


SoftMotion Basic

CoDeSys

Smart Software Solutions

We software Automation.

Training

11.06.2010

SoftMotion Basic

How to use the SoftMotion single axis functionality

Smart Software Solutions

We software Automation.

After this module you will be …

- able to use the PLCopen motion control FunctionBlocks in CoDeSys.
- able to create single axis movements.
- able to create synchronized movements.

Agenda

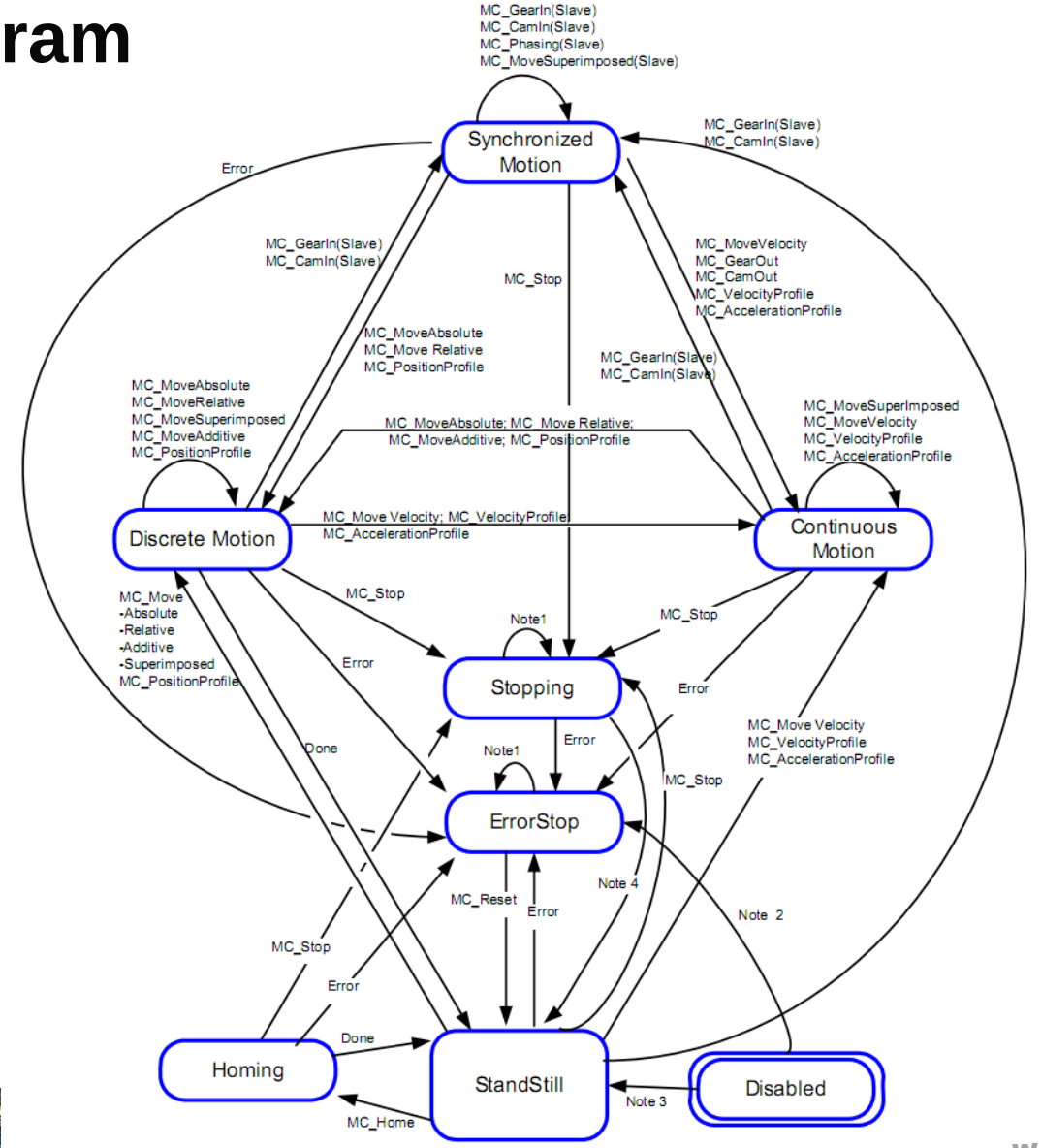
- General information
- PLCopen definitions
- Basic FunctionBlocks
- Administrative FunktionBlocks
- Homing/Probing
- Master/Slave FunctionBlocks

General information

- CoDeSys SoftMotion complies with the PLCopen motion control specifications
- CoDeSys SoftMotion can only be used with a CoDeSys Motion Control PLC
- The SoftMotion functionality is accessed via special FunktionBlocks.
  - FBs beginning with **‘SMC_’** are **3S**-specific implementations;
  - those starting with **‘MC_’** are FBs according to the **PLCopen** MC specification

PLCopen definitions

State diagram



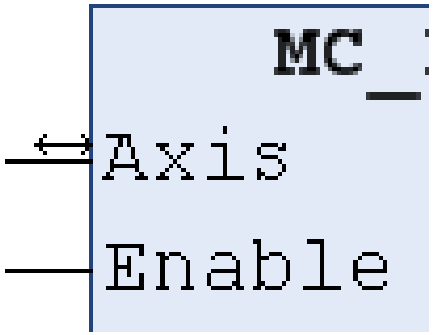
PLCopen definitions

SM3_Basic.library

- This library contains function blocks…
  - for handling, monitoring, parameterization, supervision
  - for generating movements

PLCopen definitions

Behaviour (1)

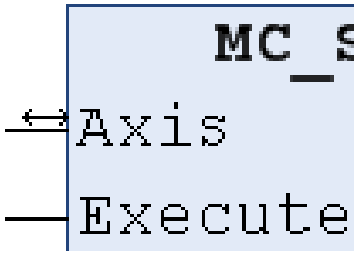


- FBs with Enable input
  - TRUE:
    - FB is active
  - FALSE:
    - FB is inactive

level sensitive

PLCopen definitions

Behaviour (2)

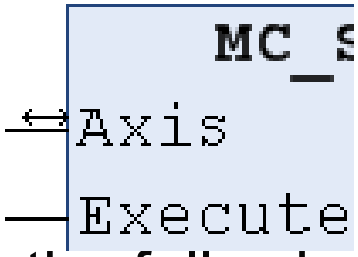


edge sensitive

- FB with Execute input
  - rising edge:
    - acceptance of inputs
    - taking control of the axis
    - start of movement
  - falling edge:
    - all outputs cleared (if this happens before action/ movement is ended, the outputs remain set for one cycle)
  - a falling edge does not stop the movement

