090

## 4 Description and use of the function blocks

Initially, this chapter comprises a description how the function blocks are used, afterwards detailed descriptions of each function block will follow.

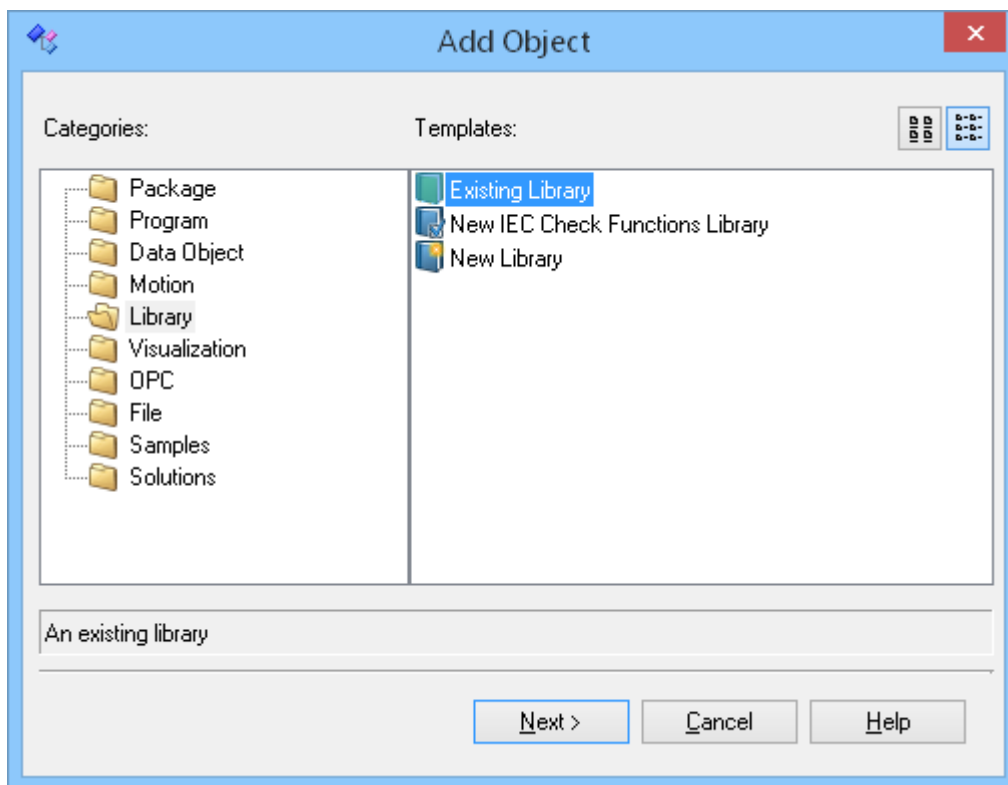
### 4.1 Adding the library “h\_w”

In the first step the library “h\_w” is added to the application project, this happens in the following way:



First download the library from the halstrup-walcher website. Then, if applicable, generate a subdirectory “Libraries” for embedded libraries in the Automation Studio project (if there’s not yet available a suitable place for embedded libraries in the project). In this directory then open the pop-up-menu with right-click and there click on “Add Object”.





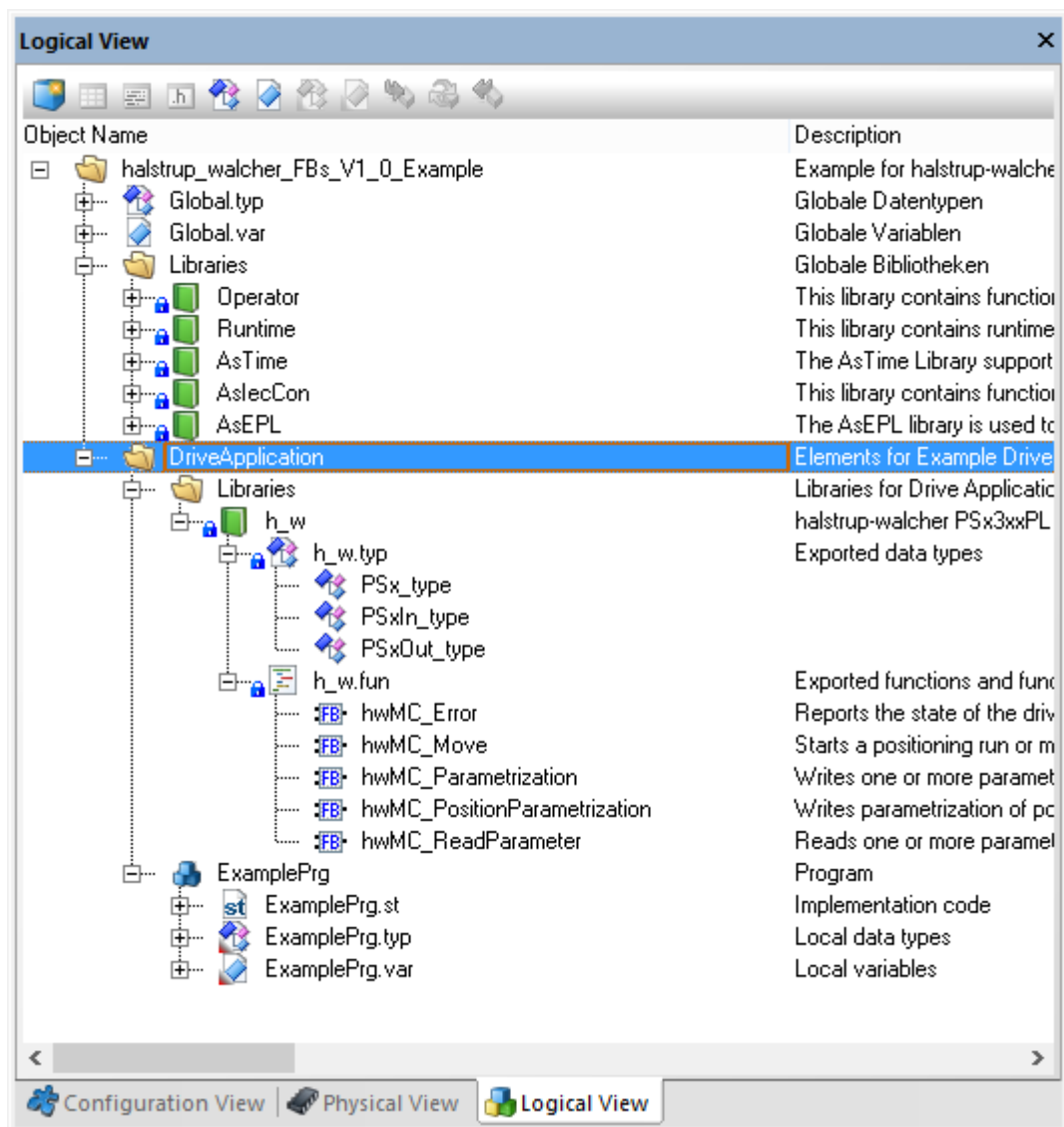
Then click on “Library → Existing Library”. In the following dialog press the button “Project folder”, then browse to the library “h\_w”, select it and click “Next”.

