DOCUMENTATION ISG-kernel

PLC library
McpPLCopen Part 1

Short description:
MCP-P1

© Copyright
ISG Industrielle Steuerungstechnik GmbH
STEP, Gropiusplatz 10
D-70563 Stuttgart
All rights reserved
www.isg-stuttgart.de
support@isg-stuttgart.de

Documentation version: 1.08
04/06/2025

Table of contents

1 Definitions 6
1.1 Abbreviations 6
1.2 Explanations of terms 6

2 PLC library “McpPLCopenP1” 8
2.1 PLCopen axis state model 8
2.2 Measurement of axis positions 10
2.2.1 Data structure TRIGGER_REF 10
2.2.2 Behaviour of MC_TouchProbe and MC_AbortTrigger 10
2.2.3 Measurement parameter definition in the axis-specific machine data 11
2.3 Cam functionality 12
2.3.1 Definition range of cams and their processing 12
2.3.2 Cam processing with closed scanning 12
2.3.3 Cam description by interpolation point tables 13
2.3.4 Handling interpolation pointer tables 15
2.3.4.1 Online interpolation of interpolation point tables 15
2.3.4.2 Closed processing for an endlessly rotating slave axis 17
2.3.4.2.1 Example: Simplest possible linear cam for endlessly rotating processing 19
2.3.5 Processing tables with LINE/POLY5 motion segments 20
2.3.5.1 Definition of the motion pattern 20
2.3.5.2 Calculating the polynomial coefficients and interpolation 21
2.3.5.3 Table format 24
2.3.5.4 Characteristics of the tables 25
2.3.5.5 Behaviour of the slave axis at the modulo limit 26
2.3.5.6 Slave positions outside the modulo range 30
2.3.5.7 Modifying tables with motion segments 32
2.3.6 Access to cam tables 33
2.3.7 File format of the cam tables (ASCII interpreter) 34
2.3.7.1 The table header 34
2.3.7.2 Table data 35
2.3.7.3 Table types 35
2.3.7.4 Function types (interpolation types) 36
2.3.7.5 Tokens and separators 36
2.3.7.6 Comments 36
2.3.7.7 Example of a correct file format 37

3 Overview of the PLCopen FBs 38
3.1 MC_Power 41
3.2 MC_ReadAxisInfo 43
3.3 MC_ReadMotionState 45
3.4 MC_ReadStatus 48
3.5 MC_ReadAxisError 50
3.6 MC_ReadActualPosition 51
3.7 MC_Reset 52
3.8 MC_Home 53
3.8.1 Parameter values when the“CamSignal” input is used 54
3.9 MC_MoveAbsolute 55

3.10 MC_MoveAdditive 58
3.11 MC_MoveRelative 61
3.12 MC_MoveSuperImposed 63
3.13 MC_MoveVelocity 66
3.14 MC_Stop 69
3.15 MC_AbortTrigger 71
3.16 MC_TouchProbe 73
3.17 MC_CamIn 75
3.17.1 Values at input “StartMode” 78
3.17.2 Values at the “TableFctType” input 79
3.17.3 Influence of axis parameters on the coupling process 80
3.18 MC_CamOut 82
3.19 MC_CamTableSelect 83
3.20 MC_GearIn 84
3.20.1 Influence of axis parameters on the coupling process 87
3.21 MC_GearOut 89
3.22 MC_Phasing 90
3.23 MC_ReadParameter 93
3.24 MC_WriteParameter 96
3.25 MC_WriteLrealParameter 99
3.26 MC_SetPosition 101
3.27 MC_SetOverride 102
3.28 MC_Halt 105

4 PLC library “McpPLCopenP1” – ISG extensions 107

4.1 Overview of the additional ISG function blocks (FB) 107
4.2 MCV_P1_Platform 110
4.3 MCV_Axis 111
4.4 MCV_CamSwitch 112
4.5 MCV_TorqueLimit 116
4.6 MCV_RmvPlug 118
4.7 MCV_SetFollowUp 120
4.8 MCV_SetDriveMode 121
4.9 MCV_SetTorqueValue 123
4.10 MCV_PosLagMonitor 124
4.11 MCV_Home 125
4.12 MCV_DisableFeedbackMon 127
4.13 MCV_ChgParamSet 128
4.14 MCV_ReqParamSetId 131
4.15 MCV_ChgParamValue 133
4.16 MCV_ActivateParamChg 135
4.17 MCV_SaveParamChg 138
4.18 MCV_ReadParamValue 140
4.19 MCV_DiscardParamChg 143
4.20 MCV_UnHome 145
4.21 MCV_DeactivateAxis 146

4.22 MCV_Decouple 147
4.23 MCV_Couple 149
4.24 MCV_ConveyorControl 151
4.25 MCV_SetReducedFilter 153
4.26 MCV_SuppressSoftLimSw 154
4.27 MCV_TorqueFreeStop 155
4.28 MCV_InitPosition 156

5 Application notes 158
5.1 Application note: Configuration of a “synchronisation master” 158
5.1.1 Configuring a simulated axis in the ISG-CNC 158
5.1.2 Setting axis parameters 158
5.1.3 Handling the control units on the HLI 158
5.1.4 GearIn coupling 159
5.1.5 Actual value filtering 159
5.1.6 CamIn coupling 159

6 References 160

7 Appendix 161
7.1 Suggestions, corrections and the latest documentation 161

Index 162

List of figures

Fig. 1: Extended PLCopen axis state model 9
Fig. 2: State diagram of the measuring channel 11
Fig. 3: Quasi mechanical coupling between the master and slave positions with cam types a and b 13
Fig. 4: Interpolation point entries and associated master position intervals 14
Fig. 5: Interpolation types: "StepDirect"," Linear" and" Polynom_3" 16
Fig. 6: Endlessly rotating vs. finite-rotating cam processing 17
Fig. 7: Use of the "MC_CamIn.StartMode" flag for an endlessly rotating slave axis 18
Fig. 8: Simplest possible linear cam for endlessly rotating processing 19
Fig. 9: LINE-POLY5 and POLY5-LINE sequences 20
Fig. 10: Master-slave intervals of a cam table 21
Fig. 11: Example of cam with POLY5-LINE 23
Fig. 12: Master/slave coupling without endless transition at the modulo limit 27
Fig. 13: Master/slave coupling with positive endless transition at the modulo limit 28
Fig. 14: Master/slave coupling with negative endless transition at the modulo limit 29
Fig. 15: Table with large transformation, endless negative 31
Fig. 16: Integration of data retention for cam tables into the overall system 33
Fig. 17: Impact of the compensation time on the switching times 114
Fig. 18: Behaviour of the "State" output when “FirstOnPosition” < “LastOnPosition” 114
Fig. 19: Behaviour of the "State" output when “FirstOnPosition” > “LastOnPosition” 115
Fig. 20: Flow chart - Decoupling an axis 148
Fig. 21: Flow chart - Coupling an axis 150

1 Definitions

1.1 Abbreviations

AXHLI	Axis-specific High-Level Interface
CM	Continuous Motion (endless rotation)
DM	Discrete Motion (positioning)
FB	Function Block
FBSD	FB State Diagram
HLI	High-Level Interface between MC and PLC
MC	Motion Controller
MCP	Motion Control Platform
MCE	Motion Control Engine
MC-FB	Motion Controller Function Block
NL Slope	Non-linear slope
PCS	Part program coordinate system
PLC	Programmable Logic Control
POE	Program Organisation Unit
SAI	Single Axis Interpolator

1.2 Explanations of terms

Axis group	A combination of axes which can execute a motion on a spatial curve coordinated by a channel while maintaining the specified values for velocity, acceleration and jerk on this spatial curve.
CoDeSys	PLC programming system from 3S Smart Software Solutions
Function block:	Internal order format of the ISG Motion Controller.
HLI library	Access to the memory interface to the ISG-MCE.
ISG-MCE	This stands for the ISG NC Kernel which, in connection with this documentation, is also referred to as the "Motion Control Engine"
Channel	Unit which coordinates the axis motions of an axis group.
MC-FB	Designates the PLC function blocks that are used to issue commands to the ISG-MC.
Multiprog	PLC programming system from KW-Software
Motion library	PLC software application that contains function blocks to move axes in conformity with the PLCopen specification as well as further FBs to assume motion generation tasks

Mandatory note on references to other documents

For the sake of clarity, links to other documents and parameters are abbreviated, e.g. [PROG] for the Programming Manual or P-AXIS-00001 for an axis parameter.

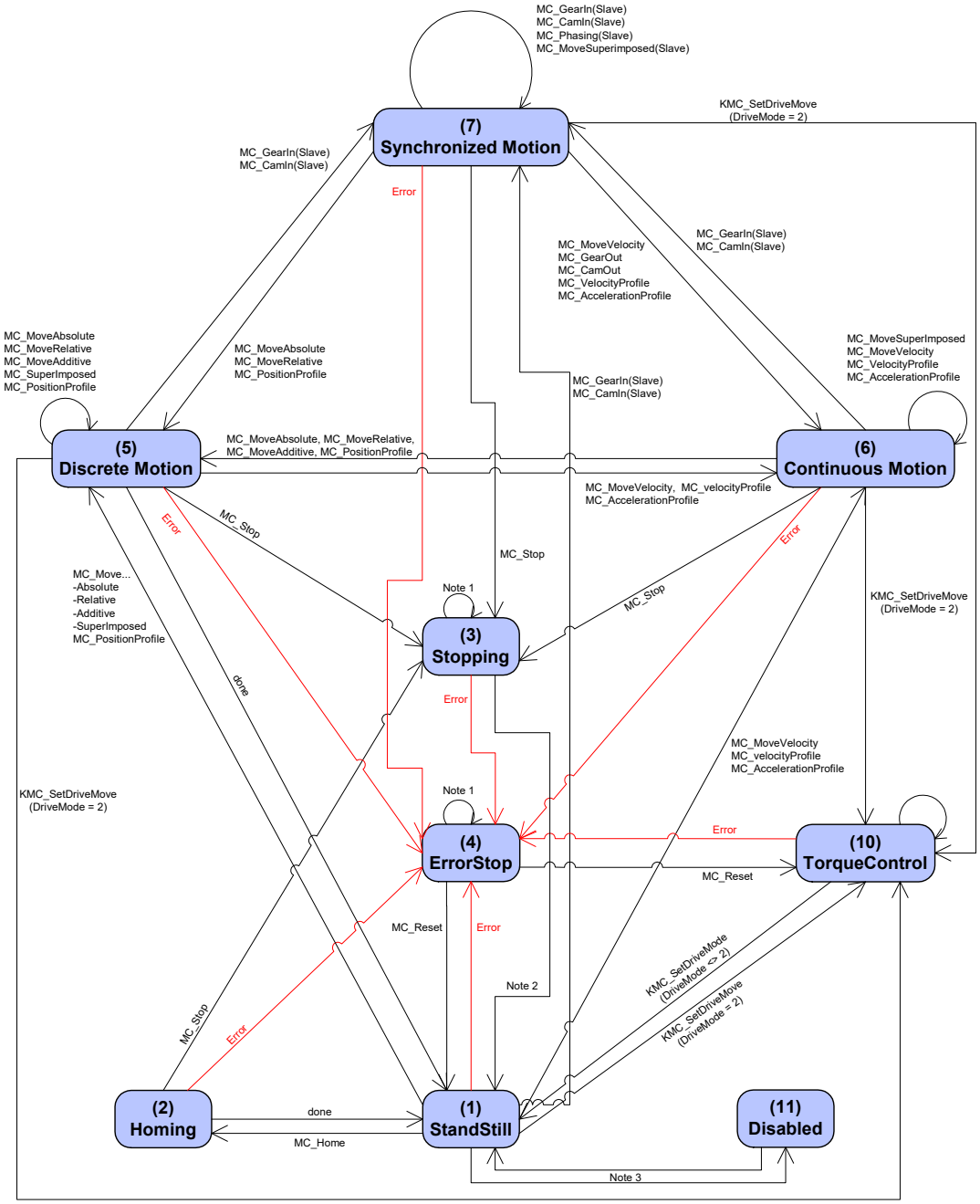
For technical reasons, these links only function in the Online Help (HTML5, CHM) but not in pdf files since pdfs do not support cross-linking.

2 PLC library “McpPLCopenP1”

This library contains function blocks that are defined as default blocks in the PLCopen Part 1 specification.

2.1 PLCopen axis state model

In addition to the axis state model of PLCopen, there is a further state for the **torque control** mode. The **Disabled** state is not yet fully implemented. The following figure shows the possible transitions.



Note 1: In state (3) Stopping or (4) ErrorStop all FBs can be called, although they are not executed. MC_Reset or error occurs - they change to the state (1) StandStill and (4) ErrorStop respectively.

Note ***2:*** MC_Stop.done AND NOT MC_Stop.Execute

Note ***3:*** Currently only MC_DeactivateAxis can change to the state (11) Disabled or (1) StandStill

MC_Reset: MC_Reset checks the context of a commissioned reset command. Is operating mode TorqueControl active, state (4) ErrorStop will direct change to state (10) TorqueControl, although the operating mode cannot be changed during this period.

Fig. 1: Extended PLCopen axis state model

2.2 Measurement of axis positions

Detection of positions independently of the interpolation or position control clock is necessary to realise precise positioning dependent on a measured event. In a control system, this can only be realised by additional hardware for latching the positions. This additional hardware is generally in a system close to the drive, if not directly in the drive itself. The function blocks MC_TouchProbe [▶ 73] and MC_AbortTrigger [▶ 71] make it possible to use this functionality.

2.2.1 Data structure TRIGGER_REF

A structure variable of type TRIGGER_REF transfers the measurement parameters to an instance of the MC_TouchProbe [▶ 73] or MC_AbortTrigger [▶ 71]. The structure variable contains

- the measuring channel number (structure element “tp_channel”)
- the mode (structure element “tp_mode”).
  The mode specifies whether the measurement is triggered on the rising or falling edge of the trigger signal. The following values are permissible for the mode:

Falling edge of the trigger signal	**1**
Rising edge of the trigger signal	**2**

Notice

The data structure TRIGGER_REF is already declared in the library supplied.

2.2.2 Behaviour of MC_TouchProbe and MC_AbortTrigger

- As opposed to motion commands, retriggering an MC_TouchProbe [▶ 73] or MC_AbortTrigger [▶ 71] instance does not result in the previous command being overwritten; instead, the new command is rejected by the FBs and no new command is sent. The retriggered FB instance indicates an FB-specific error and remains ACTIVE.
- At the FB outputs "error" and "error_id", a signal is output for one PLC cycle with which retriggering can be detected. If either AXIS_REF or TRIGGER_REF has changed at the time of retriggering, this is indicated as an error at the aforementioned FB outputs and the command is also rejected.
- The CommandAborted output can only be set if the command has been explicitly aborted by an MC_AbortTrigger [▶ 71]. The identical measuring channel number **and** mode must then be used in TRIGGER_REF since a measuring command is exclusively identified by the matching of all parameters in TRIGGER_REF.
- Contrary to the motion commands, a measurement command does not lead to a change in the axis status. As PLCopen does not specify in what states MC_TouchProbe [▶ 73] is permissible, it is defined that this is permissible in all states except "ERROR_STOP" and "HOMING".
- A separate state diagram is implemented for a measuring channel. The current state of the measuring channel is managed in “tp_state” in the AXIS_REF of the axis which was used to send the measuring command.

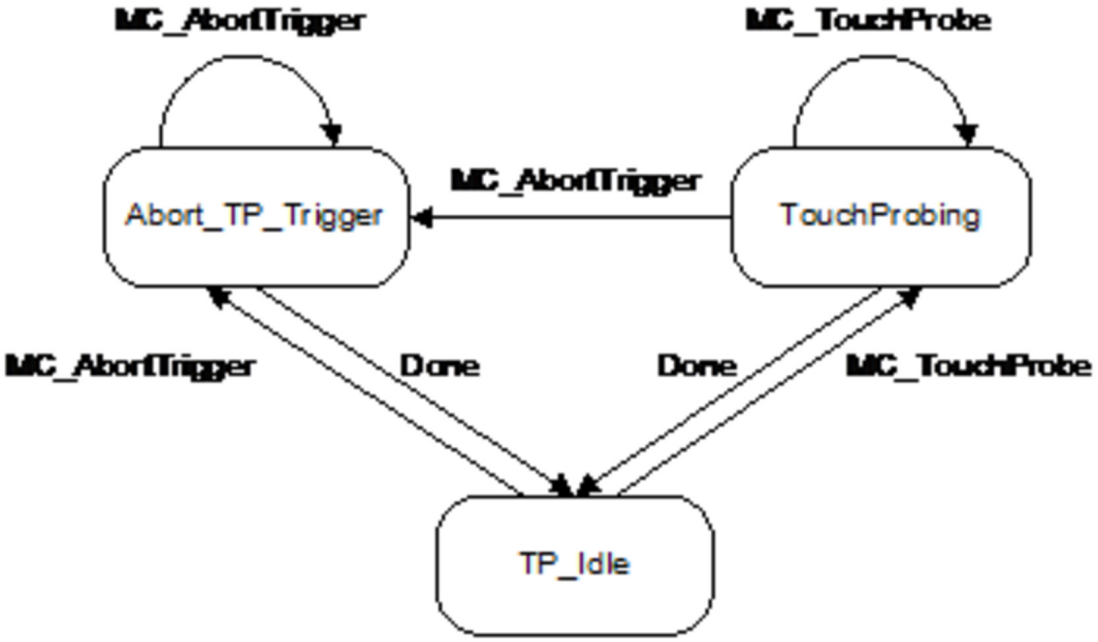


Fig. 2: State diagram of the measuring channel

- When, a maximum of as many measurement commands are possible as there are axes.

2.2.3 Measurement parameter definition in the axis-specific machine data

If positions of an axis are to be detected by measurement processes, this option must be defined by parameters in the axis-specific machine data.

On the one hand, the possibility of detecting measured values for an axis must fundamentally be enabled. This is done by means of the entry:

```
kenngr.messachse                          1
```

On the other hand, you must define where the source for the trigger signal that triggers measurement is. There are 4 different signal sources. In this context, it must be pointed out that only one of the signal sources can be defined at any one time as active and must therefore be assigned the value **1**. All other alternatives must be assigned the value 0.

The following example shows the parameter definitions that must be used when using KUKA drives. With KUKA drives, the trigger signal is supplied by the drives and so this signal source is defined as active and the others as inactive.

```
kenngr.mess_signal_achs_steuer            0
kenngr.mess_signal_taster                 0
kenngr.mess_signal_sercos                 0
kenngr.meas_signal_drive                  1
```

2.3 Cam functionality

Notice

Definition of curve

A cam is a description of a motion profile that clearly assigns a slave position to each master position for a specific master position range.

The term “**cam**“ is also used with the same meaning in this document.

2.3.1 Definition range of cams and their processing

The cam functionality in the ISG-MCE is looked upon as a mechanical analogy to a cam on a vertical shaft. Cams can only be used as an absolute coupling specification between two endlessly rotating axes. Accordingly, each cam's definition range extends over the entire modulo range of the master axis. Thus, there is always precisely one associated slave position for each absolute master position, i.e. the definition range is self-contained.

In this connection, we also speak of cam processing with **"**closed scanning" of the cam profile with the master positions.

2.3.2 Cam processing with closed scanning

In the case of cam processing with closed scanning, the cam profile is processed exclusively within the closed value range (= modulo range or "period") of the master axis.

It is assumed that this modulo range of the master axis is a property of the axis, i.e.: the axis keeps its position **automatically** within the modulo range.

Thus, there is always precisely one associated slave position for each master position. Therefore, the cam can never be scanned at an invalid position.

Only the following cam types are suitable for cam processing with closed scanning:

- Cams (a) whose velocity integral in the definition interval is exactly equal to zero, e.g. for limited oscillating motion.
- Cams (b) whose velocity integral in the definition interval corresponds to the modulo range of the slave axis, e.g. for endlessly rotating cranks.

Processing of any cam by means of closed scanning that does not correspond to the definition above leads to abrupt position changes of the slave axis whenever the master position exceeds a modulo limit.

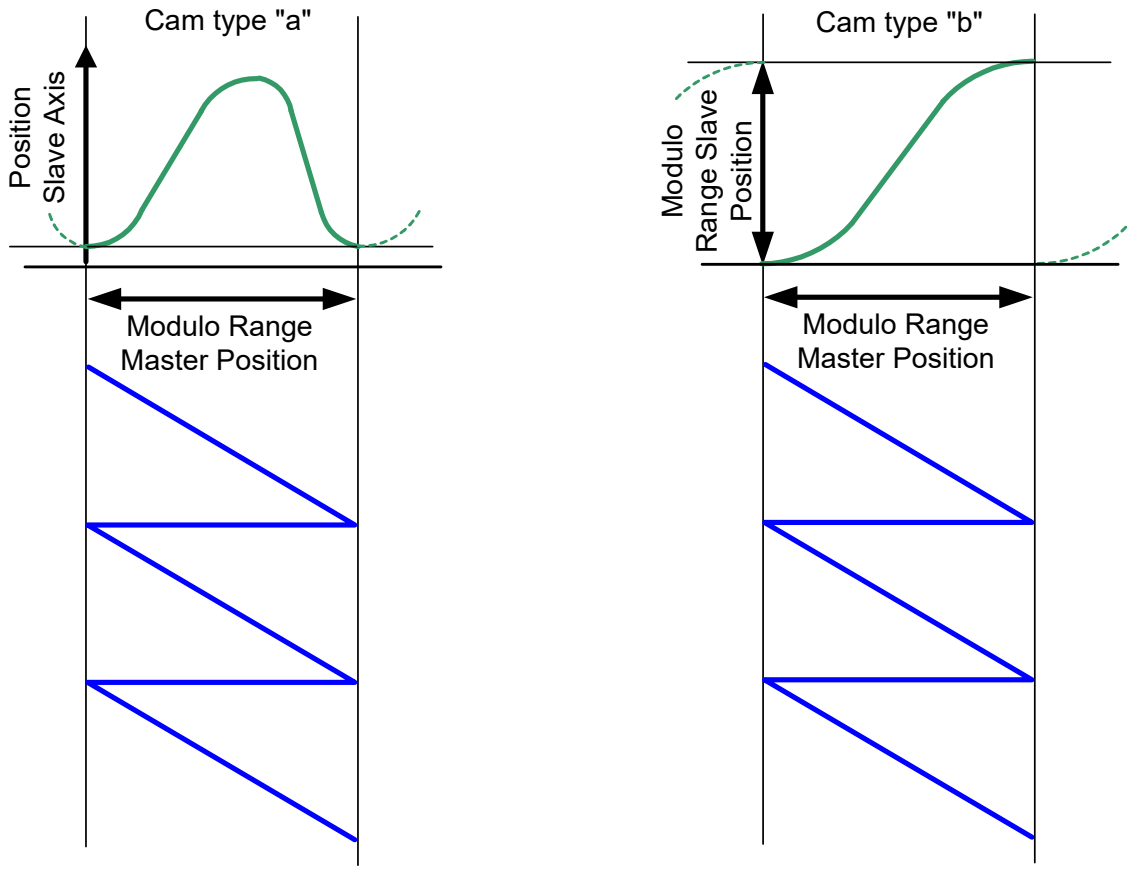


Fig. 3: Quasi mechanical coupling between the master and slave positions with cam types a and b

When scanning the cam type "a", nothing needs to be observed when the master passes through the modulo limit.

When the cam type “b” is scanned and when the master passes through the modulo limit, make sure that the slave position is corrected by a modulo range (see Section Closed processing for an endlessly rotating slave axis [▶ 17]).

2.3.3 Cam description by interpolation point tables

In the ISG-MCE, interpolation point tables and tables with alternating LINE/POLY5 motion sections are supported as cams.

The entries (= lines in the cam table) in the interpolation point tables are represented by equidistant interpolation points of the slave axis across the modulo range of the master axis. This is why no master positions are kept in this table type. When processing the cams, the modulo range of the master axis is split into sections of equal size in “number of interpolation points”. The interpolation point entries for the slave axis are represented by metric positions of the salve axis within the associated position intervals of the master axis.

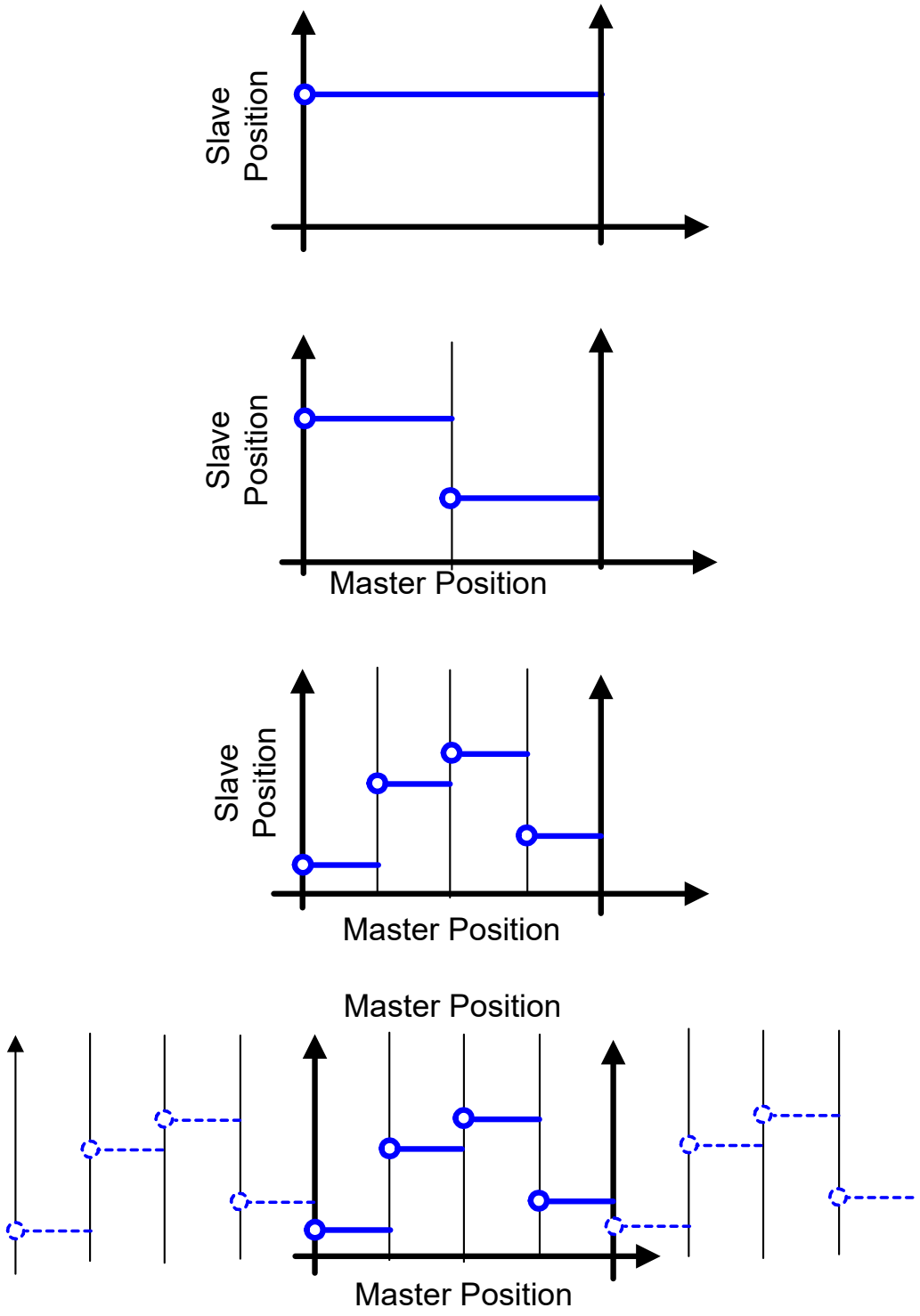


Fig. 4: Interpolation point entries and associated master position intervals

This form of cam description is only suitable for use with closed scanning because, according to the definition, the slave position at the end of the cam is implicitly assumed as being equal to the slave position at the start of the cam.

The first table above with precisely one slave position entry is therefore the simplest conceivable cam description in the form of a table. During processing, the slave would move to the entered slave position during the synchronisation phase (ramp-in) and would stay there. The interpolation of equidistant interpolation point tables is described in Sections Online interpolation of interpolation point tables [▶ 15] and Closed processing for an endlessly rotating slave axis [▶ 17].

Table entries need not be equidistant in tables containing LINE/POLY5 motion sections. For this reason, the first column always specifies the master position and the second column always specifies the slave position which defines the end of the motion section. The metric unit for master and slave positions is 1E-4° or 1E-4 mm. The Section Processing tables with LINE/POLY5 motion segments [▶ 20] describes the interpolation of tables with LINE/POLY5 motion sections.

2.3.4 Handling interpolation pointer tables

2.3.4.1 Online interpolation of interpolation point tables

As shown above, the current table description defines the slave position at the end of the cam as being = to the slave position at the start of the cam and is therefore only suitable for use with closed scanning.

If we go one step further, and want to interpolate between the interpolation points, the slave positions from the "neighbouring" modulo ranges are also used to compute information such as the pitch of the interpolated cam at a specific point.

Therefore, such an interpolation uses information that is not contained in the table as such, but in the definition of closed scanning.

Under the boundary conditions just pointed out, interpolation between the equidistant slave interpolation points is possible in linear fashion, cubically (only makes sense with very fine tables) or not at all.

The manufacturer-specific Input "MC_CamIn.TableFctType" to the PLCopen FB specifies how interpolation takes place between two table interpolation points:

```
HLI_STEP_DIRECT     : UDINT := 0;
```

Within the master position interval s_m_i, s_m_{i+1} the slave position entry s_s_i is output.

```
HLI_LINEAR          : UDINT := 1;
```

Within the master position interval s_m_i, s_m_{i+1} a linear interpolation is placed between the slave position entries s_s_i, s_s_{i+1}.

```
HLI_POLYNOM_3       : UDINT := 2;
```

Within the master position interval s_m_i, s_m_{i+1} a constant tangent cubic Bezier spline is interpolated between the four adjacent slave position entries s_s_{i-1}, s_s_i, s_s_{i+1}, s_s_{i+2}. Polynomial coefficient calculation takes place online.

These interpolation types are shown in the following figure. Obtaining of tangent information at each interpolation point and also at the modulo limits is also hinted at in the bottom figure: at every interpolation point, the tangent is a parallel line to the straight line that passes through the two neighbouring interpolation points.

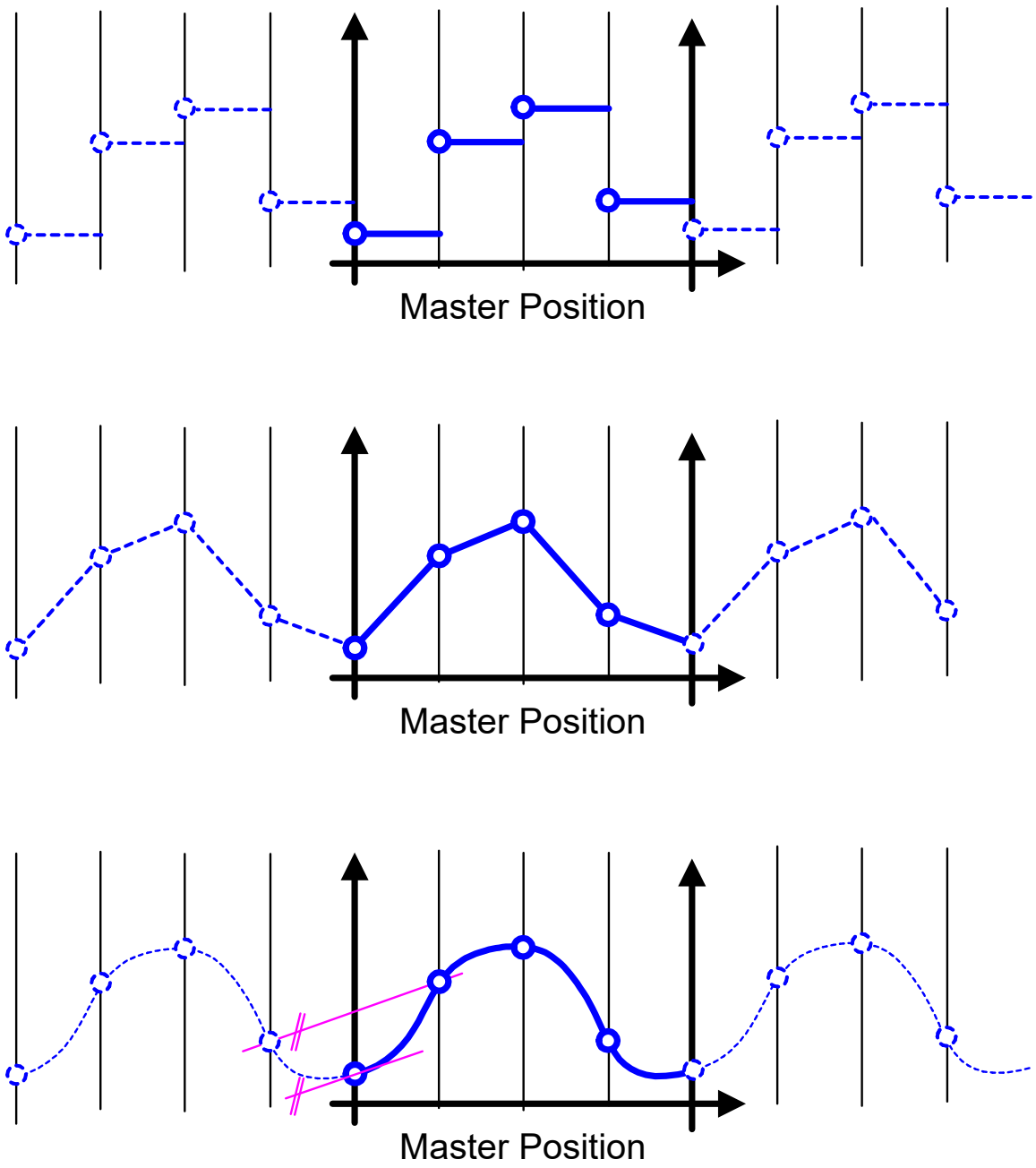


Fig. 5: Interpolation types: "StepDirect"," Linear" and" Polynom_3"

2.3.4.2 Closed processing for an endlessly rotating slave axis

In the case of interpolation between the interpolation points, there is a need to distinguish whether the slave is to process the cam endlessly or with finite rotation. This information is not included in the case of the current cam description.

As one and the same cam description can be used for both types, this distinction is not a characteristic of the cam, but is rather a component of the (CamIn) command.