PLCopen definitions

behaviour (3)



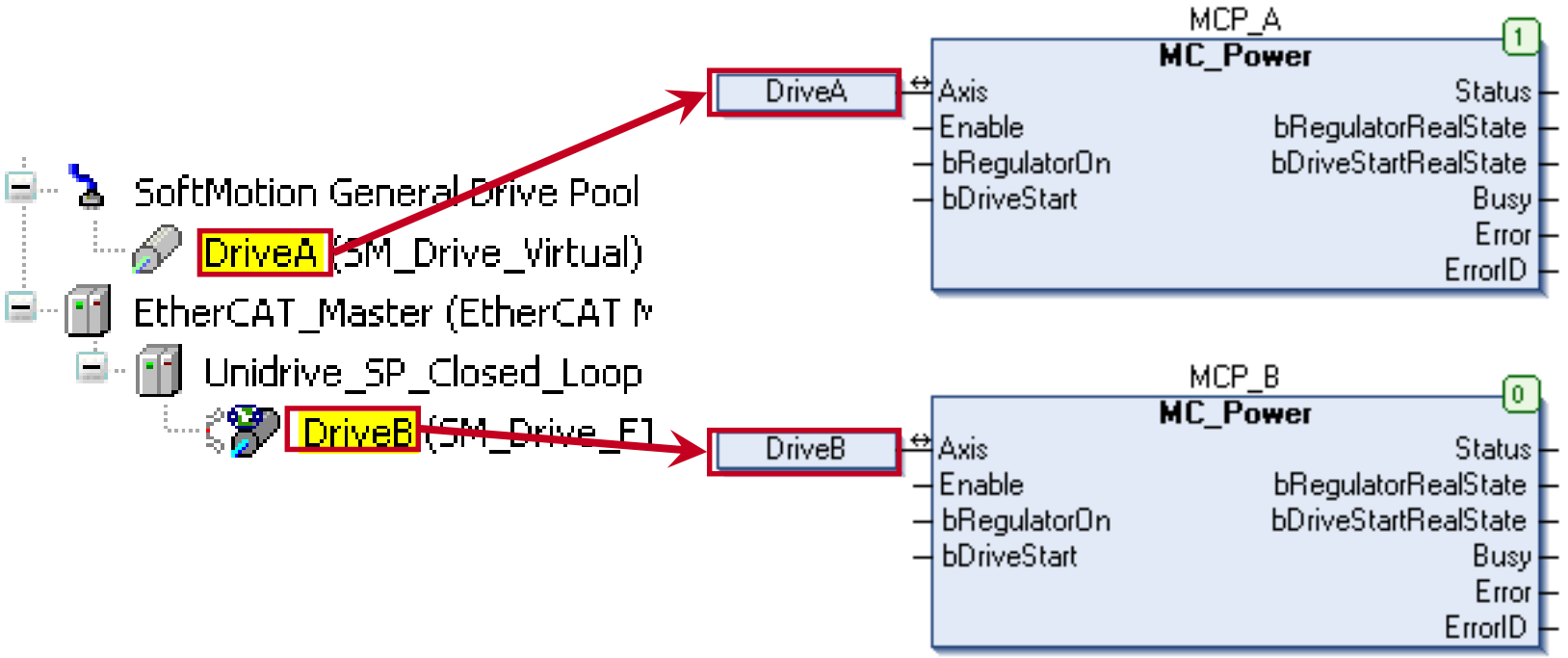
edge sensitive

- FB with Execute input has the following outputs…
  - done (or similar)
    - set, when movement (command) has been successfully completed
  - busy
    - true as long as the FB processes the ordered task
  - CommandAborted
    - during the process another FB has taken control (caused by a rising edge on the execute input) of the axis
  - Error/ErrorID
    - internal error inside FB (not implicitly in drive) (e.g. wrong parameters, drive not enabled…)

Basic FunctionBlocks

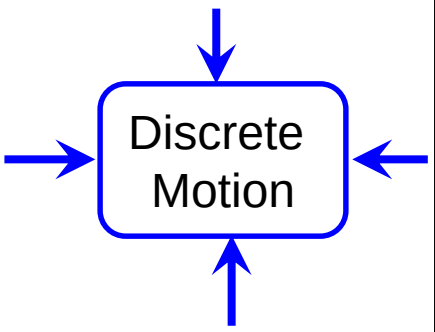
MC_POWER

- enable the power stage (bRegulatorOn)
- disable quickstop mechanism (bDriveStart)



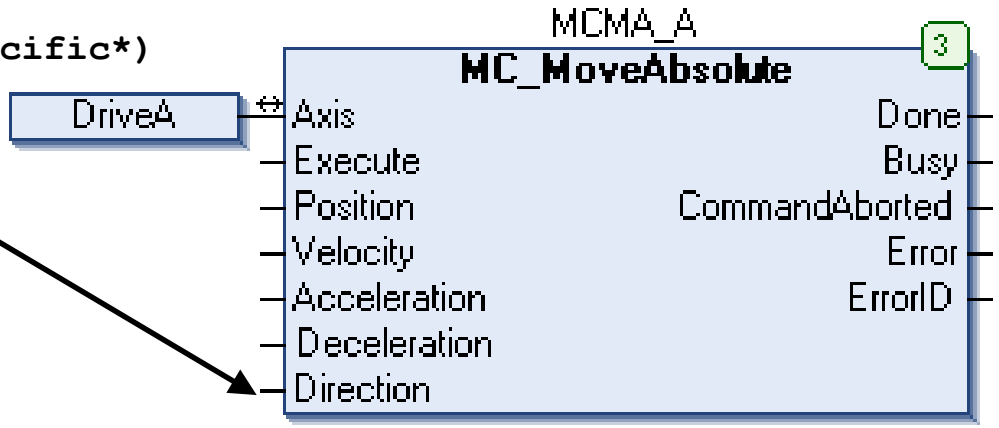
Basic FunctionBlocks

MC_MoveAbsolute



- absolute positioning
- only for rotary axes: modes of direction:
  positive, negative, current, shortest, fastest