Then in the following dialog assign the library to the active CPU and finish.



If the B&R library “AsEPL” is not yet part of the project at this moment, this library will be added automatically. AsEPL provides mechanisms that are required by the library “h\_w”. If applicable, AsEPL might be shifted from the drive specific library folder to the library folder that is comprehensive for the whole project in order to announce that AsEPL might also be used by other applications within the project.

As result, the library h\_w presents itself in the following way:



## 4.2 Generating variables for each positioning drive

For each drive that shall be accessed by one or more function blocks, a variable of the type "PSx\_type" has to be generated. These variables might be declared global, or local in that program that executes the function blocks.

E.g. for two drives, the following variables have to be generated (here in the language "ST"):

```
Motor_1: PSx_type
```

```
Motor_2: PSx_type
```

## 4.3 Assignment of channel names to process variables ("I/O Mapping")

The function blocks access the input and output data of the drives they use. In the data type "PSx\_type" already appropriate elements are provided, but still these have to be assigned to

the I/O channels of the drives. For the two drives according the example above, the following assignments result:

Assignment of the channels of motor 1:

```
ModuleOk → Motor_1.In.ModuleOK
TargetPosition_I2001Out → Motor_1.Out.TargetPosition
ActualPosition_I2003In → Motor_1.In.ActualPosition
ControlWord_I2024Out → Motor_1.Out.ControlWord
StatusWord_I2025 → Motor_1.In.StatusWord
ActualSpeed_I2030 → Motor_1.In.ActualSpeed
```

Assignment of the channels of motor 2:

```
ModuleOk → Motor_2.In.ModuleOK
TargetPosition_I2001Out → Motor_2.Out.TargetPosition
ActualPosition_I2003In → Motor_2.In.ActualPosition
ControlWord_I2024Out → Motor_2.Out.ControlWord
StatusWord_I2025 → Motor_2.In.StatusWord
ActualSpeed_I2030 → Motor_2.In.ActualSpeed
```

These assignments are generated by selecting the corresp. Powerlink node in the Physical View, opening the pop-up-menu with right-click and selecting "I/O Mapping".



Prior to this, these nodes must have been generated like usual, e.g. with the help of the System Designer. For this purpose, load the device description file (the so-called XDD file) of the PSx3xxPL into the hardware catalogue of the Automation Studio. This file is available on the halstrup-walcher website.

#### 4.4 Setting the variables "Device" and "NodeID"

In the data type "PSx\_type" there are two elements present that are accessed by the function blocks:

- Node number of the drives
- Device name of the POWERLINK interface (e.g. "IF3")

Assuming that the two drives have the node numbers 1 and 2 and that they are connected to the standard Powerlink interface, the following values should be set to these variables (here in the language "ST"):

```
Motor_1.NodeID := 1;
Motor_1.Device := 'IF3';
Motor_2.NodeID := 2;
Motor_2.Device := 'IF3';
```

These settings are fixed during the execution of the program, so e.g. in an ST program they might be placed in the INIT part.

#### 4.5 Generating instances of the desired function blocks

For each function block that shall be used in the program, now one instance per drive has to be generated.