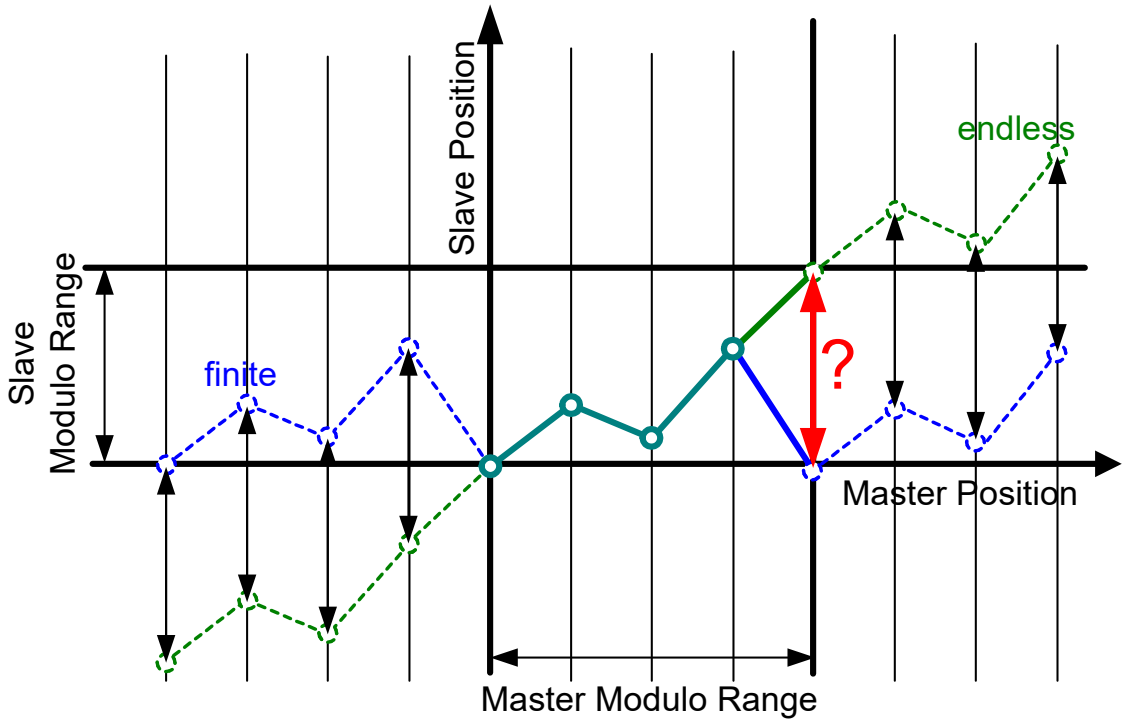


Fig. 6: Endlessly rotating vs. finite-rotating cam processing

In the case of endlessly rotating processing, the table interpolation points that are used to calculate the interpolation must be corrected by one modulo range at the cam's definition limits. The correction direction depends on whether the cam has been modelled for a forwards or backwards rotating slave.

However, the ISG slope kernel for cam processing (sl_cam_kernel) can only handle this properly and automatically if no intermediate interpolation (“MC_CamIn.TableFctType“ = HLI_STEP_DIRECT) is used between the interpolation points in the table. This ensures that the shortest modulo distance for two successive scanned slave positions is always output as the relative path for each real-time cycle. The modulo range of the slave axis is used for this calculation and this is why cam processing exclusively with modulo rotating slave axes is also possible.

If the cam has been modelled for an endlessly rotating slave axis, in connection with the use of interpolation point interpolation("MC_CamIn.TableFctType" = HLI_LINEAR or HLI_POLYNOM_3), it is necessary to specify in which direction modulo correction is to take place.

If the cam was modelled so that it passes through the modulo range of the slave in the positive direction, the bit represented by the global constant must be set in

```
HLI_CI_ENDLESS_POSITIVE    : UDINT := 32;
```

"MC_CamIn.StartMode”.

If the cam passes through the modulo range of the slave in the negative direction, the bit represented by the global constant must be set in “MC_CamIn.StartMode“.

```
HLI_CI_ENDLESS_NEGATIVE    : UDINT := 64;
```

"MC_CamIn.StartMode”.

Nevertheless, the cam profile must always match the chosen processing mode as otherwise this results in abrupt changes in the slave position.

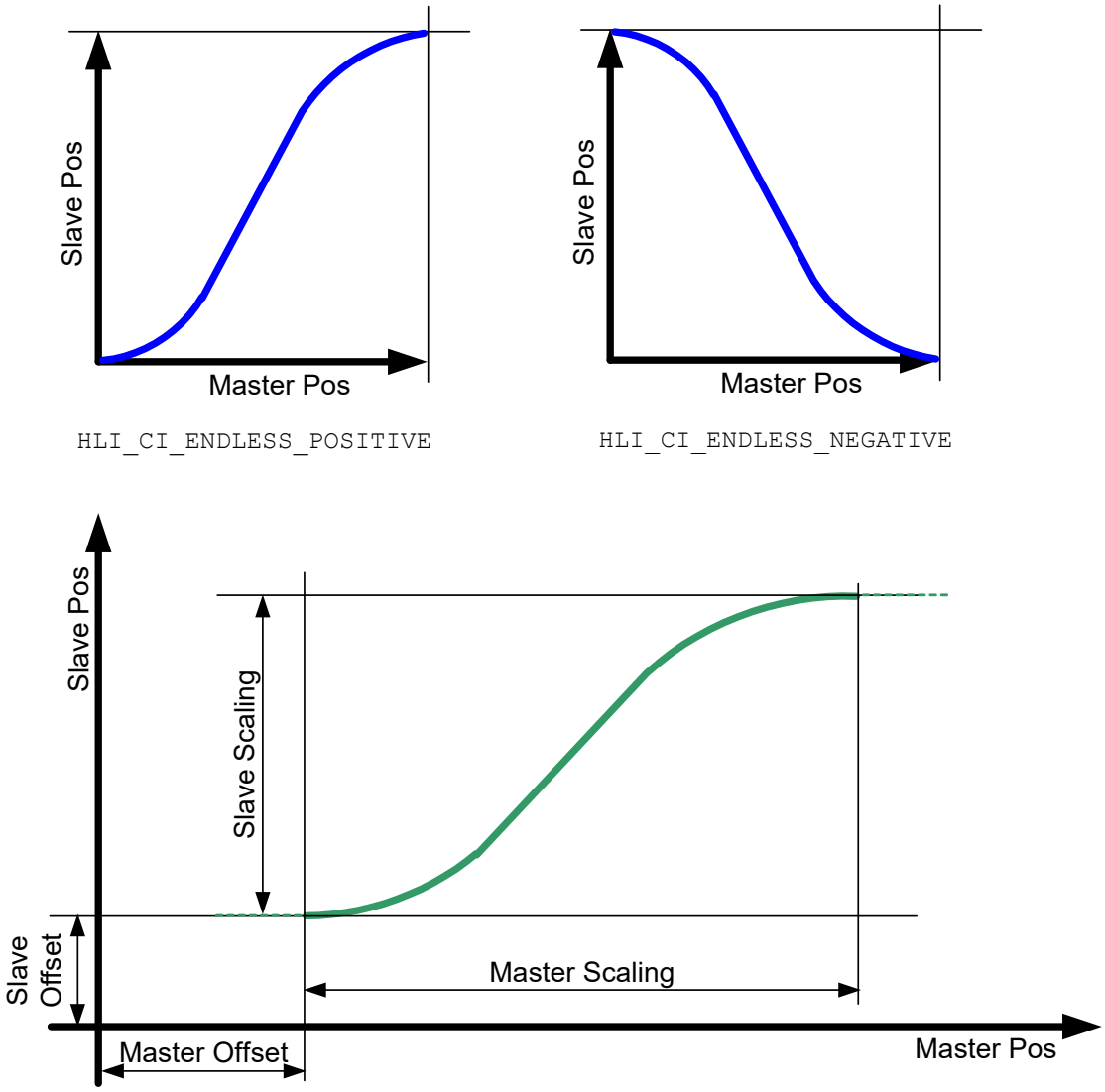


Fig. 7: Use of the "MC_CamIn.StartMode" flag for an endlessly rotating slave axis

2.3.4.2.1 Example: Simplest possible linear cam for endlessly rotating processing

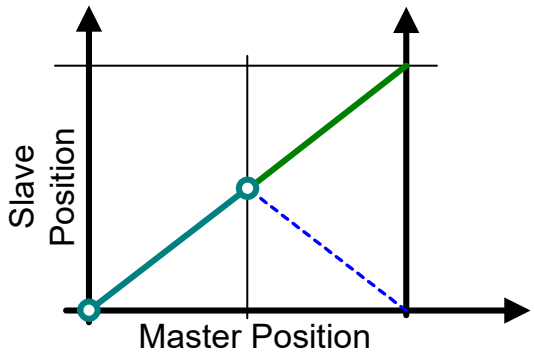


Fig. 8: Simplest possible linear cam for endlessly rotating processing

The figure above with two slave position entries is the simplest conceivable cam description of a straight line in the form of a table.

Peculiarities:

The cam must be used endlessly in the positive direction. Otherwise, the dashed curve is run through.

With closed scanning, the resulting straight line must pass exactly through the modulo range of the slave, in other words, it must have slope 1, otherwise a position jump will take place at the modulo limit. This means that a GearIn with a 1:1 ratio can be reproduced with a straight line of slope 1 as curve.

If you want to duplicate a GearIn with a straight line that has a pitch of 1, this is not possible with closed scanning!

2.3.5 Processing tables with LINE/POLY5 motion segments

2.3.5.1 Definition of the motion pattern

By analogy to programming a contouring motion via DIN 66025, the interpolation type is used to define along which trajectory the specified slave position is travelled to in the direction of ascending master positions.

Accordingly, the POLY5-LINE interpolation type defines that the first table point is travelled to via a polynomial and thus the first interval in the table is covered with constant speed.

In the case of the LINE-POLY5 type, the first slave position is accordingly travelled to at constant speed and there is a polynomial in the first interval. The figure below shows which patterns may result from LINE-POLY5 and POLY5-LINE sequences due to the position of the first interpolation point in the table.

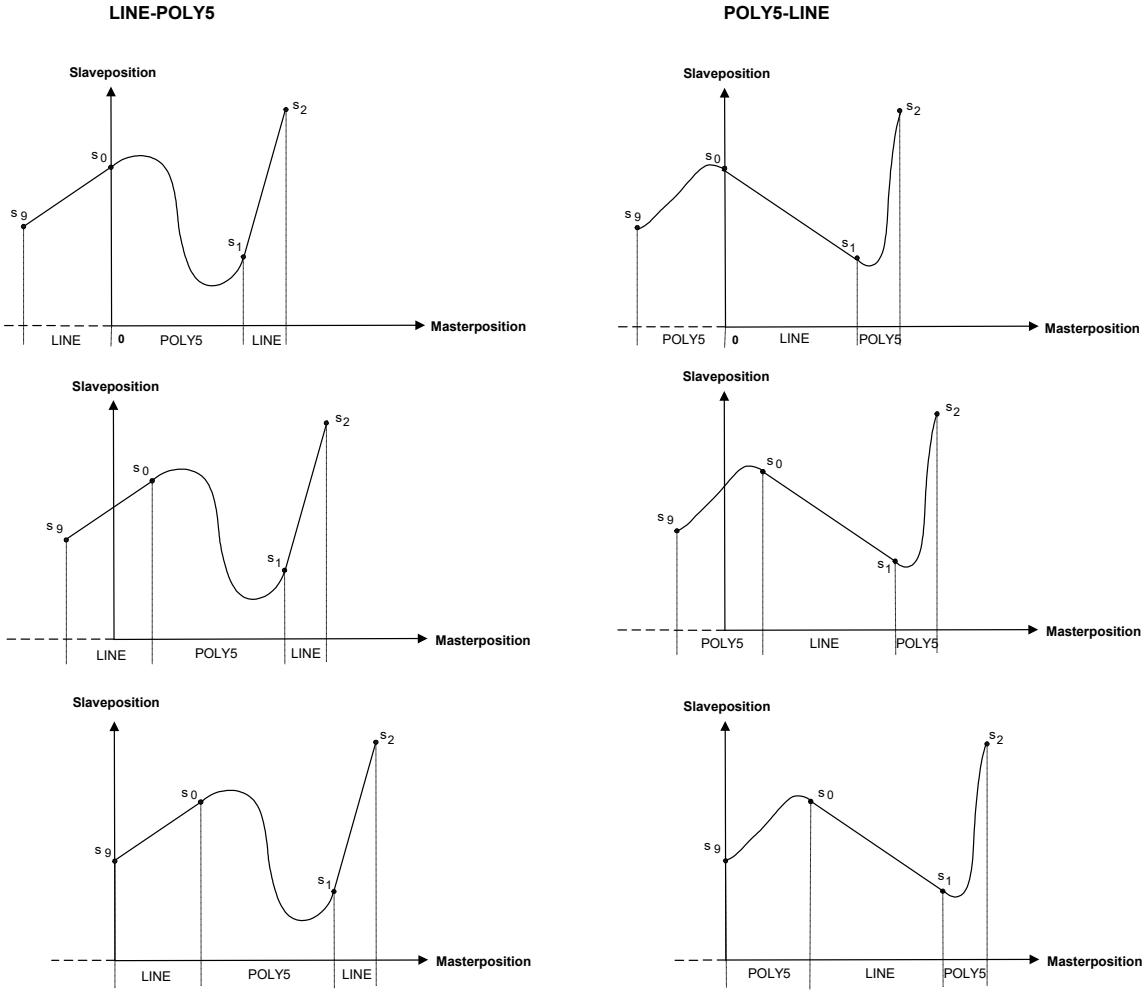


Fig. 9: LINE-POLY5 and POLY5-LINE sequences

2.3.5.2 Calculating the polynomial coefficients and interpolation

The polynomials of the fifth order for determination of the slave position s as a function of the master position t in the ranges of non-constant speed are of the form:

$$s(t) = k_5 t^5 + k_4 t^4 + k_3 t^3 + k_2 t^2 + k_1 t + k_0$$

They are defined by six boundary conditions. At the start and end points, the position s and the first (speed v) and second derivations (acceleration a) must agree with the neighbouring intervals. The second derivation, i.e. the acceleration, should be equal to 0 on the left and right.

The figure below shows 3 successive intervals which are used to calculate the polynomial coefficients:

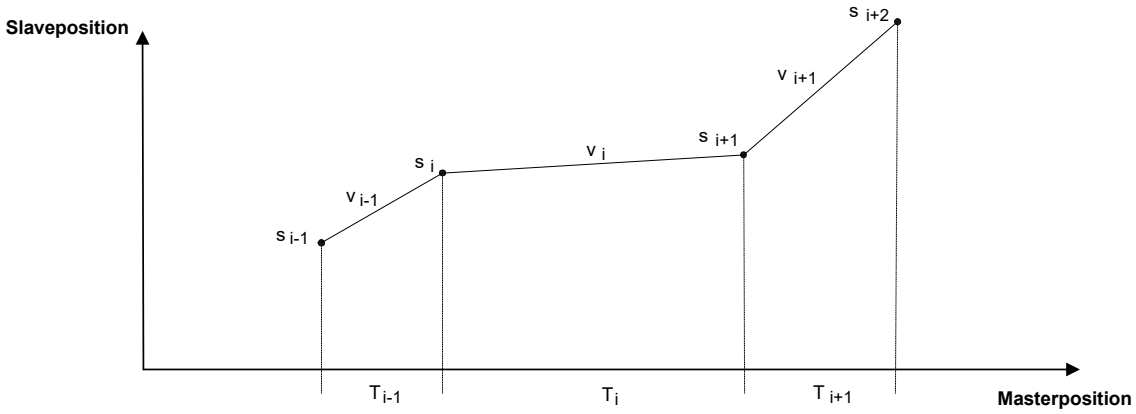
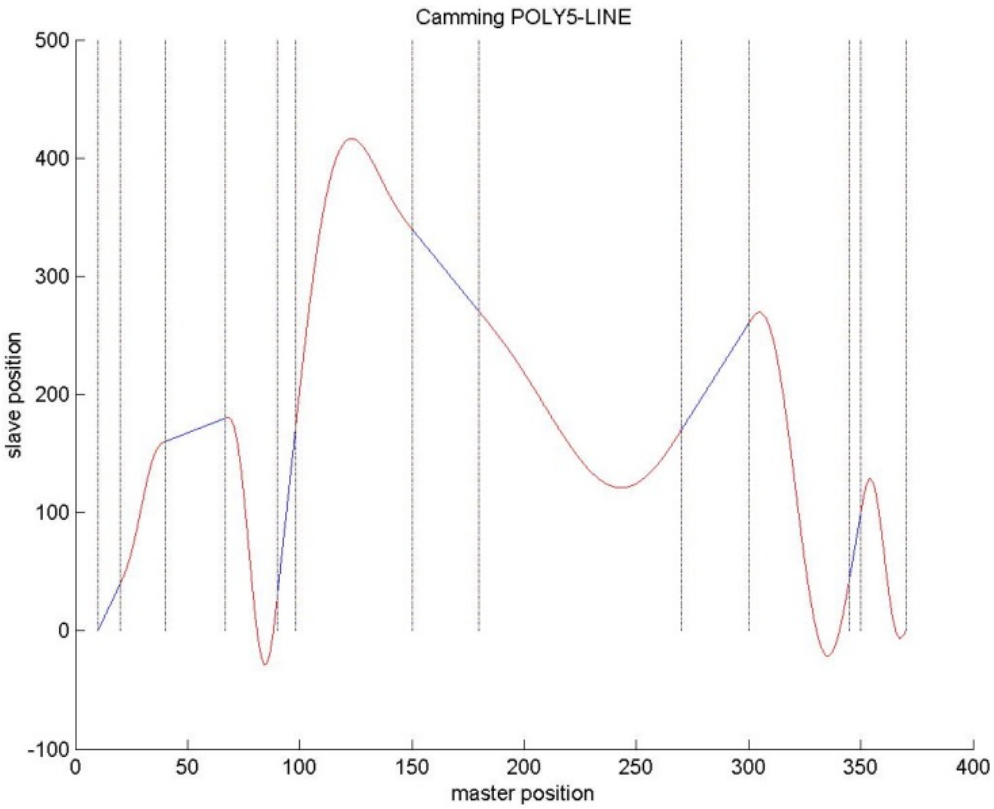


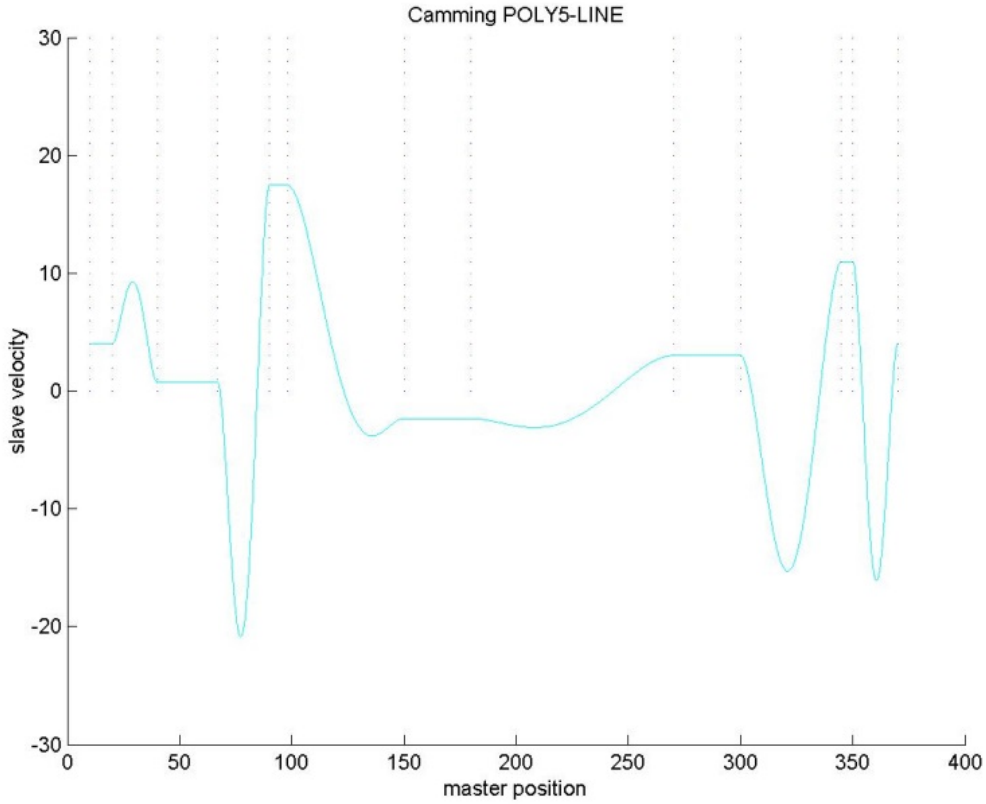
Fig. 10: Master-slave intervals of a cam table

Below, a camming table of the POLY5-LINE type is calculated as an example and is presented with the position, speed and acceleration in the individual intervals.

```
Tablename = poly5line_2  # Table name
Table-ID  = 402  # Table-ID
Tabletype = 7  # Table type
Function type = 5  # Interpolation type, 5 = POLY5-LINE
Lines = 10 # Number of lines
Begintable  # Beginning of table
100000 0
200000 400000
400000 1600000
670000 1800000
900000 300000
980000 1700000
1500000 3400000
1800000 2700000
2700000 1700000
3000000 2600000
EndTable
```

In the first diagram, the intervals with polynomials are shown in red and the intervals with constant speed are shown in blue:





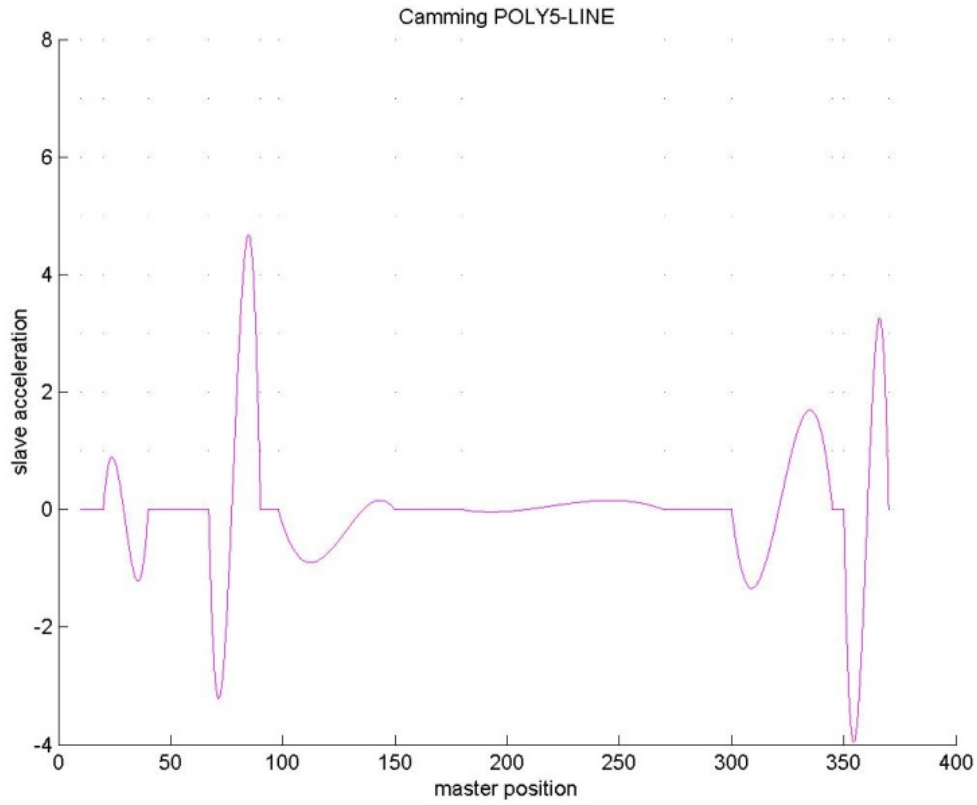


Fig. 11: Example of cam with POLY5-LINE

2.3.5.3 Table format

The table format used is the so-called zebra format, in which, as shown in the previous chapter, segments linked by straight lines (corresponding to constant speed) and segments linked with polynomials of the fifth order (Poly5 segment) alternate.

There must always be an even number of segments. It is also necessary to define whether the first segment begins with a straight line or a polynomial. The area between the last point of the table and its first point is referred to as the first segment.

The shape of the curve in the first segment is defined by the interpolation type. The LINE-POLY5 interpolation type defines that the first segment is travelled through with a straight line. In the case of the POLY5-LINE type, this segment contains a polynomial. Contrary to the interpolation point tables, in the case of these tables the LINE-POLY5 or POLY5-LINE interpolation type can be specified in the "Functiontype" token.

If this entry is to take effect, the PLC must transfer the "TableDefined" value in the "Function-Type" parameter of the PLCopen function block MC_CamIn. Otherwise, the interpolation type entered in MC_CamIn on commanding takes effect. Inconsistencies between the table entry and PLC commanding are thus avoided.

Notice

For the **LINE/POLY5 interpolation type**, the **value 4** must be specified in the table or in MC_CamIn. The **POLY5/LINE** type requires the **value 5**.

As already mentioned, for each line the useful data in the table contains master and slave value pairs, and the master position is specified in column 0 and the slave position in column 1, each in the unit 1E-4° or 1E-4 mm.

Notice

In the table's header, the **Table type in the "Tabletype" token must be set to the two-column integer format** by **entry of the value 7**.

2.3.5.4 Characteristics of the tables

The tables and table entries must have the following characteristics:

1. The master positions are sequentially ascending
2. The master positions are at a minimum distance of 0.1° from one another
3. The master and slave value pairs need not begin at the master position 0
4. The smallest table consists of two master and slave value pairs and thus of only two intervals
5. The maximum table size is limited to 360 master and slave value pairs
6. The master and slave axes must be a module axis.

Example

Layout of a typical cam table

```
Tablename = EndlessNegative  # Table name
Table-ID  = 422  # Table-ID
Tabletype = 7  # Table type
Function type = 5  # Interpolation type
Lines = 6 # Number of lines
Begintable  # Beginning of table
15000     0                        # MasterPos[0] SlavePos[0]
450000    3400000                  # MasterPos[1] SlavePos[1]
900000    2400000                  # MasterPos[2] SlavePos[2]
1350000   2000000                  # MasterPos[3] SlavePos[3]
2700000   1300000                  # MasterPos[4] SlavePos[4]
3400000   300000                   # MasterPos[5] SlavePos[5]
EndTable  # End of table
```

2.3.5.5 Behaviour of the slave axis at the modulo limit

The behaviour of the slave at the modulo limit must be specially defined.

By default, the slave will always move within its modulo limits, i.e. it maximally undergoes one revolution in the same direction.

If an endlessly rotating slave is required, either "EndlessPositive" or "EndlessNegative" must be specified in the start mode of the FB MC_CamIn. The slave will then endlessly rotate beyond the modulo limit in the positive or negative direction of rotation.

Cases for all three variants with regard to the behaviour of the slave axis at the modulo limit are shown in Figures 2, 3 and 4.

Example

The following camming table was used in all three examples:

```
Tablename = Smallest_Table # Table name
Table-ID  = 404  # Table-ID
Tabletype = 7  # Table type, 7= 2-column, SGN32 format
Functiontype = 4  # Interpol. type, 4= LINE-POLY5, 5= POLY5-LINE
Lines = 2 # Number of lines
Begintable  # Beginning of table
100000 0
1500000 3400000
EndTable
```

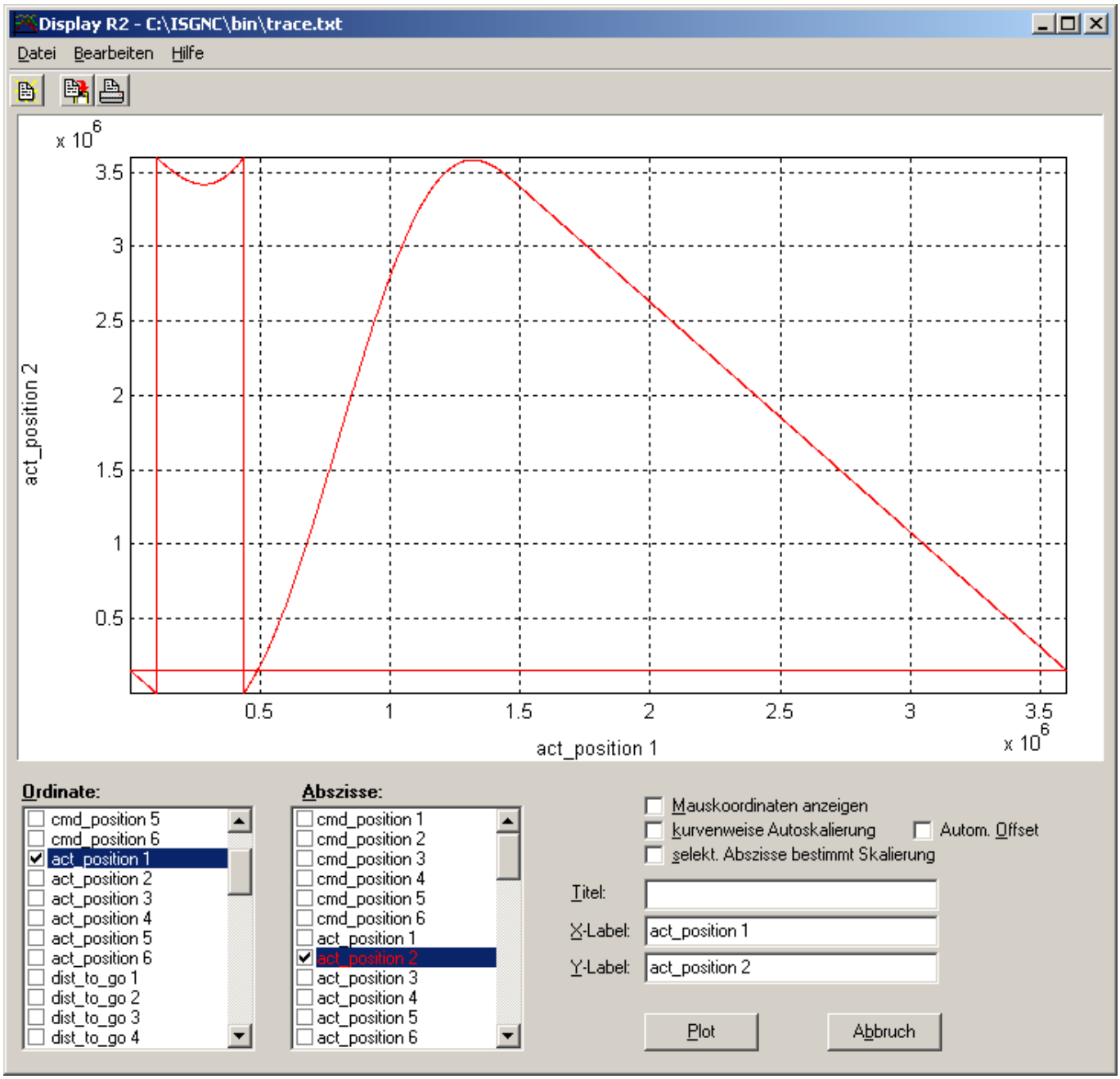


Fig. 12: Master/slave coupling without endless transition at the modulo limit

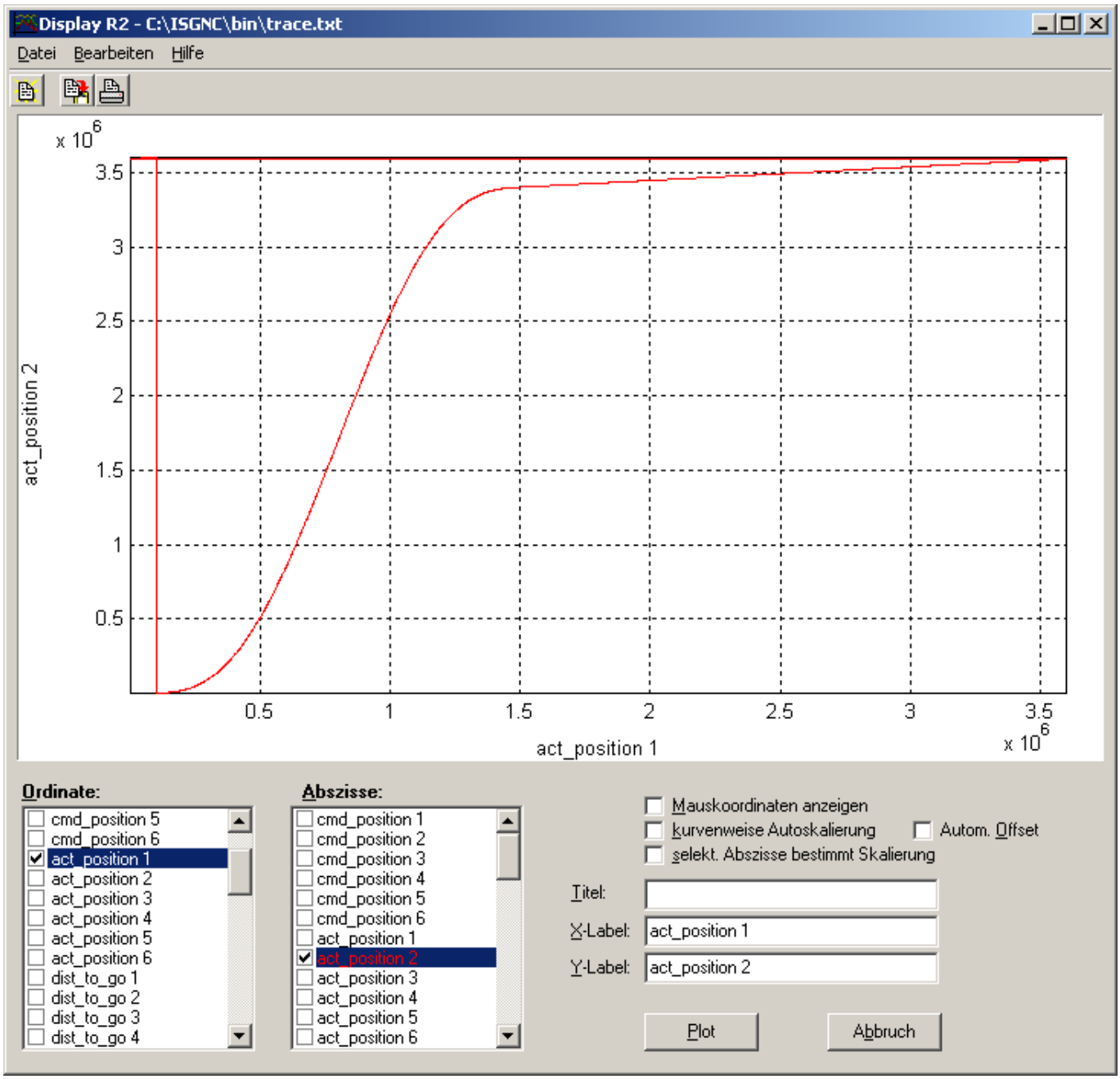


Fig. 13: Master/slave coupling with positive endless transition at the modulo limit

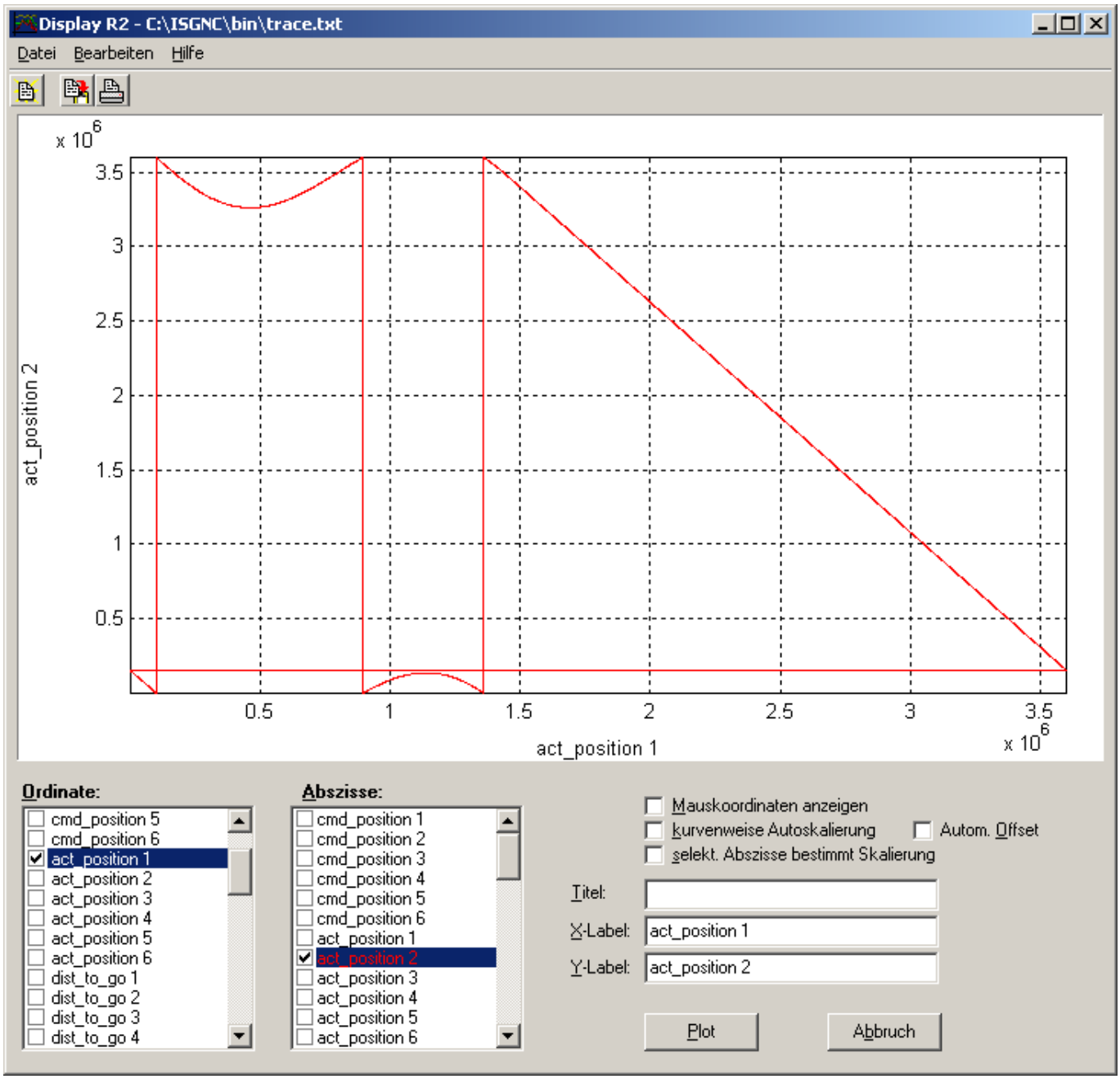


Fig. 14: Master/slave coupling with negative endless transition at the modulo limit

2.3.5.6 Slave positions outside the modulo range

If the gear ratio between the master and slave is to be very large, it may be necessary to define slave positions that lie outside the modulo range.

