DOCUMENTATION ISG-kernel

# PLC library
# McpPLCopen Part 4 Extension

Short Description:
MCP-P4 Ext

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

Documentation version
13/12/2023

# Preface

## Legal information

This documentation was produced with utmost care. The products and scope of functions described are under continuous development. We reserve the right to revise and amend the documentation at any time and without prior notice.

No claims may be made for products which have already been delivered if such claims are based on the specifications, figures and descriptions contained in this documentation.

## Personnel qualifications

This description is solely intended for skilled technicians who were trained in control, automation and drive systems and who are familiar with the applicable standards, the relevant documentation and the machining application.

It is absolutely vital to refer to this documentation, the instructions below and the explanations to carry out installation and commissioning work. Skilled technicians are under the obligation to use the documentation duly published for every installation and commissioning operation.

Skilled technicians must ensure that the application or use of the products described fulfil all safety requirements including all applicable laws, regulations, provisions and standards.

## Further information

Links below (DE)

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

or (EN)

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

contains further information on messages generated in the NC kernel, online help, PLC libraries, tools, etc. in addition to the current documentation.

## Disclaimer

It is forbidden to make any changes to the software configuration which are not contained in the options described in this documentation.

## Trade marks and patents

The name ISG®, ISG kernel®, ISG virtuos®, ISG dirigent® and the associated logos are registered and licensed trade marks of ISG Industrielle Steuerungstechnik GmbH.

The use of other trade marks or logos contained in this documentation by third parties may result in a violation of the rights of the respective trade mark owners.

## Copyright

© ISG Industrielle Steuerungstechnik GmbH, Stuttgart, Germany.

No parts of this document may be reproduced, transmitted or exploited in any form without prior consent. Non-compliance may result in liability for damages. All rights reserved with regard to the registration of patents, utility models or industrial designs.

# 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.

# Contents

**Preface** ..... **2**

**General and safety instructions** ..... **3**

**1 Definitions** ..... **6**

1.1 Abbreviations ..... 6

1.2 Explanations of terms ..... 6

**2 PLC library “McpPLCopenP4Ext”** ..... **8**

2.1 Axis group reference AXES_GROUP_REF ..... 8

2.1.1 Array for data structures of the AXES_GROUP_REF type ..... 8

**3 Overview of the FBs** ..... **9**

**4 Function blocks for block search** ..... **10**

4.1 General information about block search ..... 10

4.2 Use of function blocks for block search ..... 10

4.3 MCV_BlockSearchCtrl ..... 11

4.4 MCV_BlockSearchDeselect ..... 14

4.5 MCV_BlockSearchSetBlockCount ..... 15

4.6 MCV_BlockSearchSetBlockNumber ..... 18

4.7 MCV_BlockSearchSetFileOffset ..... 21

**5 Function blocks for manual mode** ..... **25**

5.1 MCV_ManMode ..... 25

5.2 MCV_Handwheels ..... 29

5.3 MCV_ManModeKeys ..... 30

**6 General function blocks** ..... **32**

6.1 MCV_FadeValueLreal ..... 32

6.2 MCV_GrpAllHomedStatus ..... 33

6.3 MCV_GrpBackward ..... 34

6.4 MCV_GrpFeedhold ..... 35

6.5 MCV_GrpGetNextPrevAx ..... 36

6.6 MCV_GrpKinTrafoCmds ..... 38

6.7 MCV_GrpOperationMode ..... 39

6.8 MCV_GrpOptionalStop ..... 42

6.9 MCV_GrpPrgBlockIgnore ..... 43

6.10 MCV_GrpSetRapidOverride ..... 44

6.11 MCV_GrpSingleAxSelect ..... 45

6.12 MCV_GrpSingleBlock ..... 47

6.13 MCV_ReadActualPathInfo ..... 47

**7 Function blocks for technology functions** ..... **51**

7.1 MCV_GrpHFct ..... 51

7.2 MCV_GrpMFct ..... 53

7.3 MCV_GrpTFct ..... 55

**8 Appendix ..... 57**
8.1 Suggestions, corrections and the latest documentation..... 57
**Keyword index ..... 58**

# 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 “McpPLCopenP4Ext”

This library contains function blocks which permit access to functions of the ISG-MCE. It contains function blocks which act on the ISG-MCE platform, on axis groups or also on axes. The FBs are informed on which definite instance is accessed at an input as reference to the particular instance. The MCP provides these references as global array after start-up. The definitions for these arrays are saved in the base PLC library for the MCP.

## 2.1 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 **HIilfcIdx** 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.

### 2.1.1 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.

# 3 Overview of the FBs

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

| Function block | PLC systems | | |
|---|---|---|---|
| | KW | 3S | TwinCAT |
| MCV_BlockSearchCtrl [▶ 11] | X | X | - |
| MCV_BlockSearchDeselect [▶ 14] | X | X | - |
| MCV_BlockSearchSetBlockCount [▶ 15] | X | X | - |
| MCV_BlockSearchSetBlockNumber [▶ 18] | X | X | - |
| MCV_BlockSearchSetFileOffset [▶ 21] | X | X | - |
| | | | |
| MCV_ManMode [▶ 25] | X | X | - |
| MCV_Handwheels [▶ 29] | - | X | - |
| MCV_ManModeKeys [▶ 30] | - | X | - |
| | - | | |
| MCV_FadeValueLreal [▶ 32] | - | X | - |
| MCV_GrpAllHomedStatus [▶ 33] | X | X | - |
| MCV_GrpBackward [▶ 34] | X | X | - |
| MCV_GrpFeedhold [▶ 35] | X | X | - |
| MCV_GrpGetNextPrevAx [▶ 36] | X | X | - |
| MCV_GrpKinTrafoCmds [▶ 38] | X | X | - |
| MCV_GrpOperationMode [▶ 39] | - | X | - |
| MCV_GrpOptionalStop [▶ 42] | X | X | - |
| MCV_GrpPrgBlockIgnore [▶ 43] | X | X | - |
| MCV_GrpSetRapidOverride [▶ 44] | - | X | - |
| MCV_GrpSingleAxSelect [▶ 45] | X | X | - |
| MCV_GrpSingleBlock [▶ 47] | X | X | - |
| MCV_ReadActualPathInfo [▶ 47] | X | X | - |
| | | | |
| MCV_GrpHFct [▶ 51] | X | X | |
| MCV_GrpMFct [▶ 53] | X | X | - |
| MCV_GrpTFct [▶ 55] | X | x | - |

# 4 Function blocks for block search

## 4.1 General information about block search

Block search is only used

1. to process part of an NC program with real axis motions or
2. to resume execution at the breakpoint after a disruption, for example. Real axis motions then resume at the continuation position.

## 4.2 Use of function blocks for block search

The 4 subtasks for implementing block search are listed below including the associated function blocks used:

### Subtask 1: Acquire data at a breakpoint

The current data relating to NC program execution must be saved at every reset so that block search can start at the breakpoint. The FB MCV_ReadActualPathInfo supplied the required information. For this reason it should be called cyclically and always be enabled.

### Subtask 2: Parameter definition of block search

To ensure an NC program runs in block search mode, the parameters for the required block search type must be sent to the motion controller before the NC program starts. This is implemented by setting the “Execute” input to TRUE. An FB is defined in the library for every block search type available:

- Block search defined by block counter: MCV_BlockSearchSetBlockCount [▶ 15]
- Block search defined by block number: MCV_BlockSearchSetBlockNumber [▶ 18]
- Block search defined by file offset: MCV_BlockSearchSetFileOffset [▶ 21]

### Subtask 3: Handshake with the motion controller

An NC program is started in block search in the same way as without block search. However, if a block search type is enabled by the parameters, the PLC is informed of this after NC program start-up and can then take the required measures. The PLC sends an acknowledgement to the motion controller at the end of these preparations. The motion controller then signals that the continuation position is found and this is again acknowledged by the PLC.

The FB MCV_BlockSearchCtrl [▶ 11] is used for this task.

### Subtask 4: Disable block search

The block search type remains enabled after the NC program ends. To ensure that block search is not active at the next program start-up, it must be disabled. The FB MCV_BlockSearchDeselect [▶ 14] is used for this purpose.

## 4.3 MCV_BlockSearchCtrl

This function block is used to inform the PLC of block search status.

**Enable:**

- Set “Enable” input to TRUE
- "Valid" output displays TRUE

Once the block is enabled by the PLC; the motion controller (MC) signals the state of the blocks search at the outputs of the FB.

The “Selected” output indicates whether the MC was configured for block search. The next time an NC program is started, no axis motion takes place until the continuation position is found.

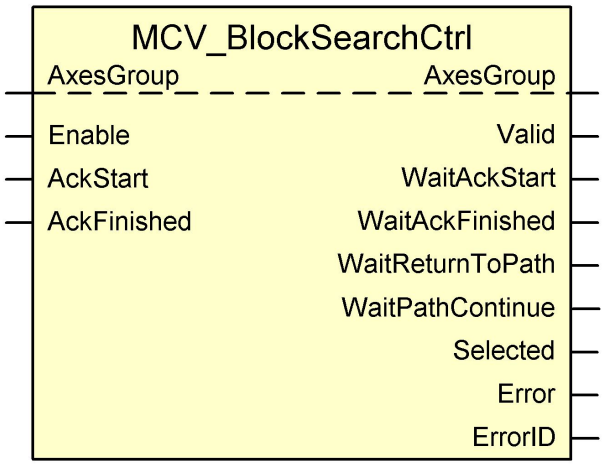
All "Wait…” outputs reflect the block search state in the MC.