If, as an example, the function blocks `hwMC_Move`, `hwMC_Error`, `hwMC_ReadParameter` and `hwMCPositionParametrization` are each designated for two drives, the following function block instances have to be generated (here in the language “ST”):

```
hwMC_Move_1 : hwMC_Move
hwMC_Error_1 : hwMC_Error;
hwMC_ReadParameter_1 : hwMC_ReadParameter;
hwMC_PositionParametrization_1 : hwMC_PositionParametrization;
hwMC_Move_2 : hwMC_Move;
hwMC_Error_2 : hwMC_Error;
hwMC_ReadParameter_2 : hwMC_ReadParameter;
hwMC_PositionParametrization_2 : hwMC_PositionParametrization;
```

#### 4.6 Setting the input variable “pDrive”

The input variable “pDrive” establishes the reference between the corresponding function block instance and the desired drive. “pDrive” has to be set for each function block, in the example above the setting would be in the following way (here in the language “ST”):

```
hwMC_Move_1.pDrive := ADR(Motor_1);
hwMC_Error_1.pDrive := ADR(Motor_1);
hwMC_ReadParameter_1.pDrive := ADR(Motor_1);
hwMC_PositionParametrization_1.pDrive := ADR(Motor_1);
hwMC_Move_2.pDrive := ADR(Motor_2);
hwMC_Error_2.pDrive := ADR(Motor_2);
hwMC_ReadParameter_2.pDrive := ADR(Motor_2);
hwMC_PositionParametrization_2.pDrive := ADR(Motor_2);
```

These settings are fixed during the execution of the program, so e.g. in an ST program they might be placed in the INIT part.

#### 4.7 Setting further input and output variables

The remaining input variables either may be set directly with the desired values or a variable of the same type is declared and assigned to the input variable.

On the same way output variables either may be evaluated or a variable of the same type is declared and assigned to the output variable.

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.

#### 4.8 Commonalities of all function blocks

The function block instances that have been declared in chapter 4.5 now have to be called in a cyclically called task.

The input “pDrive” is not listed any more separately in the following descriptions of the particular function blocks.

#### 4.9 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 service data channel of a Powerlink node.

Thereby the rule applies that when a function block is called, it remains in the state “waiting”, until the service data channel for this node is free. Then this function block reserves the service data channel for the duration of one SDO access.

For the function blocks “hwMC\_Move”, “hwMC\_ReadParameter”, “hwMC\_Parametrization” and “hwMC\_PositionParametrization”, the reservation of the service data channel results in setting the output “Active” to High.

The function block “hwMC\_Error” doesn't use the service data channel.

“hwMC\_Move” uses the service data channel in order to ensure that a run command has been taken over by the drive. The other function blocks use SDO transfers in order to write and read parameter values.

#### 4.10 hwMC\_Move

This FB serves to send run commands to the drive.

FB hwMC_Move				
→	pDrive	UDINT	<input type="checkbox"/>	VAR_INPUT
→	Release	BOOL	<input type="checkbox"/>	VAR_INPUT
→	Position	DINT	<input type="checkbox"/>	VAR_INPUT
→	ManualRunToLargerValues	BOOL	<input type="checkbox"/>	VAR_INPUT
→	ManualRunToSmallerValues	BOOL	<input type="checkbox"/>	VAR_INPUT
→	Active	BOOL	<input checked="" type="checkbox"/>	VAR_OUTPUT
→	InPosition	BOOL	<input checked="" type="checkbox"/>	VAR_OUTPUT
→	ActualPosition	DINT	<input checked="" type="checkbox"/>	VAR_OUTPUT
→	Error	BOOL	<input checked="" type="checkbox"/>	VAR_OUTPUT
→	ErrorID	UINT	<input checked="" type="checkbox"/>	VAR_OUTPUT

##### 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.



In order to move to the same target position e.g. after a blocking condition, the release has to be deasserted and asserted again. Between deasserting and asserting, it has to be scheduled enough time so that the low state is transmitted on the bus at least one time.

#### 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.



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



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: UINT
- Direction: OUTPUT

The error code 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 hwMC\_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

#### 4.11 hwMC\_Error

This FB reports the state of the drive and the FB as error bit, error code ("ErrorID") and as text. In the case that hwMC\_Error as well as hwMC\_Move are activated, the error code of hwMC\_Error is always the same as the error code of the function block hwMC\_Move.

FB	hwMC_Error			
	pDrive	UDINT	<input type="checkbox"/>	VAR_INPUT
	Enable	BOOL	<input type="checkbox"/>	VAR_INPUT
	Error	BOOL	<input checked="" type="checkbox"/>	VAR_OUTPUT
	ErrorID	UINT	<input checked="" type="checkbox"/>	VAR_OUTPUT
	ErrorDescription	STRING[80]	<input checked="" type="checkbox"/>	VAR_OUTPUT

##### 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: UINT
- 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

#### 4.12 hwMC\_ReadParameter

With this FB values of all parameters can be read by the drive

FB	hwMC_ReadParameter			
→	pDrive	UDINT	<input type="checkbox"/>	VAR_INPUT
→	Execute	BOOL	<input type="checkbox"/>	VAR_INPUT
→	StatusWord_Enable	BOOL	<input type="checkbox"/>	VAR_INPUT
→	ActualSpeed_Enable	BOOL	<input type="checkbox"/>	VAR_INPUT
→	ActualPosition_Enable	BOOL	<input type="checkbox"/>	VAR_INPUT
→	ActualTorque_Enable	BOOL	<input type="checkbox"/>	VAR_INPUT
→	MaxTorqueLastRun_Enable	BOOL	<input type="checkbox"/>	VAR_INPUT
→	U_Control_Enable	BOOL	<input type="checkbox"/>	VAR_INPUT
→	U_Motor_Enable	BOOL	<input type="checkbox"/>	VAR_INPUT
→	DeviceTemperature_Enable	BOOL	<input type="checkbox"/>	VAR_INPUT
→	ProductionDate_Enable	BOOL	<input type="checkbox"/>	VAR_INPUT
→	SerialNumber_Enable	BOOL	<input type="checkbox"/>	VAR_INPUT