This is why slave positions outside the modulo range are not transformed into it, but are used without change as an interpolation point.

The following figure shows such a table and the profile thus run through.

Programing Example

Table with large transformation, endless negative

```
Tablename = extremetable4 # Table name
Table-ID  = 414  # Table-ID
Tabletype = 7  # Table type
Function type = 5  # Interpolation type
Lines = 8 # Number of lines
Begintable  # Beginning of table
0        0
200000 0
900000 -500000
1800000 -10000000
2100000 -16000000
2700000 -12000000
3200000 -19000000
3400000 -20000000
EndTable
```

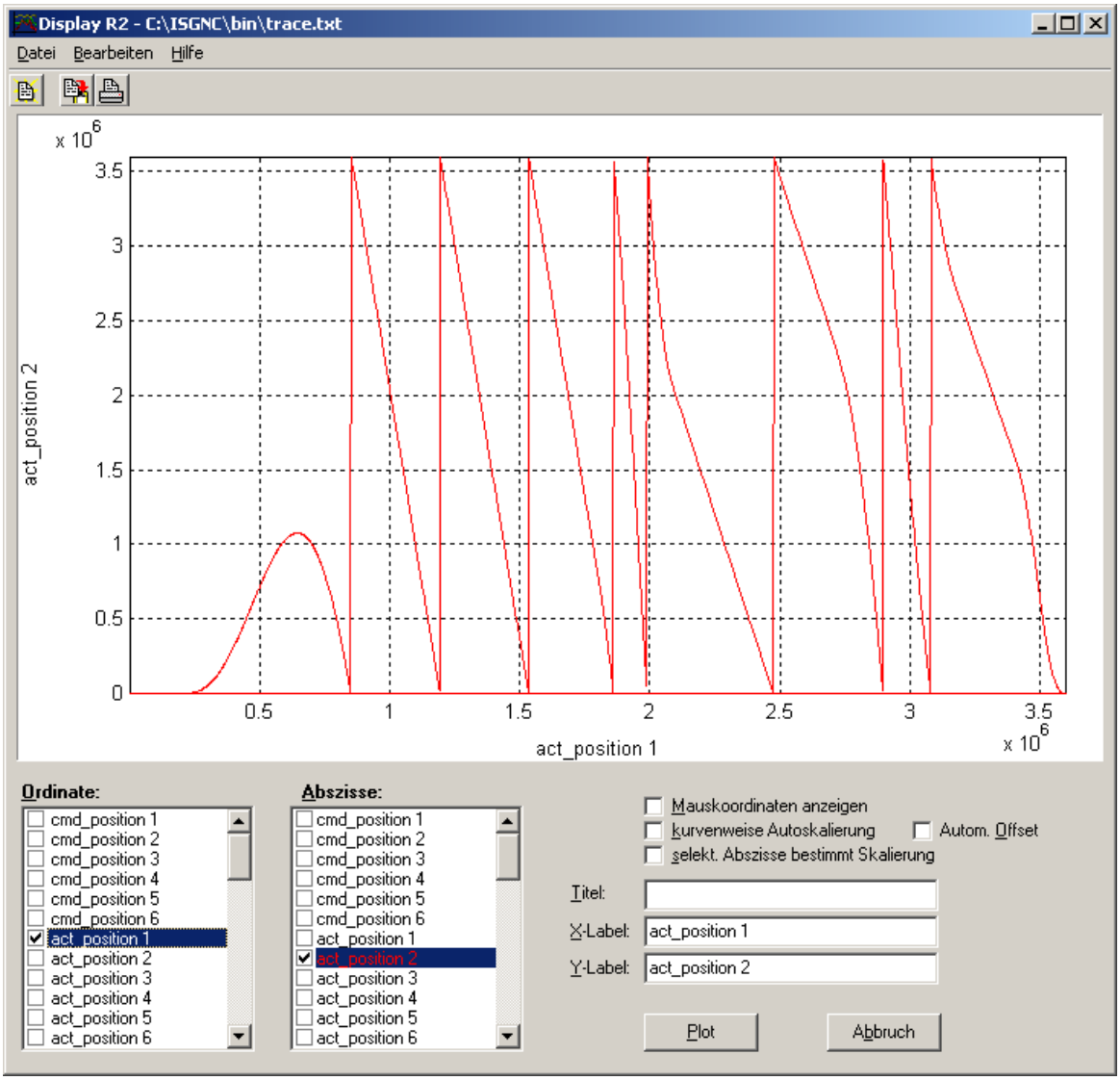


Fig. 15: Table with large transformation, endless negative

2.3.5.7 Modifying tables with motion segments

Online modification of tables with motion segments (zebra tables) is done directly in the table memory (shared memory) because the kernel only works with a copy of the active camming table. A maximum of 360 master and slave values pairs in the SGN32 format are defined for each table, the internal memory requirement is only 2880 bytes.

The modified table is activated by commanding MC_CamIn again. If possible, the table should be switched over in a safe area in which the slave is not moving (LINE with 0 speed). Despite standstill of the slave, synchronisation to a new table requires a phase of synchronisation to speed = 0 and balancing of the phase position. This operation can take a few cycles depending on the setting in the axis MDS. Presuming that the slave is not moving and does not have to perform phase compensation (characteristic of the safe range), synchronisation to the camming table will be completed in the time set in the axis MDS for dwelling within the speed window (usually for about 20 to 30 ms). If the slave is synchronised at standstill and the dwell time has already elapsed within the speed window, optimisation might have to take place here so as to ensure that the entire operation is completed in only a few cycles.

The range in which the slave is resynchronised should be of the LINE type. This ensures that the velocity over the entire interval is constant = 0. In the case of a polynomial, this will generally result in movement of the slave.

2.3.6 Access to cam tables

PLCopen requires provision of a structure with the name MC_CAM_REF, which must be passed on as an input parameter to the PLCopen FB MC_CamTableSelect. If the call is successful, the PLCopen FB MC_CamTableSelect returns a so-called MC_CAM_ID, which is also defined in PLCopen.

As defined by PLCopen, both MC_CAM_REF and MC_CAM_ID are manufacturer-specific. Both data types contain the identifier of the cam table. The MC_CAM_REF data structure also contains the name of the file created in the cam table.

Function blocks exist to edit tables for Multiprog running under KW-Software and CoDeSys from 3S. However, edited tables cannot be saved.

The "tab_ldr.lis" file is interpreted during startup. It contains the file names of the CAM tables to be loaded. All (*.tab) table files, which must also be in the list directory of the ISG-MCE, are interpreted and are loaded into the table memory.

The following figure elucidates this:

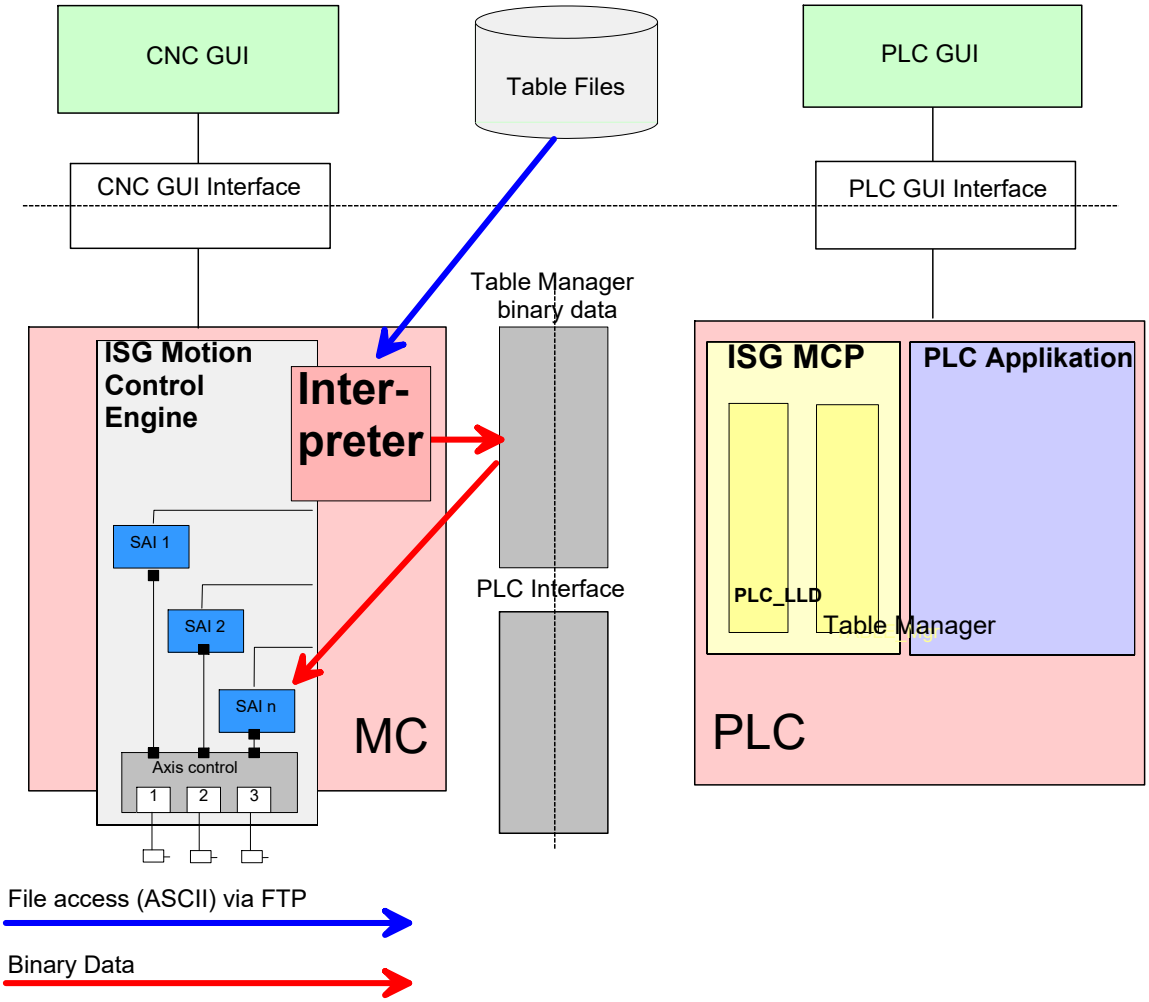


Fig. 16: Integration of data retention for cam tables into the overall system

2.3.7 File format of the cam tables (ASCII interpreter)

The cam tables are stored in ASCII format. One file is needed for each cam table. The lines of the cam table represent the interpolation points of the slave axis over the modulo range of the master axis, which is from 0° to 360°.

The ASCII interpreter is realised so that it internally works in two interpretation modes.

1. Interpretation of the table header
2. Interpretation of the table data

2.3.7.1 The table header

The table header must contain the following keywords to permit the correct interpretation of the table.

```
Tablename = Sine 360°
Tabletype       = 3          # 3 = TABLE_ONE_COLUMN_REAL64
Table-ID  = 402  # Table-ID
Functiontype = 5  # Interpolation type, 5 = POLY5-LINE
Lines           = 36
```

Example

Example of a table header

- **Tabellenname** (Tablename)**:**
  Any name without blanks.
- **Tabellentyp** (Tabletype):
  The table type to be used. This is specified in the index, see table in Tabellentypen [▶ 35].
- **Table-ID:**
  ID of the cam table.
- **Funktionstyp** (Functiontype)**:**
  Interpolation type, see Interpolation types [▶ 36].
- **Number of lines** (Lines)**:**
  Number of lines which contain the table values.

2.3.7.2 Table data

Detection of the key word **"**Begintable" switches the interpreter to the TSF interpretation mode (= Tabulator Separated Format). This means that after the keyword
“**Begintable**“ the interpreter expects columns of figures corresponding to the table type.

Example

Example of table data

```
Begintable
625133,4396
1231272,516
1800000
2314035,395
2757759,995
3382893,435
3545307,911
3600000
....
-2757759,995
-2314035,395
-1800000
-1231272,516
-625133,4396
-8,82107E-10
EndTable
```

2.3.7.3 Table types

The table below lists the available table types. The table type TABLE_MOTION_PATTERNS is only used for internal purposes (automatic test).

Index	Table type	Number of columns	Data format
1	TABLE_ONE_COLUMN_SGN16	1	Signed 16
2	TABLE_ONE_COLUMN_SGN32	1	Signed 32
3	TABLE_ONE_COLUMN_REAL64	1	Real 64
4	TABLE_4_COLUMNS_SGN32	4	Signed 32
5	TABLE_4_COLUMN_REAL64	4	Real 64
6	TABLE_MOTION_PATTERNS	-	-
7	TABLE_TWO_COLUMNS_SGN32	2	Signed 32

2.3.7.4 Function types (interpolation types)

Interpolation types for the cam tables:

Index	Function type	Description
0	STEP_DIRECT	Constant positions are read from the cam table.
1	LINEAR	The slave position between two table entries is interpolated linearly.
2	POLYNOM_3	Currently not implemented.
3	POLYNOM_3_KOEFF	Precalculated polynomial coefficients
4	LINE_POLY5	Linear – Poly5
5	POLY5_LINE	Poly5 – Linear
6	TABLE_DEFINED	Function type is defined in the cam table.

2.3.7.5 Tokens and separators

Line by line, the interpreter separates the ASCII file contents into tokens separated by separator characters. The following separator characters are permissible within the table data:

- Blank
- Tabulator

Within the table header, the

- equal sign

is also supported as a separator.

2.3.7.6 Comments

For comments, the **"#**" (hash) and ";" (semicolon) can be used as the first character in a line. If the comment characters are not the first character in the line, the entire line is interpreted quite normally, which leads to an error message. The interpreter knows the number of tokens to be taken per line and only takes exactly that number. This means that the interpreter ignores what is behind the last token of a line separated by a separator (blank or tab).

Programing Example

Examples of syntactically correct comments

```
Tablename  = CAM profile 0815 valid comment
Tabletype   = 5       # 5 = TABLE_4_COLUMN_REAL64
Table-ID  = 123  # Table-ID
Lines       = 36
#Begintable!!! With '#' Commented out!!!
0,174532925   0,173648178   0,984807753   2,342020143   valid comment
```

2.3.7.7 Example of a correct file format

Programing Example

Complete example of a correct file format

```
Tablename  = CAM profile 0815
Tabletype      = 5    # 5 = TABLE_4_COLUMN_REAL64
Table-ID  = 123 # Table-ID
Lines = 36
Begintable
0,174532925    0,173648178    0,984807753    2,342020143
0,3490658      0,342020143    0,939692621    2,64278761
0,523598776    0,5            0,866025404    2,866025404
0,698131701    0,64278761     0,766044443    2,984807753
0,872664626    0,766044443    0,64278761     2,984807753
1,047197551    0,866025404    0,5            2,866025404
1,221730476    0,939692621    0,342020143    2,64278761
1,396263402    0,984807753    0,173648178    2,342020143
1,570796327    1              6,12574E-17    2
1,745329252    0,984807753   -0,173648178    1,657979857
1,919862177    0,939692621   -0,342020143    1,35721239
2,094395102    0,866025404   -0,5            1,133974596
2,268928028    0,766044443   -0,64278761     1,015192247
2,443460953    0,64278761    -0,766044443    1,015192247
2,617993878    0,5           -0,866025404    1,133974596
2,792526803    0,342020143   -0,939692621    1,35721239
2,967059728    0,173648178   -0,984807753    1,657979857
3,141592654    1,22515E-16   -1              2
3,316125579   -0,173648178   -0,984807753    2,342020143
3,490658504   -0,342020143   -0,939692621    2,64278761
3,665191429   -0,5           -0,866025404    2,866025404
3,839724354   -0,64278761    -0,766044443    2,984807753
4,01425728    -0,766044443   -0,64278761     2,984807753
4,188790205   -0,866025404   -0,5            2,866025404
4,36332313    -0,939692621   -0,342020143    2,64278761
4,537856055   -0,984807753   -0,173648178    2,342020143
4,71238898    -1             -1,83772E-16    2
4,886921906   -0,984807753    0,173648178    1,657979857
5,061454831   -0,939692621    0,342020143    1,35721239
5,235987756   -0,866025404    0,5            1,133974596
5,410520681   -0,766044443    0,64278761     1,015192247
5,585053606   -0,64278761     0,766044443    1,015192247
5,759586532   -0,5            0,866025404    1,133974596
5,934119457   -0,342020143    0,939692621    1,35721239
6,108652382   -0,173648178    0,984807753    1,657979857
6,283185307   -2,4503E-16     1              2
Endtable
```

3 Overview of the PLCopen FBs

The table below provides an overview of the available PLCopen function blocks

Function block	PLC systems		
	3S	TwinCAT	KW
MC_Power [▶ 41]	SIMU - SERCOS	SIMU	SIMU KUKA SERCOS
MC_ReadAxisInfo [▶ 43]	SIMU - SERCOS	SIMU	SIMU KUKA SERCO)
MC_ReadMotionState [▶ 45]	SIMU - SERCOS	SIMU	SIMU KUKA SERCO)
MC_ReadStatus [▶ 48]	SIMU - SERCOS	SIMU	SIMU KUKA SERCO)
MC_ReadAxisError [▶ 50]	SIMU - SERCOS	SIMU	SIMU KUKA SERCO)
MC_ReadActualPosition [▶ 51]	SIMU - SERCOS	SIMU	SIMU KUKA SERCOS
MC_Reset [▶ 52]	SIMU - SERCOS	SIMU	SIMU KUKA SERCO)
MC_Home [▶ 53]	SIMU - SERCOS	SIMU	SIMU KUKA SERCO)
MC_MoveAbsolute [▶ 55]	SIMU - SERCOS	SIMU	SIMU KUKA SERCO)
MC_MoveAdditive [▶ 58]	SIMU - SERCOS	SIMU	SIMU KUKA SERCO)
MC_MoveRelative [▶ 61]	SIMU - SERCOS	SIMU	SIMU KUKA SERCOS

MC_MoveSuperImposed [▶ 63]	SIMU - SERCOS	SIMU	SIMU KUKA SERCO)
MC_MoveVelocity [▶ 66]	SIMU - SERCOS	SIMU	SIMU KUKA SERCO)
MC_Stop [▶ 69]	SIMU - SERCOS	SIMU	SIMU KUKA SERCO)
MC_AbortTrigger [▶ 71]	- - SERCOS	-	- KUKA SERCO)
MC_TouchProbe; [▶ 73]	- - SERCOS	-	- KUKA SERCOS
MC_CamIn [▶ 75]	SIMU - SERCOS	-	SIMU KUKA SERCO)
MC_CamOut [▶ 82]	SIMU - SERCOS	-	SIMU KUKA SERCO)
MC_CamTableSelect [▶ 83]	SIMU - SERCOS	SIMU	SIMU KUKA SERCO)
MC_GearIn [▶ 84]	SIMU - SERCOS	SIMU	SIMU KUKA SERCO)
MC_GearOut [▶ 89]	SIMU - SERCOS	SIMU	SIMU KUKA SERCOS
MC_Phasing [▶ 90]	SIMU - SERCOS	SIMU	SIMU KUKA SERCO)
MC_ReadParameter [▶ 93]	SIMU [1]) - SERCOS [1])	-	SIMU KUKA SERCO)
MC_WriteParameter [▶ 96]	SIMU [1]) - SERCOS [1])	-	SIMU KUKA SERCO)
MC_WriteLrealParameter [▶ 96]	SIMU SERCOS	-	-

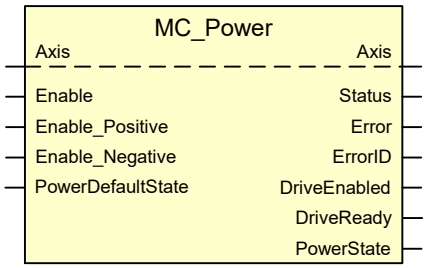
MC_SetPosition [▶ 101]	SIMU - SERCOS	SIMU	SIMU KUKA SERCO)
MC_SetOverride [▶ 101]	SIMU - SERCOS	SIMU	SIMU KUKA SERCO)
MC_Halt [▶ 102]	SIMU - SERCOS	SIMU	SIMU KUKA SERCOS

[1]) Availability is dependent on the MCE used.

3.1 MC_Power

The MC_Power FB switches the power of the drive of the axis defined by AXIS_REF on or off. It is assigned to the group of administrative FBs.

Block diagram



FB parameters

VAR_IN_OUT		
Variable name	**Data type**	**Description**
Axis	AXIS_REF	Axis reference

VAR_INPUT		
Variable name	**Data type**	**Description**
Enable	BOOL	Must have the value TRUE otherwise power cannot be switched on.
Enable_Positive	BOOL	Must have the value TRUE otherwise power cannot be switched on. **Enable only for positive motion direction is not supported.**
Enable_Negative	BOOL	Must have the value TRUE otherwise power cannot be switched on. **Enable only for negative motion direction is not supported.**
PowerDefaultState	MCV_DRIVE_POWER_STATE	**Function available for CANopen drives:** The function block sets the drive to this state as long as the “Status” output indicates FALSE, i.e. at least one of the Enable... inputs is assigned the value FALSE (see table below). The input is assigned **mcvPowerStateDefault** as default value

VAR_OUTPUT		
Variable name	**Data type**	**Description**
Status	BOOL	TRUE indicates that the drive has torque applied and is in control mode.
Error	BOOL	Indicates whether an error occurred in the FB.
ErrorID	WORD	Error identifier
DriveEnabled	BOOL	TRUE indicates that the drive is ready for enabling power.
DriveReady	BOOL	TRUE indicates that the drive is being controlled.
PowerState	MCV_DRIVE_ POWER_STATE	State in which the drive is currently (see table below).

Values of the enumeration type MCV_DRIVE_POWER_STATE

Constant	Value	Note
mcvPowerStateNotReady	0	Drive not ready to run.
mcvPowerStateReadyForPower	1	Drive ready to run.
mcvPowerStateReadyForOperation	2	Drive switched on.
mcvPowerStateDriveEnabled	3	Drive ready but no controller enable.
mcvPowerStateOperational	4	Drive ready and command values to follow.
mcvPowerStateDefault	5	The "PowerDefaultState” input is defaulted with this value. Then MC_Power only sends commands to the drive when TRUE was assigned to all 3 "Enable_...” inputs. If FALSE is assigned to one of the 3 "Enable_...” inputs, this value causes the drive to revert to the initial state configured in the drive.

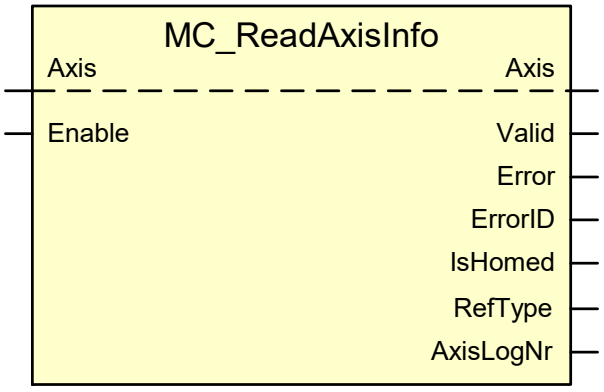
Behaviour of the FB:

- The power for the drive is connected only if all 3 inputs have the value TRUE.
- Regardless of the status of the axis, this FB can be commanded and the command can be sent to the motion controller.
- If there is an error in other axes, the MC_Power FB reacts and outputs the error P-ERR-40001 (ERR_PLC_AX_MC) at the function block and indicates the error P-ERR-294039 at its output AxisErrorID on the function block MC_ReadAxisError.
- The behaviour of the axis depends on the defined drive type. If the drive type simulation (P-AXIS-00020) is defined for the axis, the output "Status" is also simulated.

3.2 MC_ReadAxisInfo

This FB is used to obtain the current status data of an axis. It is assigned to the group of administrative FBs.

Block diagram



FB parameters

VAR_IN_OUT		
Variable name	**Data type**	**Description**
Axis	AXIS_REF	Axis reference

VAR_INPUT		
Variable name	**Data type**	**Description**
Enable	BOOL	If the value is TRUE, the axis state data is indicated continuously.

VAR_OUTPUT		
Variable name	**Data type**	**Description**
Valid	BOOL	TRUE if valid values are available.
Error	BOOL	Indicates whether an error has occurred in an FB.
ErrorID	WORD	Error identifier
IsHomed	BOOL	Homing is completed.
RefType	DINT	Reference type (single axis, path axis, spindle, etc.)
AxisLogNr	UINT	Logical axis number

Values of RefType output

Constant	Value	Note
REF_NONE	0	No reference available
REF_ANY_AXIS	1	Reference available but not yet assigned
REF_AXGRP_AXIS	2	Axis is in an axis group
REF_SAI_AXIS	3	Single axis / SAI
REF_IO_STATION	4	I/O station
REF_TO_REF	5	Reference to an axis reference
REF_ENC_AXIS	6	Encoder axis

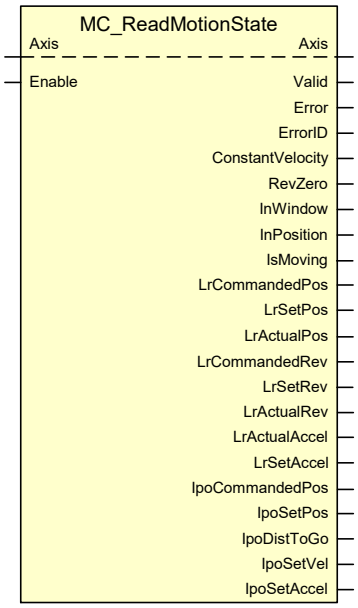
Behaviour of the FB:

- The information of an axis is updated cyclically as long as “Enable” has the value TRUE.

3.3 MC_ReadMotionState

This FB is used to obtain the current motions states of an axis. It is assigned to the group of administrative FBs.

Block diagram



FB parameters

VAR_IN_OUT		
Variable name	**Data type**	**Description**
Axis	AXIS_REF	Axis reference

VAR_INPUT		
Variable name	**Data type**	**Description**
Enable	BOOL	If the value is TRUE, the axis state data is indicated continuously.

VAR_OUTPUT		
Variable name	**Data type**	**Description**
Valid	BOOL	TRUE if valid values are available.
Error	BOOL	Indicates whether an error has occurred in an FB.
ErrorID	WORD	Error identifier
ConstantVelocity	BOOL	The drive has achieved rotational speed.
RevZero	BOOL	Drive stationary, rotational speed = 0.
InWindow	BOOL	Axis is within the control window at the target position
InPosition	BOOL	The axis is in position.
IsMoving	BOOL	The axis moves, i.e. a command value is generated for this axis for the current interpolation. Please find further information in the HLI documentation under is_moving_r.
LrCommandedPos	LREAL	Target position of the axis, default unit [0.1 µm]
LrSetPos	LREAL	Command position of the axis, Default unit [0.1 µm]
LrActualPos	LREAL	Actual position of the axis, default unit [0.1 µm]
LrCommandedRev	LREAL	Programmed speed of a single axis (SAI), default unit [10^{-3} °/s]
LrSetRev	LREAL	Command rotational speed of the axis, default unit [10^{-3} °/ s]
LrActualRev	LREAL	Actual rotational speed of the axis, default unit [10^{-3} °/ s]
LrActualAccel	LREAL	Actual acceleration of the axis, default unit [1°/s^2]
LrSetAccel	LREAL	Command acceleration of the axis, default unit [1°/s^2]
IpoCommandedPos	LREAL	Target position of a single axis (SAI), default unit [10^{-4} °]
IpoSetPos	LREAL	Command position of a single axis (SAI), default unit [10^{-4} °]
IpoDistToGo	LREAL	Distance to go of a single axis (SAI), default unit [10^{-4} °]
IpoSetVel	LREAL	Command velocity of a single axis (SAI), default unit [10^{-3}°/ s]
IpoSetAccel	LREAL	Command acceleration of a single axis (SAI), default unit [1°/s^2]

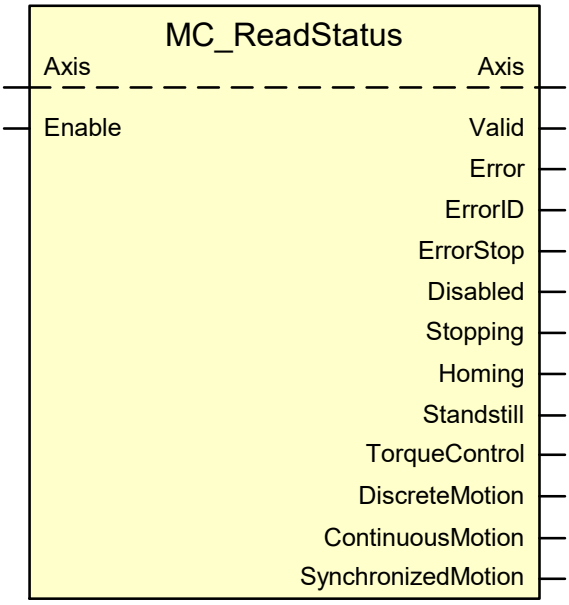
Behaviour of the FB:

- The information of an axis is updated cyclically as long as “Enable” has the value TRUE.

3.4 MC_ReadStatus

This FB is used to maintain the current PLCopen state of an axis. It is assigned to the group of administrative FBs.

Blockdiagramm



Parameters of the FB

VAR_IN_OUT		
Variable name	**Data type**	**Description**
axis	AXIS_REF	Axis reference

VAR_INPUT		
Variable name	**Data type**	**Description**
enable	BOOL	If the value is TRUE, the axis state data is indicated continuously.

VAR_OUTPUT		
Variable name	**Data type**	**Description**
Valid	BOOL	TRUE if valid values are available.
Error	BOOL	Indicates whether an error has occurred in an FB.
ErrorID	WORD	Error code.
Errorstop	BOOL	Axis is in “Errorstop” state
Disabled	BOOL	Axis is in “Disabled” state
Stopping	BOOL	Axis is in “Stopping” state
Homing	BOOL	Axis is in “Homing” state
Standstill	BOOL	Axis is in “Standstill” state
TorqueControl	BOOL	Axis is in “Torque Control” state
DiscreteMotion	BOOL	Axis is in “Discrete Motion” state
ContinuousMotion	BOOL	Axis is in “Continuous Motion” state
SynchronizedMotion	BOOL	Axis is in “Synchronised Motion” state

Behaviour of the FB

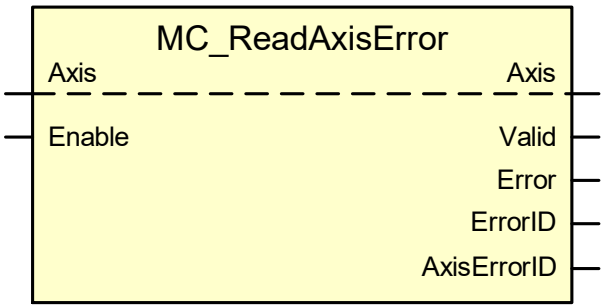
- The data of an axis is updated cyclically as long as "enable" has the value TRUE.

3.5 MC_ReadAxisError

When there is an axis error, the FB MC_ReadAxisError returns the error P-ERR-40001 (ERR_PLC_AX_MC) at its output “ErrorID”. At output “AxisErrorID” the error code of the motion controller is displayed. The associated error message texts are saved in separate text or XML files. All other error codes at the output “ErrorID” not corresponding to the value of the error P-ERR-40001 (ERR_PLC_AX_MC) indicate internal FB errors.

For explanations of the values at outputs “ErrorID” and “AxisErrorID”, refer to DIAG].