The PLC sends an acknowledgement over the “AckStart” input to indicate that it was informed by the “WaitAckStart” signal that the MC will start the NC program in block search mode after the PLC has completed the necessary measures, e.g. suppress hardware reaction after the MC outputs technology functions.

The “AckFinished” input is used by the PLC to acknowledge the “WaitAckFinished” output signal, e.g. after hardware reactions are re-enabled after the output of technology functions.

The "WaitReturnToPath" and "WaitPathContinue" output signals are acknowledged by restarting the NC program, e.g. by confirming the START button in an HMI.

**Block diagram**




**FB parameters**

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

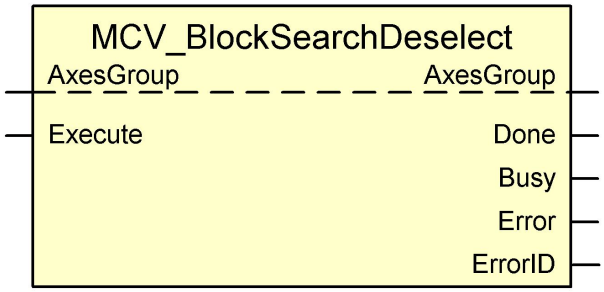
| VAR_INPUT | | |
|---|---|---|
| **Variable name** | **Data type** | **Description** |
| Enable | BOOL | If “Enable” is TRUE, the PLC is registered for the block search interface and the MC signals the state of block search over the "Wait…” outputs.<br>The MC then waits for the PLC to acknowledge the block search states signalled. |
| AckStart | BOOL | The output signal "WaitAckStart” is acknowledged on the rising edge. This is how the PLC indicates that it has completed its preparations for block search and the MC can start to find the continuation position. |
| AckFinished | BOOL | The output signal "WaitAckFinished” is acknowledged on the rising edge. This is how the PLC indicates that it has received the finished message that the continuation position was found and its preparations are completed so that real axis motions can finally be resumed. |

| VAR_OUTPUT | | |
|---|---|---|
| **Variable name** | **Data type** | **Description** |
| Valid | BOOL | When this output is TRUE, the function block outputs indicate the block search state. |
| WaitAckStart | BOOL | If the output is TRUE, an NC program was started in block search mode and the MC waits for acknowledgement from the PLC at the “AckStart” input before searching for the continuation position which was previously configured by one of the FBs MCV_BlockSearchSetBlockCount [▶ 15] or MCV_BlockSearchSetBlockNumber [▶ 18] or MCV_BlockSearchFileOffset [▶ 21]. |
| WaitAckFinished | BOOL | If the output is TRUE, the continuation position was found in the NC program and the MC waits for acknowledgement from the PLC at the “AckFinished” input. Up to this point no real axis motion takes place. |
| WaitReturnToPath | BOOL | This output is TRUE once the PLC signals via “AckFinished” that the PLC has completed the preparations to permit the execution of real axis motions. This is the signal for the user to execute an action to move the axes to the continuation position if they are not already at this position. Normally this takes place by pressing the START button in the HMI.<br>**Note:**<br>The start movement then takes place directly. |
| WaitPathContinue | BOOL | TRUE at this output indicates that the axes are located at the continuation position within the permitted deviation. After the ensuring enable, the NC program executes real axis motions starting at the continuation position. Enable normally takes place by pressing the START button in the HMI.<br>The permitted deviation was already defined when block search was configured by the "Deviation" input in the FB MCV_BlockSearchSetBlockCount [▶ 15] or MCV_BlockSearchSetBlockNumber [▶ 18] or MCV_BlockSearchFileOffset [▶ 21]. |
| Selected | BOOL | TRUE indicates that block search was selected by FB MCV_BlockSearchSetBlockCount [▶ 15] or MCV_BlockSearchSetBlockNumber [▶ 18] or MCV_BlockSearchFileOffset [▶ 21]. As a result no real axis motion takes place the next time an NC program is started until a continuation position was found. |
| Error | BOOL | TRUE indicates that an error occurred |
| ErrorID | WORD | Error identifier |

## 4.4 MCV_BlockSearchDeselect

The function block disables block search.

**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 block search function is deselected on the rising edge. |

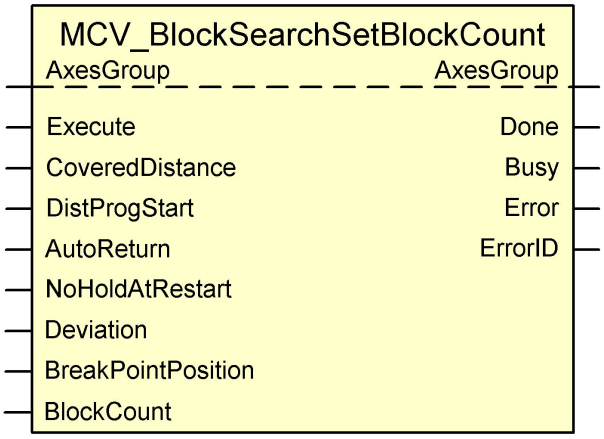
| VAR_OUTPUT | | |
|---|---|---|
| **Variable name** | **Data type** | **Description** |
| Done | BOOL | TRUE indicates that block search was deselected. |
| Busy | BOOL | TRUE indicates that the FB is executing a command |
| Error | BOOL | TRUE indicates that an error occurred. |
| ErrorID | WORD | Error identifier |

## 4.5 MCV_BlockSearchSetBlockCount

This FB defines the continuation position by the number of NC program lines read by the MC.

The block counter contains all the NC program lines read by the MC, i.e. repeatedly read lines, empty and comment lines.

**Block diagram**




**FB parameters**

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

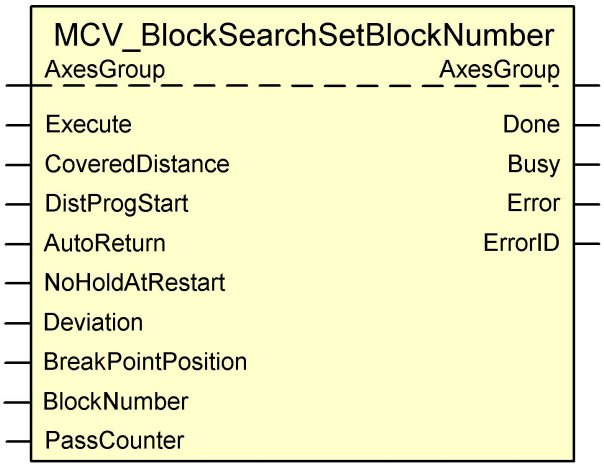
| VAR_INPUT | | |
|---|---|---|
| **Variable name** | **Data type** | **Description** |
| Execute | BOOL | The values of the inputs are transferred to the MC on the rising edge to define the search for the continuation position. |
| CoveredDistance | LREAL | If the continuation position searched for is not at the start of the NC program line defined by the value specified at the “BlockCount” input, this input specifies which portion of the distance programmed in the NC program line defines the continuation position.<br>Unit: 0.1% |
| DistProgStart | LREAL | This defines the continuation position search for by means of the distance covered since program start or since the last NC command #DISTANCE PROG START CLEAR. This can also define a continuation position located between the start and end of a motion block.<br>Unit: 0.1µm |
| AutoReturn | BOOL | This option defines whether the axes are enabled to move directly to the continuation position after the continuation position is found and after the signal that the MC is waiting for the enable to move to the contour ("WaitReturnToPath" output at the FB MCV_BlockSearchCtrl [▶ 11]).<br>If this option is TRUE, the value at the “Deviation” input is not invoked to check the deviation between the continuation position and the actual position of the axes. |
| NoHoldAtRestart | BOOL | If this input is TRUE, the NC program continues its execution immediately after the enable signal to move to the contour without any further user input. |
| Deviation | UDINT | The value defines the maximum permissible deviation of the axis position from the continuation position after the continuation position is found.<br>Unit: 0.1µm |
| BreakpointPosition | LREAL | Indicates an additional breakpoint position as distance from program start.<br>The breakpoint must be located after the continuation position.<br>Unit: 0.1µm |
| BlockCount | UDINT | The value is derived from the number of NC program lines which the MC has read since the NC program started. All the lines read by the MC are counted, i.e. repeatedly read lines, empty and comment lines. |

| VAR_OUTPUT | | |
|---|---|---|
| **Variable name** | **Data type** | **Description** |
| Done | BOOL | TRUE indicates that the block search parameters were successfully written to the motion controller. |
| Busy | BOOL | TRUE indicates that the FB is executing a command |
| Error | BOOL | TRUE indicates that an error occurred |
| ErrorID | WORD | Error identifier |

## 4.6 MCV_BlockSearchSetBlockNumber

This FB defines the continuation position by the block number (N word) of a line containing NC commands. Optionally the continuation position can also be found after the block number has occurred n times.