DeviceVariant_Enable	BOOL	<input type="checkbox"/>	VAR_INPUT
SoftwareVersion_Enable	BOOL	<input type="checkbox"/>	VAR_INPUT
ControlWord_Enable	BOOL	<input type="checkbox"/>	VAR_INPUT
TargetPosition_Enable	BOOL	<input type="checkbox"/>	VAR_INPUT
DirRotation_Enable	BOOL	<input type="checkbox"/>	VAR_INPUT
PosScaleNumerator_Enable	BOOL	<input type="checkbox"/>	VAR_INPUT
PosScaleDenominator_Enable	BOOL	<input type="checkbox"/>	VAR_INPUT
ReferencingValue_Enable	BOOL	<input type="checkbox"/>	VAR_INPUT
UpperMappingEnd_Enable	BOOL	<input type="checkbox"/>	VAR_INPUT
UpperLimit_Enable	BOOL	<input type="checkbox"/>	VAR_INPUT
LowerLimit_Enable	BOOL	<input type="checkbox"/>	VAR_INPUT
PositionWindow_Enable	BOOL	<input type="checkbox"/>	VAR_INPUT
LoopLength_Enable	BOOL	<input type="checkbox"/>	VAR_INPUT
DragError_Enable	BOOL	<input type="checkbox"/>	VAR_INPUT
Readjustment_Enable	BOOL	<input type="checkbox"/>	VAR_INPUT
DragErrorCorrection_Enable	BOOL	<input type="checkbox"/>	VAR_INPUT
TargetSpeed_Enable	BOOL	<input type="checkbox"/>	VAR_INPUT
TargetSpeedManual_Enable	BOOL	<input type="checkbox"/>	VAR_INPUT
SpeedLimitAbort_Enable	BOOL	<input type="checkbox"/>	VAR_INPUT
Acceleration_Enable	BOOL	<input type="checkbox"/>	VAR_INPUT
Deceleration_Enable	BOOL	<input type="checkbox"/>	VAR_INPUT
MaxStartTorque_Enable	BOOL	<input type="checkbox"/>	VAR_INPUT
MaxTorque_Enable	BOOL	<input type="checkbox"/>	VAR_INPUT
MaxHoldTorqueEndRun_Enable	BOOL	<input type="checkbox"/>	VAR_INPUT
MaxHoldTorque_Enable	BOOL	<input type="checkbox"/>	VAR_INPUT
AbortRunTime_Enable	BOOL	<input type="checkbox"/>	VAR_INPUT
StartTorqueTime_Enable	BOOL	<input type="checkbox"/>	VAR_INPUT
MaxHoldTorqueEndRunTime_Enable	BOOL	<input type="checkbox"/>	VAR_INPUT
DirectionChangeTime_Enable	BOOL	<input type="checkbox"/>	VAR_INPUT
BreakReleaseTimeEndRun_Enable	BOOL	<input type="checkbox"/>	VAR_INPUT
MotVoltageFilterTime_Enable	BOOL	<input type="checkbox"/>	VAR_INPUT
GeneralPurpose_1_Enable	BOOL	<input type="checkbox"/>	VAR_INPUT
GeneralPurpose_2_Enable	BOOL	<input type="checkbox"/>	VAR_INPUT
GeneralPurpose_3_Enable	BOOL	<input type="checkbox"/>	VAR_INPUT
GeneralPurpose_4_Enable	BOOL	<input type="checkbox"/>	VAR_INPUT
GeneralPurpose_5_Enable	BOOL	<input type="checkbox"/>	VAR_INPUT
GeneralPurpose_6_Enable	BOOL	<input type="checkbox"/>	VAR_INPUT
GeneralPurpose_7_Enable	BOOL	<input type="checkbox"/>	VAR_INPUT
GeneralPurpose_8_Enable	BOOL	<input type="checkbox"/>	VAR_INPUT
GeneralPurpose_9_Enable	BOOL	<input type="checkbox"/>	VAR_INPUT
GeneralPurpose_10_Enable	BOOL	<input type="checkbox"/>	VAR_INPUT
MinVoltage_Enable	BOOL	<input type="checkbox"/>	VAR_INPUT
MaxTemperature_Enable	BOOL	<input type="checkbox"/>	VAR_INPUT