Block diagram



FB parameters

VAR_IN_OUT		
Variable name	**Data type**	**Description**
Axis	AXIS_REF	Axis reference

VAR_INPUT		
Variable name	**Data type**	**Description**
Enable	BOOL	When the value is TRUE, the error codes for axis error are supplied.

VAR_OUTPUT		
Variable name	**Data type**	**Description**
Valid	BOOL	If the output is TRUE, the error codes are valid.
Error	BOOL	Is TRUE when an error occurs on the axis.
ErrorID	WORD	Error identifier
AxisErrorID	UDINT	Error code of the axis error from the motion controller.

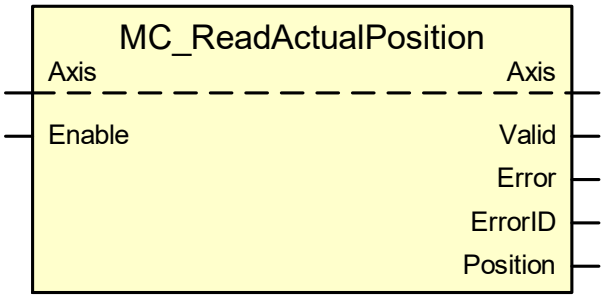
Behaviour of the FB:

- If the axis is in “Errorstop” state, the “Error” output becomes TRUE and the “ErrorID” output displays the error P-ERR-40001 (ERR_PLC_AX_MC). Then the “AxisErrorID” output shows the error code of the motion controller.

3.6 MC_ReadActualPosition

At its "position" output, the MC_ReadActualPosition returns the current position of an axis. It is assigned to the group of administrative FBs.

Blockdiagramm



Parameters of the FB

VAR_IN_OUT		
Variable name	**Data type**	**Description**
axis	AXIS_REF	Axis reference

VAR_INPUT		
Variable name	**Data type**	**Description**
enable	BOOL	If the value if TRUE, the positions are supplied continuously.

VAR_OUTPUT		
Variable name	**Data type**	**Description**
Valid	BOOL	A current position is applied to the “Position” output of the FB.
Error	BOOL	Indicates TRUE if an error occurs in the FB.
ErrorID	WORD	Error code.
Position	LREAL	Actual position of the axis, Default unit is [0.1 µm]

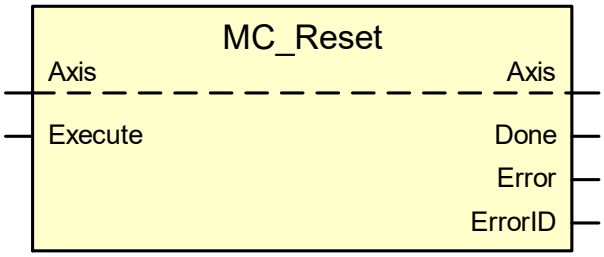
Behaviour of the FB

- If "enable" is FALSE, the value 0.0 is output at the "position" output.
- If "done" is TRUE, the "position" of the axis at the output is updated.

3.7 MC_Reset

The MC_Reset FB takes an axis from the "Errorstop" state to the "Standstill" state and discards all axis errors. Calling up MC_Reset has no influence on the outputs of the other FBs.

Blockdiagramm



Parameters of the FB

VAR_IN_OUT		
Variable name	**Data type**	**Description**
axis	AXIS_REF	Axis reference

VAR_INPUT		
Variable name	**Data type**	**Description**
Execute	BOOL	If the value if TRUE, a reset is executed on the assigned axis.

VAR_OUTPUT		
Variable name	**Data type**	**Description**
Done	BOOL	The axis is in the "Standstill“ state.
Error	BOOL	Indicates TRUE if an error occurs in the FB.
ErrorID	WORD	Error code.

Behaviour of the FB

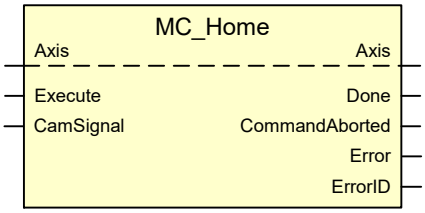
- Commanding a reset is permissible only if the axis is in the "Errorstop" state. If this is not the case, the "error" output is set to FALSE and "error_id" indicates an error value that indicates the current state of the axis.
- No new command can be sent when this FB is active. It cannot be retriggered.
- If the reset command cannot be sent to the axis within the number of FB invocations specified by the global datum MAX_RESET_RETRAILS, the "Error“ output is set to FALSE and “ErrorID” indicates the error P-ERR-44075 (ERR_PO_AX_DID_NOT_TAKE_ORDER). The value of MAX_RESET_RETRAILS can be adjusted to the specific application.
- If a reset command was successfully acknowledged after a specific number of FB invocations, this is indicated by the error P-ERR-44002 (ERR_PO_AX_ANSWER_TIME_OUT) at the “ErrorID” output and the “Error” output becomes TRUE. The number of FB calls can be specified for the specific application in the global data item MAX_RESET_WAIT_CYCLES.
- If the axis is in the TorqueControl and then assumes an error state, this state is restored after a successful reset. In all other cases, the axis assumes the "StandStill" state.

3.8 MC_Home

Reference point travel of the axis is commanded with the MC_Home FB. How an axis reacts to this command basically depends on the type of homing operation. This is specified in the MDS axis (achsmds*X*.lis) by the element P-AXIS-00299 (kenngr.homing.homing_type**)**.

The additional "CamSignal" input can specify the signal to adopt the homing position. Normally the signal source of the "CamSignal" input is the reference cam signal of the drive.

Blockdiagramm



Parameters of the FB

VAR_IN_OUT		
Variable name	**Data type**	**Description**
axis	AXIS_REF	Axis reference

VAR_INPUT		
Variable name	**Data type**	**Description**
Execute	BOOL	Homing is commanded on a rising edge at the input.
CamSignal	BOOL	Signal to command adoption of the homing position. For further application notes, see Section 2.8.1 [▶ 54].

VAR_OUTPUT		
Variable name	**Data type**	**Description**
Done	BOOL	When TRUE, homing is completed and the axis is in “Standstill” state.
CommandAborted	BOOL	TRUE indicates that the command was aborted by another command.
Error	BOOL	Is TRUE when an error occurs in the FB.
ErrorID	WORD	Error code.

General behaviour of the FB

- Commanding reference point travel is permissible only if the axis is in the "Standstill" state. If this is not the case, the "error" output is set to FALSE and "error_id" indicates an error value that designates the state the axis is currently in.
- No new command can be sent when this FB is active. It cannot be retriggered. If an attempt is made to do this anyway, the FB supplies the error P-ERR-44007 (ERR_PO_AX_HOMING_RETRIGGERD).
- The axis change to the "Homing" state directly after command of reference point travel.

3.8.1 Parameter values when the“CamSignal” input is used

Attention

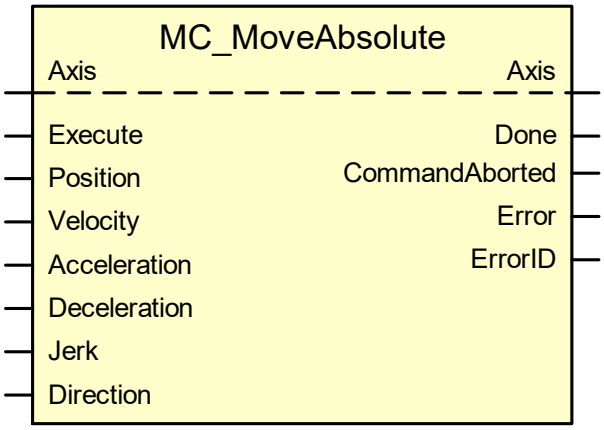
The condition for proper functioning is that the parameter P-AXIS-00036 is assigned the value 0. The parameter P-AXIS-00038 defines whether the homing position is adopted on a rising or falling edge at the “CamSignal”.

3.9 MC_MoveAbsolute

A motion of the axis to an absolute position is commanded by the FB MC_MoveAbsolute. The motion is always executed jerk-limited at constant jerk set in "Jerk". This value is valid for both "Acceleration" and "Deceleration".

If the optional parameters "Acceleration, "Deceleration" and "Jerk" are not specified or set to <= 0, the dynamic values are taken from the corresponding axis list.

Block diagram



FB parameters

VAR_IN_OUT		
Variable name	**Data type**	**Description**
Axis	AXIS_REF	Axis reference

VAR_INPUT		
Variable name	**Data type**	**Description**
Execute	BOOL	A rising edge at the input commands the motion to an absolute position.
Position	LREAL	Target position of the motion, Value range [-2147483648.0, 2147483647.0] Default unit [0.1µm or 10^{-4} °]
Velocity	LREAL	Maximum velocity of the motion. This value must be specified positive. The velocity is not necessarily reached. Value range [1.0, 2147483647.0] Default unit [1µm/s or 10^{-3} °/s]
Acceleration	LREAL	Acceleration value. This value must be specified positive. Value range [0.0, 2147483647.0] Default unit [1mm/s^2 or 1°/s^2]
Deceleration	LREAL	Deceleration value. This value must be always be specified positive. Value range [0.0, 2147483647.0] Default unit [1mm/s^2 or 1°/s^2]
Jerk	LREAL	Jerk value. This value must be always be specified positive. Value range [0.0, 2147483647.0] Default unit [1mm/s^3 or 1°/s^3]
Direction	INT	Direction in which the motion is started. One of the 4 values: **1** positive direction **2** shortest path **3** negative direction **4** current direction Specifying the direction is effective only if a modulo coordinate system was defined for the axis in the axis-specific machine data using parameter **kenngr.achs_mode** (P-AXIS-00015) since the direction is always unambiguous with linear axes.

VAR_OUTPUT		
Variable name	**Data type**	**Description**
Done	BOOL	The axis is within a range close to the target position. The range is specified by the parameter **getriebe[...].window** (P-AXIS-00236) of the axis-specific machine data.
CommandAborted	BOOL	TRUE indicates that the command was aborted by another command.
Error	BOOL	Is TRUE if an error occurs in the FB.
ErrorID	WORD	Error identifier

Behaviour of the FB:

- This motion is commanded only when the axis is in the “Standstill” or “Continuous Motion” or “Discrete Motion” or “Synchronised Motion” state. If this is not the case, the error output is set to “FALSE and “ErrorID” indicates an error value that designates the state the axis is currently in.

- The “Position, “Velocity, “Acceleration”, “Deceleration” and “Jerk” input variables are monitored only for their value range in the FB. If the values are exceeded, an error number referring to the variable is output at the “ErrorID” output and “Error” is set to TRUE. Moreover, the maximum velocity and acceleration data is limited within the motion controller. If speed values greater than **getriebe[0].dynamik.vb_max** (P-AXIS-00212) or acceleration or deceleration values greater than **getriebe[0].dynamik.a_max** (P-AXIS-00008) are commanded, the motion takes place nevertheless. However, the values are limited to the parameterised values and the motion controller issues a warning P-ERR-60211 or P-ERR-60188 to P-ERR-60194.
- Values for jerk are practical when they correspond to a minimum ramp time of T_A (cycle time of the IPO, e.g. 2 ms) and a maximum ramp time of 100s. The limit transition to non-jerk-limited velocity control lies at the minimum ramp time of T_A. With a value of 0, the maximum ramp time from the axis MDS is used to determine the jerk.
- This function block replaces active commands with new commands, making the function block retriggerable. The active command is then aborted.
- If an unreferenced axis is commended by this FB, the axis does not start. The error P-ERR-60142 is reported on the “MC_ReadAxisError” function block.
- The positions of the software limit switches (**kenngr.swe_pos** (P-AXIS-00178)**, kenngr.swe_neg** (P-AXIS-00177)**, kenngr.swe_toleranz** (P-AXIS-00179)) are monitored if the axis was defined either as a linear or a rotary axis. This means that the **kenngr.achs_typ** parameter of the axis machine data (P-AXIS-00018) has the value 0x0001 or 0x0002. If the target position commanded exceeds the position of the software limit switches, the axis starts off nevertheless. If a software limit switch position is reached, the axis is braked, keeping to the defined acceleration.

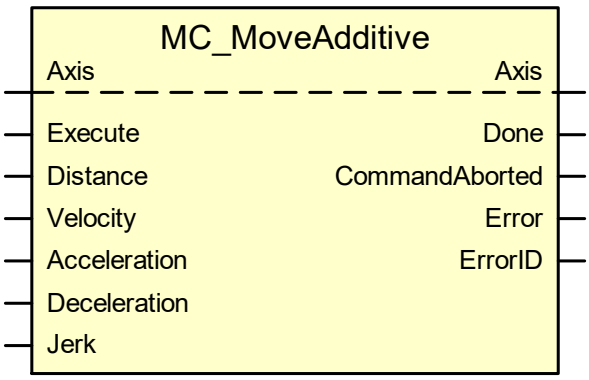
3.10 MC_MoveAdditive

With the MC_MoveAdditive FB, a relative motion is commanded, in addition to the commanded position, if the axis is in the **Discrete Motion** state. The motion is always executed jerk-limited at the constant set input "Jerk". This value is valid for both "Acceleration" and "Deceleration".

If the axis is in the "Continuous Motion" state and receives a command from this FB, the relative distance to the current position at the time of commanding is added.

If the optional parameters "Acceleration, "Deceleration" and "Jerk" are not specified or set to <= 0, the dynamic values are taken from the corresponding axis list.

Block diagram



FB parameters

VAR_IN_OUT		
Variable name	**Data type**	**Description**
Axis	AXIS_REF	Axis reference

VAR_INPUT		
Variable name	**Data type**	**Description**
Execute	BOOL	The motion starts on the rising edge of this input signal.
Distance	LREAL	Target position of the motion, Value range [-2147483648.0, 2147483647.0] Default unit [0.1µm or 10^{-4} °]
Velocity	LREAL	Maximum velocity of the motion. This value must be specified positive. The velocity is not necessarily reached. Value range [1.0, 2147483647.0] Default unit [1µm/s or 10^{-3} °/s]
Acceleration	LREAL	Acceleration value. This value must be specified positive. Value range [0.0, 2147483647.0] Default unit [1mm/s^2 or 1°/s^2]
Deceleration	LREAL	Deceleration value. This value must be always be specified positive. Value range [0.0, 2147483647.0] Default unit [1mm/s^2 or 1°/s^2]
Jerk	LREAL	Jerk value. This value must be always be specified positive. Value range [0.0, 2147483647.0] Default unit [1mm/s^3 or 1°/s^3]

VAR_OUTPUT		
Variable name	**Data type**	**Description**
Done	BOOL	The axis is within a range close to the target position. The range is specified by the parameter **getriebe[...].window** (P-AXIS-00236) of the axis-specific machine data.
CommandAborted	BOOL	TRUE indicates that the command was aborted by another command.
Error	BOOL	Is TRUE if an error occurs in the FB.
ErrorID	WORD	Error identifier

Behaviour of the FB:

- This motion is commanded only when the axis is in the “Standstill” or “Continuous Motion” or “Discrete Motion” or “Synchronised Motion” state. If this is not the case, the error output is set to “FALSE and “ErrorID” indicates an error value that designates the state the axis is currently in.
- The “Position, “Velocity, “Acceleration”, “Deceleration” and “Jerk” input variables are monitored only for their value range in the FB. If the values are exceeded, an error number referring to the variable is output at the “ErrorID” output and “Error” is set to TRUE. Moreover, the maximum velocity and acceleration data is limited within the motion controller. If speed values greater than **getriebe[0].dynamik.vb_max** (P-AXIS-00212) or acceleration or deceleration values greater than **getriebe[0].dynamik.a_max** (P-AXIS-00008) are commanded, the motion takes place nevertheless. The values are limited to the parameterised values and a warning P-ERR-60211 or P-ERR-60188 to P-ERR-60194 is output.
- Values for jerk are practical when they correspond to a minimum ramp time of T_A (cycle time of the IPO, e.g. 2 ms) and a maximum ramp time of 100s. The limit transition to non-jerk-limited velocity control lies at the minimum ramp time of T_A. With a value of 0, the maximum ramp time from the axis MDS is used to determine the jerk.

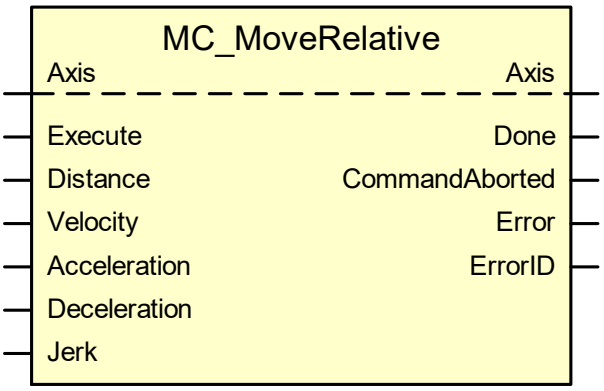
- This function block replaces active commands with new commands, making the function block retriggerable. The active command is then aborted.
- If a non-referenced axis is commanded by this FB, the axis travels at the speed defined by the **getriebe[...].vb_reflow** parameter (P-AXIS-00218) of the axis machine data.
- The positions of the software limit switches (**kenngr.swe_pos** (P-AXIS-00178)**, kenngr.swe_neg** (P-AXIS-00177)**, kenngr.swe_toleranz** (P-AXIS-00179)) are monitored if the axis was defined either as a linear or a rotary axis. This means that the **kenngr.achs_typ** parameter of the axis machine data (P-AXIS-00018) has the value 0x0001 or 0x0002. If the target position commanded exceeds the position of the software limit switches, the axis starts off nevertheless. If a software limit switch position is reached, the axis is braked, keeping to the defined acceleration.

3.11 MC_MoveRelative

A relative motion is commanded with the MC_MoveRelative FB, in addition to the current position. This is regardless of whether the axis is in "Discrete Motion" or "Continuous Motion" state. The motion is always executed jerk-limited at the constant set input "Jerk". This value is valid for both "Acceleration" and "Deceleration".

If the optional parameters “Acceleration, “Deceleration” and “Jerk” are not specified or set to <= 0, the dynamic values are taken from the corresponding axis list.

Block diagram



FB parameters

VAR_IN_OUT		
Variable name	**Data type**	**Description**
Axis	AXIS_REF	Axis reference

VAR_INPUT		
Variable name	**Data type**	**Description**
Execute	BOOL	The motion starts at the rising edge of this input signal.
Distance	LREAL	Relative distance for the motion. Value range [-2147483648.0, 2147483647.0] Default unit [0.1µm or 10^{-4} °]
Velocity	LREAL	Maximum velocity of the motion. This value must be specified positive. The velocity is not necessarily reached. Value range [1.0, 2147483647.0] Default unit [1µm/s or 10^{-3} °/s]
Acceleration	LREAL	Acceleration value. This value must be specified positive. Value range [0.0, 2147483647.0] Default unit [1mm/s^2 or 1°/s^2]
Deceleration	LREAL	Deceleration value. This value must be always be specified positive. Value range [0.0, 2147483647.0] Default unit [1mm/s^2 or 1°/s^2]
Jerk	LREAL	Jerk value. This value must be always be specified positive. Value range [0.0, 2147483647.0] Default unit [1mm/s^3 or 1°/s^3]

VAR_OUTPUT		
Variable name	**Data type**	**Description**
Done	BOOL	The axis is within a range close to the target position. The range is specified by the parameter **getriebe[...].window** (P-AXIS-00236) of the axis-specific machine data.
CommandAborted	BOOL	TRUE indicates that the command was aborted by another command.
Error	BOOL	Is TRUE if an error occurs in the FB.
ErrorID	WORD	Error identifier

Behaviour of the FB:

- This motion is commanded only when the axis is in the “Standstill” or “Continuous Motion” or “Discrete Motion” or “Synchronised Motion” state. If this is not the case, the error output is set to “FALSE and “ErrorID” indicates an error value that designates the state the axis is currently in.
- The “Position, “Velocity, “Acceleration”, “Deceleration” and “Jerk” input variables are monitored only for their value range in the FB. If the values are exceeded, an error number referring to the variable is output at the “ErrorID” output and “Error” is set to TRUE. Moreover, the maximum velocity and acceleration data is limited within the motion controller. If speed values greater than **getriebe[0].dynamik.vb_max** (P-AXIS-00212) or acceleration or deceleration values greater than **getriebe[0].dynamik.a_max** (P-AXIS-00008) are commanded, the motion takes place nevertheless. The values are limited to the parameterised values and a warning P-ERR-60211 or P-ERR-60188 to P-ERR-60194 is output.
- Values for jerk are practical if they correspond to a minimum ramp time of T_A (cycle time of the IPO, e.g. 2 ms) and a maximum ramp time of 100s. The limit transition to non-jerk-limited velocity control lies at the minimum ramp time of T_A. With a value of 0, the maximum ramp time from the axis MDS is used to determine the jerk.
- This function block replaces active commands with new commands, making the function block retriggerable. The active command is then aborted.
- If a non-referenced axis is commanded by this FB, the axis travels at the speed defined by the **getriebe[...].vb_reflow** parameter (P-AXIS-00218) of the axis machine data.
- The positions of the software limit switches (**kenngr.swe_pos** (P-AXIS-00178)**, kenngr.swe_neg** (P-AXIS-00177)**, kenngr.swe_toleranz** (P-AXIS-00179)) are monitored if the axis was defined either as a linear or a rotary axis. This means that the **kenngr.achs_typ** parameter of the axis machine data (P-AXIS-00018) has the value 0x0001 or 0x0002. If the target position commanded exceeds the position of the software limit switches, the axis starts off nevertheless. If a software limit switch position is reached, the axis is braked, keeping to the defined acceleration.

3.12 MC_MoveSuperImposed

A relative motion is commanded with the MC_MoveSuperImposed FB, in addition to motion already active. The active motion is not interrupted but is superimposed over the commanded one. The motion is always executed jerk-limited with the constant set input "Jerk". This value is valid for both "Acceleration" and "Deceleration".

If the optional parameters “Acceleration, “Deceleration” and “Jerk” are not specified or set to <= 0, the dynamic values are taken from the corresponding axis list.

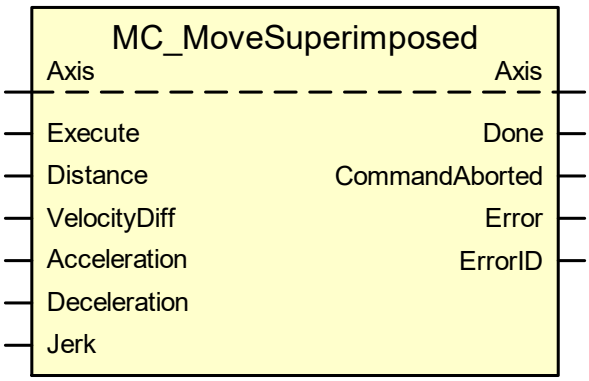
If the axis is already in a motion state and receives a command from this FB, it results in

1. a superimposed interpolation
2. a change in velocity
3. a change in the target position of the active motion.

If the axis is in the “Standstill” state, a command by the MC_MoveSuperImposed FB acts like a command by an MC_MoveRelative FB.

Since "Acceleration" values are also superimposed in the case of superimposed interpolation, corresponding axis parameters must be defined to ensure that the axis is not dynamically overloaded.

Block diagram



FB parameters

VAR_IN_OUT		
Variable name	**Data type**	**Description**
Axis	AXIS_REF	Axis reference

VAR_INPUT		
Variable name	**Data type**	**Description**
Execute	BOOL	Superimposition of the motion is started on the rising edge of this input signal.
Distance	LREAL	Relative distance of the motion Value range [-2147483648.0, 2147483647.0] Default unit [0.1µm or 10^{-4} °]
VelocityDiff	LREAL	Maximum difference to the velocity of the underlying motion. This value must be specified positive. The velocity is not necessarily reached. Value range [1. 0, 2147483647.0] Default unit [1µm/s or 10^{-3} °/s]
Acceleration	LREAL	Acceleration value. This value must be specified positive. Value range [0.0, 2147483647.0] Default unit [1mm/s^2 or 1°/s^2]
Deceleration	LREAL	Deceleration value. This value must be always be specified positive. Value range [0.0, 2147483647.0] Default unit [1mm/s^2 or 1°/s^2]
Jerk	LREAL	Jerk value. This value must be always be specified positive. Value range [0.0, 2147483647.0] Default unit [1mm/s^3 or 1°/s^3]

VAR_OUTPUT		
Variable name	**Data type**	**Description**
Done	BOOL	The relative distance of the active movement is superimposed.
CommandAborted	BOOL	TRUE indicates that the command was aborted by another command.
Error	BOOL	Is TRUE if an error occurs in the FB.
ErrorID	WORD	Error identifier

Behaviour of the FB:

- This motion is commanded only when the axis is in the “Standstill” or “Continuous Motion” or “Discrete Motion” or “Synchronised Motion” state. If this is not the case, the error output is set to “FALSE and “ErrorID” indicates an error value that designates the state the axis is currently in.
- The “Distance, “VelocityDiff, “Acceleration”, “Deceleration” and “Jerk” input variables are monitored only for their value range in the FB. If the values are exceeded, an error number referring to the variable is output at the “ErrorID” output and “Error” is set to TRUE. Moreover, the maximum velocity and acceleration data is limited within the motion controller. If speed values greater than **getriebe[0].dynamik.vb_max** (P-AXIS-00212) or acceleration or deceleration values greater than **getriebe[0].dynamik.a_max** (P-AXIS-00008) are commanded, the motion takes place nevertheless. The values are limited to the parameterised values and a warning P-ERR-60211 or P-ERR-60188 to P-ERR-60194 is output.
- Values for jerk are practical when they correspond to a minimum ramp time of T_A (cycle time of the IPO, e.g. 2 ms) and a maximum ramp time of 100s. The limit transition to non-jerk-limited velocity control lies at the minimum ramp time of T_A. With a value of 0, the maximum ramp time from the axis MDS is used to determine the jerk.

- With this function block, active commands can be modified by an additional, superimposed motion. The following rules apply to command assertion and command aborting:
- If no interpolation is active, i.e. if the axis is in the “Standstill” state**,** a command by the MC_MoveSuperImposed FB acts like a command by an MC_MoveRelative FB.
- If interpolation is already active, i.e. the axis is in one of the “Continuous Motion”, “Discrete Motion” or “Synchronised Motion” states, a command by the MC_MoveSuperImposed FB starts a further, superimposed interpolation.
- If a superimposed interpolation (cf. 2.) is already active, a new command by the MC_MoveSuperImposed FB ensures that only the currently superimposed interpolation is replaced by the new MC_MoveSuperImposed command, while the subordinate interpolation is not affected.
- If a superimposed interpolation (vgl2.) is already active, there is a renewed command with any MC_Move*** except MC_MoveSuperImposed; this causes the two active interpolations to abort. In this way, the underlying and the superimposed interpolation are replaced by the new MC_Move*** command.
- If a non-referenced axis is commanded by this FB, the axis travels at the speed defined by the **getriebe[...].vb_reflow** parameter (P-AXIS-00218) of the axis machine data.
- The positions of the software limit switches (**kenngr.swe_pos** (P-AXIS-00178)**, kenngr.swe_neg** (P-AXIS-00177)**, kenngr.swe_toleranz** (P-AXIS-00179)) are monitored if the axis was defined either as a linear or a rotary axis. This means that the **kenngr.achs_typ** parameter of the axis machine data (P-AXIS-00018) has the value 0x0001 or 0x0002. If the target position commanded exceeds the position of the software limit switches, the axis starts off nevertheless. If a software limit switch position is reached, the axis is braked, keeping to the defined acceleration.

3.13 MC_MoveVelocity

The MC_MoveVelocity FB commands an endless motion at the specified speed. The motion is always executed jerk-limited with the constant set input "Jerk". This value is valid for both "Acceleration" and "Deceleration".

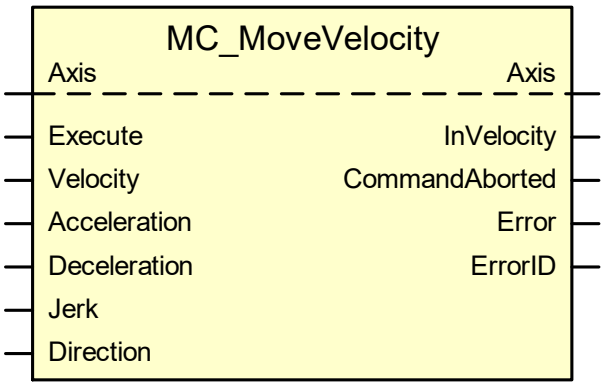
If the optional parameters “Acceleration, “Deceleration” and “Jerk” are not specified or set to <= 0, the dynamic values are taken from the corresponding axis list.

To stop the motion, the FB must be interrupted by another FB that sends a new command to the axis.

The “InVelocity” output is reset when a falling edge is detected at the “Execute” input or when the effect of the FB is ended due to the command of another FB.

In conjunction with an MC_MoveSuperImposed FB, the “InVelocity” output remains TRUE.

Block diagram



FB parameters

VAR_IN_OUT		
Variable name	**Data type**	**Description**
Axis	AXIS_REF	Axis reference

VAR_INPUT		
Variable name	**Data type**	**Description**
Execute	BOOL	The motion starts at the rising edge of this input signal.
Velocity	LREAL	Velocity value of endless motion. This value must be specified positive. Value range [1. 0, 2147483647.0] Default unit [1µm/s or 10-3 °/s]
Acceleration	LREAL	Acceleration value. This value must be specified positive. Value range [0. 0, 2147483647.0] Default unit [1mm/s^2 or 1°/s^2]
Deceleration	LREAL	Deceleration value. This value must be always be specified positive. Value range [0. 0, 2147483647.0] Default unit [1mm/s^2 or 1°/s^2]
Jerk	LREAL	Jerk value. This value must be always be specified positive. Value range [0. 0, 2147483647.0] Default unit [1mm/s^3 or 1°/s^3]
Direction	INT	Direction in which the motion is started. One of the 3 values: Positive direction **3** negative direction **4** current direction

VAR_OUTPUT		
Variable name	**Data type**	**Description**
InVelocity	BOOL	The commanded velocity was reached (for the first time).
CommandAborted	BOOL	TRUE indicates that the command was aborted by another command.
Error	BOOL	Is TRUE if an error occurs in the FB.
ErrorID	WORD	Error identifier

Behaviour of the FB:

- This motion is commanded only when the axis is in the “Standstill” or “Continuous Motion” or “Discrete Motion” or “Synchronised Motion” state. If this is not the case, the error output is set to “FALSE and “ErrorID” indicates an error value that designates the state the axis is currently in.
- The “Position, “Velocity, “Acceleration”, “Deceleration” and “Jerk” input variables are monitored only for their value range in the FB. If the values are exceeded, an error number referring to the variable is output at the “ErrorID” output and “Error” is set to TRUE. Moreover, the maximum velocity and acceleration data is limited within the motion controller. If speed values greater than **getriebe[0].dynamik.vb_max** (P-AXIS-00212) or acceleration or deceleration values greater than **getriebe[0].dynamik.a_max** (P-AXIS-00008) are commanded, the motion takes place nevertheless. The values are limited to the parameterised values and a warning P-ERR-60211 or P-ERR-60188 to P-ERR-60194 is output.
- Values that correspond to a minimum ramp time of T_A (cycle time of the IPO, e.g. 2 ms) and a maximum ramp time of 100s are practical for jerk. The limit transition to non-jerk-limited velocity control lies at the minimum ramp time of T_A. With a value of 0, the maximum ramp time from the axis MDS is used to determine the jerk.

- This function block replaces active commands with new commands, making the function block retriggerable. The active command is then aborted.
- If a non-referenced axis is commanded by this FB, the axis travels at the speed defined by the **getriebe[...].vb_reflow** parameter (P-AXIS-00218) of the axis machine data.
- The positions of the software limit switches (**kenngr.swe_pos** (P-AXIS-00178)**, kenngr.swe_neg** (P-AXIS-00177)**, kenngr.swe_toleranz** (P-AXIS-00179)) are monitored if the axis was defined either as a linear or a rotary axis. This means that the **kenngr.achs_typ** parameter of the axis machine data (P-AXIS-00018) has the value 0x0001 or 0x0002. If a software limit switch position is reached, the axis is braked, keeping to the defined acceleration.
- The acknowledgement at the "InVelocity" output depends on the setting of the **kenngr.vb_prozent** (P-AXIS-00217) value. This value defines a percentage tolerance window for the target velocity. The "InVelocity" acknowledgement appears as soon as you enter the tolerance window.

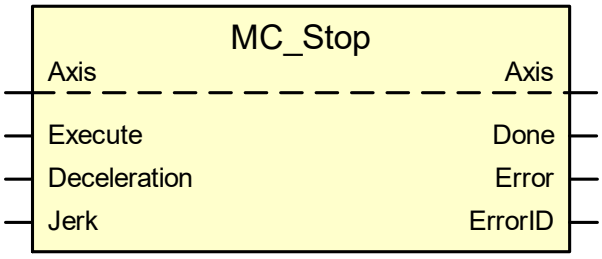
3.14 MC_Stop

The MC_Stop FB leads to a controlled motion stop and sets the axis to the “Stopping” state. The motion stop is always jerk-limited with the constant set input "jerk” for the deceleration rate.

If the optional parameters ”Deceleration” and “Jerk” are not specified or set to <= 0, the dynamic values are taken from the corresponding axis list.

It aborts every ongoing command by other motion FBs. Contrary to the definition of the state graph in PLCopen, it has proven useful in practice to also be able to abort an axis that is in the "Stopping“ state by commanding it with another motion FB. The MC_Stop FB sets the “Done” output as soon as the axis is at standstill. However, the “Stopping” state of the axis is retained for as long as the “Execute” input is still TRUE.

Block diagram



FB parameters

VAR_IN_OUT		
Variable name	**Data type**	**Description**
Axis	AXIS_REF	Axis reference
VAR_INPUT		
Variable name	**Data type**	**Description**
Execute	BOOL	Starts the command on the rising edge.
Deceleration	LREAL	Deceleration value. This value must be always be specified positive. Value range [0.0, 2147483647.0] Default unit [1mm/s^2 or 1°/s^2]
Jerk	LREAL	Jerk value. This value must be always be specified positive. Value range [0.0, 2147483647.0] Default unit [1mm/s^3 or 1°/s^3]
VAR_OUTPUT		
Variable name	**Data type**	**Description**
Done	BOOL	0 velocity was reached. **Caution:** The "Done“ output merely indicates that the axis is stationary. The automatic axis state indicator remains initially in the STOPPING state. Only when the "Execute” input is also FALSE does the automatic axis state indicator change to the “STANDSTILL” state.
Error	BOOL	Is TRUE if an error occurs in the FB.
ErrorID	WORD	Error identifier

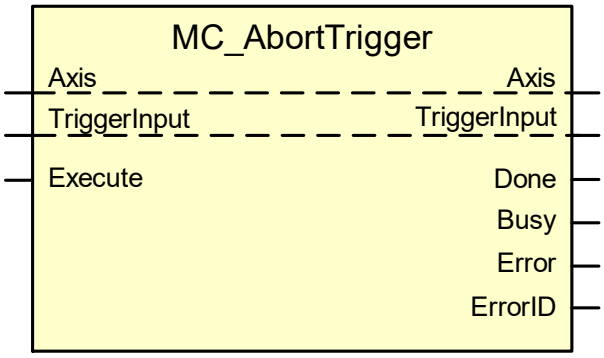
Behaviour of the FB:

- The command for stopping the axis is issued only when the axis is in the “Standstill or “Continuous Motion” or “Discrete Motion” or “Synchronized Motion” or “Homing” state. If this is not the case, the “Error” output is set to FALSE and “ErrorID” indicates an error value that designates the state the axis is currently in.
- The “Deceleration” and “Jerk” input variables are monitored only for their value ranges in the FB. If the values are exceeded, an error number referring to the variable is output at the “ErrorID” output and “Error” is set to TRUE. Moreover, the deceleration value is limited in the motion controller. If a deceleration greater than **getriebe[0].dynamik.a_max** is commanded (P-AXIS-00008) is commanded, the motion is still executed. However, the values are limited to the parameterised values and a warning is issued.
- Practical values for jerk are between the minimum jerk of deceleration/100 s and the maximum jerk of deceleration/T_A. The limit transition to non-jerk-limited velocity control lies at the minimum ramp time of T_A. When the value is 0, the maximum ramp time in P-AXIS-00195 to P-AXIS-00198 is used to determine jerk.
- No new command can be sent when this FB is active. It is not retriggerable and therefore sets the "Error“ error output to TRUE and indicates at output “ErrorID” P-ERR-44008 (ERR_PO_AX_STOP_RETRIGGERD).

3.15 MC_AbortTrigger

The MC_AbortTrigger FB is used to abort commands that are related to trigger events (e.g. MC_TouchProbe).

Blockdiagramm



Parameters of the FB

VAR_IN_OUT		
Variable name	**Data type**	**Description**
axis	AXIS_REF	Axis reference
TriggerInput	TRIGGER_REF [▶ 10]	Reference for the trigger signal source (see chapter entitled Data structure TRIGGER_REF [▶ 10]).

VAR_INPUT		
Variable name	**Data type**	**Description**
Execute	BOOL	Aborts a command of an FB MC_TouchProbe on the rising edge.

VAR_OUTPUT		
Variable name	**Data type**	**Description**
Done	BOOL	Measuring command is aborted.
Busy	BOOL	Is TRUE while a measuring command is aborted.
Error	BOOL	Is TRUE when an error occurs in the FB.
ErrorID	WORD	Error code.

Behaviour of the FB

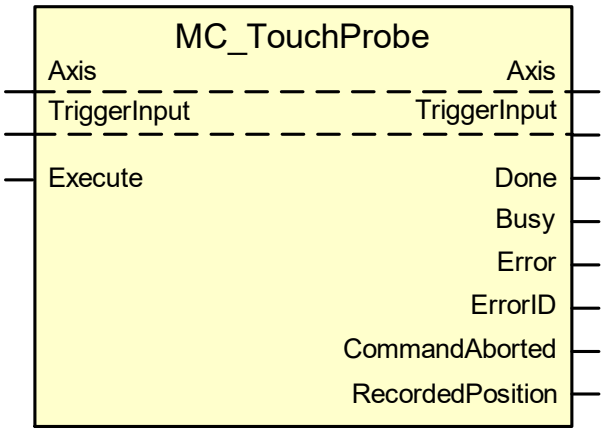
- A measurement command is aborted only when the axis is in the "Standstill" or "Continuous Motion" or "Discrete Motion" or "Synchronized Motion" or "Stopping" state. Moreover, no command for aborting of a measurement command must be active in the measurement channel specified by "TriggerInput". The measurement channel must therefore be in the "TP_IDLE" or "TOUCHPROBING" state. If this is not the case, the "error" output is set to FALSE and "error_id" indicates an error value that designates the state the axis is currently in.
- If a command for aborting a measurement command in a measurement channel is active, no further command for aborting in relation to this measurement channel can be sent. Therefore, the FB is not retriggerable. However, if the command is still sent, the “Error” output is TRUE and signals the errorP-ERR-44010 (ERR_PO_AX_TPABORT_RETRIG) at the “ErrorID” output.

- Errors are also generated if the axis reference, the measuring channel or the trigger event of the measurement process were changed when they are commanded in the active state. The following error may occur in these cases:
- P-ERR-44001 (ERR_PO_AX_REF_CHG_WHILE_ACTIVE)
- P-ERR-44011 (ERR_PO_AX_INPREF_TPCH_CHW_ACTIVE)
- P-ERR-44012 (ERR_PO_AX_INPREF_TPMD_CHW_ACTIVE)

3.16 MC_TouchProbe

The MC_TouchProbe FB records an axis position when a trigger event occurs. The measurement channel and method (rising or falling edge of the trigger signal) are defined via the reference for the trigger signal source.

Blockdiagramm



Parameters of the FB

VAR_IN_OUT		
Variable name	**Data type**	**Description**
axis	AXIS_REF	Axis reference
TriggerInput	TRIGGER_REF [▶ 10]	Reference for the trigger signal source (see chapter entitled Data structure TRIGGER_REF [▶ 10]).

VAR_INPUT		
Variable name	**Data type**	**Description**
Execute	BOOL	Starts the command on the rising edge.

VAR_OUTPUT		
Variable name	**Data type**	**Description**
Done	BOOL	Trigger event was detected.
Busy	BOOL	Is TRUE as long as the input/output waits for the trigger event.
Error	BOOL	Indicates TRUE if an error occurs in the FB.
ErrorID	WORD	Error code.
CommandAborted	BOOL	Command is aborted by another command of an FB MC_AbortTrigger.
RecordedPosition	LREAL	Axis position when the trigger event occurred. Default unit [0.1µm or 10^{-4} °]

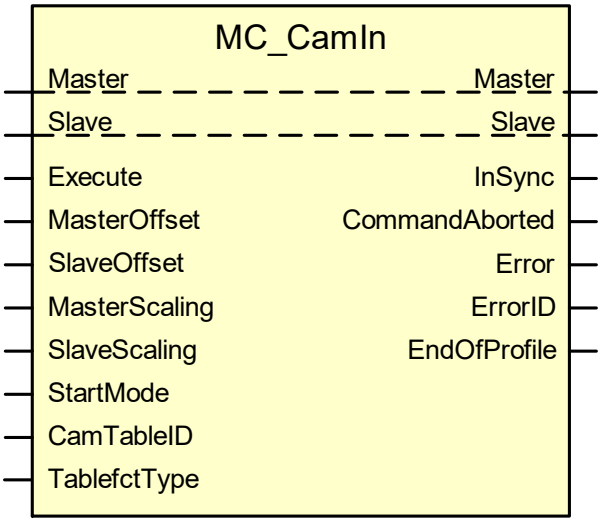
Behaviour of the FB

- A measurement command is issued only when the axis is in the "Standstill" or "Continuous Motion" or "Discrete Motion" or "Synchronized Motion" or "Stopping" state. Moreover, no measurement command must be active in the measurement channel specified by "TriggerInput". The measurement channel must therefore be in the "TP_IDLE" state. If this is not the case, the "error" output is set to FALSE and "error_id" indicates an error value that designates the state the axis is currently in.
- If a measurement command in a measurement channel is active, no further measurement command in relation to this measurement channel can be sent. Therefore, the FB is not retriggerable. If a command is still sent, the “Error“ output becomes TRUE and signals the error P-ERR-44009 (ERR_PO_AX_TPROBE_RETRIG) at the “ErrorID“ output.
- Errors are also generated when the axis reference, the measuring channel or the trigger event of the measurement process were changed while a measurement command is active. The following error may occur in these cases:
- P-ERR-44001 (ERR_PO_AX_REF_CHG_WHILE_ACTIVE)
- P-ERR-44011 (ERR_PO_AX_INPREF_TPCH_CHW_ACTIVE)
- P-ERR-44012 (ERR_PO_AX_INPREF_TPMD_CHW_ACTIVE).

3.17 MC_CamIn

The MC_CamIn FB permits travel along cams [▶ 12] listed in a table.

Block diagram



FB parameters

VAR_IN_OUT		
Variable name	**Data type**	**Description**
Master	AXIS_REF	Axis reference of the master axis.
Slave	AXIS_REF	Axis reference of the slave axis
VAR_INPUT		
Variable name	**Data type**	**Description**
Execute	BOOL	Starts the command on the rising edge.
MasterOffset	LREAL	Offset of the master axis table Value range [-2147483648.0, 2147483647.0] Default unit [0.1µm or 10^{-4} °]
SlaveOffset	LREAL	Offset of the slave axis table Value range [-2147483648.0, 2147483647.0] Default unit [0.1µm or 10^{-4} °]
MasterScaling	LREAL	Factor for the master axis profile Value range [-2147483648.0, 2147483647.0] **! Important !:** Only integer values should be currently used!
SlaveScaling	LREAL	Factor for the slave axis profile Value range [-2147483648.0, 2147483647.0]
StartMode	UDINT	See explanations in the Section Values at input “StartMode” [▶ 78].
CamTableID	UINT	Identifier of cam used for the motion.
TableFctType	UDINT	See explanations in the Section Values at the “TableFctType” input [▶ 79].
VAR_OUTPUT		
Variable name	**Data type**	**Description**
InSync	BOOL	Indicates that the motion of the slave axis is now synchronous with the cam.
CommandAborted	BOOL	TRUE indicates that the command was aborted by another command.
Error	BOOL	Is TRUE if an error occurs in the FB.
ErrorID	WORD	Error identifier
EndOfProfile	BOOL	Output which indicates the cyclical end of the cam profile.

Behaviour of the FB:

- The command to move along a cam can only be issued only when the slave axis is in one of the following states:
- Standstill
- Continuous Motion
- Discrete Motion
- Synchronised Motion

If this is not the case, the “Error” output is set to FALSE and “ErrorID” indicates an error value that designates the state the axis is currently in.

- The value range of the “MasterOffset”, “SlaveOffset, “MasterScaling“ and “SlaveScaling” input variables is monitored and, if it is exceeded, an error number referring to the variable is output at the “ErrorID” output and “Error” is set to TRUE.
- If no table is loaded and the CamIn FB is commanded, the error P-ERR-40001 (ERR_PLC_AX_MC) is issued at the “ErrorID” output and the error P-ERR-60172 is issued at the AxisErrorID output of an MC_ReadAxisError function block.
- The function block is retriggerable. Therefore, it can send a command again in the active state.
- For an axis to work as a master axis and for it to transfer the necessary synchronisation information to all axes (slaves) cyclically, the parameter **cam_gear.is_master** (P-AXIS-00288) must be set to 1. An axis configured as master can be simultaneously coupled to another master axis as slave.
- The MC_CamIn FB can only be used for rotary and linear slave axes.
- Master/slave coupling is realised with the setpoints of the master.
- A slave axis is defined with the following axis parameters:
- cam_gear.v_diff_percent [▶ 80] (P-AXIS-00289)
- cam_gear.time_in_window (P-AXIS-00290)
- cam_gear.time_out_in_window (P-AXIS-00291)
- **cam_gear.v_max_slave** (P-AXIS-00303)
- **cam_gear.a_max_slave** (P-AXIS-00304)
- cam_gear.v_phasing [▶ 80] (P-AXIS-00305)
- cam_gear.a_phasing (P-AXIS-00306)
- cam_gear.d_phasing (P-AXIS-00307)
- cam_gear.j_phasing (P-AXIS-00376)
- cam_gear.j_vel_sync [▶ 80] (P-AXIS-00377)

For further explanations, see the Sections Influence of axis parameters on the coupling process [▶ 80] and Cam functionality [▶ 12].

3.17.1 Values at input “StartMode”

```
HLI_CI_RAMP_IN              : UDINT := 1;
HLI_CI_ABSOLUTE             : UDINT := 2;
HLI_CI_RELATIVE             : UDINT := 4;
```

Ramp-In and Absolute are always used. This is why these three bits have no effect

```
HLI_CI_LOCK_TABLE           : UDINT := 8;
```

Specifies whether the table used for online value modification during coupling is locked. This should always be set for table data types to which atomic access is not possible. Otherwise, the problem of simultaneous memory access must be observed.

```
HLI_CI_F_PERIODIC           : UDINT := 16;
```

If this flag is **not** set at MC_CamIn, the saved cam is passed exactly once on each rising edge at “MC_CamIn.Execute“. Travel over the cam is synchronised to the modulo passage of the master position. If the slave axis is not already at the start position of the CAM table, the slave axis is moved immediately to the cam start position at the moment of a rising edge at "MC_CamIn.Execute” and stays there until the next time the master axis passes zero point. From there, the cam profile is run through precisely once. If "MC_CamIn.Execute” is triggered again during the cycle, the cam is run through a further time. Without retriggering, the slave axis is decoupled from the master axis at the end of the CAM table. This means that the slave axis is stopped abruptly if it is not already at standstill.

If the cam was modelled for an endlessly rotating slave axis, in connection with the use of interpolation point interpolation (“MC_CamIn.TableFctType” = HLI_LINEAR or HLI_POLYNOM_3), it is necessary to specify in which direction modulo correction is to take place.

If the cam was modelled so that it passes through the modulo range of the slave axis in the positive direction, the bit represented by the global constant

```
HLI_CI_ENDLESS_POSITIVE     : UDINT := 32;
```

must be set in “MC_CamIn.StartMode”.

If the cam passes through the modulo range of the slave axis in the negative direction, the following bit represented by the global constant

```
HLI_CI_ENDLESS_NEGATIVE     : UDINT := 64;
```

must be set in “MC_CamIn.StartMode”.

3.17.2 Values at the “TableFctType” input

The manufacturer-specific "MC_CamIn.TableFctType" input specifies how to interpolate between two table lines:

```
HLI_STEP_DIRECT              : UDINT := 0;
```

Within the position interval of the master axis s_m_i, s_m_{i+1} the position entry of the slave axis s_s_i is output.

```
HLI_LINEAR                   : UDINT := 1;
```

Within the position interval of the master axis s_m_i, s_m_{i+1} a linear interpolation is inserted between the position entries of the slave axis s_s_i, s_s_{i+1}.

```
HLI_POLYNOM_3                : UDINT := 2;
```

Within the position interval of the master axis s_m_i, s_m_{i+1} a constant tangent cubic Bezier spline is interpolated between the four adjacent position entries or the slave axis s_s_{i-1}, s_s_i, s_s_{i+1}, s_s_{i+2}. Polynomial coefficient calculation takes place online.

```
HLI_POLYNOM_3_KOEFF          : UDINT := 3;
```

Within the position interval of the master axis s_m_i, s_m_{i+1} the position entry of the slave axis s_s_i (lines containing 4 values a_i) consists of the coefficients of a polynomial with the following form

$X = a_0 + a_1t + a_2t2 + a_3t3$.

Polynomial coefficient calculation therefore takes place offline.

```
LINE_POLY5                   : UDINT := 4;
```

Motion sections alternating between LINE and POLY5, starting with LINE.

```
POLY5_LINE                   : UDINT := 5;
```

Motion sections alternating between LINE and POLY5, starting with POLY5.

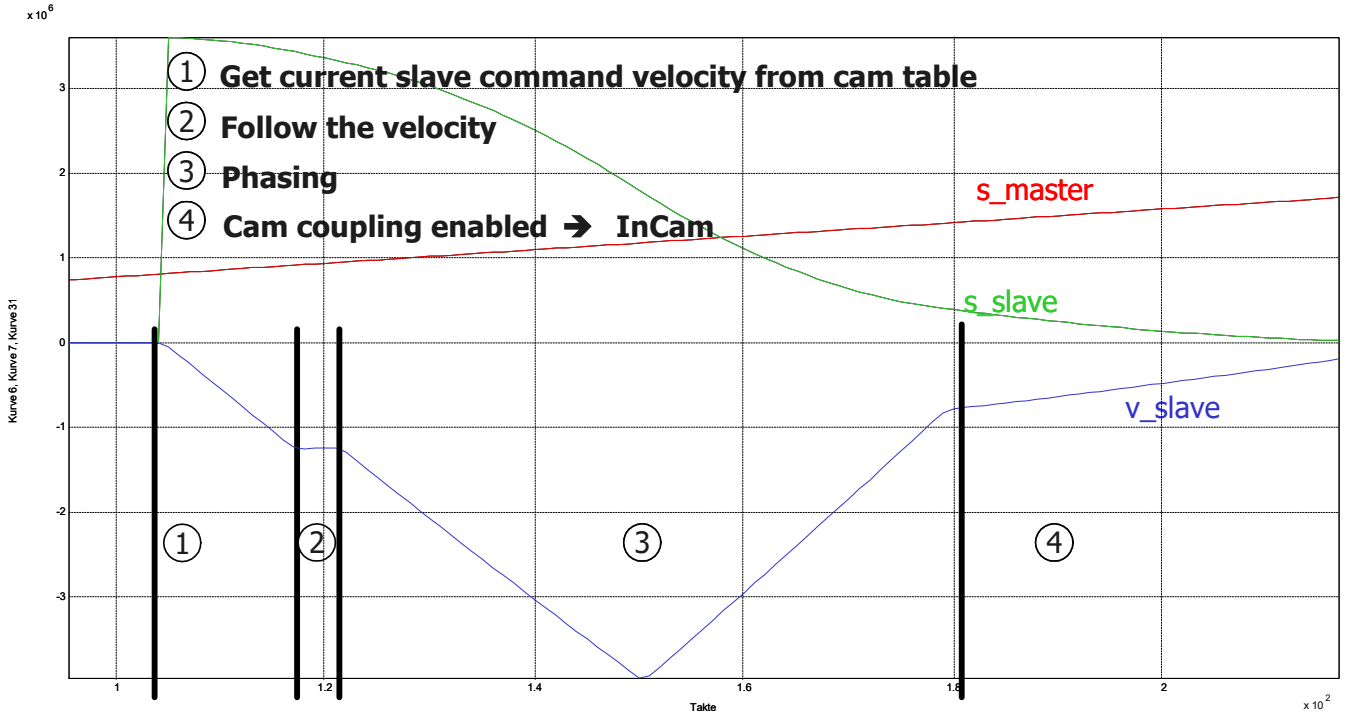
```
TABLE_DEFINED                : UDINT := 6;
```

The interpolation type is defined in the table.

3.17.3 Influence of axis parameters on the coupling process

What is referred to as a real amplitude ramp-in takes place both with a running master axis and at standstill.

With a running master axis this equates to the following mechanical analogy: Engaging a cam on a running master shaft. The coupling has slip until the cam to be driven along with it is at the right speed. The cam is then brought to the right position.)



The following parameters of the axis MDS influence this behaviour:

Axis parameters	Parameter-ID	Description
cam_gear.v_diff_percent	P-AXIS-00289	Velocity tolerance in phase 2. Default: 10
cam_gear.time_in_window	P-AXIS-00290	Min. duration of phase 2. Default: 8000[µs]
cam_gear.time_out_in_window	P-AXIS-00291	Max. duration of phase 2. Default: 1000000[µs]

Synchronisation of the speed is jerk-limited. The jerk value must be set in the axis MDS in the parameter **cam_gear.j_vel_sync** (P-AXIS-00377).

Values that correspond to a minimum ramp time of TA(cycle time of the IPO, e.g. 2 ms) and a maximum ramp time of 100s make sense as the jerk. The limit transition to non-jerk-limited speed realisation lies at the minimum ramp time of TA.

In the case of a value of 0, the maximum ramp time from the axis MDS is used to determine the jerk.

Notice

If the slave axis is a linear axis, make sure that the slave axis does not cross the software limit switch when the position is synchronised.

Notice

In the coupled state, the coupling is "holy", i.e. axis-specific feed hold, axis-specific override CANNOT act on the cam slave because this would be contradictory to the coupling specification.

The behaviour at emergency stop is implemented as follows: The MCE is only signalled about an emergency stop situation via the drive interface. In the event of an emergency stop, the coupling is released and decelerated at the current limit which is entered in the axis MDS parameter **getriebe[0].lslope_profil.a_grenz_stufe_1** (P-AXIS-00005). This default behaviour in the event of an emergency stop can be deactivated by using the parameter **cam_gear.keep_coupling_on_lr_error** (P-AXIS-00308). Then coupling is ensured even in the event of Emergency Stop as long as the drive still follows the setpoints.

In conjunction with the safety function which triggers an unconditional feedhold [HLI//Watchdog with effect on feedhold [▶ 80]] in the event of a watchdog trigger failure, the MC response to a watchdog trigger failure is to cancel the axis coupling and decelerate the axes at the current brake entered in the associated axis MDS parameter **getriebe[0].lslope_profil.a_grenz_stufe_1** (P-AXIS-00005).

This default behaviour in the event of a watchdog trigger failure by the PLC can be deactivated by the parameter **cam_gear.keep_coupling_on_fe_drop** (P-AXIS-00309). This safeguards the coupling even in the event of a watchdog trigger failure.

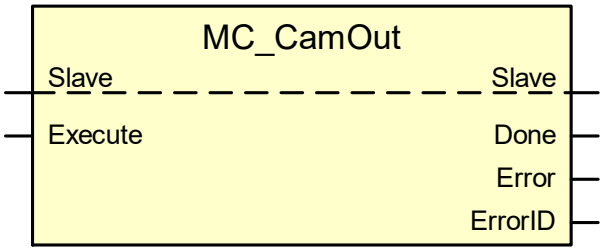
Automatic phasing is also jerk-limited after velocity synchronisation is reached. Consequently, the following parameters must be set in the slave:

- **cam_gear.v_phasing** (P-AXIS-00305) Velocity of the superimposed motion for phasing
- **cam_gear.a_phasing** (P-AXIS-00306) Acceleration of the superimposed motion for phasing
- **cam_gear.d_phasing** (P-AXIS-00307) Deceleration of the superimposed motion for phasing
- **cam_gear.j_phasing** (P-AXIS-00376) Jerk of the superimposed motion for phasing

3.18 MC_CamOut

Commanding of an MC_CamOut FB releases coupling of the slave axis to the master axis that has been specified via a cam. The current velocity of the slave is retained (endless motion).

Blockdiagramm



Parameters of the FB

VAR_IN_OUT		
Variable name	**Data type**	**Description**
Slave	AXIS_REF	Axis reference of the slave axis

VAR_INPUT		
Variable name	**Data type**	**Description**
Execute	BOOL	Starts to disengage the coupling via a cam on a rising edge.

VAR_OUTPUT		
Variable name	**Data type**	**Description**
Done	BOOL	Disengagement completed.
Error	BOOL	Is TRUE when an error occurs in the FB.
ErrorID	WORD	Error code.

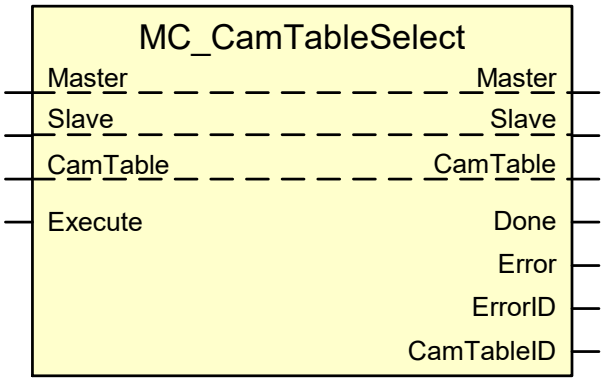
Behaviour of the FB

- Release of coupling of the slave axis to the master axis can only be commanded if the slave axis is in the Synchronized Motion state. If this is not the case, the "error" output is set to FALSE and "error_id" indicates an error value that designates the state the axis is currently in.
- Mention is made here once again of the (idiosyncratic) definition of the PLCopen standard to the effect that the commanded slave axis remains at its momentary speed after cancellation of coupling. Therefore, it does not necessarily stop.
- The function block is retriggerable. Therefore, it can send a command again in the active state.

3.19 MC_CamTableSelect

The MC_CamTableSelect FB is used to determine the CAM table ID by its filename. This can then be used as the input for the MC_CamIn function block.

Blockdiagramm



Parameters of the FB

VAR_IN_OUT		
Variable name	**Data type**	**Description**
Master	AXIS_REF	Axis reference of the master axis.
Slave	AXIS_REF	Axis reference of the slave axis
CamTable	MC_CAM_REF	This structure contains the filename of the CAM table.

VAR_INPUT		
Variable name	**Data type**	**Description**
Execute	BOOL	The CAM table identifier is supplied on the rising edge.

VAR_OUTPUT		
Variable name	**Data type**	**Description**
Done	BOOL	The CAM table identifier was supplied.
Error	BOOL	Is TRUE when an error occurs in the FB.
ErrorID	WORD	Error code.
CamTableID	MC_CAM_ID	CAM table identifier

Behaviour of the FB

- If the filename and therefore the CAM table ID are assigned in the “CamTable” input, the filename has a higher priority, i.e. the associated CAM table ID is determined.
- Commanding is possible at any time regardless of the state of the master or slave axis.

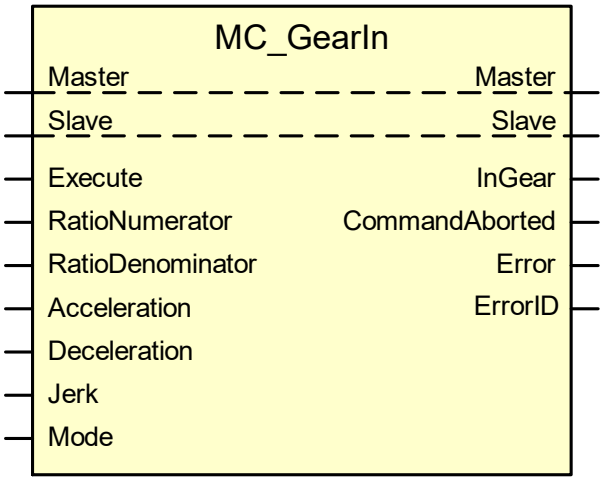
3.20 MC_GearIn

The MC_GearIn FB commands gear coupling with a gear ratio. The gear ratio defines the speed ratio between the master and slave axes. Synchronisation to velocity is jerk-limited. The jerk value must be specified as an input at the FB.

If the optional parameters “Acceleration, “Deceleration” and “Jerk” are not specified or set to <= 0, the dynamic values are taken from the corresponding axis list.

The slave axis can be linked either to master command values or to actual master values. Selection is made in the "Mode" input parameter.

Block diagram



FB parameters

VAR_IN_OUT		
Variable name	**Data type**	**Description**
Master	AXIS_REF	Axis reference of the master axis.
Slave	AXIS_REF	Axis reference of the slave axis

VAR_INPUT			
Variable name	**Data type**	**Description**	
Execute	BOOL	Starts the command on the rising edge.	
RatioNumerator	DINT	Gear ratio numerator	
RatioDenominator	DINT	Gear ratio denominator	
Acceleration	LREAL	Acceleration value. This value must be always be specified positive. Value range [0.0, 2147483647.0] Default unit [1mm/s^2 or 1°/s^2]	
Deceleration	LREAL	Deceleration value. This value must be always be specified positive. Value range [0.0, 2147483647.0] Default unit [1mm/s^2 or 1°/s^2]	
Jerk	LREAL	Jerk value. This value must be always be specified positive. Value range [0.0, 2147483647.0] Default unit [1mm/s^3 or 1°/s^3]	
Mode	UDINT	Possible values	
		0: (16#0)	Type of coupling between master axis and slave: Coupling on the command value side.
		128: (16#80)	Type of coupling between master axis and slave: Coupling on the actual value side.

VAR_OUTPUT		
Variable name	**Data type**	**Description**
InGear	BOOL	Commanded gearing completed.
Error	BOOL	Is TRUE if an error occurs in the FB.
ErrorID	WORD	Error identifier
CommandAborted	BOOL	TRUE indicates that the command was aborted by another command.

Behaviour of the FB:

- The command for gear coupling can only be issued only when the slave axis is in one of the following states:
- Standstill
- Continuous Motion
- Discrete Motion
- Synchronised Motion

If this is not the case, the “Error” output is set to FALSE and “ErrorID” indicates an error value that designates the state the axis is currently in.

- The value range of the “Acceleration”, “Deceleration” and “Jerk” input variables is monitored and, if it is exceeded, an error number referring to the variable is output at the “ErrorID” output and “Error” is set to TRUE. The “RatioDenominator” input variable is the denominator of the gear ratio and this is why it must be unequal to 0. In the event of the “RatioDenominator” **= 0** error, the error P-ERR-44040 (ERR_PO_AX_RATIO_DENOM_ZERO) is output at the “ErrorID” output and "Error“ becomes TRUE.

- Values for jerk are practical when they correspond to a minimum ramp time of T_A (cycle time of the IPO, e.g. 2 ms) and a maximum ramp time of 100s. The limit transition to non-jerk-limited velocity control lies at the minimum ramp time of T_A. With a value of 0, the maximum ramp time from the axis MDS is used to determine the jerk.
- The function block is retriggerable. Therefore, it can send a command again in the active state.
- For an axis to work as a master axis and for it to transfer the necessary synchronisation information to all axes (slaves) cyclically, the **cam_gear.is_master** parameter (P-AXIS-00288) must be set to 1 for such a master axis in the axis parameter configuration. An axis configured as master can be simultaneously coupled to another master axis as slave.
- The "Mode" input parameter can be used to set whether the master/slave coupling should use the actual values or the command values of the master.

Notice

The MC_GearIn FB can be used for linear and rotary slave axes. With linear slave axes, software limit switch monitoring is active after “Homing” is completed. For deceleration on approaching the software limit, the current limit ramp is used and gear coupling is released.

Notice

In all other situations, gear coupling is "holy", i.e. axis-specific feedhold, axis-specific override **CANNOT** act on the cam slave because this would be contradictory to the coupling specification.

3.20.1 Influence of axis parameters on the coupling process

Every SAI axis can be operated as a slave axis. The standard axis parameters below must be set in a slave axis both for actual and setpoint coupling:

- cam_gear.v_diff_percent [▶ 80] (P-AXIS-00289)
- cam_gear.time_in_window (P-AXIS-00290)
- cam_gear.time_out_in_window (P-AXIS-00291)
- **cam_gear.v_max_slave** (P-AXIS-00303)
- **cam_gear.a_max_slave** (P-AXIS-00304)
- **getriebe[0].vb_min_null** (P-AXIS-00216)

Filtering the master actual values

With an actual value coupling, the parameters that refer to filtering master actual values must be set in the slave axis:

- **cam_gear.mv_type**: (P-AXIS-00300) Digital FIR filter on/off
- **cam_gear.mv_nbr_cycles**: (P-AXIS-00301) Number of values used to filter.

Dead time compensation for constant velocity during actual value coupling

- **cam_gear.delay_time**:n (P-AXIS-00302)

Dead time compensation ensures that constant speed and a coupling ratio of 1 always results in the same phase between the master and slave. Compensation involves errors in phases of speed that is not constant. Dead time should be determined by trial and error.

The behaviour at emergency stop is implemented as follows:

The MC is only signalled about an emergency stop situation by the drive interface. In the event of an emergency stop, the coupling is released and decelerated at the current limit which is entered in the axis MDS parameter **getriebe[0].lslope_profil.a_grenz_stufe_1** (P-AXIS-00005). This default behaviour in the event of an emergency stop can be deactivated by using the parameter **cam_gear.keep_coupling_on_lr_error** (P-AXIS-00308). Then coupling is ensured even in the event of an emergency stop as long as the drive still follows the setpoints.

In conjunction with the safety function which triggers an unconditional feedhold [HLI//Watchdog with effect on feedhold] in the event of a watchdog trigger failure, the MC response to a watchdog trigger failure is to cancel the axis coupling and decelerate the axes at the current brake entered in the associated axis MDS parameter **getriebe[0].lslope_profil.a_grenz_stufe_1** (P-AXIS-00005). This default behaviour in the event of an emergency stop can be deactivated by using the parameter **cam_gear.keep_coupling_on_fe_drop** (P-AXIS-00309). This safeguards the coupling even in the event of a watchdog trigger failure.

When automatic phasing is commanded (actual value or setpoint coupling), the following axis parameters must also be set in the slave:

- **cam_gear.v_phasing** (P-AXIS-00305)

Speed of the superimposed motion for phasing.

- **cam_gear.a_phasing** (P-AXIS-00306)

Acceleration of the superimposed motion for phasing.

- **cam_gear.d_phasing** (P-AXIS-00307)

Deceleration of the superimposed motion for phasing.

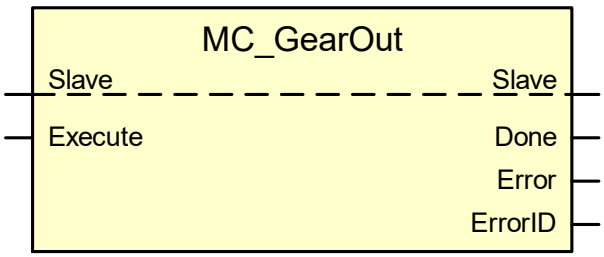
- **cam_gear.j_phasing** (P-AXIS-00376)

Jerk of the superimposed motion for phasing.

3.21 MC_GearOut

The MC_GearOut FB releases coupling of the slave axis to the master axis that has been specified via a gear ratio. The current velocity of the slave is retained (endless motion).

Blockdiagramm



Parameters of the FB

VAR_IN_OUT		
Variable name	**Data type**	**Description**
Slave	AXIS_REF	Axis reference of the slave axis

VAR_INPUT		
Variable name	**Data type**	**Description**
Execute	BOOL	Starts disengaging the velocity coupling on the rising edge.

VAR_OUTPUT		
Variable name	**Data type**	**Description**
Done	BOOL	The velocity coupling of the slave axis is disengaged.
Error	BOOL	Is TRUE when an error occurs in the FB.
ErrorID	WORD	Error code.

Behaviour of the FB

- Cancellation of gear coupling is commanded only when the axis is in the Synchronized Motion state. If this is not the case, the "Error" output is set to FALSE and "ErrorID" indicates an error value that designates the state the axis is currently in.
- Mention is made here once again of the (idiosyncratic) definition of the PLCopen standard to the effect that the commanded slave axis remains at its momentary speed after cancellation of coupling. Therefore, it does not necessarily stop.
- The function block is retriggerable, i.e. it can send a command when it is in the active state.

3.22 MC_Phasing

The MC_Phasing FB is used to achieve an offset of the slave axis with respect to the master axis. Accordingly, a phase offset of the master axis is specified from the viewpoint of the slave axis and the slave axis tries to eliminate this offset by accelerating or decelerating. The motion is always executed jerk-limited with the constant set input "Jerk”. This value is valid for both "Acceleration" and "Deceleration".

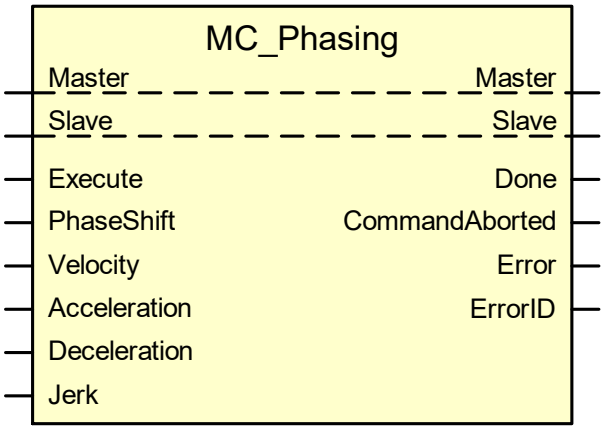
If the optional parameters “Acceleration, “Deceleration” and “Jerk” are not specified or set to <= 0, the dynamic values are taken from the corresponding axis list.

The mechanical analogy is to release the coupling of the master and slave axes for a limited period.

With **camming**, this FB results in a change in the "apparent" master position from the slave's viewpoint. With **gearing**, a phase offset is produced between the master and slave by commanding a superimposed motion in the slave. Therefore with gearing, MC_Phasing has the same effect as MC_MoveSuperImposed (to which it is actually converted in the control system).

The dynamic values: With camming, "Velocity", "Acceleration" and "Deceleration" refer to the change in the "apparent" master position from the viewpoint of the slave; but with gearing, they refer to the superimposed motion of the slave axis itself.

Block diagram



FB parameters

VAR_IN_OUT		
Variable name	**Data type**	**Description**
Master	AXIS_REF	Axis reference of the master axis.
Slave	AXIS_REF	Axis reference of the slave axis
VAR_INPUT		
Variable name	**Data type**	**Description**
Execute	BOOL	Starts the command on the rising edge.
PhaseShift	LREAL	Phase shift between master axis and slave axis, as “seen” by the slave axis. The calculated phase shift is transferred to the slave axis as the master axis position. Value range [-2147483648.0, 2147483647.0] Default unit [0.1µm or 10^{-4} °]
Velocity	LREAL	Maximum velocity value which may be reached when phasing is generated. This value must be always be specified positive. The velocity is not necessarily achieved. Value range [0.0, 2147483647.0] Default unit [1µm/s or 10^{-3}°/s]
Acceleration	LREAL	Maximum acceleration value. This value must be always be specified positive. Value range [0.0, 2147483647.0] Default unit [1mm/s^2 or 1°/s^2]
Deceleration	LREAL	Maximum deceleration value. This value must be always be specified positive. Value range [0.0, 2147483647.0] Default unit [1mm/s^2 or 1°/s^2]
Jerk	LREAL	Maximum jerk value. This value must be always be specified positive. Value range [0.0, 2147483647.0] Default unit [1mm/s^3 or 1°/s^3]
VAR_OUTPUT		
Variable name	**Data type**	**Description**
Done	BOOL	Commanded phase offset reached.
CommandAborted	BOOL	TRUE indicates that the command was aborted by another command.
Error	BOOL	Is TRUE if an error occurs in the FB.
ErrorID	WORD	Error identifier

Behaviour of the FB:

- Phase offset can only be commanded when the slave axis is in the “Synchronised Motion” state.
- If this is not the case, the “Error” output is set to FALSE and “ErrorID” indicates an error value that designates the state the axis is currently in.
- The value range of the “PhaseShift”, “Velocity”, “Acceleration”, “Deceleration” and “Jerk” input variables is monitored and, if it is exceeded, an error number referring to the variable is output at the “ErrorID” output and “Error” is set to TRUE.
- Values for jerk are practical when they correspond to a minimum ramp time of T_A (cycle time of the IPO, e.g. 2 ms) and a maximum ramp time of 100s. The limit transition to non-jerk-limited velocity control lies at the minimum ramp time of T_A. With a value of 0, the maximum ramp time from the axis MDS is used to determine the jerk.
- The function block is retriggerable, i.e. it can send a command when it is in the active state.

3.23 MC_ReadParameter

This function block corresponds to the function block described in PLCopen for reading parameters.

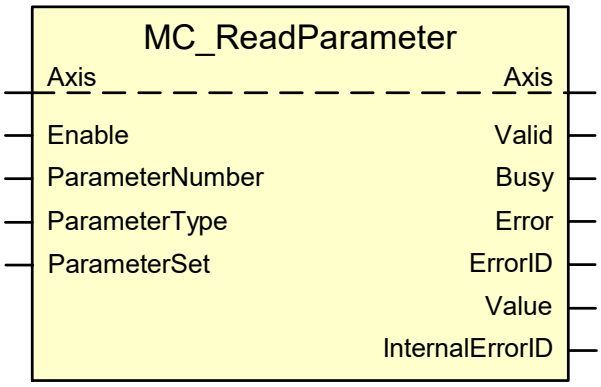
Notice

This function block is not available in TwinCAT systems..

Notice

Only parameters from the axis MDS are read. Parameters that are only defined in the default axis MDS can **not** be read.

Block diagram



FB parameters

VAR_IN_OUT		
Variable name	**Data type**	**Description**
Axis	AXIS_REF	Axis reference
VAR_INPUT		
Variable name	**Data type**	**Description**
Enable	BOOL	Value must be TRUE to start continuous read.
ParameterNumber	STRING	Identifier of the parameter whose value is to be read. See Note 1 [▶ 93].
ParameterType	STRING	Type of parameter to be read. Currently: ACHS_MDS
ParameterSet	UINT	Number of the parameter set from which the parameter value is to be read.
VAR_OUTPUT		
Variable name	**Data type**	**Description**
Valid	BOOL	Is TRUE when the parameter was successfully read.
Busy	BOOL	Is TRUE if the FB is still active.
Error	BOOL	Indicates whether an error occurred in the FB.
ErrorID	WORD	Error identifier
Value	STRING	Parameter value from the original list.
InternalErrorID	UDINT	Internal error identification of the ISG-MCP.

Behaviour of the FB:

- The behaviour of this FB corresponds to that of the MCV_ReadParamValue function block.

Example

Parameterising the Motion Controller by ASCII file lists

```
kenngr.swe_pos.
getriebe[0].vb_not_referenced
```

Example

Parameterising the Motion Controller by XML file lists

If the Motion Controller is parameterised using XML files, access to the parameter must be specified by an XPath value at the "ParameterNumber" input.

```
/AX_MDS/Head/P-AXIS-00297/@Value
/AX_MDS/Settings/P-AXIS-00015/@Value
/AX_MDS/Settings/Homing/P-AXIS-00299/@Value
/AX_MDS/Drive/Generic/address/@Value
```

or for parameters with index:

```
/AX_MDS/Gear/GearIndex[@Value="0"]/P-AXIS-00234/@Value
```

or simply use the parameter name, e.g.:

```
P-AXIS-00299
```

3.24 MC_WriteParameter

This function block (FB) corresponds to the function block for writing parameters described in PLCopen.

Notice

This function block is not available in TwinCAT systems..

3 steps are carried out when writing a parameter with this FB:

1. Writing the parameter value into the work list
2. Activating the work list (parameters take effect in the ISG-MCP)
3. Storing the work list back to the original list

This means that a parameter value that is written with this FB also takes effect immediately and is permanently stored.

This FB is not suitable for consistently writing several parameter values. In this case, use the MCV_ChgParamValue, MCV_ActivateParamChg and MCV_SaveParamChg function blocks.

Notice

Only parameters that can be read from the axis MDS are written. Parameters that are only defined in the default axis MDS cannot be written.

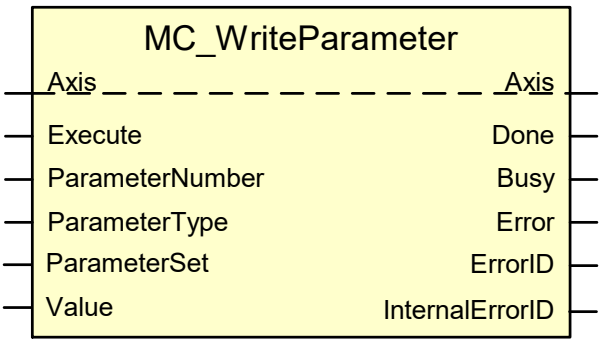
Notice

During the write operation, the axis must be at standstill and motion commands must not have been or must not be started for the axis. The axis may only be moved again after the parameter is written and the new value is active in the controller (output “done“ TRUE).

Notice

To modify a parameter, this function block requires an original list that can be read in by the controller without warnings and error messages.

Block diagram



FB parameters

VAR_IN_OUT		
Variable name	**Data type**	**Description**
Axis	AXIS_REF	Axis reference
VAR_INPUT		
Variable name	**Data type**	**Description**
Execute	BOOL	Must have the value TRUE in order to write the parameter value.
ParameterNumber	STRING	Identifier of the parameter to be written. See Note 1. [▶ 96]
ParameterType	STRING	Type of parameter to be written. Currently: ACHS_MDS
ParameterSet	UINT	Number of the parameter set from which the parameter value is to be read.
Value	STRING	New value of the parameter.
VAR_OUTPUT		
Variable name	**Data type**	**Description**
Done	BOOL	Is TRUE when the parameter was successfully written.
Busy	BOOL	Is TRUE if the FB is still active.
Error	BOOL	Indicates whether an error occurred in the FB.
ErrorID	WORD	Error identifier
InternalErrorID	UDINT	Internal error identification of the ISG-MCP.

Behaviour of the FB:

A frequent cause of ISG-MCP errors are absent write privileges or a write-protected work list or original list because, in these cases, a work list is not created or an original list cannot be replaced.

Example

Parameterising the Motion Controller by ASCII file lists

If the Motion Controller is parameterised using ASCII file lists, the name of the parameter is specified as the value at the "ParameterNumber" input.

```
kenngr.swe_pos
getriebe[0].vb_not_referenced
```

Example

Parameterising the Motion Controller by XML file lists

If the Motion Controller is parameterised using XML files, access to the parameter must be specified by an XPath value at the "ParameterNumber" input:

```
/AX_MDS/Head/P-AXIS-00297/@Value
/AX_MDS/Settings/P-AXIS-00015/@Value
/AX_MDS/Settings/Homing/P-AXIS-00299/@Value
/AX_MDS/Drive/Generic/address/@Value
```

or for parameters with index:

```
/AX_MDS/Gear/GearIndex[@Value="0"]/P-AXIS-00234/@Value
```

or simply use the parameter names.

{» e.g. P-AXIS-00299{Martin Pfizenmayer 21.06.2018 10:49:41: Ist das so richtig ? wie muss denn der Name drin stehen?}}

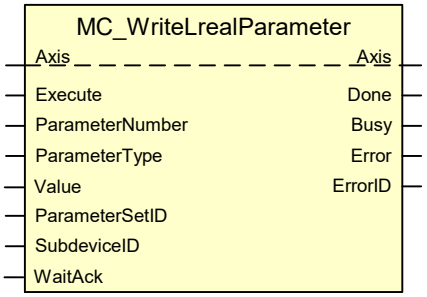
3.25 MC_WriteLrealParameter

This function block is used to change the values of control system parameters. You can change the parameters of various control system components.

Notice

Only drive parameters can be changed (ParameterType:= 'Drive'). If you try to change other parameter types, it will result in an error message.

Block diagram



FB parameters

VAR_IN_OUT		
Variable name	**Data type**	**Description**
Axis	AXIS_REF	Axis reference

VAR_INPUT		
Variable name	**Data type**	**Description**
Execute	BOOL	Must have the value TRUE in order to write the parameter value.
ParameterNumber	STRING	Identifier of the parameter to be written.
ParameterType	STRING	Kind of parameter to be written. Possible values: 'Drive'
Value	LREAL	New value of the parameter.
ParameterSetID	INT	Identifier indicating the parameter set in which the value is to be written. -1 at this input means that no parameter set was selected. The input is defaulted with this value.
SubdeviceID	INT	Identifier indicating the device for which the value is to be written. -1 at this input means that no unit was selected. The input is defaulted with this value.
WaitAck	BOOL	Input which influences the reaction of the motion controller to the command acknowledgement. TRUE – the motion controller stops the interpolation with this command. FALSE – the interpolation of the motion controller is not stopped when the command is received.

VAR_OUTPUT		
Variable name	**Data type**	**Description**
Done	BOOL	Is TRUE when the parameter was successfully written.
Busy	BOOL	Is TRUE if the FB is still active.
Error	BOOL	Indicates whether an error occurred in the FB.
ErrorID	WORD	Error identifier
InternalErrorID	UDINT	Internal error identification of the ISG-MCP.

Behaviour of the FB:

- A frequent cause of ISG-MCP errors are absent write privileges or a write-protected work list or original list because, in these cases, a work list is not created or an original list cannot be replaced.

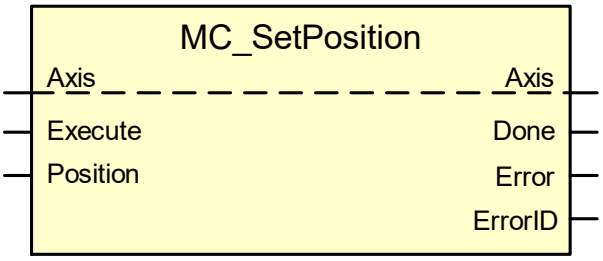
3.26 MC_SetPosition

The function block shifts the coordinate system of an axis by changing both the command position and the actual position to the same position value which is applied to the “Position” input. The axis is not moved during this operation.

To permit the block to execute its function, the parameter kenngr.set_refpos_mode (P-AXIS-00278) must be assigned in the associated axis parameter list as follows:

kenngr.set_refpos_mode **PLC**

Blockdiagramm



Parameters of the FB

VAR_IN_OUT		
Variable name	**Data type**	**Description**
axis	AXIS_REF	Axis reference

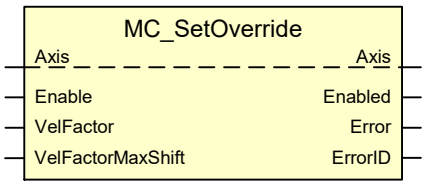
VAR_INPUT		
Variable name	**Data type**	**Description**
Execute	BOOL	Starts the command on the rising edge.
Position	LREAL	Position which is to be adopted as command and actual position if a rising edge is detected at the "Execute" input. Value range [-2147483648.0, 2147483647.0] Default unit [0.1 µm or 10^{-4} °].

VAR_OUTPUT		
Variable name	**Data type**	**Description**
Done	BOOL	Is TRUE when the position was successfully adopted.
Error	BOOL	Indicates whether an error has occurred in the FB.
ErrorID	WORD	Error code.

3.27 MC_SetOverride

The FB MC_SetOverride specifies a weighting factor that acts on the commanded velocity of the axis.

Block diagram



FB parameters

VAR_IN_OUT		
Variable name	**Data type**	**Description**
Axis	AXIS_REF	Axis reference
VAR_INPUT		
Variable name	**Data type**	**Description**
Enable	BOOL	If this value is TRUE, the weighting factor is applied to the commanded velocity.
VelFactor	LREAL	Weighting factor of the velocity Value range [0.0 to P-AXIS-00109 / 1000] See use of the “VelFactor” input.
VelFactorMaxShift	LREAL	Maximum change of the weighting factor. Value range [0.0 to P-AXIS-00109 / 1000] See use of the “VelFactorMaxShift” input.
VAR_OUTPUT		
Variable name	**Data type**	**Description**
Enabled	BOOL	If this value is TRUE, the value at the “VelFactor” input is continuously applied to the axis.
Error	BOOL	Is TRUE if an error occurs in the FB.
ErrorID	WORD	Error identifier

Behaviour of the FB:

The FB acts on all axes which have a spindle or single-axis interpolator due to the parameterisation. This can be achieved by entering the axis as

1. Spindle
2. a linear or rotary axis with activated P-AXIS-00250
3. A path axis with activated P-AXIS-00457 (where the effect of parameter P-AXIS-00491 must be considered)

in the configuration. On axes below 3 [▷ 102], the weighting factor only acts if the axis is moved by a PLCopen command.

- The weighting factor "VelFactor" can always be specified whenever the ISG-MCP was initialised correctly.
- The value specified at the “VelFactor” input is cyclically applied to the axis as long as the “Enable” input is TRUE. If “Enable” changes from TRUE to FALSE, the value which was applied to the “VelFactor” input at the transition from “Enable” acts as the velocity factor.

⚠ WARNING

This function does not act on axes that are in the "SYNCHRONIZED_MOTION" state.

Note:

In a typical packaging application environment, there are frequently intermeshing machine components that are controlled by means of cam/gear coupling.

A loss of coupling synchronicity here may lead to very severe machine damage. This is why the cam/gear coupling in the implementation is "holy", i.e. axis-specific feedhold, axis-specific override **CANNOT** act on the cam/gear slave because this would be contradictory to the coupling specification.

Example

Use of the “VelFactor” input

- The default value of the “VelFactor” input is 1.0.
- P-AXIS-00109 is the maximum permitted override value. Assuming that the maximum allowed override value in an application may not exceed 100%, the parameter P-AXIS-00109 is assigned the value 1000 since the unit 0.1% is specified for this parameter. Therefore, the range for the ‘VelFactor’ is [0.0 to 1.0].
- If the value at the "VelFactor" input is greater than P-AXIS-00109/1000, it is limited to P-AXIS-00109/1000 (typical upper limit value is 200%).

Example

Use of the “VelFactorMaxShift” input

- The “VelFactorMaxShift“ input is defaulted to the value 1.0.
- If the value at the “VelFactor” input changes, the current effective velocity factor sent to the motion controller is increased or decreased by the maximum value at the “VelFactorMaxShift” input each time the function block is invoked.
- The weighting factor sent to the motion controlled is increased or deceased until the same velocity weighting factor as specified at the “VelFactor” input acts on the motion controller. Therefore, the function block continues to be invoked until its “Enabled” output indicates TRUE again after “VelFactor” is changed.

3.28 MC_Halt

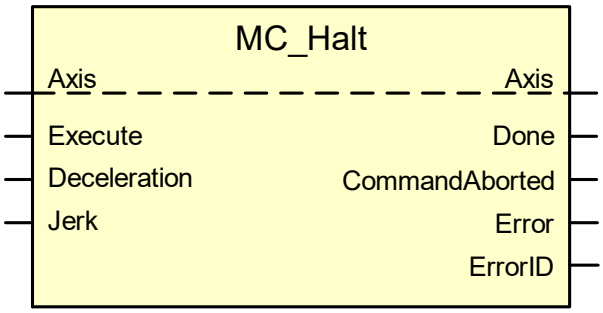
The MC_Halt FB leads to a controlled motion stop. The motion stop is always jerk-limited with the constant set input "Jerk" to build up the deceleration rate. It aborts every ongoing command by other motion FBs.

If the optional parameters "Deceleration" and "Jerk" are not specified or set to <= 0, the dynamic values are taken from the corresponding axis list.

Contrary to the FB MC_Stop, a commanded FB MC_Halt may be aborted by commanding another motion FB. The "CommandAborted" output is then set to TRUE.

The MC_Halt FB sets the "Done" output as soon as the axis is at standstill.

Block diagram



FB parameters

VAR_IN_OUT		
Variable name	**Data type**	**Description**
Axis	AXIS_REF	Axis reference
VAR_INPUT		
Variable name	**Data type**	**Description**
Execute	BOOL	Starts the command on the rising edge.
Deceleration	LREAL	Deceleration value. This value must be always be specified positive. Value range [0.0, 2147483647.0] Default unit [1mm/s^2 or 1°/s^2]
Jerk	LREAL	Jerk value. This value must be always be specified positive. Value range [0.0, 2147483647.0] Default unit [1mm/s^3 or 1°/s^3]
VAR_OUTPUT		
Variable name	**Data type**	**Description**
Done	BOOL	0 velocity was reached.
CommandAborted	BOOL	TRUE indicates that the command was aborted by another command.
Error	BOOL	Is TRUE if an error occurs in the FB.
ErrorID	WORD	Error identifier

Behaviour of the FB:

- The “Deceleration” and “Jerk” input variables are monitored only for their value ranges in the FB. If the values are exceeded, an error number referring to the variable is output at the “ErrorID” output and “Error” is set to TRUE. Moreover, the deceleration value is limited in the motion controller. If a deceleration greater than **getriebe[0].dynamik.a_max** (P-AXIS-00008) is commanded, the motion is still executed. However, the values are limited to the parameterised values and a warning is issued.
- Practical values for jerk are between the minimum jerk of deceleration/100 s and the maximum jerk of deceleration/T_A. The limit transition to non-jerk-limited velocity control lies at the minimum ramp time of T_A. When the value is 0, the maximum ramp time in P-AXIS-00195 to P-AXIS-00198 is used to determine jerk.

4 PLC library “McpPLCopenP1” – ISG extensions

The function blocks described in this chapter provide additional functions to the standard PLCopen function blocks. They are also contained in the library where the PLCopen function blocks “Function blocks for motion control” are located.

4.1 Overview of the additional ISG function blocks (FB)

The table below provides an overview of the availability of the FBs additionally provided by the ISG.

Function block	PLC systems		
	KW	3S	TwinCAT
MCV_P1_Platform [▶ 110]	X	X	X
MCV_Axis [▶ 111]	X	X	X
MCV_CamSwitch [▶ 111]	SIMU KUKA SERCOS	SIMU - SERCOS	SIMU
MCV_TorqueLimit [▶ 116]	- KUKA SERCOS	- - SERCOS	SIMU
MCV_SetFollowUp [▶ 120]	SIMU KUKA -	SIMU - -	SIMU
MCV_SetDriveMode [▶ 121]	- KUKA SERCOS	- - SERCOS	SIMU
MCV_SetTorqueValue [▶ 123]	- KUKA SERCOS	- - SERCOS	SIMU
MCV_PosLagMonitor [▶ 124]	- KUKA -	- - -	SIMU
MCV_Home [▶ 125]	- - -	- - -	SIMU
MCV_DisableFeedbackMon [▶ 127]	- KUKA -	- - -	SIMU
MCV_ChgParamSet [▶ 128]	SIMU KUKA SERCOS	SIMU [1]) - SERCOS [1])	SIMU
MCV_ReqParamSetId [▶ 131]	SIMU KUKA SERCOS	SIMU [1]) - SERCOS [1])	SIMU

MCV_ChgParamValue [▶ 133]	SIMU KUKA SERCOS	SIMU [1]) - SERCOS [1])	SIMU
MCV_ActivateParamChg [▶ 135]	SIMU KUKA SERCOS	SIMU [1]) - SERCOS [1])	SIMU
MCV_SaveParamChg [▶ 138]	SIMU KUKA SERCOS	SIMU [1]) - SERCOS [1])	SIMU
MCV_ReadParamValue [▶ 140]	SIMU KUKA SERCOS	SIMU [1]) - SERCOS [1])	SIMU
MCV_DiscardParamChg [▶ 143]	SIMU KUKA SERCOS	SIMU [1]) - SERCOS [1])	SIMU
MCV_UnHome [▶ 145]	- KUKA -	- - -	
MCV_DeactivateAxis [▶ 146]	- KUKA -	- - -	
MCV_Decouple [▶ 146]	- KUKA -	- - -	
MCV_Couple [▶ 149]	- KUKA -	- - -	
MCV_ConveyorControl [▶ 151][1])	SIMU - SERCOS	SIMU - SERCOS	SIMU - SERCOS
MCV_SetReducedFilter [▶ 153]	SIMU KUKA SERCOS	SIMU - SERCOS	SIMU - SERCOS
MCV_SuppressSoftLimSw [▶ 153]	SIMU KUKA SERCOS	SIMU - SERCOS	SIMU - SERCOS
MCV_TorqueFreeStop [▶ 155]	- KUKA SERCOS	- - SERCOS	-
MCV_InitPosition [▶ 156]	Available for all drive types	Available for all drive types	Available for all drive types

[1]) Availability is dependent on the MCE used.

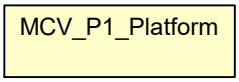
4.2 MCV_P1_Platform

This function block ensures that the motion controller is signalled during the initialisation phase that the PLC is present and that single axis commands are sent to the motion controlled. After this operation, its function is to invoke the instances of the function block MCV_Axis [▶ 111] cyclically to enable communication between the motion controller and the PLC relating to PLCopen Part 1 commands.

Notice

Only 1 instance of this block may occur in a PLC application. It must be invoked in every PLC cycle.

Blockdiagramm



Parameters of the FB

The functions block has no inputs or outputs..

4.3 MCV_Axis

The function block monitors the communication section between the motion controller and single axes. It adopts information from the motion controller to the PLC such as error messages or command acknowledgements etc.

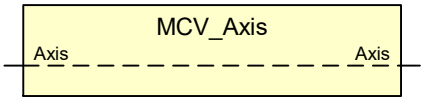
Notice

In a PLC application this function block must be invoked in every PLC cycle, where only 1 instance of the function block may exist for each single axis.

Notice

To simplify the function for the user, the library contains the function block MCV_P1_Platform [▶ 110] by instancing and invoking the required instances of the FB MCV_Axis.

Blockdiagramm



Parameters of the FB

VAR_IN_OUT		
Variable name	**Data type**	**Description**
axis	AXIS_REF	Axis reference

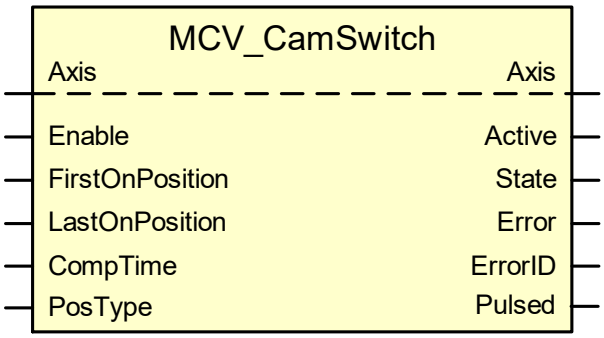
4.4 MCV_CamSwitch

The MCV_CamSwitch FB detects the output of a cam-controlled switch that is assigned to an axis. The cam range is freely definable.

The FB-internal calculations are based on a modulo coordinate system with the value range [0, 3600000] 0.1 µm or 10^{-4} °.

The prerequisite must also be fulfilled that no more than half the modulo value range is traversed by the axis in each PLC cycle.

Block diagram



FB parameters

VAR_IN_OUT		
Variable name	**Data type**	**Description**
Axis	AXIS_REF	Axis reference

VAR_INPUT		
Variable name	**Data type**	**Description**
Enable	BOOL	Activates the cam-driven switch.
FirstOnPosition	LREAL	Position of the axis at which the output of the cam-driven switch changes from FALSE to TRUE, if the axis moves in a positive direction. Default unit [0.1µm or 10^{-4} °]
LastOnPosition	LREAL	Position of the axis at which the output of the cam-driven switch changes from TRUE to FALSE, if the axis moves in a positive direction. Default unit [0.1µm or 10^{-4} °]
CompTime	LREAL	Compensation time by which the switch-on phase of the switch is advanced or delayed. Default unit [1ms]
PosType	INT	Selection whether command position or actual position of the axis should be used for comparison with the input variables and for calculation of the switch-on/switch-off position that is changed by the compensation time. **0**: Command position of the axis (ACTIVE_POS) **1**: Actual position of the axis (CURRENT_POS)

VAR_OUTPUT		
Variable name	**Data type**	**Description**
Active	BOOL	Indicates that the cam-controlled switch is active.
State	BOOL	Output of the cam-driven switch. The output is TRUE when 1. the value of the axis position lies between the “FirstOnPosition” and “LastOnPosition”. 2. The axis position at time t_0 **before** and the next t_1 **behind** the range between the “FirstOnPosition“ and the “LastOnPosition“. The signal is then TRUE for at least 1 PLC cycle.
Error	BOOL	Is TRUE if an error occurs in the FB.
ErrorID	WORD	Error identifier
Pulsed	BOOL	For visualisation purposes. If the "State" output is TRUE, the "Pulsed" output is also TRUE for 1 s.

Behaviour of the FB:

- Input values for “FirstOnPosition“ and “LastOnPosition“ which are less than the lower limit of the modulo coordinate system are set to the lower limit. Input values which are greater than the upper limit are set to the upper limit. This is done internally in the FB without acknowledgement.
- Specification of a compensation time leads in the FB to modification of the switching positions defined by “ FirstOnPosition” and “LastOnPosition”. If values should result for the modified switching positions that are outside the value range of the modulo coordinate system, they are converted to values within the range. This is done internally in the FB without acknowledgement.
- Modification of the switching positions by specifying a compensation time is executed directly in the PLC cycle by transferring them to the “CompTime“ input and takes full effect immediately.
- Specification of the compensation time "CompTime" acts equally on the “FirstOnPosition” and “LastOnPosition”. Positive values for “CompTime“ lead to a situation where the “State” output changes after a delay to reach a switching position. Negative values lead to a situation where the “State” output changes before the switching position is reached.

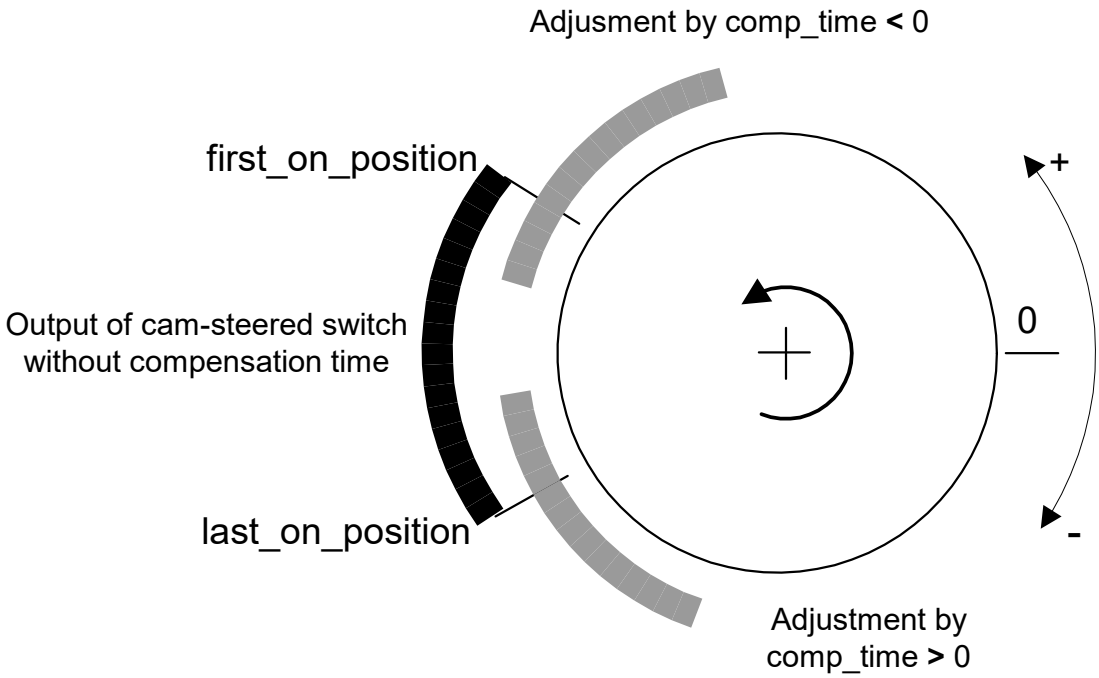


Fig. 17: Impact of the compensation time on the switching times

- Behaviour of the “State” output of the cam-controlled switch and the positions that are transferred to the “FirstOnPosition” and “LastOnPosition” inputs. The axis moves in a positive direction from the low limit of the modulo coordinate system to the high limit.

Case 1: “FirstOnPosition“ < “LastOnPosition“

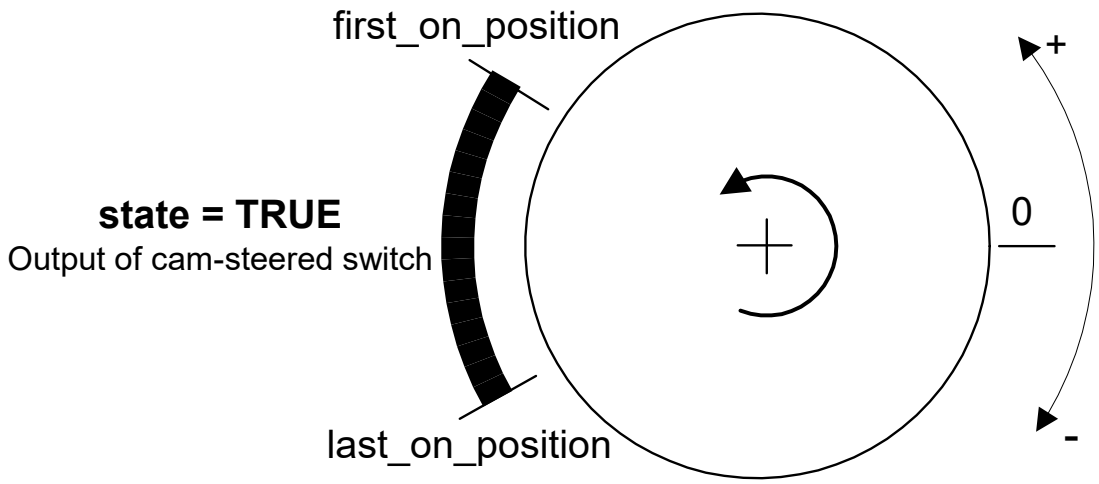


Fig. 18: Behaviour of the "State" output when “FirstOnPosition” < “LastOnPosition”

Case 2: “FirstOnPosition“ > “LastOnPosition“

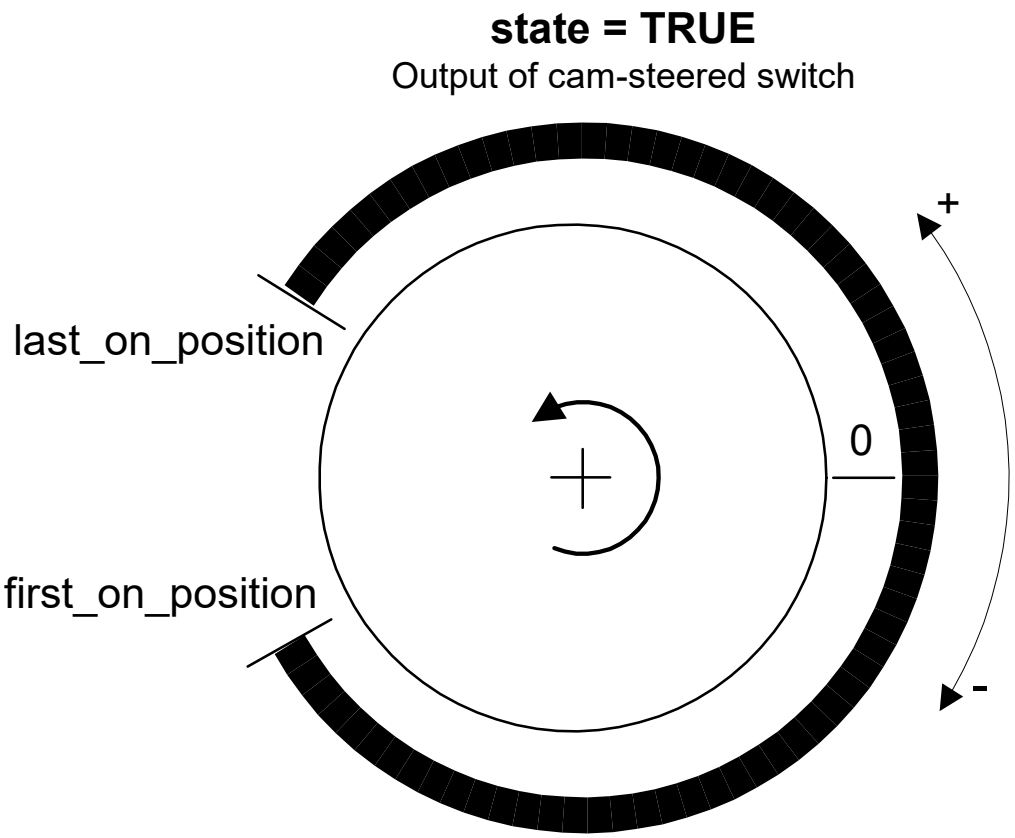


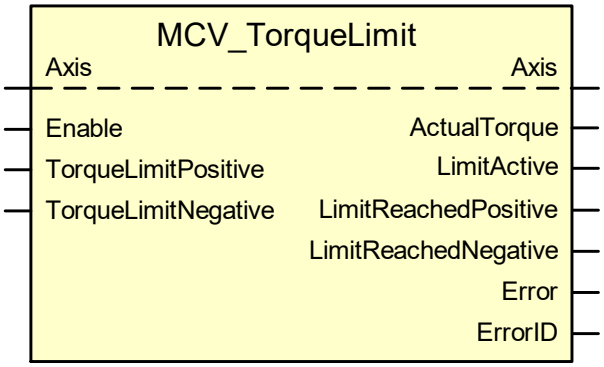
Fig. 19: Behaviour of the "State" output when “FirstOnPosition” > “LastOnPosition”

- The FB takes 2 special cases into account, which result from the fact that the PLC system is a discreet system. These are:
  - The axis position at the time t_n is before the switch-on position and the axis position at the time t_{n+1} is after the switch-off position. Therefore, the switching range is less than the distance traversed by the axis in one PLC cycle.
  - If the axis position is before the switch-on position at the time t_n, specification of a compensation time may result in a situation where the axis position at the time t_{n+1} is after the switch-off position due to modification of the switch-on points.
- In these special cases, the “State” output becomes **TRUE** for at least one PLC cycle although it was not detected within one revolution of the axis that the axis position was between switching positions.