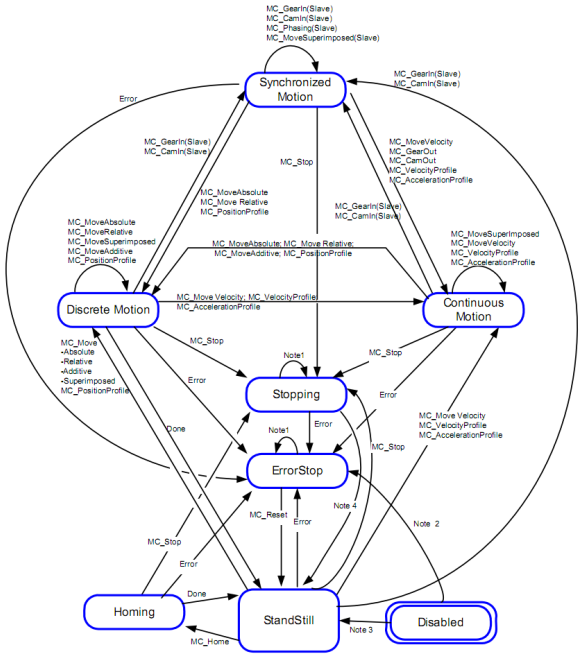
```
TYPE MC_DIRECTION :(
        fastest:=3,(*vendor specific*)
        current:=2,
        positive:=1,
        shortest:=0,
        negative:=-1);
END_TYPE
```