	Address_Enable	BOOL	<input type="checkbox"/>	VAR_INPUT
	ConfigConnectionTimeout_Enable	BOOL	<input type="checkbox"/>	VAR_INPUT
	SafePositionConnTimeout_Enable	BOOL	<input type="checkbox"/>	VAR_INPUT
	RepetitionTimeSafePosRun_Enable	BOOL	<input type="checkbox"/>	VAR_INPUT
	DeliveryState_Enable	BOOL	<input type="checkbox"/>	VAR_INPUT
	Active	BOOL	<input type="checkbox"/>	VAR_OUTPUT
	Done	BOOL	<input type="checkbox"/>	VAR_OUTPUT
	Error	BOOL	<input type="checkbox"/>	VAR_OUTPUT
	ErrorID	UINT	<input type="checkbox"/>	VAR_OUTPUT
	ErrorParameter	UINT	<input type="checkbox"/>	VAR_OUTPUT
	StatusWord_Value	UINT	<input type="checkbox"/>	VAR_OUTPUT
	ActualSpeed_Value	INT	<input type="checkbox"/>	VAR_OUTPUT
	ActualPosition_Value	DINT	<input type="checkbox"/>	VAR_OUTPUT
	ActualTorque_Value	UINT	<input type="checkbox"/>	VAR_OUTPUT
	MaxTorqueLastRun_Value	UINT	<input type="checkbox"/>	VAR_OUTPUT
	U_Control_Value	UINT	<input type="checkbox"/>	VAR_OUTPUT
	U_Motor_Value	UINT	<input type="checkbox"/>	VAR_OUTPUT
	DeviceTemperature_Value	UINT	<input type="checkbox"/>	VAR_OUTPUT
	ProductionDate_Value	UINT	<input type="checkbox"/>	VAR_OUTPUT
	SerialNumber_Value	UINT	<input type="checkbox"/>	VAR_OUTPUT
	DeviceVariant_Value	STRING[31]	<input type="checkbox"/>	VAR_OUTPUT
	SoftwareVersion_Value	UINT	<input type="checkbox"/>	VAR_OUTPUT
	ControlWord_Value	UINT	<input type="checkbox"/>	VAR_OUTPUT
	TargetPosition_Value	DINT	<input type="checkbox"/>	VAR_OUTPUT
	DirRotation_Value	UINT	<input type="checkbox"/>	VAR_OUTPUT
	PosScaleNumerator_Value	UINT	<input type="checkbox"/>	VAR_OUTPUT
	PosScaleDenominator_Value	UINT	<input type="checkbox"/>	VAR_OUTPUT
	ReferencingValue_Value	DINT	<input type="checkbox"/>	VAR_OUTPUT
	UpperMappingEnd_Value	DINT	<input type="checkbox"/>	VAR_OUTPUT
	UpperLimit_Value	DINT	<input type="checkbox"/>	VAR_OUTPUT
	LowerLimit_Value	DINT	<input type="checkbox"/>	VAR_OUTPUT
	PositionWindow_Value	UINT	<input type="checkbox"/>	VAR_OUTPUT
	LoopLength_Value	DINT	<input type="checkbox"/>	VAR_OUTPUT
	DragError_Value	UINT	<input type="checkbox"/>	VAR_OUTPUT
	Readjustment_Value	USINT	<input type="checkbox"/>	VAR_OUTPUT
	DragErrorCorrection_Value	UINT	<input type="checkbox"/>	VAR_OUTPUT
	TargetSpeed_Value	UINT	<input type="checkbox"/>	VAR_OUTPUT
	TargetSpeedManual_Value	UINT	<input type="checkbox"/>	VAR_OUTPUT
	SpeedLimitAbort_Value	UINT	<input type="checkbox"/>	VAR_OUTPUT
	Acceleration_Value	UINT	<input type="checkbox"/>	VAR_OUTPUT
	Deceleration_Value	UINT	<input type="checkbox"/>	VAR_OUTPUT
	MaxStartTorque_Value	UINT	<input type="checkbox"/>	VAR_OUTPUT
	MaxTorque_Value	UINT	<input type="checkbox"/>	VAR_OUTPUT

MaxHoldTorqueEndRun_Value	UINT		VAR_OUTPUT
MaxHoldTorque_Value	UINT		VAR_OUTPUT
AbortRunTime_Value	UINT		VAR_OUTPUT
StartTorqueTime_Value	UINT		VAR_OUTPUT
MaxHoldTorqueEndRunTime_Value	UINT		VAR_OUTPUT
DirectionChangeTime_Value	UINT		VAR_OUTPUT
BreakReleaseTimeEndRun_Value	UINT		VAR_OUTPUT
MotVoltageFilterTime_Value	UINT		VAR_OUTPUT
GeneralPurpose_1_Value	UDINT		VAR_OUTPUT
GeneralPurpose_2_Value	UDINT		VAR_OUTPUT
GeneralPurpose_3_Value	UDINT		VAR_OUTPUT
GeneralPurpose_4_Value	UDINT		VAR_OUTPUT
GeneralPurpose_5_Value	UDINT		VAR_OUTPUT
GeneralPurpose_6_Value	UDINT		VAR_OUTPUT
GeneralPurpose_7_Value	UDINT		VAR_OUTPUT
GeneralPurpose_8_Value	UDINT		VAR_OUTPUT
GeneralPurpose_9_Value	UDINT		VAR_OUTPUT
GeneralPurpose_10_Value	UDINT		VAR_OUTPUT
MinVoltage_Value	UINT		VAR_OUTPUT
MaxTemperature_Value	UINT		VAR_OUTPUT
Address_Value	UINT		VAR_OUTPUT
ConfigConnectionTimeout_Value	UINT		VAR_OUTPUT
SafePositionConnTimeout_Value	DINT		VAR_OUTPUT
RepetitionTimeSafePosRun_Value	DINT		VAR_OUTPUT
DeliveryState_Value	INT		VAR_OUTPUT

#### Execute

Start of a reading process

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

Description:

When issuing a rising edge, a reading process is started. Those parameters are consecutively read for which the corresponding FB input is set (ending "...\_Enable"). The read value is available at the corresp. output of the FB. If an error occurs, no more parameters are read and the value of the parameter that is actually read stays unchanged (default value: 0).