4.5 MCV_TorqueLimit

The MCV_TorqueLimit FB limits the maximum torque of the drive for every motion direction independently.

Blockdiagramm



Parameters of the FB

VAR_IN_OUT		
Variable name	**Data type**	**Description**
axis	AXIS_REF	Axis reference

VAR_INPUT		
Variable name	**Data type**	**Description**
enable	BOOL	If "Enable“ is TRUE, the torque limit is active.
TorqueLimitPositive	INT	Maximum drive torque which may occur in the positive motion direction. Default unit [0.1%]
TorqueLimitNegative	INT	Maximum drive torque which may occur in the negative motion direction. Default unit [0.1%]

VAR_OUTPUT		
Variable name	**Data type**	**Description**
ActualTorque	INT	Currently effective drive torque. Default unit [0.1%]
LimitActive	BOOL	The torque limit is activated.
LimitReachedPositive	BOOL	The maximum torque for the positive motion direction is reached.
LimitReachedNegative	BOOL	The maximum torque for the negative motion direction is reached.
Error	BOOL	Indicates TRUE if an error occurs in the FB.
ErrorID	WORD	Error code.

Behaviour of the FB

- The torque limit can always be specified whenever the ISG-MCP has been initialised correctly.
- The torque limit must be specified separately for each motion direction of the axis. If no value is specified for the torque limit via the "TorqueLimitPos" or "TorqueLimitNeg" inputs, the FB specifies the default of 100%. Input values less than 0% and values greater than 130% are converted to 0% or 100% without any further message.
- The MVC_PosLagMonitor FB is used to deactivate the position lag error monitor.
- The "LimitReachedPos" and "LimitReachedNeg" outputs become TRUE whenever the currently active torque corresponds to at least 95% of the torque commanded by this FB.

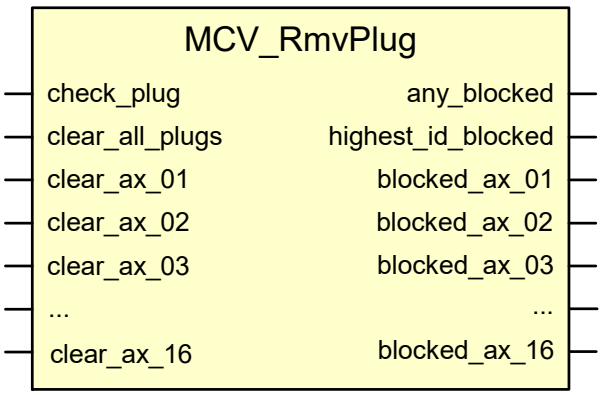
4.6 MCV_RmvPlug

The MCV_RmvPlug FB is used to check whether and which of the 16 axis interfaces are blocked by unacknowledged commands of the PLCopen FBs. After checking, this FB can be used to remove detected blockages.

Notice

This FB serves diagnostic purposes and must only be used during the development phase of a project.

Blockdiagramm



Parameters of the FB

VAR_INPUT		
Variable name	**Data type**	**Description**
check_plug	BOOL	Starts the check to detect which of the maximum 16 axis interfaces is blocked by a command which is not acknowledged by the PLC.
clear_all_plugs	BOOL	Acknowledges all commands which are blocking an axis interface.
clear_ax_01	BOOL	These 16 inputs can be used to cancel the block of the axis interface of each axis separately caused by an unacknowledged command.
…	BOOL	
clear_ax_16	BOOL	

VAR_OUTPUT		
Variable name	**Data type**	**Description**
any_blocked	BOOL	Is TRUE if the block caused by an unacknowledged command was detected at any axis interface after the start of the check executed for the axis interfaces.
highest_id_blocked	BOOL	Indicates the identification number of the axis with the highest number that is blocked by an unacknowledged command.
blocked_ax_01	BOOL	There are 16 outputs, each of which indicates exactly where each axis is blocked by an unacknowledged command. If the value is TRUE, the axis is blocked.
…	BOOL	
blocked_ax_16	BOOL	

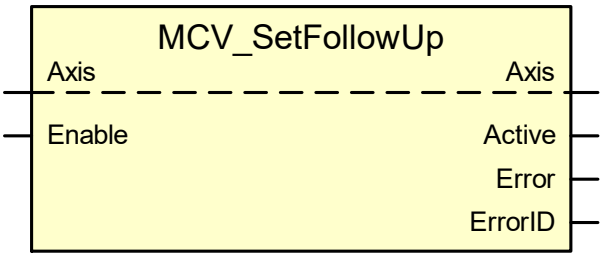
Behaviour of the FB

- The function block can be command independently of the state of the axis.
- The check for blocked axis interfaces is started via the "check_plug" input. An axis interface is considered to be block and thus incapable of reporting other commands issued to the assigned axis if a command is applied at the interface as processed for more than **10** successive PLC cycles, but has not been acknowledged by the PLC. The blocking command is identified by its ordinal number and this is stored.
- The "any_blocked" output becomes TRUE if at least one axis interface is blocked.
- If an axis is blocked, the "blocked_ax_01" "... blocked_ax_16" assigned to an axis becomes TRUE.
- If any axis is blocked, the "highest_id_blocked" output indicates the highest ordinal number of all blocked axes.
- Via the "clear_all_plugs" input, for every blocked axis precisely the command is acknowledged by this FB whose ordinal number has been stored for the axis.
- For each axis, blockage can be individually removed via the "clear_ax_01 ..." "clear_ax_16 inputs".

4.7 MCV_SetFollowUp

Tracking mode is activated or deactivated with the ISG_SetFollowUp FB.

Block diagram



FB parameters

VAR_IN_OUT		
Variable name	**Data type**	**Description**
Axis	AXIS_REF	Axis reference
VAR_INPUT		
Variable name	**Data type**	**Description**
Enable	BOOL	If “Enable” is TRUE; tracking mode is active.
VAR_OUTPUT		
Variable name	**Data type**	**Description**
Active	BOOL	Tracking mode is active.
Error	BOOL	Is TRUE if an error occurs in the FB.
ErrorID	WORD	Error identifier

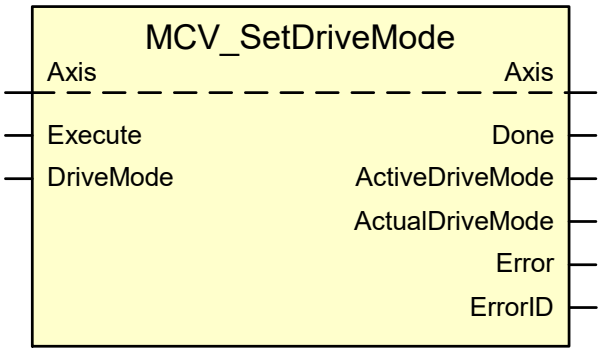
Behaviour of the FB:

- Tracking mode is activated or deactivated via the “Enable” input.
- After the “Enable” input is deactivated, the function must wait for several cycles (at least 10) until tracking mode is terminated in the interpolator.
- It is not possible to command tracking mode during the interpolation, otherwise this leas to the error P-ERR-44051 (ERR_PO_AX_FOLLOW_UP_NOT_ALLOWED) at the FB output “ErrorID”.

4.8 MCV_SetDriveMode

Different operation modes can be set with the MCV_SetDriveMode FB. The table below lists the possible operation modes.

Block diagram



FB parameters

VAR_IN_OUT			
Variable name	**Data type**	**Description**	
Axis	AXIS_REF	Axis reference	
VAR_INPUT			
Variable name	**Data type**	**Description**	
Execute	BOOL	A rising flank at the input signal causes a change to the specified drive mode	
DriveMode	INT	**Value**	**Operation mode**
		0	Position control
		1	Velocity control
		2	Torque control
		3	Position control with torque pilot control
VAR_OUTPUT			
Variable name	**Data type**	**Description**	
Done	BOOL	Is TRUE when a drive mode switch-over was completed successfully.	
ActiveDriveMode	BOOL	The FB MCV_SetDriveMode is active.	
ActualDriveMode	INT	Current drive mode	
Error	BOOL	Is TRUE if an error occurs in the FB.	
ErrorID	WORD	Error code	

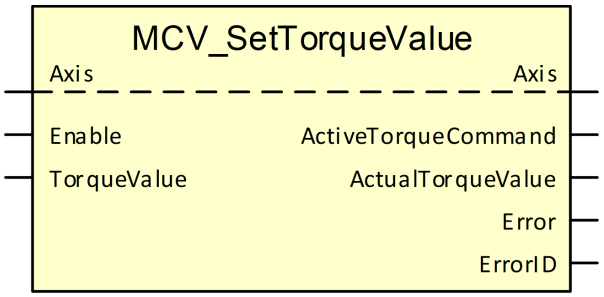
Behaviour of the FB:

- To prevent the occurrence of a position lag error during torque control, the user must deactivate the position lag monitor (see MCV_PosLagMonitor FB).
- Use of an invalid drive mode sets the "Error“ output to TRUE. The “ErrorID“ output indicates the error P-ERR-44049 (ERR_PO_AX_INVALID_DRIVE_MODE).
- A change in drive mode is only permitted in the “Disabled” and “Standstill” states. If this is not the case, the “Error“ output becomes TRUE and “ErrorID“ indicates the error P-ERR-44050 (ERR_PO_AX_TNA_TORQUE_CTRL_ACTIVE).

4.9 MCV_SetTorqueValue

The value for the maximum torque in ±0,1% is set with the MCV_SetTorqueValue FB.

Blockdiagramm



Parameters of the FB

VAR_IN_OUT		
Variable name	**Data type**	**Description**
axis	AXIS_REF	Axis reference

VAR_INPUT		
Variable name	**Data type**	**Description**
enable	BOOL	Activates the specified torque for torque control and torque pilot control.
TorqueValue	INT	Maximum torque in ±0.1%

VAR_OUTPUT		
Variable name	**Data type**	**Description**
ActiveTorqueCommand	BOOL	Torque is active.
ActualTorqueValue	INT	Current torque in ±0.1%
Error	BOOL	Is TRUE when an error occurs in the FB.
ErrorID	WORD	Error code.

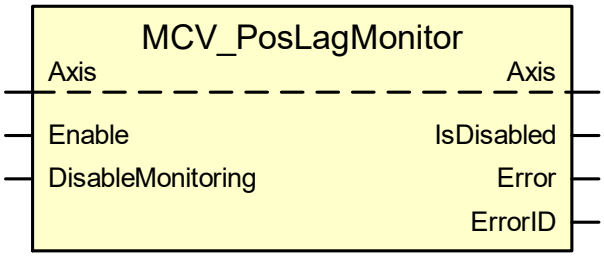
Behaviour of the FB

- The maximum torque is specified in the value range -1000 ≤ 0 ≤ +1000. The unit is 0.1% of the maximum torque; in the positive or negative direction of rotation in each case.
- The entered value is active as soon as "enable" is set to TRUE. The entered value is accepted immediately if the FB is permanently activated.

4.10 MCV_PosLagMonitor

The FB MCV_PosLagMonitor can deactivate position lag monitoring. The default setting for position lag monitoring is active.

Blockdiagramm



Parameters of the FB

VAR_IN_OUT		
Variable name	**Data type**	**Description**
axis	AXIS_REF	Axis reference

VAR_INPUT		
Variable name	**Data type**	**Description**
enable	BOOL	If “Enable” is TRUE, the FB is active.
DisableMonitoring	BOOL	If “DisableMonitoring” is TRUE, position lag monitoring is deactivated.

VAR_OUTPUT		
Variable name	**Data type**	**Description**
IsDisabled	BOOL	Is TRUE when position lag monitoring is deactivated.
Error	BOOL	Indicates TRUE if an error occurs in the FB.
ErrorID	WORD	Error code.

Behaviour of the FB

- The “DisableMonitoring” input signal acts as the on/off switch for the position lag monitor. The setting is activated by "enable".
- Coupling of the two inputs is possible.

4.11 MCV_Home

Attention

This function block is discontinued. For new projects please use the FB MC_Home [▶ 53] which also has the “CamSignal“ input. MC_Home has the identical behaviour.

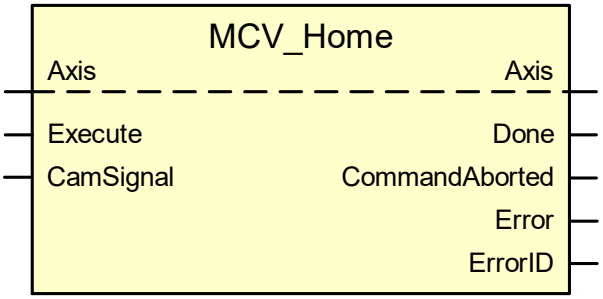
Reference point travel of the axis is commanded with the MCV_Home FB. MCV_Home extends the functionality of the PLCopen FB MC_Home [▶ 53] by an additional "CamSignal" input which is evaluated as the signal to adopt the homing position. Normally, the signal source of "CamSignal" is the homing cam signal of the drive.

Attention

The condition for proper functioning is that the parameter P-AXIS-00036 is assigned the value 0.

The parameter P-AXIS-00038 defines whether adoption of the homing position at the “CamSignal” should take place on the rising or falling edge.

Blockdiagramm



Parameters of the FB

VAR_IN_OUT		
Variable name	**Data type**	**Description**
axis	AXIS_REF	Axis reference

VAR_INPUT		
Variable name	**Data type**	**Description**
Execute	BOOL	A rising edge at the input commands homing.
CamSignal	BOOL	Signal to command adoption of the homing position.

VAR_OUTPUT		
Variable name	**Data type**	**Description**
Done	BOOL	When TRUE, homing is completed and the axis is in “Standstill” state.
CommandAborted	BOOL	TRUE indicates that the command was aborted by another command.
Error	BOOL	Indicates TRUE if an error occurs in the FB.
ErrorID	WORD	Error code.

Behaviour of the FB

- Commanding reference point travel is permissible only if the axis is in the "Standstill" state. If this is not the case, the "error" output is set to FALSE and "error_id" indicates an error value that designates the state the axis is currently in.
- No new command can be sent when this FB is active. It cannot be retriggered. If an attempt is made to do this anyway, the FB supplies the error P-ERR-44007 (ERR_PO_AX_HOMING_RETRIGGERD).
- The axis change to the "Homing" state directly after command of reference point travel.

4.12 MCV_DisableFeedbackMon

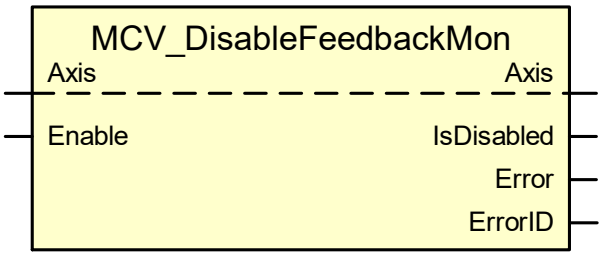
The FB MCV_DisableFeedbackMon disables the motor encoder monitoring, e.g. to permit online decoupling.

The FB deactivates the following KUKA drive error messages:

- Encoder errors
- Actual velocity errors
- Position lag errors
- Resolver errors
- Motor temperature errors

By default, the motor encoder monitoring is enabled.

Blockdiagramm



Parameters of the FB

VAR_IN_OUT		
Variable name	**Data type**	**Description**
axis	AXIS_REF	Axis reference

VAR_INPUT		
Variable name	**Data type**	**Description**
enable	BOOL	If this input is TRUE, the motor encoder is disabled; when FALSE, it is enabled.

VAR_OUTPUT		
Variable name	**Data type**	**Description**
IsDisabled	BOOL	Is TRUE when the motor encoder monitor is disabled.
Error	BOOL	Indicates TRUE if an error occurs in the FB.
ErrorID	WORD	Error code.

4.13 MCV_ChgParamSet

Change parameter set

This function block changes the parameter set of an axis. The parameter set to be activated must already exist and is selected by assigning the input “NCParamSetIDIn”.

For example, this function is used for a tool change to notify the Motion Controller of the changed tool dimensions.

Notice

This function block is not available in TwinCAT systems..

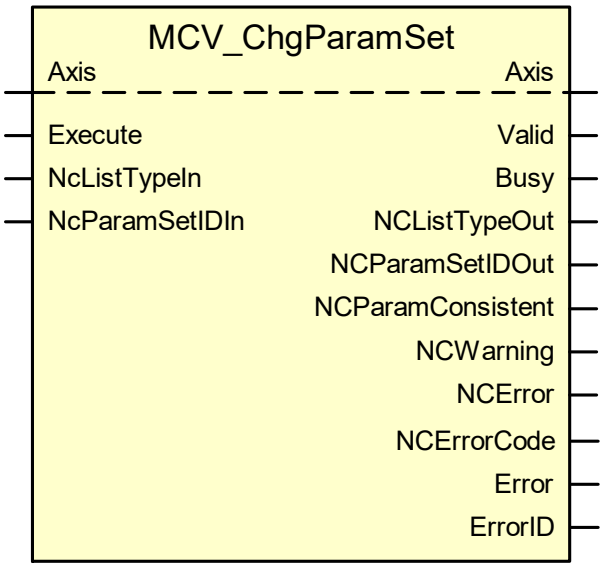
Attention

When a parameter set is changed, the axis must be at standstill and no motion commands may have been started or be started for the axis. The axis may only be moved again after all the parameters of the new parameter set are read in and activated in the controller (“valid“ = TRUE).

Attention

When the parameter set is changed, the behaviour of the controller may change. It may also change or deactivate safety-critical functions.

Block diagram



FB parameters

VAR_IN_OUT		
Variable name	**Data type**	**Description**
Axis	AXIS_REF	Axis reference
VAR_INPUT		
Variable name	**Data type**	**Description**
Execute	BOOL	A change of parameter set is executed on a rising edge.
NcListTypeIn	STRING	Identifier for parameter or list type e.g. “ACHS_MDS“
NcParamSetIDIn	UINT	Number of the parameter set to be switched to. The value 0 for the parameter set ID designates the default parameter set that is interpreted and activated when the Motion Controller is started. Value range [0 to 10] Please note the specification for alternative parameter sets. [▶ 128]
VAR_OUTPUT		
Variable name	**Data type**	**Description**
Valid	BOOL	Is TRUE when the parameter set change was successfully executed.
Busy	BOOL	Is TRUE if the FB is still active.
NCListTypeOut	STRING	Identifier for parameter or list type e.g. “ACHS_MDS“
NCParamSetIDOut	UINT	Number of the parameter set switched to/to be switched to.
NCParamConsistent	BOOL	Is FALSE if the parameters were not fully adopted in the ISG-MCP when the parameter set was changed.
NCWarning	BOOL	Is TRUE if a warning occurred in the ISG-MCP.
NCError	BOOL	Is TRUE if an error occurred in the ISG-MCP while a parameter set is changed.
NCErrorCode	UDINT	Error code of the error or warning occurring in the ISG-MCP.
Error	BOOL	Is TRUE if the FB is in error state. The change in parameter set was not executed successfully.
ErrorID	WORD	Error code

[1]) Availability is dependent on the MCE used.

Behaviour of the FB:

- The FB is in an error state if "Error" = TRUE. The change in parameter set was not executed correctly. An error occurred either in the function block itself or in the ISG-MCP. The "NCError" output parameter indicates whether an error occurred in the ISG-MCP.
- If “NCError“ = TRUE and “NCParamConsistent“ = TRUE, the new parameter set was not interpreted correctly. No new parameters were read in and the old parameters are retained unchanged.

Attention

If the function block enters error state when the parameter set is changed, i.e. “Error” = TRUE, and at the same time “NCError” = TRUE and “NCParamConsistent” = FALSE, the controller has inconsistent data.

Either a new parameter set must be read in correctly or the controller must be rebooted.

- If the FB returns a warning (“NCWarning” = TRUE), at least one parameter was not adopted and the value of this parameter was corrected internally by the controller.

Example

Applying the “NCParamSetIDIn” block

If parameter sets are saved to files, as in the case with sets of the ACHS_MDS type, the default parameter set is the one specified in the start-up list for the associated axis and is interpreted and activated when the motion controller is started up. Example of the 1st axis in the system

```
achs_mds[0]      isgnc\listen\achsmds1.lis
```

The default parameter set is selected by applying the value 0 to the input !NCParamSetIDIn!,

The names of all further files with alternative parameter sets for this axis are derived from the name of the default parameter set and an additional number separated by an underscore.

```
achsmds1_1.lis
achsmds1_2.lis
...
```

The appended number is applied to the “NCParamSetIDIn” input if an alternative parameter set is to be activated.

The files containing the alternative parameter sets must be located in the same directory as the file of the default parameter set.

4.14 MCV_ReqParamSetId

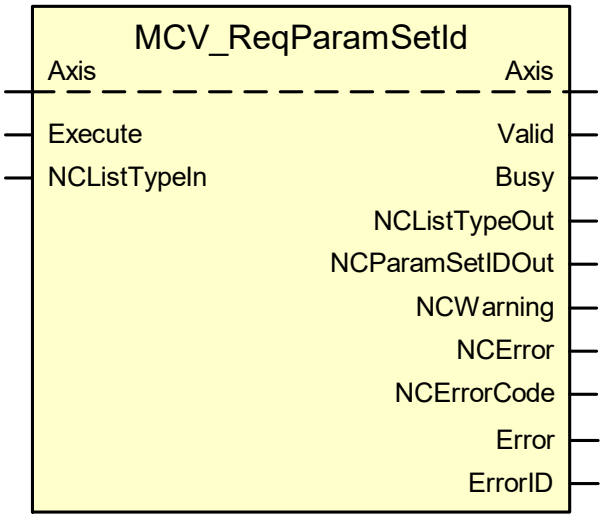
Request parameter set number

The axis administration can request the number of the active parameter set so that the PLC knows which parameter set is currently in use for an axis.

Notice

This function block is not available in TwinCAT systems..

Block diagram



FB parameters

VAR_IN_OUT		
Variable name	**Data type**	**Description**
Axis	AXIS_REF	Axis reference
VAR_INPUT		
Variable name	**Data type**	**Description**
Execute	BOOL	The request for the current parameter set number starts on the rising edge.
NCListTypeIn	STRING	Identifier for parameter or list type e.g. “ACHS_MDS“

VAR_OUTPUT		
Variable name	**Data type**	**Description**
Valid	BOOL	Is TRUE if the current parameter set number was successfully read.
Busy	BOOL	Is TRUE if the FB is still active.
NCListTypeOut	STRING	Identifier for parameter or list type e.g. “ACHS_MDS“
NCParamSetIdOut	UINT	Number of the parameter set that is currently used for this axis.
NCWarning	BOOL	Is TRUE if a warning occurred in the ISG-MCP.
NCError	BOOL	Is TRUE if an error occurred in the ISG-MCP while a parameter set is changed.
NCErrorCode	UDINT	Error code of the error or warning occurring in the ISG-MCP.
Error	BOOL	Is TRUE if the FB is in error state. The request for the parameter set number failed.
ErrorID	WORD	Error code

[1]) Availability is dependent on the MCE used.

Behaviour of the FB:

- The FB is in an error state if "Error" = TRUE. The active parameter set number could not be read. An error occurred either in the function block itself or in the ISG-MCP. The "NCError" output parameter indicates whether an error occurred in the ISG-MCP.

4.15 MCV_ChgParamValue

Change parameter value

To prevent problems caused by incorrect parameterisation, this block does not change the original parameter set but creates a copy and writes the changes to it. The copy is referred to as the **work list** in this documentation.

The original parameter set need not correspond to the currently active parameter data record. It is selected by specifying its identifier at the ”NcParamSetIDIn” input.

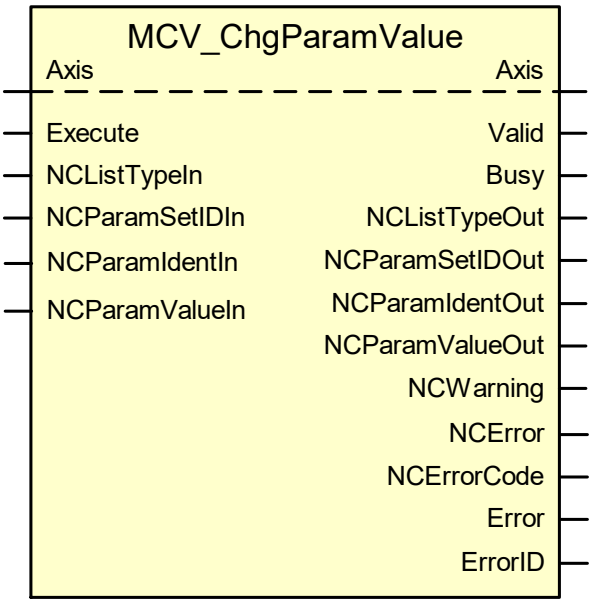
Notice

This function block is not available in TwinCAT systems..

Notice

The modified parameters in the work list [▶ 133] are only adopted by the ISG-MCP and only take effect after they are activated by the **MCV_ActivateParamChg** function block.

Block diagram



FB parameters

VAR_IN_OUT		
Variable name	**Data type**	**Description**
Axis	AXIS_REF	Axis reference

VAR_INPUT		
Variable name	**Data type**	**Description**
Execute	BOOL	A change of parameter value is executed on a rising edge.
NCListTypeIn	STRING	Identifier for parameter or list type e.g. “ACHS_MDS“
NCParamSetIDIn	UINT	Number of the parameter set where the parameter value is to be changed.
NCParamIdentIn	STRING	Identifier of parameter to be changed, e.g. “knngr.swe_pos“
NCParamValueIn	STRING	New value of the parameter..
VAR_OUTPUT		
Variable name	**Data type**	**Description**
Valid	BOOL	Is TRUE when the parameter set change was executed successfully.
Busy	BOOL	Is TRUE if the FB is still active.
NCListTypeOut	STRING	Identifier for parameter or list type e.g. “ACHS_MDS“
NCParamSetIDOut	UINT	Number of the parameter set where the parameter value was/is to be exchanged.
NCParamIdentOut	STRING	Identifier of parameter to be changed e.g. “knngr.swe_pos“
NCParamValueOut	STRING	New value of the parameter..
NCWarning	BOOL	Is TRUE if a warning occurred in the ISG-MCP.
NCError	BOOL	Is TRUE if an error occurred in the ISG-MCP while a parameter value is changed.
NCErrorCode	UDINT	Error code of the error or warning occurring in the ISG-MCP.
Error	BOOL	Is TRUE if the FB is in error state. The change in the parameter value was not executed successfully.
ErrorID	WORD	Error code

[1]) Availability is dependent on the MCE used.

Behaviour of the FB:

- The FB is in an error state if "Error" = TRUE. The parameter was not written to the work list [▶ 133]. An error occurred either in the function block itself or in the ISG-MCP. The "NCError" output parameter indicates whether an error occurred in the ISG-MCP.
- A frequent cause of ISG-MCP errors are absent write access or a write-protected work list since in these cases, a work list [▶ 133] was not created or a parameter cannot be written.

4.16 MCV_ActivateParamChg

Activate modified parameters from the work list

As a result of this activation, the modified parameters stored in the work list [▶ 133] are adopted by the ISG-MCP and take effect in the controller. At the same time, the parameter values are checked for plausibility.

Notice

This function block is not available in TwinCAT systems..

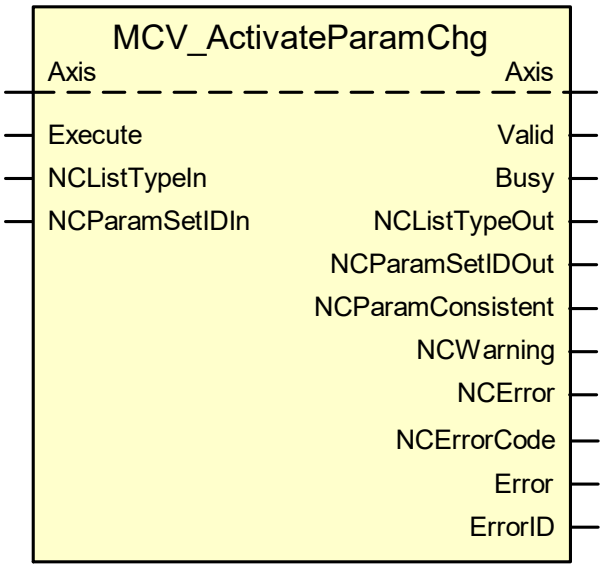
Attention

When a parameter set is activated, the axis must be at standstill and no motion commands may have been started or be started for the axis. The axis may only be moved again after all parameters have been completely activated ("valid" = TRUE).

Attention

When a parameter set is activated with the work list [▶ 133], the behaviour of the controller may change. It may also change or deactivate safety-critical functions.

Block diagram



FB parameters

VAR_IN_OUT		
Variable name	**Data type**	**Description**
Axis	AXIS_REF	Axis reference
VAR_INPUT		
Variable name	**Data type**	**Description**
Execute	BOOL	A change of the parameter is executed on a rising edge.
NCListTypeIn	STRING	Identifier for parameter or list type e.g. “ACHS_MDS“
NCParamSetIDIn	UINT	Number of the parameter set which is to be activated.
VAR_OUTPUT		
Variable name	**Data type**	**Description**
Valid	BOOL	Is TRUE when the activation was successfully executed.
Busy	BOOL	Is TRUE if the FB is still active.
NCListTypeOut	STRING	Identifier for parameter or list type e.g. “ACHS_MDS“
NCParamSetIDOut	UINT	Number of the parameter set which was activated.
NCParamConsistent	BOOL	Is FALSE if an error occurred while updating a parameter in the ISG-MCP and values from different parameter sets are used in the ISG-MCP.
NCWarning	BOOL	Is TRUE if a warning occurred in the ISG-MCP.
NCError	BOOL	Is TRUE if an error occurred in the ISG-MCP during activation.
NCErrorCode	UDINT	Error code of the error or warning occurring in the ISG-MCP.
Error	BOOL	Is TRUE if the FB is in error state. Parameter activation was not executed successfully.
ErrorID	WORD	Error code

[1]) Availability is dependent on the MCE used.

Behaviour of the FB:

- The FB is in an error state if "Error" = TRUE. Updating the parameter set was not executed successfully. An error occurred either in the function block itself or in the ISG-MCP. The "NCError" output parameter indicates whether an error occurred in the ISG-MCP.
- If “NCError“ = TRUE and “NCParamConsistent“ = TRUE, the new parameter set was not interpreted correctly. No new parameters were read from the work list [▶ 133] and the old parameters are retained unchanged in the controller.

Attention

If the function block enters error state when the parameter set with the work list [▶ 133] is updated, i.e. “Error” = TRUE and at the same time “NCError” = TRUE and “NCParamConsistent” = FALSE, the controller has inconsistent data.

Either a new parameter set must be read in correctly or the controller must be rebooted.

- If the FB returns a warning (“NCWarning” = TRUE), at least one parameter was not adopted and the value of this parameter was corrected internally by the controller.
- Successful activation of the work list [▶ 133] is a condition for it being written back later as an original list using the MCV_SaveParamChg function block.
- If the parameters in the work list [▶ 133] were not activated successfully, they can be deleted using the function block MCV_DiscardParamChg. This discards all parameter changes made.

4.17 MCV_SaveParamChg

Save modified parameters

This function block saves the work list [▶ 133] as an original work list after it is activated successfully. After it is successfully saved, the previous work list is deleted.

After a controller reboot or a parameter set change, the modified parameters are then remanently available in the original list.

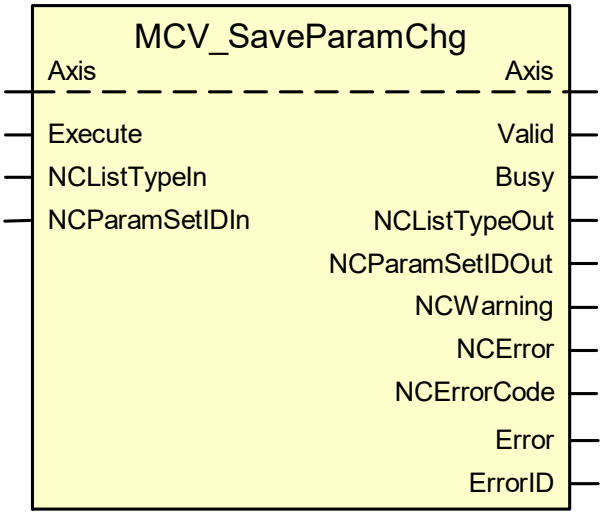
Notice

This function block is not available in TwinCAT systems..

Notice

If a warning occurred when the work list [▶ 133] was activated, at least one parameter was corrected in the controller. The corrected value is not included in the original list. When the original list is read in, the warning occurs again and the parameter value is corrected a second time.

Block diagram



FB parameters

VAR_IN_OUT		
Variable name	**Data type**	**Description**
Axis	AXIS_REF	Axis reference
VAR_INPUT		
Variable name	**Data type**	**Description**
Execute	BOOL	The changed parameters are saved on the rising edge.
NCListTypeIn	STRING	Identifier for parameter or list type e.g. “ACHS_MDS“
NCParamSetIDIn	UINT	Number of the parameter set which is to be saved.
VAR_OUTPUT		
Variable name	**Data type**	**Description**
Valid	BOOL	Is TRUE if the change parameters were saved successfully.
Busy	BOOL	Is TRUE if the FB is still active.
NCListTypeOut	STRING	Identifier for parameter or list type e.g. “ACHS_MDS“.
NCParamSetIDOut	UINT	Number of the parameter set which was saved/is to be saved.
NCWarning	BOOL	Is TRUE if a warning occurred in the ISG-MCP.
NCError	BOOL	Is TRUE if an error occurred in the ISG-MCP while the modified parameters were saved.
NCErrorCode	UDINT	Error code of the error or warning occurring in the ISG-MCP.
Error	BOOL	Is TRUE if the FB is in error state. Saving the changed parameters was not successful.
ErrorID	WORD	Error code

[1]) Availability is dependent on the MCE used.

Behaviour of the FB:

- The FB is in an error state if "Error" = TRUE. Saving the work list was not executed successfully. An error occurred either in the function block itself or in the ISG-MCP. The "NCError" output parameter indicates whether an error occurred in the ISG-MCP.
- A frequent cause of ISG-MCP errors are absent write access or a write-protected original list since in these cases the original list cannot be replaced.

4.18 MCV_ReadParamValue

Read parameter value

This function block reads the parameter values from the original list or work list [▶ 133] of a selected parameter set. If the parameter searched for is in the original list that still exists, it is returned as the "old parameter". If a work list [▶ 133] exists and contains the parameter value, it is returned as the "new parameter".

Notice

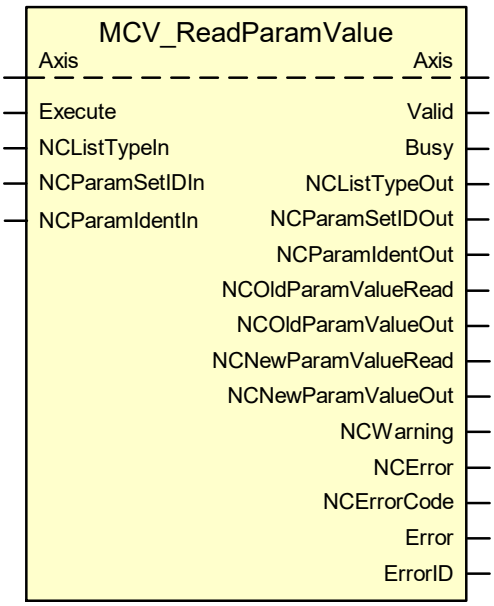
This function block is not available in TwinCAT systems..

Notice

The parameter value effective in the controller may differ from the two parameter values read in if the parameter value was corrected in the controller due to a warning when interpreting the original or work list [▶ 133].

After activation without a warning, the parameter value in the work list is active in the controller. Otherwise, the value of the original list is taken.

Block diagram



FB parameters

VAR_IN_OUT		
Variable name	**Data type**	**Description**
Axis	AXIS_REF	Axis reference
VAR_INPUT		
Variable name	**Data type**	**Description**
Execute	BOOL	The changed parameters are saved on the rising edge.
NCListTypeIn	STRING	Identifier for parameter or list type e.g. “ACHS_MDS“
NCParamSetIDIn	UINT	Number of the parameter set from which the parameter value is to be read.
NCParamIdentIn	STRING	Identifier of the parameter whose value is to be read, e.g. “knngr.swe_pos“
VAR_OUTPUT		
Variable name	**Data type**	**Description**
Valid	BOOL	Is TRUE if the change parameters were saved successfully.
Busy	BOOL	Is TRUE if the FB is still active.
NCListTypeOut	STRING	Identifier for parameter or list type e.g. “ACHS_MDS“
NCParamSetIDOut	UINT	Number of the parameter set which was read from the parameter value.
NCParamIdentOut	STRING	Identifier of parameter to be changed, e.g. “knngr.swe_pos“
NCOldParamValueRead	BOOL	Is TRUE if the parameter identifier was in the original list.
NCOldParamValueOut	STRING	Parameter value from the original list.
NCNewParamValueRead	BOOL	Is TRUE if the parameter identifier was in the work list.
NCNewParamValueOut	STRING	Parameter value from the work list.
NCWarning	BOOL	Is TRUE if a warning occurred in the ISG-MCP.
NCError	BOOL	Is TRUE if an error occurred in the MCE when a parameter value is read.
NCErrorCode	UDINT	Error code of the error or warning occurring in the ISG-MCP.
Error	BOOL	Is TRUE if the FB is in error state. Saving the changed parameters was not successful.
ErrorID	WORD	Error code

[1]) Availability is dependent on the MCE used.

Behaviour of the FB:

- The FB is in an error state if "Error" = TRUE. The parameter value could not be read correctly. An error occurred either in the function block itself or in the ISG-MCP. The "NCError" output parameter indicates whether an error occurred in the ISG-MCP.
- The ISG-MCP returns an error if the parameter value is not in any of the lists.
- If the value of a parameter is determined from the active parameter set, the code of the active parameter set is first determined by an instance of the **MCV_ReqParamSetID**. This code is applied to the “NCParamSetIDIn” input of the MCV_ReadParamValue The parameter value from the active parameter set is then applied to the “NCOldParamValueOut” output and is valid when the “NCOldParamValueRead” output = TRUE. If a work list exists at the same time, the value from it is indicated at the “NCNewParamValueOut” output.

4.19 MCV_DiscardParamChg

Discard changes

This function block is used to delete the work list [▶ 133]. This discards all parameter changes in the work list.

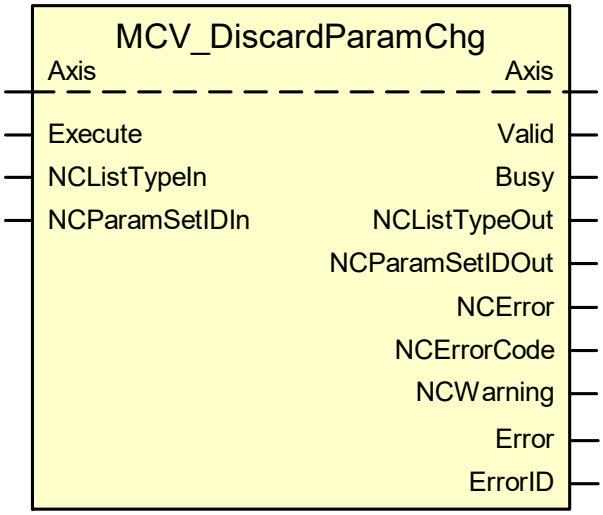
Notice

This function block is not available in TwinCAT systems..

Notice

If the work list was already activated by the **MCV_ActivateParamChg** function block, the changes in the controller are effective and this function block can no longer reverse this situation. In order to reverse the changed parameter values, the original list must be read in using the **MCV_ChgParamSet** function block.

Block diagram



FB parameters

VAR_IN_OUT		
Variable name	**Data type**	**Description**
Axis	AXIS_REF	Axis reference
VAR_INPUT		
Variable name	**Data type**	**Description**
Execute	BOOL	Discarding changes is executed on a rising edge.
NCListTypeIn	STRING	Identifier for parameter or list type e.g. “ACHS_MDS“
NCParamSetIDIn	UINT	Number of the parameter set whose changes are to be discarded.
VAR_OUTPUT		
Variable name	**Data type**	**Description**
Valid	BOOL	Is TRUE when the parameter set changes were discarded successfully.
Busy	BOOL	Is TRUE if the FB is still active.
NCListTypeOut	STRING	Identifier for parameter or list type, e.g. “ACHS_MDS“
NCParamSetIDOut	UINT	Number of the parameter set whose changes are to be discarded.
NCWarning	BOOL	Is TRUE if a warning occurred in the ISG-MCP.
NCError	BOOL	Is TRUE if an error occurred in the MCE while changes were discarded.
NCErrorCode	UDINT	Error code of the error or warning occurring in the ISG-MCP.
Error	BOOL	Is TRUE if the FB is in error state. The changes in the parameter set were not discarded successfully.
ErrorID	WORD	Error code

[1]) Availability is dependent on the MCE used.

Behaviour of the FB:

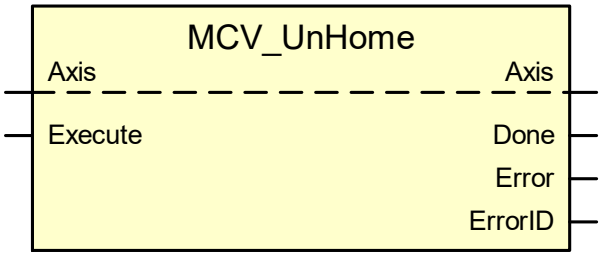
- The FB is in an error state if "Error" = TRUE. Parameter changes could not be cancelled correctly. An error occurred either in the function block itself or in the ISG-MCP. The "NCError" output parameter indicates whether an error occurred in the ISG-MCP.

A frequent cause of ISG-MCP errors are absent write accesses or a write-protected work list [▶ 133] since in these cases the work list cannot be deleted.

4.20 MCV_UnHome

This function block unreferences an axis.

Blockdiagramm



Parameters of the FB

VAR_IN_OUT		
Variable name	**Data type**	**Description**
axis	AXIS_REF	Axis reference

VAR_INPUT		
Variable name	**Data type**	**Description**
Execute	BOOL	An axis is unreferenced on a rising edge.

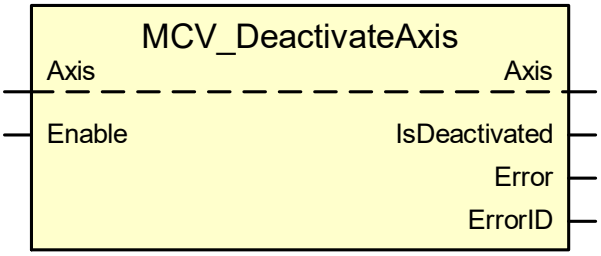
VAR_OUTPUT		
Variable name	**Data type**	**Description**
Done	BOOL	Unreferencing was executed successfully.
Error	BOOL	Indicates TRUE if an error occurs in the FB.
ErrorID	WORD	Error code.

4.21 MCV_DeactivateAxis

The FB deactivates or parks an axis. In this state it is no longer possible to move the axis and no further error messages are displayed.

The axis is activated by default

Blockdiagramm



Parameters of the FB

VAR_IN_OUT		
Variable name	**Data type**	**Description**
axis	AXIS_REF	Axis reference

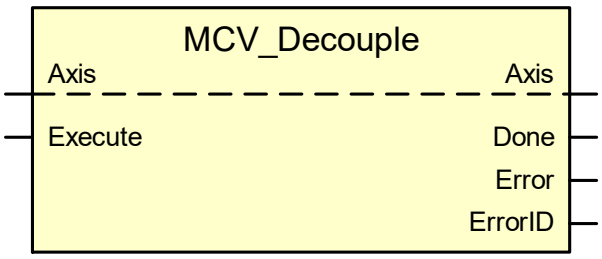
VAR_INPUT		
Variable name	**Data type**	**Description**
enable	BOOL	Is TRUE when the axis is deactivated.

VAR_OUTPUT		
Variable name	**Data type**	**Description**
IsDeactivated	BOOL	Is TRUE when the axis is deactivated.
Error	BOOL	Indicates TRUE if an error occurs in the FB.
ErrorID	WORD	Error code.

4.22 MCV_Decouple

The “MCV_Decouple” FB decouples an axis, i.e. it is deactivated in order to replace a motor, for example. The sequence of the internally commanded FBs is shown in the flow chart below.

Block diagram



FB parameters

VAR_IN_OUT		
Variable name	**Data type**	**Description**
Axis	AXIS_REF	Axis reference
VAR_INPUT		
Variable name	**Data type**	**Description**
Execute	BOOL	The axis coupling sequence is triggered on a rising edge
VAR_OUTPUT		
Variable name	**Data type**	**Description**
Done	BOOL	The axis was decoupled successfully.
Error	BOOL	Is TRUE if an error occurs in the FB.
ErrorID	WORD	Error code

Behaviour of the FB:

- The sequence of axis decoupling is listed in the diagram below.
- At the beginning of the sequence the current axis position is saved to parameter P-AXIS-00152 (pos_refpkt).

Flow chart

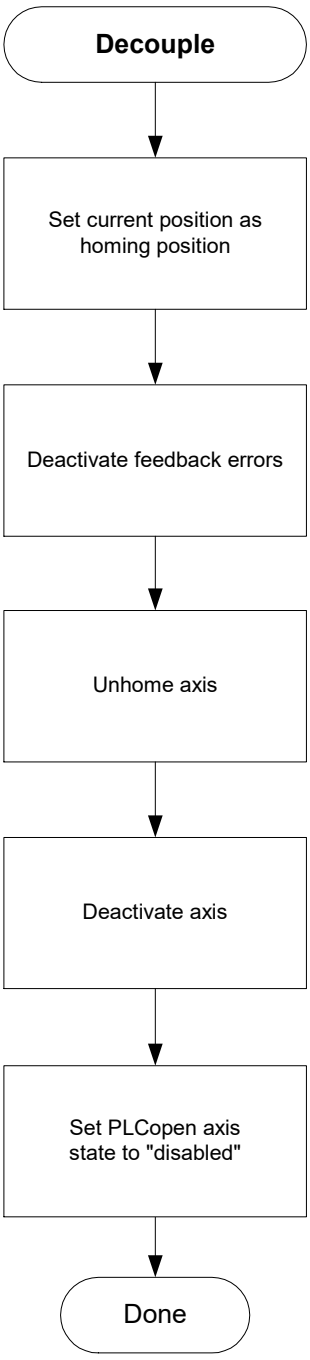
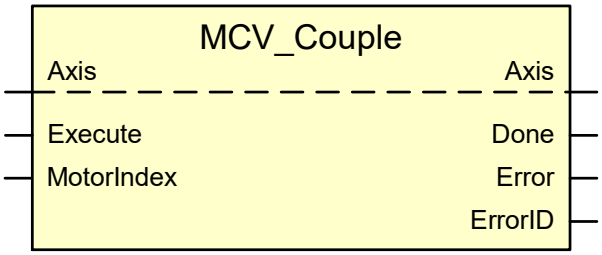


Fig. 20: Flow chart - Decoupling an axis

4.23 MCV_Couple

With the MCV_Couple FB an axis may be recoupled, i.e. it will be reactivated., e.g. after decoupling to replace a motor. The sequence of the internally commanded FBs is shown in the flow chart below.

Block diagram



FB parameters

VAR_IN_OUT		
Variable name	**Data type**	**Description**
Axis	AXIS_REF	Axis reference
VAR_INPUT		
Variable name	**Data type**	**Description**
Execute	BOOL	The axis coupling sequence is triggered on a rising edge.
MotorIndex	UDINT	The value at this input determines which parameter set is activated when an axis is coupled. Axes can therefore be coupled to different drive motors.
VAR_OUTPUT		
Variable name	**Data type**	**Description**
Done	BOOL	The axis was coupled successfully.
Error	BOOL	Indicates TRUE if an error occurs in the FB.
ErrorID	WORD	Error code

Flow chart

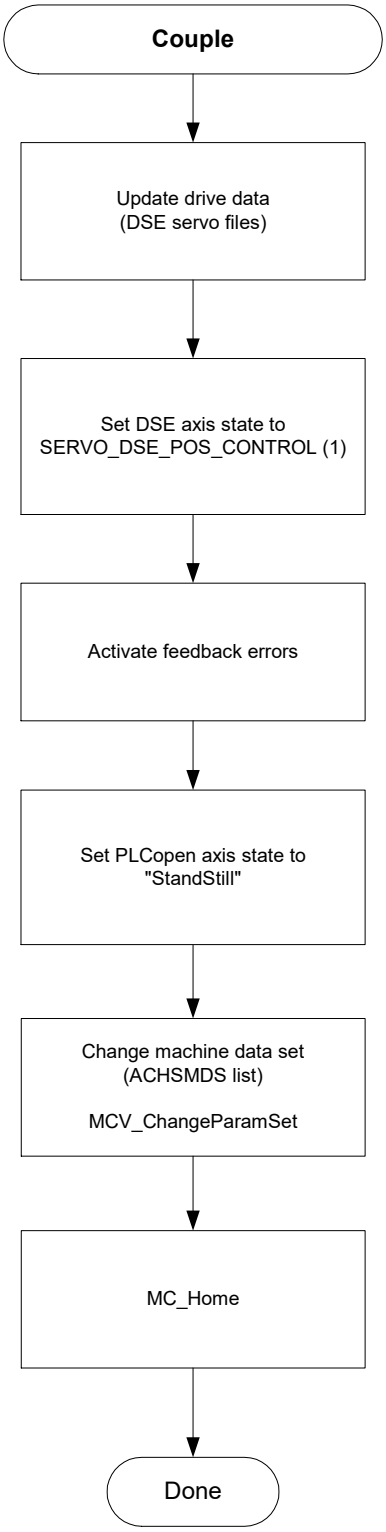


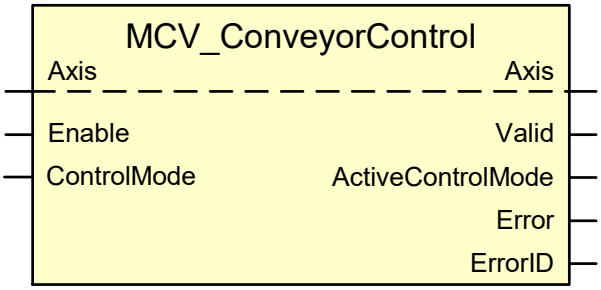
Fig. 21: Flow chart - Coupling an axis

4.24 MCV_ConveyorControl

The “MCV_ConveyorControl” FB signals the dynamic state of an axis to the motion controller when the axis is not subject to its control. This axis is parameterised as an encoder axis (see P-AXIS-00015).

This information influences the synchronisation motion of an axis group for which this encoder axis was parameterised as a master axis.

Block diagram



FB parameters

VAR_IN_OUT			
Variable name	**Data type**	**Description**	
Axis	AXIS_REF	Axis reference	
VAR_INPUT			
Variable name	**Data type**	**Description**	
Enable	BOOL	TRUE, dynamic state of the encoder axis is sent to the motion controller.	
ControlMode	UDINT	**Value**	**Constant**
		0	HLI_CONVEYOR_SPEED_CONST
		1	HLI_CONVEYOR_DECELERATION
		2	HLI_CONVEYOR_DECELERATION
		3	HLI_CONVEYOR_SPEED_ZERO
		4	HLI_CONVEYOR_MACH_WITH_SPEED_ZERO
VAR_OUTPUT			
Variable name	**Data type**	**Description**	
Valid	BOOL	TRUE, value at the “ActiveControlMode” output is valid.	
ActiveControlMode	UDINT	Current dynamic state of the encoder axis which was signalled to the motion controller.	
Error	BOOL	Indicates TRUE if an error occurs in the FB.	
ErrorID	WORD	Error code	

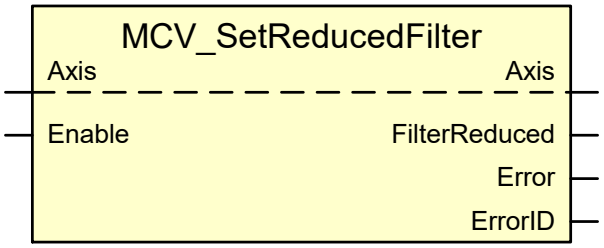
Behaviour of the FB:

- This function is not available in all motion controller variants.
- In the variants in which this function is available, the parameters of the group conv_sync must be defined in the axis parameter list ([AXIS]) and the parameters of the conveyor_sync group must be set in the channel parameter list ([CHAN]).

4.25 MCV_SetReducedFilter

The “MCV_SetReducedFilter” FB influences the filtering of actual values of an axis and is usually used in combination with an axis configured as an encoder axis (see P-AXIS-00015) which is not controlled by the motion controller. The FB is only effective if a slave axis was configured for this encoder axis.

Blockdiagramm



Parameters of the FB

VAR_IN_OUT		
Variable name	**Data type**	**Description**
axis	AXIS_REF	Axis reference

VAR_INPUT		
Variable name	**Data type**	**Description**
enable	BOOL	TRUE, the parameter for actual value filtering is activated for a reduced number of values.

VAR_OUTPUT		
Variable name	**Data type**	**Description**
FilterReduced	BOOL	TRUE; Actual value filtering is then executed by the reduced number of values which were parameterised.
Error	BOOL	Indicates TRUE if an error occurs in the FB.
ErrorID	WORD	Error code.

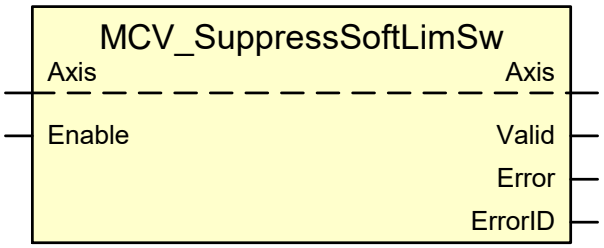
Behaviour of the FB

- To parameterise the filter effect, the parameters of the conv_sync group must be configured accordingly in the axis parameter list ([AXIS]).

4.26 MCV_SuppressSoftLimSw

The “MCV_SuppressSoftLimSw” FB executes the output of diagnosis messages in conjunction with software limits switches.

Blockdiagramm



Parameters of the FB

VAR_IN_OUT		
Variable name	**Data type**	**Description**
axis	AXIS_REF	Axis reference

VAR_INPUT		
Variable name	**Data type**	**Description**
enable	BOOL	TRUE, the output of diagnosis messages referring to software limit switches is suppressed.

VAR_OUTPUT		
Variable name	**Data type**	**Description**
Valid	BOOL	TRUE, output suppression is activated.
Error	BOOL	Indicates TRUE if an error occurs in the FB.
ErrorID	WORD	Error code.

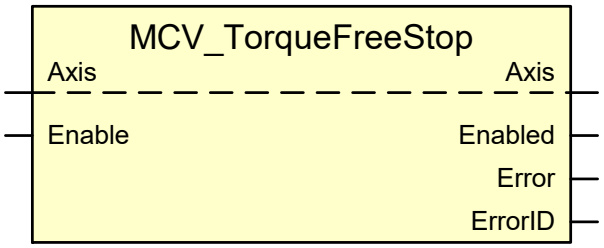
Behaviour of the FB

- Suppressed diagnosis messages include: P-ERR-50041, P-ERR-50042, P-ERR-60158 and P-ERR-60159.

4.27 MCV_TorqueFreeStop

The “MCV_TorqueFreeStop” FB influences whether a drive stops when torque is applied after power to the drive is switched off by an instance of MC_Power or whether it coasts freely to a stop without application of torque.

Blockdiagramm



Parameters of the FB

VAR_IN_OUT		
Variable name	**Data type**	**Description**
axis	AXIS_REF	Axis reference

VAR_INPUT			
Variable name	**Data type**	**Description**	
enable	BOOL	TRUE	When power is switched off, the drive coasts freely to a stop without torque application.
		FALSE	When power is switched off, the drive stops when torque is applied.

VAR_OUTPUT		
Variable name	**Data type**	**Description**
Enabled	BOOL	TRUE; coasting freely without torque application when power is switched off.
Error	BOOL	Indicates TRUE if an error occurs in the FB.
ErrorID	WORD	Error code.

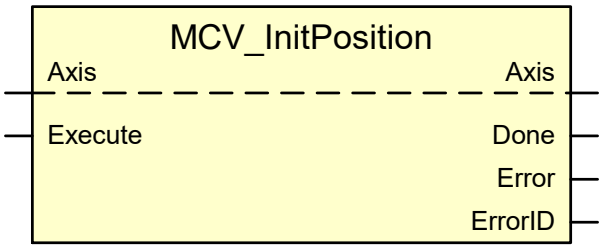
Behaviour of the FB

- Currently, the FB is only available for SERCOS and KUKA drives.

4.28 MCV_InitPosition

The “MCV_InitPosition” FB initialises the command position of the interpolator of an SAI axis with the current command position of the position controller of this SAI axis. This FB is always used to compensate for position differences between of the interpolator and the position controller caused by commanding an instance of the FB MCV_AddCmdValues (see [MCP-CTRL] or by a axis coupling.

Blockdiagramm



Parameters of the FB

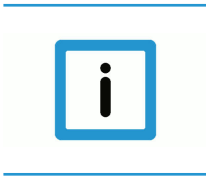
VAR_IN_OUT		
Variable name	**Data type**	**Description**
axis	AXIS_REF	Axis reference

VAR_INPUT		
Variable name	**Data type**	**Description**
Execute	BOOL	TRUE, starts the initialisation of the command position of the interpolator with the command position of the position controller of the SAI axis.

VAR_OUTPUT		
Variable name	**Data type**	**Description**
Done	BOOL	TRUE, the command position of the interpolator of the SAI axis is initialised with the command position of the position controller.
Error	BOOL	Is TRUE when an error occurs in the FB.
ErrorID	WORD	Error code.

Behaviour of the FB

- If an axis was shifted in the position controller after control unit start, it is not recognised if it is positioned later in the single axis interpolator. By commanding an MCV_InitPosition function block before positioning, the user can ensure that the specified absolute target position refers to the original coordinate system which was not shifted. In other words, this FB acts as if a possible shift is cleared (cf. NC command #CHANNEL INIT[CMDPOS] of the NC channel).
- The command is also updated by commanding MCV_SetFollowUp after tracking mode is deactivated.



Notice

Several CNC cycles are required to synchronise the position, i.e. this FB should not be used before each positioning for time-critical actions without good reason.

5 Application notes

5.1 Application note: Configuration of a “synchronisation master”

The following procedure shows an option to implement a type of “synchronisation master”.

5.1.1 Configuring a simulated axis in the ISG-CNC

hochlauf.lis:

```
zahl_mds  increment by 1 and include other axes
achs_mds[X-1]    isgnc\listen\achsmdsX.lis
```

achsmds*X*.lis:

```
#-------------------------------------------------
#  General attributes
#-------------------------------------------------
kenngr.antr_typ   4 /* 1=KONV, 2=SERCOS, 4=SIMU, 17=EMI */
kenngr.achs_mode  0x104
kenngr.achs_typ   4
```

The axis is already simulated in the ISG-CNC. Therefore, no changes are required in the servo files (number of KUKA axes “NumAx” in “global.servo” retains the same setting).

5.1.2 Setting axis parameters

To deactivate the “real” behaviour of a simulated axis (e.g. controller behaviour with position lag), set the following axis MDS parameter:

```
antr_simu.ist_gleich_soll   1
```

In addition, the resolution factors must be set to 1:

```
getriebe[0].wegaufz          1
getriebe[0].wegaufn          1
```

5.1.3 Handling the control units on the HLI

The following control unit must be activated at the PLC interface (HLI):

```
hli.axis[X-1].lr_mc_control.add_cmd_values.enable_w = TRUE
```

It is then possible to specify additive values at the current interpolator setpoint input of the axis.

Additive position setpoint input:

```
hli.axis[X-1].lr_mc_control.add_cmd_values.command_w.m_add_pos_value
```

Additive velocity setpoint input:

```
hli.axis[X-1].lr_mc_control.add_cmd_values.command_w.m_add_speed_value
```

These values are added to the current interpolator position or velocity in every interpolator cycle and output to the drive (so be careful when using the control unit with real drives).

In other words, if an axis was moved to position 90 using MC_MoveAbsolute, an additive position setpoint of 10 causes the axis to move to position 100. This position is then retained until a new position is specified.

To move an axis continuously, we recommend the use of the additive velocity interface. If the accuracy is insufficient, the additive position setpoints must be specified continuously without modulo break to prevent the axis from moving backwards.

5.1.4 GearIn coupling

The actual position of the “synchronised master” can now be specified using the variable described above. Please note that only position control values may be modified (“MC_ReadStatus.lr*”). The interpolator setpoint is not affected. Then coupling must take place on the actual values of this “synchronised master“ axis ("MC_GearIn.Mode" = 16#80).

5.1.5 Actual value filtering

The actual values of the “Synchronised Master” (MC_GearIn-Master) can be filtered in the known way (cam_gear.mv_type, P-AXIS-00300).

5.1.6 CamIn coupling

If the “Synchronised Master” is required as master axis for MC_CamIn, a further simulated axis must be configured in the ISG CNC (see above). This axis can then be coupled to the synchronised master as a slave with an actual value based MC_GearIn. The slave can also be used as a master for the MC_CamIn coupling.

6 References

[1] PLCopen specifications: TC2 Task Force Motion Control “Function Blocks for motion control” Version 1.0, dated 23 Nov. 2001

[2] CNC PLC overall control system documentation

[3] The PLCopen Compliance Statement V1.0 from ISG can be found on the PLCopen website (www.plcopen.org).

7 Appendix

7.1 Suggestions, corrections and the latest documentation

Did you find any errors? Do you have any suggestions or constructive criticism? Then please contact us at documentation@isg-stuttgart.de.

The latest documentation is posted in our Online Help (DE/EN):

QR code link: https://www.isg-stuttgart.de/documentation-kernel/

The link above forwards you to:

https://www.isg-stuttgart.de/fileadmin/kernel/kernel-html/index.html

Notice

Change options for favourite links in your browser;

Technical changes to the website layout concerning folder paths or a change in the HTML framework and therefore the link structure cannot be excluded.

We recommend you to save the above "QR code link" as your primary favourite link.

PDFs for download:

DE:
https://www.isg-stuttgart.de/produkte/softwareprodukte/isg-kernel/dokumente-und-downloads

EN:
https://www.isg-stuttgart.de/en/products/softwareproducts/isg-kernel/documents-and-downloads

Email: documentation@isg-stuttgart.de

Index

D

D-MCP-P1-MC_AbortTrigger ... 71
D-MCP-P1-MC_CamIn ... 75
D-MCP-P1-MC_CamOut ... 82
D-MCP-P1-MC_CamTableSelect ... 83
D-MCP-P1-MC_GearIn ... 84
D-MCP-P1-MC_GearOut ... 89
D-MCP-P1-MC_Halt ... 105
D-MCP-P1-MC_Home ... 53
D-MCP-P1-MC_MoveAbsolute ... 55
D-MCP-P1-MC_MoveAdditive ... 58
D-MCP-P1-MC_MoveRelative ... 61
D-MCP-P1-MC_MoveSuperImposed ... 63
D-MCP-P1-MC_MoveVelocity ... 66
D-MCP-P1-MC_Phasing ... 90
D-MCP-P1-MC_Power ... 41
D-MCP-P1-MC_ReadActualPosition ... 51
D-MCP-P1-MC_ReadAxisError ... 50
D-MCP-P1-MC_ReadAxisInfo ... 43
D-MCP-P1-MC_ReadMotionState ... 45
D-MCP-P1-MC_ReadParameter ... 93
D-MCP-P1-MC_ReadStatus ... 48
D-MCP-P1-MC_Reset ... 52
D-MCP-P1-MC_SetOverride ... 102
D-MCP-P1-MC_SetPosition ... 101
D-MCP-P1-MC_Stop ... 69
D-MCP-P1-MC_TouchProbe ... 73
D-MCP-P1-MC_WriteLrealParameter ... 99
D-MCP-P1-MC_WriteParameter ... 96
D-MCP-P1-MCV_ActivateParamChg ... 135
D-MCP-P1-MCV_Axis ... 111
D-MCP-P1-MCV_CamSwitch ... 112
D-MCP-P1-MCV_ChgParamSet ... 128
D-MCP-P1-MCV_ChgParamValue ... 133
D-MCP-P1-MCV_ConveyorControl ... 151
D-MCP-P1-MCV_Couple ... 149
D-MCP-P1-MCV_DeactivateAxis ... 146
D-MCP-P1-MCV_Decouple ... 147
D-MCP-P1-MCV_DisableFeedbackMon ... 127
D-MCP-P1-MCV_DiscardParamChg ... 143
D-MCP-P1-MCV_Home ... 125
D-MCP-P1-MCV_InitPosition ... 156
D-MCP-P1-MCV_P1_Platform ... 110
D-MCP-P1-MCV_PosLagMonitor ... 124
D-MCP-P1-MCV_ReqParamSetId ... 131
D-MCP-P1-MCV_RmvPlug ... 118
D-MCP-P1-MCV_SaveParamChg ... 138
D-MCP-P1-MCV_SetDriveMode ... 121
D-MCP-P1-MCV_SetFollowUp ... 120
D-MCP-P1-MCV_SetReducedFilter ... 153
D-MCP-P1-MCV_SetTorqueValue ... 123
D-MCP-P1-MCV_SuppressSoftLimSw ... 154
D-MCP-P1-MCV_TorqueFreeStop ... 155
D-MCP-P1-MCV_TorqueLimit ... 116
D-MCP-P1-MCV_UnHome ... 145

M

MC_AbortTrigger ... 71
MC_CamIn ... 75
MC_CamOut ... 82
MC_CamTableSelect ... 83
MC_GearIn ... 84
MC_GearOut ... 89
MC_Halt ... 105
MC_Home ... 53
MC_MoveAbsolute ... 55
MC_MoveAdditive ... 58
MC_MoveRelative ... 61
MC_MoveSuperImposed ... 63
MC_MoveVelocity ... 66
MC_Phasing ... 90
MC_Power ... 41
MC_ReadActualPosition ... 51
MC_ReadAxisError ... 50
MC_ReadAxisInfo ... 43
MC_ReadMotionState ... 45
MC_ReadParameter ... 93
MC_ReadStatus ... 48
MC_Reset ... 52
MC_SetOverride ... 102
MC_SetPosition ... 101
MC_Stop ... 69
MC_TouchProbe ... 73
MC_WriteLrealParameter ... 99
MC_WriteParameter ... 96
MCV_ActivateParamChg ... 135
MCV_Axis ... 111
MCV_CamSwitch ... 112
MCV_ChgParamSet ... 128
MCV_ChgParamValue ... 133
MCV_ConveyorControl; ... 151
MCV_Couple ... 149
MCV_DeactivateAxis ... 146
MCV_Decouple ... 147
MCV_DisableFeedbackMon ... 127
MCV_DiscardParamChg ... 143
MCV_Home ... 125
MCV_InitPosition ... 156
MCV_P1_Platform ... 110
MCV_PosLagMonitor ... 124
MCV_ReqParamSetId ... 131
MCV_RmvPlug ... 118
MCV_SaveParamChg ... 138
MCV_SetDriveMode ... 121
MCV_SetFollowUp ... 120
MCV_SetReducedFilter ... 153
MCV_SetTorqueValue ... 123
MCV_SuppressSoftLimSw ... 154
MCV_TorqueFreeStop ... 155
MCV_TorqueLimit ... 116
MCV_UnHome ... 145

© Copyright
ISG Industrielle Steuerungstechnik GmbH
STEP, Gropiusplatz 10
D-70563 Stuttgart
All rights reserved
www.isg-stuttgart.de
support@isg-stuttgart.de