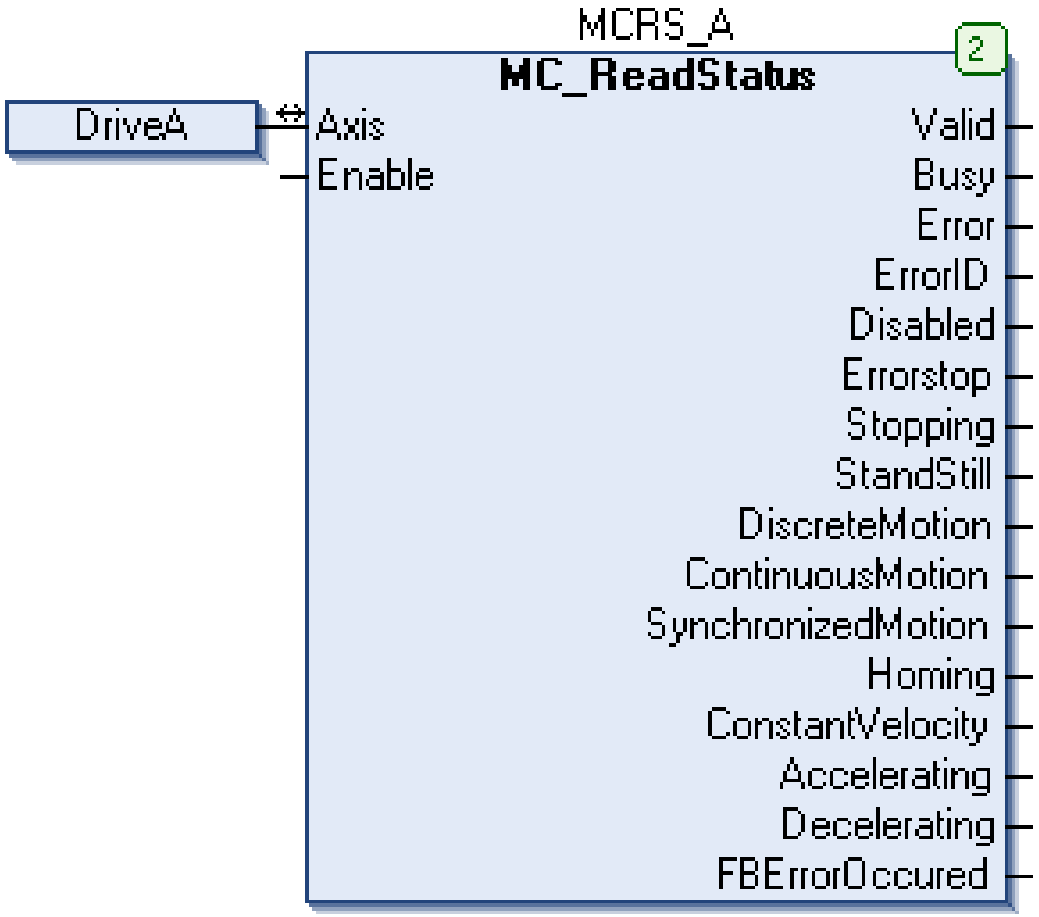


Basic FunctionBlocks

MC_ReadStatus

- reads the state of an axis defined by the PLCopen

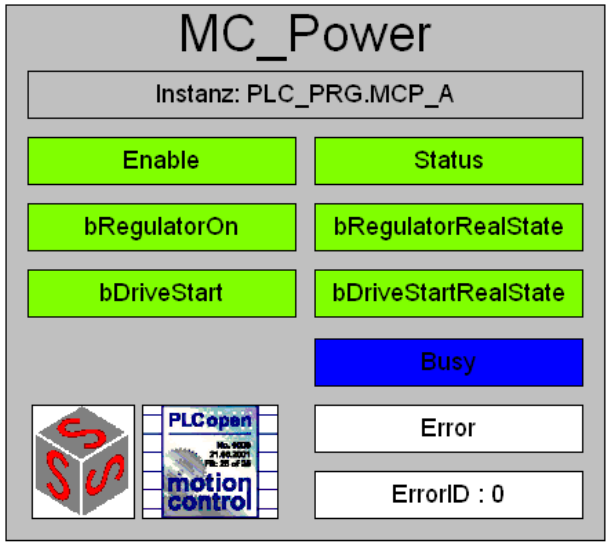


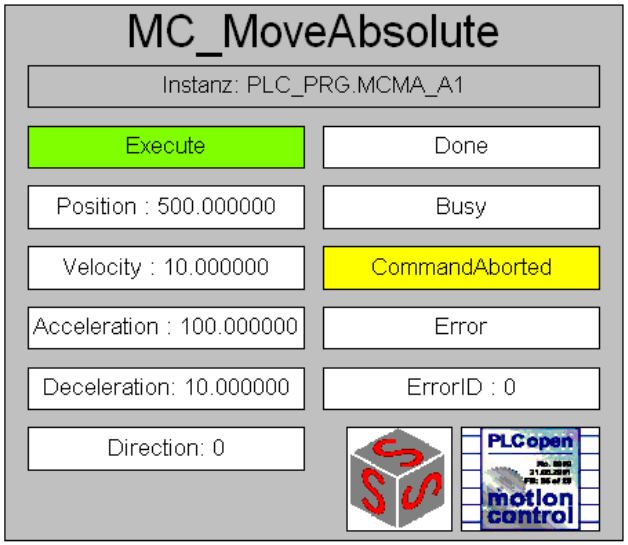


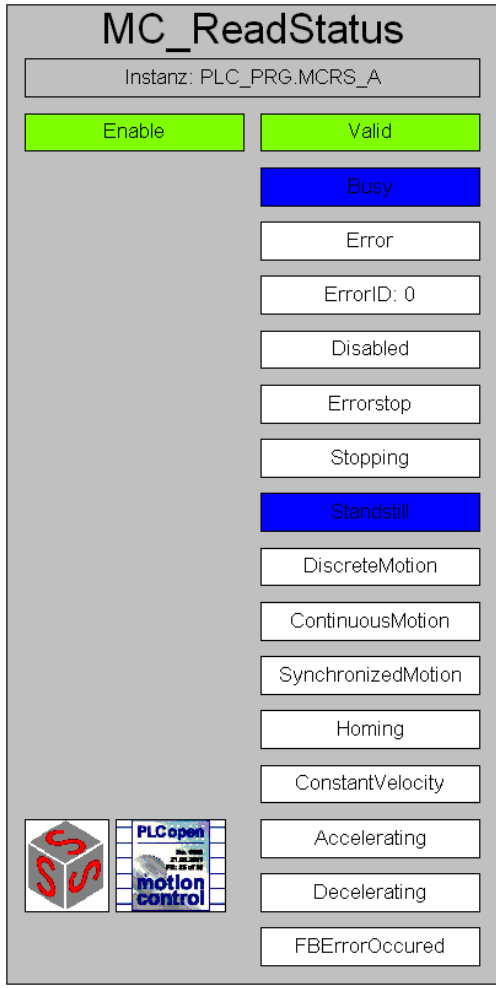
Basic FunctionBlocks

Visualization templates

- The SoftMotion libraries provide visualization templates for all FBs which are especially useful for initial operation.







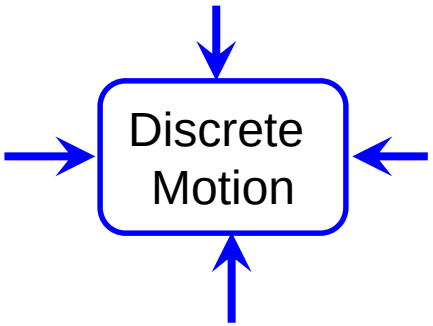
SM_BASIC_EX1

Write a little application for one linear drive, containing an MC_Power FB and two MC_MoveAbsolute FBs!
Use the MC_ReadStatus to evaluate the PLCopen states!

Test the start behaviour of the FBs (Execute, CommandAborted, Done..)!

Use the visualization templates!

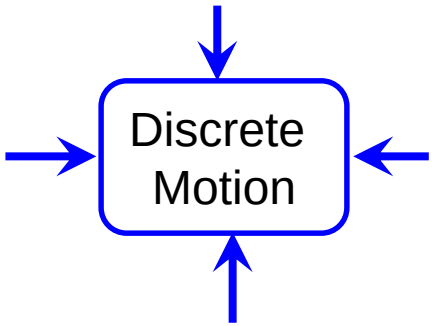
MC_MoveRelative



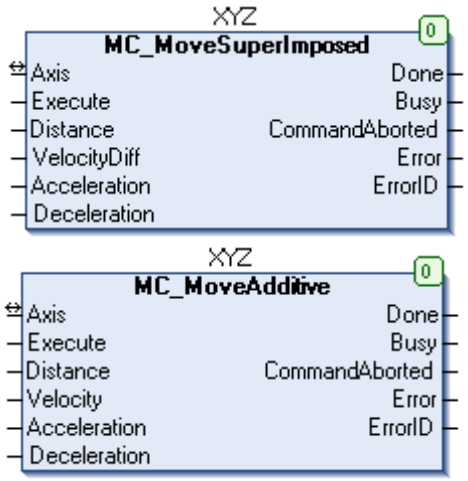
- Moves the drive by a distance relative to the last set position.

Basic FunctionBlocks

Other discrete Motion FBs



- MC_MoveSuperImposed
- MC_MoveAdditive
- MC_PositionProfile



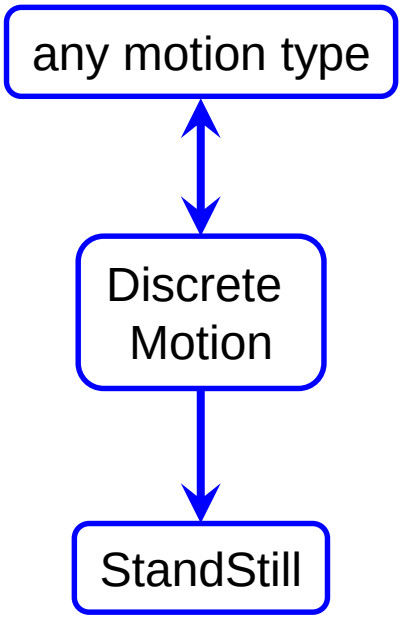
- Need detailed information?
- → refer to CoDeSys Help

Online Help

Basic FunctionBlocks

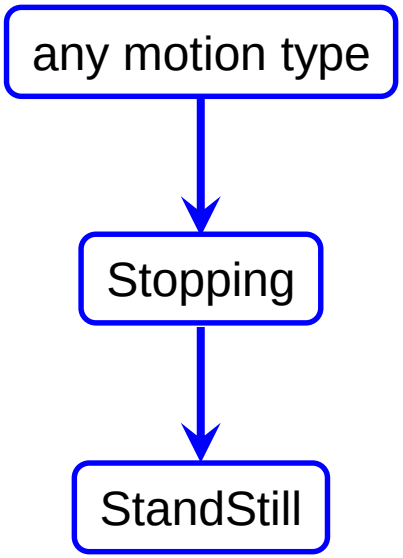
MC_Halt

- slow axis down to standstill



Basic FunctionBlocks

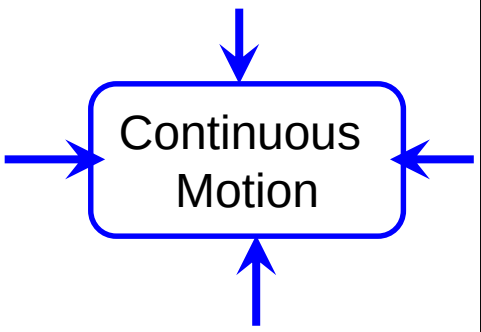
MC_Stop



- slow axis down to standstill
- stopping cannot be interrupted by other FBs
- remains in stopping state
  - until the axis has reached velocity zero
  - and Execute is False