For a new reading process, a new rising edge has to be generated. When deasserting the bit, the values already read still stay available.

#### Active

Bit is set as long as the reading process is active

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

The bit is deasserted as soon as all requested values have been read or an error occurred or "Execute" is deasserted.

#### Done

Bit is set as soon as all requested values have been read successfully and are available in "...\_Value"

- Type: BOOL

- Default value: FALSE
- Direction: OUTPUT

The bit is deasserted when “Execute” is deasserted.

#### Error

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

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

The bit is deasserted when “Execute” is deasserted.

#### ErrorID

Error code (see table “ErrorID” in chapter 3)

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

Drive errors are not considered when reading a parameter. The value is set to 0 when “Execute” is deasserted.

#### ErrorParameter

SDO number of the parameter that in case of an error has caused the error while reading

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

The value is set to 0 when “Execute” is deasserted.

### 4.13 hwMC\_Parametrization

With this FB parameter values can be written into the drive.

FB hwMC_Parametrization			
◆ pDrive	UDINT	<input type="checkbox"/>	VAR_INPUT
◆ Execute	BOOL	<input type="checkbox"/>	VAR_INPUT
◆ DeliveryState	BOOL	<input type="checkbox"/>	VAR_INPUT
◆ DirRotation_Enable	BOOL	<input type="checkbox"/>	VAR_INPUT
◆ DirRotation_Value	UINT	<input type="checkbox"/>	VAR_INPUT
◆ PosScaleNumerator_Enable	BOOL	<input type="checkbox"/>	VAR_INPUT
◆ PosScaleNumerator_Value	UINT	<input type="checkbox"/>	VAR_INPUT
◆ PosScaleDenominator_Enable	BOOL	<input type="checkbox"/>	VAR_INPUT
◆ PosScaleDenominator_Value	UINT	<input type="checkbox"/>	VAR_INPUT

ActualPosition_Enable	BOOL	<input type="checkbox"/>	VAR_INPUT
ActualPosition_Value	DINT	<input type="checkbox"/>	VAR_INPUT
ReferencingValue_Enable	BOOL	<input type="checkbox"/>	VAR_INPUT
ReferencingValue_Value	DINT	<input type="checkbox"/>	VAR_INPUT
UpperMappingEnd_Enable	BOOL	<input type="checkbox"/>	VAR_INPUT
UpperMappingEnd_Value	DINT	<input type="checkbox"/>	VAR_INPUT
UpperLimit_Enable	BOOL	<input type="checkbox"/>	VAR_INPUT
UpperLimit_Value	DINT	<input type="checkbox"/>	VAR_INPUT
LowerLimit_Enable	BOOL	<input type="checkbox"/>	VAR_INPUT
LowerLimit_Value	DINT	<input type="checkbox"/>	VAR_INPUT
PositionWindow_Enable	BOOL	<input type="checkbox"/>	VAR_INPUT
PositionWindow_Value	UINT	<input type="checkbox"/>	VAR_INPUT
LoopLength_Enable	BOOL	<input type="checkbox"/>	VAR_INPUT
LoopLength_Value	DINT	<input type="checkbox"/>	VAR_INPUT
DragError_Enable	BOOL	<input type="checkbox"/>	VAR_INPUT
DragError_Value	UINT	<input type="checkbox"/>	VAR_INPUT
Readjustment_Enable	BOOL	<input type="checkbox"/>	VAR_INPUT
Readjustment_Value	USINT	<input type="checkbox"/>	VAR_INPUT
DragErrorCorrection_Enable	BOOL	<input type="checkbox"/>	VAR_INPUT
DragErrorCorrection_Value	UINT	<input type="checkbox"/>	VAR_INPUT
TargetSpeed_Enable	BOOL	<input type="checkbox"/>	VAR_INPUT
TargetSpeed_Value	UINT	<input type="checkbox"/>	VAR_INPUT
TargetSpeedManual_Enable	BOOL	<input type="checkbox"/>	VAR_INPUT
TargetSpeedManual_Value	UINT	<input type="checkbox"/>	VAR_INPUT
SpeedLimitAbort_Enable	BOOL	<input type="checkbox"/>	VAR_INPUT
SpeedLimitAbort_Value	UINT	<input type="checkbox"/>	VAR_INPUT
Acceleration_Enable	BOOL	<input type="checkbox"/>	VAR_INPUT
Acceleration_Value	UINT	<input type="checkbox"/>	VAR_INPUT
Deceleration_Enable	BOOL	<input type="checkbox"/>	VAR_INPUT
Deceleration_Value	UINT	<input type="checkbox"/>	VAR_INPUT
MaxStartTorque_Enable	BOOL	<input type="checkbox"/>	VAR_INPUT
MaxStartTorque_Value	UINT	<input type="checkbox"/>	VAR_INPUT
MaxTorque_Enable	BOOL	<input type="checkbox"/>	VAR_INPUT
MaxTorque_Value	UINT	<input type="checkbox"/>	VAR_INPUT
MaxHoldTorqueEndRun_Enable	BOOL	<input type="checkbox"/>	VAR_INPUT
MaxHoldTorqueEndRun_Value	UINT	<input type="checkbox"/>	VAR_INPUT
MaxHoldTorque_Enable	BOOL	<input type="checkbox"/>	VAR_INPUT
MaxHoldTorque_Value	UINT	<input type="checkbox"/>	VAR_INPUT
AbortRunTime_Enable	BOOL	<input type="checkbox"/>	VAR_INPUT
AbortRunTime_Value	UINT	<input type="checkbox"/>	VAR_INPUT
StartTorqueTime_Enable	BOOL	<input type="checkbox"/>	VAR_INPUT
StartTorqueTime_Value	UINT	<input type="checkbox"/>	VAR_INPUT
MaxHoldTorqueEndRunTime_Enable	BOOL	<input type="checkbox"/>	VAR_INPUT

