DOCUMENTATION ISG-kernel

PLC library
McpPLCopen Part 4

Short Description:
MCP-P4

www.isg-stuttgart.de
support@isg-stuttgart.de

Documentation version: 1.04
13/12/2023

General and safety instructions

Icons used and their meanings

This documentation uses the following icons next to the safety instruction and the associated text. Please read the (safety) instructions carefully and comply with them at all times.

Icons in explanatory text

➢ Indicates an action.

⇨ Indicates an action statement.

⚠ DANGER

Acute danger to life!

If you fail to comply with the safety instruction next to this icon, there is immediate danger to human life and health.

⚠ CAUTION

Personal injury and damage to machines!

If you fail to comply with the safety instruction next to this icon, it may result in personal injury or damage to machines.

Attention

Restriction or error

This icon describes restrictions or warns of errors.

Notice

Tips and other notes

This icon indicates information to assist in general understanding or to provide additional information.

Example

General example

Example that clarifies the text.

Programing Example

NC programming example

Programming example (complete NC program or program sequence) of the described function or NC command.

Release Note

Specific version information

Optional or restricted function. The availability of this function depends on the configuration and the scope of the version.

Table of contents

List of figures

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 “McpPLCopenP4”

This library contains function blocks that are defined as default blocks in the PLCopen Part 4 specification.

2.1 PLCopen axis group state diagram

Among other things the axis group state diagram is defined in this specification to show the states which an axis group can adopt. The diagram also shows how jobs sent by a PLCopen-defined function block change the state of an axis group and which jobs are actually permitted in which axis group state.

This state diagram is the basis for implementing the PLC library “McpPlcopenP4“.

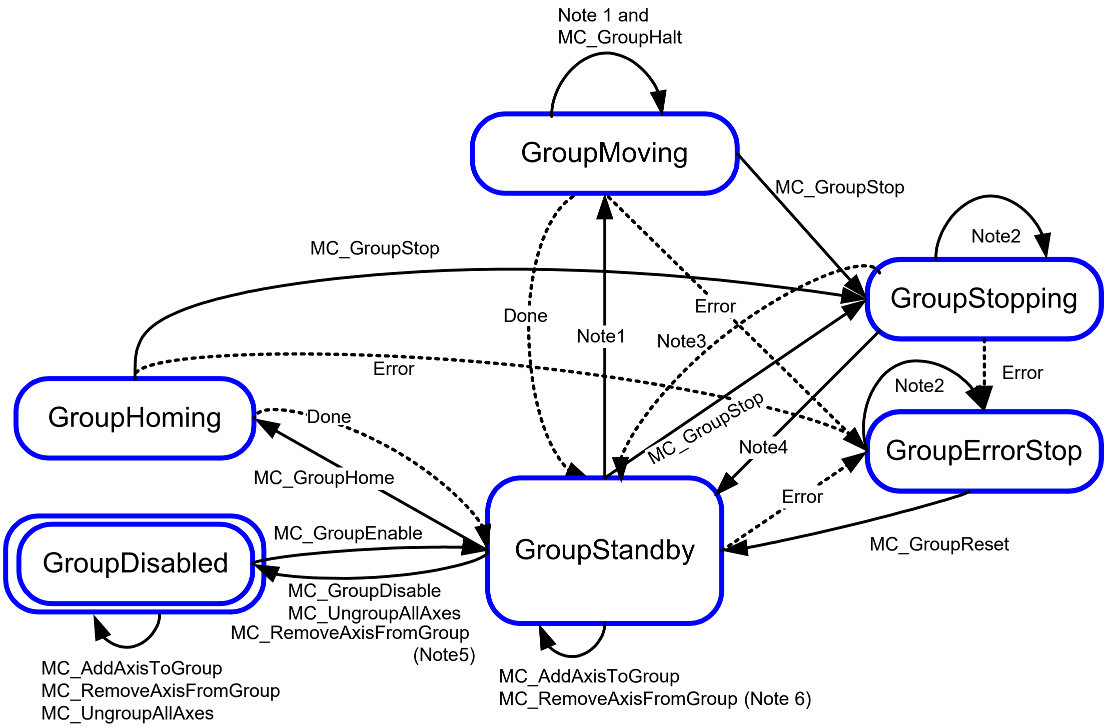


Fig. 1: Axis group state diagram of PLCopen Part 4 specification

2.2 Axis group reference AXES_GROUP_REF

The PLCopen Part4 specification requires the provision of a structure named AXES_GROUP_REF which must be transferred to the individual FBs as a VAR_IN_OUT parameter to identify the axis group commanded. The contents of the AXES_GROUP_REF structure are manufacturer-specific as stipulated in the PLCopen definition.

The ISG implementation, the AXES_GROUP_REF contains a variable **HlilfcIdx** which can be used for the **axis group index** which addresses an axis group-specific area on the **HLI**.

This axis index is assigned once in every PLC project by calling the FB MCV_PlatformBase.

In addition, the structure contains an array for the last six error messages that were notified by the motion controller for an axis group. However, the structure is not defined in the “McpPlcopenP4” library but in the “McpBase” library.

2.2.1 Array for data structures of the AXES_GROUP_REF type

The data structure AXES_GROUP_REF must exist for each axis group, and must be available in all programs pertaining to the PLC project's tasks. To ensure this, all AXES_GROUP_REF structures are managed in an array named **gAxesGroupRef**.

For PLC applications based on the CoDeSys (3S) environment the array **gAxesGroupRef** is provided by the library McpBase.lib as global variable.

2.3 Other references defined according to PLCopen Part 4

Besides the axis group reference, the specification defines other references as data type. Their structure content is not defined since they are manufacturer-specific.

Since it depends on the PLC application whether these references are required at all, only the data types are defined for these references. The definitions are contained in the “McpBase” library. These data types can be defined in a PLC application as required.

A short description of the data types for these references used in the “McpPlcopenP4” library is provided below.

2.3.1 Data structure IDENT_IN_GROUP_REF

This structure is used to define an axis in an axis group and to identify the axis by assigning a name to it. When an axis is added to an axis group (MC_AddAxisToGrp), the variable **ChAxIdx** defines the position where this axis is assigned in the axis group. When an axis is removed from an axis group (MC_RemAxisFromGrp), this variable identifies the axis to be removed.

The variable **Name** defines how the axis must be addressed in the NC program when it receives a motion job.

```
TYPE IDENT_IN_GROUP_REF :
  STRUCT
(* Index of a channel axis status data)
(  at channel specific HLI interface *)
    ChAxIdx :UINT;
    Name    :MCV_STR_AX_NAME;
  END_STRUCT;
END_TYPE
```

2.3.2 Data structure MC_PATH_DATA_REF

This data structure represents a reference to the data of the programmed path which is to be travelled by the axes of the axis group.

In the ISG implementation the reference contains the string **FileName** which includes either the absolute directory path or only the filename of a file and an NC program. The structure is used in relation with a job sent to the function block MC_MovePath.

```
TYPE MC_PATH_DATA_REF :
  STRUCT
(* Name of nc program to be executed *)
    Filename : MCV_STR_PROG_NAME;END_STRUCT;
END_TYPE
```

2.4 Path blending and transition modes

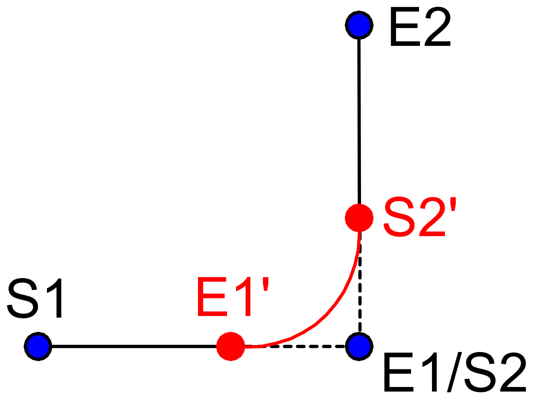
2.4.1 Overview of transition conditions on contouring (blending and transition modes)

If more than one motion FB is executed, the traversed path and the velocity profile will depend on the transition and blending processes transferred to the FBs.

The table below provides an overview of the permitted transitions:

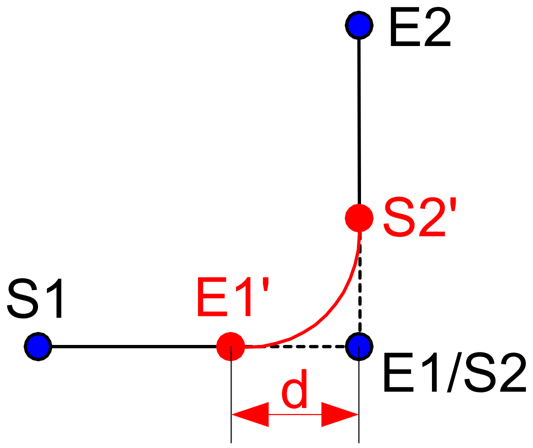
BufferMode	TransitionMode	Description
mcAborting		Aborts the current active motion command and starts a separate motion command.
mcBuffered		The motion command only becomes active when all the previously commanded motions have been executed. When the job is completed, the velocity of the axis group is 0. As a result there is no blending to the next motion job and therefore the "TransitionMode" input is not evaluated.
mcBlending Previous	TMNone	No blending curve is inserted between 2 commands of the motion FB. The transition velocity between 2 commands depends on whether the command of the 2nd FB changes the motion direction.
	TMConstantVelocity	A blending curve is inserted in all these cases. An FB with one of these values at the "TransitionMode" input is not executed immediately since the following NC block is required to calculate the blending curve. With the last FB in a chained sequence of FBs, **Buffered** value must be applied to the "BufferMode" input.
	TMCornerDistance	
	TMMaxCornerDeviation	
	TMConstDeviation	
	TMIntermediatePoint	
	TMDynamicOptimized	

The following definitions for blending paths are contained in the PLCopen specification:

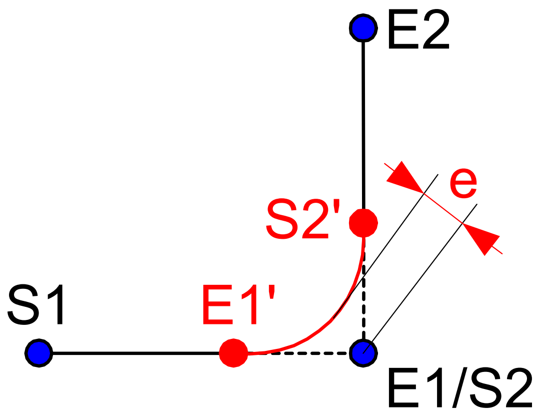


Blending curve: Inserted curve which modifies the original programmed path (E1' - S2').
Pre-block Motion block before the blending curve (S1 – E1)
Post-block Motion block after the blending curve (S2 – E2)

Corner distance Distance (d) from the start point of the blending curve (E1’) to the programmed target point (E1), see figure below.



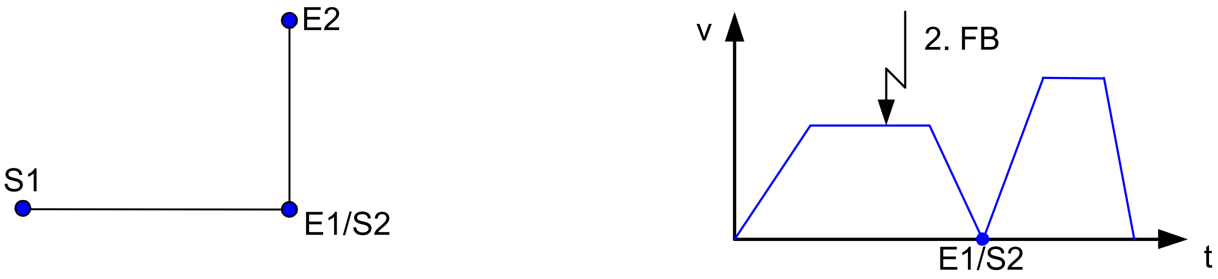
Corner deviation The shortest distance between the programmed corner point (E1/S2) and the blending curve, see figure below.



2.4.2 Transition mode: TMNone

The jobs of the function blocks which generate a motion are not modified and no transition curve is inserted. This is the only valid transition mode when the value **mcBuffered** was assigned to the “BufferMode” input.

When this transition mode is selected, the same motion is obtained for all permitted values at the “BufferMode” input as if the value **mcBuffered** had been applied to the “BufferMode” input (see figure below).

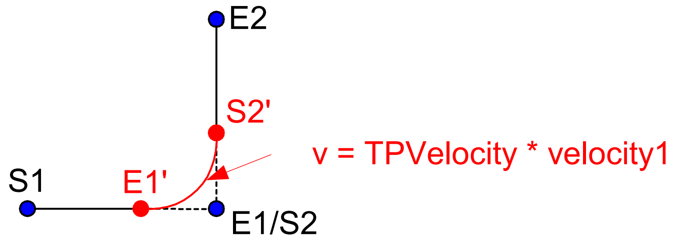


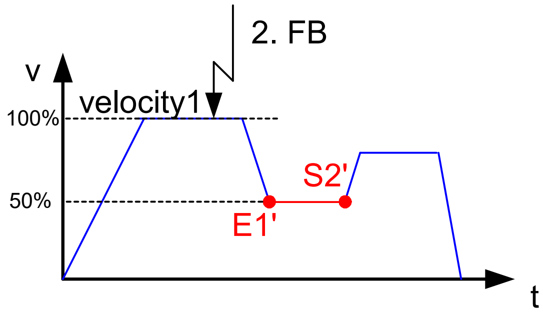
2.4.3 Transition mode: TMConstantVelocity

This type of transition mode generates the transition curve so that the path is moved at a constant specified velocity. The velocity is then defined by the percentage weighting of **TPVelocity** to the velocity which was commanded by the previous function block. I.e., when **TPVelocity** has a value of 50%, the transition curve is moved at half the velocity at which the motion of the function block immediately before the last function block was executed.

Due to the curvature of the transition curve and the maximum permitted acceleration of the axes involved in the motion, it may happen that the specified velocity is not reached.

The graphic below shows the transition curve when **mcBuffered** is transferred at the “Buffer-Mode“ input.





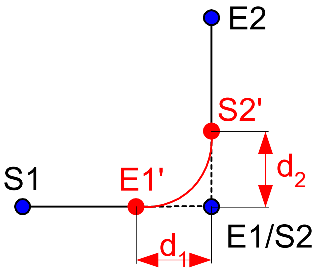
Weighting takes place using the parameters listed in the table.

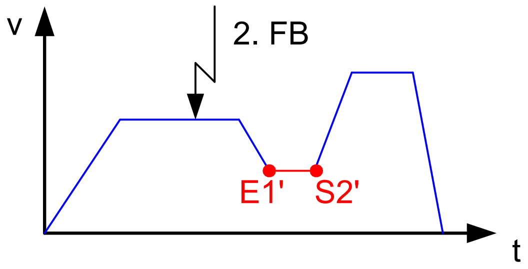
Transition parameters	Description
TransitionParameter[0]	Velocity at which the machine moves over the transition curve. It is specified as the weighting factor relating to the velocity of the previous motion job.

2.4.4 Transition mode: TMCornerDistance

Once it is known where the programmed path can be left with two successive function blocks and when this path can be resumed, the user can explicitly specify the distance between these two points and the original corner point of the programmed path for this transition mode. The original motion jobs can then be shortened by these distances.

The motion velocity on the transition curve is the velocity of the first of two successive function blocks. Due to the curvature of the transition curve and the maximum permitted acceleration of the axes involved in the motion, it may happen that the specified velocity is not reached.





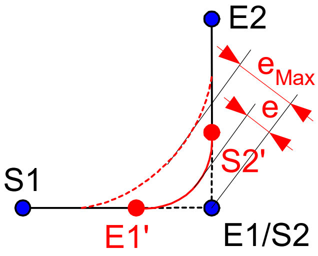
Transition parameters	Description
TransitionParameter[0]	This parameter defines the point at which the programmed path can be left. It is specified as the distance (d_1) to the corner at which the following motion starts.
TransitionParameter[1]	This parameter defines the point at which the programmed path can be resumed. It is specified as the distance (d_2) between the point at which the following motion starts and the return point.

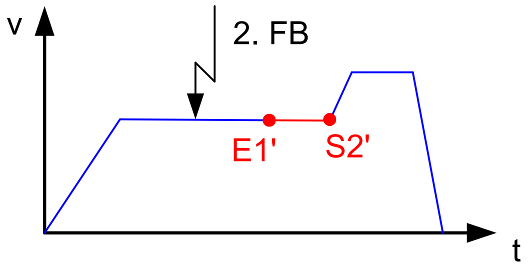
2.4.5 Transition mode: TMMaxCornerDeviation

With this transition mode the distances by which the motions of two successive function blocks are shortened are automatically determined based on geometric calculations. The boundary condition for the calculation is that the calculated deviation of the transition curve is not greater than the maximum specified deviation.

The motion velocity on the transition curve is the velocity of the first of two successive function blocks. Due to the curvature of the transition curve and the maximum permitted acceleration of the axes involved in the motion, it may happen that the specified velocity is not reached.

If it is possible to maintain the velocity of the previous function block on the transition curve, the actual corner deviation is smaller than the specified maximum permitted corner deviation.



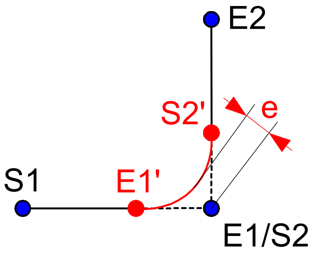


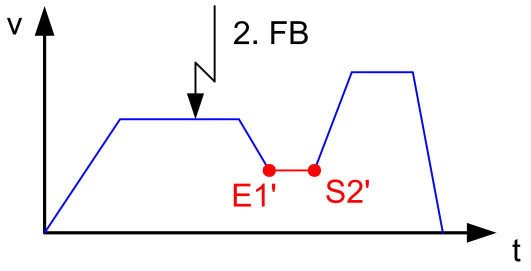
Transition parameters	Description
TransitionParameter[0]	If the maximum permitted corner deviation is (e_{Max}); the shortest distance between the corner point of the path motion and the new calculated transition curve.

2.4.6 Transition mode: TMConstDeviation

With this transition mode the distances by which the motions of two successive function blocks are shortened are automatically determined based on geometric calculations. The boundary condition for the calculation is that the calculated deviation of the transition curve is the specified deviation.

The motion velocity on the transition curve is the velocity of the first of two successive function blocks. Due to the curvature of the transition curve and the maximum permitted acceleration of the axes involved in the motion, it may happen that the specified velocity is not reached.

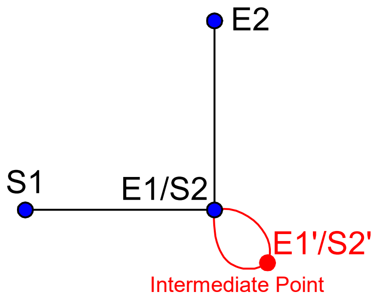


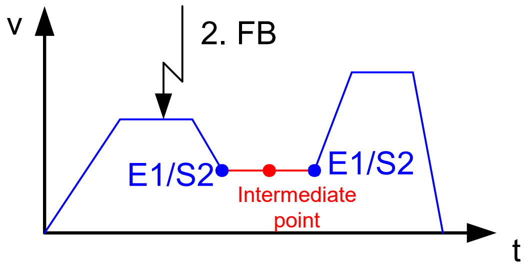


Transition parameters	Description
TransitionParameter[0]	Is the corner deviation (e) and the shortest distance between the corner point and the path motion and the new calculation transition curve.

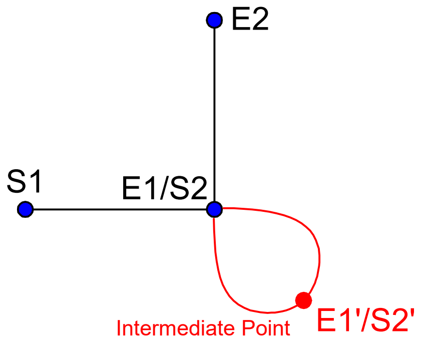
2.4.7 Transition mode: TMIntermediatePoint

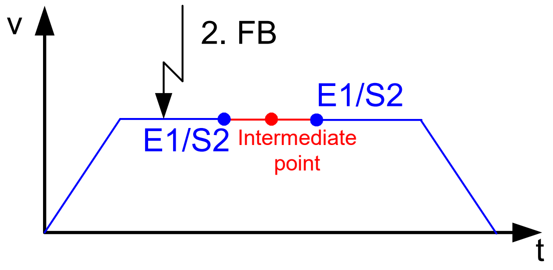
This transition mode generates a transition curve by which the compensation motion passes through an intermediate point defined by the user. The intermediate point is defined by its X, Y and Z coordinates.





The second example shows how to maintain the velocity constant on the transition curve by selecting the intermediate point at a suitable distance from E1/S2.





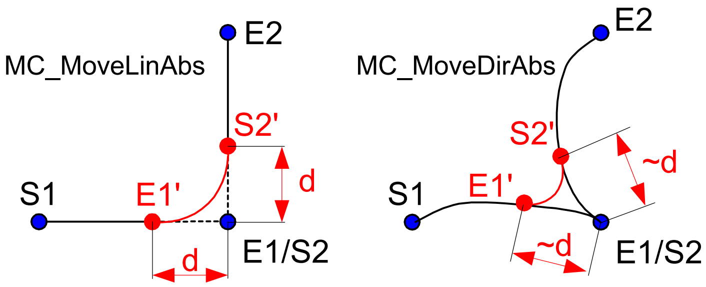
Transition parameters	Description
TransitionParameter[0]	X coordinate of intermediate point
TransitionParameter[1]	Y coordinate of intermediate point
TransitionParameter[2]	Z coordinate of intermediate point

2.4.8 Transition mode: TMDynamicOptimized

This transition mode optimises the axis-specific acceleration and jerk. No geometric path is calculated.

The only parameter that must be specified when this transition mode is used defines the distance from the point where the transition starts up to the corner point of the successive motions. This is also the distance from the corner point to the point where the transition motion should end. The distance d can be interpreted as the radius of a ball with the corner point E1/E2 as its centre point. Outside this ball, the tool centre point follows the programmed path. The transition takes place inside the ball.

This transition mode is recommended for point-to-point (PTP) motions (MC_MoveDirAbs).



Transition parameters	Description
TransitionParameter[0]	The distance (d) is from the point where the programmed path is left up to the corner point of the successive motions. If this transition mode is used in conjunction with an MC_MoveDirAbs FB and a robot (PTP motions), the distance d is not exactly the distance between E1’ and E1 or between S2 and S2’.

3 Overview of the PLCopen FBs

The table below provides an overview of the availability of PLCopen FBs in PLC and drive systems supported by ISG. The PLCopen Part 4 specification provides the option to use a short form for function block identifiers. The library makes use of these options.

Function block	**PLC systems**		
	3S	**TwinCAT**	**KW**
MC_AddAxisToGrp [▸ 19]	SIMU -	SIMU -	SIMU KUKA
MC_GrpContinue [▸ 21]	SIMU -	SIMU -	SIMU KUKA
MC_GrpDisable [▸ 21]	SIMU -	SIMU -	SIMU KUKA
MC_GrpEnable [▸ 23]	SIMU -	SIMU -	SIMU KUKA
MC_GrpHalt [▸ 24]	SIMU -	SIMU -	SIMU KUKA
MC_GrpInterrupt [▸ 25]	SIMU -	SIMU -	SIMU KUKA
MC_GrpReadActPos [▸ 26]	SIMU -	SIMU -	SIMU KUKA
MC_GrpReadError [▸ 28]	SIMU -	SIMU -	SIMU KUKA
MC_GrpReadInfo [▸ 28]	SIMU -	SIMU -	SIMU KUKA
MC_GrpReadStatus [▸ 30]	SIMU -	SIMU -	SIMU KUKA
MC_GrpReset [▸ 31]	SIMU -	SIMU -	SIMU KUKA
MC_GrpSetOverride [▸ 32]	SIMU -	SIMU -	SIMU KUKA
MC_GrpStop [▸ 35]	SIMU -	SIMU -	SIMU KUKA
MC_MoveCircAbs [▸ 37]	SIMU -	SIMU -	SIMU KUKA

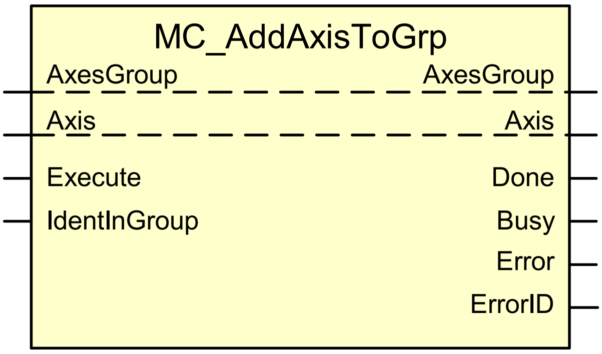
MC_MoveCircRel [▸ 41]	SIMU -	SIMU -	SIMU KUKA
MC_MoveDirAbs [▸ 45]	SIMU -	SIMU -	SIMU KUKA
MC_MoveLinAbs [▸ 47]	SIMU -	SIMU -	SIMU KUKA
MC_MoveLinRel [▸ 49]	SIMU -	SIMU -	SIMU KUKA
MC_MovePath [▸ 51]	SIMU -	SIMU -	SIMU KUKA
MC_RemAxisFromGrp [▸ 52]	SIMU -	SIMU -	SIMU KUKA
MC_SetCartTrans [▸ 54]	SIMU -	SIMU -	SIMU KUKA
MC_SetKinTrans [▸ 57]	SIMU -	SIMU -	SIMU KUKA
MC_UngroupAllAxes [▸ 58]	SIMU -	SIMU -	SIMU KUKA

3.1 MC_AddAxisToGrp

The function block adds an axis to an axis group. This is an administrative FB, i.e. no motion job is generated.

The added axis is identified within the axis group by a unique name and an index. The two values are transferred to the FB by the data structure IDENT_IN_GROUP_REF (see Section Axis group reference AXES_GROUP_REF [▸ 8]).

Block diagram



FB parameters

VAR_IN_OUT		
Variable name	**Data type**	**Description**
AxesGroup	AXES_GROUP_REF	Axis group reference
Axis	AXIS_REF	Axis reference
VAR_INPUT		
Variable name	**Data type**	**Description**
Execute	BOOL	The axis is added to the axis group on a rising edge.
IdentInGroup	IDENT_IN_GROUP_REF	Identification of the axis in the axis group.
VAR_OUTPUT		
Variable name	**Data type**	**Description**
Done	BOOL	TRUE indicates that axis was added to the axis group.
Busy	BOOL	TRUE indicates that the FB is executing a job.
Error	BOOL	TRUE indicates that an error occurred.
ErrorID	WORD	Error code

Notice

The following conditions must be met to permit this function block to work correctly when the axis reference at the “Axis” input is an axis:

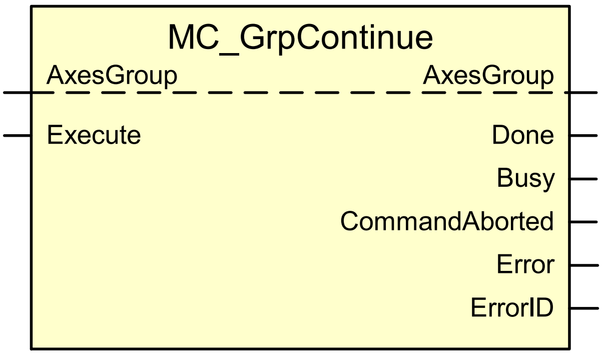
The spindle axis must be notified to the NC channel by specifying it in the channel parameter list (see [FCT-S1//NC spindle]).

The identifier to address the spindle in the channel and defined by the channel parameter P-CHAN-00007 must be formed from the letter “S” followed by the logical axis number of the spindle (P-CHAN-00036). For example, if P-CHAN-00036 = 2, then S2 must be specified for P-CHAN-00007.

3.2 MC_GrpContinue

This function block cancels feedhold that was commanded by a “MC_GrpInterrupt”. Therefore command values are output to the axes of the axis group when a motion command was interrupted by the “MC_GrpInterrupt” command or a new command becomes active.

Block diagram



FB parameters

VAR_IN_OUT		
Variable name	**Data type**	**Description**
AxesGroup	AXES_GROUP_REF	Axis group reference
VAR_INPUT		
Variable name	**Data type**	**Description**
Execute	BOOL	Feedhold is cancelled on a rising edge.
VAR_OUTPUT		
Variable name	**Data type**	**Description**
Done	BOOL	TRUE notifies that command values can now be sent to the axes of an axis group.
Busy	BOOL	TRUE indicates that the FB is executing a job.
CommandAborted	BOOL	TRUE indicates that the command was aborted by another command.
Error	BOOL	TRUE indicates that an error occurred.
ErrorID	WORD	Error code

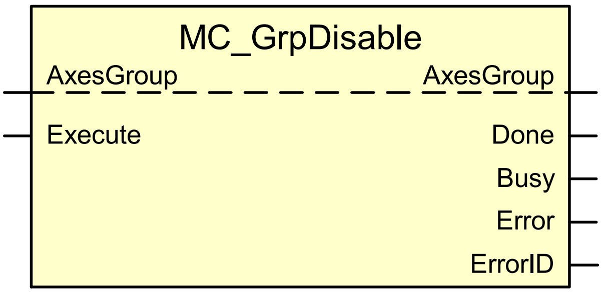
Behaviour of the FB:

- Limits for acceleration and velocity are the configured values in the channel or axis parameter lists.

3.3 MC_GrpDisable

This is an administrative function block which sets the axis group to the “GroupDisabled” state.

Block diagram



FB parameters

VAR_IN_OUT		
Variable name	**Data type**	**Description**
AxesGroup	AXES_GROUP_REF	Axis group reference
VAR_INPUT		
Variable name	**Data type**	**Description**
Execute	BOOL	Rising edge sets the axes group to the “GroupDisabled” state.
VAR_OUTPUT		
Variable name	**Data type**	**Description**
Done	BOOL	TRUE indicates that axes group is set to the “GroupDisabled” state.
Busy	BOOL	TRUE indicates that the FB is executing a job.
Error	BOOL	TRUE indicates that an error occurred.
ErrorID	WORD	Error code

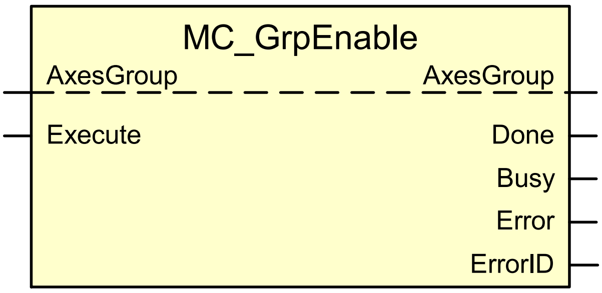
Behaviour of the FB:

- The command of this FB can only be executed successfully if the axis group is in the “GroupStandby“ state or is already in the “GroupDisabled“ state.
- In the “GroupDisabled” state, only commands of the FBs MC_AddAxisToGrp [▸ 19], MC_RemAxisFromGrp [▸ 52], MC_UngroupAllAxes [▸ 58] or MC_GrpEnable [▸ 23] can be executed successfully. No motion commands are therefore possible.

3.4 MC_GrpEnable

This is an administrative FB that sets the axis group to the “GroupStandby” state. This function block must always be commanded whenever the axis group is in the “GroupDisabled” state since no motion job is accepted in the “GroupDisabled” state.

Block diagram



FB parameters

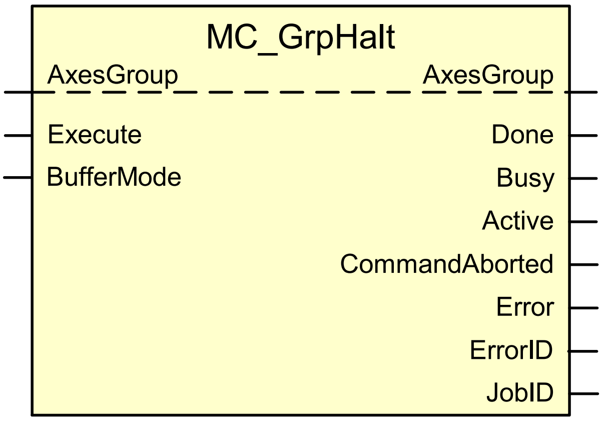
VAR_IN_OUT		
Variable name	**Data type**	**Description**
AxesGroup	AXES_GROUP_REF	Axis group reference
VAR_INPUT		
Variable name	**Data type**	**Description**
Execute	BOOL	Rising edge sets the axis group to the “GroupDisabled” state.
VAR_OUTPUT		
Variable name	**Data type**	**Description**
Done	BOOL	TRUE indicates that axis group is set to the “GroupDisabled” state.
Busy	BOOL	TRUE indicates that the FB is executing a job.
Error	BOOL	TRUE indicates that an error occurred.
ErrorID	WORD	Error code

3.5 MC_GrpHalt

This function block leads to a controlled motion stop. It aborts every ongoing command by other motion FBs.

The axis group changes to the “GroupMoving state until velocity 0 is reached. When the “Done” is set, the axis group state changes to “GroupStandby”.

Block diagram



FB parameters

VAR_IN_OUT		
Variable name	**Data type**	**Description**
AxesGroup	AXES_GROUP_REF	Axis group reference
VAR_INPUT		
Variable name	**Data type**	**Description**
Execute	BOOL	Starts the command on the rising edge.

BufferMode	MC_BUFFER_MODE	The input defines when a job is activated provided that other jobs are already active when the FB is commanded or are waiting for execution. The following modes are supported:
		mcAborting = 0
		mcBuffered = 1

VAR_OUTPUT		
Variable name	**Data type**	**Description**
Done	BOOL	0 velocity was reached; the axis is at standstill.
Busy	BOOL	TRUE indicates that the FB is executing a job.
Active	BOOL	The command is not yet completed.
CommandAborted	BOOL	The stop job was aborted by another job.
Error	BOOL	TRUE indicates that an error occurred.
ErrorID	WORD	Error code
JobID	UDINT	Ordinal number of the last job sent by the FB.

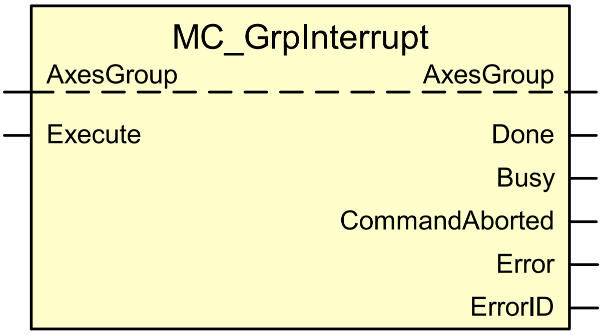
3.6 MC_GrpInterrupt

The function block leads to a controlled motion stop. It does not abort an active motion job. This is why the “CommandAborted“ output is not set at the function block whose job is active at this time. However, the “Active” output of this FB is reset and only the “Busy” output remains TRUE.

The axis group itself retains the state which was indicated at the time when the motion was aborted.

To end the effect of the function block, a job is sent by an instance of the MC_GrpContinue [▸ 21] FB.

Block diagram



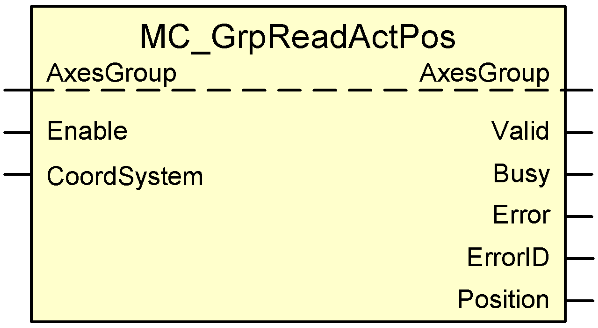
FB parameters

VAR_IN_OUT		
Variable name	**Data type**	**Description**
AxesGroup	AXES_GROUP_REF	Axis group reference
VAR_INPUT		
Variable name	**Data type**	**Description**
Execute	BOOL	Starts the controlled motion stop on the rising edge.
VAR_OUTPUT		
Variable name	**Data type**	**Description**
Done	BOOL	TRUE indicates that the active job was interrupted and the command to stop the axis was activated.
Busy	BOOL	TRUE indicates that the FB is executing a job.
CommandAborted	BOOL	The stop job was aborted by another job.
Error	BOOL	TRUE indicates that an error occurred.
ErrorID	WORD	Error code

3.7 MC_GrpReadActPos

This function block supplies the current axis positions for the axes of an axis group.

Block diagram



FB parameters

VAR_IN_OUT		
Variable name	**Data type**	**Description**
AxesGroup	AXES_GROUP_REF	Axis group reference
VAR_INPUT		
Variable name	**Data type**	**Description**
Enable	BOOL	When “Enable” is TRUE, this FB supplies the current position of all axes in an axis group in the PCS at the Position output.

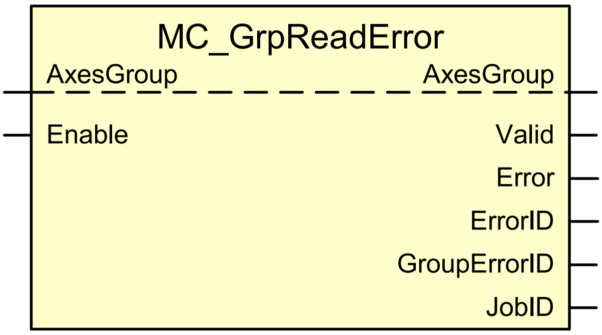
CoordSystem	MC_COORDINATE_SYSTEM	Defines the coordinate system used to display the current axis positions.
		mcACS = 0: Axis coordinate system
		mcPCS = 2: Programming coordinate system
		The input is defaulted to the value mcPCS.

VAR_OUTPUT		
Variable name	**Data type**	**Description**
Valid	BOOL	When this output is TRUE, the values at the Position output are valid.
Busy	BOOL	A long as the output is TRUE, the FB updates the values at the “Position” output.
Error	BOOL	TRUE indicates that an error occurred.
ErrorID	WORD	Error code
Position	ARRAY [0..HLI_CH_AX_MAXIDX] OF LREAL	Array with current positions Default unit [0.1 µm] If CoordSystem = mcACS, the axis coordinate system is used for the positions. The positions of gantry slave axes are also displayed. If CoordSystem = mcPCS, the programming coordinate system is used for the positions. The positions of gantry slave axes are not displayed.

3.8 MC_GrpReadError

This function block supplies the error number when the axis group is in the “GroupErrorStop state. This is an administrative FB, i.e. no motion job is generated.

Block diagram



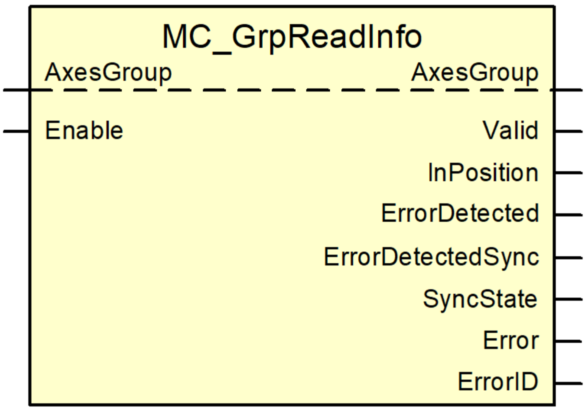
FB parameters

VAR_IN_OUT		
Variable name	**Data type**	**Description**
AxesGroup	AXES_GROUP_REF	Axis group reference
VAR_INPUT		
Variable name	**Data type**	**Description**
Enable	BOOL	As long as input is TRUE, the current error is indicated at the “GroupErrorID” output when the axis is in the “GroupErrorStop” state.
VAR_OUTPUT		
Variable name	**Data type**	**Description**
Valid	BOOL	TRUE if the FB is active.
Error	BOOL	TRUE indicates that an error occurred.
ErrorID	WORD	Error identifier.
GroupErrorID	UDINT	Error identifier of the axis group.
JobID	UDINT	Ordinal number of the command when the displayed error occurred.

3.9 MC_GrpReadInfo

This function block supplies information concerning an axis group. This is an administrative FB, i.e. no motion job is generated.

Block diagram



FB parameters

VAR_IN_OUT		
Variable name	**Data type**	**Description**
AxesGroup	AXES_GROUP_REF	Axis group reference
VAR_INPUT		
Variable name	**Data type**	**Description**
Enable	BOOL	If TRUE, supplies cyclic information about the axis group.
VAR_OUTPUT		
Variable name	**Data type**	**Description**
Valid	BOOL	TRUE when the information is available.
InPosition	BOOL	Target position is reached.
ErrorDetected	BOOL	Is TRUE when an error is detected in the motion controller. This means that the emergency strategy is initiated.
ErrorDetectedSync	BOOL	Is TRUE when an error related to conveyor belt synchronisation is detected in the motion controller. This means that the emergency strategy is initiated.
SyncState	UDINT	Current state of an axis group relating to a command that leads to synchronisation with a conveyor belt. Table 1 shows the possible values for the synchronisation state.
Error	BOOL	TRUE indicates that an error occurred.
ErrorID	WORD	Error code

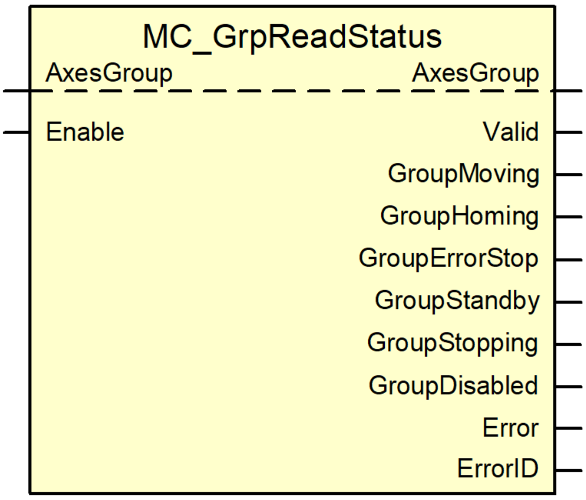
Values at the “SyncState” output

Constant	Value	Note
HLI_GRP_NO_SYNCH	0	No synchronisation command is active.
HLI_GRP_SYNCH_IN	1	The axis group executes a synchronisation command
HLI_GRP_SYNCH_LINKED	2	The axis group motions to synchronise with a conveyor belt axis are completed and the conveyor belt axis and the axis group now move synchronously.
HLI_GRP_SYNCH_OUT	3	The state commanding an axis group to move synchronously with a conveyor belt is cancelled.
HLI_GRP_SYNCH_STOPPED	4	An error occurred in an axis group that moves synchronously with a conveyor belt axis. The axis group starts its strategy in the event of an error.

3.10 MC_GrpReadStatus

This function block supplies the axis group state depending on the axis group state diagram defined in PLCopen Part 4 specification. This is an administrative FB, i.e. no motion job is generated.

Block diagram



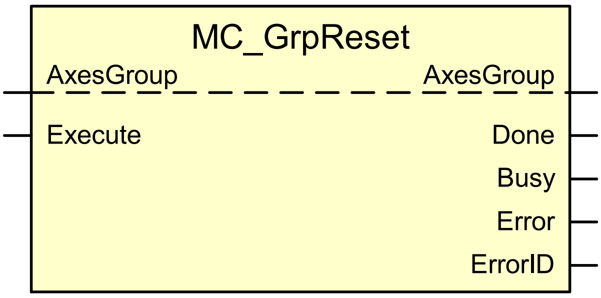
FB parameters

VAR_IN_OUT		
Variable name	**Data type**	**Description**
AxesGroup	AXES_GROUP_REF	Axis group reference
VAR_INPUT		
Variable name	**Data type**	**Description**
Enable	BOOL	Supplies state values cyclically when TRUE.
VAR_OUTPUT		
Variable name	**Data type**	**Description**
Valid	BOOL	TRUE when state values are available.
GroupMoving	BOOL	Axis group is in the “GroupMoving” state
GroupHoming	BOOL	Axis group is in the “GroupHoming“ state
GroupErrorStop	BOOL	Axis group is in the “GroupErrorStop“ state
GroupStandby	BOOL	Axis group is in the “GroupStandby“ state
GroupStopping	BOOL	Axis group is in the “GroupStopping“ state
GroupDisabled	BOOL	Axis group is in the “GroupDisabled“ state
Error	BOOL	TRUE indicates that an error occurred.
ErrorID	WORD	Error code

3.11 MC_GrpReset

This function block executes a rest on all axes in the specified axis group. The axis group then changes from the “GroupErrorStop” state to the “GroupStandby” state.

Block diagram



FB parameters

VAR_IN_OUT		
Variable name	**Data type**	**Description**
AxesGroup	AXES_GROUP_REF	Axis group reference
VAR_INPUT		
Variable name	**Data type**	**Description**
Execute	BOOL	A axis group is reset on a rising edge.
VAR_OUTPUT		
Variable name	**Data type**	**Description**
Done	BOOL	The reset was successfully executed.
Busy	BOOL	TRUE indicates that the FB is executing a job.
Error	BOOL	TRUE indicates that an error occurred.
ErrorID	WORD	Error code

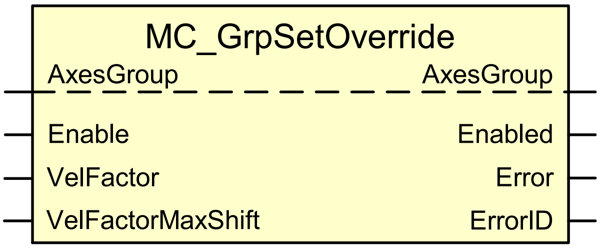
Behaviour of the FB:

- The function block can only be executed successfully if the axis group is in the “GroupErrorStop” state. Otherwise the “Error” output is TRUE and “ErrorID” indicates the error identifier P-ERR-46007 (ERR_PO_GRP_TNA_NOT_ERROR).

3.12 MC_GrpSetOverride

The FB MC_GrpSetOverride specifies a weighting factor which acts on the programmed path velocity of the specified axis group.

Block diagram



FB parameters

VAR_IN_OUT		
Variable name	**Data type**	**Description**
AxesGroup	AXES_GROUP_REF	Axis group reference
VAR_INPUT		
Variable name	**Data type**	**Description**
Enable	BOOL	If TRUE, the value at the "VelFactor" input is applied to the programmed velocity.
VelFactor	LREAL	Weighting factor for the programmed path velocity Range [0.0 P-CHAN-00056 / 1000]. See Note 1
VelFactorMaxShift	LREAL	Maximum change of the weighting factor. Value range [0.0 to P-CHAN-00056 / 1000] See Note 2
VAR_OUTPUT		
Variable name	**Data type**	**Description**
Enabled	BOOL	TRUE indicates that the velocity weighting factor is applied to the programmed path velocity.
Error	BOOL	TRUE indicates that an error occurred.
ErrorID	WORD	Error code

Behaviour of the FB:

- Under no circumstance does the FB change the axes group state. This means that the value 0.0 for "VelFactor" leads to a stop of the axes in the axis group but the state of the axis group does not change after "GroupStop".
- The value at "VelFactor" retains its validity until a new value is set. Accordingly, if the "Enable" input is set to FALSE, the weighting factor which was last TRUE at "Enable" continues to be valid.

Value range and determination of the "VelFactor" input

- The default value of the "VelFactor" input is 1.0.
- P-CHAN-00056 is the maximum permitted override value. Assuming that the maximum allowed override value in an application may not exceed 100%, the parameter P-CHAN-00056 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-CHAN-00056/1000, the value is limited to P-CHAN-00056/1000 (typical upper limit value is 200%).

Notice

No message is output if an excessive "VelFactor" value is set.

The value is limited to P-CHAN-00056/1000.

Value range and determination of the “VelFactorMaxShift” input

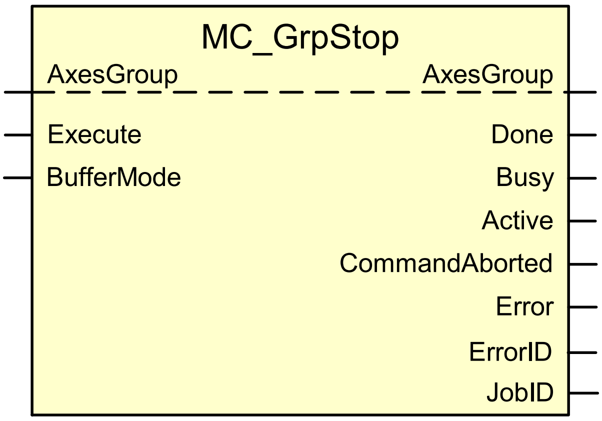
- The “VelFactorMaxShift“ input is defaulted to the value 1.0.
- If the value at the “VelFactor” input changes, the current effective velocity factor sent to the motion controller is increased or decreased by the maximum value at the “VelFactorMaxShift” input each time the function block is invoked.
- The weighting factor sent to the motion controller is increased or decreased until the same velocity weighting factor as specified at the “VelFactor” input acts on the motion controller. To achieve this, the function block must be invoked at least until its “Enabled” output indicates TRUE again after a change occurs at the “VelFactor” input.

3.13 MC_GrpStop

The function block leads to a controlled motion stop. It aborts every ongoing command by other motion FBs.

The axis group changes to the “GroupStopping” state until velocity 0 is reached. If the “Done” and “Execute” outputs are set to FALSE, the axis group state changes to “GroupStandby”. The function block is not interruptible.

Block diagram



FB parameters

VAR_IN_OUT		
Variable name	**Data type**	**Description**
AxesGroup	AXES_GROUP_REF	Axis group reference
VAR_INPUT		
Variable name	**Data type**	**Description**
Execute	BOOL	Starts the command on the rising edge.

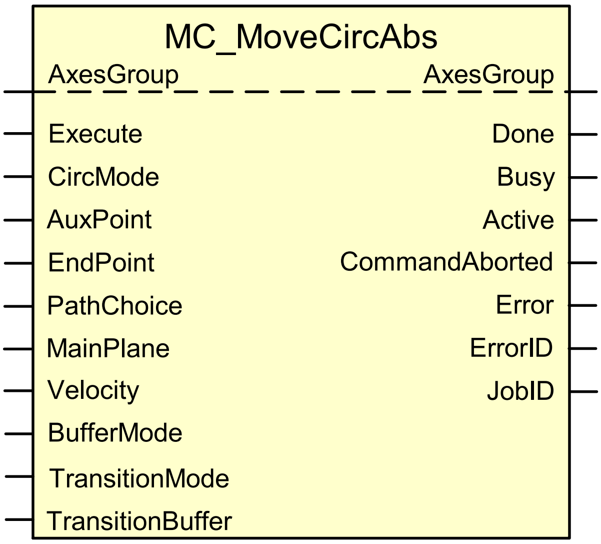
BufferMode	MC_BUFFER_MODE	The input defines when a job is activated provided that other jobs are already active when the FB is commanded or are waiting for execution. The following modes are supported:
		mcAborting = 0
		mcBuffered = 1

VAR_OUTPUT		
Variable name	**Data type**	**Description**
Done	BOOL	TRUE indicates that 0 velocity was reached. The axes are at standstill.
Busy	BOOL	TRUE indicates that the function block is executing a job.
Active	BOOL	The command is not yet completed.
CommandAborted	BOOL	The command to stop was aborted by another job.
Error	BOOL	TRUE indicates that an error occurred.
ErrorID	WORD	Error code
JobID	UDINT	Ordinal number of the last job sent by the FB.

3.14 MC_MoveCircAbs

This function block commands a circular interpolated motion of the axes of an axis group by specifying absolute target positions.

Block diagram



FB parameters

VAR_IN_OUT		
Variable name	**Data type**	**Description**
AxesGroup	AXES_GROUP_REF	Axis group reference
VAR_INPUT		
Variable name	**Data type**	**Description**
Execute	BOOL	The circular interpolated motion of the axes starts on a rising edge depending on the positions specified as absolute coordinates.

CircMode	MC_CIRC_MODE	This input defines the meaning of the values at the “AuxPoint” and “PathChoice” inputs. The permitted values for this input are:
		mcCenter = 1
		mcRadius = 2
		and the are described in more retail in Meaning of the “CircMode“ input with absolute programming [▸ 39].

AuxPoint	ARRAY [0..HLI_CH_AX_MAXIDX] OF LREAL	Absolute coordinates of a point in the current coordinate system. The type of additional point is defined by the value at the “CircMode“ input. Default unit [0.1 µm]
EndPoint	ARRAY [0..HLI_CH_AX_MAXIDX] OF LREAL	Absolute coordinates of the end point of a circular motion in the current coordinate system. Default unit [0.1 µm]
PathChoice	MC_CIRC_PATHCHOICE	Determines whether the motion is travelled from the start point to the end point in a clockwise or counter clockwise direction on the circular path defined by the values of “CircMode”, “AuxPoint” and “EndPoint”. The permitted values for this input are described in the Meaning of the “PathChoice“ input [▸ 40] input.

MainPlane	MCV_MAIN_PLANE	The value of this input defines the main plane of the programming coordinate system where the circle is interpolated.
		The permitted values for this input are:
		mcvMpXY = 0
		mcvMpZX = 1
		mcvMpYZ = 2

Velocity	LREAL	Maximum path velocity of the motion. This value need not be reached. Default unit [10^{-3} °/ s]

BufferMode	MC_BUFFER_MODE	The input defines when a job is activated provided that other jobs are already active when the FB is commanded or are waiting for execution. It also defines the path generated at the transition between 2 jobs. The following values are possible:
		mcAborting = 0
		mcBuffered = 1
		mcBlendingPrev = 3

TransitionMode	UDINT	0: TMNone (no transition motion is inserted, default setting) 2: TMConstantVelocity 3: TMCornerDistance 4: TMMaxCornerDeviation 10: TMConstDeviation 11: TMIntermediatePoint 12: TMDynamicOptimized
TransitionParameter	MCV_ARRAY_TRANS_PARAM	See Section Path blending and transition modes [▸ 11]
VAR_OUTPUT		
Variable name	**Data type**	**Description**
Done	BOOL	TRUE indicates that the axes have reached their end position.
Busy	BOOL	TRUE indicates that the function block is executing a job.
Active	BOOL	TRUE indicates that the function block sent the job to the MC.
CommandAborted	BOOL	TRUE indicates that the currently active command sent by the FB was aborted by another job.
Error	BOOL	TRUE indicates that an error occurred.
ErrorID	WORD	Error code
JobID	UDINT	Ordinal number of the last job sent by the FB.

Notice

If, due to the values specified at the “AuxPoint“ and “EndPoint“ inputs, a distance results in the direction of the 3rd main axis of the orthogonal coordinate system in the plane of the circular path, the application of this FB leads to a helical motion (see [PROG//Helikalinterpolation]).

3.14.1 Meaning of the “CircMode“ input with absolute programming

The table below lists values that can be applied to the “CircMode” input when the function block MC_MoveCircAbs is used and describes their implication on the interpretation of the values at the “AuxPoint” inputs.

Values for the “CircMode” input with MC_MoveCircAbs

Constant	Value	Description
mcCenter	1	The coordinates at the “AuxPoint” input are absolute coordinates of the circle centre point. The coordinates at the “EndPoint” input are absolute coordinates of the end point of the circular path.
mcRadius	2	The value specified at the “AuxPoint[0]“ input is the length of the radius. Since there are 2 possible motion paths with this type of programming, the sign of the “AuxPoint[0]“ value and therefore the radius defines whether the shorter or the longer motion path is programmed. See also [PROG//Radius programming] The coordinates specified at the “EndPoint” input are absolute coordinates of the end point of the circular path.

3.14.2 Meaning of the “PathChoice“ input

Based on a right-handed Cartesian coordinate system where two coordinate axes are positioned in the plane of the circular path, a “counter clockwise” motion on the circular path corresponds to the mathematical positive direction of rotation.

This equates to the “right-hand rule”: The right hand comprises the 3rd main axis with the thumb pointing in the direction of the positive 3rd main axis; the curvature of the fingers representing the 3rd main axis corresponds to the positive direction of rotation.

The values defined in the table below can be specified in the FB at the “PathChoice“ input to determine the motion on the circular path.

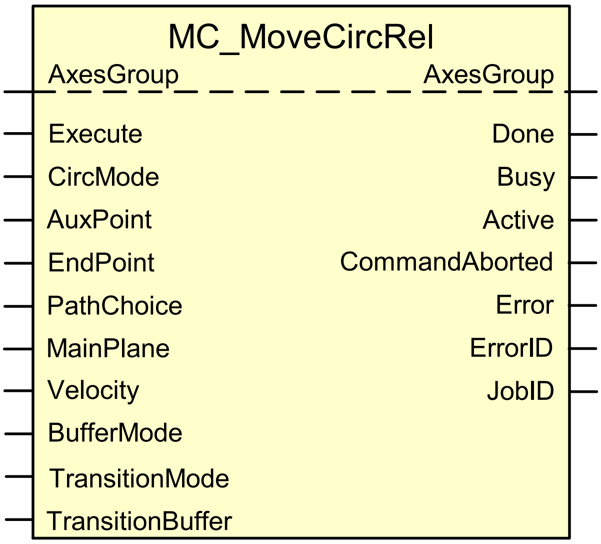
Values for the “PathChoice” input with MC_MoveCircAbs

Constant	Value	Description
mcClockwise	0	Passing the circular path in the clockwise direction
mcCounterClockwise	1	Passing the circular path in the counter clockwise direction

3.15 MC_MoveCircRel

This function block commands a circular interpolated motion of the axes of an axis group by specifying the target position by specifying relative distances.

Block diagram



FB parameters

VAR_IN_OUT		
Variable name	**Data type**	**Description**
AxesGroup	AXES_GROUP_REF	Axis group reference
VAR_INPUT		
Variable name	**Data type**	**Description**
Execute	BOOL	The linear interpolated motion of the axes starts on a rising edge corresponding to the specified relative distances.

CircMode	MC_CIRC_MODE	This input defines the meaning of the values at the “AuxPoint” and “PathChoice” inputs. The permitted values for this input are:
		mcCenter = 1
		mcRadius = 2
		and they are described in more detail in Meaning of the “CircMode“ input with relative programming [▸ 43].

AuxPoint	ARRAY [0..HLI_CH_AX_MAXIDX] OF LREAL	Relative coordinates of a point in the current coordinate system. The type of additional point is defined by the value at the “CircMode“ input. Default unit [0.1 µm]
EndPoint	ARRAY [0..HLI_CH_AX_MAXIDX] OF LREAL	Relative coordinates of the end point of a circular motion in the current coordinate system. Default unit [0.1 µm]
PathChoice	MC_CIRC_PATHCHOICE	Determines whether the motion is travelled from the start point to the end point in a clockwise or counter clockwise direction on the circular path defined by the values of “CircMode”, “AuxPoint” and “EndPoint”. The permitted values for this input are described in the Meaning of the “PathChoice“ input [▸ 44] input.

MainPlane	MCV_MAIN_PLANE	The value of this input defines the main plane of the programming coordinate system where the circle is interpolated.
		The permitted values for this input are:
		mcvMpXY = 0
		mcvMpZX = 1
		mcvMpYZ = 2

Velocity	LREAL	Maximum path velocity of the motion. This value need not be reached. Default unit [10^{-3} °/ s]

BufferMode	MC_BUFFER_MODE	The input defines when a job is activated provided that other jobs are already active when the FB is commanded or are waiting for execution. It also defines the path generated at the transition between 2 jobs. The following values are possible:
		mcAborting = 0
		mcBuffered = 1
		mcBlendingPrev = 3

TransitionMode	UDINT	0: TMNone (no transition motion is inserted, default setting) 2: TMConstantVelocity 3: TMCornerDistance 4: TMMaxCornerDeviation 10: TMConstDeviation 11: TMIntermediatePoint 12: TMDynamicOptimized
TransitionParameter	MCV_ARRAY_TRANS_PARAM	See Section Path blending and transition modes [▶ 11]
VAR_OUTPUT		
Variable name	**Data type**	**Description**
Done	BOOL	TRUE indicates that the axes have reached their end position.
Busy	BOOL	TRUE indicates that the function block is executing a job.
Active	BOOL	TRUE indicates that the function block sent the command to the MC.
CommandAborted	BOOL	TRUE indicates that the currently active command sent by the FB was aborted by another command.
Error	BOOL	TRUE indicates that an error occurred.
ErrorID	WORD	Error code
JobID	UDINT	Ordinal number of the last job sent by the FB.

Notice

If, due to the values specified at the “AuxPoint“ and “EndPoint“ inputs, a distance results in the direction of the 3rd main axis of the orthogonal coordinate system in the plane of the circular path, the application of this FB leads to a helical motion (see [PROG//Helical interpolation]).

3.15.1 Meaning of the “CircMode“ input with relative programming

The table below lists values that can be applied to the “CircMode” input when the function block MC_MoveCircRel is used and describes their implication on the interpretation of the values at the “AuxPoint” inputs.

Values for the “CircMode” input with MC_MoveCircRel

Constant	Value	Description
mcCenter	1	The coordinates at the “AuxPoint” input are circle centre point coordinates specified relative to the start point of the circular motion. The coordinates at the “EndPoint” input are coordinates of the end point of the circular path specified relative to the start point of the circular motion.
mcRadius	2	The value specified at the “AuxPoint[0]“ input is the length of the radius. Since there are 2 possible motion paths with this type of programming, the sign of the “AuxPoint[0]“ value and therefore the radius defines whether the shorter or the longer motion path is programmed. See also [PROG//Radius programming] The coordinates at the “EndPoint” input are relative coordinates of the end point of the circular path relative to the start point of the circular motion.

3.15.2 Meaning of the “PathChoice“ input

Based on a right-handed coordinate system where two coordinate axes are positioned in the plane of the circular path, a “counter clockwise” motion on the circular path corresponds to the mathematical positive direction of rotation.

This equates to the “right-hand rule”: The right hand comprises the 3rd main axis with the thumb pointing in the direction of the positive 3rd main axis; the curvature of the fingers representing the 3rd main axis corresponds to the positive direction of rotation.

The values defined in the table below can be specified in the FB at the “PathChoice“ input to determine the motion on the circular path.

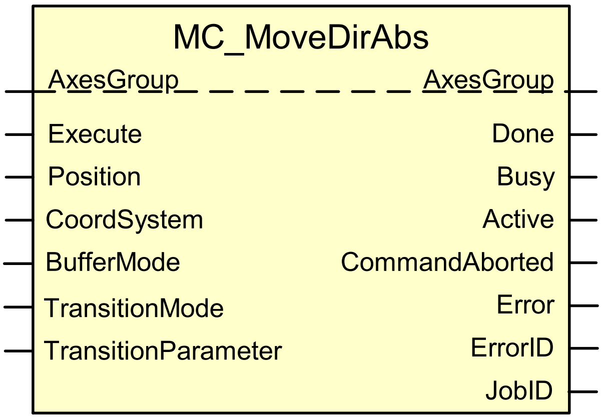
Values for the “PathChoice” input with MC_MoveCircRel

Constant	Value	Description
mcClockwise	0	Passing the circular path in the clockwise direction
mcCounterClockwise	1	Passing the circular path in the counter clockwise direction

3.16 MC_MoveDirAbs

The function block commands a motion by specifying an absolute target position. No path motion is interpolated here but every axis moves at the maximum possible velocity.

Block diagram



FB parameters

VAR_IN_OUT		
Variable name	**Data type**	**Description**
AxesGroup	AXES_GROUP_REF	Axis group reference
VAR_INPUT		
Variable name	**Data type**	**Description**
Execute	BOOL	The interpolated motion of the axes starts on a rising edge corresponding to the specified absolute positions.
Position	ARRAY [0..HLI_CH_AX_MAXIDX] OF LREAL	Array of absolute end positions in the current coordinate system. Default unit [0.1 µm]

CoordSystem	MC_COORDINATE_SYSTEM	Defines the coordinate system of the absolute positions:
		mcACS = 0 (axis coordinate system)
		mcPCS = 2 (programming coordinate system)

BufferMode	MC_BUFFER_MODE	The input defines when a job is activated provided that other jobs are already active when the FB is commanded or are waiting for execution. It also defines the path generated at the transition between 2 jobs. The following values are possible:
		mcAborting = 0
		mcBuffered = 1
		mcBlendingPrev = 3

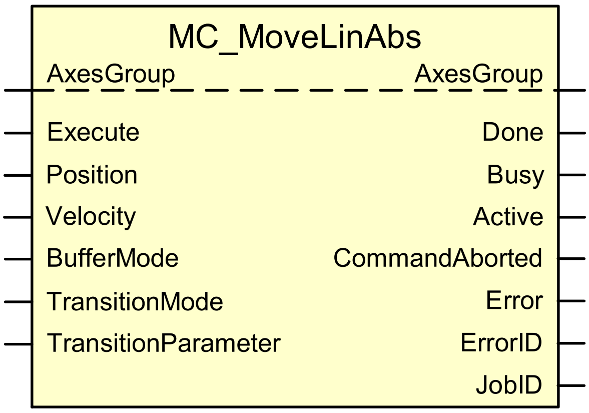
TransitionMode	UDINT	0: TMNone (no transition motion is inserted, default setting) 2: TMConstantVelocity 3: TMCornerDistance 4: TMMaxCornerDeviation 10: TMConstDeviation 11: TMIntermediatePoint 12: TMDynamicOptimized
TransitionParameter	MCV_ARRAY_TRANS_PARAM	See Section Path blending and transition modes [▸ 11]
VAR_OUTPUT		
Variable name	**Data type**	**Description**
Done	BOOL	TRUE indicates that the moved axes reached their end position. The signal is activated as soon as the axes are located within the window around the target position defined by the related axis parameter **getriebe[...].window** (P-AXIS-00236).
Busy	BOOL	TRUE indicates that the function block is executing a job.
Active	BOOL	TRUE indicates that the function block sent the command to the MC.

CommandAborted	BOOL	TRUE indicates that the currently active command sent by the FB was aborted by another command.
Error	BOOL	TRUE indicates that an error occurred.
ErrorID	WORD	Error code
JobID	UDINT	Ordinal number of the last job sent by the FB.

3.17 MC_MoveLinAbs

This function block commands a linear interpolated motion of the axes of an axis group by specifying absolute target positions.

Block diagram



FB parameters

VAR_IN_OUT		
Variable name	**Data type**	**Description**
AxesGroup	AXES_GROUP_REF	Axis group reference
VAR_INPUT		
Variable name	**Data type**	**Description**
Execute	BOOL	The linear interpolated motion of the axes starts on a rising edge corresponding to the specified absolute positions.
Position	ARRAY [0..HLI_CH_AX_MAXIDX] OF LREAL	Array of absolute end positions in the current coordinate system. Default unit [0.1 µm]
Velocity	LREAL	Maximum path velocity of the motion. This value need not be reached. Default unit [10^{-3} °/ s]

BufferMode	MC_BUFFER_MODE	The input defines when a job is activated provided that other jobs are already active when the FB is commanded or are waiting for execution. It also defines the path generated at the transition between 2 jobs. The following values are possible:
		mcAborting = 0
		mcBuffered = 1
		mcBlendingPrev = 3

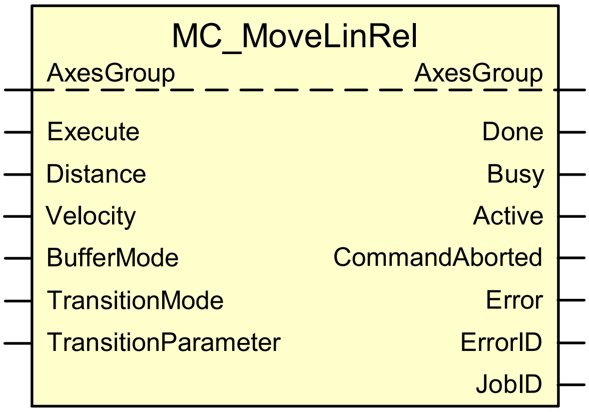
TransitionMode	UDINT	0: TMNone (no transition curve is inserted; default setting) 2: TMConstantVelocity 3: TMCornerDistance 4: TMMaxCornerDeviation 10: TMConstDeviation 11: TMIntermediatePoint 12: TMDynamicOptimized
TransitionParameter	MCV_ARRAY_TRANS_PARAM	See Section Path blending and transition modes [▸ 11]
VAR_OUTPUT		
Variable name	**Data type**	**Description**
Done	BOOL	TRUE indicates that the axes have reached their end position.
Busy	BOOL	TRUE indicates that the function block is executing a job.
Active	BOOL	TRUE indicates that the function block sent the command to the MC.
CommandAborted	BOOL	TRUE indicates that the currently active command sent by the FB was aborted by another command.
Error	BOOL	TRUE indicates that an error occurred.

ErrorID	WORD	Error code
JobID	UDINT	Ordinal number of the last job sent by the FB.

3.18 MC_MoveLinRel

This function block commands a linear interpolated motion of the axes of an axis group by specifying the target position by specifying relative distances.

Block diagram



FB parameters

VAR_IN_OUT		
Variable name	**Data type**	**Description**
AxesGroup	AXES_GROUP_REF	Axis group reference
VAR_INPUT		
Variable name	**Data type**	**Description**
Execute	BOOL	The interpolated motion relative to the current coordinate starts on the rising edge.
Distance	ARRAY [0..HLI_CH_AX_MAXIDX] OF LREAL	Array of relative distances in the current coordinate system. Default unit [0.1 µm]
Velocity	LREAL	Maximum path velocity of the motion. This value need not be reached. Default unit [10^{-3} °/ s]

BufferMode	MC_BUFFER_MODE	The input defines when a job is activated provided that other jobs are already active when the FB is commanded or are waiting for execution. It also defines the path generated at the transition between 2 jobs. The following values are possible:
		mcAborting = 0
		mcBuffered = 1
		mcBlendingPrev = 3

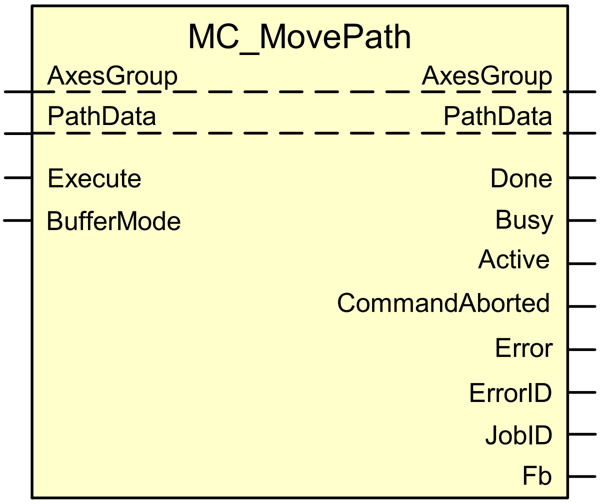
TransitionMode	UDINT	0: TMNone (no transition motion is inserted, default setting) 2: TMConstantVelocity 3: TMCornerDistance 4: TMMaxCornerDeviation 10: TMConstDeviation 11: TMIntermediatePoint 12: TMDynamicOptimized
TransitionParameter	MCV_ARRAY_TRANS_PARAM	See Section Path blending and transition modes [▸ 11]
VAR_OUTPUT		
Variable name	**Data type**	**Description**
Done	BOOL	TRUE indicates that the axes have reached their end position.
Busy	BOOL	TRUE indicates that the function block is executing a job.
Active	BOOL	TRUE indicates that the function block sent the command to the MC.
CommandAborted	BOOL	TRUE indicates that the currently active command sent by the FB was aborted by another job for the axis group.
Error	BOOL	TRUE indicates that an error occurred.

ErrorID	WORD	Error code
JobID	UDINT	Ordinal number of the last job sent by the FB.

3.19 MC_MovePath

When implemented, this function block enables the execution of an NC program. The name of the file containing the NC program is specified in the data structure MC_PATH_DATA_REF [▸ 10].

Block diagram



FB parameters

VAR_IN_OUT		
Variable name	**Data type**	**Description**
AxesGroup	AXES_GROUP_REF [▸ 8]	Axis group reference
PathData	MC_PATH_DATA_REF [▸ 10]	Reference to the path description. This implementation means an NC program.
VAR_INPUT		
Variable name	**Data type**	**Description**
Execute	BOOL	Starts the command on the rising edge.

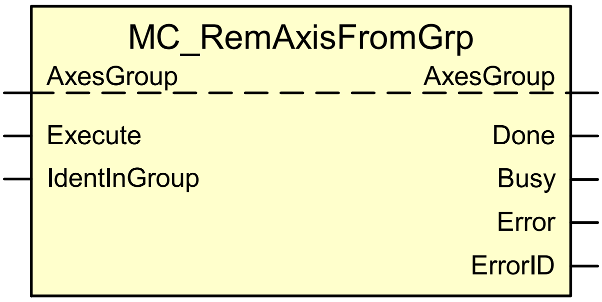
BufferMode	MC_BUFFER_MODE	The input defines when a job is activated provided that other jobs are already active when the FB is commanded or are waiting for execution. It also defines the path generated at the transition between 2 jobs. The following values are possible:
		mcAborting = 0
		mcBuffered = 1
		mcBlendingPrev = 3

VAR_OUTPUT		
Variable name	**Data type**	**Description**
Done	BOOL	NC program execution is completed.
Busy	BOOL	TRUE indicates that the function block is executing a job.
Active	BOOL	The command for the FB is not yet completed.
CommandAborted	BOOL	NC program execution was aborted.
Error	BOOL	TRUE indicates that an error occurred.
ErrorID	WORD	Error code
JobID	UDINT	Ordinal number of the last job sent by the FB.
Fb	MCV_FB_REF	Optional for use in applications with job management (MCV_GrpReadJobAck) [▸ 83] This output is available as of CNC Build V3.1.3103.1.

3.20 MC_RemAxisFromGrp

Removes an axis from the axis group. This is an administrative FB, i.e. no motion job is generated.

Block diagram



FB parameters

VAR_IN_OUT		
Variable name	**Data type**	**Description**
AxesGroup	AXES_GROUP_REF	Axis group reference
VAR_INPUT		
Variable name	**Data type**	**Description**
Execute	BOOL	The axis is removed from the axis group on a rising edge.
IdentInGroup	IDENT_IN_GROUP_REF	Identification of the axis in the axis group.
VAR_OUTPUT		
Variable name	**Data type**	**Description**
Done	BOOL	The axis was successfully removed from the axis group.
Busy	BOOL	TRUE indicates that the function block is executing a job.
Error	BOOL	TRUE indicates that an error occurred.
ErrorID	WORD	Error code

Behaviour of the FB:

- Only the axis name need be specified to remove an axis from an axis group. Therefore, only the “Name” element needs to be assigned accordingly at the “IdentInGroup“ input in the IDENT_IN_GROUP_REF [▸ 10] data structure.

Notice

The following conditions must be met to permit this function block to work correctly when the axis reference at the “Axis” input is an axis:

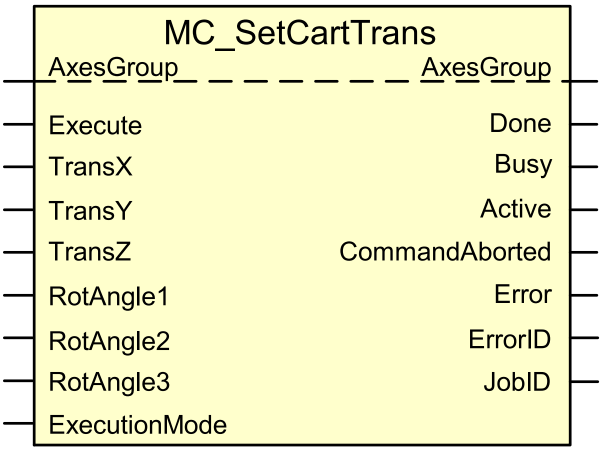
The spindle axis must be notified to the NC channel by specifying it in the channel parameter list (see [FCT-S1//NC spindle]).

The identifier to address the spindle in the channel and defined by the channel parameter P-CHAN-00007 must be formed from the letter “S” followed by the logical axis number of the spindle (P-CHAN-00036). For example, if P-CHAN-00036 = 2, then S2 must be specified for P-CHAN-00007.

3.21 MC_SetCartTrans

This function block activates a Cartesian coordinate transformation. This type of transformation can offset and rotate the PCS relative to the MCS. The offset and rotation specifications are activated relative to the origin of the currently active PCS.

Block diagram



FB parameters

VAR_IN_OUT		
Variable name	**Data type**	**Description**
AxesGroup	AXES_GROUP_REF	Axis group reference
VAR_INPUT		
Variable name	**Data type**	**Description**
Execute	BOOL	A rising edge activates a transformation.
TransX	LREAL	X coordinate of the offset vector (V_x)
TransY	LREAL	Y coordinate of the offset vector (V_y)
TransZ	LREAL	Z coordinate of the offset vector (V_z)
RotAngle1	LREAL	Rotation angle about X axis (j_1).
RotAngle2	LREAL	Rotation angle about Y axis (j_2).
RotAngle3	LREAL	Rotation angle about Z axis (j_3)
ExecutionMode	UDINT	mcQueued = 2
VAR_OUTPUT		
Variable name	**Data type**	**Description**
Done	BOOL	The Cartesian coordinate transformation was activated.
Busy	BOOL	TRUE indicates that the function block is executing a job.
Active	BOOL	TRUE indicates that the function block sent the job to the MC.
CommandAborted	BOOL	TRUE indicates that the currently active command sent by the FB was aborted by another command for the axis group.
Error	BOOL	TRUE indicates that an error occurred.
ErrorID	WORD	Error code
JobID	UDINT	Ordinal number of the last job sent by the FB.

When the rotations j_1, j_2 and j_3 are defined, they must be rotated in the sequence listed below and in the mathematical positive direction of rotation:

1. Rotation at angle j_3 about the Z axis
2. Rotation at angle j_2 about the Y axis (y´)
3. Rotation at angle j_1 about the X axis (e.g. x´´)

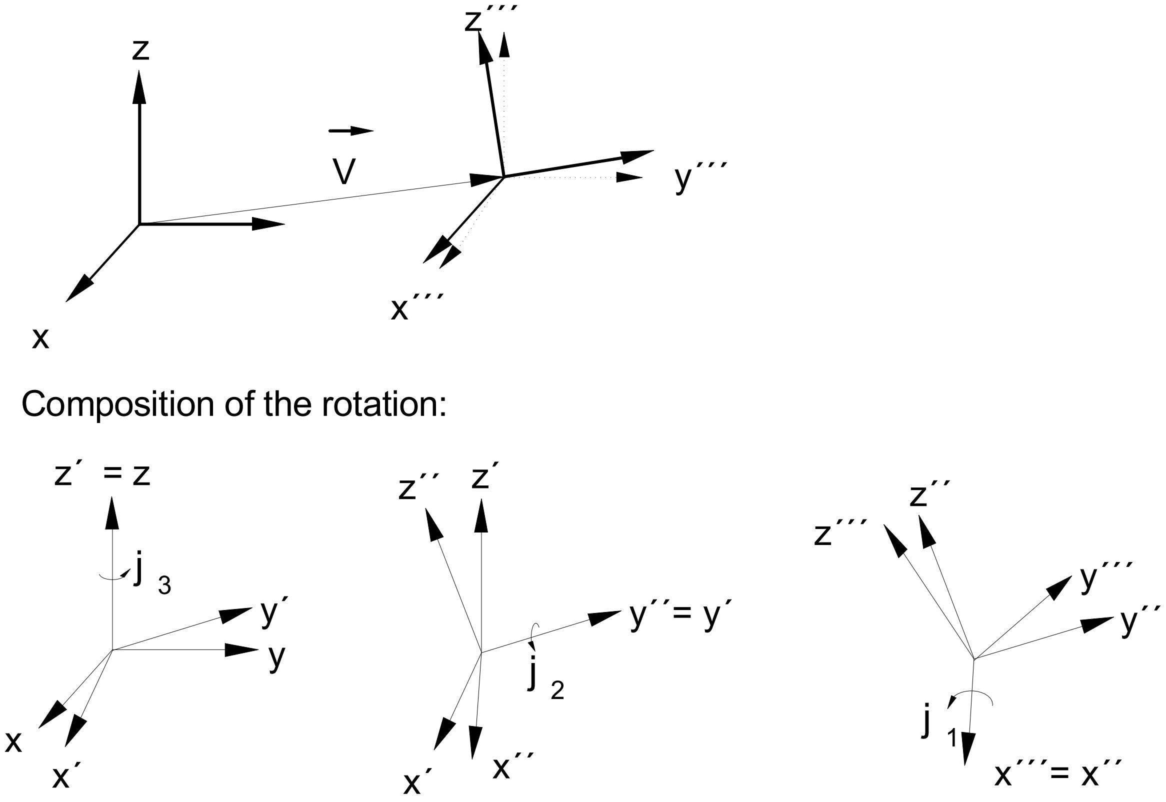
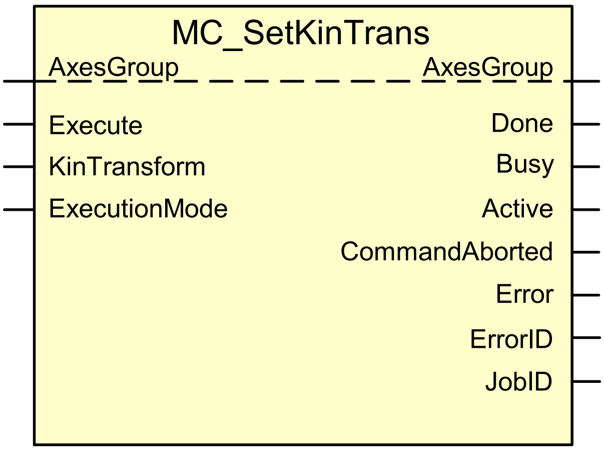


Fig. 2: Composition of the rotation

3.22 MC_SetKinTrans

This function block activates a kinematic coordinate transformation which creates the relationship between the axis coordinate system (ACS) and the machine coordinate system (MCS) by selecting one of the kinematic models from a row.

Block diagram



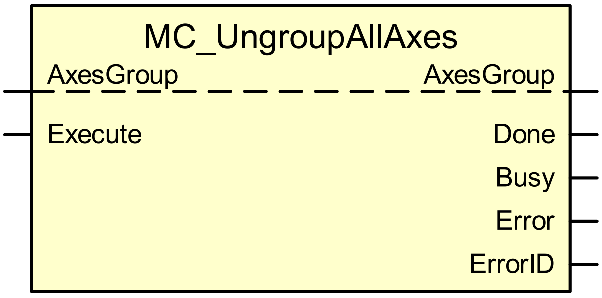
FB parameters

VAR_IN_OUT		
Variable name	**Data type**	**Description**
AxesGroup	AXES_GROUP_REF	Axis group reference
VAR_INPUT		
Variable name	**Data type**	**Description**
Execute	BOOL	Activates a kinematic coordinate transformation on a rising edge taking into consideration the time of execution specified at the “ExecutionMode” pin.
KinTransform	UDINT	Reference for a kinematic model.
ExecutionMode	UDINT	Defines the time when the command of the administrative block is activated. mcQueued = 2
VAR_OUTPUT		
Variable name	**Data type**	**Description**
Done	BOOL	The kinematic coordinate transformation was activated.
Busy	BOOL	TRUE indicates that the function block is executing a job.
Active	BOOL	TRUE indicates that the function block sent the command to the MC.
CommandAborted	BOOL	TRUE indicates that the currently active command sent by the FB was aborted by another command for the axis group.
Error	BOOL	TRUE indicates that an error occurred.
ErrorID	WORD	Error code
JobID	UDINT	Ordinal number of the last job sent by the FB.

3.23 MC_UngroupAllAxes

This function block removes all the axes from an axis group. This is an administrative FB, i.e. no motion job is generated.

Block diagram



FB parameters

VAR_IN_OUT		
Variable name	**Data type**	**Description**
AxesGroup	AXES_GROUP_REF	Axis group reference
VAR_INPUT		
Variable name	**Data type**	**Description**
Execute	BOOL	All the axes are removed from an axis group on a rising edge.
VAR_OUTPUT		
Variable name	**Data type**	**Description**
Done	BOOL	All the axes are removed from an axis group on a rising edge.
Busy	BOOL	TRUE indicates that the function block is executing a job.
Error	BOOL	TRUE indicates that an error occurred.
ErrorID	WORD	Error code

Notice

The following conditions must be met to permit this function block to work correctly when the axis reference at the “Axis” input is an axis:

The spindle axis must be notified to the NC channel by specifying it in the channel parameter list (see [FCT-S1//NC spindle]).

The identifier to address the spindle in the channel and defined by the channel parameter P-CHAN-00007 must be formed from the letter “S” followed by the logical axis number of the spindle (P-CHAN-00036). For example, if P-CHAN-00036 = 2, then S2 must be specified for P-CHAN-00007.

4 PLC library “McpPLCopenP4” – ISG extensions

The function blocks described in this section provide additional functions.

4.1 Overview of the additional ISG FBs

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

FB	PLC systems		
	KW	3S	TwinCAT
MCV_P4_Platform [▸ 61]	X	X	X
MCV_AxesGroup [▸ 62]	X	X	X
MCV_GrpDisplayAxes [▸ 63]	X	X	X
MCV_GrpReadErrorHistory [▸ 66]	X	X	X
MCV_GrpReadErrorInfo [▸ 66]	X	X	X
MCV_GrpReadMotionState [▸ 69]	X	X	X
MCV_GrpReadSetPos [▸ 71]	X	X	X
MCV_SetKinParam [▸ 72]	X	X	X
MCV_GrpPathMode [▸ 78]	-	X	-
MCV_GrpPathPrepare [▸ 80]		X	-
MCV_GrpReadJobAck [▸ 83]	-	X	-
MCV_GrpReadJobStatusList [▸ 85]	-	X	-
MCV_GrpResetForced [▸ 85]	-	X	-

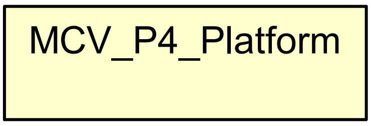
4.2 MCV_P4_Platform

This function block ensures that the motion controller is signalled in the initialisation phase that the PLC is present and commands for axis groups are sent to the motion controller. Afterwards, its task is to invoke cyclically the instances of the function block MCV_AxesGroup [▸ 62] thus ensuring communication between the motion controller and the PLC relating to PLCopen Part 4 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_AxesGroup

This function block is used to monitor the communication path between the motion controller and the axis group and to transfer information available from the motion controller, such as error messages or command acknowledgements, to the PLC.

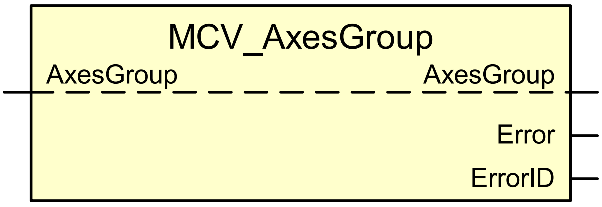
Notice

This function block must be called in a PLC application in every PLC cycle.

There may only be one instance of the function block per axis group.

To simplify matters for the user, the library provides the function block MCV_P4_Platform in which the required instances of the FB MCV_AxesGroup are instanced and called.

Block diagram



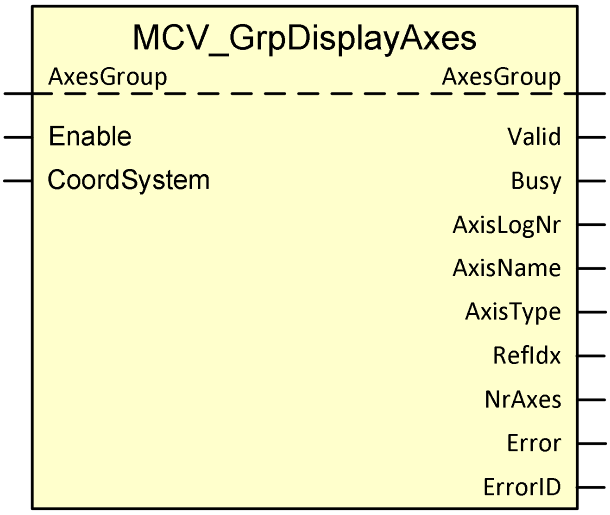
FB parameters

VAR_IN_OUT		
Variable name	**Data type**	**Description**
AxesGroup	AXES_GROUP_REF	Axis group reference
VAR_OUTPUT		
Variable name	**Data type**	**Description**
Error	BOOL	TRUE indicates that an error occurred.
ErrorID	WORD	Error code

4.4 MCV_GrpDisplayAxes

This function block displays which axes currently belong to which axis group. Various items of information is displayed for the axes in the axis group. The information index corresponds to the index of the axis in each context, i.e. axes in the axis configuration or an axis group.

Block diagram



FB parameters

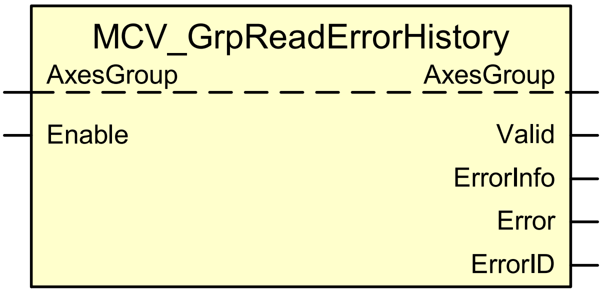
VAR_IN_OUT		
Variable name	**Data type**	**Description**
AxesGroup	AXES_GROUP_REF	Axis group reference
VAR_INPUT		
Variable name	**Data type**	**Description**
Enable	BOOL	As long as the input is TRUE, the function displays the axes included in the axis group.
CoordSystem	MC_COORDINATE_SYSTEM	Defines the display of information about the axes in an axis group: mcACS = 0: Information about the axes configured in the control system belonging to the axis group applied to the AxesGroup input. mcPCS = 2: Information about the axes belonging to an axis group applied to the AxesGroup input as viewed by an NC channel. The input is defaulted to the value mcPCS.
VAR_OUTPUT		
Variable name	**Data type**	**Description**
Valid	BOOL	TRUE indicates that the data at the "AxisLogNr" and "AxisName" outputs are valid and show the composition of the axis group.
Busy	BOOL	TRUE indicates that the FB is executing a job.
AxisLogNr	ARRAY [0..HLI_CH_AX_MAXIDX] OF UINT	Array of logical axis numbers (see P-AXIS-00016) of the axes included in the axis group.
AxisName	ARRAY [0..HLI_CH_AX_MAXIDX] OF STRING(HLI_ACHS_NAME_LAENGE)	Array of axis names of the axes included in the axis group.
AxisType	ARRAY [0..HLI_CH_AX_MAXIDX] OF UINT	Array containing the axis type of an axis,. For value range, see P-AXIS-00018 If **CoordSystem = mcACS**, the value for the axis type configured in the axis list is displayed. If **CoordSystem = mcPCS**, the axis type activated by programming in the NC program is displayed.
RefIdx	ARRAY [0..HLI_CH_AX_MAXIDX] OF DINT	Array representing the sequence of axes in each context and containing the logical axis number of each axis. Valid axis numbers have a value > 0.

		If **CoordSystem = mcACS**, the context is all axes configured for a control application. The array contains the valid axis number in consecutive order (with no gaps). The axis number 0 indicates that the axis was not configured. If **CoordSystem = mcPCS,** , this array displays the logical axis numbers of all axes in the axis group. The sequence of axis numbers is dependent on the classification of the axes in the axis group (e.g. by programming), i.e. valid axis numbers are not necessarily consecutive (with gaps).
NrAxes	UINT	Number of axes currently belonging to the axis group. If **CoordSystem = mcACS**, gantry slave axes are included in the count if the gantry master axis is included in the axis group. If **CoordSystem = mcPCS,** , gantry slave axes are not included in the count.
Error	BOOL	TRUE indicates that an error occurred.
ErrorID	WORD	Error identifier

4.5 MCV_GrpReadErrorHistory

The function block displays the errors and warnings occurring in the motion controller in the sequence they occur at the “ErrorInfo” output. The first error occurring since the last command of a MC_GrpReset [▸ 31] is the error with the lowest array index.

Blockdiagramm



Parameters of the FB

VAR_IN_OUT		
Variable name	**Data type**	**Description**
AxesGroup	AXES_GROUP_REF	Axis group reference

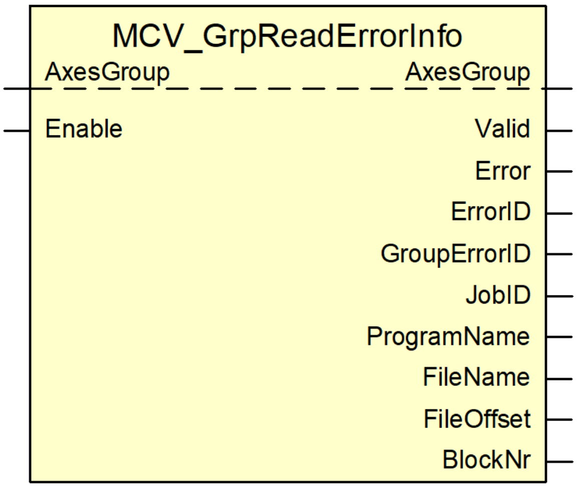
VAR_INPUT		
Variable name	**Data type**	**Description**
enable	BOOL	As long as the input is TRUE, errors and warnings of the Motion Controller are listed in the sequence of their occurrence at the “ErrorInfo” output.

VAR_OUTPUT		
Variable name	**Data type**	**Description**
Valid	BOOL	TRUE indicates that the list at the “ErrorInfo” output is up to date and valid.
ErrorInfo	ARRAY [0..MCV_AGR_ERR_BUF_MAXIDX] OF MC_ERROR_INFO_REF	Error identifiers of the motion controller are output in the sequence of their occurrence. The current error is always the one with the highest array index irrespective of the number of error identifiers displayed, even if the list is completely full.
Error	BOOL	TRUE indicates that an error occurred.
ErrorID	WORD	Error code.

4.6 MCV_GrpReadErrorInfo

The function block displays additional information concerning the actual error in the motion controller to facilitate the diagnosis and elimination of the error.

Block diagram



FB parameters

VAR_IN_OUT		
Variable name	**Data type**	**Description**
AxesGroup	AXES_GROUP_REF	Axis group reference

VAR_INPUT		
Variable name	**Data type**	**Description**
Enable	BOOL	As long as input is TRUE, the function block displays additional information on the actual error of the motion controller.

VAR_OUTPUT		
Variable name	**Data type**	**Description**
Valid	BOOL	TRUE indicates that the information concerning the actual error is valid.
Error	BOOL	TRUE indicates that an error occurred when the information in the current error is read.
ErrorID	WORD	Error identifier
GroupErrorID	UDINT	Error number of the actual axis group error.
JobID	UDINT	Ordinal number of the command when the displayed error occurred.
ProgramName	HLI_PROG_NAME	Name of the NC program that was started by a function block MC_MovePath [▸ 51].
FileName	HLI_FILE_NAME	Name of file that contains the NC program that is displayed at output “ProgramName”.
FileOffset	UDINT	Offset in the NC program file that determines the location where the actual error occurred. Unit: Bytes
BlockNr	DINT	Displays the NC block number of the NC program where the current error occurred if the NC block was addressed by the N function ([PROG//N function]).

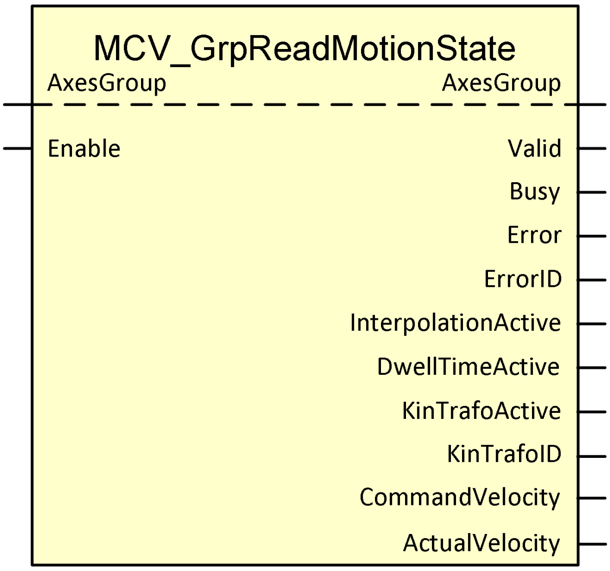
Behaviour of the FB:

- The information of the actual error is displayed at the outputs until a MC_GrpReset is commanded successful.

4.7 MCV_GrpReadMotionState

This function block supplies the current motion states of an axis group.
It is assigned to the group of administrative function blocks.

Block diagram



FB parameters

VAR_IN_OUT		
Variable name	**Data type**	**Description**
AxesGroup	AXES_GROUP_REF	Axis group reference
VAR_INPUT		
Variable name	**Data type**	**Description**
Enable	BOOL	If the value is TRUE, the output data of the axes group is displayed continuously.
VAR_OUTPUT		
Variable name	**Data type**	**Description**
Valid	BOOL	Is TRUE if valid values are available.
Busy	BOOL	TRUE indicates that the FB is executing a job.
Error	BOOL	Indicates whether an error occurred in the FB.
ErrorID	WORD	Error identifier
InterpolationActive	BOOL	Is TRUE if the interpolator is active.
DwellTimeActive	BOOL	Is TRUE if the function waits due to a dwell time.
KinTrafoActive	BOOL	Is TRUE if a kinematic transformation is active.
KinTrafoID	UINT	If the output KinTrafoActive is TRUE, then this is the identifier of the currently active kinematic transformation. The identifier corresponds to the numbers in the description [KITRA/Classification/kinematic types]
CommandVelocity	LREAL	Programmed velocity
ActualVelocity	LREAL	Current velocity

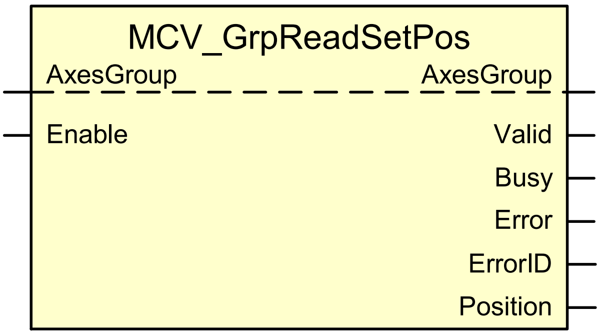
Behaviour of the FB:

- The axis group data is updated cyclically as long as the input “Enable” has the value TRUE.
- If an axis group is operated with kinematic transformation (typical with Cartesian programming of the TCP of a robot kinematic), the outputs "CommandVelocity" and "ActualVelocity" must be linked to the output "KinTrafoActive", because, if kinematic transformation is deselected, the velocity outputs do not show the real behaviour of the TCP. This linking of outputs is not required with Cartesian machines.

4.8 MCV_GrpReadSetPos

The function block supplies the command positions of the tool centre point in the specified Cartesian machine coordinate system (MCS).

Block diagram



FB parameters

VAR_IN_OUT		
Variable name	**Data type**	**Description**
AxesGroup	AXES_GROUP_REF	Axis group reference
VAR_INPUT		
Variable name	**Data type**	**Description**
Enable	BOOL	As long as the input is TRUE, the function block indicates the command positions of the tool centre point in the MCS at the “Position” output.
VAR_OUTPUT		
Variable name	**Data type**	**Description**
Valid	BOOL	Is TRUE as long as command positions are supplied.
Busy	BOOL	TRUE indicates that the FB is active.
Error	BOOL	TRUE indicates that an error occurred when the information in the current error is read.
ErrorID	WORD	Error code
Position	ARRAY [0..HLI_CH_AX_MAXIDX] OF LREAL	Array containing the command positions of the tool centre point based on the MCS.

4.9 MCV_SetKinParam

This function block is used to change the parameter values of a kinematic transformation, e.g. a robot kinematic. The kinematic transformation is identified by the “KinematicID” identification number assigned to it. The individual parameters of a kinematic transformation are identified by an index. The meaning of the individual parameters depends on the kinematic transformation whose parameters are to be changed.

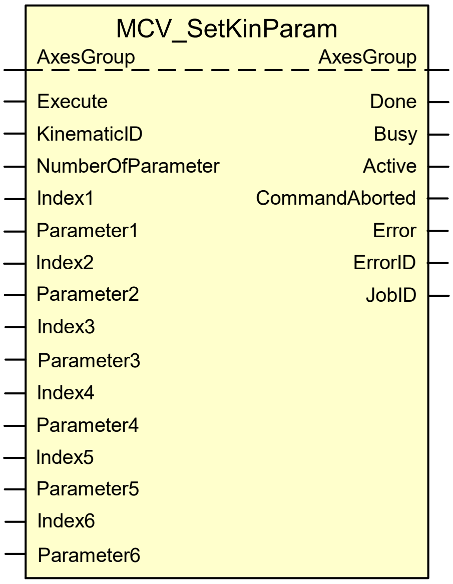
The function block has no “BufferMode” input but behaves like a function block with the “Buffer-Mode” mcBuffered.

After the controller is reset, all changes executed by this type of function block are lost.

Attention

Since this FB can change kinematic parameters during operation, it should be noted that incorrect values for the written parameters may lead to unexpected axis motions.

Block diagram



FB parameters

VAR_IN_OUT		
Variable name	**Data type**	**Description**
AxesGroup	AXES_GROUP_REF	Axis group reference
VAR_INPUT		
Variable name	**Data type**	**Description**
Execute	BOOL	The parameters of the kinematic transformation are sent to the motion controller on a rising edge.
KinematicID	UDINT	Identification number of the kinematic transformation ([KITRA])
NumberOfParameter	UDINT	Number N of value pairs (IndexX, ParameterX) which are to be sent, where N = [0.6] and X=1..N. For N=0 no parameters are sent to the motion controller. If a value > 6 is applied, it is converted in the FB to 6.
Index1	UDINT	Index of the parameter whose value is specified at the “Parameter1” input. The index value depends on the kinematic transformation selected at the “KinematicID” input and which parameter is to be sent.
Parameter1	DINT	Parameter value to be sent. Length specifications are in 0.1µm, angles in 10_{-4} °.
Index2	UDINT	Optional: Index for 2nd parameter
Parameter2	DINT	Optional: Value of 2nd parameter
Index3	UDINT	Optional: Index for 3rd parameter
Parameter3	DINT	Optional: Value of 3rd parameter
Index4	UDINT	Optional: Index for 4th parameter
Parameter4	DINT	Optional: Value of 4th parameter
Index5	UDINT	Optional: Index for 5th parameter
Parameter5	DINT	Optional: Value of 5th Parameter
Index6	UDINT	Optional: Index for 6th parameter
Parameter6	DINT	Optional: Value of 6th parameter
VAR_OUTPUT		
Variable name	**Data type**	**Description**
Done	BOOL	TRUE indicates that the parameters were adopted by the motion controller.
Busy	BOOL	TRUE indicates that the function block is executing a job.
Active	BOOL	TRUE indicates that the function block sent the command to the MC.
CommandAborted	BOOL	TRUE indicates that the currently active command sent by the FB was aborted by another command.

Error	BOOL	TRUE indicates that an error occurred.
ErrorID	WORD	Error code
JobID	UDINT	Ordinal number of the last job sent by the FB.

4.9.1 Example 1

The graphic below shows that param[7] of the kinematic transformation with ID=45 (articulated robot with 6 machine axes) is the length of the robot arm between joints 2 and 3 and how this information can be transferred to the motion controller via MCV_SetKinParam [▸ 72].

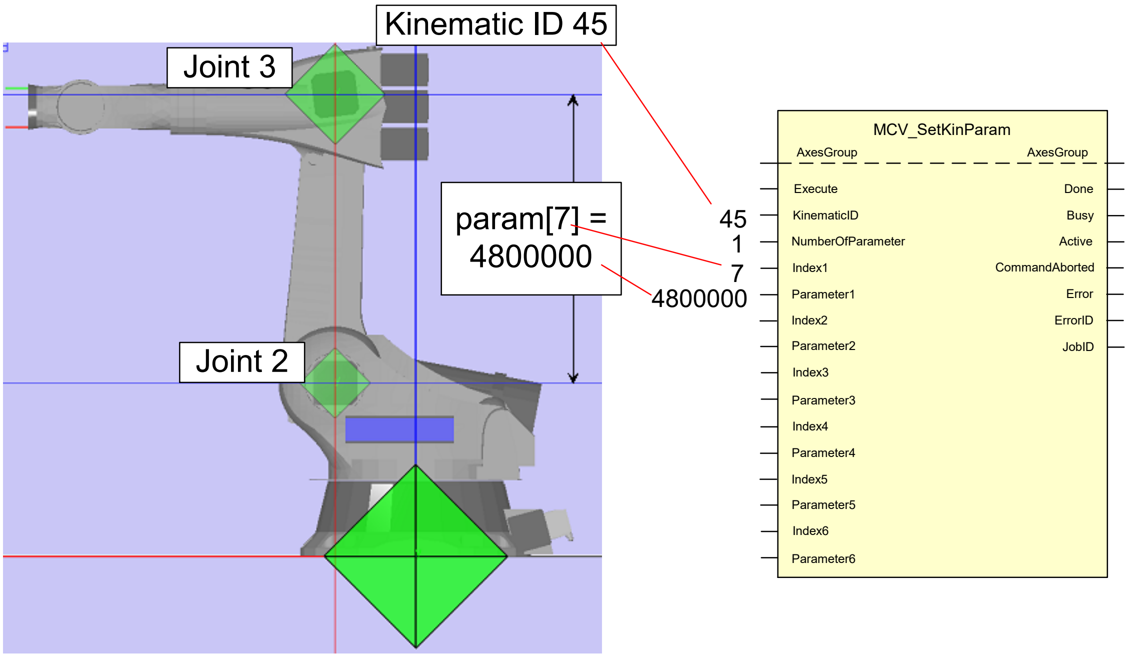


Fig. 3: Articulated robot kinematic transformation ID=45

4.9.2 Example 2

The second example shows how tool parameters are changed if they are to be considered in the parameters for the kinematic transformation ID=45, as is the case here.

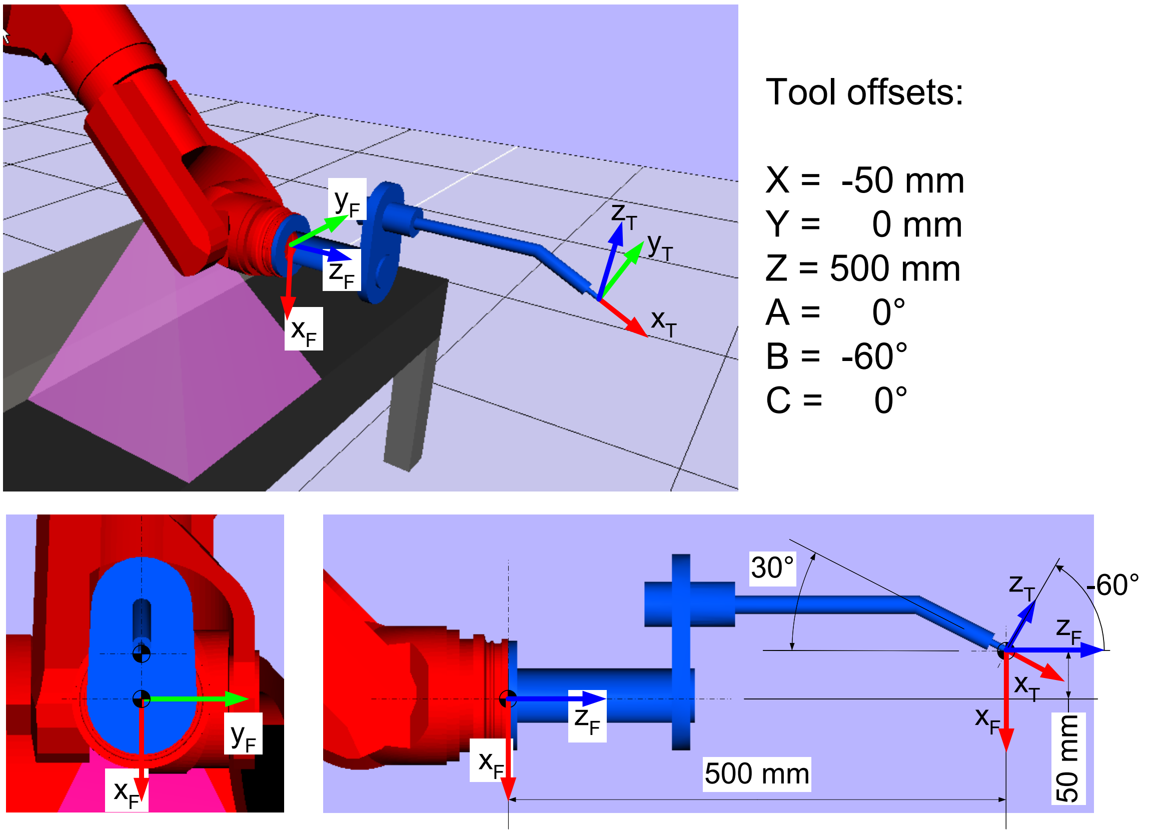


Fig. 4: Tool offsets for kinematic transformation IS=45

If tool parameters are preassigned in the axis group parameter list, the identifiers for the tool offsets (P-CHAN-00094) are:

kinematik[45].param[0]	5000000	Tool centre point offset in Z direction of a coordinate system in the tool flange (Z_F)
kinematik[45].param[1]	-500000	Tool centre point offset in X direction of a coordinate system in the tool flange (X_F)
kinematik[45].param[2]	0	Tool centre point offset in Y direction of a coordinate system in the tool flange (Y_F)
kinematik[45].param[3]	0	Tool orientation offset on rotation about the X'' axis (3rd rotation)
kinematik[45].param[4]	-600000	Tool orientation offset on rotation about the Y’ axis (2nd rotation)
kinematik[45].param[5]	0	Tool orientation offset on rotation about the Z axis (1st rotation)

If the same parameters are to be assigned by an instance of an MCV_SetKinParam FB, the inputs of this instance should be assigned as follows:

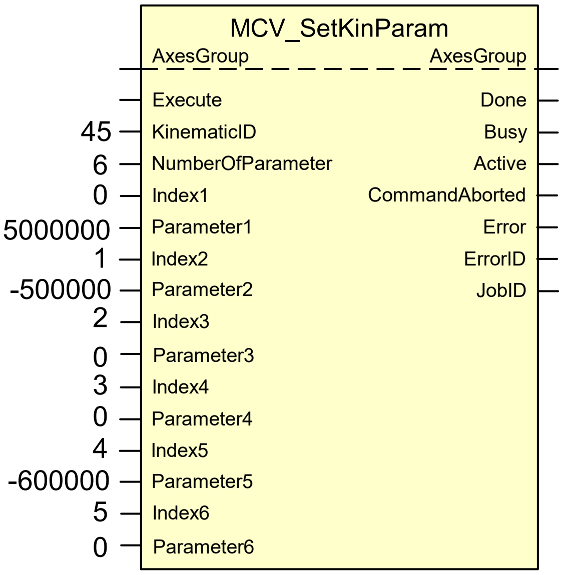


Fig. 5: Example assignment of MCV_SetKinParam with KIN_ID 45

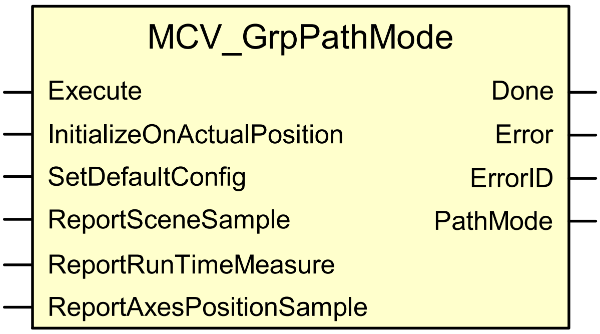
4.10 MCV_GrpPathMode

Release Note

This function block is available as of CNC Build V3.1.3103.1.

The function block is used for entering options to execute a job to be started in the data structure MCV_PATH_MODE [▸ 88] of the "PathMode" output. The output is linked to the “PathMode” input of an instance of the type MCV_GrpPathPrepare [▸ 80].

Block diagram



FB parameters

VAR_INPUT		
Variable name	**Data type**	**Description**
Execute	BOOL	On the rising edge at the input, the values of the input pins referring to start options are entered in a data structure of the type PATH_MODE and output at the PathMode output.
InitializeOnActualPosition	MCV_START_MODE [▸ 88]	
SetDefaultConfig	MCV_START_MODE [▸ 88]	
ReportSceneSample	MCV_START_MODE [▸ 88]	
ReportRunTimeMeasure	MCV_START_MODE [▸ 88]	
ReportAxesPositionSample	MCV_START_MODE [▸ 88]	
VAR_OUTPUT		
Variable name	**Data type**	**Description**
Done	BOOL	TRUE indicates that the values of the start options at the PathMode output are available in a structure of the type MCV_PATH_MODE.
Error	BOOL	TRUE indicates that an error occurred.
ErrorID	WORD	Error code
PathMode	MCV_START_MODE [▸ 88]	Program mode parameters of a job

4.11 MCV_GrpPathPrepare

Release Note

This function block is available as of CNC Build V3.1.3103.1.

The function block is used to transfer job data, program parameters and program modes to a data structure of the MC_PATH_DATA_REF [▸ 10] type. This structure is output at the PathData output and must be transferred to the corresponding input of an MC_MovePath [▸ 51] instance.

The function block inputs can be written by the user. Alternatively, the function block can be linked to the outputs of an MCV_GrpGetJobRequest instance to forward jobs generated by an HMI or another NC channel, for example.

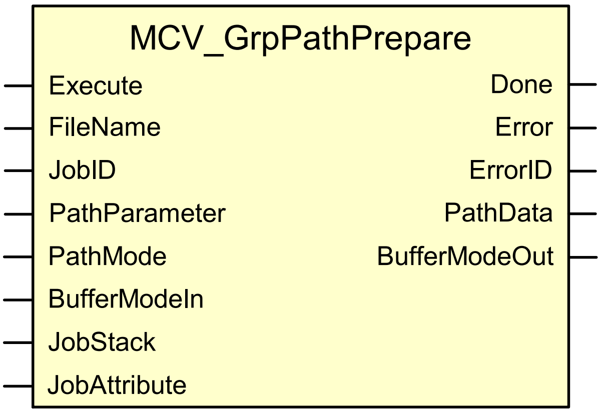
Notice

The structure MC_PATH_DATA_REF may only be described by instances of the type MCV_GrpPathPrepare. This ensures that future changes to the library have no impacts on existing PLC programs.

Notice

What remains permitted is the direct entry of a name or path for an NC program on MC_PATH_DATA_REF. Existing PLC applications can then be used unchanged without job management or parameterisation. In this case, it is not permitted to use MCV_GrpPathPrepare.

Block diagram



FB parameters

VAR_INPUT		
Variable name	**Data type**	**Description**
Execute	BOOL	The function block is executed on a rising edge at the input.
FileName	STRING(MCV_PROG_NAME_STRLEN)	Program name of a job.
JobID	MCV_GRP_JOB_ID [▸ 88]	Job information
PathParameter	MCV_PATH_PARAM [▸ 88]	Program parameters of a job. Parameters can be accessed in the called NC program (FileName). PL[0] corresponds to the variable **@PL1** and PL[19] corresponds to the variable **@PL20**. Unused parameters are prefixed by “0”.
PathMode	MCV_PATH_MODE [▸ 88]	Program mode parameters of a job. The following are available: - InitializeOnActualPosition, - SetDefaultConfig, - ReportSceneSample, - ReportRunTimeMeasure, - ReportAxesPositionSample where 0 = HLI_MC_START_MODE_NOT_USED 1 = HLI_MC_START_MODE_OFF 2 = HLI_MC_START_MODE_ON 3 = HLI_MC_START_MODE_USE_ACTUAL

BufferModeIn	MC_BUFFERMODE	BufferMode of a job.
		The following values (similar to the BufferMode interface in the MC_MovePath [▸ 51] function block) are possible:
		mcAborting= 0
		mcBuffered= 1
		mcBlendingPrev= 3

JobStack	HLI_MC_JOB_STACK [▸ 89]	Information on the job source Only needed if the job was read in by an instance of the MCV_GrpGetJobRequest. Otherwise, the input must remain unassigned.
JobAttribute	UDINT	Identification number which is freely assignable by the user.
VAR_OUTPUT		
Variable name	**Data type**	**Description**
Done	BOOL	TRUE indicates that the data structure MC_PATH_DATA_REF [▸ 10] was correctly filled out and can be transferred together with the value of the "BufferModeOut” output to an instance of the MC_MovePath [▸ 51] function block.
Error	BOOL	TRUE indicates that an error occurred.
ErrorID	WORD	Error code

PathData	MC_PATH_DATA_REF [▸ 10]	Structure for use at the “PathData” input of an instgance of the MC_MovePath [▸ 51].
BufferModeOut	MC_BUFFERMODE	BufferMode for use at the “BufferMode” input of an instance of the MC_MovePath function block.

4.12 MCV_GrpReadJobAck

Release Note

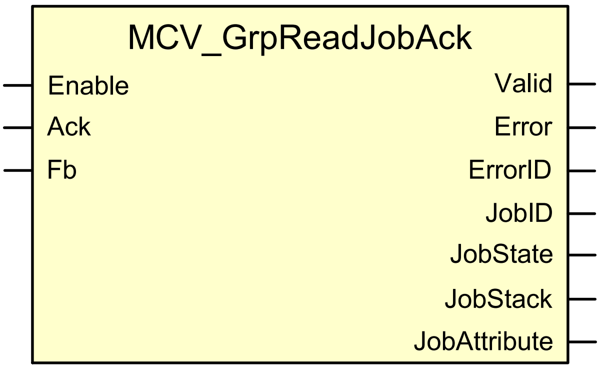
This function block is available as of CNC Build V3.1.3103.1.

The function block is used to evaluate the end or abort operations of jobs commanded by an MC_MovePath [▸ 51] . The instance of the MC_MovePath [▸ 51] whose job acknowledgements are evaluated is identified via the “Fb” input.

Notice

When the function block is created and the “Enable” input is set to TRUE, the function block must be called cyclically. Otherwise, job acknowledgements to any function blocks are blocked, even to other PLCopen Part 4 function blocks.

Block diagram



FB parameters

VAR_INPUT		
Variable name	**Data type**	**Description**
Enable	BOOL	As long as “Enable” is TRUE, job acknowledgements are read. Only acknowledgements are displayed if they refer to jobs sent by the instance of an MC_MovePath [▸ 51] registered at the “Fb” input. New acknowledgements are then only displayed if the “Ack” input is FALSE.
Ack	BOOL	The input defines that the information indicated at the function block outputs was read.
Fb	MCV_FB_REF	Reference to an instance of an MC_MovePath [▸ 51] function block with the output of the same type.
VAR_OUTPUT		
Variable name	**Data type**	**Description**
Valid	BOOL	TRUE indicates that new data is ready to be fetched.
Error	BOOL	TRUE indicates that an error occurred.
ErrorID	WORD	Error code
JobID	MCV_GRP_JOB_INFO [▸ 91]	Job description and job state.
JobState :	MCV_JOB_STATE [▸ 89]	Job information
JobStack	HLI_MC_JOB_STACK [▸ 89]	Information on the job source Only needed if the job is to be read by an instance of the MCV_GrpSetJobResponse.
JobAttribute	UDINT	Identification number specified by the user and output together with the state of a job.

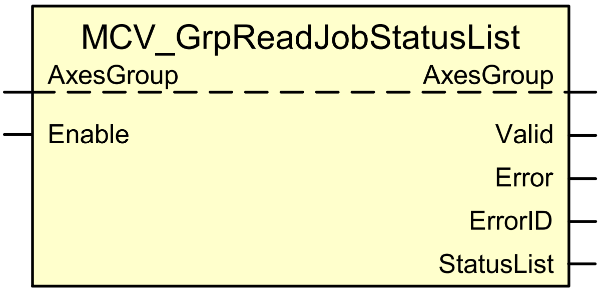
4.13 MCV_GrpReadJobStatusList

Release Note

This function block is available as of CNC Build V3.1.3103.1.

The function block is used to display the states of all jobs currently commanded for an axis group.

Block diagram



FB parameters

VAR_IN_OUT		
Variable name	**Data type**	**Description**
AxesGroup	AXES_GROUP_REF [▸ 8]	Axis group reference
VAR_INPUT		
Variable name	**Data type**	**Description**
Enable	BOOL	If the input is TRUE, the function block outputs are updated.
VAR_OUTPUT		
Variable name	**Data type**	**Description**
Valid	BOOL	TRUE indicates that the requested update was executed.
Error	BOOL	TRUE indicates that an error occurred.
ErrorID	WORD	Error code
StatusList	MCV_GRP_JOB_DISPLAY [▸ 90]	Structure that describes the states of all current MC_MovePath [▸ 51] commands in a channel.

4.14 MCV_GrpResetForced

Release Note

This function block is available as of CNC Build V3.1.3103.1.

The function block leads to a controlled motion stop. It aborts every ongoing command by other motion FBs.

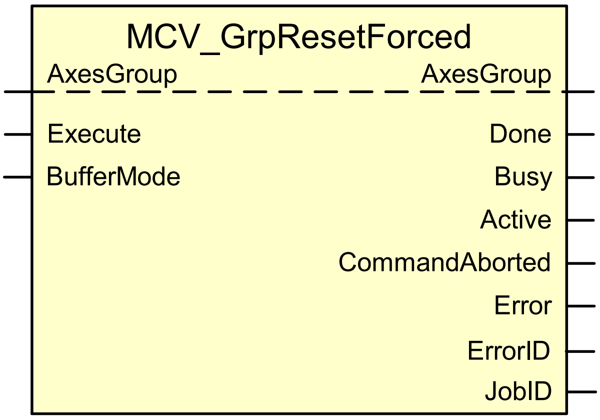
The axis group changes to the “GroupStopping” state until velocity 0 is reached. If the “Done” and “Execute” outputs are set to FALSE, the axis group state changes to “GroupStandby”.

The function block is not interruptible.

Notice

In addition to the motion stop which is implemented identically to the MC_GrpStop [▸ 35], this FB reverts the axis group to its initial state.

Block diagram



FB parameters

VAR_IN_OUT		
Variable name	**Data type**	**Description**
AxesGroup	AXES_GROUP_REF [▸ 8]	Axis group reference
VAR_INPUT		
Variable name	**Data type**	**Description**
Execute	BOOL	Starts the command on the rising edge.

BufferMode	MC_BUFFER_MODE	The input defines when a job is activated provided that other jobs are already active when the FB is commanded or are waiting for execution. The following modes are supported:
		mcAborting = 0
		mcBuffered = 1

VAR_OUTPUT		
Variable name	**Data type**	**Description**
Done	BOOL	TRUE indicates that 0 velocity was reached. The axes are at standstill.
Busy	BOOL	TRUE indicates that the function block is executing a job.
Active	BOOL	The command is not yet completed.
CommandAborted	BOOL	The command to stop was aborted by another command.
Error	BOOL	TRUE indicates that an error occurred.
ErrorID	WORD	Error code
JobID	UDINT	Ordinal number of the last job sent by the FB.

4.15 Data structures for job management

MCV_GRP_JOB_ID

```
TYPE MCV_GRP_JOB_ID :
STRUCT
  JobID : UDINT;
  IfcID : UINT;
END_STRUCT
END_TYPE
```

MCV_PATH_PARAM

```
TYPE MCV_PATH_PARAM :
STRUCT
  PL : ARRAY[0..MCV_PATH_PARAM_MAXIDX] OF LREAL;
END_STRUCT
END_TYPE

MCV_PATH_PARAM_MAXIDX : UDINT := 19;
```

MCV_PATH_MODE

```
TYPE MCV_PATH_MODE :
STRUCT
  InitialiseOnActualPosition : MCV_START_MODE;
  SetDefaultConfig : MCV_START_MODE;
  ReportSceneSample : MCV_START_MODE;
  ReportRunTimeMeasure : MCV_START_MODE;
  ReportAxesPositionSample : MCV_START_MODE;
END_STRUCT
END_TYPE

TYPE MCV_START_MODE : UDINT;
```

HLI_MC_JOB_STACK

```
TYPE HLI_MC_IFC_INDEX : UINT;
TYPE HLI_MC_IFC_CLASS : UINT;


TYPE HLI_MC_IFC_ID :
STRUCT
  i_index : HLI_MC_IFC_INDEX;
  i_class : HLI_MC_IFC_CLASS;
END_STRUCT
END_TYPE


TYPE HLI_MC_JOB_DESCR :
STRUCT
  job_id : UDINT;
  ifc_id : HLI_MC_IFC_ID;
END_STRUCT
END_TYPE


TYPE ARRAY_HLI_MC_JOB_DESC : ARRAY[0..HLI_MC_JOB_STACK_MAXIDX] OF
HLI_MC_JOB_DESCR;
END_TYPE

TYPE HLI_MC_JOB_STACK :
STRUCT
  count : UINT;
  iterator : UINT;
  fill_up_1 : UDINT;
  job : ARRAY_HLI_MC_JOB_DESC;
END_STRUCT
END_TYPE


HLI_MC_JOB_STACK_MAXIDX : INT := 11;
```

MCV_GRP_JOB_STATE

```
TYPE MCV_GRP_JOB_STATE :
STRUCT
  TimeStamp : UDINT;
  ErrorID : WORD;
  State : MCV_JOB_STATE;
  (* See global constants in McpBase.lib MCV_JS_...*)
END_STRUCT
END_TYPE
```

MCV_GRP_JOB_STATUS

```
TYPE MCV_GRP_JOB_STATUS :
STRUCT
  JobDesc : MCV_GRP_JOB_DESCRIPTION;
  JobID : MCV_GRP_JOB_ID;
  JobState : MCV_GRP_JOB_STATE;
END_STRUCT
END_TYPE
```

Paths of this structure:

```
NN.JobDesc.Comment
NN.JobID.JobNumber
NN.JobID.SourceNumber
NN.JobState.TimeStamp
NN.JobState.ErrorID
NN.JobState.State
```

MCV_GRP_JOB_DISPLAY

```
TYPE MCV_GRP_IFC_INDEX : UINT;
TYPE MCV_GRP _IFC_CLASS : UINT;


TYPE MCV_GRP_IFC_ID :
STRUCT
  i_index : MCV_GRP_IFC_INDEX;
  i_class : MCV_GRP_IFC_CLASS;
END_STRUCT
END_TYPE


TYPE MCV_GRP_JOB_DESCR :
STRUCT
  ifc_id : MCV_GRP_IFC_ID;
  job_id : UDINT;
END_STRUCT
END_TYPE


TYPE MCV_JOB_DISPLAY :
STRUCT
  job : MCV_GRP_JOB_DESCR;
  state : DINT;
END_STRUCT
END_TYPE


TYPE MCV_ARRAY_JOB_DISPLAY : ARRAY[0..MCV_JOB_STATE_LIST_MAXIDX] OF
MCV_JOB_DISPLAY;
END_TYPE


MCV_JOB_STATE_LIST_MAXIDX : UINT := 19;

TYPE MCV_GRP_JOB_DISPLAY :
STRUCT
  state : DINT;
  number_of_jobs : UDINT;
  job : MCV_ARRAY_JOB_DISPLAY;
```

```
END_STRUCT
END_TYPE
```

Paths of this structure:

```
NN.JobCount
NN.JobList[0].JobID.JobNumber
NN.JobList[0].JobID.SourceNumber
```

MCV_JOB_STOP_MODE

```
TYPE MCV_JOB_STOP_MODE:
STRUCT
  JobID : MCV_GRP_JOB_ID;
  JobSysInfo : MCV_GRP_JOB_SYS_INFO;
  StopMode : MCV_STOP_MODE;
END_STRUCT
END_TYPE
```

```
HLI_MC_START_MODE_NOT_USED : INT := 0;
HLI_MC_START_MODE_OFF : INT := 1;
HLI_MC_START_MODE_ON : INT := 2;
HLI_MC_START_MODE_USE_ACTUAL : INT := 3;
```

MCV_GRP_JOB_INFO

```
TYPE MCV_GRP_JOB_INFO :
STRUCT
  JobID : MCV_GRP_JOB_ID;
  JobState : MCV_JOB_STATE;
  JobSysInfo : MCV_GRP_JOB_SYS_INFO;
END_STRUCT
END_TYPE
```

5 References

[1] PLCopen specifications: TC2 Task Force Motion Control “Function Blocks for motion control”: Part4 – Coordinated Motion” Version 1.0 dated 3 Dec. 2008

[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).

6 Appendix

Index