Basic FunctionBlocks

MC_MoveVelocity

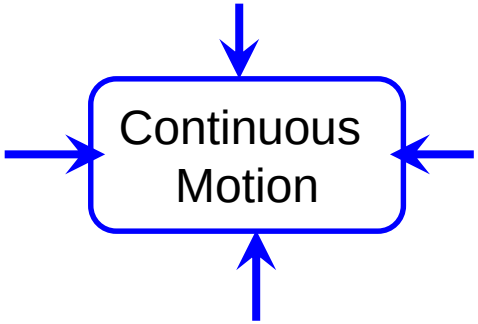


- endless movement of an axis with constant velocity
- output InVelocity shows when the set velocity has been reached
- velocity is always non-negative
- direction is set with input Direction positive/negative/current

```
TYPE MC_DIRECTION :(
        fastest:=3,(*vendor specific*)
        current:=2,
        positive:=1,
        shortest:=0,
        negative:=-1);
END_TYPE
```

Basic FunctionBlocks

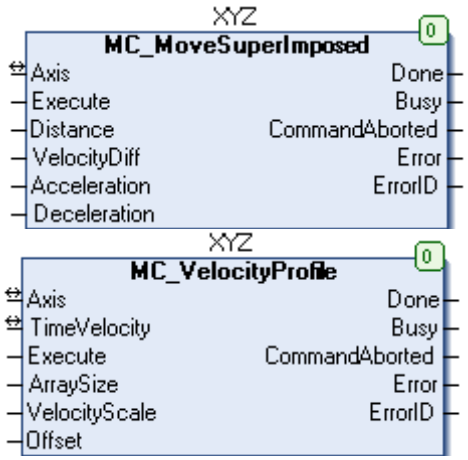
Other continuous Motion FBs



- MC_MoveSuperImposed
- MC_VelocityProfile
- MC_AccelerationProfile

- Need detailed information?
- → refer to CoDeSys Help

Online Help



SM_BASIC_EX2

Program a two-velocity hand control for a linear axis:

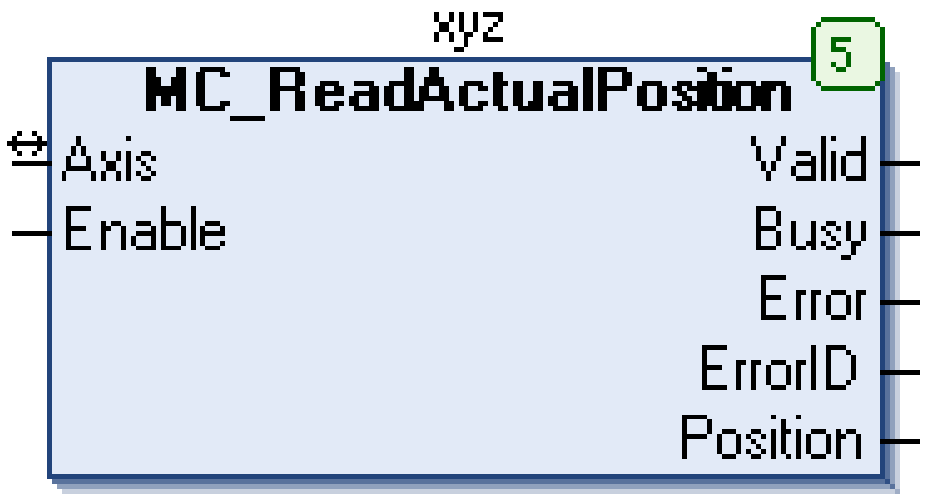
Four buttons (of the HMI) make the axis move at two different velocities either to the right or to the left.
Another button moves the axis back to position zero.

Use visualization templates for testing your program!

Administrative FunctionBlocks

MC_ReadActualPosition

- read actual position
- besides using the FB, reading the values of the AxisREF is also possible
  <DriveName>.fActPosition



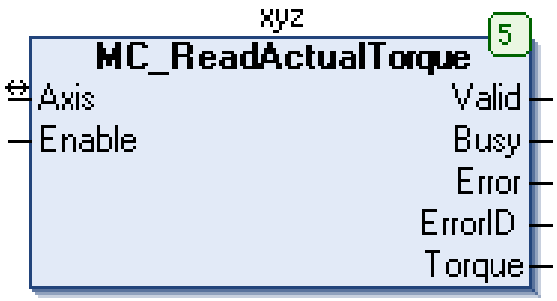
Administrative FunctionBlocks

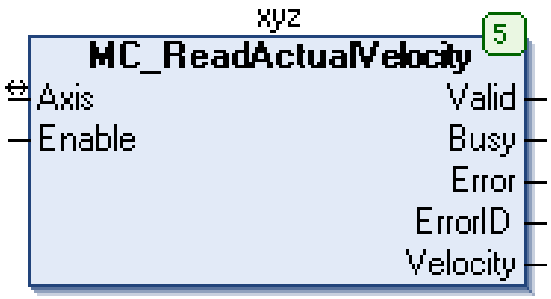
Other FBs for reading the actual values

- MC_ReadActualTorque
- MC_ReadActualVelocity

- Need detailed information?
- → refer to CoDeSys Help

Online Help

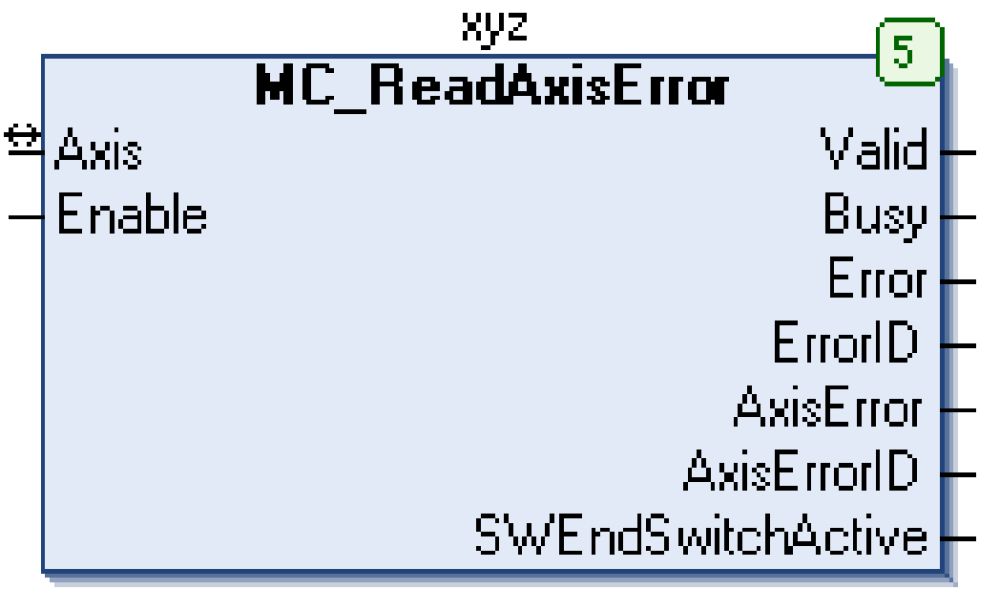




Administrative FunctionBlocks

MC_ReadAxisError

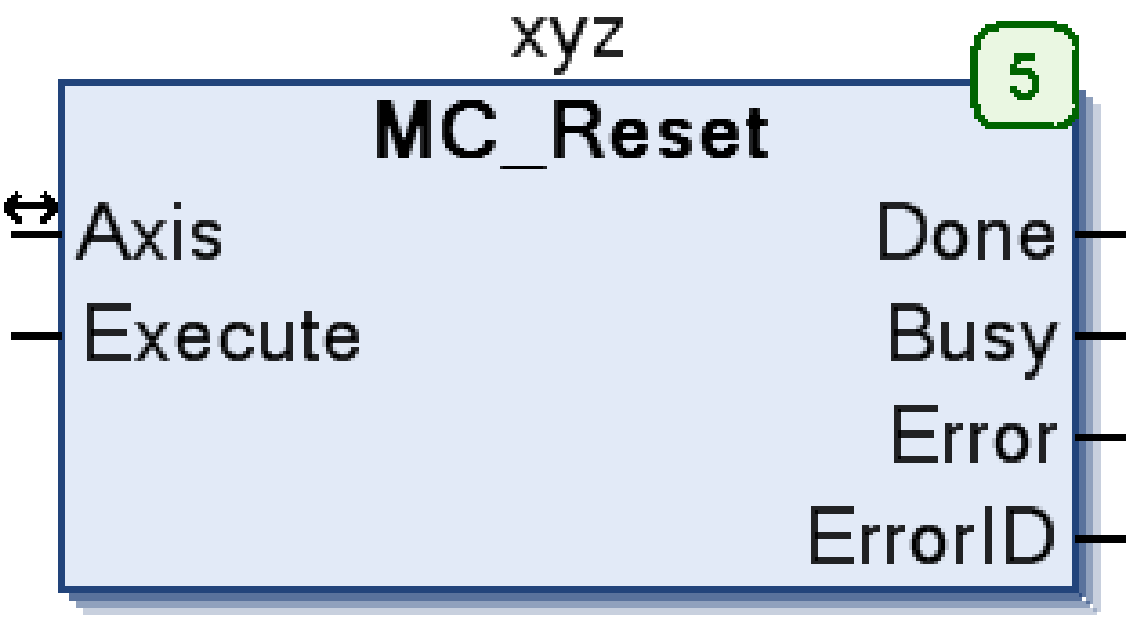
- read drive internal error
- returned AxisErrorID is vendor specific



Administrative FunctionBlocks

MC_Reset

- reset errors



Administrative FunctionBlocks

Handling the error history

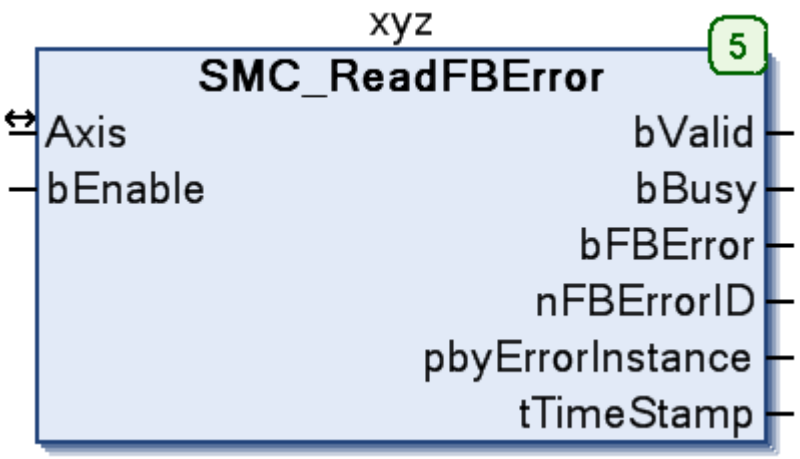
- every axis contains an FB error history

Expression	Type	Value
Device.Application.DriveA.fbeFBError	ARRAY [0..g_SMC_N...	
fbeFBError[0]	SMC_FBERROR	
wID	SMC_ERROR	SMC_AXIS_NOT_READY_FOR_MOTION
pbyErrorInstance	POINTER TO BYTE	16#0316F73E
tTimeStamp	TIME	T#3h13m42s911ms
fbeFBError[1]	SMC_FBERROR	
fbeFBError[2]	SMC_FBERROR	
fbeFBError[3]	SMC_FBERROR	
wID	SMC_ERROR	SMC_REGULATOR_OR_START_NOT_SET
pbyErrorInstance	POINTER TO BYTE	16#0316F778
tTimeStamp	TIME	T#3h27m51s452ms
fbeFBError[4]	SMC_FBERROR	
fbeFBError[5]	SMC_FBERROR	

Administrative FunctionBlocks

Handling the error history

- Read the first error which has occurred at an FB using this axis.



- Clear this error.

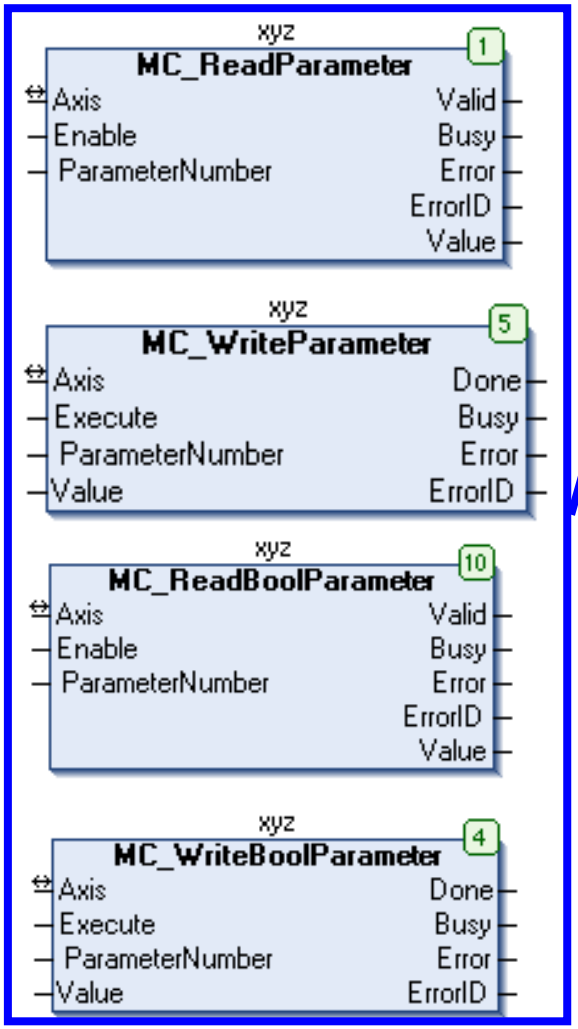
Administrative FunctionBlocks

MC_SetPosition

- Moves the origin of the coordinate system.
- Mode:
  - TRUE: moves the origin by Position
  - FALSE: current position becomes Position

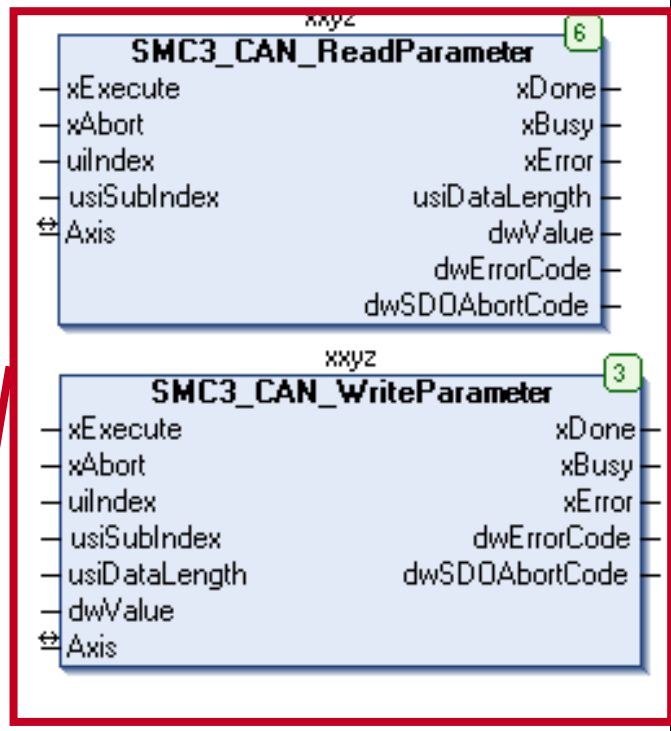
Administrative FunctionBlocks

MC_Read/Write (Bool)Parameter



read/write parameters according to PLCopen

also available bus dependent FBs for reading and writing parameters



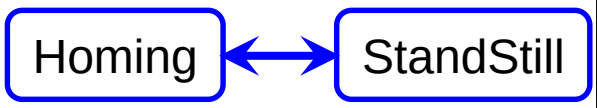
SM_BASIC_EX3

Use the SM_BASIC_EX2 project.

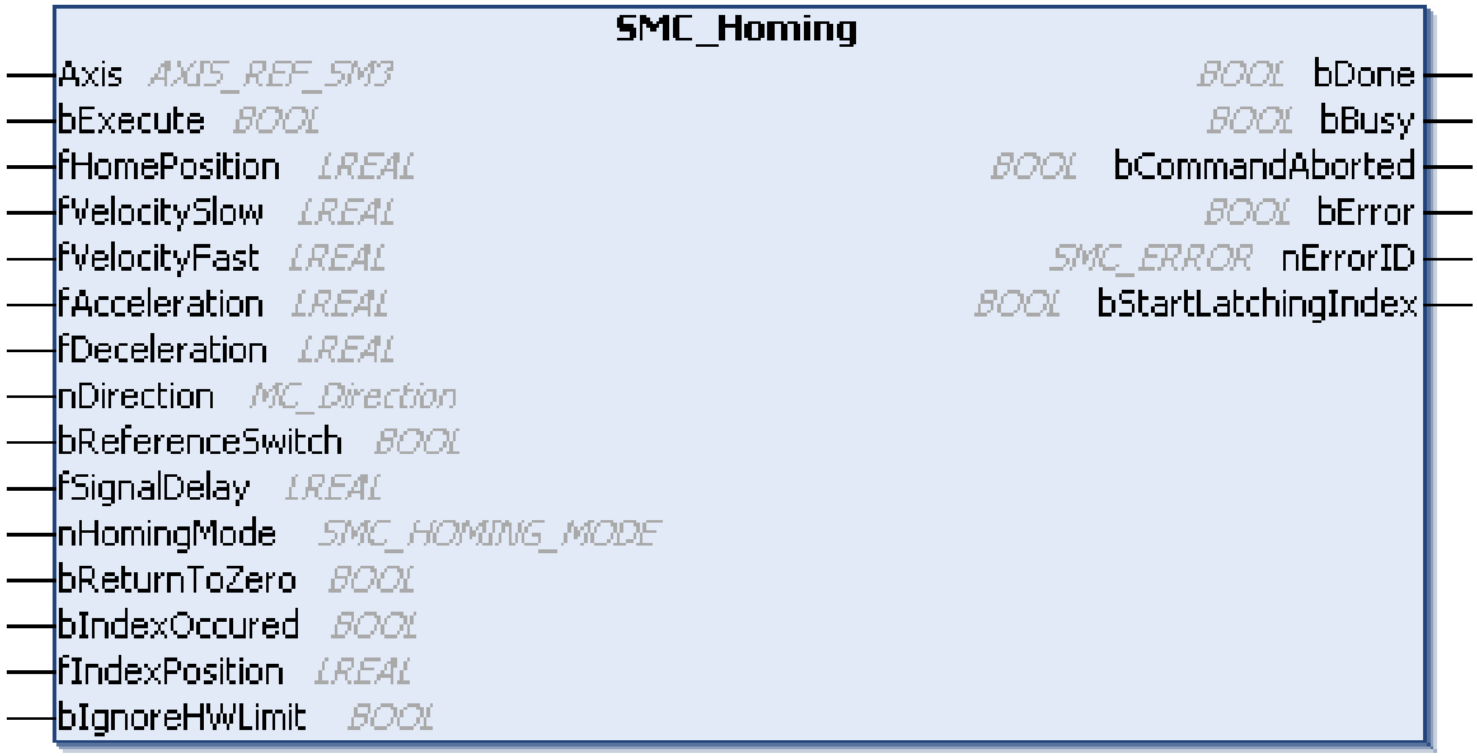
Add an MC_Rest FunctionBlock to reset the drive.
Use an MC_ReadActualPosition to get the current position values.

Use visualization templates for testing your program!

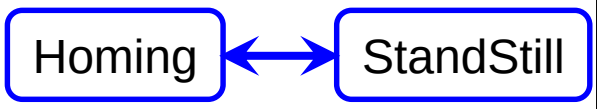
SMC_Homing



- PLC controlled homing



MC_Home



- Drive controlled homing (ordered homing)
  - If homing is not supported by the drive use SMC_Homing

MC_TouchProbe

```
TYPE TRIGGER_REF :
STRUCT

bFastLatching:BOOL:=TRUE;
  iTriggerNumber:INT:=-1;
  bInput:BOOL;
  bActive:BOOL:=FALSE;
END_STRUCT
END_TYPE
```

- TriggerInput: reference to trigger signal
- WindowOnly
  - FALSE: all signals cause a probe
  - TRUE: only between FirstPosition and LastPosition
    - (not supported by all drives)

MC_AbortTrigger

- aborts active probe

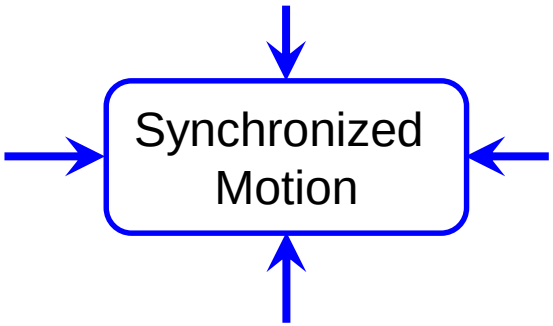
```
TYPE TRIGGER_REF :
STRUCT

bFastLatching:BOOL:=TRUE;
  iTriggerNumber:INT:=-1;
  bInput:BOOL;
  bActive:BOOL:=FALSE;
END_STRUCT
END_TYPE
```

Master/Slave FunctionBlocks

MC_GearIn

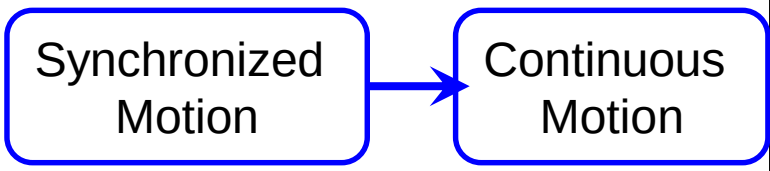
- electronic gear



$$V_{Slave} = \frac{RatioNumerator}{RatioDenomiator} \bullet V_{Master}$$

Master/Slave FunctionBlocks

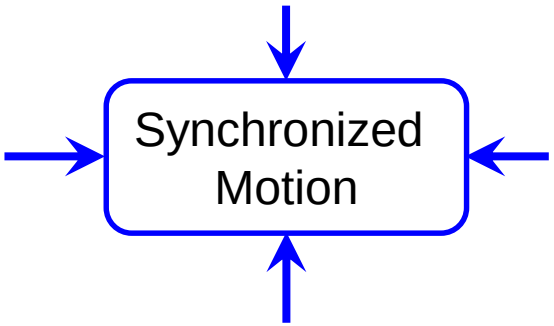
MC_GearOut



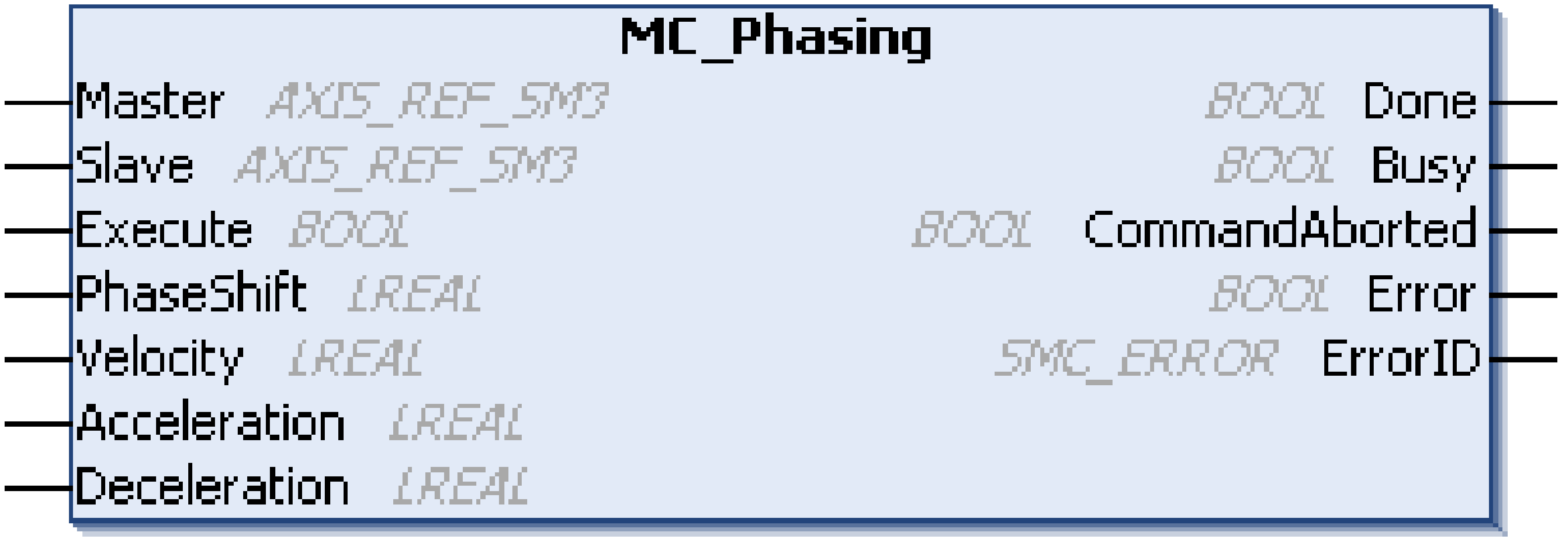
- ungear the slave axis
- slave axis maintains last velocity

Master/Slave FunctionBlocks

MC_Phasing



- phase shifting



SM_BASIC_EX4

Create a program for two axes:

Axis1 is controlled by MC_Power and MC_MoveVelocity (with visualization templates).

Axis2 is always linked to Axis1 with a gearing ratio of 2/3.

Summary

- SoftMotion means
  - FBs according to PLCopen motion control
  - additional FBs
- There are FBs for
  - creating movements
  - supervison
  - administrative tasks
- Further information needed?
  - → refer to CoDeSys Help

Online Help

Thank you for your interest

Smart Software Solutions

We software Automation.