➡	MaxHoldTorqueEndRunTime_Value	UINT	<input type="checkbox"/>	VAR_INPUT
➡	DirectionChangeTime_Enable	BOOL	<input type="checkbox"/>	VAR_INPUT
➡	DirectionChangeTime_Value	UINT	<input type="checkbox"/>	VAR_INPUT
➡	BreakReleaseTimeEndRun_Enable	BOOL	<input type="checkbox"/>	VAR_INPUT
➡	BreakReleaseTimeEndRun_Value	UINT	<input type="checkbox"/>	VAR_INPUT
➡	MotVoltageFilterTime_Enable	BOOL	<input type="checkbox"/>	VAR_INPUT
➡	MotVoltageFilterTime_Value	UINT	<input type="checkbox"/>	VAR_INPUT
➡	GeneralPurpose_1_Enable	BOOL	<input type="checkbox"/>	VAR_INPUT
➡	GeneralPurpose_1_Value	UDINT	<input type="checkbox"/>	VAR_INPUT
➡	GeneralPurpose_2_Enable	BOOL	<input type="checkbox"/>	VAR_INPUT
➡	GeneralPurpose_2_Value	UDINT	<input type="checkbox"/>	VAR_INPUT
➡	GeneralPurpose_3_Enable	BOOL	<input type="checkbox"/>	VAR_INPUT
➡	GeneralPurpose_3_Value	UDINT	<input type="checkbox"/>	VAR_INPUT
➡	GeneralPurpose_4_Enable	BOOL	<input type="checkbox"/>	VAR_INPUT
➡	GeneralPurpose_4_Value	UDINT	<input type="checkbox"/>	VAR_INPUT
➡	GeneralPurpose_5_Enable	BOOL	<input type="checkbox"/>	VAR_INPUT
➡	GeneralPurpose_5_Value	UDINT	<input type="checkbox"/>	VAR_INPUT
➡	GeneralPurpose_6_Enable	BOOL	<input type="checkbox"/>	VAR_INPUT
➡	GeneralPurpose_6_Value	UDINT	<input type="checkbox"/>	VAR_INPUT
➡	GeneralPurpose_7_Enable	BOOL	<input type="checkbox"/>	VAR_INPUT
➡	GeneralPurpose_7_Value	UDINT	<input type="checkbox"/>	VAR_INPUT
➡	GeneralPurpose_8_Enable	BOOL	<input type="checkbox"/>	VAR_INPUT
➡	GeneralPurpose_8_Value	UDINT	<input type="checkbox"/>	VAR_INPUT
➡	GeneralPurpose_9_Enable	BOOL	<input type="checkbox"/>	VAR_INPUT
➡	GeneralPurpose_9_Value	UDINT	<input type="checkbox"/>	VAR_INPUT
➡	GeneralPurpose_10_Enable	BOOL	<input type="checkbox"/>	VAR_INPUT
➡	GeneralPurpose_10_Value	UDINT	<input type="checkbox"/>	VAR_INPUT
➡	MinVoltage_Enable	BOOL	<input type="checkbox"/>	VAR_INPUT
➡	MinVoltage_Value	UINT	<input type="checkbox"/>	VAR_INPUT
➡	MaxTemperature_Enable	BOOL	<input type="checkbox"/>	VAR_INPUT
➡	MaxTemperature_Value	UINT	<input type="checkbox"/>	VAR_INPUT
➡	Address_Enable	BOOL	<input type="checkbox"/>	VAR_INPUT
➡	Address_Value	UINT	<input type="checkbox"/>	VAR_INPUT
➡	ConfigConnectionTimeout_Enable	BOOL	<input type="checkbox"/>	VAR_INPUT
➡	ConfigConnectionTimeout_Value	UINT	<input type="checkbox"/>	VAR_INPUT
➡	SafePositionConnTimeout_Enable	BOOL	<input type="checkbox"/>	VAR_INPUT
➡	SafePositionConnTimeout_Value	DINT	<input type="checkbox"/>	VAR_INPUT
➡	RepetitionTimeSafePosRun_Enable	BOOL	<input type="checkbox"/>	VAR_INPUT
➡	RepetitionTimeSafePosRun_Value	DINT	<input type="checkbox"/>	VAR_INPUT
➡	SaveSettings	BOOL	<input type="checkbox"/>	VAR_INPUT
➡	Active	BOOL	<input checked="" type="checkbox"/>	VAR_OUTPUT
➡	Done	BOOL	<input checked="" type="checkbox"/>	VAR_OUTPUT
➡	Error	BOOL	<input checked="" type="checkbox"/>	VAR_OUTPUT
➡	ErrorID	UINT	<input checked="" type="checkbox"/>	VAR_OUTPUT
➡	ErrorParameter	UINT	<input checked="" type="checkbox"/>	VAR_OUTPUT

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" 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" 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, a currently running parametrization process is aborted.