**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 values of the inputs are transferred to the MC on the rising edge to define the search for the continuation position. |
| CoveredDistance | LREAL | If the continuation position searched for is not at the start of the NC program line defined by the value specified at the “BlockCount” input, this input specifies which portion of the distance programmed in the NC program line defines the continuation position.<br>Unit: 0.1% |
| DistProgStart | LREAL | This defines the continuation position search for by means of the distance covered since program start or since the last NC command #DISTANCE PROG START CLEAR. This can also define a continuation position located between the start and end of a motion block.<br>Unit: 0.1µm |
| AutoReturn | BOOL | This option defines whether the axes are enabled to move directly to the continuation position after the continuation position is found and after the signal that the MC is waiting for the enable to move to the contour ("WaitReturnToPath" output at the FB MCV_BlockSearchCtrl [▶ 11]).<br>If this option is TRUE, the value at the “Deviation” input is not invoked to check the deviation between the continuation position and the actual position of the axes. |
| NoHoldAtRestart | BOOL | If this input is TRUE, the NC program continues its execution immediately after the enable signal to move to the contour without any nurther user input. |
| Deviation | UDINT | The value defines the maximum permissible deviation of the axis position from the continuation position after the continuation position is found.<br>Unit: 0.1µm |
| BreakpointPosition | LREAL | Indicates an additional breakpoint position as distance from program start.<br>The breakpoint must be located after the continuation position.<br>Unit: 0.1µm |
| BlockNumber | UDINT | This input defines the continuation position by the block number. The block number is the number (N word) of an NC line in the NC program. |

| | | |
|---|---|---|
| | | **Note**<br>To generate the NC program, the operator or the system is responsible for providing every line which is relevant later, even only once, with a unique block number. |
| PassCounter | UDINT | Since the block number is detected several times, e.g. in loops, it is not always unique on its own. In this case, the continuation position may be optionally defined by the additional specification of a pass counter. |

| **VAR_OUTPUT** | | |
|---|---|---|
| **Variable name** | **Data type** | **Description** |
| Done | BOOL | TRUE indicates that the block search parameters were successfully written to the Motion Controller. |
| Busy | BOOL | TRUE indicates that the FB is executing a command |
| Error | BOOL | TRUE indicates that an error occurred |
| ErrorID | WORD | Error identifier |

## 4.7 MCV_BlockSearchSetFileOffset

This block search defines a range in which real axis motions can be executed by specifying a continuation/end position (start/end flag) using file offset.

The continuation position is defined by the values at the “Start...” inputs of the function block. The end position is defined by the values at the “End...” inputs.

### Notice

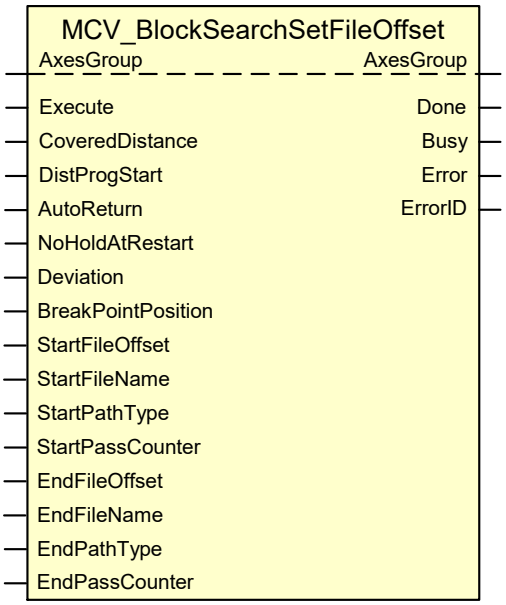
File offset must be specified so that it describes the start of a line in a particular NC program (main program, global subroutine). A possible existing block number (N word) also belongs to the NC line.

**Options** also include assigning values for the "StartPassCounter" input and **all** "End…” inputs.

If the "End…" inputs assign **no** values, the NC program is executed with real axis motions up to NC program end M30 after the continuation position is found.

If the “End...” inputs were assigned values and the defined end position is reached, execution is aborted and the NC program ends immediately.

### 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 values of the inputs are transferred to the MC on the rising edge to define the search for the continuation position. |
| CoveredDistance | LREAL | If the continuation position searched for is not at the start of the NC program line defined by the value specified at the “BlockCount” input, this input specifies which portion of the distance programmed in the NC program line defines the continuation position.<br>Unit: 0.1% |
| DistProgStart | LREAL | This defines the continuation position search for by means of the distance covered since program start or since the last NC command #DISTANCE PROG START CLEAR. This can also define a continuation position located between the start and end of a motion block.<br>Unit: 0.1µm |
| AutoReturn | BOOL | This option defines whether the axes are enabled to move directly to the continuation position after the continuation position is found and after the signal that the MC is waiting for the enable to move to the contour ("WaitReturnToPath" output at the FB MCV_BlockSearchCtrl [▶ 11]).<br>If this option is TRUE, the value at the “Deviation” input is not invoked to check the deviation between the continuation position and the actual position of the axes. |
| NoHoldAtRestart | BOOL | If this input is TRUE, the NC program continues its execution immediately after the enable signal to move to the contour without any further user input. |
| Deviation | UDINT | The value defines the maximum permissible deviation of the axis position from the continuation position after the continuation position is found.<br>Unit: 0.1µm |
| BreakpointPosition | LREAL | Indicates an additional breakpoint position as distance from program start.<br>The breakpoint must be located after the continuation position.<br>Unit: 0.1µm |

| | | |
|---|---|---|
| StartFileOffset | DINT | Defines the continuation position by specifying the file offset of an NC line in the NC program.<br>**Note:**<br>File offset must therefore always point to the start of a line in the NC program. It may also contain an existing block number. |
| StartFileName | MCV_FILENAME_STR | This is the file name of the file containing the NC program in which the continuation position is located. |
| StartPathType | UINT | Defines whether the file specified at the “StartFileName” input is located in the path of the NC main programs or NC subroutines:<br>Possible values:<br>0 – main program path (default value)<br>1 – subroutine path |
| StartPassCounter | DINT | The counter indicating the number of times the program is to start before the value is checked by the StartFileOffset input is checked to determine the continuation position.<br>**Note:**<br>The value 0 (default value) for the counter means that only the StartFileOffset determines the continuation position. This means that the same behaviour is obtained for block search for the values 0 and 1. |
| EndFileOffset | DINT | Defines the end position by specifying the file offset of an NC line in the NC program.<br>**Note:**<br>File offset must therefore always point to the start of a line in the NC program. It may also contain an existing block number. |
| EndFileName | MCV_FILENAME_STR | This is the file name of the file containing the NC program in which the end position is located. |
| EndPathType | UINT | Defines whether the file specified at the “EndFileName” input is located in the path of the NC main programs or NC subroutines:<br>Possible values:<br>0 – main program path (default value)<br>1 – subroutine path |

| | | |
|---|---|---|
| EndPassCounter | DINT | The counter indicating the number of times the program is to start before the value is checked by the EndFileOffset input is checked to determine the breakpoint.<br>**Note:**<br>The value 0 (default value) for the counter means that only EndFileOffset determines the continuation position. This means that the same behaviour is obtained for block search for the values 0 and 1. |

| **VAR_OUTPUT** | | |
|---|---|---|
| **Variable name** | **Data type** | **Description** |
| Done | BOOL | TRUE indicates that the block search parameters were successfully written to the motion controller. |
| Busy | BOOL | TRUE indicates that the FB is executing a command |
| Error | BOOL | TRUE indicates that an error occurred |
| ErrorID | WORD | Error identifier |

# 5 Function blocks for manual mode

## 5.1 MCV_ManMode

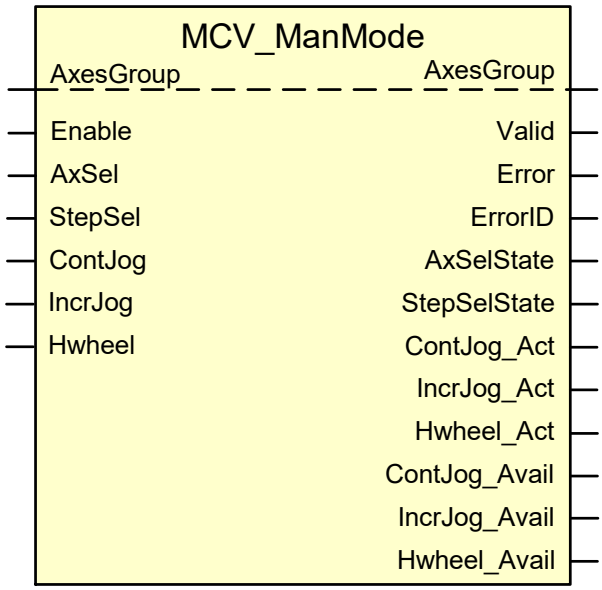
This FB acts on an axis group (NC channel) when it is in manual mode. (See FB MCV_GrpOperationMode)

For every axis in the axis group, it defines

- whether an axis is moved in manual mode
- which operating element (key pair or handwheel) supplies the information to move the axis

and what the parameters of the selected manual mode are.

**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 input is TRUE, the commands applied to the inputs are commanded for each of the axis groups (NC channel) if they are in manual mode. |
| AxSel | MCV_AY_MM_AX_SEL | This input enables axes for manual mode and connects to an operating element (key pair or handwheel).<br>This is an array containing elements of the data structure MCV_MM_AX_SEL [▶ 27]. Every array element corresponds with the axis which is sorted by the same index in the sequence of the axes in the axis group. |
| StepSel | MCV_AY_MM_STEP_SEL | Several parameter data records can be configured for each axis for each manual mode. This input is used to select the parameter data records to be applied in the current manual mode situation.<br>This input defines which parameter records are to be applied by specifying the index under which the parameter data record is entered in the axis parameter list.<br>Reference to AXIS (internal |
| ContJog | BOOL | Selects continuous jog mode on a rising edge. Deselects by selecting IncrJog or Hwheel. |
| IncrJog | BOOL | Selects incremental jog mode on a rising edge. Deselects by selecting ContJog or Hwheel. |
| Hwheel | BOOL | Selects handwheel mode on a rising edge. Deselects by selecting ContJog or IncrJog. |

| VAR_OUTPUT | | |
|---|---|---|
| **Variable name** | **Data type** | **Description** |
| Valid | BOOL | TRUE when the inputs are evaluated. |
| Error | BOOL | Indicates TRUE when an error occurs. |
| ErrorID | WORD | Error identifier |
| AxSelState | MCV_AY_MM_AX_SEL_STATE | This output indicates the current situation, which axes are enabled for manual mode and the operating element they are connected to.<br>This is an array containing elements of the data structure MCV_MM_AX_STATE [▶ 27]. Every array element corresponds to the axis which is sorted by the same index in the sequence of the axes in the axis group. |
| StepSelState | MCV_AY_MM_STEP_SEL_STATE | This output indicates whether each selectable parameter data record was enabled or is ready to be enabled.<br>This is an array containing elements of the data structure MCV_MM_STEP_SEL_STATE [▶ 28] |
| ContJog_Act | BOOL | Continuous jog mode is activated. |
| IncrJog_Act | BOOL | Incremental jog mode is activated |
| Hwheel_Act | BOOL | Handwheel mode is activated. |
| ContJog_Avail | BOOL | Continuous jog mode is ready. |
| IncrJog_Avail | BOOL | Incremental jog mode is ready. |
| Hwheel_Avail | BOOL | Handwheel mode is ready. |

## Data structure MCV_MM_AX_SEL

Before an axis can move in manual mode, it must be enabled for manual mode and be connected to an operating element. The structure variables are used as follows:

- **SelAxis**: This element defines whether an axis was enabled for manual mode (TRUE) or disabled (FALSE).
- **CtrlElemId**: The identification number defines which operating element acts on the axis in manual mode. Depending on the manual mode type, this is one of the values which were defined by the manual mode parameter P-MANU-00002 or P-MANU-00009.

## Data structure MCV_MM_AX_SEL_STATE

This data structure indicates the current state of an axis relating to manual mode. The structure variable

- **AxAct**: indicates with TRUE that the axis was enables for manual mode.
- **AxAvail**: indicates with TRUE that the axis is ready for manual mode but was not yet selected.
- **CtrlElemId**: indicates the identification number of the operating element which influences the motion of the axis in manual mode.

## Data structure MCV_MM_STEP_SEL_STATE

This data structure indicates which of the parameter data records defined in the axis parameter list was enabled for the currently selected manual mode type.

This data structure indicates the currently selected manual mode type for the parameter data records configured in the axis parameter list, whether the single parameter data record was enabled or is ready to enable.

- **StepAct**: indicates with TRUE that a particular parameter data record is enabled.
- **StepAvail**: indicates with TRUE that the parameter data record is selectable.

**Notice**

**The elements StepAct and StepAvail in a parameter data record may never be TRUE simultaneously.**

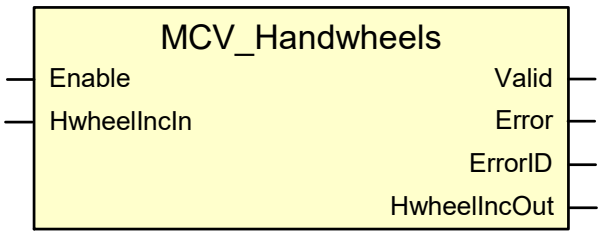
## 5.2 MCV_Handwheels

The increments output by handwheels are appended to this function block in the form of a counter state. The following criteria must be met to ensure that a change in handwheel counter state executes an axis:

The axis group to which an axis belongs must be in **manual mode** (see FB MCV_GrpOperationMode [▶ 39]).

**Handwheel** mode must be activated for the axis and the axis must be linked to a handwheel (see FB MCV_ManMode [▶ 25]).

### Block diagram




### FB parameters

| VAR_INPUT | | |
|---|---|---|
| **Variable name** | **Data type** | **Description** |
| Enable | BOOL | If the input is TRUE, the motion controller transfers the counter states of the handwheel increments to the axes whose axis group is in **manual mode** and whose handwheel interface was linked to a handwheel. |
| HwheelIncIn | MCV_AY_MM_HW_INC | The signed counter state of a handwheel is transferred to the motion controller by each element in this array. |

| VAR_OUTPUT | | |
|---|---|---|
| **Variable name** | **Variable name** | **Variable name** |
| Valid | BOOL | Is TRUE when TRUE is present at the “Enable” input. |
| Error | BOOL | Indicates TRUE when an error occurs. |
| ErrorID | INT | Error identifier |
| HwheelIncOut | MCV_AY_MM_HW_INC | Every element in the array indicates the counter state of the corresponding input “HwheelIncIn[*x*]” which was transferred to the motion controller. |

**Attention**

If axes in different axis groups which are in **manual mode** are connected to the same handwheel, all these axes move when the handwheel counter state changes.

## 5.3 MCV_ManModeKeys

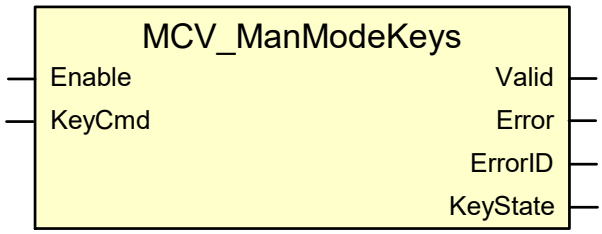
The outputs of keys used as input elements to move axes in incremental or continuous jog mode are connected to this function block.

The following criteria must be met to ensure that an axis moves when these key are pressed:

The axis group to which an axis belongs must be in **manual mode** (see FB MCV_GrpOperationMode [▶ 39]).

One of the **incremental** or **continuous jog mode** operation modes must be activated for the axis and the axis must be linked to an input element (see FB MCV_ManualMode [▶ 25]).

### Block diagram




### FB parameters

| VAR_INPUT | | |
|---|---|---|
| **Variable name** | **Data type** | **Description** |
| Enable | BOOL | If the input is TRUE, the commands applied to the inputs are commanded for each of the axis groups (NC channel) if they are in manual mode. |
| KeyCmd | MCV_AY_MM_KEY_CMD | This input transfers the signals of manual mode keys so that the motion controller starts the corresponding motion.<br>This is an array containing elements of the data structure MCV_MM_KEY_CMD [▶ 31].<br>The array element [x] then affects all axes which are connected to the operating element where CtrlElemId = x + 1 (see FB MCV_ManMode [▶ 25], data structure MCV_MM_AX_SEL [▶ 27]). |

| VAR_OUTPUT | | |
|---|---|---|
| **Variable name** | **Data type** | **Description** |
| Valid | BOOL | Is TRUE when TRUE is present at the “Enable” input. The function block then acts on the axis group in manual mode. |
| Error | BOOL | Indicates TRUE when an error occurs. |
| ErrorID | INT | Error identifier |
| KeyState | MCV_AY_MM_KEY_STATE | The signals of the manual mode keys are output via this output.<br>This is an array containing elements of the data structure MCV_MM_KEY_STATE [▶ 31]. |

**Attention**

If axes in different axis groups which are in **manual mode** are connected to the same key pair, all these axes move when one of the key pair is pressed.

## Data structure MCV_MM_KEY_CMD

This structure is used to transfer axis motion signals to the motion controller in incremental or continuous jog mode. The structure variables are used as follows:

- **PosDir**: is assigned TRUE if the key pair is assigned to move the connected axes in a positive direction.
- **NegDir**: is assigned TRUE if the key pair is assigned to move the connected axes in a negative direction.
- **Rapid**: is assigned TRUE if the connected axes are to execute the motion in rapid traverse.

## Data structure MCV_MM_KEY_STATE

This data structure indicates the current state or a key pair.

- PosDirAct: indicates TRUE if a positive motion direction is commanded.
- NegDirAct: indicates TRUE if a negative motion direction is commanded.
- RapidAct: indicates TRUE if rapid traverse is selected.

- PosDirAvail: TRUE indicates that positive motion direction is not selected.
- NegDirAvail: TRUE indicates that negative motion direction is not selected.
- RapidAvail: TRUE indicates that rapid traverse is not selected.

# 6 General function blocks

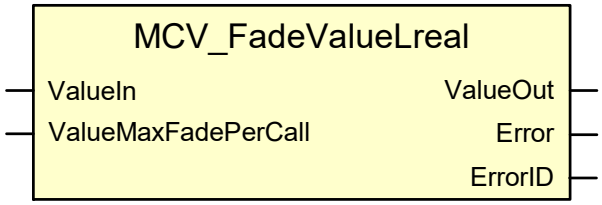
## 6.1 MCV_FadeValueLreal

This FB is a utility FB which can be used in a PLC application.

This function block is used to output a delayed change in the value at the “ValueIn” input to the “ValueOut” output.

Here the user has the option to influence the change speed by specifying “ValueMaxFadePerCall” at the input and the amount by which the current value should change at the “ValueOut” per call until the value present at “ValueIn” is reached.

**Block diagram**




**FB parameters**

| **VAR_INPUT** | | |
|---|---|---|
| **Variable name** | **Data type** | **Description** |
| ValueIn | LREAL | Target value which should be output at the “ValueOut” output.<br>The “ValueIn“ input is defaulted to the value 0.0. |
| ValueMaxFadePerCall | LREAL | Defines the change in the value currently output at the “ValueOut” output every time the function block is called.<br>The change is effective until the same value is output at the “ValueOut” output as is present at the “ValueIn” input. |

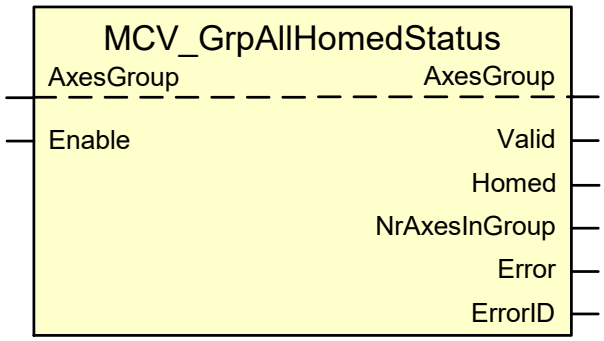
| **VAR_OUTPUT** | | |
|---|---|---|
| **Variable name** | **Data type** | **Description** |
| ValueOut | LREAL | Current output value |
| Error | BOOL | Indicates TRUE when an error occurs. |
| ErrorID | WORD | Error identifier |

## 6.2 MCV_GrpAllHomedStatus

This FB is a utility FB which can be used in a PLC application.

The FB checks whether all axes assigned to an axis group are referenced and indicates how many axes are assigned to this 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 | It TRUE is applied to the input, the FB checks every time it is called whether all axes in an axis group are referenced. |

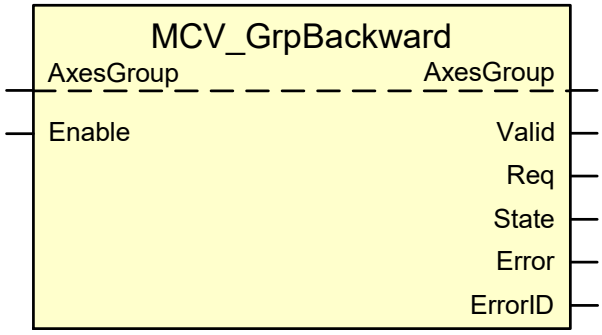
| VAR_OUTPUT | | |
|---|---|---|
| **Variable name** | **Data type** | **Description** |
| Valid | BOOL | TRUE when the Enable input is TRUE. |
| Homed | BOOL | Is TRUE when all axes in the axis group are referenced. |
| NrAxesInGroup | UINT | Number of axes present in the axis group |
| Error | BOOL | Indicates TRUE when an error occurs. |
| ErrorID | WORD | Error identifier |

## 6.3 MCV_GrpBackward

This FB causes the NC kernel to move backwards along the programmed path.

For further information on backward motion, see the functional description [FCT-C7].

**Block diagram**




**FB parameters**

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

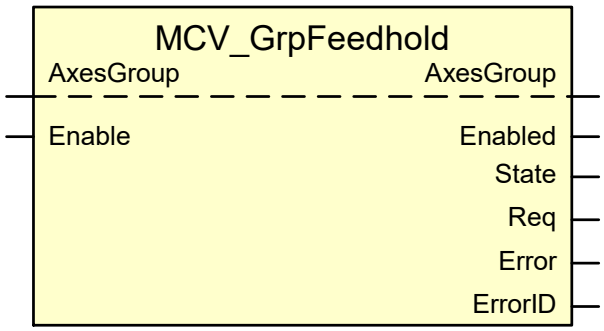
| **VAR_INPUT** | | |
|---|---|---|
| **Variable name** | **Data type** | **Description** |
| Enable | BOOL | The axes move backward on the path as long as the input is TRUE. |

| **VAR_OUTPUT** | | |
|---|---|---|
| **Variable name** | **Data type** | **Description** |
| Valid | BOOL | TRUE indicates that backward motion is active. |
| Req | BOOL | Indicates that another function commanded backward motion e.g. a GUI.<br>This command only becomes active when it is applied to the “Enable” input of the FB. |
| State | BOOL | TRUE indicates that backward motion is activated. |
| Error | BOOL | Indicates TRUE when an error occurs. |
| ErrorID | UDINT | Error identifier |

## 6.4 MCV_GrpFeedhold

This FB triggers a feed hold for an axis group during the interpolation. Deceleration takes place down to the feedrate 0 at the acceleration values entered in the axis group-specific configuration list (see [CHAN]).

**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 | Feed hold is activated for the axis group as long as the input is TRUE. |

| VAR_OUTPUT | | |
|---|---|---|
| **Variable name** | **Data type** | **Description** |
| Enabled | BOOL | Is TRUE when feed hold is activated. |
| State | BOOL | Status from the MCP that feed hold<br>is activated. |
| Req | BOOL | TRUE indicates that another control system unit intends to command a feed hold. If this is to be implemented, the Enable input must be assigned TRUE in the PLC application. |
| Error | BOOL | Indicates whether an error has occurred in an FB. |
| ErrorID | WORD | Error identifier |

## 6.5 MCV_GrpGetNextPrevAx

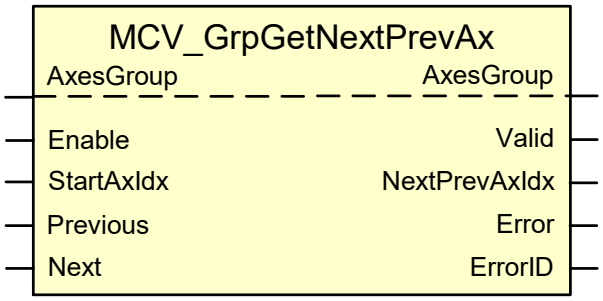
This FB is a utility FB which can be used in a PLC application.

It is used by the FB MCV_SingleAxisSelect.

With reference to a specified axis, it is used to determine the previous or next axis from the list of axes contained in the axis group.

The result is output as a numerical value. The numerical value is the index which is used to find the searched-for axis from the list of axes in 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 | Provided TRUE is present at the input, the index of the searched-for axis in the axis group is determined dependent on the inputs “StartAxIdx”, “Previous” and “Next”. |
| StartAxIdx | INT | Index of the axis in the axis group (channel) at which the search for the previous or next axis is started, if a search is made.<br>The search is started when the “Enable” input is set to TRUE for the first time. |
| Previous | BOOL | If the input is TRUE, a search is made for a lower index in this sequence after the next axis in the sequence of axes in the axis group. |
| Next | BOOL | If the input is TRUE, a search is made for a higher index in this sequence after the next axis in the sequence of axes in the axis group. |

| VAR_OUTPUT | | |
|---|---|---|
| **Variable name** | **Data type** | **Description** |
| Valid | BOOL | TRUE when the inputs are evaluated. |
| NextPrevAxIdx | INT | Index of the searched-for axis in the sequence of axes in the axis group. |
| Error | BOOL | Indicates TRUE when an error occurs. |
| ErrorID | WORD | Error identifier |

**Behaviour of the FB:**

On a rising edge, the “Prev“ and “Next“ inputs cause a change in the outputs.

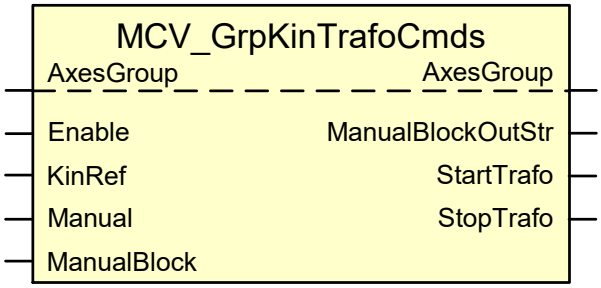
## 6.6 MCV_GrpKinTrafoCmds

This FB is a utility FB which is used by the McpPLCopen Part4 Extension library itself.

Only when **manual mode** is active for an axis group, the string which is the parameter for manual mode is extended to select or deselect a kinematic transformation depending on the signal at the Enable input.

To achieve this, the string at the “ManualBlock” input is placed in front of the NC command to activate or deactivate.

**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 | TRUE activates the kinematic transformation specified by KinRef. If FALSE is applied to this input, the currently kinematic transformation is deselected. |
| KinRef | MC_KIN_REF | Reference to the kinematic transformation. |
| Manual | BOOL | The signal that indicates that manual mode is active is applied to this input. |
| ManualBlock | STRING(HLI_IMCM_MODE-_STATE_PARAM_STRL) | The string that acts as parameter when manual mode is activated is applied to this input. |

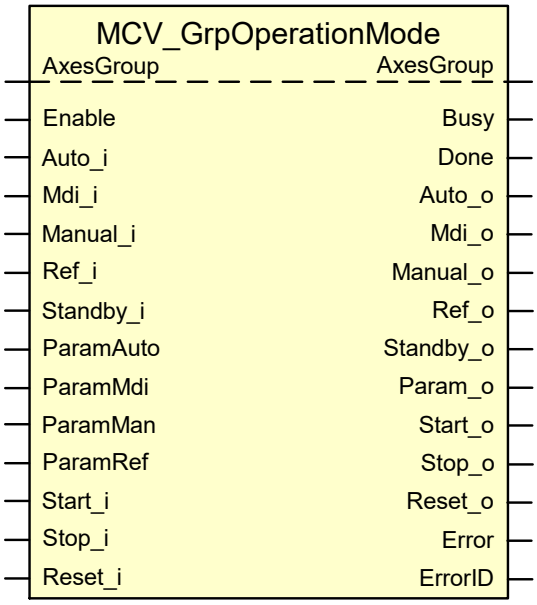
| **VAR_OUTPUT** | | |
|---|---|---|
| **Variable name** | **Data type** | **Description** |
| ManualBlockOutStr | BOOL | Supplies the string which must be applied when hand mode is activated if activating and deactivating a kinematic transformation is to be considered. |
| StartTrafo | BOOL | TRUE indicates that kinematic transformation is activated. |
| StopTrafo | BOOL | TRUE indicates that kinematic transformation is deactivated. |

## 6.7 MCV_GrpOperationMode

This FB is a utility FB which is used by the McpPLCopen Part4 Extension library itself.

It is used to switch an NC channel to the required mode. The input signals are evaluated, the required commands are generated and sent to the NC kernel via the interface.

**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 this input is TRUE, the FB can command the modes of the NC kernel. |
| Auto_i | BOOL | Rising edge switches the NC kernel to AUTOMATIC mode. |
| Mdi_i | BOOL | Rising edge switches the NC kernel to MANUAL BLOCK mode. |
| Manual_i | BOOL | Rising edge switches the NC kernel to MANUAL MODE. |
| Ref_i | BOOL | Rising edge switches the NC kernel to HOMING mode. |
| Standby_i | BOOL | Rising edge switches the NC kernel to BASE STATE mode. |
| ParamAuto | STRING(HLI_IMCM_MODE_STATE_PARAM_STRL) | NC program name in AUTOMATIC mode |
| ParamMdi | STRING(HLI_IMCM_MODE_STATE_PARAM_STRL) | Manual block in AUTOMATIC mode |
| ParamMan | STRING(HLI_IMCM_MODE_STATE_PARAM_STRL) | If nothing or an empty string is transferred, manual mode is activated. G200 is then commanded internally in the motion controller.<br>If a string is transferred, it must be a valid NC block which must terminate at its end with G200.<br>**Example**: Assume a transformation is to be active in manual mode.<br>#TRAFO ON$R$NG200<br>The term $R$N is a string defining a line break in structure text according to IEC 61131. |
| ParamRef | STRING(HLI_IMCM_MODE_STATE_PARAM_STRL) | If nothing or an empty string is transferred, the NC program with the name rpf.nc is started. The program must be located in one of the paths which were defined by the parameters P-STUP-00018 to P-STUP-00021 in the start-up list or which are definable depending on the control system.<br>If a string is transferred, it have be a valid NC block.<br>**Example 1**: Homing programmed via manual block:<br>G74 X1 Y2 Z3<br><br>**Example 2**: Automatic program can only be called as subroutine<br>L sup_rpf.nc |
| Start_i | BOOL | Start execution of NC programs or manual blocks in the operation mode activated. |
| Stop_i | BOOL | Stop processing an operation mode. |
| Reset_i | BOOL | Reset an operation mode. |

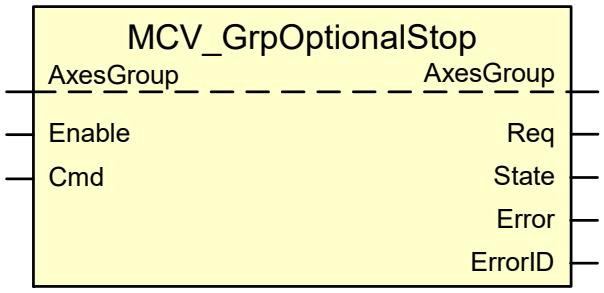
| VAR_OUTPUT | | |
|---|---|---|
| **Variable name** | **Data type** | **Description** |
| Busy | BOOL | TRUE as long as a command is set to the NC kernel. |
| Done | BOOL | TRUE when the command to change the mode was successfully sent to the NC kernel. |
| Auto_o | BOOL | TRUE when AUTOMATIC mode is selected. |
| Mdi_o | BOOL | TRUE when MANUAL BLOCK mode is selected. |
| Manual_o | BOOL | TRUE when MANUAL MODE is selected. |
| Ref_o | BOOL | TRUE when HOMING mode is selected. |
| Standby_o | BOOL | TRUE when BASIC STATE mode is selected. |
| Param_o | STRING(HLI_IMCM_MODE_STATE_PARAM_STRL) | Indicates the currently active parameter of the currently selected mode (value of one of the 4 parameter inputs) |
| Start_o | BOOL | TRUE when a mode was activated. |
| Stop_o | BOOL | TRUE when processing in a mode was stopped. |
| Reset_o | BOOL | TRUE while a mode is reset. |
| Error | BOOL | Indicates TRUE when an error occurs. |
| ErrorID | INT | Error identifier |

## 6.8 MCV_GrpOptionalStop

The FB defines whether the processing of an NC program is stopped when an M01 occurs in the NC code or whether this M01 is ignored.

For further information on the Optional Stop, see the programming manual [PROG// Optional stop].

**Block diagram**




**FB parameters**

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

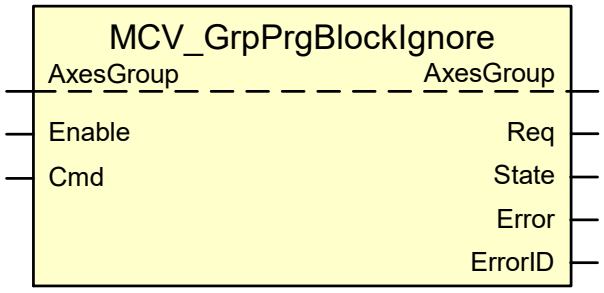
| VAR_INPUT | | |
|---|---|---|
| **Variable name** | **Data type** | **Description** |
| Enable | BOOL | A command can be sent via the “Cmd” input as long as the input is TRUE. |
| Cmd | BOOL | The optional stop is activated as long as TRUE is applied to the input. |

| VAR_OUTPUT | | |
|---|---|---|
| **Variable name** | **Data type** | **Description** |
| Req | BOOL | Indicates that another user commanded the optional stop, e.g. a GUI.<br>This command only becomes active when it is applied to the “Enable” input of the FB. |
| State | BOOL | TRUE indicates that the optional stop function is activated. |
| Error | BOOL | Indicates TRUE when an error occurs. |
| ErrorID | UDINT | Error identifier |

## 6.9 MCV_GrpPrgBlockIgnore

This FB can hide NC blocks which are then not executed when it is preceded by a **"/"** in the NC program (see [PROG//Hiding blocks]).

**Block diagram**




**FB parameters**

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

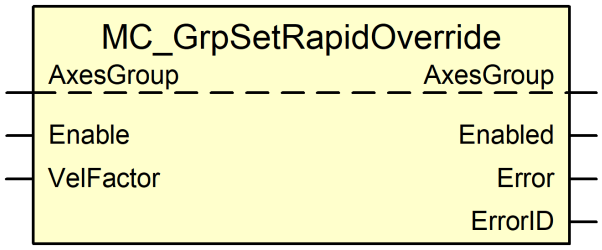
| **VAR_INPUT** | | |
|---|---|---|
| **Variable name** | **Data type** | **Description** |
| Enable | BOOL | A command can be sent via the "Cmd" input as long as the input is TRUE. |
| Cmd | BOOL | Specially marked NC blocks are skipped and not executed as long as TRUE is applied to the input. |

| **VAR_OUTPUT** | | |
|---|---|---|
| **Variable name** | **Data type** | **Description** |
| Req | BOOL | Indicates that another user commanded the hiding of NC blocks, e.g. a GUI.<br>This command only becomes active when it is applied to the "Enable" input of the FB. |
| State | BOOL | TRUE indicates that the function for hiding NC blocks is activated. |
| Error | BOOL | Indicates TRUE when an error occurs. |
| ErrorID | UDINT | Error identifier |

## 6.10 MCV_GrpSetRapidOverride

The FB MCV_GrpSetRapidOverride specifies a weighting factor which acts on the programmed rapid traverse 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 rapid traverse.<br>The channel parameters P-CHAN-00181 defines the impact of the weighting factor on rapid traverse. |
| VelFactor | LREAL | Weighting factor for the programmed rapid traverse.<br>The input is defaulted to the value 1.0. |

| VAR_OUTPUT | | |
|---|---|---|
| **Variable name** | **Data type** | **Description** |
| Enabled | BOOL | TRUE indicates that the velocity weighting factor is applied to the programmed rapid traverse. |
| Error | BOOL | TRUE indicates that an error occurred. |
| ErrorID | WORD | Error identifier |

**Behaviour of the FB:**

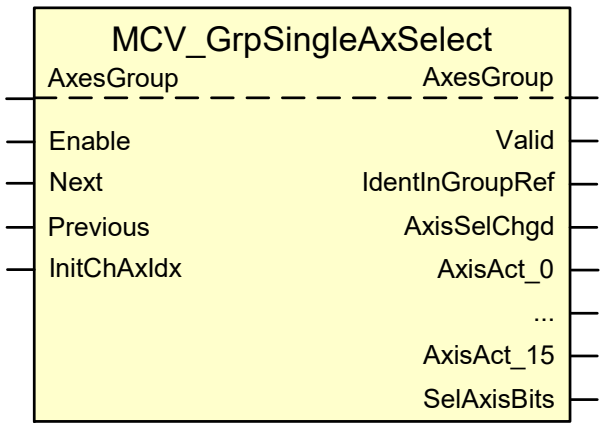
- Under no circumstance does the FB change the axes group state. This means that the value 0.0 for “VelFactor” leads to a stop of the axes in the axis group but the state of the axis group does not change after “GroupStop”.
- The value at “VelFactor” retains its validity until a new value is set. Accordingly,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

## 6.11 MCV_GrpSingleAxSelect

This FB is a utility FB which can be used in a PLC application.

With reference to a currently selected axis, it is used to select the previous or next axis from the list of axes contained in the axis group. The result is accessible at the FB both as a Boolean value and a binary encoded value. In addition the output conforming to PLCopen is supported as IDENT_IN_GROUP_REF.

**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 is indicated at the input, rising edges are evaluated at the “Next” and “Previous” inputs and the value is considered at the “InitChAxIdx” input. | |
| Next | BOOL | A rising edge has the result that with respect to the axis just selected, the next axis present in the axis group is detected and the corresponding output is set to TRUE.<br>In this context the next axis means an axis with a higher index than the sequence criterion in the list of axes present in the axis group. | |
| Previous | BOOL | A rising edge has the result that with respect to the axis just selected, the previous axis present in the axis group is detected and the corresponding output is set to TRUE.<br>In this context the previous axis means an axis with a lower index than the sequence criterion in the list of axis present in the axis group. | |
| InitChAxIdx | INT | Index of the axis in the axis group (channel) which is selected after the “Enable” input was set to TRUE the first time. | |
| | | -1 | no output is selected |
| | | 0 | “AxisAct_0“ output becomes TRUE |
| | | … | … |
| | | 15 | “AxisAct_15“ output becomes TRUE |

| VAR_OUTPUT | | |
|---|---|---|
| **Variable name** | **Data type** | **Description** |
| Valid | BOOL | TRUE when the inputs are evaluated. |
| IdentInGroupRef | IDENT_IN_GROUP_REF | Reference to the assignment of an axis in an axis group. |
| AxisSelChgd | BOOL | Is TRUE for one PLC cycle when selection has changed and therefore the value changed at the “SelAxisBits” output. |
| AxisAct_0 | BOOL | TRUE indicates the axis in an axis group which is the previous or next axis. |
| … | | |
| AxisAct_15 | | |
| SelAxisBits | WORD | Displays the successful selection as binary value. |

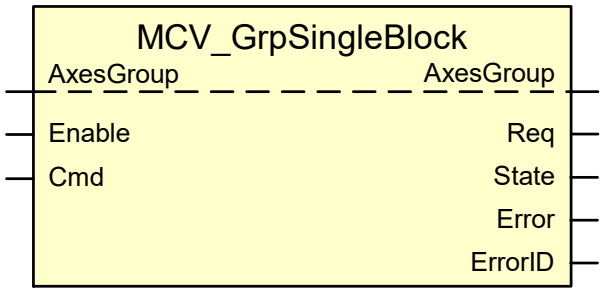
**Behaviour of the FB:**

- On a rising edge, the “Prev“ and “Next“ inputs cause a change in the outputs.
- The FB considers the option that the axes in an axis group may be assigned any position and the list of the axes in an axis group may contain gaps.

## 6.12 MCV_GrpSingleBlock

This FB activates single step mode while NC blocks are processed.

**Block diagram**




**FB parameters**

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

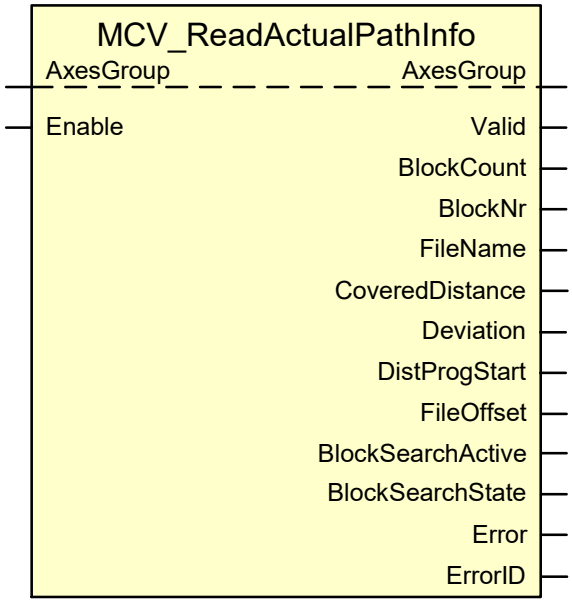
| VAR_INPUT | | |
|---|---|---|
| **Variable name** | **Data type** | **Description** |
| Enable | BOOL | A command can be sent via the “Cmd” input as long as the input is TRUE. |
| Cmd | BOOL | Single step mode is active when NC blocks are processed as long as TRUE is applied to the input. |

| VAR_OUTPUT | | |
|---|---|---|
| **Variable name** | **Data type** | **Description** |
| Req | BOOL | Indicates that another user commanded single step mode, e.g. a GUI.<br>This command only becomes active when it is applied to the “Enable” input of the FB. |
| State | BOOL | TRUE indicates that the single step function is activated. |
| Error | BOOL | Indicates TRUE when an error occurs. |
| ErrorID | UDINT | Error identifier |

## 6.13 MCV_ReadActualPathInfo

This block shows data to the currently executed NC program and moves the axes of the 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 Data type** | **Description** |
| Enable | BOOL | The FB supplies state values cyclically when this input is TRUE. |

| VAR_OUTPUT | | |
|---|---|---|
| **Variable name** | **Data type** | **Description** |
| Valid | BOOL | Indicates TRUE when valid state values are presented to the outputs. |
| BlockCount | UDINT | The number of NC program lines which the MC has read since the NC program started. All the lines read by the MC are counted, i.e. repeatedly read lines, empty and comment lines. |
| BlockNr | UINT | The block number (N word) of an NC line in the NC program. |
| FileName | MCV_FILENAME_STR | File name of the file that contains the NC program. |
| CoveredDistance | LREAL | This data item indicates what part of the distance programmed in the NC command was covered by an NC command which results in a motion.<br>Unit: 0.1% |
| Deviation | UDINT | Deviation from the axis position from the continuation position when block search was enabled and the continuation position is found. |
| DistProgStart | LREAL | The sum of the distance traversed since program start of the last occurrence of the NC command #DISTANCE PROG START CLEAR.<br>The NC commands #DISTANCE PROG START ON, #DISTANCE PROG START OFF control the distances contained in this sum. |
| FileOffset | DINT | File offset of the start of the NC program line which results in the currently executed path motion. |
| BlockSearchActive | BOOL | As long the block search process remains in the HLI_BS_ACTIVE [▶ 47] or HLI_BS_WAIT_FOR_PLC_OFF [▶ 47] state, this value indicates TRUE. |
| BlockSearchState | INT | Indicates the state of the block search.<br>The possible states are listed in Table 1. |
| Error | BOOL | TRUE indicates that an error occurred. |
| ErrorID | WORD | Error identifier |

### Block search states

| Value | Constant | Description |
|---|---|---|
| 0 | HLI_BS_INACTIVE | A block search was selected. |
| 1 | HLI_BS_WAIT_FOR_PLC_ON | An NC program was started in block search mode.<br>The MC waits for an acknowledgement from the PLC and then starts the search for the continuation position. |
| 2 | HLI_BS_ACTIVE | The MC searches for the continuation position. |
| 3 | HLI_BS_WAIT_FOR_PLC_OFF | The MC has found the continuation position and waits for an acknowledgement from the PLC: |
| 4 | HLI_BS_WAIT_RETURN_TO_CONTOUR | The MC waits for the signal to move the participating axes directing to the continuation position if they are not yet at the continuation position. |
| 5 | HLI_BS_RETURNING_TO_CONTOUR | The participating axes move directly to the continuation position. |
| 6 | HLI_BS_WAIT_FOR_CONTINUE_CONTOUR | The axes are located at the continuation position and the MC waits for the signal to continue the NC program. Once this signal is received, the axes move as programmed in the NC program. |

**Notice**

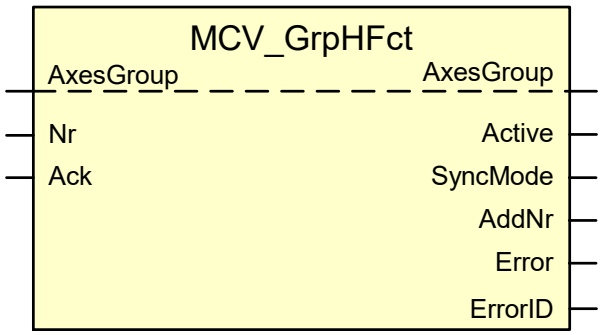
The signal is the continued execution of the NC program already started in block search and takes places by restarting the NC program on the user interface (press the START button in automatic mode) or by the PLC by commanding the “Continue motion” control unit on the PLC interface HLI.

# 7 Function blocks for technology functions

## 7.1 MCV_GrpHFct

This FB signals the output of an H technology function. It is acknowledged by the PLC provided it is a technology function that was parameterised in the channel parameter list as an axis-specific output function.

**Block diagram**




**FB parameters**

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

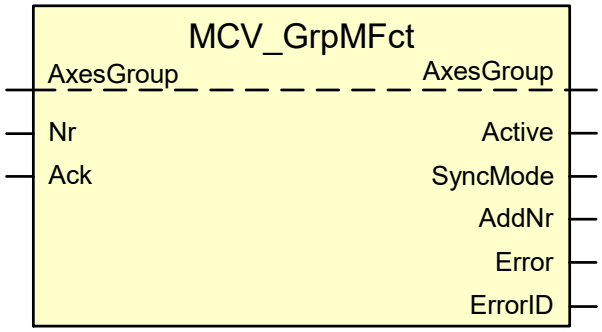
| VAR_INPUT | | |
|---|---|---|
| **Variable name** | **Data type** | **Description** |
| Nr | UINT | The number of the H technology function handled by the function block is defined via the input. |
| Ack | BOOL | The H technology function is acknowledged on the rising edge. |

| VAR_OUTPUT | | |
|---|---|---|
| **Variable name** | **Data type** | **Description** |
| Active | BOOL | The output indicates the value TRUE when the technology function was output to the PLC. The output remains TRUE until a rising edge is applied to the “Ack” input. |
| SyncMode | DWORD | Synchronisation type of an H technology function.<br>This is the value used to define the synchronisation type of the H technology functions in the channel parameter list with the parameter P-CHAN-00027. |
| AddNr | DINT | A number assigned to an H technology function by programming in the NC program and output when the H technology function is output to the the axis-specific interface in addition to the technology function number.<br>See [PROG//M/H functions with additional information]. |
| Error | BOOL | Is TRUE if an error occurs in the FB. |
| ErrorID | WORD | Error identifier |

## 7.2 MCV_GrpMFct

This FB signals the output of an M technology function. It is acknowledged by the PLC provided it is a technology function that was parameterised in the channel parameter list as an axis-specific output function.

**Block diagram**




**FB parameters**

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

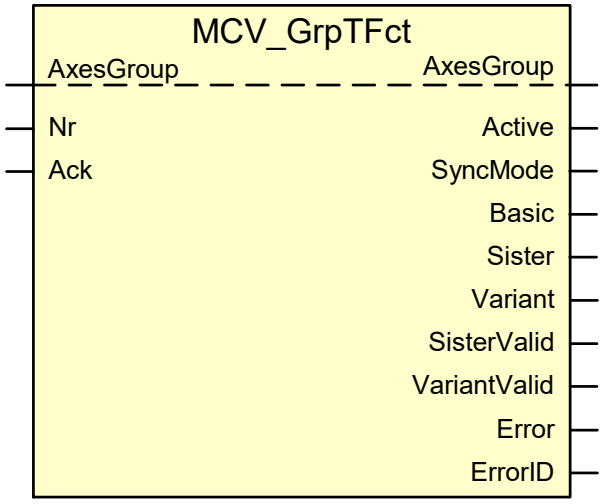
| VAR_INPUT | | |
|---|---|---|
| **Variable name** | **Data type** | **Description** |
| Nr | UINT | The number of the M technology function handled by the function block is defined via the input. |
| Ack | BOOL | The M technology function is acknowledged on the rising edge. |

| VAR_OUTPUT | | |
|---|---|---|
| **Variable name** | **Data type** | **Description** |
| Active | BOOL | The output indicates the value TRUE when the technology function was output to the PLC. The output remains TRUE until a rising edge is applied to the “Ack” input. |
| SyncMode | DWORD | Synchronisation type of an M technology function.<br>This is the value used to define the synchronisation type of the M technology functions in the channel parameter list with the parameter P-CHAN-00041. |
| AddNr | DINT | A number assigned to an M technology function by programming in the NC program and output when the M technology function is output to the the axis-specific interface in addition to the technology function number.<br>See [PROG//M/H- functions with additional information]. |
| Error | BOOL | Is TRUE if an error occurs in the FB. |
| ErrorID | WORD | Error identifier |

## 7.3 MCV_GrpTFct

This FB signals the output of a tool technology function. It is acknowledged by the PLC provided it is a technology function that was parameterised in the channel parameter list as an axis-specific output function.

**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** |
| Nr | UINT | The number of the tool technology function handled by the function block is defined via the input. |
| Ack | BOOL | The tool technology function is acknowledged on the rising edge. |

| VAR_OUTPUT | | |
|---|---|---|
| **Variable name** | **Data type** | **Description** |
| Active | BOOL | The output indicates the value TRUE when the technology function was output to the PLC. The output remains TRUE until a rising edge is applied to the “Ack” input. |
| SyncMode | DWORD | Synchronisation type of a tool technology function<br>This is the value used to define the synchronisation type of the tool technology functions in the channel parameter list with the parameter P-CHAN-00086. |
| Basic | DINT | Base number of the tool used to identify the tool type. |
| Sister | DINT | Sister number of the tool used to designate a definite specific tool item. |
| Variant | UDINT | The alternate number of a tool; used to identify different data blocks for a tool. |
| SisterValid | UINT | Identifier for the sister number of the tool |
| VariantValid | UINT | Identifier for the alternate number of the tool |
| Error | BOOL | Indicates TRUE if an error occurs in the FB. |
| ErrorID | WORD | Error identifier |

# 8 Appendix

## 8.1 Suggestions, corrections and the latest documentation

Did you find any errors? Do you have any suggestions or constructive criticism? Then please contact us at documentation@isg-stuttgart.de. The latest documentation is posted in our Online Help (DE/EN):

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

**The link above forwards you to:**

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

**Notice**

**Change options for favourite links in your browser;**

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

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

**PDFs for download:**

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

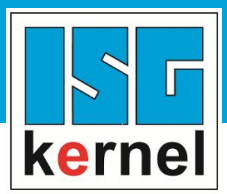
EN:
https://www.isg-stuttgart.de/en/products/softwareproducts/isg-kernel/documents-and-downloads

**E-Mail:** documentation@isg-stuttgart.de

# Keyword index

## M

MCV_BlockSearchCtrl ..... 11
MCV_BlockSearchDeselect ..... 14
MCV_BlockSearchSetBlockCount ..... 15
MCV_BlockSearchSetBlockNumber ..... 18
MCV_BlockSearchSetFileOffset ..... 21
MCV_FadeValueLreal ..... 32
MCV_GrpAllHomedStatus ..... 33
MCV_GrpBackward ..... 34
MCV_GrpFeedhold ..... 35
MCV_GrpGetNextPrevAx ..... 36
MCV_GrpHFct ..... 51
MCV_GrpKinTrafoCmds ..... 38
MCV_GrpMFct ..... 53
MCV_GrpOperationMode ..... 39
MCV_GrpOptionalStop ..... 42
MCV_GrpPrgBlockIgnore ..... 43
MCV_GrpSetRapidOverride ..... 44
MCV_GrpSingleAxSelect ..... 45
MCV_GrpSingleBlock ..... 47
MCV_GrpTFct ..... 55
MCV_Handwheels ..... 29
MCV_ManMode ..... 25
MCV_ManModeKeys ..... 30
MCV_ReadActualPathInfo ..... 48




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