#### DeliveryState

Loading of the delivery state (initially without saving)

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

#### 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

Corresponding to 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\_\_-PL.

#### 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 deasserted as soon as the parametrization has been finished successfully or an error occurred or “Execute” is deasserted.

#### Done

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

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

The bit is deasserted when “Execute” is deasserted.

#### Error

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

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

The bit is deasserted when “Execute” is deasserted.

#### ErrorID

Error code (see table “ErrorID” in chapter 3)

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

Drive errors are not considered when doing a parametrization. The value is set to 0 when “Execute” is deasserted.

#### ErrorParameter

SDO number of the parameter that in case of an error has caused the error while doing the parametrization

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

The value is set to 0 when “Execute” is deasserted.

### **4.14 hwMC\_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).

FB	hwMC_PositionParametrization			
→	pDrive	UDINT	<input type="checkbox"/>	VAR_INPUT
→	Execute	BOOL	<input type="checkbox"/>	VAR_INPUT
→	Direction	UINT	<input type="checkbox"/>	VAR_INPUT
→	StepsPerTurn	UINT	<input type="checkbox"/>	VAR_INPUT
→	LowerLimit	DINT	<input type="checkbox"/>	VAR_INPUT
→	UpperLimit	DINT	<input type="checkbox"/>	VAR_INPUT
→	SetPoint	DINT	<input type="checkbox"/>	VAR_INPUT
→	SaveSettings	BOOL	<input type="checkbox"/>	VAR_INPUT
→	Active	BOOL	<input type="checkbox"/>	VAR_OUTPUT
→	Done	BOOL	<input type="checkbox"/>	VAR_OUTPUT
→	Error	BOOL	<input type="checkbox"/>	VAR_OUTPUT
→	ErrorID	UINT	<input type="checkbox"/>	VAR_OUTPUT
→	ErrorParameter	UINT	<input type="checkbox"/>	VAR_OUTPUT

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 (SDO #202C) = Direction
  - Position scaling, numerator (SDO #2010) = 400
  - Position scaling, denominator (SDO #2011) = StepsPerTurn
  - Actual value (SDO #2003) = SetPoint
  - If (SetPoint > UpperLimit):
    - Upper mapping end (SDO #2028) = SetPoint + (3 x StepsPerTurn)
  - otherwise:
    - Upper mapping end (SDO #2028) = UpperLimit + (3 x StepsPerTurn)
  - Lower limit (SDO #2017) = LowerLimit
  - Upper limit (SDO #2016) = UpperLimit
- The number of steps per revolution "StepsPerTurn" directly results in the value of the parameter "Position scaling, denominator" (SDO #2011). Thereby it is assumed that the value of "Position scaling, numerator" (SDO #2010) 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#50A0	SDO #2011
StepsPerTurn ≤ 10000	16#5090	SDO #2011
LowerLimit ≤ UpperLimit	16#50A0	SDO #2016
(UpperLimit - LowerLimit) / StepsPerTurn ≤ 250	16#50A0	SDO #2017
If SetPoint < LowerLimit: (UpperLimit - SetPoint) / StepsPerTurn ≤ 250	16#50A0	SDO #2003
If SetPoint > UpperLimit: (SetPoint - LowerLimit) / StepsPerTurn ≤ 250	16#5090	SDO #2003

- 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, a currently running parametrization process is aborted.

#### 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 or "Execute" is deasserted.

#### Done

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

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

The bit is deasserted when "Execute" is deasserted.

#### Error

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

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

The bit is deasserted when "Execute" is deasserted.

#### ErrorID

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

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

Drive errors are not considered when doing a parametrization. The value is set to 0 when "Execute" is deasserted.

#### ErrorParameter

SDO number of the parameter that in case of an error has caused the error while doing the parametrization

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

The value is set to 0 when "Execute" is deasserted.

## 5 Example project

The project "halstrup\_walcher\_FBs\_V1\_0\_Example.apj" contains an example that shows the implementation of the function block library "h\_w". Two positioning drives are introduced ("Motor\_1" and "Motor\_2"), with the program "ExamplePrg" these drives can be run. It's also possible to start a continuous test. In this test, every time when one motor has finished a positioning, the other motor starts. Moreover, in every 10<sup>th</sup> PLC cycle the reading of parameter values is activated. Thereby for example the cyclical reading of the maximum torque occurring during the most recent run could be started.

The FBs “hwMC\_Parametrization” and “hwMC\_PositionParametrization” are called cyclically, the associated inputs might be set (also like additional requirements for the reading of parameter values) in the watch window of “ExamplePrg”:

