

Rexroth HydraulicDrive Rexroth IndraMotion MLD (2G) Libraries as of HDx-18

RE 30336-RE/2015-04

Replaces: 06.14

Reference Book



Title Rexroth HydraulicDrive
Rexroth IndraMotion MLD (2G)
Libraries as of HDx-18

Type of Documentation Reference Book

Document Typecode DOK-HYDRV*-MLD-SYSLIB3-RE01-DE-P

Internal File Reference RS-1ce05741d6fb40d20a6846a501cc2734-1-en-US-4

Purpose of Documentation This documentation describes the functions, function blocks and data types contained in the target system "HyDrive HDx-18", as well as a listing of the possible error reactions.

Record of Revision

Edition	Release Date	Notes
DOK-HYDRV*-MLD-SYSLIB3-RE01-DE-P	25.10.2013	See chapter "About This Documentation", marginal note "Editions of This Documentation"

Copyright © Bosch Rexroth AG 2013

This document, as well as the data, specifications and other information set forth in it, are the exclusive property of Bosch Rexroth AG. It may not be reproduced or given to third parties without its consent.

Liability The specified data is intended for product description purposes only and shall not be deemed to be a guaranteed characteristic unless expressly stipulated in the contract. All rights are reserved with respect to the content of this documentation and the availability of the product.

Published by Bosch Rexroth AG

Zum Eisengießer 1 ■ D-97816 Lohr a. Main

Telephone +49 (0) 93 52 / 18-0

<http://www.boschrexroth.com/>

E-mail: dokusupport@boschrexroth.de

Table of Contents

	Page
1 Introduction.....	5
1.1 About This Documentation.....	5
1.2 Reference Documentations.....	6
1.3 General Information.....	7
2 Libraries.....	9
2.1 Supported Libraries.....	9
2.2 Dependencies Between Parameterization of the Axis, Functional Package and Selected Function Block.....	10
3 Signal Time Diagrams of Most Frequently Used Inputs and Outputs.....	11
4 Error Handling.....	17
4.1 General Information.....	17
4.2 Error Reference Lists.....	18
5 Data Types.....	23
6 lecSfc.....	25
7 MY_Base.....	27
7.1 Introduction.....	27
8 MX_CheckRtv.....	29
9 MX_Debug.....	31
10 MX_PLCopen.....	33
10.1 Introduction.....	33
10.2 "Drive control" - Functions for Diagnostics.....	33
10.2.1 MX_fCheckCmdRequest.....	33
10.2.2 MX_fGetDriveWarning.....	33
10.2.3 MX_fSetDriveError.....	34
10.2.4 MX_fSetDriveErrorText.....	34
10.2.5 MX_fSetDriveWarning.....	35
10.2.6 MX_fSetDriveWarningText.....	35
10.2.7 MX_SetCmdState.....	36
10.3 "Motion" - Function Blocks/Functions for Drive Control.....	37
10.3.1 General Properties.....	37
10.3.2 MB_ChangeCamData.....	37
10.3.3 MB_ChangeProfileSet	37

Table of Contents

	Page
10.3.4 MB_ChangeProfileStep.....	37
10.3.5 MB_ClearAllError.....	38
10.3.6 MB_GearInPos.....	39
10.3.7 MB_Home.....	39
10.3.8 MB_MotionProfile.....	40
10.3.9 MB_Phasing.....	40
10.3.10 MB_PhasingSlave.....	40
10.3.11 MB_PreSetMode.....	40
10.3.12 MB_SetPositionControlMode.....	41
10.3.13 MB_Stop.....	43
10.3.14 MB_TorqueControl.....	44
10.3.15 MB_ValveControl.....	47
10.3.16 MC_CamIn.....	50
10.3.17 MC_CamOut.....	50
10.3.18 MC_GearIn.....	50
10.3.19 MC_GearOut.....	51
10.3.20 MC_Jog.....	51
10.3.21 MC_MoveAbsolute.....	53
10.3.22 MC_MoveAdditive.....	55
10.3.23 MC_MoveRelative.....	57
10.3.24 MC_MoveVelocity.....	59
10.3.25 MC_Power.....	61
10.3.26 MC_ReadStatus.....	63
10.3.27 MC_Reset.....	64
10.3.28 MC_Stop.....	65
10.3.29 MC_TorqueControl.....	67
10.3.30 MX_MoveAbsolute.....	68
10.3.31 MX_MoveAdditive.....	70
10.3.32 MX_MoveRelative.....	72
10.3.33 MX_SetControl.....	74
10.3.34 MX_SetDeviceMode.....	76
10.3.35 MX_SetOpMode.....	77
10.4 "Parameters" - Function Blocks/Functions for Parameters.....	79
10.4.1 MB_Command.....	79
10.4.2 MB_ReadListParameter.....	85
10.4.3 MB_ReadMaxRealValue.....	87
10.4.4 MB_ReadMaxValue.....	89
10.4.5 MB_ReadMinRealValue.....	90
10.4.6 MB_ReadMinValue.....	92
10.4.7 MB_ReadName.....	93
10.4.8 MB_ReadParameter.....	95
10.4.9 MB_ReadRealListParameter.....	97
10.4.10 MB_ReadRealParameter.....	99
10.4.11 MB_ReadSercosAttribute.....	101
10.4.12 MB_ReadSercosDataStatus.....	102
10.4.13 MB_ReadStringParameter.....	104

Table of Contents

	Page
10.4.14 MB_ReadUnit.....	106
10.4.15 MB_WriteListParameter	107
10.4.16 MB_WriteParameter.....	110
10.4.17 MB_WriteRealListParameter	112
10.4.18 MB_WriteRealParameter.....	114
10.4.19 MB_WriteStringParameter.....	116
10.4.20 MX_Command.....	118
10.4.21 MX_fReadParamDINT.....	121
10.4.22 MX_fReadStringParam.....	121
10.4.23 MX_fSetParamLimits.....	122
10.4.24 MX_fSetParamName.....	123
10.4.25 MX_fSetParamUnit.....	123
10.4.26 MX_fWriteParamDINT.....	124
10.4.27 MX_fWriteStringParam.....	125
10.4.28 MX_ReadParamDINT.....	126
10.4.29 MX_WriteParamDINT.....	127
10.5 "Scaling" - Functions for Scaling.....	129
10.5.1 MX_fDINT_AccTo_REAL.....	129
10.5.2 MX_fDINT_DistTo_REAL	129
10.5.3 MX_fDINT_VelTo_REAL	130
10.5.4 MX_fREAL_AccTo_DINT	130
10.5.5 MX_fREAL_DistTo_DINT	131
10.5.6 MX_fREAL_TorqueRampTo_DINT	131
10.5.7 MX_fREAL_TorqueTo_DINT	132
10.5.8 MX_fREAL_VelTo_DINT	132
10.6 "Tools" - General Functions.....	133
10.6.1 MX_fGetFreeTicks.....	133
10.6.2 MX_fGetHighResTime.....	133
10.6.3 MX_fHighResTimerTicks_to_us.....	133
10.6.4 MX_IECTaskGetLoad.....	134
10.6.5 MX_PositionLoopEventInfo.....	135
11 Basic Libraries.....	137
11.1 MX_CommonTypes.....	137
11.2 RIL_HMI_Utility.....	137
11.3 RIL_SocketComm.....	137
11.4 RIL_Utility.....	137
12 System Libraries.....	139
12.1 General Information.....	139
12.2 Standard.....	139
12.3 Util.....	139
12.4 SysDir.....	139
12.4.1 General Information.....	139
12.4.2 Notes On Utilization.....	139

Table of Contents

	Page
12.5 SysFile.....	140
12.5.1 General Information.....	140
12.5.2 Notes On Utilization.....	140
12.6 SyslecTasks23.....	140
12.7 SysMem.....	141
12.8 SysCallback23.....	141
12.9 SysSem.....	141
12.10 SysSockets.....	141
12.11 SysStr23	141
12.11.1 Overview.....	141
12.11.2 SysStrCmp	142
12.11.3 SysStrCmpl	142
12.11.4 SysStrCmpN	142
12.11.5 SysStrCmpNI	143
12.11.6 SysStrCpy	143
12.11.7 SysStrLen	143
 13 Free PLC Library by "OSCAT".....	 145
 Index.....	 147

1 Introduction

1.1 About This Documentation

Editions of This Documentation

Edition	Release date	Notes
DOK-HYDRV*-MLD-SYSLIB3-RE01-DE-P	25.10.2013	Contains descriptions of libraries for target "HydrDrive HDx-18"

Tab. 1-1: Record of Revisions

Means of Representation in This Documentation

To facilitate reading this documentation, the table below contains the means of representation and notations of recurring terms.

What?	How?	For example...
Paths and sequences of menus...	...are represented step by step in boldface	Click Window ► Show view ► Properties
Buttons and keys...	... are represented in angle brackets	Click <Add> or Press <Ctrl>+<Alt>
Important facts which are to be highlighted in the body text	Boldface	For remote axes , "Active"=TRUE signals that ...
Parameter names, diagnostic message names, function designations	Quotation marks	The function block "MC_Reset" is used to reset drive errors with the command "S-0-0099, C0500 Reset class 1 diagnostics"; drive enable remains unchanged.

Tab. 1-2: Conventions of Notation

All important notes are highlighted. A symbol tells you what kind of note is used in the text. The symbols have the following significances:

DANGER

...

DANGER indicates a hazardous situation which, if not avoided, **will** result in death or serious injury.

WARNING

...

WARNING indicates a hazardous situation which, if not avoided, **could** result in death or serious injury.

CAUTION

...

In case of non-compliance with this safety instruction, minor or moderate injury **could** occur.

NOTICE

...

In case of non-compliance with this safety instruction, property damage **could** occur.

Introduction



This box provides important information which should be observed.



This symbol highlights useful tips and tricks.

Documentation as Online Help

This documentation has been included in the HydraulicDrive help system.

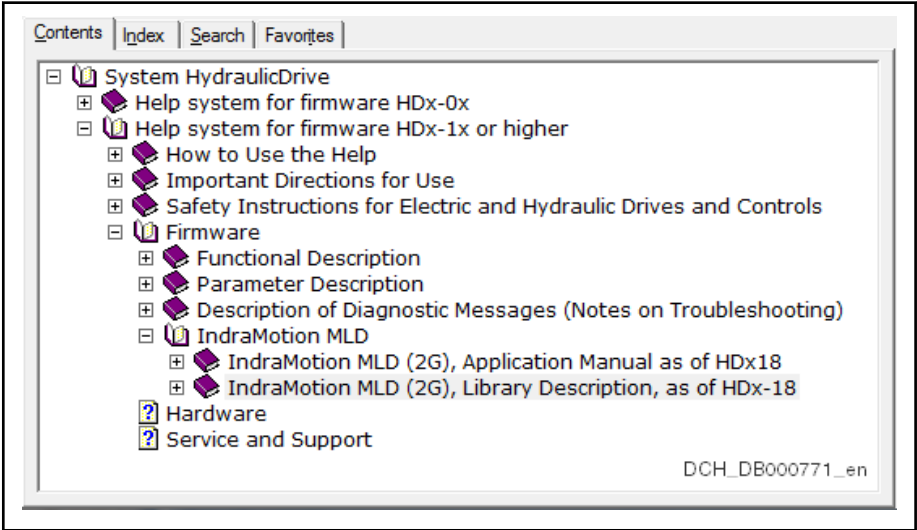


Fig. 1-1: Description of "Rexroth HydraulicDrive Firmware Libraries" in the HydraulicDrive Help System

1.2 Reference Documentations

Only deal with the product when you have the documentation identified with the book symbol  at hand and have observed it:

	Title	Document number	Document type
	High-response valve with integrated digital controller (IAC-Multi-Ethernet)	RE 29391	Technical data sheet Available on the Internet at www.boschrexroth.com
	High-response valve with integrated digital axis controller (IAC Multi-Ethernet)	RE 29391-B	Operating Instructions Available on the Internet at www.boschrexroth.com
	High-response valve with integrated digital axis controller (IAC Multi-Ethernet), pilot operated	RE 29288	Technical data sheet Available on the Internet at www.boschrexroth.com
	Rexroth HydraulicDrive HDS-16, HDB-17 and HDx-18	RE 30330-PA	Parameter description Available on the Internet at www.boschrexroth.com
	Rexroth HydraulicDrive HDS-16, HDB-17 and HDx-18	RE 30330-WA	Diagnostic Messages Available on the Internet at www.boschrexroth.com

	Title	Document number	Document type
	Rexroth HydraulicDrive HDS-16, HDB-17	RE 30330-FK	Functional Description Available on the Internet at www.boschrexroth.com
	Rexroth HydraulicDrive HDx-18	RE 30331-FK	Functional Description Available on the Internet at www.boschrexroth.com
	Digital axis control HMC	RE 30239	Technical data sheet Available on the Internet at www.boschrexroth.com
	Rexroth IndraMotion MLD (2G), Libraries as of HDx-18	RE 30336-RE	Library Description Available on the Internet at www.boschrexroth.com
	Rexroth IndraMotion MLD (2G), Libraries as of HDx-18	RE 30336-AP	Application Manual Available on the Internet at www.boschrexroth.com

Tab. 1-3: Additional Documentations

1.3 General Information

The PLC programming system IndraLogic 2G allows programming and commissioning PLC components of a control unit of the IndraLogic XLC / IndraMotion MLC, IndraMotion MLD or IndraMotion MTX types, or support them by means of an internally used visualization. IndraLogic supports the management of different libraries.

In addition, the functional expansion package "IndraMotion MLD" must have been enabled (see also Functional Description of firmware: "Enabling of Functional Packages").

When a project with an IndraDrive target or a HydraulicDrive target is created, some Rexroth libraries are automatically included in the project. The library manager allows subsequently adding libraries to a project to use their functions.

Libraries can contain both, function blocks and declarations and/or variables. The drive-integrated PLC (Rexroth IndraMotion MLD) is supplied together with basic libraries containing fundamental function blocks and definitions/variables.

As regards the function blocks, it is possible to create IEC functions/function blocks or define firmware function blocks according to the library type.

So-called "internal libraries" contain the IEC code.

2 Libraries

2.1 Supported Libraries

NOTICE

Property damage caused when non-supported libraries are included.

Only the libraries listed below are allowed to be included.

Library	Description
IecSfc	Makes available IEC steps conforming to standard in sequential language ["Sequential Function Chart" (SFC)]
MX_Base (IndraDrive) / MY_Base (HydraulicDrive)	• Data Types or Structures • Cyclic Parameters as Direct Variables (System-Wide Variables) • Axis Structures [for Multi-Axis System (MLD System Mode)]
MX_CanL2 (IndraDrive only)	Contains data types, functions and function blocks for CAN communication on level 2
MX_CheckRtv	For program-internal use only!: Functions for checking or signaling runtime errors
MX_Debug	For internal test purposes only (laboratory)!
MX_PLCOpen	• Functions for Diagnostics • Function Blocks/Functions for Drive Control • Function Blocks/Functions for Parameters • Functions for Scaling • General Functions
MX_CommonTypes	Contains data types and structures most of which are only used internally
RIL_Fieldbus	Makes available common data types (field bus types) for the Rexroth field bus libraries
RIL_HMI_Utility	Contains function blocks supporting the HMI devices
RIL_ModbusTCP	Makes available function blocks which enable the communication between Ethernet devices supporting the Modbus protocol
RIL_SercosIII (IndraDrive only)	Makes available functions which enable the communication between the PLC programming environment and the sercos III nodes
RIL_SocketComm	Functions and function blocks which allow using the TCP/UDP communication
RIL_Utility	Functions and function blocks for converting and influencing different data types
CmplecTask	System library containing, amongst others, interface functions used to switch off a watchdog (and switch it on again afterwards) (see MLD Application Manual: "Task Monitoring (Watchdog)")
SysDir / SysDir23 ^{*1}	Functions for synchronously accessing a file directory system on the target
SysFile / SysLibFile23 ^{*1}	Functions for synchronously accessing files
SysFileAsync / SysFileAsync23 ^{*1}	Functions for asynchronously accessing files
SyslecTasks23	Supports the "SysIETaskGetConfig" function by means of which the task configuration can be read
SysMem / SysMem23 ^{*1}	Functions for memory management
NetVarUdp	For program-internal use only!: Functions for processing network variables
Standard	Functions and function blocks that are required by IEC 61131-3 as standard blocks for an IEC programming system
SysCallback23 ^{*1}	Functions for activating defined callback functions for runtime events

Libraries

Library	Description
SysSem / SysSem23 ^{*1}	Functions for creating and using semaphores for task synchronization
SysSockets / SysSockets23 ^{*1}	Functions supporting the access to sockets for communication via TCP/IP and UDP
SysStr23 ^{*1}	Functions for handling strings
SysTime	Makes available functions allowing to read and set the real-time clock of the local system, as well as enabling various conversions of the time data. NOTE: To begin with, please check whether the "RIL_Uilities" library can fulfill your requirements.
Util	Function blocks that can be used for BCD conversion, bit/byte functions, mathematical auxiliary functions, as controllers, signal generators, function manipulators and for analog value processing
OSCAT	Free PLC library "OSCAT" ("Open Source Community for Automation Technology"). This open-source library contains many useful functions in the areas of "automation technology" and "building automation".

*1 Libraries with the extension "23" (such as in "SysSockets23") are used to port projects; for new projects, use the libraries without the extension "23"



"*.library" is the standard format for libraries.

The codes of the libraries supplied via IndraWorks are protected, they have the "*.compiled-library-ge33" format.

2.2 Dependencies Between Parameterization of the Axis, Functional Package and Selected Function Block

Depending on the functional package and the parameterization of the axis, not all function blocks are operable. The table below shows the dependency between the parameterization of the axis, the functional package and the selected operation mode.

P-0-2003 (1)	Drive-controlled positioning (MX_MoveRelative / MC_MoveRelative, MX_MoveAbsolute / MC_MoveAbsolute, MX_MoveAdditive / MC_MoveAdditive, MC_Jog)	Closed-loop torque/force control (MC_TorqueControl) (MB_TorqueControl)	Direct valve control (MB_ValveControl)	Closed-loop velocity control (MC_MoveVelocity)
0x00080201 (axis controller + OpenLoop)	-	+	+	-
0x00080202 (axis controller + ClosedLoop)	+	+	+	+
0x00080212 (axis controller + ClosedLoop + Servo)	+	+	+	+

- Function block is **not** operable

+ Function block is operable

1 P-0-2003, Selection of functional packages

Tab.2-1: Dependencies Between Parameterization of the Axis, Functional Package and Selected Function Block

Signal Time Diagrams of Most Frequently Used Inputs and Outputs

3 Signal Time Diagrams of Most Frequently Used Inputs and Outputs

Function blocks according to PLCopen use input and output signals with fixed functions.

Input Signals "Execute" / "Enable"

To trigger the processing of a function block or to activate it, a function block has an "Execute" input, an "ExecuteLock" input or an "Enable" input. All three inputs are of the BOOL type.

The "Execute" activation input is used at function blocks which work in edge-controlled form. With a positive edge of "Execute", the input variables are collected and at the same time the command triggering of a one-time execution of the function block takes place. The outputs "Done" or "InVelocity", "InTorque", "In_xxxx" depend on the input. New input values only take effect via the repeated positive edge of "Execute". When command triggering has not yet been completed and is interrupted, this is signaled by the fact that the "CommandAborted" output is set.



Retriggering is possible for edge-controlled function blocks with "Execute" input; this means that when the activation input, with active function block, gets a repeated edge, the old task is rejected, the inputs are applied again and the new task is continued with the new values.

The "ExecuteLock" activation input behaves like the "Execute" input. But as long as this input is TRUE, the function block cannot be interrupted. Exception: The function block "MC_Stop" can interrupt the function block "MB_Stop".

The "Enable" activation input is used at function blocks which work in status-controlled form. The input variables are applied with the positive edge of "Enable". New input values only take effect via the repeated positive edge of "Enable". If the input variables are cyclically transmitted (e.g. with process controller function blocks), this has to be explicitly documented and marked at the function block input. The function of the function block is deactivated with the low signal of the "enable".

Output Signal "Active"

For remote axes, "Active"=TRUE signals that the function block has accepted the processing command (positive edge at "Execute", "ExecuteLock" or "Enable").



The "Active" signal can be delayed at the start of the function block, as possibly required pre-processing has not been completed.

"Active" becomes FALSE, when "Done", "CommandAborted" or "Error" become TRUE. For status-controlled function blocks, "Active" becomes FALSE when "Enable" is reset.



In the case of **parameter function blocks**, "Active" is not set for a **single axis!**

Output Signal "Done"

Function blocks which have an administrative task ("MC_Power", ...) or motion function blocks (such as "MC_MoveAbsolute", "MC_MoveRelative", ...) which end in the "standstill" state, signal via the "Done" output that their task has been completed and are no longer active after they have been executed; in other words: The "Done" output signals

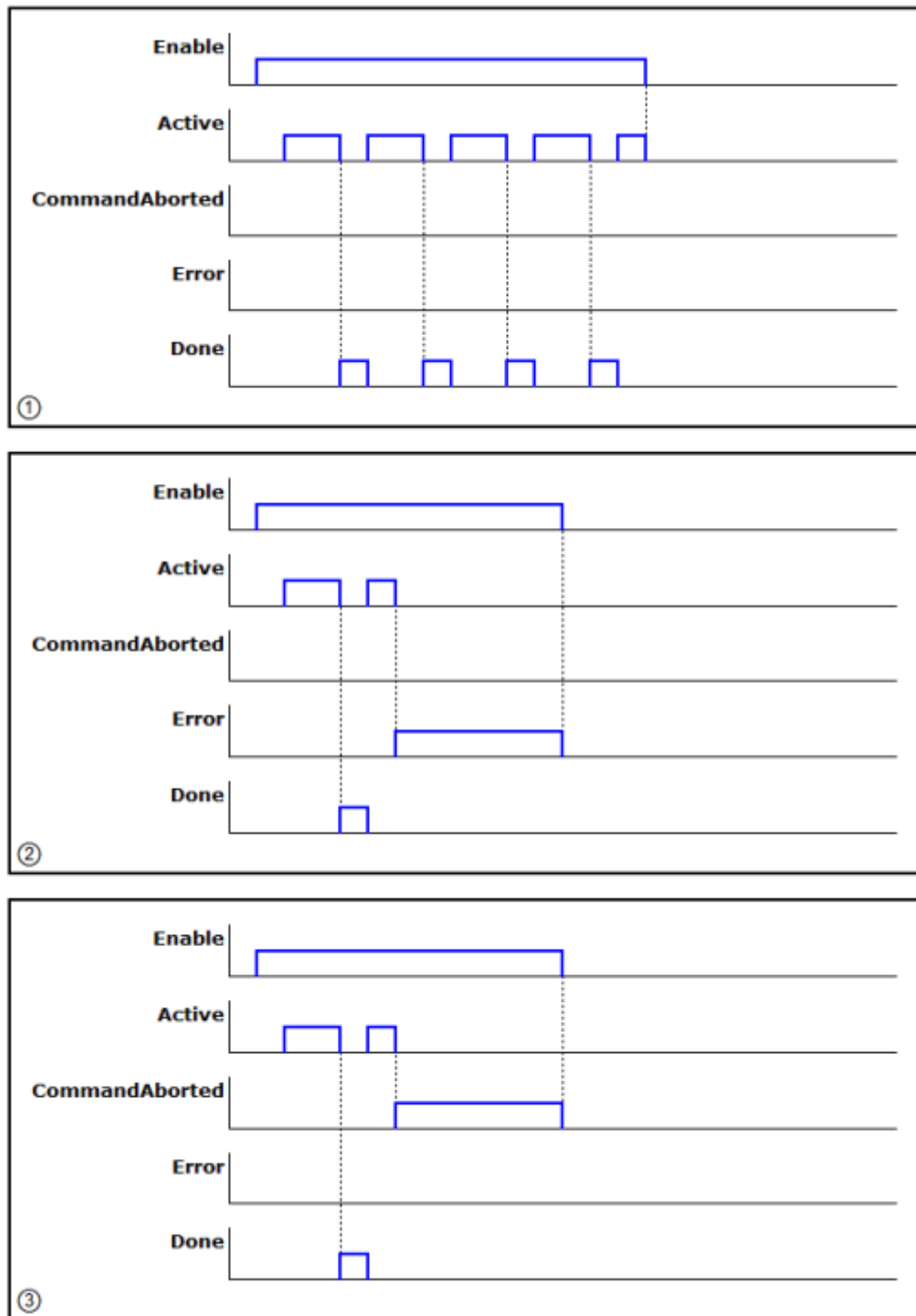
Signal Time Diagrams of Most Frequently Used Inputs and Outputs

that a function block has successfully completed its task and data possibly provided are valid.

For edge-controlled function blocks, "Done" remains TRUE for one cycle when the "Execute" input had been FALSE at the point of time at which "Done" was set to TRUE. If the input "Execute"=TRUE when "Done" is set, "Done" remains TRUE until "Execute" is set to FALSE.

For status-controlled function blocks, when "Enable"=TRUE, "Done" after successful processing remains TRUE for exactly one cycle before processing starts again.

Signal Time Diagrams of Most Frequently Used Inputs and Outputs



①

②

③

Fig. 3-1:

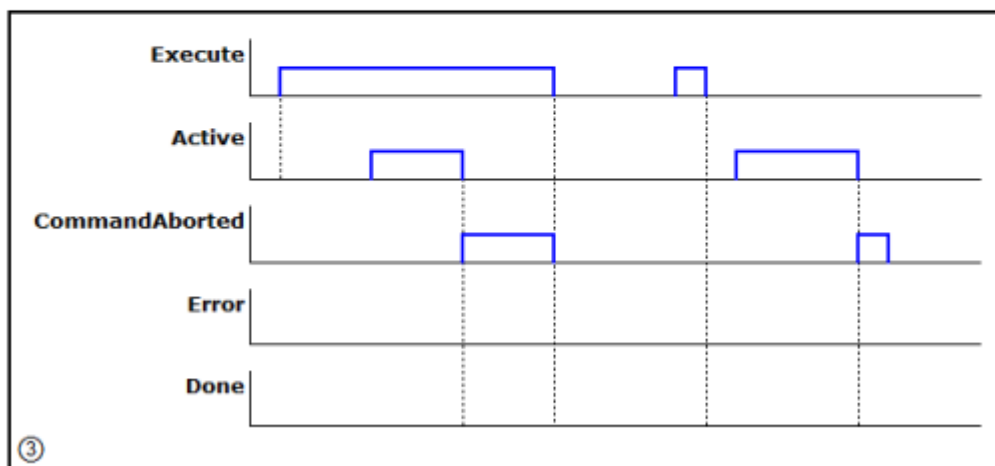
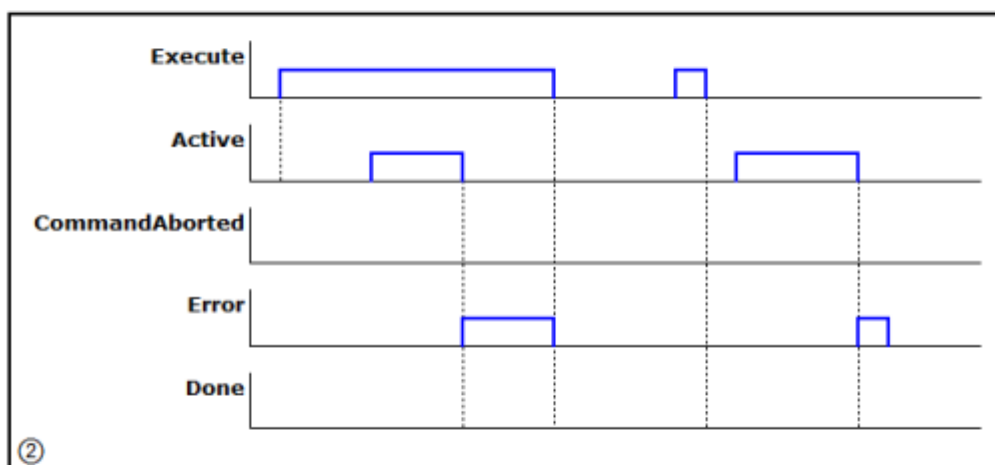
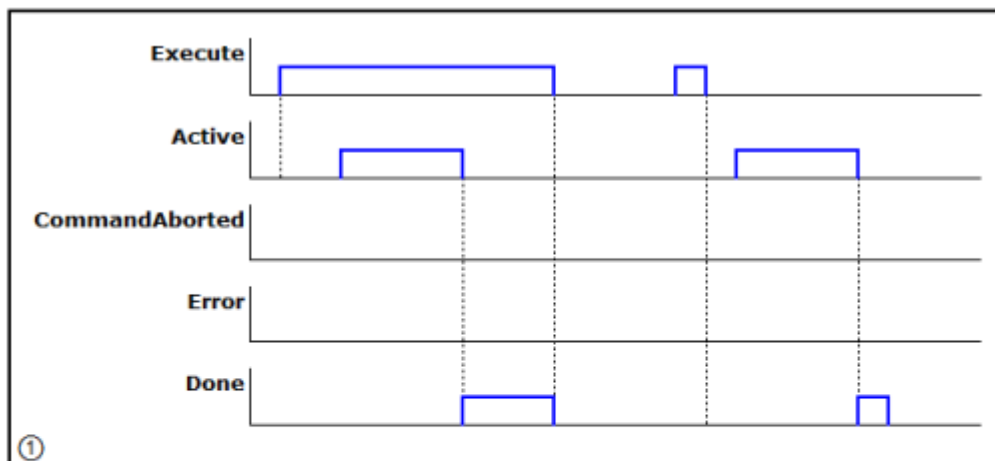
Status-controlled function block processing successfully completed

Status-controlled function block processing aborted with error

Status-controlled function block processing interrupted

Signal-Time Behavior of Status-Controlled Function Blocks (with "Enable" Input); in the Case of Parameter Function Blocks, "Active" is not Set for a Single Axis!

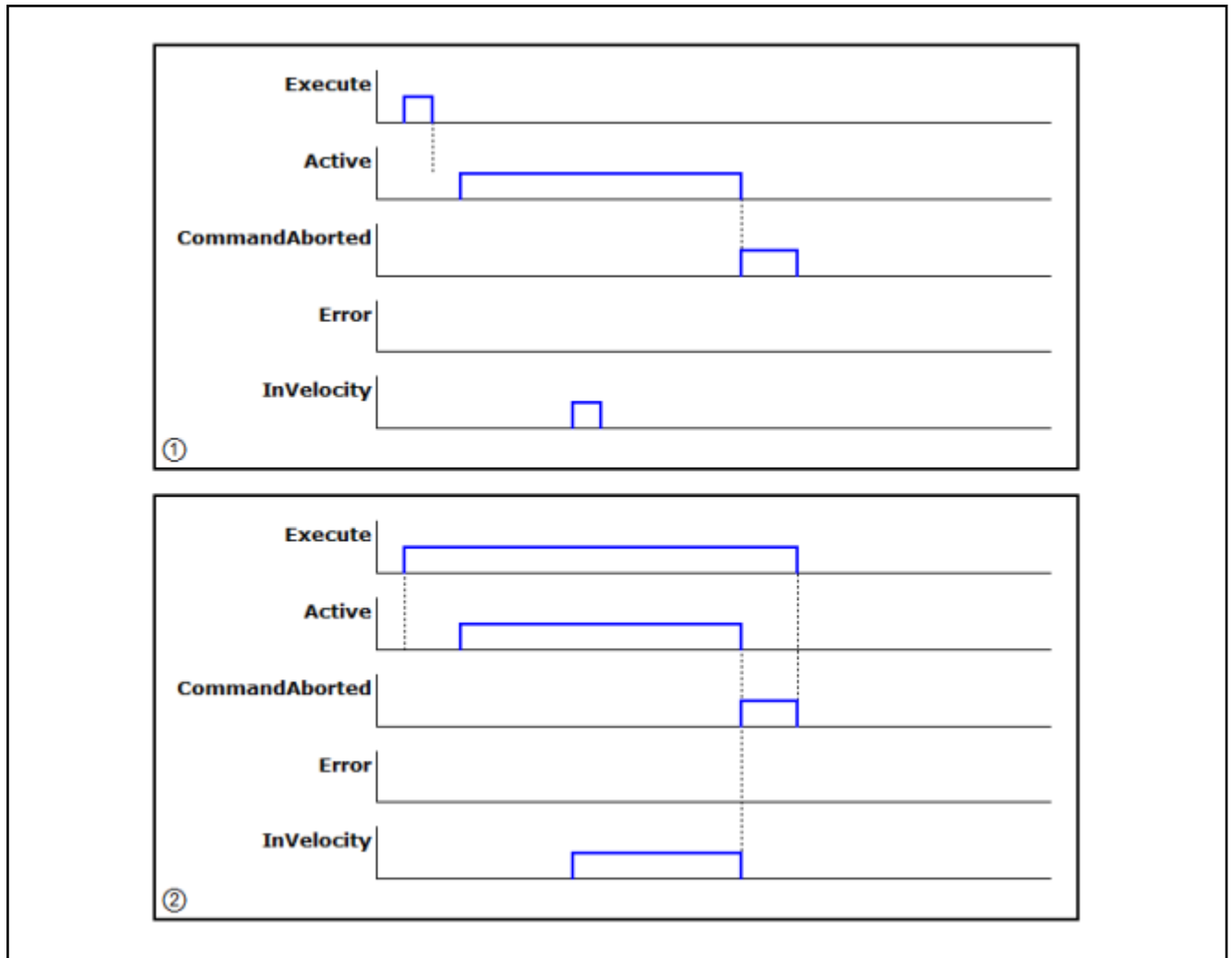
Signal Time Diagrams of Most Frequently Used Inputs and Outputs



- ① Edge-controlled function block processing successfully completed
 ② Edge-controlled function block processing aborted with error
 ③ Edge-controlled function block processing interrupted
- Fig.3-2: Signal-Time Behavior of Edge-Controlled Function Blocks (with "Execute" Input); in the Case of Parameter Function Blocks, "Active" is not Set for a Single Axis!*

Signal Time Diagrams of Most Frequently Used Inputs and Outputs

There are function blocks (such as "MC_MoveVelocity", "MC_TorqueControl", "MB_MotionProfile") which continue their motion or force activity after their task has been fulfilled and which are aborted only by "MC_Stop" or another function block being called. They signal this state by remaining "Active"; i.e. "Active" and, for example, "InVelocity", etc. are TRUE at the same time.



①+②

Fig. 3-3:

Edge-controlled function block processing interrupted
Signal-Time Behavior of Edge-Controlled Function Blocks (with
"Execute" Input); "Active" in Combination with "InVelocity"

Output Signal "CommandAborted"

The "CommandAborted" output signals that the function block was unable to complete its task, because it had been interrupted by another motion function block or the axis had been switched off due to an error.

For edge-controlled function blocks, the "CommandAborted" output remains TRUE for one cycle when the "Execute" input had been FALSE at the point of time at which "CommandAborted" was set to TRUE. If the input "Execute"=TRUE when "CommandAborted" is set, "CommandAborted" remains TRUE until "Execute" is set to FALSE.

For status-controlled function blocks, the "CommandAborted" output is reset when the "Enable" input is reset.

4 Error Handling

4.1 General Information

Depending on their target the function blocks have two or three outputs which contain information in case an error occurs:

- "Error",
- "ErrorID" and
- "ErrorIdent"

Error If processing of a task could not be successfully completed, this is signaled by the "Error" output. The "Error" output is reset with the inputs "Execute"=FALSE or "Enable"=FALSE; an additional reset input therefore is not required.

For edge-controlled function blocks, the "Error" output remains TRUE for one cycle when the "Execute" input had been FALSE at the point of time at which the "Error" output was set to TRUE. If the input "Execute"=TRUE when "Error" is set, the "Error" output remains TRUE until the "Execute" input is set to FALSE.

For status-controlled function blocks, the "Error" output remains TRUE until the "Enable" input is set to FALSE.

ErrorID The "ErrorID" output is of the type ERROR_CODE. When an error has occurred at the function block, the error classification can be seen at output "ErrorID".



The output values of "ErrorID" must be interpreted differently according to the respective target!

The ERROR_CODE type has been defined in the "CommonTypes" library of the respective target.

ErrorIdent The "ErrorIdent" output signals the exact cause of the error. "ErrorIdent" is of the type ERROR_STRUCT. ERROR_STRUCT contains the following elements:

- "Table",
- "Additional1" and
- "Additional2".



The structure and its elements have been defined in the "CommonTypes" library of the target.

Element	Element data type	Description
Table	ERROR_TABLE	The value of the element "Table" contains the information indicating which error table [INDRV_TABLE, SERCOS_TABLE or MLD_TABLE] has to be used for decoding the error code of the element "Additional1"
Additional1	DWORD	Element "Additional1" contains the error code for exact specification of the occurred error
Additional2	DWORD	"Additional2" contains additional information (if available)

Tab.4-1: Elements of "ERROR_STRUCT"

Error Handling

4.2 Error Reference Lists



When debugging a PLC program, you can select via **Debug ► Display Mode** in IndraWorks whether binary data types (BYTE, WORD, DWORD) are to be displayed in a decimal, hexadecimal or binary form. The activated selection is identified by a check mark.

MLD_TABLE At the "ErrorIdent" output, the "Table" element refers to the error table "MLD_TABLE".



When there is not any error present, the element "Table" has the value "NO_TABLE_USED".

"ErrorID"	"ErrorIdent", Element "Additional1" (hexadecimal)	"ErrorIdent", Element "Additional2"	Description
-	16#00000000	Transmitted axis number	When there is not any error present, the element "Table" has the value "NO_TABLE_USED"
INPUT_INVALID_ERROR	16#00000001	Transmitted axis number	Selected function is not available for this axis
RESOURCE_ERROR	16#00000001	Transmitted axis number	Invalid axis reference / undefined axis. Indicated axis (also slave axis) is not available
ACCESS_ERROR	16#00000002	Transmitted axis number	The function block used may only have one instance in the system (e.g.: "MX_SynchronControl")
STATE_MACHINE_ERROR	16#00000003	Transmitted axis number	Drive is not in operating status "Ab" / communication phase 4 - or - With master axis available, this axis has not been activated (P-0-0917, Control word of master axis generator)
STATE_MACHINE_ERROR	16#00000004	Transmitted axis number	Drive is not in operating status "Ab"
ACCESS_ERROR	16#00000005	Transmitted axis number	MLD has neither permanent (P-0-1367) nor temporary control over the indicated (local) axis
ACCESS_ERROR	16#00000006	Transmitted axis number	Motion command or "MC_Stop": Drive is torque-free / not under torque
STATE_MACHINE_ERROR	16#00000006	Transmitted axis number	"MC_Power": Drive is not ready for operation. Drive was switched off externally
RESOURCE_ERROR	16#00000007	Transmitted axis number	Function block not possible with current functional package / current control type (see table " Libraries "). - or - A parameter required by a motion function block has already been cyclically configured and therefore is write-protected.
STATE_MACHINE_ERROR	16#00000011	Transmitted axis number	After the activation with "MB_PreSetMode" there was an attempt to switch on with "MC_Power" without motion command triggering
STATE_MACHINE_ERROR	16#00000012	Transmitted axis number	An instance of "MB_Stop" tries to command an axis while an instance of "MC_Stop" is active - or - The axis is in "Stopping" and must not be commanded ("MB_Stop"/"MC_Stop" with active "Execute" input)
INPUT_RANGE_ERROR	16#00000013	Transmitted axis number	The "CamTableID" input is not within the valid range (1 .. 8)
STATE_MACHINE_ERROR	16#00000014	Transmitted axis number	MLD is in "ErrorStop" state. Remedy: Execute "MC_Reset"

Error Handling

"ErrorID"	"ErrorIdent", Element "Additional1" (hexadecimal)	"ErrorIdent", Element "Additional2"	Description
STATE_MACHINE_ERROR	16#00000015	Transmitted axis number	Activation of "MC_Power". The enable signal "DriveEnable" in the axis control word (P-0-0116, bit14) is missing
ACCESS_ERROR	16#00000016	Transmitted axis number	MLD in slave axis has permanent control. Therefore, the MLD-M master cannot control the slave
INPUT_RANGE_ERROR	16#00000017	Transmitted axis number	Parameterized acceleration is ≤ 0
INPUT_RANGE_ERROR	16#00000018	Transmitted axis number	Parameterized deceleration is ≤ 0
INPUT_RANGE_ERROR	16#00000019	Transmitted axis number	Parameterized velocity is < 0
ACCESS_ERROR	16#00000020	Index of the second task	Several tasks use the same instance of a function block, this is not allowed
STATE_MACHINE_ERROR	16#00000021	Transmitted axis number	An additive motion command was started which is not allowed in this axis status
ACCESS_ERROR	16#00000023	Transmitted axis number	Motion command could not be processed due to buffer overflow
OTHER_ERROR	16#00000199	Transmitted axis number	Internal processing error (undefined status)
ACCESS_ERROR	16#00000200	Parameter IDN	Internal error: Undefined parameter data format, parameter cannot be converted into FLOAT
ACCESS_ERROR	16#00000201	Parameter IDN	With this function block it is impossible to read/write an ASCII parameter
ACCESS_ERROR	16#00000202	Parameter IDN	With this function block it is impossible to read/write a list parameter
INPUT_INVALID_ERROR	16#00000203	Parameter IDN	Parameter IDN does not belong to any command
OTHER_ERROR	16#00000204	Transmitted axis number	Internal processing error (string too short)
ACCESS_ERROR	16#00000205	Parameter IDN	The required number of bytes is smaller than the actual length of the list
INPUT_INVALID_ERROR	16#00000206	Transmitted axis number / transmitted sercos address	Invalid pointer at the input of the function block [address of the applied array is outside of the PLC range or is in the memory range of a function (use global array)]. Only for library "RIL_SercosIII.lib": "Additional2" contains the transmitted sercos address
ACCESS_ERROR	16#00000207	Parameter IDN	With this function block it is only possible to process ASCII parameters
CCD_ERROR	16#00000208	16#00000200	Too many simultaneous read/write tasks on remote axes
SYSTEM_ERROR	16#00000208	CCD error number	Internal CCD system error; for error number, see "Additional2"
SYSTEM_ERROR	16#00000209	CoSeMa error number	Internal CoSeMa system error; for error number, see "Additional2"
ACCESS_ERROR	16#00000210	99	Reset command S-0-0099 incorrectly executed
ACCESS_ERROR	16#00000211	Transmitted axis number	"MC_Reset" cannot reset the error (timeout)
RESOURCE_ERROR	16#00000220	Transmitted axis number	No external encoder available
RESOURCE_ERROR	16#00000250	Transmitted axis number	Master communication synchronizes the drive
RESOURCE_ERROR	16#00000251	Transmitted axis number	NC cycle time (S-0-0001) does not match MLD task cycle time
INPUT_RANGE_ERROR	16#00000252	Transmitted axis number	The "CmdDelay" input is greater than the maximum value

Error Handling

"ErrorID"	"ErrorIdent", Element "Additional1" (hexadecimal)	"ErrorIdent", Element "Additional2"	Description
RESOURCE_ERROR	16#00000253	Transmitted axis number	The real-time channel may use only one MLD task
RESOURCE_ERROR	16#00000254	Transmitted axis number	Too many real-time variables used. At present, there are 4 RTcW_ variables and 4 RTcR_ variables allowed
CALCULATION_ERROR	16#00000255	Transmitted axis number	Internal error (variable offset too great)
CALCULATION_ERROR	16#00000256	Transmitted axis number	Internal error (incorrect variable access)
RESOURCE_ERROR	16#00000257	Transmitted axis number	Real-time channel actual values but no command values defined
ACCESS_ERROR	16#00000258	Transmitted axis number	Synchronous motion task (CCD / master communication) of the type "triggered by external event" does not support local real-time channel
ACCESS_ERROR	16#00000259	Transmitted axis number	The local real-time channel has not been configured for the task
ACCESS_ERROR	16#00000300	Transmitted axis number	MLD already has permanent control. Additional switching to temporary control does not make sense
ACCESS_ERROR	16#00000301	Transmitted axis number	Temporary control is impossible with virtual slave
ACCESS_ERROR	16#00000302	Transmitted axis number	Manual mode active, temporary control impossible
STATE_MACHINE_ERROR	16#00000320	Transmitted axis number	Master communication internally synchronizes the drive / does not allow phase switch (P-0-4086, bit 0=0)
STATE_MACHINE_ERROR	16#00000321	Transmitted axis number	Drive initialization not yet completed (incl. reaching of target phase) In the case of master communication with analog or parallel interface, it is necessary to wait until the automatic phase switch to "P4" is over. When a transition command error occurs during this first phase switch, you have to wait for 10 s; only then is it allowed to input the phase with the function block "MX_SetDeviceMode"
ACCESS_ERROR	16#00000322	Transmitted axis number	A transition command was started or cleared from a different device
DEVICE_ERROR	16#00000350	Transmitted axis number	Unknown secondary operation mode ("NewOpMode">7 was set)
RESOURCE_ERROR	16#00000355	Transmitted axis number	Axis applied at the "Master" input is not a valid master axis. Only those axes are supported as master axes that are either contained in the master axis configuration list (P-0-1820.0.1) or that are locally available in the slave axis (VmAxisInt, RmAxisInt and LinkAxisInt).
INPUT_RANGE_ERROR	16#00000370	Incorrect "CmdState"	Invalid "CmdState" input (wrong enum value)
INPUT_RANGE_ERROR	16#00000371	Incorrect "CmdState"	"CmdState" not allowed, e.g. "CmdState"="MX_CMD_NO_ACTIVE" while "P-0-1449, C4900 PLC command"="3"
INPUT_RANGE_ERROR	16#00000380	Incorrect diagnostic message number	Invalid input of diagnostic message number (N<1 or N>4)
ACCESS_ERROR	16#00000400	Parameter IDN	The indicated parameter cannot be read / written with this function block. For example, parameter of data type HEX or BIN with "MB_WriteRealParameter"
ACCESS_ERROR	16#00000401	Parameter IDN	With this function block it is impossible to read/write a FLOAT parameter. PLC registers (e.g. P-0-1370) can be configured as FLOAT.

Error Handling

"ErrorID"	"ErrorIdent", Element "Additional1" (hexadecimal)	"ErrorIdent", Element "Additional2"	Description
INPUT_RANGE_ERROR	16#00000402	Transmitted axis number / transmitted sercos address	The applied parameter value does not fit in the data format of the parameter Only for library "RIL_SercosIII.lib": "Additional2" contains the transmitted sercos address
INPUT_INVALID_ERROR	16#00000403	Transmitted axis number	For the function block "MB_ReadListParameter" or "MB_WriteListParameter", the address applied at the "ValueAdr" input is not WORD-compatible (2Byte); i.e. the address cannot be divided by 2.
INPUT_INVALID_ERROR	16#00000404	Transmitted axis number	For the function block "MB_ReadListParameter" or "MB_WriteListParameter", the address applied at the "ValueAdr" input is not DWORD-compatible (4Byte); i.e. the address cannot be divided by 4.
ACCESS_ERROR	16#00000420	Transmitted axis number	A deactivated axis was addressed.
ACCESS_ERROR	16#00000421	Transmitted axis number	The service channel is not available, because a phase switch (PM→OM) is taking place just now.
ACCESS_ERROR	16#00000422	Transmitted axis number	The service channel is no longer available (phase < 2).
ACCESS_ERROR	16#00000423	Transmitted axis number	Double address assignment. At least 2 axes have the same sercos address.
SYSTEM_ERROR	16#00000424	Transmitted axis number	Queue error. Too many tasks were transmitted to an axis.
INPUT_RANGE_ERROR	16#00000500	Transmitted port	A value for "Port"<>"COM1" was transmitted
ACCESS_ERROR	16#00000501	Transmitted port	COM1 has already been opened

Tab.4-2: Error Table MLD_TABLE

SERCOS_TABLE

Error code		
Hexadecimal	Decimal	Explanation
0x1001	4097	No IDN
0x1009	4105	Invalid access to element 1
0x2001	8193	No name
0x2002	8194	Name transmission too short
0x2003	8195	Name transmission too long
0x2004	8196	Name cannot be changed (read only)
0x2005	8197	Name is write-protected at this time
0x3002	12290	Attribute transmission too short
0x3003	12291	Attribute transmission too long
0x3004	12292	Attribute cannot be changed (read only)
0x3005	12293	Attribute is write-protected at this time
0x4001	16385	No units
0x4002	16386	Unit transmission too short
0x4003	16387	Unit transmission too long

Error Handling

Error code		
Hexadecimal	Decimal	Explanation
0x4004	16388	Unit cannot be changed (read only)
0x4005	16389	Unit is write-protected at this time
0x5001	20481	No minimum input value
0x5002	20482	Minimum input value transmission too short
0x5003	20483	Minimum input value transmission too long
0x5004	20484	Minimum input value cannot be changed (read only)
0x5005	20485	Minimum input value is write-protected at this time
0x6001	24577	No maximum input value
0x6002	24578	Maximum input value transmission too short
0x6003	24579	Maximum input value transmission too long
0x6004	24580	Maximum input value cannot be changed (read only)
0x6005	24581	Maximum input value is write-protected at this time
0x7002	28674	Operation data transmission too short
0x7003	28675	Operation data transmission too long
0x7004	28676	Operation data cannot be changed (read only)
0x7005	28677	Operation data is write-protected at this time (reason: communication phase or mode)
0x7006	28678	Operation data is smaller than the minimum input value
0x7007	28679	Operation data is greater than the maximum input value
0x7008	28680	Invalid operation data (e.g. IDN not supported, invalid bit number, invalid bit combination, invalid list length)
0x7009	28681	Operation data write protected by a password
0x700A	28682	Operation data is write protected, it is configured cyclically (IDN is configured in the MDT or AT. Therefore writing via the service channel is not allowed).
0x700B	28683	Invalid indirect addressing (e.g., data container, list handling)
0x700C	28684	Operation data is write protected, due to other settings (e.g., parameter, operation mode, drive enable, drive on etc.)
0x7010	28688	Procedure command already active
0x7011	28689	Procedure command not interruptible
0x7012	28690	Procedure command at this time not executable (e.g., in this phase the procedure command cannot be activated)
0x7013	28691	Procedure command not executable (invalid or false parameters)

Tab.4-3: *sercos Error Codes*

INDRV_TABLE



"INDRV_TABLE" is no real table in the classical sense, but serves as reference to the documentation "Troubleshooting Guide". The error number specified (diagnostic message number) provides information about the cause of the error and its remedy.

5 Data Types

Data type	Bit size	Description
BOOL	With indication of address (e.g. "AT %MX1.2"): 1	"BOOL" can have the values "TRUE" or "FALSE"; without indication of address, 8 bits are used
	Without indication of address: 8	
BYTE	8	Byte value 0 to 255
WORD	16	Word value 0 to 65535
DWORD	32	Double word values 0 to $(2^{32})-1$
LWORD	64	Long word values 0 to $(2^{64})-1$
INT	16	Integer (16 bits) from -32768 to +32767
DINT	32	Integer (32 bits) from -2^{31} to $(2^{31})-1$
SINT	8	Integer (8 bits) from -128 to +128
USINT	8	Integer (8 bits) from 0 to +255
UINT	16	Integer (16 bits) from 0 to +65535
UDINT	32	Integer (32 bits) from 0 to +4294967295
LINT	64	Integer (64 bits) from -2^{63} to $+2^{63}-1$
ULINT	64	Integer (64 bits) from 0 to $+2^{64}-1$
REAL	32	IEEE floating point number
LREAL	64	IEEE floating point number
TIME	32	IEC time in intervals of 1 ms
LTIME	64	Like "TIME". In addition, "µs" (microseconds) and "ns" (nanoseconds) can be used
DATE	32	IEC date in intervals of 1 day
DATE_AND_TIME, DT	32	Time in steps of 1s: year-month-day-hour:minute:second
TIME_OF_DAY, TOD	32	Time in steps of 1ms: hour (0-23), min. (0-59), second (0-59), ms (0-999)

Tab.5-1: General Data Types

6 IecSfc

The "IecSfc.library" library is automatically included in PLC projects. The library makes available the "SFCActionControl" function.

To configure the way in which an action assigned to an IEC step is to be performed, different qualifiers can be used. The qualifiers are processed by the "SFCActionControl" function.

See also documentation "IndraLogic 2G, PLC Programming System", chapter "Qualifiers for Actions in SFC".

7 MY_Base

7.1 Introduction

The "MY_Base.library" library makes available the following items:

- Data Types or Structures
- Cyclic Parameters as Direct Variables (System-Wide Variables)
- Axis Structures [for Multi-Axis System (MLD System Mode)]

8 MX_CheckRtv



The functions of the "MX_CheckRtv" library are implicitly called by the PLC program and must not be included in a user program!

The "MX_CheckRtv" library contains functions for checking or signaling runtime errors, such as division by zero.

When a new project with the "IndraDrive" or "HydraulicDrive" target is created, the corresponding library is automatically included and visible.

Operating Principle

The functions of the "CheckRtv" library are called with each corresponding action (e.g. division) and check the access. If access is allowed, the program runs without restriction. If access is not allowed, a drive error (F6010) is generated and the PLC is stopped.

See also "Notes on Application" in the Application Manual on Rexroth IndraMotion MLD.

Functions

- **CheckBounds**
- **CheckDivReal**
- **CheckDivLReal**
- **CheckPointer**
- **CheckRangeSigned**
- **CheckRangeUnsigned**

9 MX_Debug

The functions contained in the "MX_Debug" library are required for internal testing purposes (laboratory).

** WARNING**

Property damage, death or serious injury by incorrect control of drives when using functions from the "MX_Debug" library!

The functions from the "MX_Debug" library are required for internal testing purposes and must not be utilized by the user!

10 MX_PLCOpen

10.1 Introduction

With the "MX_PLCOpen.library" library, the following function blocks/functions are made available:

- Functions for Diagnostics
- Function Blocks/Functions for Drive Control
- Function Blocks/Functions for Parameters
- Functions for Scaling
- General Functions

10.2 "Drive control" - Functions for Diagnostics

10.2.1 MX_fCheckCmdRequest

The firmware function "MX_fCheckCmdRequest" is used to return the command input which was preset via the drive parameter "P-0-1449, C4900 PLC command".

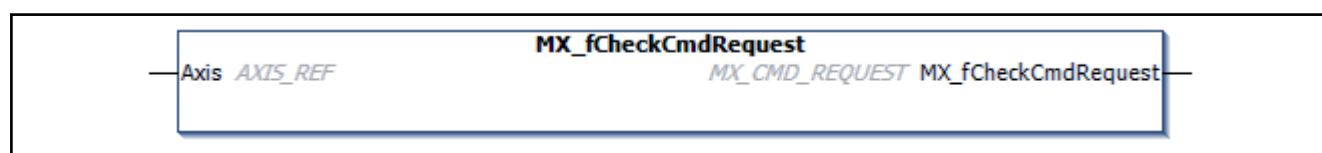


Fig. 10-1: Firmware Function "MX_fCheckCmdRequest"

Input variable	Data type	Description
Axis	AXIS_REF	Axis number (only the global variable "Axis1" can be transmitted here)

Tab. 10-1: Input Variables of the Firmware Function "MX_fCheckCmdRequest"

Return value	Data type	Description
MX_fCheckCmdRequest	MX_CMD_REQUEST	The function "MX_fCheckCmdRequest" can return the following values: • MX_CMD_REQ_ERROR: An invalid axis number was transmitted • MX_CMD_REQ_STOP: Command is to be terminated • MX_CMD_REQ_BREAK: Command is to be interrupted • MX_CMD_REQ_START: Command is to be started

Tab. 10-2: Return Value of the Firmware Function "MX_fCheckCmdRequest"

10.2.2 MX_fGetDriveWarning

The firmware function "MX_fGetDriveWarning" allows polling whether a certain warning diagnosis is active in the drive.

Depending on the value at the "WarningNo" input (1 to 7), the warning messages E2011 ... E2017 are checked.

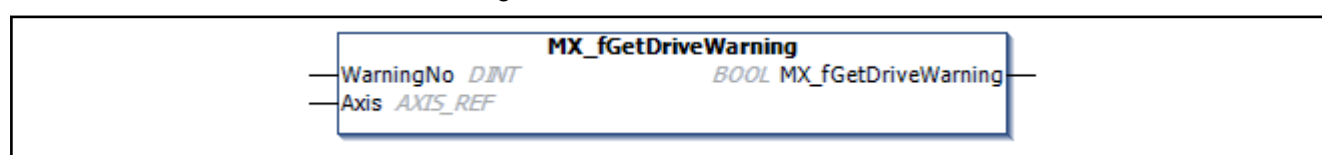


Fig. 10-2: Firmware Function "MX_fGetDriveWarning"

MX_PLCOpen

Input variable	Data type	Description
WarningNo	DINT	Order number of warning to be read
Axis	AXIS_REF	Axis number (only the global variable "Axis1" can be transmitted here)

Tab. 10-3: Input Variables of the Firmware Function "MX_fGetDriveWarning"

Return value	Data type	Description
MX_fGetDriveWarning	BOOL	If the warning to be checked is active in the drive, the function provides TRUE as return value, otherwise FALSE

Tab. 10-4: Return Value of the Firmware Function "MX_fGetDriveWarning"

10.2.3 MX_fSetDriveError

By means of the firmware function "MX_fSetDriveError", an error diagnosis can be triggered in the drive.

Depending on the value at the "ErrorNo" input (1 to 7), the errors F2011 ... F2017 are triggered in the drive.

The error diagnosis can be cleared by executing the function block "MC_Reset".

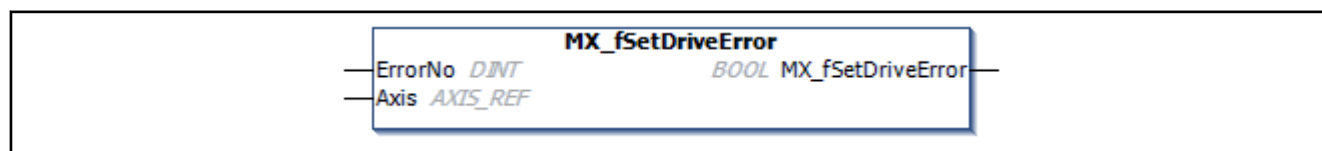


Fig. 10-3: Firmware Function "MX_fSetDriveError"

Input variable	Data type	Description
ErrorNo	DINT	Order number of error to be triggered
Axis	AXIS_REF	Axis number (only the global variable "Axis1" can be transmitted here)

Tab. 10-5: Input Variables of the Firmware Function "MX_fSetDriveError"

Return value	Data type	Description
MX_fSetDriveError	BOOL	If the value of "ErrorNo" is within the allowed range (1 to 7), the function provides TRUE as return value, otherwise FALSE

Tab. 10-6: Return Value of the Firmware Function "MX_fSetDriveError"

10.2.4 MX_fSetDriveErrorText

With the firmware function "MX_fSetDriveErrorText", the diagnostic text of the PLC errors F2011 to F2017 can be changed.

By switching the drive off and on again, the PLC errors are initialized again with their standard diagnostic texts (e.g. "PLC - Error no. 1").

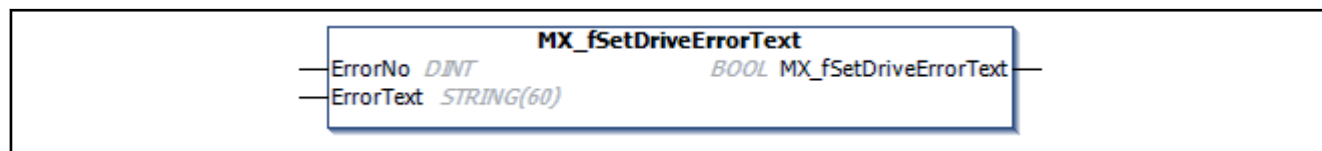


Fig. 10-4: Firmware Function "MX_fSetDriveErrorText"

Input variable	Data type	Description
ErrorNo	DINT	Order number of the PLC error the diagnostic text of which is to be changed: "F2011 PLC - Error no. 1": "ErrorNo": "1" "F2012 PLC - Error no. 2": "ErrorNo": "2" ... "F2017 PLC - Error no. 7": "ErrorNo": "7"
ErrorText	STRING (60)	Diagnostic text to be changed; maximum length: 60 characters (including string end "\0")

Tab. 10-7: Input Variable of the Firmware Function "MX_fSetDriveErrorText"

Return value	Data type	Description
MX_fSetDriveErrorText	BOOL	When the diagnostic text was successfully changed, the function provides the return value TRUE, otherwise FALSE

Tab. 10-8: Return Value of the Firmware Function "MX_fSetDriveErrorText"

10.2.5 MX_fSetDriveWarning

The firmware function "MX_fSetDriveWarning" is used to set or clear a warning diagnosis in the drive.

Depending on the value at the "WarningNo" input (1 to 7), the warning messages E2011 ... E2017 can be triggered in the drive.

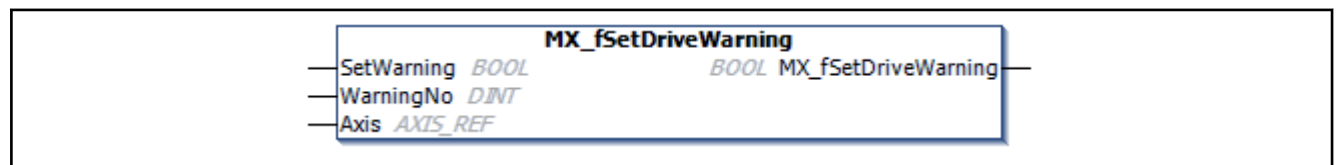


Fig. 10-5: Firmware Function "MX_fSetDriveWarning"

Input variable	Data type	Description
SetWarning	BOOL	TRUE: set warning, FALSE: clear warning
WarningNo	DINT	Order number of warning to be triggered
Axis	AXIS_REF	Axis number (only the global variable "Axis1" can be transmitted here)

Tab. 10-9: Input Variables of the Firmware Function "MX_fSetDriveWarning"

Return value	Data type	Description
MX_fSetDriveWarning	BOOL	If the value of "WarningNo" is within the allowed range (1 to 5), the function provides TRUE as return value, otherwise FALSE

Tab. 10-10: Return Value of the Firmware Function "MX_fSetDriveWarning"

10.2.6 MX_fSetDriveWarningText

With the firmware function "MX_fSetDriveWarningText", the diagnostic text of the PLC warnings E2011 to E2017 can be changed.

By switching the drive off and on again, the PLC warnings are initialized again with their standard diagnostic texts (e.g. "PLC - Warning no. 1").

MX_PLCOpen

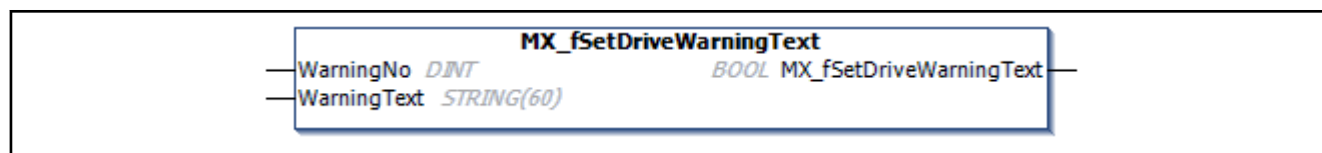


Fig.10-6: Firmware Function "MX_fSetDriveWarningText"

Input variable	Data type	Description
WarningNo	DINT	Order number of the PLC warning the diagnostic text of which is to be changed: "E2011 PLC - Warning no. 1": "WarningNo": "1" "E2012 PLC - Warning no. 2": "WarningNo": "2" ... "E2017 PLC - Warning no. 7": "WarningNo": "7"
ErrorText	STRING (60)	Diagnostic text to be changed; maximum length: 60 characters (including string end "\0")

Tab.10-11: Input Variable of the Firmware Function "MX_fSetDriveWarningText"

Return value	Data type	Description
MX_fSetDriveWarningText	BOOL	When the diagnostic text was successfully changed, the function provides the return value TRUE, otherwise FALSE

Tab.10-12: Return Value of the Firmware Function "MX_fSetDriveWarningText"

10.2.7 MX_SetCmdState

With the firmware function block "MX_SetCmdState" it is possible to set the command status of "P-0-1449, C4900 PLC command".

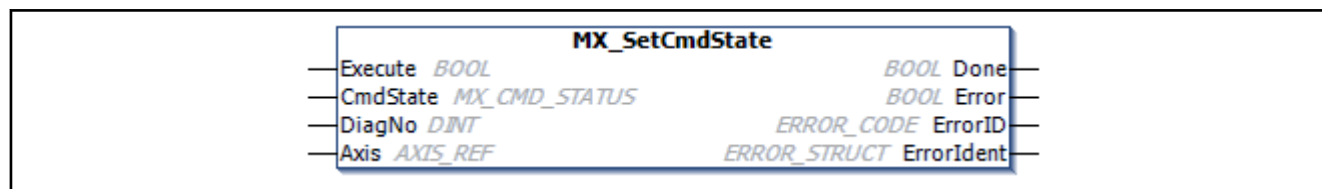


Fig.10-7: Firmware Function "MX_SetCmdState"

Input variable	Data type	Description
Execute	BOOL	See "Signal Time Diagrams"
CmdStatus	MX_CMD_STATUS	At the "CmdStatus" input, the following values can be transmitted to the function block: "MX_CMD_NO_ACTIVE": Command has not been set "MX_CMD_SETTING": Command has been set "MX_CMD_END_OK": Command completed without error "MX_CMD_IS_BREAK": Command interrupted "MX_CMD_IS_WORK": Command has been set and enabled "MX_CMD_BREAK_ERROR": Command aborted with error

Input variable	Data type	Description
DiagNo	DINT	For the case "CmdStatus"==MX_CMD_BREAK_ERROR, a command error number can be transmitted. 1: C4901 PLC command error no. 1 2: C4902 PLC command error no. 2 3: C4903 PLC command error no. 3 4: C4904 PLC command error no. 4
Axis	AXIS_REF	Axis number (only the global variable "Axis1" can be transmitted here)

Tab. 10-13: Input Variables of the Firmware Function "MX_SetCmdState"

Output variable	Data type	Description
Done	BOOL	With "Done"=TRUE the function block signals that the processing of the task has been successfully completed
Error	BOOL	With "Error"=TRUE the function block signals that an error is present
ErrorID *	ERROR_CODE	"ErrorID" gives a rough error description
ErrorIdent *	ERROR_STRUCT	Via the elements: Table, Additional1 and Additional2, "ErrorIdent" shows the exact cause of the error

* See note below

Tab. 10-14: Output Variables of the Firmware Function Block "MX_SetCmdState"



NOTE: For the list of all possible errors, see: "Error Reference Lists"

10.3 "Motion" - Function Blocks/Functions for Drive Control

10.3.1 General Properties

The function blocks/functions for drive control have the following general properties:

- Positioning function blocks have no input for "jerk"
- Function blocks have "AXIS_REF" as "VAR_IN_OUT"
- Positions and distances are transmitted as REAL (floating point)

10.3.2 MB_ChangeCamData



This function block is, at present, available only for IndraDrive. If it is called by the HydraulicDrive, the block always signals an error.

10.3.3 MB_ChangeProfileSet



This function block is, at present, available only for IndraDrive. If it is called by the HydraulicDrive, the block always signals an error.

10.3.4 MB_ChangeProfileStep



This function block is, at present, available only for IndraDrive. If it is called by the HydraulicDrive, the block always signals an error.

MX_PLCOpen

10.3.5 MB_ClearAllError

 I/O modules can, at present, only be addressed by IndraDrive. For the HydraulicDrive, this functionality is, at present, not available.

The firmware function block "MB_ClearAllError" is used to reset errors of all real activated drives and of all activated I/O modules with the command "S-0-0099, C0500 Reset class 1 diagnostics".

In order that the function block can correctly complete the reset command, it has to be cyclically called until it signals "Done"=TRUE or "Error" in the case of error.

If the command "S-0-0099, C0500 Reset class 1 diagnostics" is already active for a drive or I/O module, it should first be completed before "MB_ClearAllError" is called.

If an error of a drive or I/O module cannot be reset via the command "S-0-0099, C0500 Reset class 1 diagnostics", there is **no** error output at the function block.

In the case of deactivated axes or I/O modules, the function block does **not** output an error.

In the case of remote axes, the task is returned to the service channel (sercos III). Only at a later call can the function block signal "Done". Therefore, the resulting times for the function block calls themselves are always short. The outputs "Done" and "Error" have to be cyclically polled. The data transmission can take several seconds, depending on the load of the service channel. As long as the sercos ring is not yet in phase 2 - and temporarily during the transition from phase 3 to phase 4 - it is impossible to exchange data with remote axes. In this case, the function block signals an error.

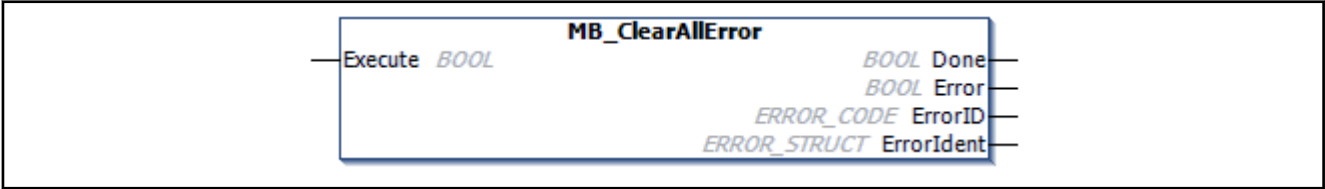


Fig.10-8: Firmware Function Block "MB_ClearAllError"

Input variable	Data type	Description
Execute	BOOL	See "Signal Time Diagrams"

Tab.10-15: Input Variable of the Firmware Function Block "MB_ClearAllError"

Output variable	Data type	Description
Done	BOOL	With "Done"=TRUE the function block signals that processing of the task has been successfully completed.
Error	BOOL	With "Error"=TRUE the function block signals that an error is present
ErrorID *	ERROR_CODE	"ErrorID" gives a rough error description.
ErrorIdent *	ERROR_STRUCT	Via the elements: "Table", "Additional1" and "Additional2", "ErrorIdent" shows the exact cause of the error.

* See note below
Tab.10-16: Output Variables of the Firmware Function Block "MB_ClearAllError"



NOTE: For the list of all possible errors, see: "Error Reference Lists"

10.3.6 MB_GearInPos



This function block is, at present, available only for IndraDrive. If it is called by the HydraulicDrive, the block always signals an error.

10.3.7 MB_Home

With the function block "MB_Home" it is possible to carry out the "drive-controlled homing procedure" in the selected axis.

It is thereby possible to establish the position data reference for non-absolute encoders.

Details for carrying out the homing procedure have to be set in the homing parameter. After the function block was carried out, the axis is in standstill.

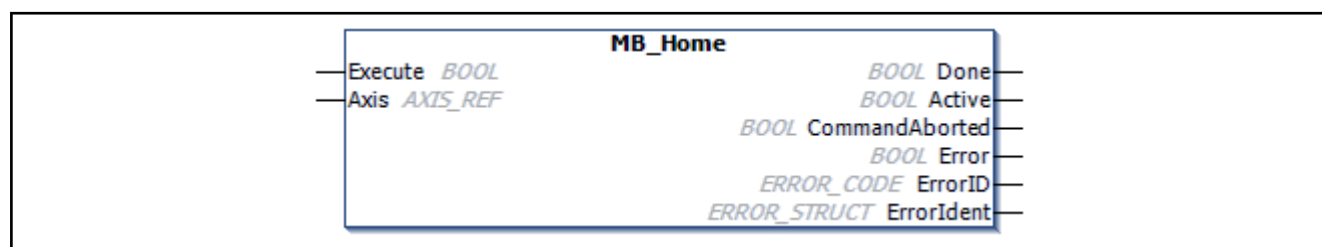


Fig. 10-9: Firmware Function Block "MB_Home"

Input variable	Data type	Description
Execute	BOOL	See "Signal Time Diagrams"

Tab. 10-17: Input Variable of the Firmware Function Block "MB_Home"

Output variable	Data type	Description
Done	BOOL	Homing completed, velocity 0
Active	BOOL	With "Active"=TRUE the firmware function block signals that it determines the axis motion. "Active" remains TRUE until "CommandAborted" or "Error" is set.
CommandAborted	BOOL	With "CommandAborted"=TRUE the function block signals that another motion function block was activated or a drive error was present.
Error	BOOL	With "Error"=TRUE the function block signals that an error is present
ErrorID *	ERROR_CODE	"ErrorID" gives a rough error description.
ErrorIdent *	ERROR_STRUCT	Via the elements: "Table", "Additional1" and "Additional2", "ErrorIdent" shows the exact cause of the error.

* See note below

Tab. 10-18: Output Variables of the Firmware Function Block "MB_Home"



NOTE: For the list of all possible errors, see: "Error Reference Lists"

MX_PLCOpen

Input/output variable (VAR_IN_OUT)	Data type	Description
Axis	AXIS_REF	Defines the axis. The following axes are possible: MPB (IndraDrive) and HDC (HydraulicDrive): Axis1 MPC (IndraDrive): - Axis1 (master) or - Axis2...Axis10 (a total of 9 slaves / remote axes) The remote axes contained in the list "P-0-1801.0.10, CCD: Addresses of projected drives" correspond to the order of Axis2 to Axis10 (Axis2 acts on the drive of list element 0 of P-0-1801.0.10,...)

Tab. 10-19: Input/Output Variables of the Firmware Function Block "MB_Home"

10.3.8 MB_MotionProfile



This function block is, at present, available only for IndraDrive. If it is called by the HydraulicDrive, the block always signals an error.

10.3.9 MB_Phasing



This function block is, at present, available only for IndraDrive. If it is called by the HydraulicDrive, the block always signals an error.

10.3.10 MB_PhasingSlave



This function block is, at present, available only for IndraDrive. If it is called by the HydraulicDrive, the block always signals an error.

10.3.11 MB_PreSetMode

With the function block "MB_PreSetMode" the axis can be prepared for command triggering before power is switched on (PreSetMode / preselecting operation modes).

"MB_PreSetMode" already commands the axis with the corresponding data, but the axis only starts when power is switched on with "MC_Power" and then directly follows the input values.



"MB_PreSetMode" can only be activated in the operating mode without power and without error. After activation with a rising "Execute" edge it is allowed to activate a motion function block (such as "MC_MoveVelocity") or "MX_SetOpMode".

The following points exemplify a typical sequence when using the function block "MB_PreSetMode":

1. Switch drive on
2. Switch or wait until drive in operating mode ("bb")
3. Activate "MB_PreSetMode" with rising "Execute" edge
4. Start a motion function block (in "Ab")
5. Switch power on with "MC_Power"

The axis immediately starts moving in the desired way

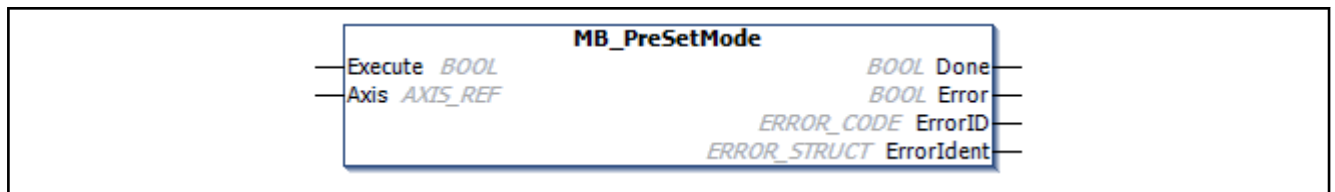


Fig. 10-10: Firmware Function Block "MB_PreSetMode"

Input variable	Data type	Description
Execute	BOOL	See "Signal Time Diagrams"

Tab. 10-20: Input Variable of the Firmware Function Block "MB_PreSetMode"

Output variable	Data type	Description
Done	BOOL	With "Done"=TRUE the function block signals that processing of the task has been successfully completed
Error	BOOL	With "Error"=TRUE the function block signals that an error is present
ErrorID *	ERROR_CODE	"ErrorID" gives a rough error description.
ErrorIdent *	ERROR_STRUCT	Via the elements: "Table", "Additional1" and "Additional2", "ErrorIdent" shows the exact cause of the error.
Value	DINT	If the output "Done"=TRUE, "Value" signals the value of the selected element

* See note below

Tab. 10-21: Output Variables of the Firmware Function Block "MB_PreSetMode"



NOTE: For the list of all possible errors, see: "Error Reference Lists"

Input/output variable (VAR_IN_OUT)	Data type	Description
Axis	AXIS_REF	Defines the axis. The following axes are possible: MPB (IndraDrive) and HDC (HydraulicDrive): Axis1 MPC (IndraDrive): - Axis1 (master) or - Axis2...Axis10 (a total of 9 slaves / remote axes) The remote axes contained in the list "P-0-1801.0.10, CCD: Addresses of projected drives" correspond to the order of Axis2 to Axis10 (Axis2 acts on the drive of list element 0 of P-0-1801.0.10,...)

Tab. 10-22: Input / Output Variable of the Firmware Function Block "MB_PreSetMode"

10.3.12 MB_SetPositionControlMode

The function block "MB_SetPositionControlMode" is used to set the position controller.

MX_PLCOpen



Fig. 10-11: Firmware Function Block "MB_SetPositionControlMode"

Input variable	Data type	Description
Execute	BOOL	See "Signal Time Diagrams"
LaglessPositionControl	BOOL	The value at the "LaglessPositionControl" input influences bit 2 of "S-0-0520, Control word of axis controller": FALSE: Position control with lag error TRUE: Lagless position control
ExternalEncoder	BOOL	The value at the "ExternalEncoder" input influences bit 0 of "S-0-0520, Control word of axis controller": FALSE: Control with encoder 1 (actual position value-1) TRUE: Control with encoder 2 (actual position value-2)
MoveAbsUseCmdPos	BOOL	The value at the "MoveAbsUseCmdPos" input influences bit 4 of "S-0-0346, Positioning control word": FALSE: Current actual position value as dedicated point for relative positioning command values TRUE: Last effective target position as dedicated point for relative positioning command values

Tab. 10-23: Input Variables of the Firmware Function Block "MB_SetPositionControlMode"

Output variable	Data type	Description
Done	BOOL	With "Done"=TRUE the function block signals that processing of the task has been successfully completed
Active	BOOL	The function block is processed very quickly; the change of states of the "Active" output cannot be evaluated with IndraMotion MLD
Error	BOOL	With "Error"=TRUE the function block signals that an error is present
ErrorID *	ERROR_CODE	"ErrorID" gives a rough error description.
ErrorIdent *	ERROR_STRUCT	Via the elements: "Table", "Additional1" and "Additional2", "ErrorIdent" shows the exact cause of the error.

* See note below

Tab. 10-24: Output Variables of the Firmware Function Block "MB_SetPositionControlMode"



NOTE: For the list of all possible errors, see: "Error Reference Lists"

Input/output variable (VAR_IN_OUT)	Data type	Description
Axis	AXIS_REF	Defines the axis. The following axes are possible: MPB (IndraDrive) and HDC (HydraulicDrive): Axis1 MPC (IndraDrive): - Axis1 (master) or - Axis2...Axis10 (a total of 9 slaves / remote axes) The remote axes contained in the list "P-0-1801.0.10, CCD: Addresses of projected drives" correspond to the order of Axis2 to Axis10 (Axis2 acts on the drive of list element 0 of P-0-1801.0.10,...)

Tab. 10-25: Input / Output Variable of the Firmware Function Block "MB_SetPositionControlMode"

10.3.13 MB_Stop

The firmware function block "MB_Stop" is used to bring the drive to the status STOP.

With the rising edge at "ExecuteLock" the stopping process is started.

If the PLC does not have control over the drive, an error is returned.

As long as the "ExecuteLock" input is TRUE, no other motion function block can be started. For this purpose, the drive is brought to the status "AH" (Drive Halt). Further command triggering is only possible when the axis is in standstill. With "ExecuteLock"=FALSE the drive remains in "AH".

With the deceleration set at "Deceleration" the drive is brought to a standstill.

When several instances are called by "MC_Stop" or "MB_Stop" in a project, it is always the last instance which takes effect.

"MC_Stop" has higher priority than "MB_Stop"; this means that a running "MC_Stop" cannot be aborted by an "MB_Stop".



To test the function block, the instance variable "ActivateCounter" can be used. It counts the positive edges of the "ExecuteLock" signal and thereby signals whether the function block is run with an edge.

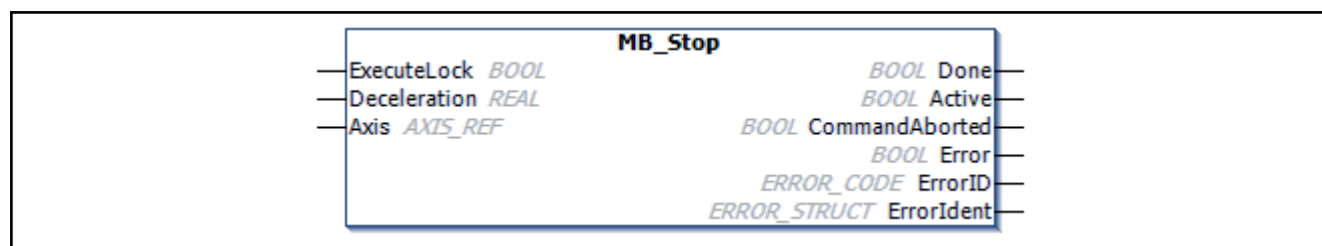


Fig. 10-12: Firmware Function Block "MB_Stop"

Input variable	Data type	Description
ExecuteLock	BOOL	With a rising edge at "ExecuteLock" the drive is switched to "AH".
Deceleration	REAL	This input sets the deceleration. The set deceleration is written to parameter "S-0-0372, Drive Halt acceleration bipolar".

Tab. 10-26: Input Variables of the Firmware Function Block "MB_Stop"

MX_PLCOpen

Output variable	Data type	Description
Done	BOOL	With "Done"=TRUE (S-0-0331, bit0="1") the firmware function block signals that the drive has stopped. Only when the "ExecuteLock" input is TRUE is the "Done" output generated
Active	BOOL	With "Active"=TRUE the firmware function block signals that it determines the axis motion
CommandAborted	BOOL	With "CommandAborted"=TRUE the function block signals that another function block has commanded another status (other "MC_Stop" or "MB_Stop")
Error	BOOL	With "Error"=TRUE the function block signals that an error is present
ErrorID *	ERROR_CODE	"ErrorID" gives a rough error description.
ErrorIdent *	ERROR_STRUCT	Via the elements: "Table", "Additional1" and "Additional2", "ErrorIdent" shows the exact cause of the error.

* See note below

Tab. 10-27: *Output Variables of the Firmware Function Block "MB_Stop"*



NOTE: For the list of all possible errors, see: "Error Reference Lists"

Input/output variable (VAR_IN_OUT)	Data type	Description
Axis	AXIS_REF	Defines the axis. The following axes are possible: MPB (IndraDrive) and HDC (HydraulicDrive): Axis1 MPC (IndraDrive): - Axis1 (master) or - Axis2...Axis10 (a total of 9 slaves / remote axes) The remote axes contained in the list "P-0-1801.0.10, CCD: Addresses of projected drives" correspond to the order of Axis2 to Axis10 (Axis2 acts on the drive of list element 0 of P-0-1801.0.10,...)

Tab. 10-28: *Input/Output Variable of the Firmware Function Block "MB_Stop"*

10.3.14 MB_TorqueControl

The function block activates the closed-loop torque/force control operation mode, and the drives follows the command value input.

The command value is a torque/force command value.

Command values can be changed via a ramp.

With "**MC_TorqueControl**", only the parameter for the ramp height has to be written for the ramp. The ramp time is taken from S-0-0823, Torque/force ramp time and must therefore be set accordingly in advance, if required.

In the case of "**MB_TorqueControl**", the block additionally writes the ramp time. This will not result in changes in the time characteristics and functionality of the block.

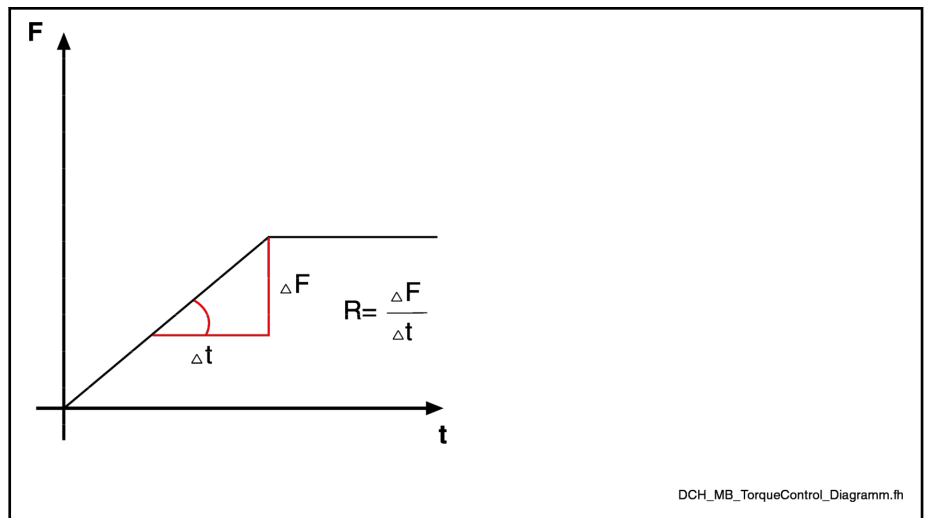


Fig. 10-13: Torque/Force Command Value Ramp



The outputs of the function block "MB_TorqueControl" behave like those of "MC_MoveVelocity", that is, the Active output remains TRUE as long as the torque control operation mode is active.

Target	Library	Range
IndraDrive MPx18 HydraulicDrive HDx-18	MX_PLCOpen.lib	Motion

Tab. 10-29: Library Assignment

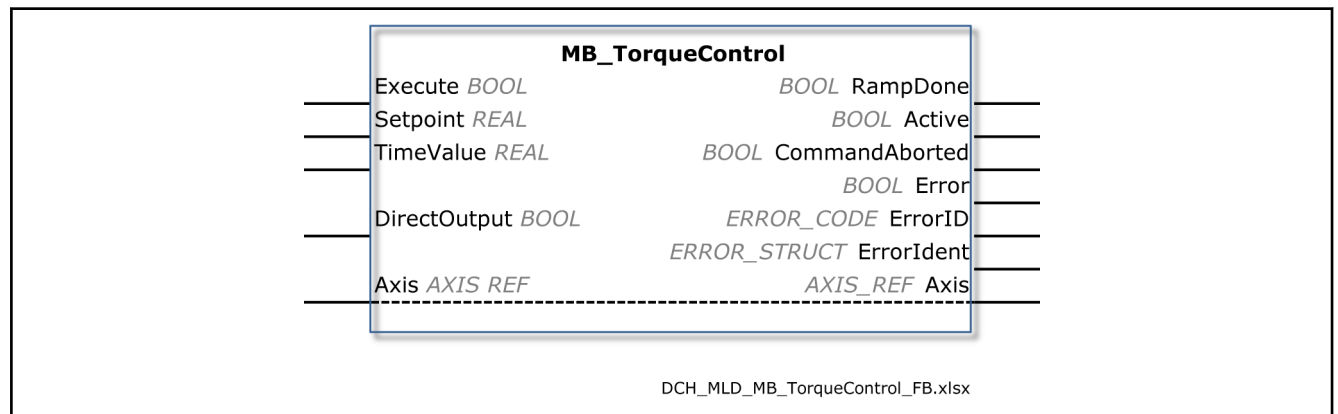


Fig. 10-14: Function Block "MB_TorqueControl"

Operation mode parameters	PLC mode parameters
S-0-0080, Torque/force command value	P-0-1465, PLC/setting-up mode Torque/force command value
S-0-0822, Torque/force ramp	P-0-1466, PLC Torque/force ramp
S-0-0823, Torque/force ramp time	P-0-1467, PLC/setting-up mode, torque/force ramp time

Tab. 10-30: Comparison of Operation Mode Parameters

MX_PLCOpen

Input variable	Data type	Description
Execute	BOOL	A rising edge starts processing of the function block.
Torque	REAL	S-0-0080 Torque/force command value
TorqueRamp	REAL	S-0-0822 Torque/force ramp
RampTime	REAL	S-0-0823 Torque/force ramp time
Axis	AXIS_REF	Axis1 (only axis for MLD-S or local axis for MLD-M). Axis2 to Axis8 correspond to the remote axes. The remote axes contained in the list "P-0-1601, CCD: Addresses of projected drives" correspond to the order of Axis2 to Axis8 (Axis2 acts on the drive of list element 0 of P-0-1601)

Tab. 10-31: Input Variables of the Firmware Function Block "MC_TorqueControl"

Return value	Data type	Description
InTorque	BOOL	Torque/force command value has been reached
Active	BOOL	Function block or operation mode is active
CommandAborted	BOOL	Function block was aborted
Error	BOOL	An error occurred while processing
ErrorID	ERROR_CODE	"ErrorID" gives a rough error description
Errorident	ERROR_STRUCTUR	Via the elements: Table, Additional1 and Additional2, "Errorident" shows the exact cause of the error

Tab. 10-32: Return values of the function block "MC_TorqueControl"



Parameter S-0-0826, Torque/force window, is used to determine, when the function block provides the message InTorque.

If the TorqueRamp input or the RampTime input is set to 0, the torque/force command value is provided abruptly.

The figure below shows the timing of the function block:

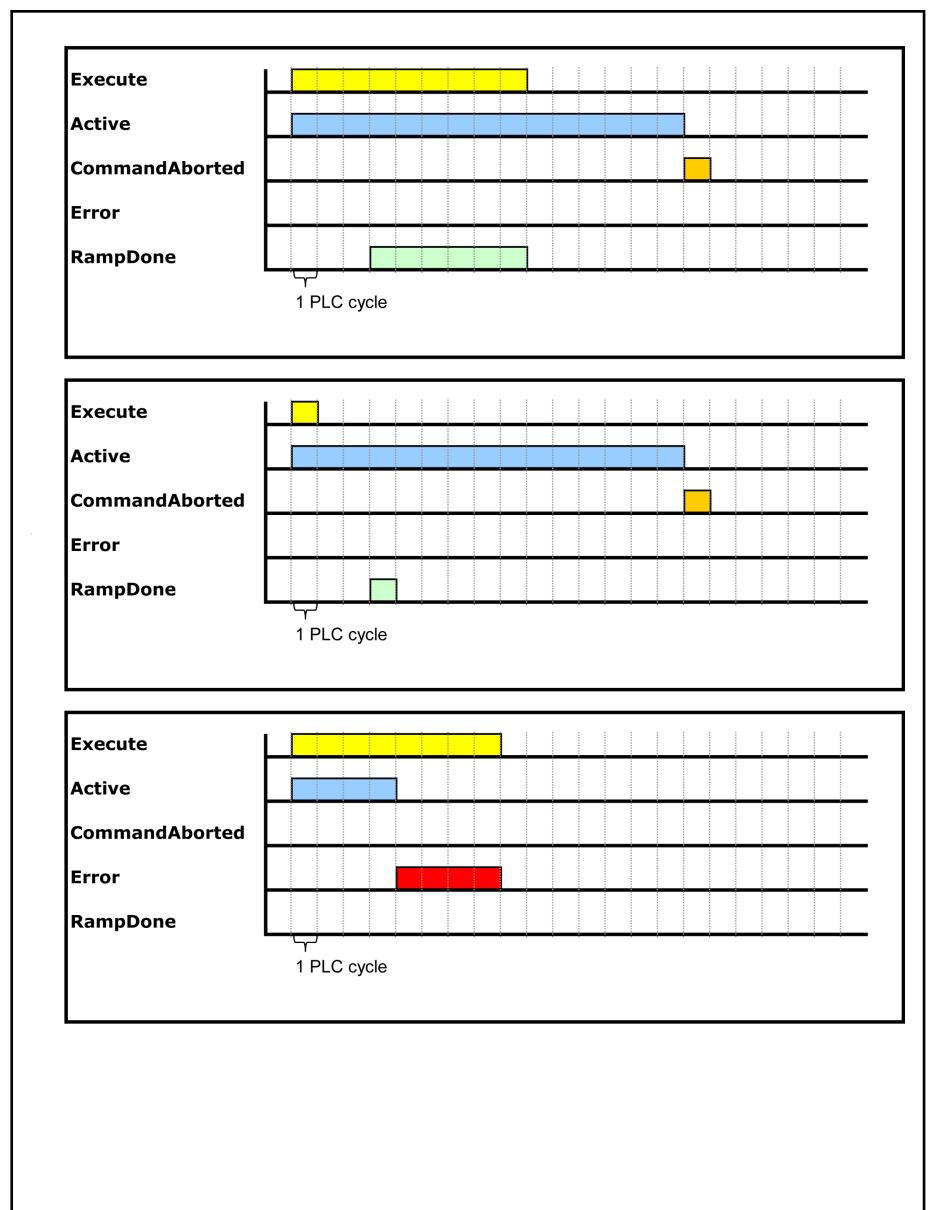


Fig. 10-15: Timing of Function Block MB_TorqueControl

10.3.15 MB_ValveControl

The function block "MB_ValveControl" activates the operation mode "direct valve control" and the drive follows the command value input.

The command value is a valve opening in % (-100 % to +100 %)

Command values can be changed via a ramp.

The input "DirectOutput" can be used to deactivate control variable adjustment (adjustment to a real valve), if necessary.

MX_PLCOpen

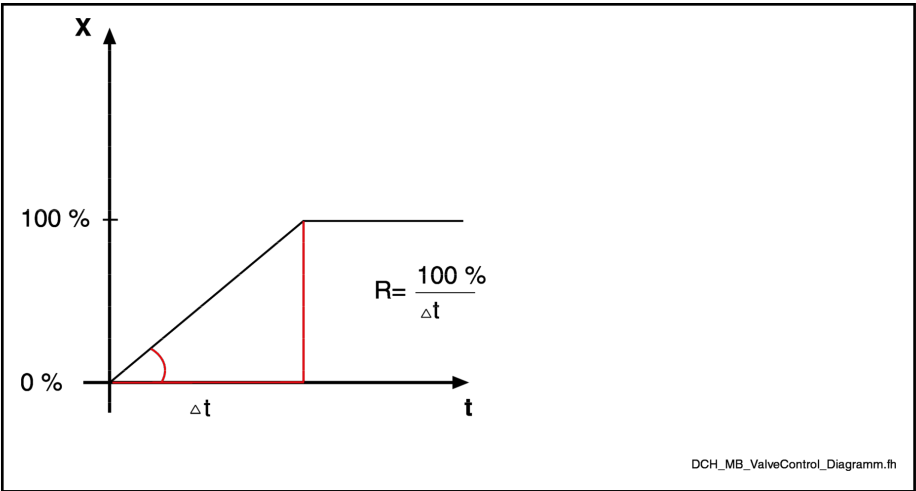


Fig.10-16: Control Variable Adjustment

 The outputs of the function block "MB_ValveControl" behave like those of "MC_MoveVelocity", that is, the Active output remains TRUE as long as the direct valve control operation mode is active.

Target	Library	Range
HydraulicDrive HDx-18	MX_PLCOpen.lib	Motion

Tab.10-33: Library Assignment

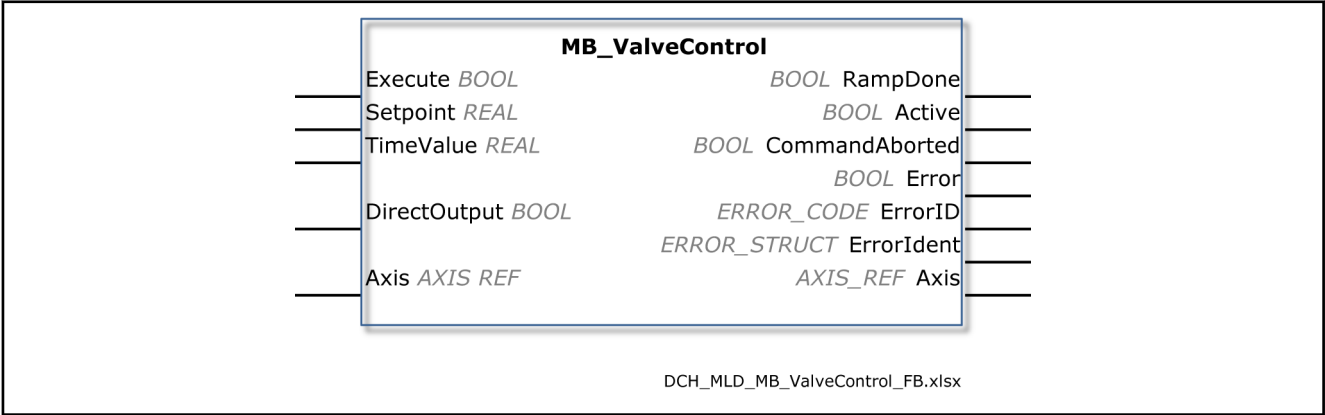


Fig.10-17: Function Block "MB_ValveControl"

Operation mode parameters	PLC mode parameters
S-0-0860, Spool command value	P-0-1468, PLC/setting-up mode, valve command value
S-0-0869, Spool ramp time	P-0-1469, PLC/setting-up mode, valve ramp time
P-0-2928.0.1, Valve command value control word, Bit 1: Direct valve command value output	

Tab.10-34: Comparison of Operation Mode Parameters

Input variable	Data type	Description
Execute	BOOL	A rising edge starts processing of the function block.
TargetValue	REAL	S-0-0080 Torque/force command value

Input variable	Data type	Description
TimeValue	REAL	S-0-0869 Spool ramp time referred to 100 % valve opening.
DirectOutput	BOOL	Input for bypassing control variable adjustment (P-0-2928.0.1, bit 1). If this input is omitted, control variable adjustment is used.
Axis	AXIS_REF	Local Axes: Axis1 and Axis2 (if available).

Tab. 10-35: Input variables of the function block "MC_ValveControl"

Return value	Data type	Description
RampDone	BOOL	Valve command value has been reached
Active	BOOL	Function block or operation mode is active
CommandAborted	BOOL	Function block was aborted
Error	BOOL	An error occurred while processing
ErrorID	ERROR_CODE	"ErrorID" gives a rough error description
ErrorIdent	ERROR_STRUCTUR	Via the elements: Table, Additional1 and Additional2, "ErrorIdent" shows the exact cause of the error

Tab. 10-36: Return values of the function block "MC_TorqueControl"



When the ramp has been completed or the output of the valve command value generator is equal to the valve command value, the function block provides the message RampDone. When the input TimeValue is set to 0, the valve command value is fed forward abruptly. The ramp is completed immediately.

The table below shows the timing of the function block:

MX_PLCOpen

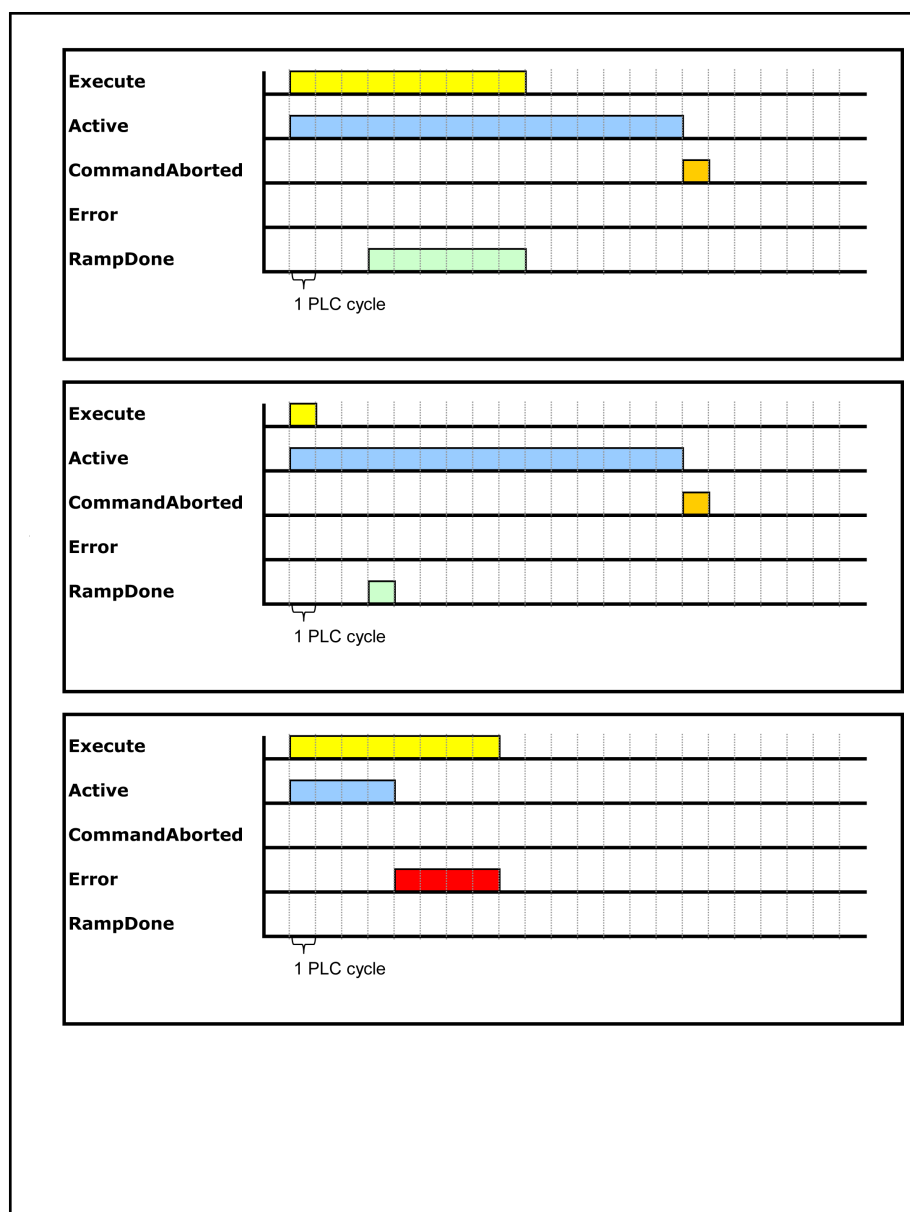


Fig.10-18: Timing of Function Block MB_ValveControl

10.3.16 MC_CamIn



This function block is, at present, available only for IndraDrive. If it is called by the HydraulicDrive, the block always signals an error.

10.3.17 MC_CamOut



This function block is, at present, available only for IndraDrive. If it is called by the HydraulicDrive, the block always signals an error.

10.3.18 MC_GearIn



This function block is, at present, available only for IndraDrive. If it is called by the HydraulicDrive, the block always signals an error.

10.3.19 MC_GearOut



This function block is, at present, available only for IndraDrive. If it is called by the HydraulicDrive, the block always signals an error.

10.3.20 MC_Jog

The firmware function block "MC_Jog" is used to jog the drive ("infinite travel" positive or "infinite travel" negative). For this purpose, the jog mode is internally activated in the operation mode "drive-controlled positioning".

For permanent control and remote axes, the secondary operation mode 3 and its parameters are used. For temporary control, the secondary PLC operation mode 11 and its parameters are used.

The jog velocity must be set at the "JogVelo" input. Via the "JogAcc" and "JogDec" inputs, you must set how the jog velocity is reached.



Only one command triggering can be preset in a PLC cycle.

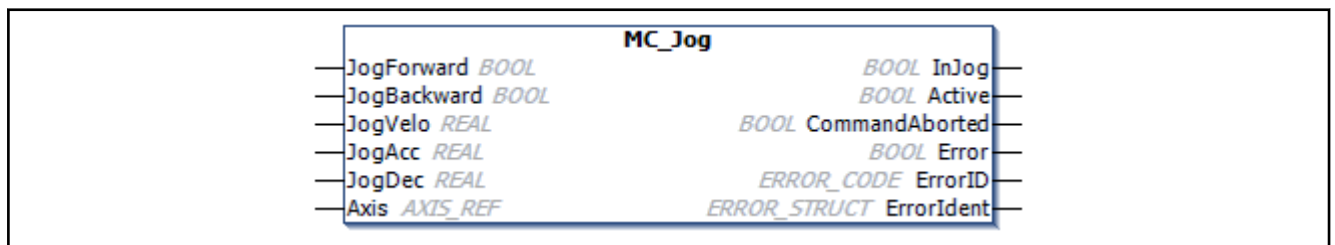


Fig. 10-19: Firmware Function Block "MC_Jog"

Input variable	Data type	Description
JogForward	BOOL	With a rising edge, the "JogVelo", "JogAcc", "JogDec" and "Axis" inputs are applied. The operation mode "drive-controlled positioning" is activated and "jog positive" is started [positioning control word (real axis: S-0-0346, Bit1/0=11)]. When "JogForward" is reset (FALSE), positioning stop is commanded (real axis: S-0-0346, Bit2/1/0=111)
JogBackward	BOOL	With a rising edge, the "JogVelo", "JogAcc", "JogDec" and "Axis" inputs are applied. The operation mode "drive-controlled positioning" is activated and "jog negative" is started [positioning control word (real axis: S-0-0346, Bit2/1/0=101)]. When "JogBackward" is reset (FALSE), positioning stop is commanded (real axis: S-0-0346, Bit2/1/0=111).
JogVelo	REAL	Positioning velocity (real axis: S-0-0259) must be ≥ 0 . Note: When the function block is active, the "JogVelo" input is cyclically evaluated and applied in the case of changes.
JogAcc	REAL	Positioning acceleration (real axis: S-0-0260) must be > 0 . Note: When the function block is active, the "JogAcc" input is cyclically evaluated and applied in the case of changes.
JogDec	REAL	Positioning deceleration (real axis: S-0-0359) must be > 0 . Note: When the function block is active, the "JogDec" input is cyclically evaluated and applied in the case of changes.

Tab. 10-37: Input Variables of the Firmware Function Block "MC_Jog"

MX_PLCOpen

Output variable	Data type	Description
InJog	BOOL	With "InJog"=TRUE the function block signals that the actual velocity is outside of the standstill window (real axis: S-0-0124).
Active	BOOL	With "Active"=TRUE the firmware function block signals that it determines the axis motion.
CommandAborted	BOOL	With "CommandAborted"=TRUE the function block signals that another motion function block was activated or a drive error was present.
Error	BOOL	With "Error"=TRUE the function block signals that an error is present
ErrorID *	ERROR_CODE	"ErrorID" gives a rough error description.
ErrorIdent *	ERROR_STRUCT	Via the elements: "Table", "Additional1" and "Additional2", "ErrorIdent" shows the exact cause of the error.

* See note below
 Tab. 10-38: *Output Variables of the Firmware Function Block "MC_Jog"*



NOTE: For the list of all possible errors, see: "Error Reference Lists"

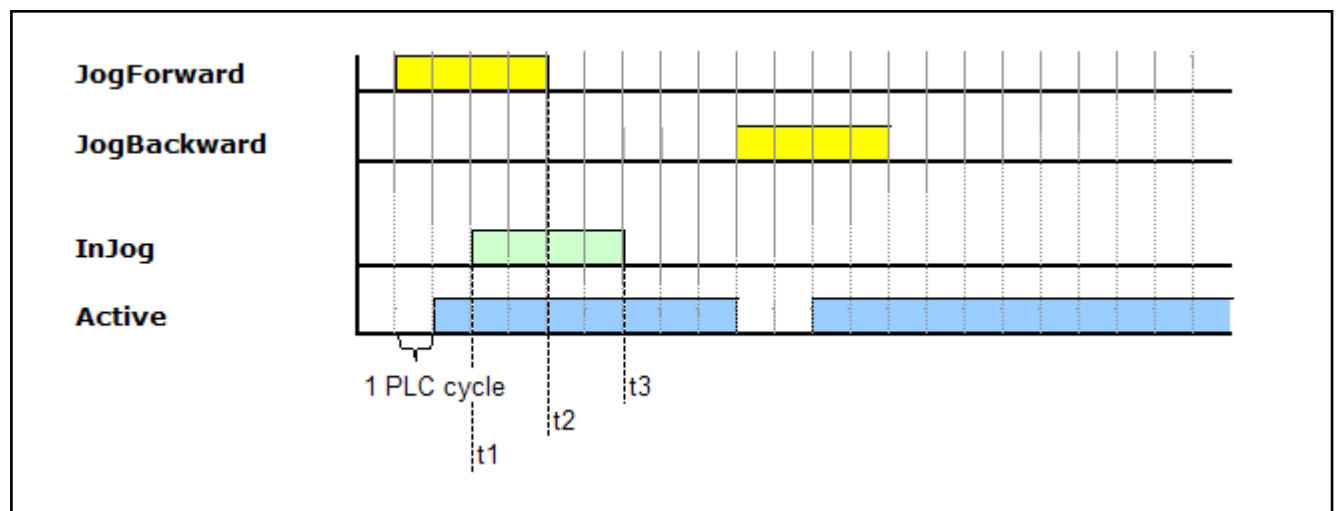
Input/output variable (VAR_IN_OUT)	Data type	Description
Axis	AXIS_REF	Defines the axis. The following axes are possible: Defines the axis. The following axes are possible: MPB (IndraDrive) and HDC (HydraulicDrive): Axis1 MPC (IndraDrive): - Axis1 (master) or - Axis2...Axis10 (a total of 9 slaves / remote axes) The remote axes contained in the list "P-0-1801.0.10, CCD: Addresses of projected drives" correspond to the order of Axis2 to Axis10 (Axis2 acts on the drive of list element 0 of P-0-1801.0.10,...)

Tab. 10-39: *Input/Output Variable of the Firmware Function Block "MC_Jog"*

Timing Diagrams

The figure below shows the timing diagram with active jog mode ("jog positive" or "jog negative"). By a positive edge at the "JogForward" input, the input parameters ("JogVelo", etc.) are applied and infinite travel in positive direction (jog positive) is activated in the operation mode "drive-controlled positioning". The "InJog" output is set when the actual velocity is greater than the standstill window (S-0-0124); "InJog" is removed again when the actual velocity is smaller than the standstill window.

The "Active" output remains set until the function block is restarted (commanded) or interrupted by another function block instance (e.g. "MC_MoveAbsolute").



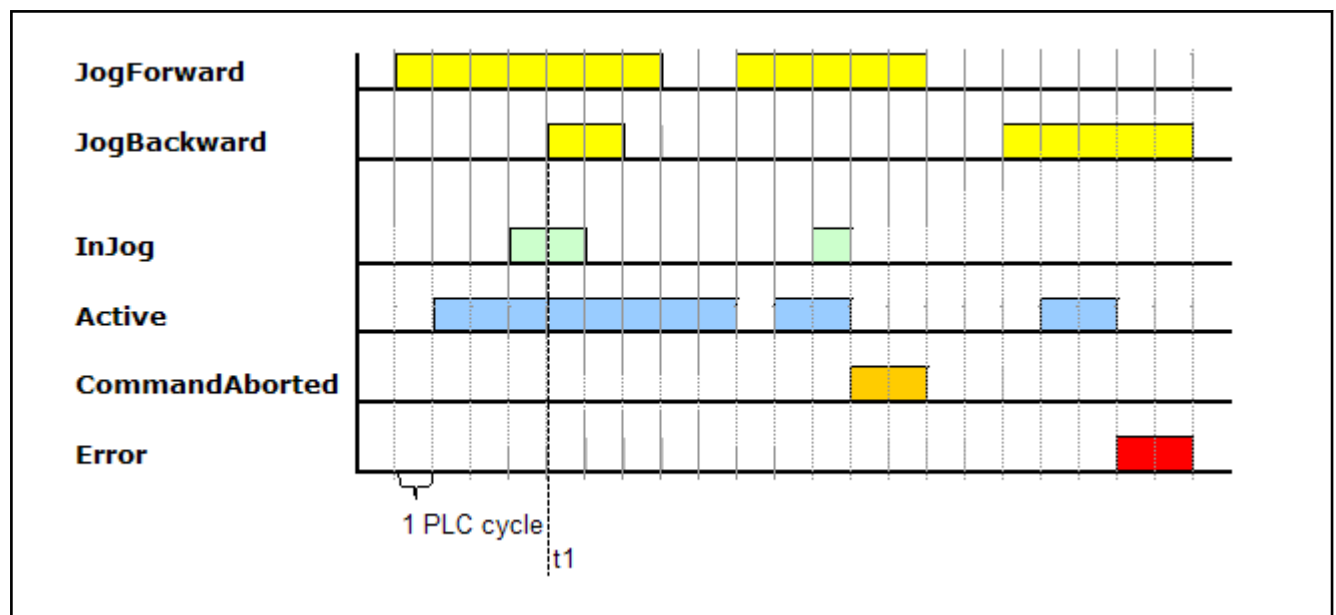
t1 Actual velocity outside of standstill window (real axis: S-0-0124)

t2 Axis is stopped

t3 Actual velocity within standstill window (real axis: S-0-0124 / virtual master axis: P-0-0912)

Fig. 10-20: Timing of the "Active" Output

The figure below shows the timing behavior of the other standard outputs.



t1 Axis is stopped

Fig. 10-21: Timing of "Active", "CommandAborted" and "Error" Outputs

10.3.21 MC_MoveAbsolute

The firmware function block "MC_MoveAbsolute" is used to move the drive to a preset absolute position. Internally the operation mode "drive-controlled positioning" is activated for this purpose. Make the settings of the position controller (with/without lag error, actual position value 1/2) with the function block "MB_SetPositionControlMode".

For permanent control and remote axes, the secondary operation mode 3 and its parameters are used. For temporary control, the secondary PLC operation mode 11 and its parameters are used.

MX_PLCOpen

The target position must be set at the "Position" input. Via the inputs "Velocity", "Acceleration" and "Deceleration" you have to set how the drive moves to the target position (motion profile).



Via the firmware function block "MC_MoveAbsolute" the "P-0-0758, Virtual master axis, actual position value" of the master axis generator can be positioned in absolute form, too. For this purpose, the master axis generator has to be activated via "P-0-0917, Control word of master axis generator", bit0="1". To convert the virtual actual position value (P-0-0758) into the internal virtual master axis position (VmAxis1, "P-0-0761, Master axis position for slave axis"), the parameter "P-0-0758, Virtual master axis, actual position value" must be written to parameter "P-0-0916, Master axis format converter signal selection".

See also Functional Description of firmware "Master Axis Generator" and "Master Axis Format Converter"

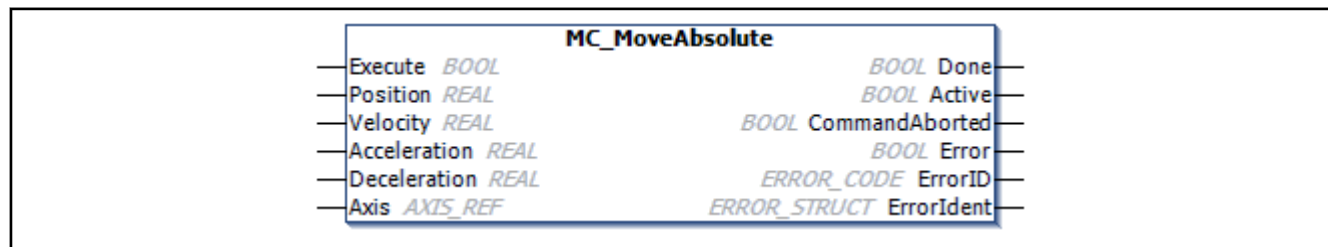


Fig. 10-22: Firmware Function Block "MC_MoveAbsolute"

Operation mode parameters	PLC mode parameters
S-0-0282, Positioning command value	P-0-1450, PLC Positioning command value
S-0-0259, Positioning velocity	P-0-1451, PLC Positioning velocity
S-0-0260, Positioning acceleration	P-0-1452, PLC/setting-up mode, positioning acceleration
S-0-0346, Positioning control word	P-0-1454, PLC Positioning command value acceptance
S-0-0359, Positioning deceleration	P-0-1453, PLC Positioning deceleration

Tab. 10-40: Comparison of Operation Mode Parameters

Input variable	Data type	Description
Execute	BOOL	See "Signal Time Diagrams"
Position	REAL	The absolute position must be preset at the "Position" input.
Velocity	REAL	Positioning velocity in physical unit
Acceleration	REAL	Positioning acceleration in physical unit
Deceleration	REAL	Positioning deceleration in physical unit

Tab. 10-41: Input Variables of the Firmware Function Block "MC_MoveAbsolute"

Output variable	Data type	Description
Done	BOOL	With "Done"=TRUE the firmware function block signals that the absolute target position has been reached. The processing is completed until the next 0-1 edge at the "Execute" input. When a 1-0 edge is applied to the "Execute" input before the firmware function block signals that it has been completed ("Done"=TRUE), the "Done" output is set to TRUE for one call when the motion target has been reached and processing of the firmware function block has not been interrupted by an error or another firmware function block. Real axes: The output is set to TRUE when bit 2 is set in parameter "S-0-0437, Positioning status".
Active	BOOL	With "Active"=TRUE the firmware function block signals that it determines the axis motion.
CommandAborted	BOOL	With "CommandAborted"=TRUE the function block signals that another motion function block was activated or a drive error was present. A drive error is displayed via the Error outputs of "MC_Power"
Error	BOOL	With "Error"=TRUE the function block signals that an error is present
ErrorID *	ERROR_CODE	"ErrorID" gives a rough error description.
ErrorIdent *	ERROR_STRUCT	Via the elements: "Table", "Additional1" and "Additional2", "ErrorIdent" shows the exact cause of the error.

* See note below

Tab. 10-42: Output Variables of the Firmware Function Block "MC_MoveAbsolute"



NOTE: For the list of all possible errors, see: "Error Reference Lists"

Input/output variable (VAR_IN_OUT)	Data type	Description
Axis	AXIS_REF	Defines the axis. The following axes are possible: - Defines the axis. The following axes are possible: MPB (IndraDrive) and HDC (HydraulicDrive): Axis1 MPC (IndraDrive): - Axis1 (master) or - Axis2...Axis10 (a total of 9 slaves / remote axes) The remote axes contained in the list "P-0-1801.0.10, CCD: Addresses of projected drives" correspond to the order of Axis2 to Axis10 (Axis2 acts on the drive of list element 0 of P-0-1801.0.10,...)

Tab. 10-43: Input/Output Variables of the Firmware Function Block "MC_MoveAbsolute"

10.3.22 MC_MoveAdditive

The firmware function block "MC_MoveAdditive" is used to move the drive in relative form by a distance in addition to the target position. The new target position results from the old target position, plus a distance. The distance is determined by the "Distance" input. The inputs "Velocity", "Acceleration" and "Deceleration" determine the motion profile.

Internally the operation mode "drive-controlled positioning" is activated. Make the settings of the position controller (with/without lag error, actual position value 1/2) with the function block "MB_SetPositionControlMode".

MX_PLCOpen

For permanent control and remote axes, the secondary operation mode 3 and its parameters are used. For temporary control, the secondary PLC operation mode 11 and its parameters are used.

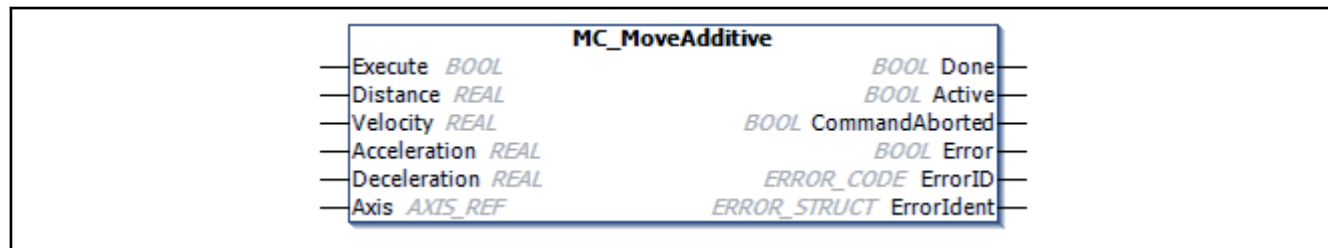


Fig. 10-23: Firmware Function Block "MC_MoveAdditive"

Operation mode parameters	PLC mode parameters
S-0-0282, Positioning command value	P-0-1450, PLC Positioning command value
S-0-0259, Positioning velocity	P-0-1451, PLC Positioning velocity
S-0-0260, Positioning acceleration	P-0-1452, PLC/setting-up mode, positioning acceleration
S-0-0346, Positioning control word	P-0-1454, PLC Positioning command value acceptance
S-0-0359, Positioning deceleration	P-0-1453, PLC Positioning deceleration

Tab. 10-44: Comparison of Operation Mode Parameters

Input variable	Data type	Description
Execute	BOOL	With a rising edge at "Execute" the data at the inputs are applied to the firmware function block; processing starts. With a falling edge, all outputs are cleared. The firmware function block remains active until one of the two outputs "Done" and "Error" is set to TRUE. The status persists for one call before the outputs are reset
Distance	REAL	At the "Distance" input the relative distance must be preset; the new target position results from the old target position and the value set at the "Distance" input.
Velocity	REAL	Positioning velocity in physical unit
Acceleration	REAL	Positioning acceleration in physical unit
Deceleration	REAL	Positioning deceleration in physical unit

Tab. 10-45: Input Variables of the Firmware Function Block "MC_MoveAdditive"

Output variable	Data type	Description
Done	BOOL	With "Done"=TRUE the firmware function block signals that the target position has been reached. The processing is completed until the next 0-1 edge at the "Execute" input. When a 1-0 edge is applied to the "Execute" input before the firmware function block signals that it has been completed ("Done"=TRUE), the "Done" output is set to TRUE for one call when the motion target has been reached and processing of the firmware function block has not been interrupted by an error or another firmware function block. Real axes: The output is set to TRUE when bit 2 is set in parameter "S-0-0437, Positioning status".
Active	BOOL	With "Active"=TRUE the firmware function block signals that it determines the axis motion.
CommandAborted	BOOL	With "CommandAborted"=TRUE the firmware function block signals that another motion function block was activated or a drive error was present. A drive error is displayed via the Error outputs of "MC_Power".
Error	BOOL	With "Error"=TRUE the function block signals that an error is present
ErrorID *	ERROR_CODE	"ErrorID" gives a rough error description.
ErrorIdent *	ERROR_STRUCT	Via the elements: "Table", "Additional1" and "Additional2", "ErrorIdent" shows the exact cause of the error.

* See note below

Tab. 10-46: Output Variables of the Firmware Function Block "MC_MoveAdditive"



NOTE: For the list of all possible errors, see: "Error Reference Lists"

Input/output variable (VAR_IN_OUT)	Data type	Description
Axis	AXIS_REF	Defines the axis. The following axes are possible: - Defines the axis. The following axes are possible: MPB (IndraDrive) and HDC (HydraulicDrive): Axis1 MPC (IndraDrive): - Axis1 (master) or - Axis2...Axis10 (a total of 9 slaves / remote axes) The remote axes contained in the list "P-0-1801.0.10, CCD: Addresses of projected drives" correspond to the order of Axis2 to Axis10 (Axis2 acts on the drive of list element 0 of P-0-1801.0.10,...)

Tab. 10-47: Input/Output Variable of the Firmware Function Block "MC_MoveAdditive"

10.3.23 MC_MoveRelative

The firmware function block "MC_MoveRelative" is used to move the drive in relative form by a distance, starting from the current actual position.

Internally the operation mode "drive-controlled positioning" is activated for this purpose. Make the settings of the position controller (with/without lag error, actual position value 1/2) with the function block "MB_SetPositionControlMode".

MX_PLCOpen

For permanent control and remote axes, the secondary operation mode 3 and its parameters are used. For temporary control, the secondary PLC operation mode 11 and its parameters are used.

The relative distance to be moved has to be preset at the "Distance" input. Via the inputs "Velocity", "Acceleration" and "Deceleration" you have to set the motion profile (determines how the relative distance is traveled).

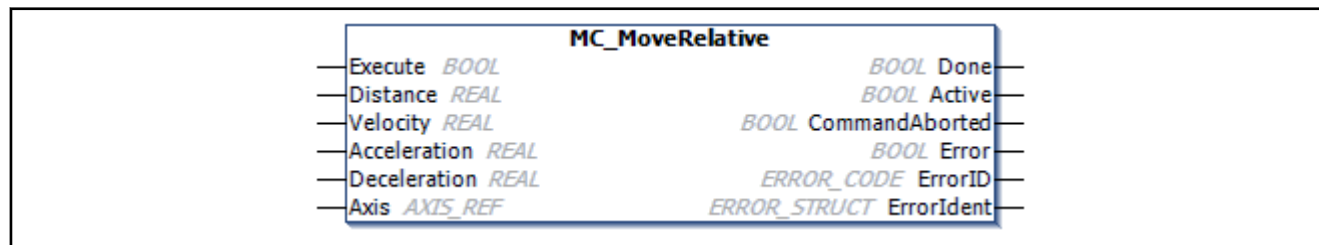


Fig. 10-24: Firmware Function Block "MC_MoveRelative"

Operation mode parameters	PLC mode parameters
S-0-0282, Positioning command value	P-0-1450, PLC Positioning command value
S-0-0259, Positioning velocity	P-0-1451, PLC Positioning velocity
S-0-0260, Positioning acceleration	P-0-1452, PLC/setting-up mode, positioning acceleration
S-0-0346, Positioning control word	P-0-1454, PLC Positioning command value acceptance
S-0-0359, Positioning deceleration	P-0-1453, PLC Positioning deceleration

Tab. 10-48: Comparison of Operation Mode Parameters

Input variable	Data type	Description
Execute	BOOL	See "Signal Time Diagrams"
Distance	REAL	At the "Distance" input you have to preset the relative distance which is to be traveled starting from the current actual position. The number of possible decimal places depends on the parameterized scaling (see drive documentation). If the applied value has more decimal places, they are cut off
Velocity	REAL	Positioning velocity in physical unit
Acceleration	REAL	Positioning acceleration in physical unit
Deceleration	REAL	Positioning deceleration in physical unit

Tab. 10-49: Input Variables of the Firmware Function Block "MC_MoveRelative"

Output variable	Data type	Description
Done	BOOL	With "Done"=TRUE the firmware function block signals that the preset distance has been traveled. The processing is completed until the next 0-1 edge at the "Execute" input. When a 1-0 edge is applied to the "Execute" input before the firmware function block signals that it has been completed ("Done"=TRUE), the "Done" output is set to TRUE for one call when the motion target has been reached and processing of the firmware function block has not been interrupted by an error or another firmware function block. Real axes: The output is set to TRUE when bit 2 is set in parameter "S-0-0437, Positioning status".
Active	BOOL	With TRUE the "Active" output signals that it determines the axis motion.

Output variable	Data type	Description
CommandAborted	BOOL	With "CommandAborted"=TRUE the function block signals that another motion function block was activated or a drive error was present. A drive error is displayed via the Error outputs of "MC_Power"
Error	BOOL	With "Error"=TRUE the function block signals that an error is present
ErrorID *	ERROR_CODE	"ErrorID" gives a rough error description.
ErrorIdent *	ERROR_STRUCT	Via the elements: "Table", "Additional1" and "Additional2", "ErrorIdent" shows the exact cause of the error.

* See note below

Tab. 10-50: Output Variables of the Firmware Function Block "MC_MoveRelative"



NOTE: For the list of all possible errors, see: "Error Reference Lists"

Input/output variable (VAR_IN_OUT)	Data type	Description
Axis	AXIS_REF	Defines the axis. The following axes are possible: - Defines the axis. The following axes are possible: MPB (IndraDrive) and HDC (HydraulicDrive): Axis1 MPC (IndraDrive): - Axis1 (master) or - Axis2...Axis10 (a total of 9 slaves / remote axes) The remote axes contained in the list "P-0-1801.0.10, CCD: Addresses of projected drives" correspond to the order of Axis2 to Axis10 (Axis2 acts on the drive of list element 0 of P-0-1801.0.10,...)

Tab. 10-51: Input/Output Variable of the Firmware Function Block "MC_MoveRelative"

10.3.24 MC_MoveVelocity

The firmware function block "MC_MoveVelocity" is used to preset a velocity for the drive. As the drive moves in velocity control, the firmware function block can be used for open-loop drives, too.

For permanent control and remote axes the primary operation mode and its parameters are used. For temporary control, the secondary PLC operation mode 8 and its parameters are used.



If they are scaling-dependent and unless otherwise described, the units of the inputs which define the drive motion are taken from the scaling parameters.

- S-0-0044, Velocity data scaling type
- S-0-0076, Position data scaling type
- S-0-0086, Torque/force data scaling type

MX_PLCOpen

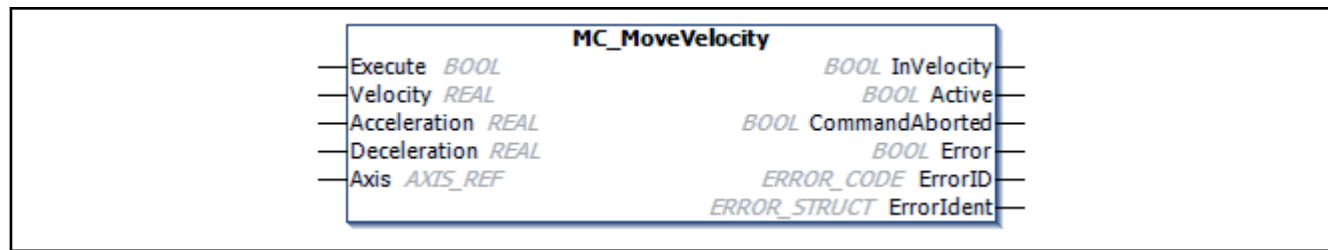


Fig. 10-25: Firmware Function Block "MC_MoveVelocity"

Operation mode parameters	PLC mode parameters
S-0-0036, Velocity command value	P-0-1460, PLC Velocity command value
P-0-1203, Ramp 2 pitch	P-0-1461, PLC Ramp 2 pitch
P-0-1213, Deceleration ramp 2	P-0-1463, PLC Deceleration ramp 2

Tab. 10-52: Comparison of Operation Mode Parameters

Input variable	Data type	Description
Execute	BOOL	See "Signal Time Diagrams"
Velocity	REAL	Velocity in physical unit. The number of possible decimal places depends on the parameterized scaling (see drive documentation). If the applied value has more decimal places, they are cut off.
Acceleration	REAL	Acceleration in physical unit. The number of possible decimal places depends on the parameterized scaling (see drive documentation). If the applied value has more decimal places, they are cut off.
Deceleration	REAL	Deceleration in physical unit. The number of possible decimal places depends on the parameterized scaling (see drive documentation). If the applied value has more decimal places, they are cut off.

Tab. 10-53: Input Variables of the Firmware Function Block "MC_MoveVelocity"

Output variable	Data type	Description
InVelocity	BOOL	With "InVelocity"=TRUE the firmware function block signals that the velocity preset by "Velocity" has been reached. When a 1-0 edge is applied to the "Execute" input before the firmware function block signals that it has reached the command velocity ("InVelocity"=TRUE), the "InVelocity" output is set to TRUE for one call if the actual velocity corresponds to the command velocity. Real axes: "Axis1" etc. InVelocity = "P-0-0115, Device control: Status word", bit 12 of the parameterized axis. In this application, bit 12 of P-0-0115 corresponds to "S-0-0330, Status "n_feedback = n_command"".
Active	BOOL	With "Active"=TRUE the firmware function block signals that it determines the axis motion. "Active" remains TRUE until "CommandAborted" or "Error" is set.
CommandAborted	BOOL	With "CommandAborted"=TRUE the function block signals that another motion function block was activated or a drive error was present. A drive error is displayed via the Error outputs of "MC_Power"

Output variable	Data type	Description
Error	BOOL	With "Error"=TRUE the function block signals that an error is present
ErrorID *	ERROR_CODE	"ErrorID" gives a rough error description.
ErrorIdent *	ERROR_STRUCT	Via the elements: "Table", "Additional1" and "Additional2", "ErrorIdent" shows the exact cause of the error.

* See note below

Tab. 10-54: Output Variables of the Firmware Function Block "MC_MoveVelocity"



NOTE: For the list of all possible errors, see: "Error Reference Lists"

Input/output variable (VAR_IN_OUT)	Data type	Description
Axis	AXIS_REF	Defines the axis. The following axes are possible: - Defines the axis. The following axes are possible: MPB (IndraDrive) and HDC (HydraulicDrive): Axis1 MPC (IndraDrive): - Axis1 (master) or - Axis2...Axis10 (a total of 9 slaves / remote axes) The remote axes contained in the list "P-0-1801.0.10, CCD: Addresses of projected drives" correspond to the order of Axis2 to Axis10 (Axis2 acts on the drive of list element 0 of P-0-1801.0.10,...)

Tab. 10-55: Input/Output Variable of the Firmware Function Block "MC_MoveVelocity"

10.3.25 MC_Power

When the internal PLC has permanent or temporary control over the drive, the drive enable can be activated with this function block. Independent of the "Enable" input, the "Status" output returns the current drive status.

The function block "MC_Power" sets the PLC-internal signal "ON" to TRUE and thereby switches the drive on.



The "ON" signal is the "ON" signal of the PLC that becomes active, linked to the external signal, in the drive with a possible delay; when several edges are generated (e.g. "TRUE - FALSE - TRUE"), this delay time must be taken into account.

When a WATCHDOG or an exception occurs in the PLC, the PLC-internal "ON" signal is automatically set to FALSE and maintained FALSE until the next RESET of the PLC.

Further calls of "MC_Power" with TRUE are signaling an error.

If the drive is not yet ready for power on (e.g. not in "Ab"), the "Status" output provides FALSE and the "Error" output is TRUE.

MX_PLCOpen

 With the query IF AxisData[Axis1.AxisNo].Axis_InAb THEN, for example, you can check whether or not the drive is ready for power on; for this query, the option for supporting the AxisData structure must have been activated in the PLC configuration (P-0-1367).

 "MC_Power" also provides an error, when a motion function block has detected an error and shut down the axis.

After every error ("Error"=TRUE), a new 0-1 edge has to be applied to the "Enable" input to activate drive enable in the drive.

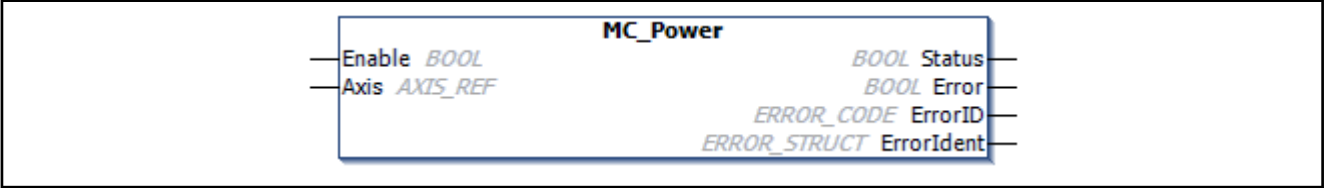


Fig. 10-26: Firmware Function Block "MC_Power"

Input variable	Data type	Description
Enable	BOOL	Via the "Enable" input, the drive enable can be set for the drive. For reasons of safety, the input takes effect in edge-controlled form. Drive enable is set with a rising edge and with a falling edge it is removed again.

Tab. 10-56: Input Variables of the Firmware Function Block "MC_Power"

Output variable	Data type	Description
Status	BOOL	When the output "Status"=TRUE, the drive enable is active and the drive is under torque. Independent of the "Enable" input, the "Status" output returns the current drive status. For this purpose, the combination "drive in control and no error" is internally checked. Output "Status":= (P115.14 AND P115.15 AND NOT P115.13) The "Status" output therefore goes to FALSE in the case of drive errors
Error	BOOL	With "Error"=TRUE the function block signals that an error is present
ErrorID *	ERROR_CODE	"ErrorID" gives a rough error description.
ErrorIdent *	ERROR_STRUCT	Via the elements: "Table", "Additional1" and "Additional2", "ErrorIdent" shows the exact cause of the error.

* See note below
Tab. 10-57: Output Variables of the Firmware Function Block "MC_Power"

 NOTE: For the list of all possible errors, see: "Error Reference Lists"

Input/output variable (VAR_IN_OUT)	Data type	Description
Axis	AXIS_REF	Defines the axis. The following axes are possible: MPB (IndraDrive) and HDC (HydraulicDrive): Axis1 MPC (IndraDrive): - Axis1 (master) or - Axis2...Axis10 (a total of 9 slaves / remote axes) The remote axes contained in the list "P-0-1801.0.10, CCD: Addresses of projected drives" correspond to the order of Axis2 to Axis10 (Axis2 acts on the drive of list element 0 of P-0-1801.0.10,...)

Tab. 10-58: Input/Output Variable of the Firmware Function Block "MC_Power"

10.3.26 MC_ReadStatus

With the function block "MC_ReadStatus" it is possible to read the current status of the axis. The status of the axis is output in decoded form.



Any change of the axis at the input of the function block requires a repeated 0/1 transition at "Enable".

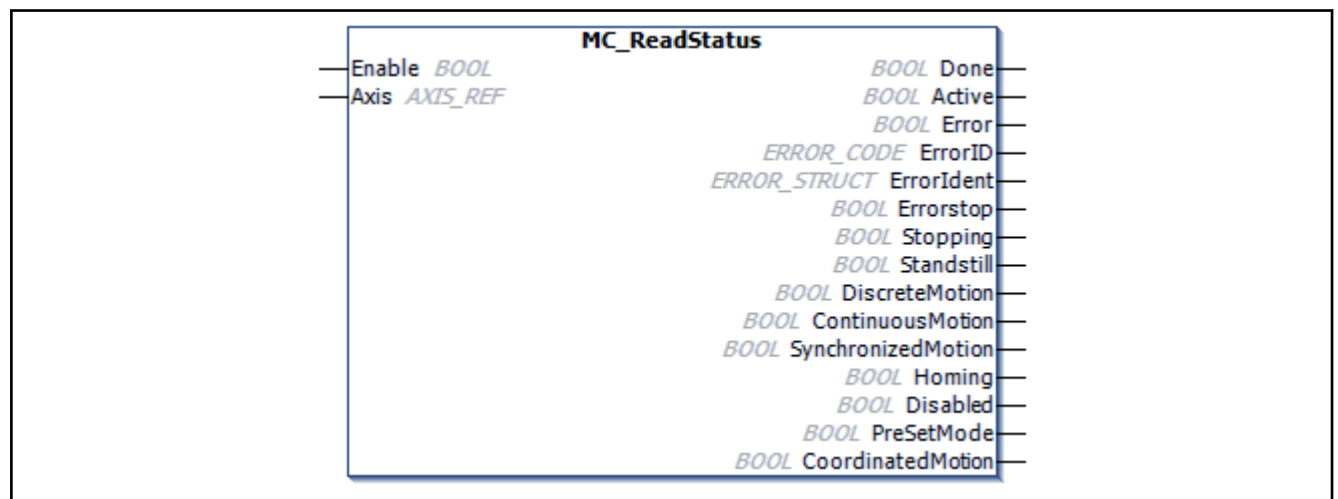


Fig. 10-27: Firmware Function Block "MC_ReadStatus"

Input variable	Data type	Description
Enable	BOOL	See "Signal Time Diagrams"

Tab. 10-59: Input Variable of the Firmware Function Block "MC_ReadStatus"

Output variable	Data type	Description
Done	BOOL	Valid value is available
Active	BOOL	Processing of the data is running after completion of pre-processing
Error	BOOL	With "Error"=TRUE the function block signals that an error is present
ErrorID *	ERROR_CODE	"ErrorID" gives a rough error description.
ErrorIdent *	ERROR_STRUCT	Via the elements: "Table", "Additional1" and "Additional2", "ErrorIdent" shows the exact cause of the error.
Errorstop	BOOL	Drive/axis in status "Errorstop"

MX_PLCOpen

Output variable	Data type	Description
Stopping	BOOL	Drive/axis in status "Stopping"
StandStill	BOOL	Drive/axis in status "StandStill"
DiscreteMotion	BOOL	Drive/axis in status "DiscreteMotion"
ContinuousMotion	BOOL	Drive/axis in status "ContinuousMotion"
SynchronizedMotion	BOOL	Drive/axis in status "SynchronizedMotion" (not with HydraulicDrive)
Homing	BOOL	Drive in status "Homing"
Disabled	BOOL	Drive/axis in status "Disabled"
PreSetMode	BOOL	Drive in status "PreSetMode"
CoordinatedMotion	BOOL	Drive/axis in status "CoordinatedMotion"; not supported at present

* See note below
Tab. 10-60: Output Variables of the Firmware Function Block "MC_ReadStatus"



NOTE: For the list of all possible errors, see: "Error Reference Lists"

Input/output variable (VAR_IN_OUT)	Data type	Description
Axis	AXIS_REF	Defines the axis. The following axes are possible: MPB (IndraDrive) and HDC (HydraulicDrive): Axis1 MPC (IndraDrive): - Axis1 (master) or - Axis2...Axis10 (a total of 9 slaves / remote axes) The remote axes contained in the list "P-0-1801.0.10, CCD: Addresses of projected drives" correspond to the order of Axis2 to Axis10 (Axis2 acts on the drive of list element 0 of P-0-1801.0.10,...)

Tab. 10-61: Input/Output Variables of the Firmware Function Block "MC_ReadStatus"

10.3.27 MC_Reset

The function block "MC_Reset" is used to reset drive errors with the command "S-0-0099, C0500 Reset class 1 diagnostics"; drive enable remains unchanged.

In order that the function block can correctly complete the reset command, it has to be cyclically called until it signals "Done"=TRUE or "Error" in the case of error.



If the command "S-0-0099, C0500 Reset class 1 diagnostics" is already active, it should first be completed before "MC_Reset" is called. If "MC_Reset" had been activated and has not yet been completed, it is possible to trigger another reset with a rising edge at "Execute".

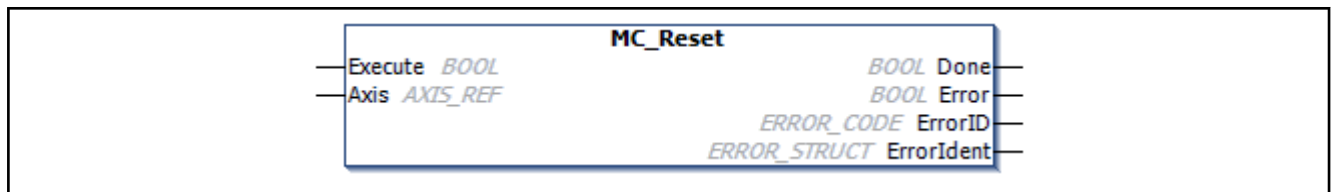


Fig. 10-28: Firmware Function Block "MC_Reset"

Input variable	Data type	Description
Execute	BOOL	See "Signal Time Diagrams"

Tab. 10-62: Input Variables of the Firmware Function Block "MC_Reset"

Output variable	Data type	Description
Done	BOOL	With "Done"=TRUE the function block signals that processing of the task has been successfully completed
Error	BOOL	With "Error"=TRUE the function block signals that an error is present
ErrorID *	ERROR_CODE	"ErrorID" gives a rough error description.
ErrorIdent *	ERROR_STRUCT	Via the elements: "Table", "Additional1" and "Additional2", "ErrorIdent" shows the exact cause of the error.

* See note below

Tab. 10-63: Output Variables of the Firmware Function Block "MC_Reset"



NOTE: For the list of all possible errors, see: "Error Reference Lists"

Input/output variable (VAR_IN_OUT)	Data type	Description
Axis	AXIS_REF	Defines the axis. The following axes are possible: MPB (IndraDrive) and HDC (HydraulicDrive): Axis1 MPC (IndraDrive): - Axis1 (master) or - Axis2...Axis10 (a total of 9 slaves / remote axes) The remote axes contained in the list "P-0-1801.0.10, CCD: Addresses of projected drives" correspond to the order of Axis2 to Axis10 (Axis2 acts on the drive of list element 0 of P-0-1801.0.10,...)

Tab. 10-64: Input/Output Variable of the Firmware Function Block "MC_Reset"

10.3.28 MC_Stop

The firmware function block "MC_Stop" is used to bring the drive to the status STOP.

With the rising edge of "Execute" the stopping process is started.

When the firmware function block "MC_Stop" is activated via a rising edge at the "Execute" input, the drive goes to the status "AH" (Drive Halt). Further command triggering of a motion function block will only be possible when the axis is in standstill and the "Execute" input has the value FALSE.

With the deceleration set at "Deceleration" the drive is brought to a standstill.

MX_PLCOpen

When several instances are called by "MC_Stop" in a project, it is always the last instance which takes effect. An active "MC_Stop" cannot be interrupted by an "MB_Stop".

When an instance is called with an edge at "Execute", other instances signal CommandAborted = TRUE when called (without edge).

"MC_Stop" has higher priority than "MB_Stop"; this means that a running "MC_Stop" cannot be aborted by an "MB_Stop".



To test the function block "MC_Stop", the instance variable "ActivateCounter" can be used. It counts the positive edges of the "Execute" signal and thereby signals whether the function block is run with an edge.

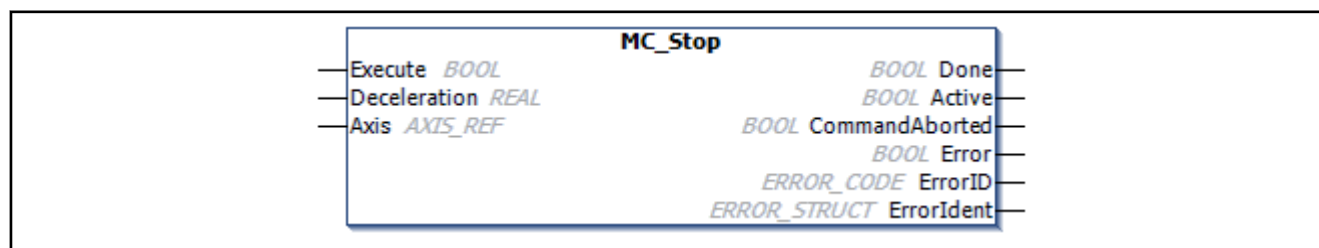


Fig. 10-29: Firmware Function Block "MC_Stop"

Input variable	Data type	Description
Execute	BOOL	With a rising edge at "Execute" the drive is switched to "AH".
Deceleration	REAL	This input sets the deceleration. The set deceleration is written to parameter "S-0-0372, Drive Halt acceleration bipolar" of the following axis (real axis).

Tab. 10-65: Input Variables of the Firmware Function Block "MC_Stop"

Output variable	Data type	Description
Done	BOOL	With "Done"=TRUE the firmware function block signals that the axis has stopped. (S-0-0331, Bit0=1)
Active	BOOL	With "Active"=TRUE the firmware function block signals that it determines the axis motion
CommandAborted	BOOL	With "CommandAborted"=TRUE the function block signals that another "MC_Stop" was activated or a drive error was present. A drive error is displayed via the Error outputs of "MC_Power"
Error	BOOL	With "Error"=TRUE the function block signals that an error is present
ErrorID *	ERROR_CODE	"ErrorID" gives a rough error description.
ErrorIdent *	ERROR_STRUCT	Via the elements: "Table", "Additional1" and "Additional2", "ErrorIdent" shows the exact cause of the error.

* See note below

Tab. 10-66: Output Variables of the Firmware Function Block "MC_Stop"



NOTE: For the list of all possible errors, see: "Error Reference Lists"

Input/output variable (VAR_IN_OUT)	Data type	Description
Axis	AXIS_REF	Defines the axis. The following axes are possible: - Defines the axis. The following axes are possible: MPB (IndraDrive) and HDC (HydraulicDrive): Axis1 MPC (IndraDrive): - Axis1 (master) or - Axis2...Axis10 (a total of 9 slaves / remote axes) The remote axes contained in the list "P-0-1801.0.10, CCD: Addresses of projected drives" correspond to the order of Axis2 to Axis10 (Axis2 acts on the drive of list element 0 of P-0-1801.0.10,...)

Tab. 10-67: Input/Output Variable of the Firmware Function Block "MC_Stop"

10.3.29 MC_TorqueControl

The firmware function block "MC_TorqueControl" is used to activate the operation mode "torque/force control"; the drive follows the command value input.

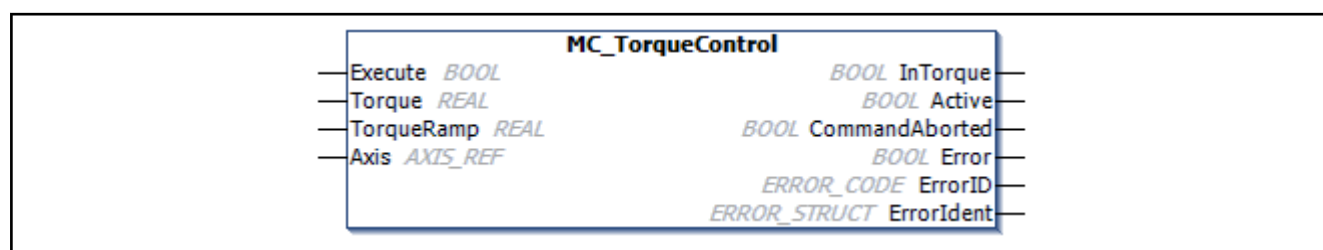


Fig. 10-30: Firmware Function Block "MC_TorqueControl"

Operation mode parameters	PLC mode parameters
S-0-0080, Torque/force command value	P-0-1465, PLC Torque/force command value
S-0-0822, Torque/force ramp	P-0-1466, PLC Torque/force ramp

Tab. 10-68: Comparison of Operation Mode Parameters

Input variable	Data type	Description
Execute	BOOL	See "Signal Time Diagrams"
Torque	REAL	Torque/force command value (S-0-0080)
TorqueRamp	REAL	Torque/force ramp (S-0-0822) with reference to the torque/force ramp time (S-0-0823) Note: If the "TorqueRamp" input is set to "0", the torque/force command value is input abruptly.

Tab. 10-69: Input Variables of the Firmware Function Block "MC_TorqueControl"

Output variable	Data type	Description
InTorque	BOOL	Torque/force command value has been reached. Note: The parameter "S-0-0826, Torque/force window" is used to set when the function block provides the "InTorque" message.
Active	BOOL	Function block or operation mode is active
CommandAborted	BOOL	Function block was aborted

MX_PLCOpen

Output variable	Data type	Description
Error	BOOL	With "Error"=TRUE the function block signals that an error is present
ErrorID *	ERROR_CODE	"ErrorID" gives a rough error description.
ErrorIdent *	ERROR_STRUCT	Via the elements: "Table", "Additional1" and "Additional2", "ErrorIdent" shows the exact cause of the error.

* See note below
 Tab. 10-70: *Output Variables of the Firmware Function Block "MC_TorqueControl"*



NOTE: For the list of all possible errors, see: "Error Reference Lists"

Input/output variable (VAR_IN_OUT)	Data type	Description
Slave	AXIS_REF	Defines the axis. The following axes are possible: MPB (IndraDrive) and HDC (HydraulicDrive): Axis1 MPC (IndraDrive): - Axis1 (master) or - Axis2...Axis10 (a total of 9 slaves / remote axes) The remote axes contained in the list "P-0-1801.0.10, CCD: Addresses of projected drives" correspond to the order of Axis2 to Axis10 (Axis2 acts on the drive of list element 0 of P-0-1801.0.10,...)

Tab. 10-71: *Input/Output Variables of the Firmware Function Block "MC_TorqueControl"*

10.3.30 MX_MoveAbsolute

The firmware function block "MX_MoveAbsolute" is used to move the drive to a preset absolute position. For this purpose, the secondary PLC operation mode "drive-controlled positioning" is internally activated (see Functional Description of firmware "Drive-Controlled Positioning"). Make the settings of the position controller (with/without lag error, actual position value 1/2) with the function block "MB_SetPositionControlMode".

For permanent control and remote axes, the secondary operation mode 3 and its parameters are used. For temporary control, the secondary PLC operation mode 11 and its parameters are used.

The target position must be set at the "Position" input. Via the inputs "Velocity", "Acceleration" and "Deceleration" you have to set how the drive moves to the target position (motion profile).

Operation mode parameters	PLC mode parameters
S-0-0282, Positioning command value	P-0-1450, PLC Positioning command value
S-0-0259, Positioning velocity	P-0-1451, PLC Positioning velocity
S-0-0260, Positioning acceleration	P-0-1452, PLC/setting-up mode, positioning acceleration
S-0-0346, Positioning control word	P-0-1454, PLC Positioning command value acceptance
S-0-0359, Positioning deceleration	P-0-1453, PLC Positioning deceleration

Tab. 10-72: *Comparison of Operation Mode Parameters*

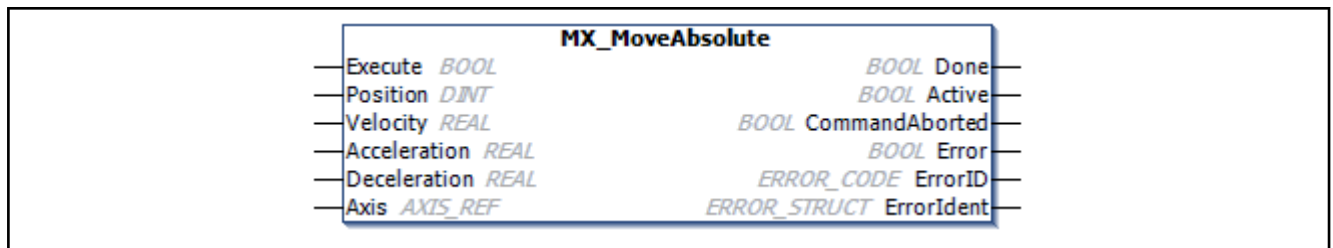


Fig. 10-31: Firmware Function Block "MX_MoveAbsolute"

Input variable	Data type	Description
Execute	BOOL	See "Signal Time Diagrams"
Position	DINT	At "Position" you have to preset the target position in fixed point format. To preset the position correctly, the number of decimal places of parameter "S-0-0282, Positioning command value" must be known. With preferred scaling (see Functional Description of firmware "Scaling Types"), the parameter has 4 decimal places. Example: If the drive is to be moved to a target position of 100.25°, the value 1002500 must be written at "Position".
Velocity	REAL	Positioning velocity in physical unit
Acceleration	REAL	Positioning acceleration in physical unit
Deceleration	REAL	Positioning deceleration in physical unit

Tab. 10-73: Input Variables of the Firmware Function Block "MX_MoveAbsolute"

Output variable	Data type	Description
Done	BOOL	With "Done"=TRUE the firmware function block signals that the absolute target position has been reached. The processing is completed until the next 0-1 edge at the "Execute" input. When a 1-0 edge is applied to the "Execute" input before the firmware function block signals that it has been completed ("Done"=TRUE), the "Done" output is set to TRUE for one call when the motion target has been reached and processing of the firmware function block has not been interrupted by an error or another firmware function block. Real axis: The output is set to TRUE when bit 2 is set in parameter "S-0-0437, Positioning status".
Active	BOOL	With "Active"=TRUE the firmware function block signals that it determines the axis motion. "Active" remains TRUE until "CommandAborted" or "Error" is set.
CommandAborted	BOOL	With "CommandAborted"=TRUE the function block signals that another motion function block was activated or a drive error was present. A drive error is displayed via the Error outputs of "MC_Power"
Error	BOOL	With "Error"=TRUE the function block signals that an error is present
ErrorID *	ERROR_CODE	"ErrorID" gives a rough error description.
ErrorIdent *	ERROR_STRUCT	Via the elements: "Table", "Additional1" and "Additional2", "ErrorIdent" shows the exact cause of the error.

* See note below

Tab. 10-74: Output Variables of the Firmware Function Block "MX_MoveAbsolute"

MX_PLCOpen



NOTE: For the list of all possible errors, see: "Error Reference Lists"

Input/output variable (VAR_IN_OUT)	Data type	Description
Axis	AXIS_REF	Defines the axis. The following axes are possible: - Defines the axis. The following axes are possible: MPB (IndraDrive) and HDC (HydraulicDrive): Axis1 MPC (IndraDrive): - Axis1 (master) or - Axis2...Axis10 (a total of 9 slaves / remote axes) The remote axes contained in the list "P-0-1801.0.10, CCD: Addresses of projected drives" correspond to the order of Axis2 to Axis10 (Axis2 acts on the drive of list element 0 of P-0-1801.0.10,...)

Tab. 10-75: Input/Output Variable of the Firmware Function Block "MX_MoveAbsolute"

10.3.31 MX_MoveAdditive

The firmware function block "MX_MoveAdditive" is used to move the drive in relative form by a distance in addition to the target position. The new target position results from the old target position, plus a distance. The distance is determined by the "Distance" input. The inputs "Velocity", "Acceleration" and "Deceleration" determine the motion profile.

For this purpose, the secondary PLC operation mode "drive-controlled positioning" is internally activated (see Functional Description of firmware "Drive-Controlled Positioning"). Make the settings of the position controller (with/without lag error, actual position value 1/2) with the function block "MB_SetPositionControlMode".

For permanent control and remote axes, the secondary operation mode 3 and its parameters are used. For temporary control, the secondary PLC operation mode 11 and its parameters are used.

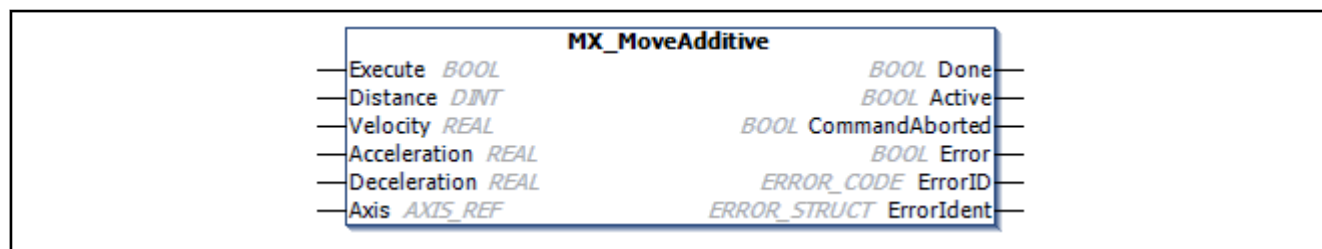


Fig. 10-32: Firmware Function Block "MX_MoveAdditive"

The corresponding PLC mode parameters are used as internal command values instead of the usual operation mode parameters. They can be read for diagnostic purposes:

Operation mode parameters	PLC mode parameters
S-0-0282, Positioning command value	P-0-1450, PLC Positioning command value
S-0-0259, Positioning velocity	P-0-1451, PLC Positioning velocity
S-0-0260, Positioning acceleration	P-0-1452, PLC/setting-up mode, positioning acceleration

S-0-0346, Positioning control word	P-0-1454, PLC Positioning command value acceptance
S-0-0359, Positioning deceleration	P-0-1453, PLC Positioning deceleration

Tab. 10-76: Comparison of Operation Mode Parameters

Input variable	Data type	Description
Execute	BOOL	See "Signal Time Diagrams"
Distance	DINT	At "Distance" you have to preset the relative distance in fixed point format. To preset the distance correctly, the number of decimal places of parameter "S-0-0282, Positioning command value" must be known. With preferred scaling (see Functional Description of firmware "Scaling Types"), the parameter has 4 decimal places. Example: If the drive is to be moved by a distance of 100.25°, the value 1002500 must be written to "Distance"
Velocity	REAL	Positioning velocity in physical unit
Acceleration	REAL	Positioning acceleration in physical unit
Deceleration	REAL	Positioning deceleration in physical unit

Tab. 10-77: Input Variables of the Firmware Function Block "MX_MoveAdditive"

Output variable	Data type	Description
Done	BOOL	With "Done"=TRUE the firmware function block signals that the target position has been reached. The processing is completed until the next 0-1 edge at the "Execute" input. When a 1-0 edge is applied to the "Execute" input before the firmware function block signals that it has been completed ("Done"=TRUE), the "Done" output is set to TRUE for one call when the motion target has been reached and processing of the firmware function block has not been interrupted by an error or another firmware function block. Real axes: The output is set to TRUE when bit 2 is set in parameter "S-0-0437, Positioning status".
Active	BOOL	With "Active"=TRUE the firmware function block signals that it determines the axis motion.
CommandAborted	BOOL	With "CommandAborted"=TRUE the firmware function block signals that another motion function block was activated or a drive error was present. A drive error is displayed via the Error outputs of "MC_Power".
Error	BOOL	With "Error"=TRUE the function block signals that an error is present
ErrorID *	ERROR_CODE	"ErrorID" gives a rough error description.
ErrorIdent *	ERROR_STRUCT	Via the elements: "Table", "Additional1" and "Additional2", "ErrorIdent" shows the exact cause of the error.

* See note below

Tab. 10-78: Output Variables of the Firmware Function Block "MX_MoveAdditive"



NOTE: For the list of all possible errors, see: "Error Reference Lists"

MX_PLCOpen

Input/output variable (VAR_IN_OUT)	Data type	Description
Axis	AXIS_REF	Defines the axis. The following axes are possible: - Defines the axis. The following axes are possible: MPB (IndraDrive) and HDC (HydraulicDrive): Axis1 MPC (IndraDrive): - Axis1 (master) or - Axis2...Axis10 (a total of 9 slaves / remote axes) The remote axes contained in the list "P-0-1801.0.10, CCD: Addresses of projected drives" correspond to the order of Axis2 to Axis10 (Axis2 acts on the drive of list element 0 of P-0-1801.0.10,...)

Tab. 10-79: *Input / Output Variable of the Firmware Function Block "MX_MoveAdditive"*

10.3.32 MX_MoveRelative

The firmware function block "MX_MoveRelative" is used to move the drive in relative form by a distance, starting from the current actual position. For this purpose, the secondary PLC operation mode "drive-controlled positioning" is internally activated (see Functional Description of firmware "Drive-Controlled Positioning").



Internally the operation mode "drive-controlled positioning" is activated. Make the settings of the position controller (with/without lag error, actual position value 1/2) with the function block "MB_SetPositionControlMode".

For permanent control and remote axes, the secondary operation mode 3 and its parameters are used. For temporary control, the secondary PLC operation mode 11 and its parameters are used.

The relative distance to be moved has to be preset at the "Distance" input. Via the inputs "Velocity", "Acceleration" and "Deceleration" you have to set the motion profile (determines how the relative distance is traveled).

The corresponding PLC mode parameters are used as internal command values instead of the usual operation mode parameters. They can be read for diagnostic purposes:

Operation mode parameters	PLC mode parameters
S-0-0282, Positioning command value	P-0-1450, PLC Positioning command value
S-0-0259, Positioning velocity	P-0-1451, PLC Positioning velocity
S-0-0260, Positioning acceleration	P-0-1452, PLC/setting-up mode, positioning acceleration
S-0-0346, Positioning control word	P-0-1454, PLC Positioning command value acceptance
S-0-0359, Positioning deceleration	P-0-1453, PLC Positioning deceleration

Tab. 10-80: *Comparison of Operation Mode Parameters*

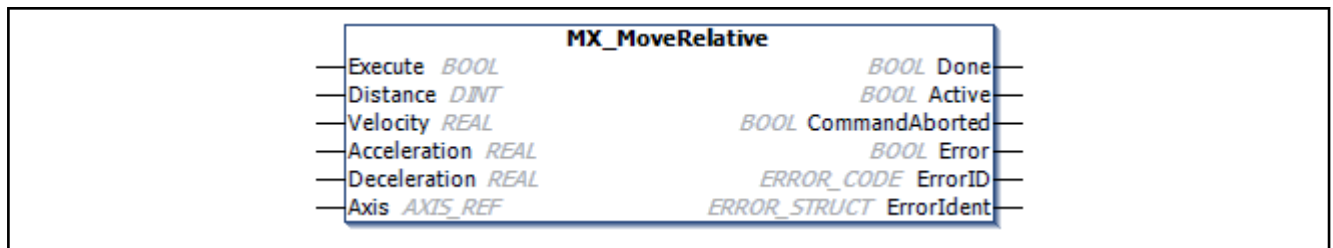


Fig. 10-33: Firmware Function Block "MX_MoveRelative"

Input variable	Data type	Description
Execute	BOOL	See "Signal Time Diagrams"
Distance	DINT	At "Distance" you have to preset the relative distance to be traveled in fixed point format. To preset the distance correctly, the number of decimal places of parameter "S-0-0282, Positioning command value" must be known. With preferred scaling (see Functional Description of firmware "Scaling Types"), the parameter has 4 decimal places. Example: If the drive is to be moved by a distance of 100.25° starting from the current actual position, the value 1002500 must be written to "Distance".
Velocity	REAL	Positioning velocity in physical unit
Acceleration	REAL	Positioning acceleration in physical unit
Deceleration	REAL	Positioning deceleration in physical unit

Tab. 10-81: Input Variables of the Firmware Function Block "MX_MoveRelative"

Output variable	Data type	Description
Done	BOOL	With "Done"=TRUE the firmware function block signals that the preset distance has been traveled. The processing is completed until the next 0-1 edge at the "Execute" input. When a 1-0 edge is applied to the "Execute" input before the firmware function block signals that it has been completed ("Done"=TRUE), the "Done" output is set to TRUE for one call when the motion target has been reached and processing of the firmware function block has not been interrupted by an error or another firmware function block. Real axes: The output is set to TRUE when bit 2 is set in parameter "S-0-0437, Positioning status".
Active	BOOL	With TRUE the "Active" output signals that it determines the axis motion.
CommandAborted	BOOL	With "CommandAborted"=TRUE the function block signals that another motion function block was activated or a drive error was present. A drive error is displayed via the Error outputs of "MC_Power"
Error *	BOOL	With "Error"=TRUE the function block signals that an error is present
ErrorID *	ERROR_CODE	"ErrorID" gives a rough error description.
ErrorIdent *	ERROR_STRUCT	Via the elements: "Table", "Additional1" and "Additional2", "ErrorIdent" shows the exact cause of the error.

* See note below

Tab. 10-82: Output Variables of the Firmware Function Block "MX_MoveRelative"



NOTE: For the list of all possible errors, see: "Error Reference Lists"

MX_PLCOpen

Input/output variable (VAR_IN_OUT)	Data type	Description
Axis	AXIS_REF	Defines the axis. The following axes are possible: - Defines the axis. The following axes are possible: MPB (IndraDrive) and HDC (HydraulicDrive): Axis1 MPC (IndraDrive): - Axis1 (master) or - Axis2...Axis10 (a total of 9 slaves / remote axes) The remote axes contained in the list "P-0-1801.0.10, CCD: Addresses of projected drives" correspond to the order of Axis2 to Axis10 (Axis2 acts on the drive of list element 0 of P-0-1801.0.10,...)

Tab. 10-83: *Input/Output Variable of the Firmware Function Block "MX_MoveRelative"*

10.3.33 MX_SetControl

The function block "MX_SetControl" is used to switch control between master communication and internal PLC; this makes sense when the axis is first controlled via a master communication and the internal PLC is to continue controlling the axis in a certain situation (e.g. error).



When the axis generally is to be moved with the internal PLC, permanent control should be activated.

With "PLCControl"=TRUE the internal PLC gets control over the drive, with "PLCControl"=FALSE the external master communication gets control over the drive.

The signals "Drive Halt" (AH), "Drive on" (ON) and the operation mode selection are switched. With internal PLC control, the signals "AH" and "ON" are only operated internally. The "Drive Halt" signal is controlled by the function block "MB_Stop" or by stopping the PLC. The "ON" signal can be controlled with the function block "MC_Power".

If the corresponding function blocks are not used, the internal signals are set to TRUE.

When switching to temporary control, the internal "ON" signal is set to TRUE. This results in the following behavior when temporary control was got with "MX_SetControl":

- Switching off:** With the firmware function block "MC_Power".
 In order that the firmware function block can detect a TRUE/FALSE transition at its Enable input, it must be called at least once with Enable=TRUE.
- Switching on:** With the firmware function block "MC_Power".
 In order that the firmware function block can detect a FALSE/TRUE transition at its Enable input, it must be called at least once with "Enable"=FALSE. This measure is not necessary when switching from the parameter mode to the operating mode takes place between the call of "MX_SetControl" and "MC_Power".

MX_PLCOpen



The function block "MX_SetControl" supplies error if permanent control is already active (setting via "P-0-1367, PLC configuration").

While switching is running ("Done" and "Error" not yet TRUE), another switching process can be started with an "Execute" edge.

In the case of program download (no online change) or PLC reset, temporary control is automatically returned again.



To be observed for PLCopen Motion function blocks

- With **temporary** control, the secondary PLC operation modes are used.
- With **permanent** control, the primary mode of operation and the secondary operation modes 1-7 are parameterized and used.



Fig. 10-34: Firmware Function Block "MX_SetControl"

Input variable	Data type	Description
Execute	BOOL	The function block is activated with a rising edge at "Execute". All outputs, except for the "PLCControlStat" output, are reset with a falling edge at "Execute". Independent of "Execute", "PLCControlStat" provides the actual value.
PLCControl	BOOL	With "PLCControl"=TRUE the internal PLC gets temporary control over the drive at the next 0-1 edge at "Execute". With "PLCControl"=FALSE control is returned to the master communication at the next 0-1 edge at "Execute".

Tab. 10-84: Input Variables of the Firmware Function Block "MX_SetControl"

Output variable	Data type	Description
Done	BOOL	With "Done"=TRUE the function block signals that processing of the task has been successfully completed
Error	BOOL	With "Error"=TRUE the function block signals that an error is present
ErrorID *	ERROR_CODE	"ErrorID" gives a rough error description.
ErrorIdent *	ERROR_STRUCT	Via the elements: "Table", "Additional1" and "Additional2", "ErrorIdent" shows the exact cause of the error.
PLCControlStat	BOOL	With "PLCControlStat"=TRUE the drive-internal PLC has control over the drive. With "PLCControlStat"=FALSE the master communication has control over the drive. Independent of "Execute", "PLCControlStat" provides the actual value.

* See note below

Tab. 10-85: Output Variables of the Firmware Function Block "MX_SetControl"

MX_PLCOpen



NOTE: For the list of all possible errors, see: "Error Reference Lists"

Input/output variable (VAR_IN_OUT)	Data type	Description
Axis	AXIS_REF	Axis number (only the global variable "Axis1" can be transmitted here)

Tab. 10-86: Input/Output Variable of the Firmware Function Block "MX_SetControl"

10.3.34 MX_SetDeviceMode

The function block "MX_SetDeviceMode" is used to switch the drive to the parameter or operating mode.

If switching is not allowed ("P-0-4086, Master communication status", bit 0="0"), the function block returns an error via "Error".

A mode is set by a rising edge of "Execute". Independent of "Execute", the current status is displayed in the "ActOperationMode" output.



During the switching process, "MX_SetDeviceMode" should be called until the "Done" output or the "Error" output becomes TRUE. Otherwise, direct switching might not be possible again, as "MX_SetDeviceMode" then possibly is not in its initial status.

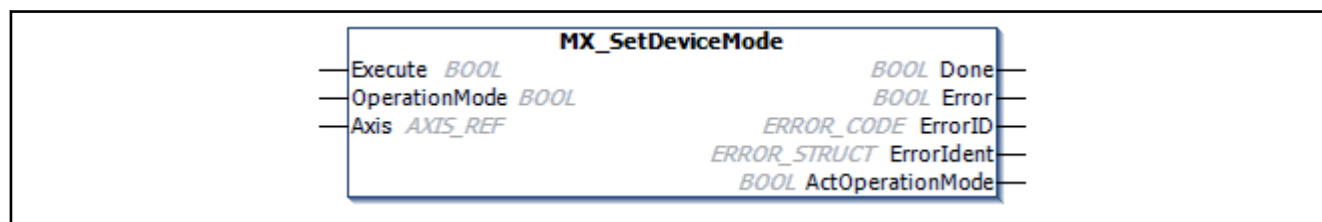


Fig. 10-35: Firmware Function Block "MX_SetDeviceMode"

Input variable	Data type	Description
Execute	BOOL	The function block is activated with a rising edge at "Execute". All outputs, except for the "ActOperationMode" output, are reset with a falling edge at "Execute". Independent of "Execute", the current communication phase is displayed at "ActOperationMode".
OperationMode	BOOL	With "OperationMode"=TRUE the drive is switched to operating mode (phase 4) with the next positive edge at "Execute"; for this purpose, the function block starts the drive command C0200 (S-0-0422). With "OperationMode"=FALSE the drive is switched to parameter mode (phase 2) with the next positive edge at "Execute"; for this purpose, the function block starts the drive command C0400 (S-0-0420).

Tab. 10-87: Input Variables of the Firmware Function Block "MX_SetDeviceMode"

Output variable	Data type	Description
Done	BOOL	With "Done"=TRUE the function block signals that processing of the task has been successfully completed
Error	BOOL	With "Error"=TRUE the function block signals that an error is present
ErrorID *	ERROR_CODE	"ErrorID" gives a rough error description.

Output variable	Data type	Description
ErrorIdent *	ERROR_STRUCT	Via the elements: "Table", "Additional1" and "Additional2", "ErrorIdent" shows the exact cause of the error.
ActOperationMode	BOOL	With "ActOperationMode"=TRUE the drive is in operating mode (phase 4). With "ActOperationMode"=FALSE the drive is in parameter mode (2<=current phase 4). Independent of "Execute", the current communication phase is displayed at "ActOperationMode".

* See note below

Tab. 10-88: Output Variables of the Firmware Function Block "MX_SetDevice-Mode"



NOTE: For the list of all possible errors, see: "Error Reference Lists"

Input/output variable (VAR_IN_OUT)	Data type	Description
Axis	AXIS_REF	Axis number (only the global variable "Axis1" can be transmitted here)

Tab. 10-89: Input/Output Variable of the Firmware Function Block "MX_SetDevice-Mode"

10.3.35 MX_SetOpMode

The function block "MX_SetOpMode" is used to directly switch the operation mode of the drive over.

The current operation mode is provided independently of "Execute". Here, the primary operation mode corresponds to value "0", the secondary operation modes correspond to values "1" to "7". The internal secondary operation modes are not permitted.



"MX_SetOpMode" is provided for direct control of the drive via its operation modes and their parameters. The operation mode that was set is activated in the following position controller cycle.

If, after the drive was switched over, it is operated using, for example, an "MX_MoveRelative", the corresponding operation mode is set implicitly.

During the switching process, "MX_SetOpMode" should be called until the "Done" output or the "Error" output becomes TRUE. Otherwise, direct switching might not be possible again, as "MX_SetOpMode" then possibly is not in its initial status.

The switching process must not be activated by several instances at the same time!

When the PLCopen profile is used for motion control, "MX_SetOpMode" must not be used. The secondary PLC operation modes are not allowed as "NewOpMode" and internally provided for the PLCopen profile of the PLC.

MX_PLCopen



- With HydraulicDrive and permanent control, the first 6 operation modes have a fixed parameter setting and must not be changed.
- The secondary operation modes 6 and 7 can be freely parameterized and selected in operation with the function block "MX_SetOpMode".
- When the function block "MB_PreSetMode" is used, the drive, after it has been switched on, immediately switches to this secondary operation mode without briefly remaining in "Drive Halt" before.
- Active motion function blocks signal "CommandAborted" when a new operation mode is set with "MX_SetOpMode".
- Upon the command "MX_SetOpMode", the axis changes from "AH" to "AF".



The operation mode must not be switched by several instances at the same time; this is prevented by the system.

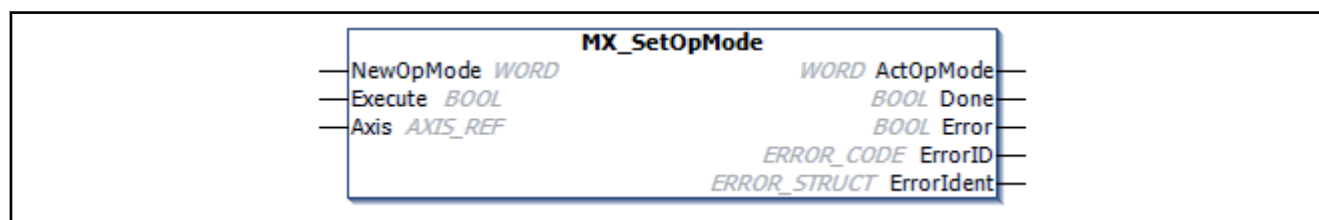


Fig. 10-36: Firmware Function Block "MX_SetOpMode"

Input variable	Data type	Description
NewOpMode	WORD	The operation mode set at "NewOpMode" is activated by a 0-1 edge at "Execute". "NewOpMode" can presently have the following values: • 0: S-0-0032, Primary operation mode • 1: S-0-0033, Secondary operation mode 1 • 2: S-0-0034, Secondary operation mode 2 • 3: S-0-0035, Secondary operation mode 3 • 4: S-0-0284, Secondary operation mode 4 • 5: S-0-0285, Secondary operation mode 5 • 6: S-0-0286, Secondary operation mode 6 • 7: S-0-0287, Secondary operation mode 7
Execute	BOOL	The function block is activated with a rising edge at "Execute". All outputs, except for the "ActOpMode" output, are reset with a falling edge at "Execute". Independent of "Execute", the current operation mode is displayed at "ActOpMode".

Tab. 10-90: Input Variables of the Firmware Function Block "MX_SetOpMode"

Output variable	Data type	Description
ActOpMode	WORD	"ActOpMode" signals which operation mode is active in the drive in the case of drive enable. The output adopts the value set at "NewOpMode".
Done	BOOL	With "Done"=TRUE the function block signals that processing of the task has been successfully completed

Output variable	Data type	Description
Error	BOOL	With "Error"=TRUE the function block signals that an error is present
ErrorID *	ERROR_CODE	"ErrorID" gives a rough error description.
ErrorIdent *	ERROR_STRUCT	Via the elements: "Table", "Additional1" and "Additional2", "ErrorIdent" shows the exact cause of the error.

* See note below

Tab. 10-91: Output Variables of the Firmware Function Block "MX_SetOpMode"



NOTE: For the list of all possible errors, see: "Error Reference Lists"

Input/output variable (VAR_IN_OUT)	Data type	Description
Axis	AXIS_REF	Defines the axis. The following axes are possible: MPB (IndraDrive) and HDC (HydraulicDrive): Axis1 MPC (IndraDrive): - Axis1 (master) or - Axis2...Axis10 (a total of 9 slaves / remote axes) The remote axes contained in the list "P-0-1801.0.10, CCD: Addresses of projected drives" correspond to the order of Axis2 to Axis10 (Axis2 acts on the drive of list element 0 of P-0-1801.0.10,...)

Tab. 10-92: Input/Output Variable of the Firmware Function Block "MX_SetOpMode"

10.4 "Parameters" - Function Blocks/Functions for Parameters

10.4.1 MB_Command

Brief Description The function block "MB_Command" is used to execute, monitor and abort commands.

Interface Description

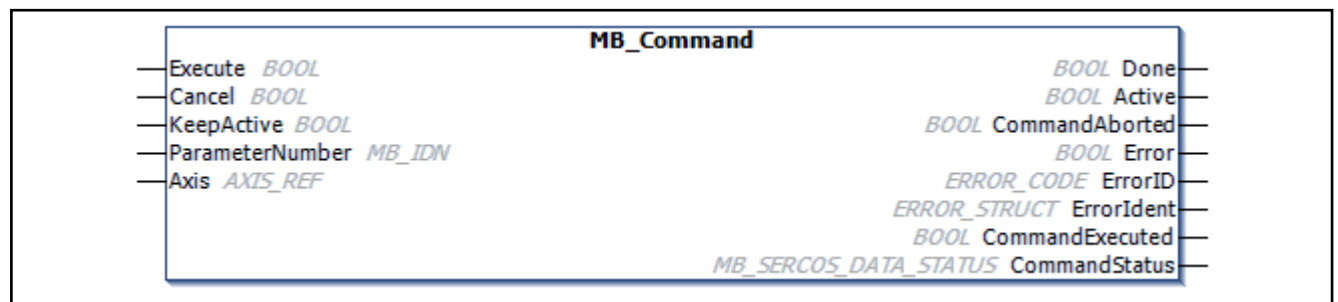


Fig. 10-37: Function Block "MB_Command"

Interface Variables

MX_PLCOpen

	Name	Type	Comment
VAR_IN_OUT	Axis	AXIS_REF	Reference of axis
VAR_INPUT	Execute	BOOL	Function block is enabled (one-time, edge-controlled)
	Cancel	BOOL	Command execution is aborted (one-time, edge-controlled)
	KeepActive	BOOL	Keep activation of command active upon execution (permanent, status-controlled)
	ParameterNumber	MB_IDN	Command
VAR_OUTPUT	Done	BOOL	Command successfully completed and reset
	Active	BOOL	Command execution active
	CommandAborted	BOOL	Command successfully aborted and reset
	Error	BOOL	With "Error"=TRUE the function block signals that an error is present
	ErrorID	ERROR_CODE	"ErrorID" gives a rough error description.
	ErrorIdent	ERROR_STRUCT	Via the elements: "Table", "Additional1" and "Additional2", "ErrorIdent" shows the exact cause of the error.
	CommandExecuted	BOOL	Command successfully completed
	CommandStatus	MB_SERCOS_DATA_STATUSES	Status of the command execution

Tab. 10-93: Interface Variables Function Block MB_Command

Signal-Time Diagram

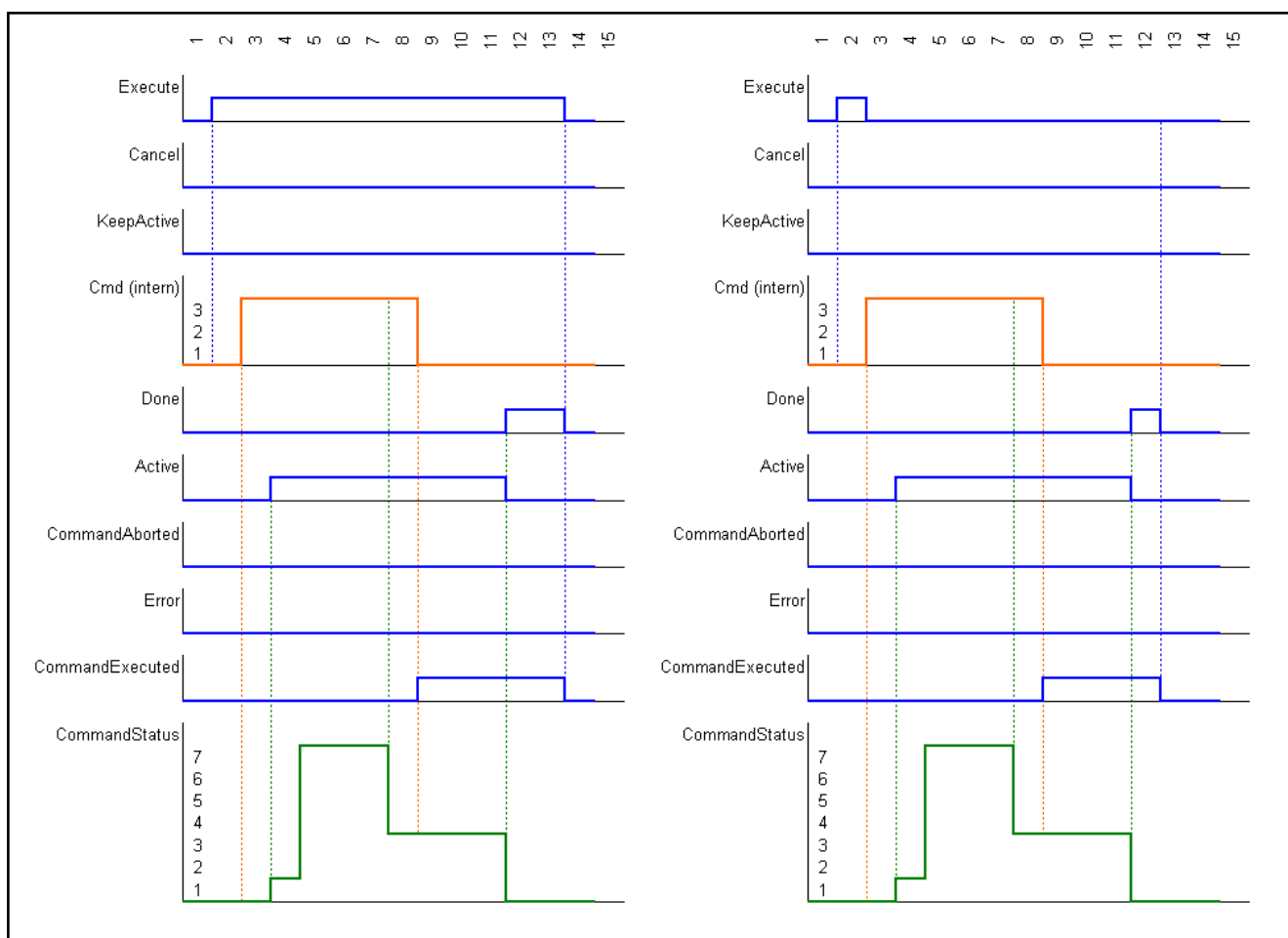


Fig. 10-38: Signal-Time Diagram for Signal Sequence "Execute-Done"

MX_PLCOpen

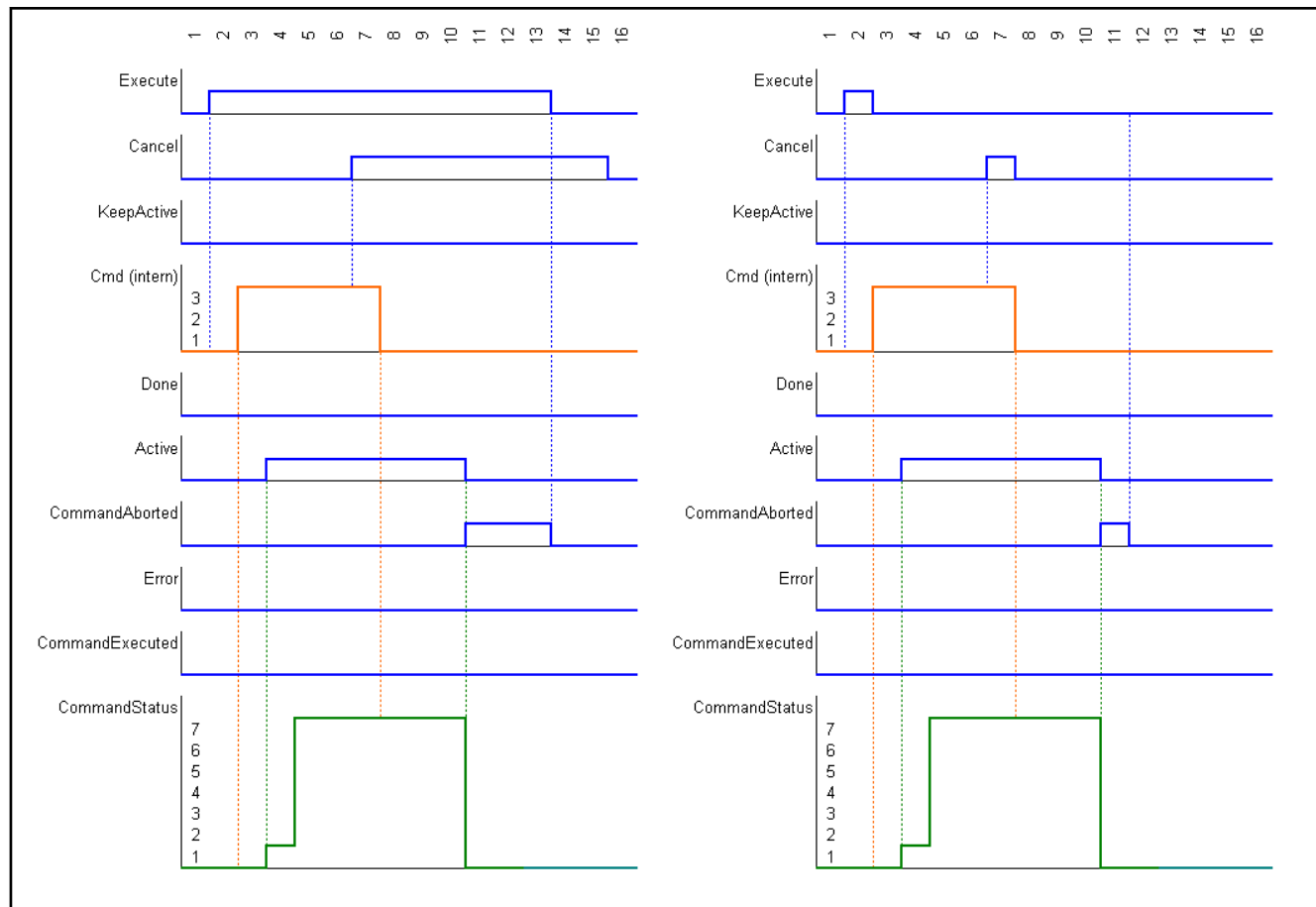


Fig. 10-39: Signal-Time Diagram for Signal Sequence "Execute-Cancel-CommandAborted"

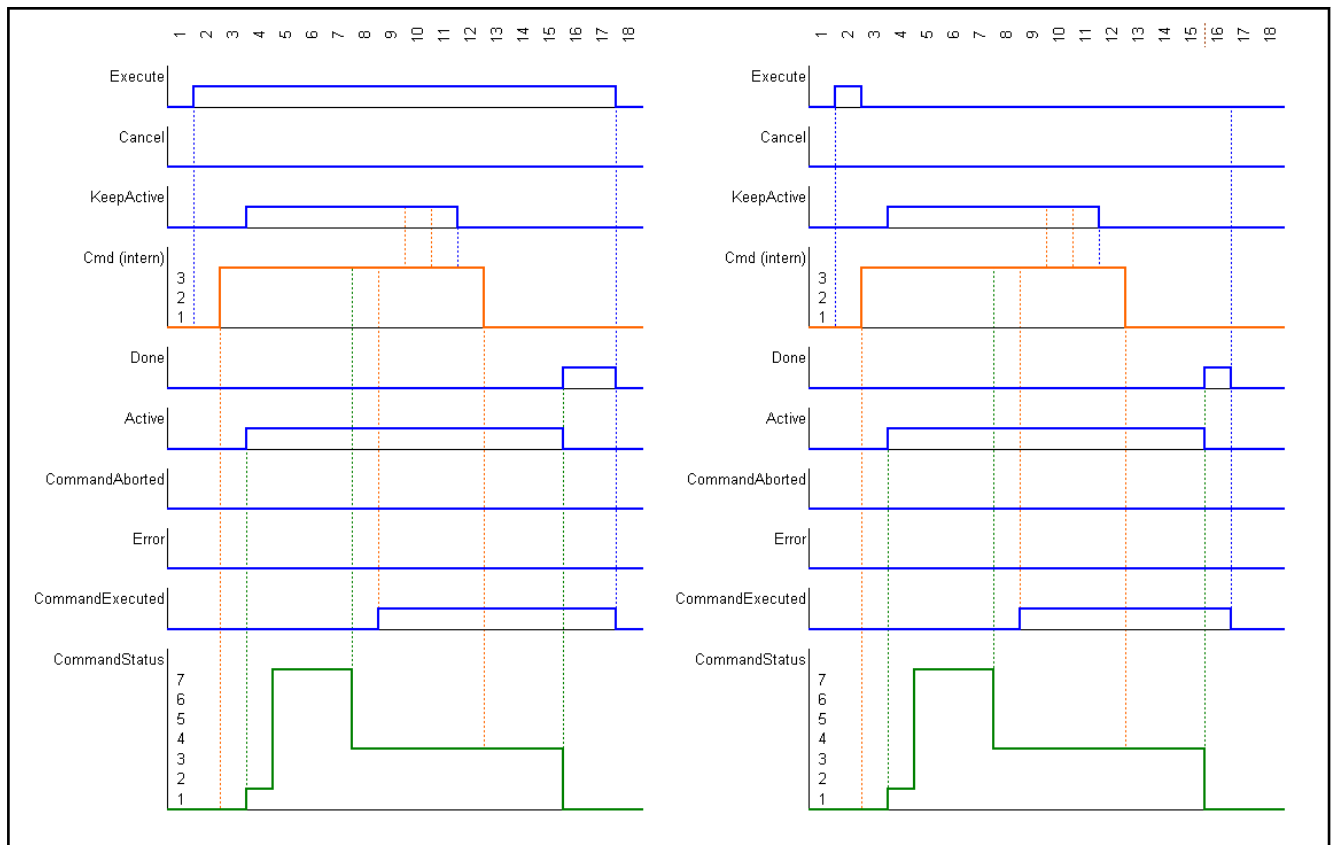


Fig. 10-40: Signal-Time Diagram for Signal Sequence "Execute-KeepActive-Done"

MX_PLCOpen

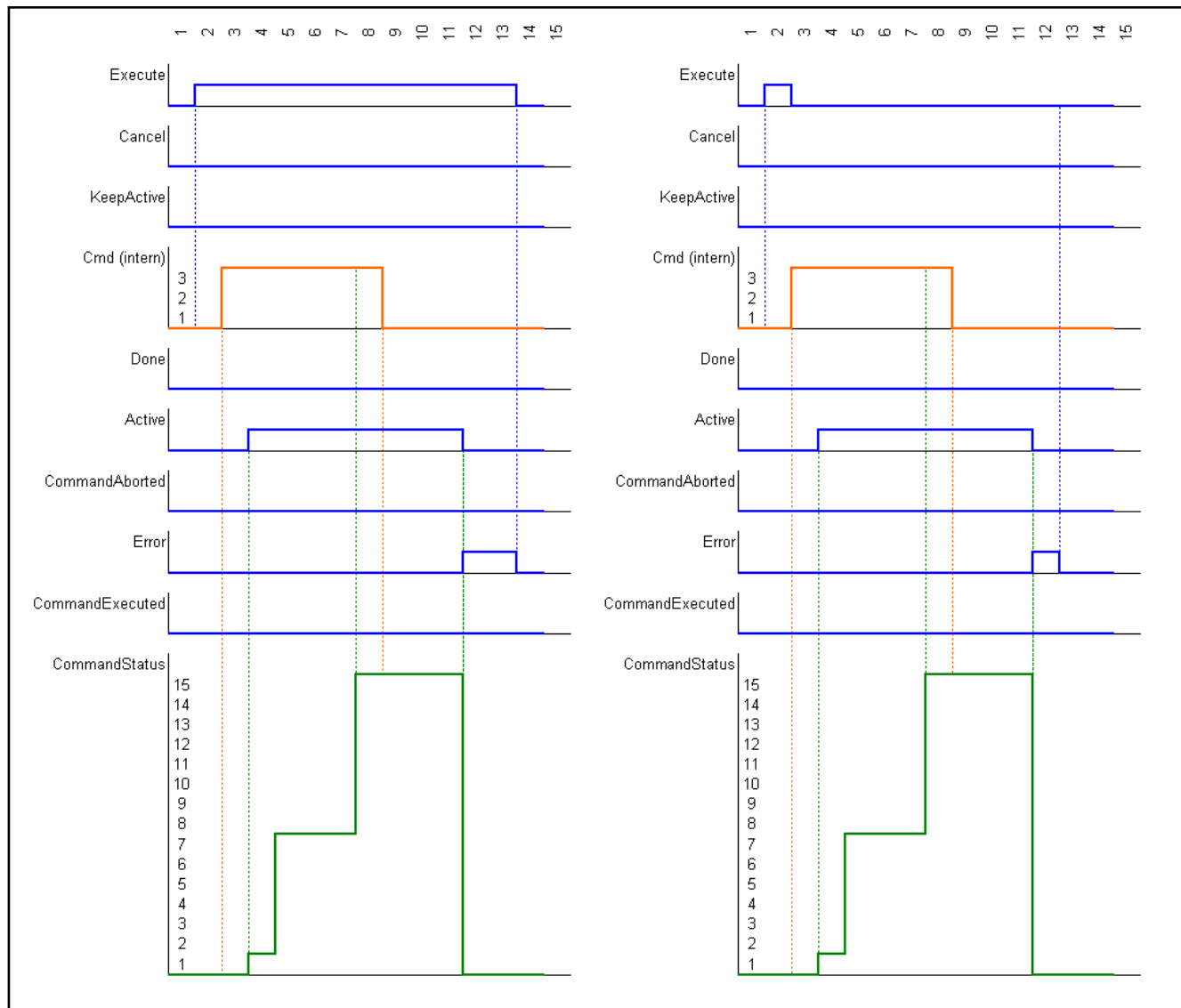


Fig. 10-41: Signal-Time Diagram for Signal Sequence "Execute-Error"

Functional Description

After the function block has been enabled, the function block "MB_Command" by means of "Execute" starts the command execution of the command defined via the "ParameterNumber" input. Generally, the command is automatically reset when processing has been completed. The command must be specified in the parameter IDN format in accordance with the sercos standard. Apart from this direct specification, it is possible to use a predefined constant from the global variables of the basic library (e.g. "ML_Base") or a conversion function from the "RIL_Uilities" library. Depending on the command itself, the command execution can be aborted via the "Cancel" input at any time. The "KeepActive" input allows keeping the command active upon its execution. When the "KeepActive" input is deactivated, the command is automatically reset. For the duration of at least one call, the outputs "Done" or "CommandAborted" signal that a command was successfully completed or aborted. The "Active" output and the corresponding data at the "CommandStatus" output show the active processing of the command. The current status of the command execution is permanently signaled at the "CommandStatus" output; the "CommandExecuted" output acknowledges that the command execution was

completed. If an error occurs during the processing of the command, this is signaled via the "Error" output and via the elements of error classification (ErrorID, ErrorIdent) for the duration of at least one call.

Error Handling

The function block "MB_Command" uses the error tables **F_RELATED_TABLE**, 16#0170, and **INDRV_TABLE**, 16#0070. In Additional1 and Additional2 it can generate the errors contained in the table below. In addition to this error, the included function blocks "MB_WriteParameter", "MB_ReadSercosDataStatus", "MB_ReadParameter" and "MB_ReadSercosAttribute" can signal other errors.

ErrorID	Additional1	Additional2	Description
INPUT_RANGE_ERROR, 16#0006	16#0002	16#0001	Invalid input - The function block "MB_Command" does not support retriggering of the command execution.
INPUT_RANGE_ERROR, 16#0006	16#0002	16#0002	Invalid input - Control commands do not support the inputs "Cancel" or "KeepActive".
INPUT_RANGE_ERROR, 16#0006	16#0002	16#0003	Invalid input - The "ParameterNumber" input does not contain any valid command.
DEVICE_ERROR, 16#0008	16#1701	16#0000	Error during command execution (control)
STATE_MACHINE_ERROR, 16#0005	16#0006	16#????	Invalid status of the state machine 16#???? specifies the invalid status

Tab. 10-94: Error Numbers of the Error Table "F_RELATED_TABLE", Caused by "MB_Command"

ErrorID	Additional1	Additional2	Description
DEVICE_ERROR, 16#0008	16#?????	16#0000	Error during command execution (drive) - 16#???? contains the current value from S-0-0390.

Tab. 10-95: Error Numbers of the Error Table "INDRV_TABLE", Caused by "MB_Command"

10.4.2 MB_ReadListParameter

With the function block "MB_ReadListParameter" it is possible to read the current value of a list parameter.

The IDN of the list parameter to be read has to be applied at the "ParameterNumber" input. The number of bytes to be read has to be indicated at the "NoOfBytes" input. At the "ValueAdr" input indicate the initial address of the field to which the bytes that have been read are to be written. The processing of the function block is started with a rising edge at the "Execute" input ("Execute"=TRUE).



To test the function block "MB_ReadListParameter", the instance variable "ActivateCounter" can be used. It counts the positive edges of the "Execute" signal and thereby signals whether the function block is run with an edge.

Local Axis

When the function block is used for a local axis, all list elements of the selected parameter are immediately read when the function block is called.

MX_PLCOpen

After the list elements have been read, control goes back to the program and TRUE is provided at the "Done" output.



You should observe that the reading process can take some time, depending on the volume of data to be read; if necessary, you should use a separate task.

Example: Reading all list elements of parameter P-0-0072 (1024 long words) takes several 100 µs.

Remote Axes



Accessing parameters of remote axes is possible in the MLD-M system mode, in the CCD basic mode and in the CCD system mode.

When the function block is used for remote axes, the read task is returned to the service channel (sercos III). Only at a later call can the function block signal "Done".

The outputs "Done", "Active" and "Error" have to be cyclically polled.

The data transmission can take several seconds, depending on the data volume.



As long as the sercos ring is not yet in phase 2 - and temporarily during the transition from phase 3 to phase 4 - it is impossible to exchange data with remote axes. In this case, the function block signals an error.

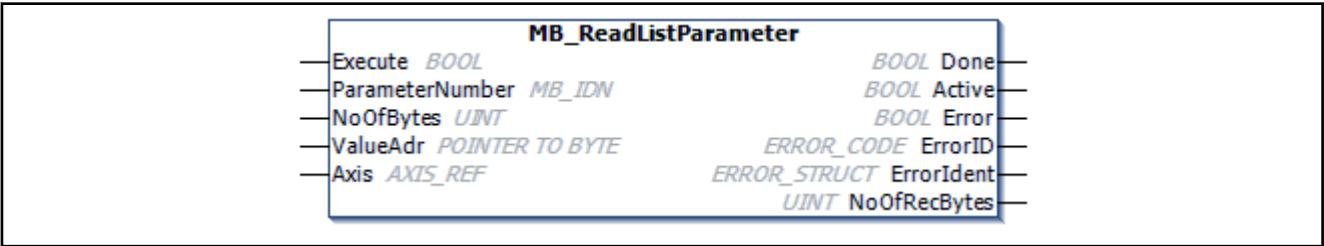


Fig. 10-42: Firmware Function Block "MB_ReadListParameter"

Input variable	Data type	Description
Execute	BOOL	See "Signal Time Diagrams"
ParameterNumber	MB_IDN	IDN of the parameter to be read
NoOfBytes	UINT	The number of bytes to be read has to be indicated at the "NoOfBytes" input. Example: NoOfBytes:=SIZEOF(arFeld1)
ValueAdr	POINTER TO BYTE	At the "ValueAdr" input indicate the initial address of the field to which the bytes that have been read are to be written. Example: ValueAdr := ADR(arFeld1)

Tab. 10-96: Input Variables of the Firmware Function Block "MB_ReadListParameter"

Output variable	Data type	Description
Done	BOOL	With "Done"=TRUE the function block signals that the reading of the data value of the list parameter was successful
Active	BOOL	For remote axes, "Active"=TRUE signals that the function block has accepted the processing command, for single axes this output is not set!

Output variable	Data type	Description
Error	BOOL	With "Error"=TRUE the function block signals that an error is present
ErrorID *	ERROR_CODE	"ErrorID" gives a rough error description.
ErrorIdent *	ERROR_STRUCT	Via the elements: "Table", "Additional1" and "Additional2", "ErrorIdent" shows the exact cause of the error.
NoOfRecBytes	UINT	The "NoOfRecBytes" output signals how many bytes were actually read

* See note below

Tab. 10-97: Output Variables of the Firmware Function Block "MB_ReadListParameter"



NOTE: For the list of all possible errors, see: "Error Reference Lists"

Input / output variable (VAR_IN_OUT)	Data type	Description
Axis	AXIS_REF	Defines the axis. The following axes are possible: MPB (IndraDrive) and HDC (HydraulicDrive): Axis1 MPC (IndraDrive): - Axis1 (master) or - Axis2...Axis10 (a total of 9 slaves / remote axes) The remote axes contained in the list "P-0-1801.0.10, CCD: Addresses of projected drives" correspond to the order of Axis2 to Axis10 (Axis2 acts on the drive of list element 0 of P-0-1801.0.10,...)

Tab. 10-98: Input / Output Variable of the Firmware Function Block "MB_ReadListParameter"

10.4.3 MB_ReadMaxRealValue

With the function block "MB_ReadMaxRealValue" it is possible to read the maximum allowed value of a parameter. The value is returned in REAL format.

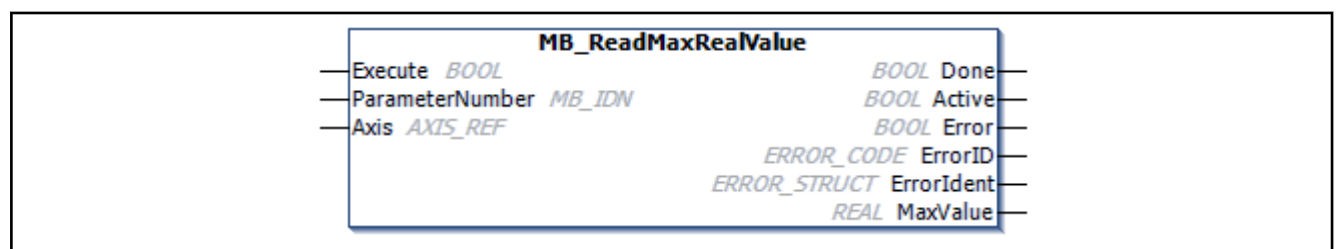


Fig. 10-43: Firmware Function Block "MB_ReadMaxRealValue"

Input variable	Data type	Description
Execute	BOOL	See "Signal Time Diagrams"
ParameterNumber	MB_IDN	At the "ParameterNumber" input apply the IDN of the parameter the maximum value of which is to be read

Tab. 10-99: Input Variables of the Firmware Function Block "MB_ReadMaxRealValue"

MX_PLCOpen

Output variable	Data type	Description
Done	BOOL	With "Done"=TRUE the function block signals that the processing of the task has been successfully completed
Active	BOOL	For remote axes, "Active"=TRUE signals that the function block has accepted the processing command, for single axes this output is not set!
Error	BOOL	With "Error"=TRUE the function block signals that an error is present
ErrorID *	ERROR_CODE	"ErrorID" gives a rough error description.
ErrorIdent *	ERROR_STRUCT	Via the elements: "Table", "Additional1" and "Additional2", "ErrorIdent" shows the exact cause of the error.
MaxValue	REAL	If "Done"=TRUE, "MaxValue" signals the maximum input value of the selected parameter. Note: According to the scaling, the input value of the parameter has decimal places which have to be taken into account and used for calculation

* See note below

Tab. 10-100: Output Variables of the Firmware Function Block "MB_ReadMaxReal-Value"



NOTE: For the list of all possible errors, see: "Error Reference Lists"

Input / output variable (VAR_IN_OUT)	Data type	Description
Axis	AXIS_REF	Defines the axis. The following axes are possible: MPB (IndraDrive) and HDC (HydraulicDrive): Axis1 MPC (IndraDrive): - Axis1 (master) or - Axis2...Axis10 (a total of 9 slaves / remote axes) The remote axes contained in the list "P-0-1801.0.10, CCD: Addresses of projected drives" correspond to the order of Axis2 to Axis10 (Axis2 acts on the drive of list element 0 of P-0-1801.0.10,...) Concerning the access to parameters of remote axes, please observe the notes below!

Tab. 10-101: Input / Output Variable of the Firmware Function Block "MB_Read-MaxRealValue"



In the case of a remote axis, the task is returned to the service channel (sercos III). Only at a later call can the function block signal "Done"=TRUE. Therefore, the resulting times for the function block calls themselves are always short. The outputs "Done", "Active" and "Error" have to be cyclically polled. The data transmission can take several seconds, depending on the load of the service channel.

As long as the sercos ring is not yet in phase 2 - and temporarily during the transition from phase 3 to phase 4 - it is impossible to exchange data with remote axes. In this case, the function block signals an error.



Accessing parameters of remote axes is possible in the MLD-M system mode, in the CCD basic mode and in the CCD system mode.

10.4.4 MB_ReadMaxValue

With the function block "MB_ReadMaxValue" it is possible to read the maximum allowed value of a parameter. The value is returned in DINT format.

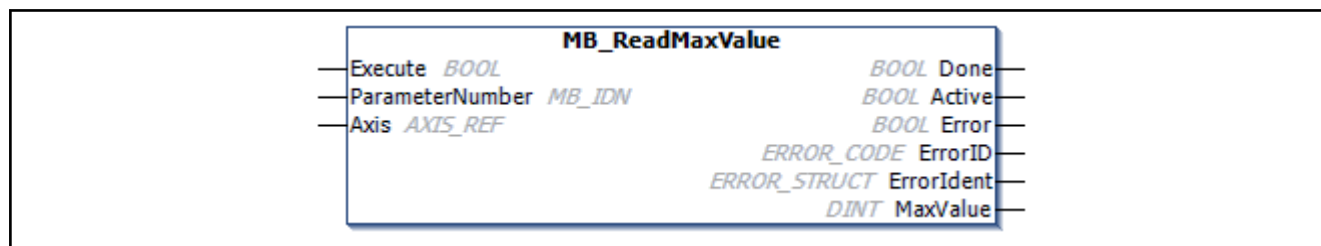


Fig. 10-44: Firmware Function Block "MB_ReadMaxValue"

Input variable	Data type	Description
Execute	BOOL	See "Signal Time Diagrams"
ParameterNumber	MB_IDN	At the "ParameterNumber" input apply the IDN of the parameter the maximum value of which is to be read

Tab. 10-102: Input Variables of the Firmware Function Block "MB_ReadMaxValue"

Output variable	Data type	Description
Done	BOOL	With "Done"=TRUE the function block signals that the processing of the task has been successfully completed
Active	BOOL	For remote axes, "Active"=TRUE signals that the function block has accepted the processing command, for single axes this output is not set!
Error	BOOL	With "Error"=TRUE the function block signals that an error is present
ErrorID *	ERROR_CODE	"ErrorID" gives a rough error description.
ErrorIdent *	ERROR_STRUCT	Via the elements: "Table", "Additional1" and "Additional2", "ErrorIdent" shows the exact cause of the error.
MaxValue	DINT	If "Done"=TRUE, "MaxValue" signals the maximum input value of the selected parameter. Note: According to the scaling, the input value of the parameter has decimal places which have to be taken into account and used for calculation

* See note below

Tab. 10-103: Output Variables of the Firmware Function Block "MB_ReadMaxValue"



NOTE: For the list of all possible errors, see: "Error Reference Lists"

MX_PLCOpen

Input / output variable (VAR_IN_OUT)	Data type	Description
Axis	AXIS_REF	Defines the axis. The following axes are possible: MPB (IndraDrive) and HDC (HydraulicDrive): Axis1 MPC (IndraDrive): - Axis1 (master) or - Axis2...Axis10 (a total of 9 slaves / remote axes) The remote axes contained in the list "P-0-1801.0.10, CCD: Addresses of projected drives" correspond to the order of Axis2 to Axis10 (Axis2 acts on the drive of list element 0 of P-0-1801.0.10,...) Concerning the access to parameters of remote axes, please observe the notes below!

Tab. 10-104: Input / Output Variable of the Firmware Function Block "MB_ReadMaxValue"



In the case of a remote axis, the task is returned to the service channel (sercos III). Only at a later call can the function block signal "Done"="TRUE". Therefore, the resulting times for the function block calls themselves are always short. The outputs "Done", "Active" and "Error" have to be cyclically polled. The data transmission can take several seconds, depending on the load of the service channel.

As long as the sercos ring is not yet in phase 2 - and temporarily during the transition from phase 3 to phase 4 - it is impossible to exchange data with remote axes. In this case, the function block signals an error.



Accessing parameters of remote axes is possible in the MLD-M system mode, in the CCD basic mode and in the CCD system mode.

10.4.5 MB_ReadMinRealValue

With the function block "MB_ReadMinRealValue" it is possible to read the minimum allowed value of a parameter. The value is returned in REAL format.



Fig. 10-45: Firmware Function Block "MB_ReadMinRealValue"

Input variable	Data type	Description
Execute	BOOL	See "Signal Time Diagrams"
ParameterNumber	MB_IDN	At the "ParameterNumber" input apply the IDN of the parameter the minimum value of which is to be read

Tab. 10-105: *Input Variables of the Firmware Function Block "MB_ReadMinReal-Value"*

Output variable	Data type	Description
Done	BOOL	With "Done"=TRUE the function block signals that the processing of the task has been successfully completed
Active	BOOL	For remote axes, "Active"=TRUE signals that the function block has accepted the processing command, for single axes this output is not set!
Error	BOOL	With "Error"=TRUE the function block signals that an error is present
ErrorID *	ERROR_CODE	"ErrorID" gives a rough error description.
ErrorIdent *	ERROR_STRUCT	Via the elements: "Table", "Additional1" and "Additional2", "ErrorIdent" shows the exact cause of the error.
MinValue	REAL	If "Done"=TRUE, "MinValue" signals the minimum input value of the selected parameter. Note: According to the scaling, the input value of the parameter has decimal places which have to be taken into account and used for calculation

* See note below

Tab. 10-106: *Output Variables of the Firmware Function Block "MB_ReadMinReal-Value"*



NOTE: For the list of all possible errors, see: "Error Reference Lists"

Input / output variable (VAR_IN_OUT)	Data type	Description
Axis	AXIS_REF	Defines the axis. The following axes are possible: MPB (IndraDrive) and HDC (HydraulicDrive): Axis1 MPC (IndraDrive): - Axis1 (master) or - Axis2...Axis10 (a total of 9 slaves / remote axes) The remote axes contained in the list "P-0-1801.0.10, CCD: Addresses of projected drives" correspond to the order of Axis2 to Axis10 (Axis2 acts on the drive of list element 0 of P-0-1801.0.10,...) Concerning the access to parameters of remote axes, please observe the notes below!

Tab. 10-107: *Input / Output Variable of the Firmware Function Block "MB_ReadMin-RealValue"*

MX_PLCOpen



In the case of a remote axis, the task is returned to the service channel (sercos III). Only at a later call can the function block signal "Done"="TRUE". Therefore, the resulting times for the function block calls themselves are always short. The outputs "Done", "Active" and "Error" have to be cyclically polled. The data transmission can take several seconds, depending on the load of the service channel.

As long as the sercos ring is not yet in phase 2 - and temporarily during the transition from phase 3 to phase 4 - it is impossible to exchange data with remote axes. In this case, the function block signals an error.



Accessing parameters of remote axes is possible in the MLD-M system mode, in the CCD basic mode and in the CCD system mode.

10.4.6 MB_ReadMinValue

With the function block "MB_ReadMinValue" it is possible to read the minimum allowed value of a parameter. The value is returned in DINT format.

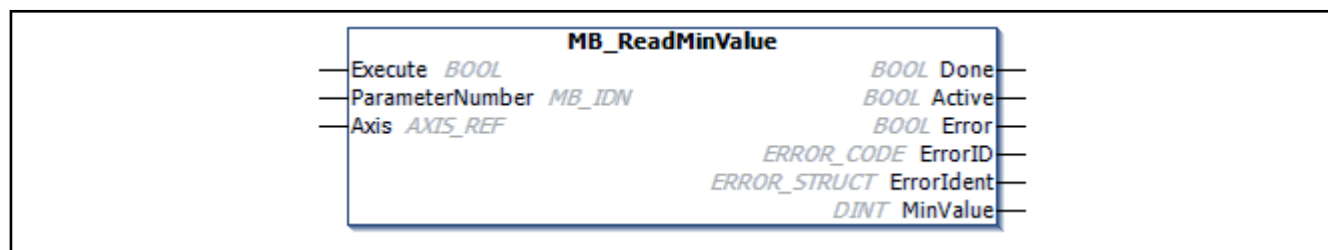


Fig. 10-46: Firmware Function Block "MB_ReadMinValue"

Input variable	Data type	Description
Execute	BOOL	See "Signal Time Diagrams"
ParameterNumber	MB_IDN	At the "ParameterNumber" input apply the IDN of the parameter the minimum value of which is to be read

Tab. 10-108: Input Variables of the Firmware Function Block "MB_ReadMinValue"

Output variable	Data type	Description
Done	BOOL	With "Done"=TRUE the function block signals that the processing of the task has been successfully completed
Active	BOOL	For remote axes, "Active"=TRUE signals that the function block has accepted the processing command, for single axes this output is not set!
Error	BOOL	With "Error"=TRUE the function block signals that an error is present
ErrorID *	ERROR_CODE	"ErrorID" gives a rough error description.

Output variable	Data type	Description
ErrorIdent *	ERROR_STRUCT	Via the elements: "Table", "Additional1" and "Additional2", "ErrorIdent" shows the exact cause of the error.
MinValue	DINT	If "Done"=TRUE, "MinValue" signals the minimum input value of the selected parameter. Note: According to the scaling, the input value of the parameter has decimal places which have to be taken into account and used for calculation

* See note below

Tab. 10-109: Output Variables of the Firmware Function Block "MB_ReadMinValue"



NOTE: For the list of all possible errors, see: "Error Reference Lists"

Input / output variable (VAR_IN_OUT)	Data type	Description
Axis	AXIS_REF	Defines the axis. The following axes are possible: MPB (IndraDrive) and HDC (HydraulicDrive): Axis1 MPC (IndraDrive): - Axis1 (master) or - Axis2...Axis10 (a total of 9 slaves / remote axes) The remote axes contained in the list "P-0-1801.0.10, CCD: Addresses of projected drives" correspond to the order of Axis2 to Axis10 (Axis2 acts on the drive of list element 0 of P-0-1801.0.10,...) Concerning the access to parameters of remote axes, please observe the notes below!

Tab. 10-110: Input / Output Variable of the Firmware Function Block "MB_ReadMinValue"



In the case of a remote axis, the task is returned to the service channel (sercos III). Only at a later call can the function block signal "Done"=TRUE. Therefore, the resulting times for the function block calls themselves are always short. The outputs "Done", "Active" and "Error" have to be cyclically polled. The data transmission can take several seconds, depending on the load of the service channel.

As long as the sercos ring is not yet in phase 2 - and temporarily during the transition from phase 3 to phase 4 - it is impossible to exchange data with remote axes. In this case, the function block signals an error.



Accessing parameters of remote axes is possible in the MLD-M system mode, in the CCD basic mode and in the CCD system mode.

10.4.7 MB_ReadName

With the function block "MB_ReadName" it is possible to read the name of a parameter.

At the "ParameterNumber" input you have to apply the parameter number of the parameter the name of which is to be read. The processing of the

MX_PLCOpen

function block is started with a rising edge at the "Execute" input. Error-free processing is signaled by "Done"=TRUE. Incorrect processing is signaled by "Error"=TRUE. In the case of error, a detailed description of the error is output at the outputs "ErrorID" and "ErrorIdent".

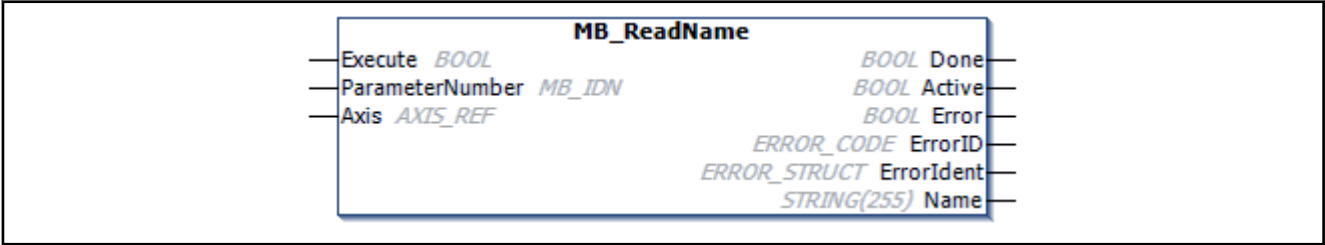


Fig. 10-47: Firmware Function Block "MB_ReadName"

Input variable	Data type	Description
Execute	BOOL	See "Signal Time Diagrams"
ParameterNumber	MB_IDN	At the "ParameterNumber" input apply the IDN of the parameter the name of which is to be read

Tab. 10-111: Input Variables of the Firmware Function Block "MB_ReadName"

Output variable	Data type	Description
Done	BOOL	With "Done"=TRUE the function block signals that the processing of the task has been successfully completed
Active	BOOL	For remote axes, "Active"=TRUE signals that the function block has accepted the processing command, for single axes this output is not set!
Error	BOOL	With "Error"=TRUE the function block signals that an error is present
ErrorID *	ERROR_CODE	"ErrorID" gives a rough error description.
ErrorIdent *	ERROR_STRUCT	Via the elements: "Table", "Additional1" and "Additional2", "ErrorIdent" shows the exact cause of the error.
Name	STRING (255)	If "Done"=TRUE, "Name" signals the name of the selected parameter.

* See note below

Tab. 10-112: Output Variables of the Firmware Function Block "MB_ReadName"



NOTE: For the list of all possible errors, see: "Error Reference Lists"

Input / output variable (VAR_IN_OUT)	Data type	Description
Axis	AXIS_REF	Defines the axis. The following axes are possible: MPB (IndraDrive) and HDC (HydraulicDrive): Axis1 MPC (IndraDrive): - Axis1 (master) or - Axis2...Axis10 (a total of 9 slaves / remote axes) The remote axes contained in the list "P-0-1801.0.10, CCD: Addresses of projected drives" correspond to the order of Axis2 to Axis10 (Axis2 acts on the drive of list element 0 of P-0-1801.0.10,...) Concerning the access to parameters of remote axes, please observe the notes below!

Tab. 10-113: Input / Output Variable of the Firmware Function Block "MB_Read-Name"



In the case of a remote axis, the task is returned to the service channel (sercos III). Only at a later call can the function block signal "Done"="TRUE". Therefore, the resulting times for the function block calls themselves are always short. The outputs "Done", "Active" and "Error" have to be cyclically polled. The data transmission can take several seconds, depending on the load of the service channel.

As long as the sercos ring is not yet in phase 2 - and temporarily during the transition from phase 3 to phase 4 - it is impossible to exchange data with remote axes. In this case, the function block signals an error.



Accessing parameters of remote axes is possible in the MLD-M system mode, in the CCD basic mode and in the CCD system mode.

10.4.8 MB_ReadParameter

With the function block "MB_ReadParameter" it is possible to read the current value of a parameter. The return value is DINT.



With the function block "MB_ReadParameter", the input variables are only applied with a 0-1 edge at the "Enable" input.



List and string parameters cannot be read with the function block.

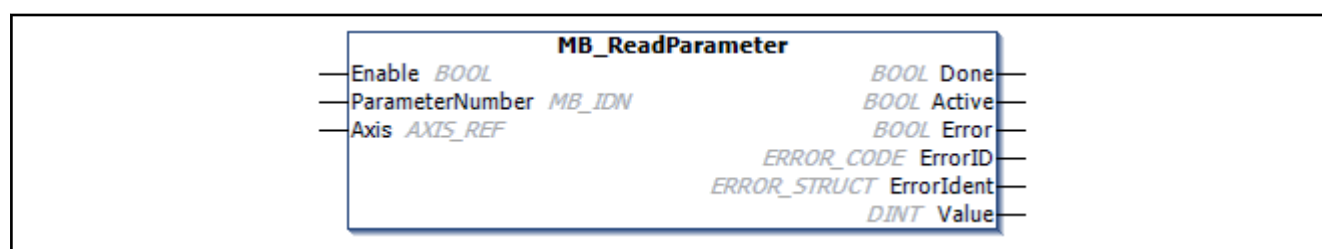


Fig. 10-48: Firmware Function Block "MB_ReadParameter"

MX_PLCOpen

Input variable	Data type	Description
Enable	BOOL	At a positive edge at the "Enable" input, the input variables are applied
ParameterNumber	MB_IDN	At the "ParameterNumber" input apply the IDN of the parameter (e.g. FP_S_0_0052) the current value of which is to be read

Tab. 10-114: Input Variables of the Firmware Function Block "MB_ReadParameter"

Output variable	Data type	Description
Done	BOOL	With "Done"=TRUE the function block signals that the processing of the task has been successfully completed
Active	BOOL	For remote axes, "Active"=TRUE signals that the function block has accepted the processing command, for single axes this output is not set!
Error	BOOL	With "Error"=TRUE the function block signals that an error is present
ErrorID *	ERROR_CODE	"ErrorID" gives a rough error description.
ErrorIdent *	ERROR_STRUCT	Via the elements: "Table", "Additional1" and "Additional2", "ErrorIdent" shows the exact cause of the error.
Value	DINT	If "Done"=TRUE, "Value" signals the value of the selected parameter. According to the scaling, the parameter has decimal places which have to be taken into account and used for calculation

* See note below

Tab. 10-115: Output Variables of the Firmware Function Block "MB_ReadParameter"



NOTE: For the list of all possible errors, see: "Error Reference Lists"

Input / output variable (VAR_IN_OUT)	Data type	Description
Axis	AXIS_REF	Defines the axis. The following axes are possible: MPB (IndraDrive) and HDC (HydraulicDrive): Axis1 MPC (IndraDrive): - Axis1 (master) or - Axis2...Axis10 (a total of 9 slaves / remote axes) The remote axes contained in the list "P-0-1801.0.10, CCD: Addresses of projected drives" correspond to the order of Axis2 to Axis10 (Axis2 acts on the drive of list element 0 of P-0-1801.0.10,...) Concerning the access to parameters of remote axes, please observe the notes below!

Tab. 10-116: Input / Output Variable of the Firmware Function Block "MB_ReadParameter"



In the case of a remote axis, the task is returned to the service channel (sercos III). Only at a later call can the function block signal "Done"="TRUE". Therefore, the resulting times for the function block calls themselves are always short. The outputs "Done", "Active" and "Error" have to be cyclically polled. The data transmission can take several seconds, depending on the load of the service channel.

As long as the sercos ring is not yet in phase 2 - and temporarily during the transition from phase 3 to phase 4 - it is impossible to exchange data with remote axes. In this case, the function block signals an error.



Accessing parameters of remote axes is possible in the MLD-M system mode, in the CCD basic mode and in the CCD system mode.

10.4.9 MB_ReadRealListParameter

With the function block "MB_ReadRealListParameter" it is possible to read the current list of a list parameter.

The IDN of the list parameter to be read has to be applied at the "ParameterNumber" input. The number of bytes to be read has to be indicated at the "NoOfBytes" input. At the "ValueAdr" input indicate the initial address of the field to which the bytes that have been read are to be written. The processing of the function block is started with a rising edge at the "Execute" input ("Execute"="TRUE").



To test the function block "MB_ReadRealListParameter", the instance variable "ActivateCounter" can be used. It counts the positive edges of the "Execute" signal and thereby signals whether the function block is run with an edge.

Local Axis

When the function block is used for a local axis, all list elements of the selected parameter are immediately read when the function block is called. After the list elements have been read, control goes back to the program and TRUE is provided at the "Done" output.



You should observe that the reading process can take some time, depending on the volume of data to be read; if necessary, you should use a separate task.

Example: Reading all list elements of parameter P-0-0072 (1024 long words) takes several 100 µs.

Remote Axes



Accessing parameters of remote axes is possible in the MLD-M system mode, in the CCD basic mode and in the CCD system mode.

When the function block is used for remote axes, the read task is returned to the service channel (sercos III). Only at a later call can the function block signal "Done"="TRUE".

The outputs "Done", "Active" and "Error" have to be cyclically polled.

The data transmission can take several seconds, depending on the data volume.

MX_PLCOpen



As long as the sercos ring is not yet in phase 2 - and temporarily during the transition from phase 3 to phase 4 - it is impossible to exchange data with remote axes. In this case, the function block signals an error.

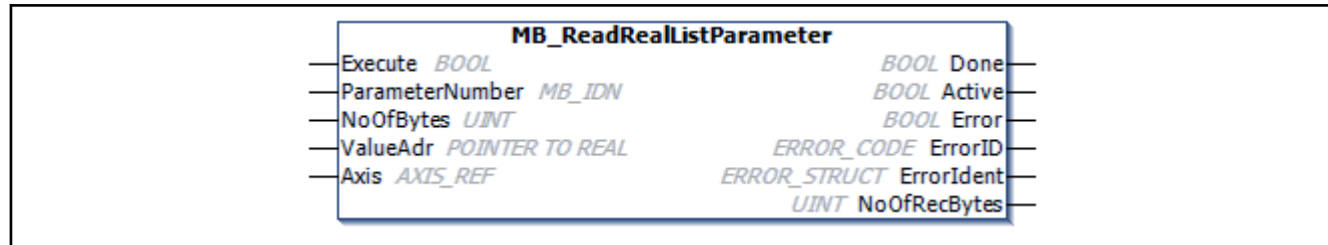


Fig. 10-49: Firmware Function Block "MB_ReadRealListParameter"

Input variable	Data type	Description
Execute	BOOL	See "Signal Time Diagrams"
ParameterNumber	MB_IDN	IDN of the parameter to be read
NoOfBytes	UINT	The number of bytes to be read has to be indicated at the "NoOfBytes" input Example: NoOfBytes:=SIZEOF(arFeld1)
ValueAdr	POINTER TO REAL	At the "ValueAdr" input indicate the initial address of the field to which the bytes that have been read are to be written. Example: ValueAdr := ADR(arFeld1)

Tab. 10-117: Input Variables of the Firmware Function Block "MB_ReadRealList-Parameter"

Output variable	Data type	Description
Done	BOOL	With "Done"=TRUE the function block signals that the cyclic reading of the data value of the parameter was successful
Active	BOOL	For remote axes, "Active"=TRUE signals that the function block has accepted the processing command, for single axes this output is not set!
Error	BOOL	With "Error"=TRUE the function block signals that an error is present
ErrorID *	ERROR_CODE	"ErrorID" gives a rough error description.
ErrorIdent *	ERROR_STRUCT	Via the elements: "Table", "Additional1" and "Additional2", "ErrorIdent" shows the exact cause of the error.
NoOfRecBytes	UINT	The "NoOfRecBytes" output signals how many bytes were actually read

* See note below

Tab. 10-118: Output Variables of the Firmware Function Block "MB_ReadRealList-Parameter"



NOTE: For the list of all possible errors, see: "Error Reference Lists"

Input / output variable (VAR_IN_OUT)	Data type	Description
Axis	AXIS_REF	Defines the axis. The following axes are possible: MPB (IndraDrive) and HDC (HydraulicDrive): Axis1 MPC (IndraDrive): - Axis1 (master) or - Axis2...Axis10 (a total of 9 slaves / remote axes) The remote axes contained in the list "P-0-1801.0.10, CCD: Addresses of projected drives" correspond to the order of Axis2 to Axis10 (Axis2 acts on the drive of list element 0 of P-0-1801.0.10,...) Concerning the access to parameters of remote axes, please observe the note below!

Tab. 10-119: Input / Output Variable of the Firmware Function Block "MB_ReadRealListParameter"



Accessing parameters of remote axes is possible in the MLD-M system mode, in the CCD basic mode and in the CCD system mode.

10.4.10 MB_ReadRealParameter

With the function block "MB_ReadRealParameter" it is possible to read the current value of a parameter. The return value is REAL.



List and string parameters cannot be read with the function block "MB_ReadRealParameter".

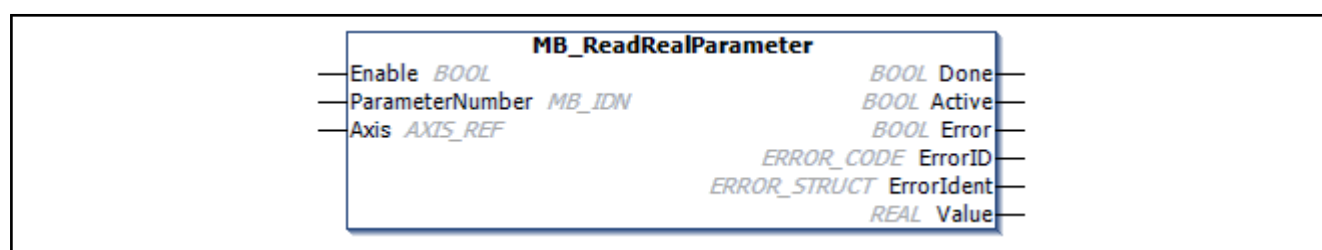


Fig. 10-50: Firmware Function Block "MB_ReadRealParameter"

Input variable	Data type	Description
Enable	BOOL	At a positive edge at the "Enable" input, the input variables are applied
ParameterNumber	MB_IDN	At the "ParameterNumber" input apply the IDN of the parameter the current value of which is to be read Any new parameter number will only be applied with the next positive edge at the "Enable" input

Tab. 10-120: Input Variables of the Firmware Function Block "MB_ReadRealParameter"

MX_PLCOpen

Output variable	Data type	Description
Done	BOOL	With "Done"=TRUE the function block signals that the processing of the task has been successfully completed
Active	BOOL	For remote axes, "Active"=TRUE signals that the function block has accepted the processing command, for single axes this output is not set!
Error	BOOL	With "Error"=TRUE the function block signals that an error is present
ErrorID *	ERROR_CODE	"ErrorID" gives a rough error description.
ErrorIdent *	ERROR_STRUCT	Via the elements: "Table", "Additional1" and "Additional2", "ErrorIdent" shows the exact cause of the error.
Value	REAL	If "Done"=TRUE, "Value" signals the value of the selected parameter

* See note below

Tab. 10-121: *Output Variables of the Firmware Function Block "MB_ReadRealParameter"*



NOTE: For the list of all possible errors, see: "Error Reference Lists"

Input / output variable (VAR_IN_OUT)	Data type	Description
Axis	AXIS_REF	Defines the axis. The following axes are possible: MPB (IndraDrive) and HDC (HydraulicDrive): Axis1 MPC (IndraDrive): - Axis1 (master) or - Axis2...Axis10 (a total of 9 slaves / remote axes) The remote axes contained in the list "P-0-1801.0.10, CCD: Addresses of projected drives" correspond to the order of Axis2 to Axis10 (Axis2 acts on the drive of list element 0 of P-0-1801.0.10,...) Concerning the access to parameters of remote axes, please observe the notes below!

Tab. 10-122: *Input / Output Variable of the Firmware Function Block "MB_ReadRealParameter"*



In the case of a remote axis, the task is returned to the service channel (sercos III). Only at a later call can the function block signal "Done"=TRUE. Therefore, the resulting times for the function block calls themselves are always short. The outputs "Done", "Active" and "Error" have to be cyclically polled. The data transmission can take several seconds, depending on the load of the service channel.

As long as the sercos ring is not yet in phase 2 - and temporarily during the transition from phase 3 to phase 4 - it is impossible to exchange data with remote axes. In this case, the function block signals an error.



Accessing parameters of remote axes is possible in the MLD-M system mode, in the CCD basic mode and in the CCD system mode.

10.4.11 MB_ReadSercosAttribute

With the function block "MB_ReadSercosAttribute" it is possible to read the sercos attribute of a parameter.

At the "ParameterNumber" input you have to apply the parameter number of the parameter the sercos attribute of which is to be read. The processing of the function block is started with a rising edge at the "Execute" input. Error-free processing is signaled by "Done"=TRUE. Incorrect processing is signaled by "Error"=TRUE. In the case of error, a detailed description of the error is output at the outputs "ErrorID" and "ErrorIdent".

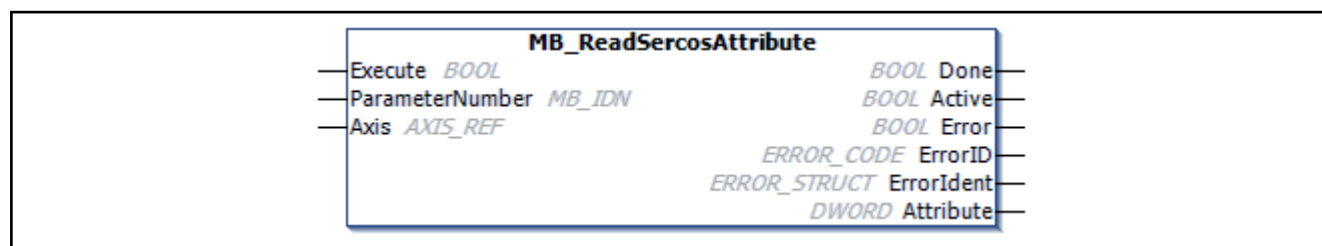


Fig. 10-51: Firmware Function Block "MB_ReadSercosAttribute"

Input variable	Data type	Description
Execute	BOOL	See "Signal Time Diagrams"
ParameterNumber	MB_IDN	At the "ParameterNumber" input you have to apply the parameter number of the parameter the sercos attribute of which is to be read. NOTE: Any new parameter number will only be applied with the next positive edge at the "Execute" input.

Tab. 10-123: Input Variables of the Firmware Function Block "MB_ReadSercosAttribute"

Output variable	Data type	Description
Done	BOOL	With "Done"=TRUE the function block signals that the cyclic reading of the sercos attribute is successfully running
Active	BOOL	For remote axes, "Active"=TRUE signals that the function block has accepted the processing command, for single axes this output is not set!
Error	BOOL	With "Error"=TRUE the function block signals that an error is present
ErrorID *	ERROR_CODE	"ErrorID" gives a rough error description.
ErrorIdent *	ERROR_STRUCT	Via the elements: "Table", "Additional1" and "Additional2", "ErrorIdent" shows the exact cause of the error.
Attribute	DWORD	If "Done"=TRUE, "Attribute" signals the attribute of the selected parameter

* See note below

Tab. 10-124: Output Variables of the Firmware Function Block "MB_ReadSercosAttribute"



NOTE: For the list of all possible errors, see: "Error Reference Lists"

MX_PLCOpen

Input / output variable (VAR_IN_OUT)	Data type	Description
Axis	AXIS_REF	Defines the axis. The following axes are possible: MPB (IndraDrive) and HDC (HydraulicDrive): Axis1 MPC (IndraDrive): - Axis1 (master) or - Axis2...Axis10 (a total of 9 slaves / remote axes) The remote axes contained in the list "P-0-1801.0.10, CCD: Addresses of projected drives" correspond to the order of Axis2 to Axis10 (Axis2 acts on the drive of list element 0 of P-0-1801.0.10,...) Concerning the access to parameters of remote axes, please observe the notes below!

Tab. 10-125: Input / Output Variable of the Firmware Function Block "MB_ReadSercosAttribute"



In the case of a remote axis, the task is returned to the service channel (sercos III). Only at a later call can the function block signal "Done"="TRUE". Therefore, the resulting times for the function block calls themselves are always short. The outputs "Done", "Active" and "Error" have to be cyclically polled. The data transmission can take several seconds, depending on the load of the service channel.

As long as the sercos ring is not yet in phase 2 - and temporarily during the transition from phase 3 to phase 4 - it is impossible to exchange data with remote axes. In this case, the function block signals an error.



Accessing parameters of remote axes is possible in the MLD-M system mode, in the CCD basic mode and in the CCD system mode.

10.4.12 MB_ReadSercosDataStatus

With the function block "MB_ReadSercosDataStatus" it is possible to read the status of a parameter.

At the "ParameterNumber" input you have to apply the parameter number of the parameter the status of which is to be read.

Error-free processing is signaled by "Done"=TRUE. Incorrect processing is signaled by "Error"=TRUE. In the case of error, a detailed description of the error is output at the outputs "ErrorID" and

"ErrorIdent".



Fig. 10-52: Firmware Function Block "MB_ReadSercosDataStatus"

Input variable	Data type	Description
Enable	BOOL	The function block is status-controlled. The input variables are applied with the positive edge of "Enable". New input values only take effect via the repeated positive edge of "Enable".
ParameterNumber	MB_IDN	At the "ParameterNumber" input you have to apply the parameter number of the parameter the status of which is to be read

Tab. 10-126: *Input Variables of the Firmware Function Block "MB_ReadSercosDataStatus"*

Output variable	Data type	Description
Done	BOOL	With "Done"=TRUE the function block signals that the processing of the function block has been successfully completed
Active	BOOL	For remote axes, "Active"=TRUE signals that the function block has accepted the processing command, for single axes this output is not set!
Error	BOOL	With "Error"=TRUE the function block signals that an error is present
ErrorID *	ERROR_CODE	"ErrorID" gives a rough error description.
ErrorIdent *	ERROR_STRUCT	Via the elements: "Table", "Additional1" and "Additional2", "ErrorIdent" shows the exact cause of the error.
Status	MB_SERCOS_DATA_STATUS	The output displays the current status of the parameter. Enum values for "Status" output: "MB_SERCOS_DATA_NOT_SET": Command has not been set "MB_SERCOS_DATA_SETTING": Command has been set "MB_SERCOS_DATA_END_OK": Command completed without error "MB_SERCOS_DATA_INTERRUPTED": Command interrupted "MB_SERCOS_DATA_IN_PROCESS": Command being processed "MB_SERCOS_DATA_BREAK_ERROR": Command aborted with error "MB_SERCOS_DATA_DATUM_FAULT": Invalid operating data

* See note below

Tab. 10-127: *Output Variables of the Firmware Function Block "MB_ReadSercosDataStatus"*



NOTE: For the list of all possible errors, see: "Error Reference Lists"

MX_PLCOpen

Input / output variable (VAR_IN_OUT)	Data type	Description
Axis	AXIS_REF	Defines the axis. The following axes are possible: MPB (IndraDrive) and HDC (HydraulicDrive): Axis1 MPC (IndraDrive): - Axis1 (master) or - Axis2...Axis10 (a total of 9 slaves / remote axes) The remote axes contained in the list "P-0-1801.0.10, CCD: Addresses of projected drives" correspond to the order of Axis2 to Axis10 (Axis2 acts on the drive of list element 0 of P-0-1801.0.10,...) Concerning the access to parameters of remote axes, please observe the notes below!

Tab. 10-128: Input / Output Variable of the Firmware Function Block "MB_ReadSercosDataStatus"



In the case of a remote axis, the task is returned to the service channel (sercos III). Only at a later call can the function block signal "Done"="TRUE". Therefore, the resulting times for the function block calls themselves are always short. The outputs "Done", "Active" and "Error" have to be cyclically polled. The data transmission can take several seconds, depending on the load of the service channel.

As long as the sercos ring is not yet in phase 2 - and temporarily during the transition from phase 3 to phase 4 - it is impossible to exchange data with remote axes. In this case, the function block signals an error.



Accessing parameters of remote axes is possible in the MLD-M system mode, in the CCD basic mode and in the CCD system mode.

10.4.13 MB_ReadStringParameter

With the function block "MB_ReadStringParameter" it is possible to read a parameter the operating data of which has the ASCII format (string).

At the "ParameterNumber" input you have to apply the parameter number of the parameter the operating data of which is to be read. The processing of the function block is started with a rising edge at the "Execute" input.



To test the function block, the instance variable "ActivateCounter" can be used. It counts the positive edges of the "Execute" signal and thereby signals whether the function block is run with an edge.

Local Axis

When the function block is used for a local axis, all characters of the selected ASCII parameter are immediately read when the function block is called. After that, control goes back to the program and TRUE is provided at the "Done" output.



The reading process can take some time, depending on the text length. If necessary, a separate task should be used.

Remote Axes



Accessing parameters of remote axes is possible in the MLD-M system mode, in the CCD basic mode and in the CCD system mode.

When the function block is used for remote axes, the read task is returned to the service channel (sercos III). Only at a later call can the function block signal "Done".

The outputs "Done", "Active" and "Error" have to be cyclically polled.

The data transmission can take several milliseconds, depending on the text length.



As long as the sercos ring is not yet in phase 2 - and temporarily during the transition from phase 3 to phase 4 - it is impossible to exchange data with remote axes. In this case, the function block signals an error.

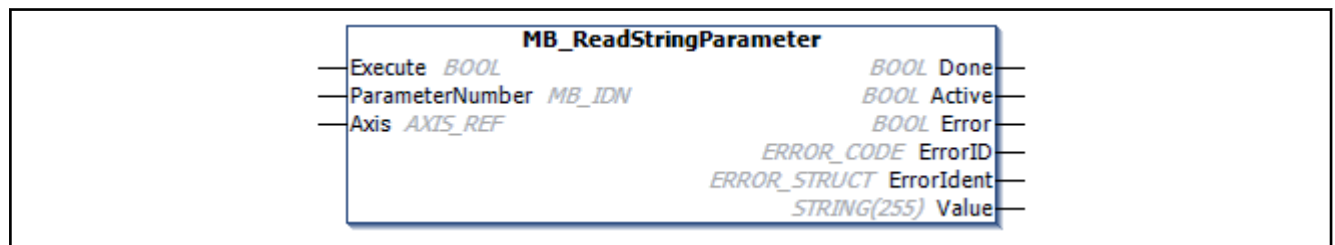


Fig. 10-53: Firmware Function Block "MB_ReadStringParameter"

Input variable	Data type	Description
Execute	BOOL	See "Signal Time Diagrams"
ParameterNumber	DINT	At the "ParameterNumber" input you have to apply the parameter number of the parameter the operating data of which is to be read

Tab. 10-129: Input Variables of the Firmware Function Block "MB_ReadStringParameter"

Output variable	Data type	Description
Done	BOOL	With "Done"=TRUE the function block signals that the processing of the task has been successfully completed
Active	BOOL	For remote axes, "Active"=TRUE signals that the function block has accepted the processing command, for single axes this output is not set!
Error	BOOL	With "Error"=TRUE the function block signals that an error is present
ErrorID *	ERROR_CODE	"ErrorID" gives a rough error description.
ErrorIdent *	ERROR_STRUCT	Via the elements: "Table", "Additional1" and "Additional2", "ErrorIdent" shows the exact cause of the error.
Value	STRING (255)	"Value" provides the operating data of the selected parameter

* See note below

Tab. 10-130: Output Variables of the Firmware Function Block "MB_ReadStringParameter"



NOTE: For the list of all possible errors, see: "Error Reference Lists"

MX_PLCOpen

Input / output variable (VAR_IN_OUT)	Data type	Description
Axis	AXIS_REF	Defines the axis. The following axes are possible: MPB (IndraDrive) and HDC (HydraulicDrive): Axis1 MPC (IndraDrive): - Axis1 (master) or - Axis2...Axis10 (a total of 9 slaves / remote axes) The remote axes contained in the list "P-0-1801.0.10, CCD: Addresses of projected drives" correspond to the order of Axis2 to Axis10 (Axis2 acts on the drive of list element 0 of P-0-1801.0.10,...)

Tab. 10-131: Input / Output Variable of the Firmware Function Block "MB_ReadStringParameter"

10.4.14 MB_ReadUnit

With the function block "MB_ReadUnit" it is possible to read the unit of a parameter.

At the "ParameterNumber" input you have to apply the parameter number of the parameter the unit of which is to be read. The processing of the function block is started with a rising edge at the "Execute" input. Error-free processing is signaled by "Done"=TRUE. Incorrect processing is signaled by "Error"=TRUE. In the case of error, a detailed description of the error is output at the outputs "ErrorID" and "ErrorIdent".

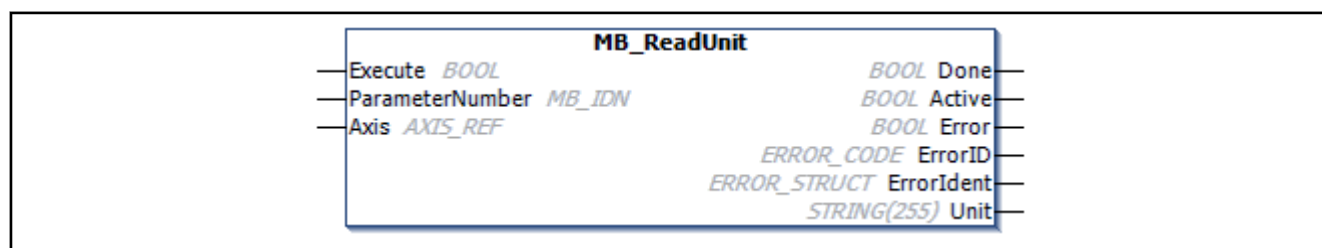


Fig. 10-54: Firmware Function Block "MB_ReadUnit"

Input variable	Data type	Description
Execute	BOOL	See "Signal Time Diagrams"
ParameterNumber	MB_IDN	At the "ParameterNumber" input you have to apply the parameter number of the parameter the unit of which is to be read

Tab. 10-132: Input Variables of the Firmware Function Block "MB_ReadUnit"

Output variable	Data type	Description
Done	BOOL	With "Done"=TRUE the function block signals that the processing of the task has been successfully completed
Active	BOOL	For remote axes, "Active"=TRUE signals that the function block has accepted the processing command, for single axes this output is not set!
Error	BOOL	With "Error"=TRUE the function block signals that an error is present
ErrorID *	ERROR_CODE	"ErrorID" gives a rough error description.

Output variable	Data type	Description
ErrorIdent *	ERROR_STRUCT	Via the elements: "Table", "Additional1" and "Additional2", "ErrorIdent" shows the exact cause of the error.
Unit	STRING (255)	If "Done"=TRUE, "Unit" signals the unit of the selected parameter

* See note below

Tab. 10-133: Output Variables of the Firmware Function Block "MB_ReadUnit"



NOTE: For the list of all possible errors, see: "Error Reference Lists"

Input / output variable (VAR_IN_OUT)	Data type	Description
Axis	AXIS_REF	Defines the axis. The following axes are possible: MPB (IndraDrive) and HDC (HydraulicDrive): Axis1 MPC (IndraDrive): - Axis1 (master) or - Axis2...Axis10 (a total of 9 slaves / remote axes) The remote axes contained in the list "P-0-1801.0.10, CCD: Addresses of projected drives" correspond to the order of Axis2 to Axis10 (Axis2 acts on the drive of list element 0 of P-0-1801.0.10,...) Concerning the access to parameters of remote axes, please observe the notes below!

Tab. 10-134: Input / Output Variable of the Firmware Function Block "MB_ReadUnit"



In the case of a remote axis, the task is returned to the service channel (sercos III). Only at a later call can the function block signal "Done"=TRUE. Therefore, the resulting times for the function block calls themselves are always short. The outputs "Done", "Active" and "Error" have to be cyclically polled. The data transmission can take several seconds, depending on the load of the service channel.

As long as the sercos ring is not yet in phase 2 - and temporarily during the transition from phase 3 to phase 4 - it is impossible to exchange data with remote axes. In this case, the function block signals an error.



Accessing parameters of remote axes is possible in the MLD-M system mode, in the CCD basic mode and in the CCD system mode.

10.4.15 MB_WriteListParameter

With the function block "MB_WriteListParameter" it is possible to write defined values to a list parameter.

The IDN of the parameter to be written has to be applied at the "ParameterNumber" input.

MX_PLCOpen



Depending on parameter "S-0-0269, Storage mode", the changed parameter value is stored in the device-internal, non-volatile memory ("flash"), either immediately or by executing the command "S-0-0264, C2200 Backup working memory procedure command".

NOTICE

Destruction of the fixed value parameter memory by constant activation/execution of the function block!

When the function block is frequently executed, set "S-0-0269, Storage mode" in such a way that changed parameter values are only stored in the device-internal, non-volatile memory ("flash") by executing the command "S-0-0264, C2200 Backup working memory procedure command".



To test the function block "MB_WriteListParameter", the instance variable "ActivateCounter" can be used. It counts the positive edges of the "Execute" signal and thereby signals whether the function block is run with an edge.

Local Axis

When the function block is used for a local axis, all list elements of the applied array are immediately written when the function block is called. After that, control goes back to the program and TRUE is provided at the "Done" output.



You should observe that the writing process can take some time, depending on the volume of data to be written; if necessary, you should use a separate task.

Example: Writing all list elements of parameter P-0-0072 (1024 long words) takes several milliseconds.

It is better to write only the actually required elements; if, for example, 8 elements of parameter "P-0-0072, Cam table 1" are to be written (one element of parameter P-0-0072 consists of 4 bytes), "32" has to be applied to "NoOfBytes" (8*4).

Remote Axes

Accessing parameters of remote axes is possible in the MLD-M system mode, in the CCD basic mode and in the CCD system mode.

When the function block is used for remote axes, the write request is returned to the service channel (sercos III). Only at a later call can the function block signal "Done".

The outputs "Done", "Active" and "Error" have to be cyclically polled.

The data transmission can take several seconds, depending on the data volume.



As long as the sercos ring is not yet in phase 2 - and temporarily during the transition from phase 3 to phase 4 - it is impossible to exchange data with remote axes. In this case, the function block signals an error.



Fig. 10-55: Firmware Function Block "MB_WriteListParameter"

Input variable	Data type	Description
Execute	BOOL	See "Signal Time Diagrams"
ParameterNumber	MB_IDN	At the "ParameterNumber" input, apply the IDN of the list parameter to be written
NoOfBytes	UINT	The number of bytes to be written has to be indicated at the "NoOfBytes" input. Example: NoOfBytes:=SIZEOF(arFeld1)
ValueAdr	POINTER TO BYTE	At the "ValueAdr" input, indicate the address of the field which contains the data/elements to be written for the list parameter. Example: ValueAdr := ADR(arFeld1)

Tab. 10-135: Input Variables of the Firmware Function Block "MB_WriteListParameter"

Output variable	Data type	Description
Done	BOOL	With "Done"=TRUE the function block signals that the processing of the task has been successfully completed
Active	BOOL	For remote axes, "Active"=TRUE signals that the function block has accepted the processing command, for single axes this output is not set!
Error	BOOL	With "Error"=TRUE the function block signals that an error is present
ErrorID *	ERROR_CODE	"ErrorID" gives a rough error description.
ErrorIdent *	ERROR_STRUCT	Via the elements: "Table", "Additional1" and "Additional2", "ErrorIdent" shows the exact cause of the error.

* See note below

Tab. 10-136: Output Variables of the Firmware Function Block "MB_WriteListParameter"



NOTE: For the list of all possible errors, see: "Error Reference Lists"

MX_PLCOpen

Input / output variable (VAR_IN_OUT)	Data type	Description
Axis	AXIS_REF	Defines the axis. The following axes are possible: MPB (IndraDrive) and HDC (HydraulicDrive): Axis1 MPC (IndraDrive): - Axis1 (master) or - Axis2...Axis10 (a total of 9 slaves / remote axes) The remote axes contained in the list "P-0-1801.0.10, CCD: Addresses of projected drives" correspond to the order of Axis2 to Axis10 (Axis2 acts on the drive of list element 0 of P-0-1801.0.10,...)

Tab. 10-137: Input / Output Variable of the Firmware Function Block "MB_WriteList-Parameter"

10.4.16 MB_WriteParameter

With the function block "MB_WriteParameter" it is possible to write a defined value to a parameter. To do this, you have to apply, at the "ParameterNumber" input, the parameter number of the parameter the value of which is to be written. The value which is to be written to the parameter has to be applied at the "Value" input and has to be of the DINT type.



List and string parameters cannot be written with "MB_WriteParameter".



According to the scaling, the parameter has decimal places which have to be taken into account and used for calculation!



Depending on parameter "S-0-0269, Storage mode", the changed parameter value is stored in the device-internal, non-volatile memory ("flash"), either immediately or by executing the command "S-0-0264, C2200 Backup working memory procedure command".

NOTICE

Destruction of the fixed value parameter memory by constant activation/execution of the function block!

When the function block is frequently executed, set "S-0-0269, Storage mode" in such a way that changed parameter values are only stored in the device-internal, non-volatile memory ("flash") by executing the command "S-0-0264, C2200 Backup working memory procedure command".



To test the function block "MB_WriteParameter", the instance variable "ActivateCounter" can be used. It counts the positive edges of the "Execute" signal and thereby signals whether the function block is run with an edge.



Fig. 10-56: Firmware Function Block "MB_WriteParameter"

Input variable	Data type	Description
Execute	BOOL	See "Signal Time Diagrams"
ParameterNumber	MB_IDN	At the "ParameterNumber" input, apply the parameter number of the parameter to be written
Value	DINT	At the "Value" input, apply the value which is to be written to the parameter. Take the scaling of the selected parameter into account

Tab. 10-138: Input Variables of the Firmware Function Block "MB_WriteParameter"

Output variable	Data type	Description
Done	BOOL	With "Done"=TRUE the function block signals that the processing of the task has been successfully completed
Active	BOOL	For remote axes, "Active"=TRUE signals that the function block has accepted the processing command, for single axes this output is not set!
Error	BOOL	With "Error"=TRUE the function block signals that an error is present
ErrorID *	ERROR_CODE	"ErrorID" gives a rough error description.
ErrorIdent *	ERROR_STRUCT	Via the elements: "Table", "Additional1" and "Additional2", "ErrorIdent" shows the exact cause of the error.

* See note below

Tab. 10-139: Output Variables of the Firmware Function Block "MB_WriteParameter"



NOTE: For the list of all possible errors, see: "Error Reference Lists"

Input / output variable (VAR_IN_OUT)	Data type	Description
Axis	AXIS_REF	Defines the axis. The following axes are possible: MPB (IndraDrive) and HDC (HydraulicDrive): Axis1 MPC (IndraDrive): - Axis1 (master) or - Axis2...Axis10 (a total of 9 slaves / remote axes) The remote axes contained in the list "P-0-1801.0.10, CCD: Addresses of projected drives" correspond to the order of Axis2 to Axis10 (Axis2 acts on the drive of list element 0 of P-0-1801.0.10,...) Concerning the access to parameters of remote axes, please observe the notes below!

Tab. 10-140: Input / Output Variable of the Firmware Function Block "MB_WriteParameter"

MX_PLCOpen



In the case of a remote axis, the task is returned to the service channel (sercos III). Only at a later call can the function block signal "Done"="TRUE". Therefore, the resulting times for the function block calls themselves are always short. The outputs "Done", "Active" and "Error" have to be cyclically polled. The data transmission can take several seconds, depending on the load of the service channel.

As long as the sercos ring is not yet in phase 2 - and temporarily during the transition from phase 3 to phase 4 - it is impossible to exchange data with remote axes. In this case, the function block signals an error.



Accessing parameters of remote axes is possible in the MLD-M system mode, in the CCD basic mode and in the CCD system mode.

10.4.17 MB_WriteRealListParameter

With the function block "MB_WriteRealListParameter" it is possible to write defined values to a list parameter.

The IDN of the list parameter to be written has to be applied at the "ParameterNumber" input.



Depending on parameter "S-0-0269, Storage mode", the changed parameter value is stored in the device-internal, non-volatile memory ("flash"), either immediately or by executing the command "S-0-0264, C2200 Backup working memory procedure command".

NOTICE

Destruction of the fixed value parameter memory by constant activation/execution of the function block!

When the function block is frequently executed, set "S-0-0269, Storage mode" in such a way that changed parameter values are only stored in the device-internal, non-volatile memory ("flash") by executing the command "S-0-0264, C2200 Backup working memory procedure command".



To test the function block "MB_WriteRealListParameter", the instance variable "ActivateCounter" can be used. It counts the positive edges of the "Execute" signal and thereby signals whether the function block is run with an edge.

The inputs "NoOfBytes" and "ValueAdr" should be used as follows:

Example:

```
inst_MB_WriteRealListParameter : MB_WriteRealListParameter;
arFeld1 : ARRAY[1..16] OF REAL;
inst_MB_WriteRealListParameter(...,
NoOfBytes:=SIZEOF(arFeld1), ValueAdr:=ADR(arFeld1), ...);
```

Local Axis

When the function block is used for a local axis, all list elements of the applied array are immediately written when the function block is called. After

that, control goes back to the program and TRUE is provided at the "Done" output.



You should observe that the writing process can take some time, depending on the volume of data to be written. If necessary, a separate task should be used.

Example: Writing all list elements of parameter P-0-0072 (1024 long words) takes several milliseconds.

Remote Axes



Accessing parameters of remote axes is possible in the MLD-M system mode, in the CCD basic mode and in the CCD system mode.

When the function block is used for remote axes, the write request is returned to the service channel (sercos III). Only at a later call can the function block signal "Done"=TRUE.

The outputs "Done", "Active" and "Error" have to be cyclically polled.

The data transmission can take several seconds, depending on the data volume.



As long as the sercos ring is not yet in phase 2 - and temporarily during the transition from phase 3 to phase 4 - it is impossible to exchange data with remote axes. In this case, the function block signals an error.

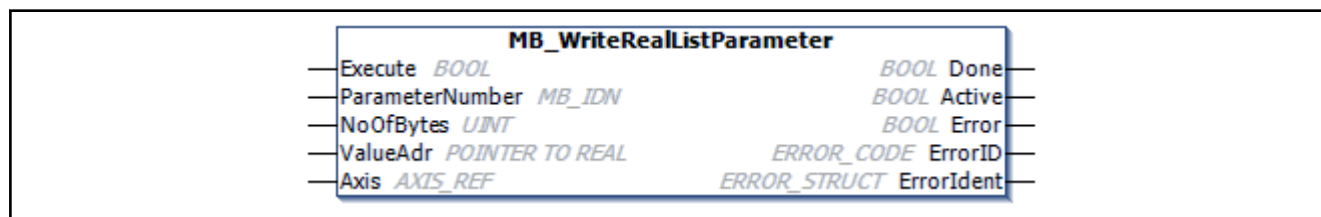


Fig. 10-57: Firmware Function Block "MB_WriteRealListParameter"

Input variable	Data type	Description
Execute	BOOL	See "Signal Time Diagrams"
ParameterNumber	MB_IDN	At the "ParameterNumber" input, apply the IDN of the list parameter to be written
NoOfBytes	UINT	The number of bytes to be written has to be indicated at the "NoOfBytes" input Example: NoOfBytes := SIZEOF(arFeld1); Note: As the list elements are stored in a REAL array (REAL data type = 4-byte), the NoOfBytes is calculated as follows: NoOfBytes = 4 * (number of elements)
ValueAdr	POINTER TO REAL	At the "ValueAdr" input indicate the initial address of the field to which the bytes that have been read are to be written. Example: ValueAdr := ADR(arFeld1); Note: The applied field must be of the REAL data type.

Tab. 10-141: Input Variables of the Firmware Function Block "MB_WriteRealList-Parameter"

MX_PLCOpen

Output variable	Data type	Description
Done	BOOL	With "Done"=TRUE the function block signals that the processing of the task has been successfully completed
Active	BOOL	For remote axes, "Active"=TRUE signals that the function block has accepted the processing command, for single axes this output is not set!
Error	BOOL	With "Error"=TRUE the function block signals that an error is present
ErrorID *	ERROR_CODE	"ErrorID" gives a rough error description.
ErrorIdent *	ERROR_STRUCT	Via the elements: "Table", "Additional1" and "Additional2", "ErrorIdent" shows the exact cause of the error.

* See note below

Tab. 10-142: *Output Variables of the Firmware Function Block "MB_WriteRealList-Parameter"*



NOTE: For the list of all possible errors, see: "Error Reference Lists"

Input / output variable (VAR_IN_OUT)	Data type	Description
Axis	AXIS_REF	Defines the axis. The following axes are possible: MPB (IndraDrive) and HDC (HydraulicDrive): Axis1 MPC (IndraDrive): - Axis1 (master) or - Axis2...Axis10 (a total of 9 slaves / remote axes) The remote axes contained in the list "P-0-1801.0.10, CCD: Addresses of projected drives" correspond to the order of Axis2 to Axis10 (Axis2 acts on the drive of list element 0 of P-0-1801.0.10,...)

Tab. 10-143: *Input / Output Variable of the Firmware Function Block "MB_WriteRealListParameter"*

10.4.18 MB_WriteRealParameter

With the function block "MB_WriteRealParameter" you can write a defined value to the parameter belonging to the parameter number. The data format of the parameter to be written must be DEC_OV, DEC_MV or FLOAT; parameters of the data formats BIN, HEX and IDN cannot be written.



Depending on parameter "S-0-0269, Storage mode", the changed parameter value is stored in the device-internal, non-volatile memory ("flash"), either immediately or by executing the command "S-0-0264, C2200 Backup working memory procedure command".

NOTICE

Destruction of the fixed value parameter memory by constant activation/execution of the function block!

When the function block is frequently executed, set "S-0-0269, Storage mode" in such a way that changed parameter values are only stored in the device-internal, non-volatile memory ("flash") by executing the command "S-0-0264, C2200 Backup working memory procedure command".



List and string parameters cannot be written with the function block "MB_WriteRealParameter"; to write list parameters, please use the function block "MB_WriteListParameter" or "MB_WriteRealListParameter".



Fig. 10-58: Firmware Function Block "MB_WriteRealParameter"

Input variable	Data type	Description
Execute	BOOL	See "Signal Time Diagrams"
ParameterNumber	MB_IDN	At the "ParameterNumber" input, apply the IDN of the list parameter to be written
Value	REAL	At the "Value" input, apply the value which is to be written to the parameter. Take the scaling of the selected parameter into account

Tab. 10-144: Input Variables of the Firmware Function Block "MB_WriteRealParameter"

Output variable	Data type	Description
Done	BOOL	With "Done"=TRUE the function block signals that the processing of the task has been successfully completed
Active	BOOL	For remote axes, "Active"=TRUE signals that the function block has accepted the processing command, for single axes this output is not set!
Error	BOOL	With "Error"=TRUE the function block signals that an error is present
ErrorID *	ERROR_CODE	"ErrorID" gives a rough error description.
ErrorIdent *	ERROR_STRUCT	Via the elements: "Table", "Additional1" and "Additional2", "ErrorIdent" shows the exact cause of the error.

* See note below

Tab. 10-145: Output Variables of the Firmware Function Block "MB_WriteRealParameter"



NOTE: For the list of all possible errors, see: "Error Reference Lists"

MX_PLCOpen

Input / output variable (VAR_IN_OUT)	Data type	Description
Axis	AXIS_REF	Defines the axis. The following axes are possible: MPB (IndraDrive) and HDC (HydraulicDrive): Axis1 MPC (IndraDrive): - Axis1 (master) or - Axis2...Axis10 (a total of 9 slaves / remote axes) The remote axes contained in the list "P-0-1801.0.10, CCD: Addresses of projected drives" correspond to the order of Axis2 to Axis10 (Axis2 acts on the drive of list element 0 of P-0-1801.0.10,...) Concerning the access to parameters of remote axes, please observe the notes below!

Tab. 10-146: *Input / Output Variable of the Firmware Function Block "MB_WriteRealParameter"*



In the case of a remote axis, the task is returned to the service channel (sercos III). Only at a later call can the function block signal "Done"="TRUE". Therefore, the resulting times for the function block calls themselves are always short. The outputs "Done", "Active" and "Error" have to be cyclically polled. The data transmission can take several seconds, depending on the load of the service channel.

As long as the sercos ring is not yet in phase 2 - and temporarily during the transition from phase 3 to phase 4 - it is impossible to exchange data with remote axes. In this case, the function block signals an error.



Accessing parameters of remote axes is possible in the MLD-M system mode, in the CCD basic mode and in the CCD system mode.

10.4.19 MB_WriteStringParameter

With the function block "MB_WriteStringParameter" it is possible to write a parameter the operating data of which has the ASCII format (string).

At the "ParameterNumber" input you have to apply the parameter number of the parameter the operating data of which is to be written. The processing of the function block is started with a rising edge at the "Execute" input.



Depending on parameter "S-0-0269, Storage mode", the changed parameter value is stored in the device-internal, non-volatile memory ("flash"), either immediately or by executing the command "S-0-0264, C2200 Backup working memory procedure command".

NOTICE

Destruction of the fixed value parameter memory by constant activation/execution of the function block!

When the function block is frequently executed, set "S-0-0269, Storage mode" in such a way that changed parameter values are only stored in the device-internal, non-volatile memory ("flash") by executing the command "S-0-0264, C2200 Backup working memory procedure command".



To test the function block, the instance variable "ActivateCounter" can be used. It counts the positive edges of the "Execute" signal and thereby signals whether the function block is run with an edge.

Local Axis

When the function block is used for a local axis, all characters of the selected ASCII parameter are immediately written when the function block is called. After that, control goes back to the program and TRUE is provided at the "Done" output.



The writing process can take some time, depending on the text length. If necessary, a separate task should be used.

Remote Axes



Accessing parameters of remote axes is possible in the MLD-M system mode, in the CCD basic mode and in the CCD system mode.

When the function block is used for remote axes, the write request is returned to the service channel (sercos III). Only at a later call can the function block signal "Done".

The outputs "Done", "Active" and "Error" have to be cyclically polled.

The data transmission can take several milliseconds, depending on the text length.



As long as the sercos ring is not yet in phase 2 - and temporarily during the transition from phase 3 to phase 4 - it is impossible to exchange data with remote axes. In this case, the function block signals an error.

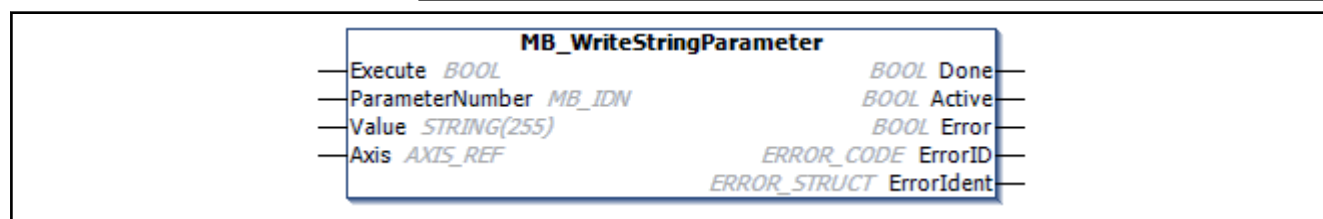


Fig. 10-59: Firmware Function Block "MB_WriteStringParameter"

Input variable	Data type	Description
Execute	BOOL	See "Signal Time Diagrams"
ParameterNumber	DINT	At the "ParameterNumber" input you have to apply the parameter number of the parameter the operating data of which is to be written

Tab. 10-147: Input Variables of the Firmware Function Block "MB_WriteStringParameter"

MX_PLCOpen

Output variable	Data type	Description
Done	BOOL	With "Done"=TRUE the function block signals that the processing of the task has been successfully completed
Active	BOOL	For remote axes, "Active"=TRUE signals that the function block has accepted the processing command, for single axes this output is not set!
Error	BOOL	With "Error"=TRUE the function block signals that an error is present
ErrorID *	ERROR_CODE	"ErrorID" gives a rough error description.
ErrorIdent *	ERROR_STRUCT	Via the elements: "Table", "Additional1" and "Additional2", "ErrorIdent" shows the exact cause of the error.
Value	STRING (255)	"Value" provides the operating data of the selected parameter

* See note below
Tab. 10-148: Output Variables of the Firmware Function Block "MB_WriteString-Parameter"

 NOTE: For the list of all possible errors, see: "Error Reference Lists"

Input / output variable (VAR_IN_OUT)	Data type	Description
Axis	AXIS_REF	Defines the axis. The following axes are possible: MPB (IndraDrive) and HDC (HydraulicDrive): Axis1 MPC (IndraDrive): - Axis1 (master) or - Axis2...Axis10 (a total of 9 slaves / remote axes) The remote axes contained in the list "P-0-1801.0.10, CCD: Addresses of projected drives" correspond to the order of Axis2 to Axis10 (Axis2 acts on the drive of list element 0 of P-0-1801.0.10,...)

Tab. 10-149: Input / Output Variable of the Firmware Function Block "MB_WriteStringParameter"

10.4.20 MX_Command

The function block "MX_Command" is used to execute a drive command.
The command is activated or deactivated with the "Enable" input. If TRUE is provided at the input, the command is started by writing "3" (dec.). In the case of FALSE, the command is aborted/terminated by writing "0".



Fig. 10-60: Firmware Function Block "MX_Command"

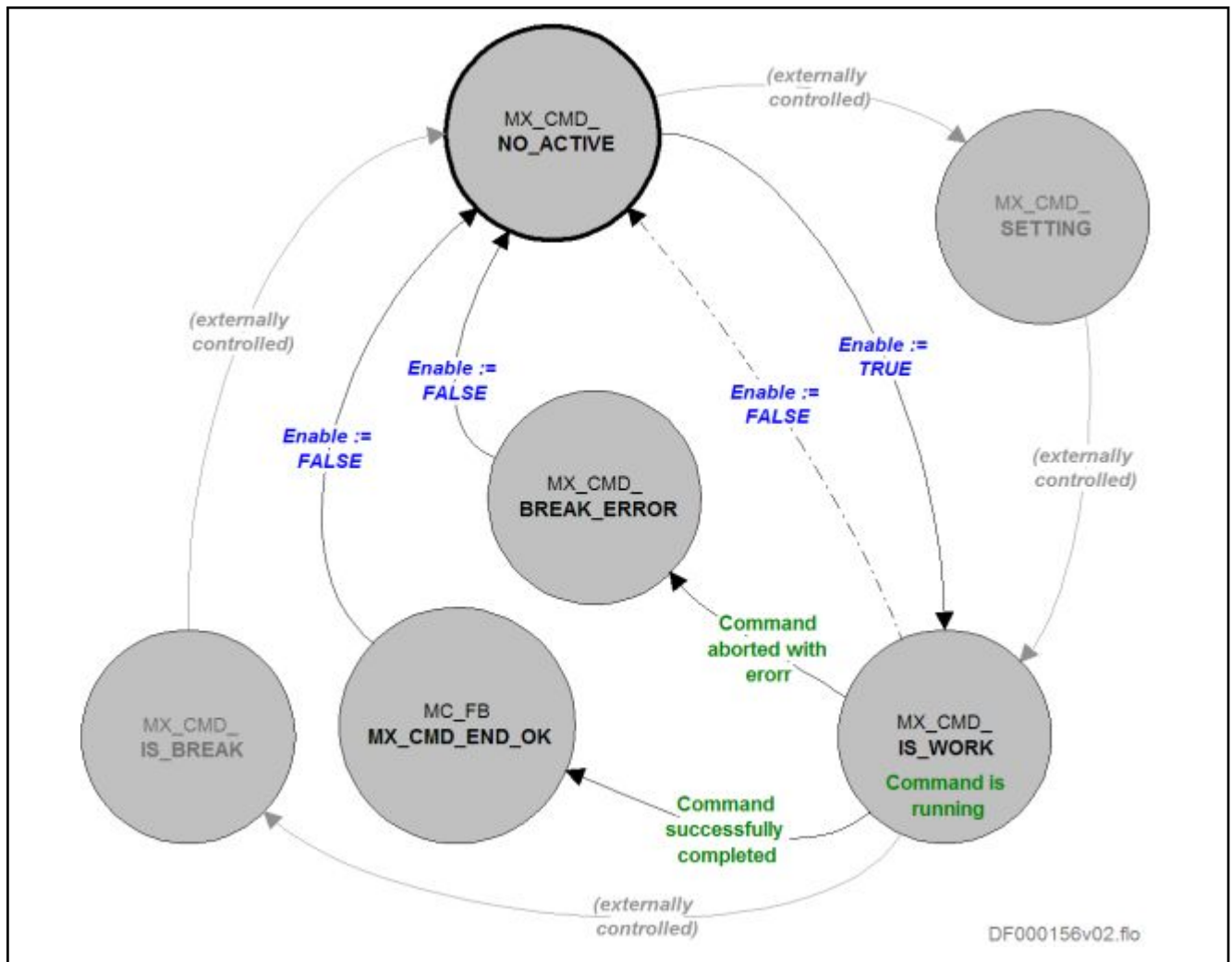


Fig.10-61: Executing a Drive Command



In the case of a remote axis, the task is returned to the service channel (sercos III). Only at a later call can "CmdStatus" signal "MX_CMD_END_OK". Therefore, the resulting times for the function block calls themselves are always short. The outputs "CmdStatus" and "Error" have to be cyclically polled. The data transmission can take several seconds, depending on the load of the service channel.

As long as the sercos ring is not yet in phase 2 - and temporarily during the transition from phase 3 to phase 4 - it is impossible to exchange data with remote axes. In this case, the function block signals an error.

MX_PLCOpen

Input variable	Data type	Description
Enable	BOOL	The function block is status-controlled. The command is started with "Enable"=TRUE, it is cleared with "Enable"=FALSE. After the input has been set to TRUE, there normally is a waiting time until the output displays "CmdStatus"=MX_CMD_END_OK or "CmdStatus"=MX_CMD_BREAK_ERROR. After the input has been reset, there should be a waiting time until the output displays "CmdStatus"=MX_CMD_NO_ACTIVE. (See also command description and "Signal Time Diagrams")
CmdIdent	MB_IDN	IDN of the command

Tab. 10-150: Input Variables of the Firmware Function Block "MX_Command"

Output variable	Data type	Description
CmdStatus	MX_CMD_STATUS	The output displays the current status of the command. Enum values for "CmdStatus" output: "MX_CMD_NO_ACTIVE": Command has not been set "MX_CMD_SETTING": Command has been set "MX_CMD_END_OK": Command completed without error "MX_CMD_IS_BREAK": Command interrupted "MX_CMD_IS_WORK": Command being processed "MX_CMD_BREAK_ERROR": Command aborted with error
Error	BOOL	With "Error"=TRUE the function block signals that an error is present
ErrorID *	ERROR_CODE	"ErrorID" gives a rough error description.
ErrorIdent *	ERROR_STRUCT	Via the elements: "Table", "Additional1" and "Additional2", "ErrorIdent" shows the exact cause of the error.

* See note below

Tab. 10-151: Output Variables of the Firmware Function Block "MX_Command"



NOTE: For the list of all possible errors, see: "Error Reference Lists"

Input / output variable (VAR_IN_OUT)	Data type	Description
Axis	AXIS_REF	Defines the axis. The following axes are possible: MPB (IndraDrive) and HDC (HydraulicDrive): Axis1 MPC (IndraDrive): - Axis1 (master) or - Axis2...Axis10 (a total of 9 slaves / remote axes) The remote axes contained in the list "P-0-1801.0.10, CCD: Addresses of projected drives" correspond to the order of Axis2 to Axis10 (Axis2 acts on the drive of list element 0 of P-0-1801.0.10,...) Note: The input is only applied at a rising edge of the Enable input. If several axes are to be controlled, a separate instance should be used for each axis.

Tab. 10-152: Input / Output Variable of the Firmware Function Block "MX_Command"

10.4.21 MX_fReadParamDINT

The firmware function "MX_fReadParamDINT" is used to read a parameter value and return it as DINT.

With this function it is at first impossible to recognize whether the reading of the parameter was successful or not. The function block "MX_ReadParamDINT" is suited for this purpose.

For unsigned parameters with 32 bits the return value can simply be converted with the function "DINT_TO_UDINT".



Fig. 10-62: Firmware Function "MX_fReadParamDINT"

Input variable	Data type	Description
Ident	MB_IDN	At the "Ident" input apply the IDN of the parameter the value of which is to be read
ListElem	INT	It is possible to read the value of a single parameter, but also the values of list parameters. To read a single parameter , apply "0" to "ListElem". For list parameters , the zeroth element is read with "0", the first element with "1", the second element with "2" etc. With "-1" at "ListElem" it is the actual length, with "-2" it is the maximum length of a list parameter which is read
Axis	AXIS_REF	Axis number (only the global variable "Axis1" can be transmitted here)

Tab. 10-153: Input Variables of the Firmware Function "MX_fReadParamDINT"

Return value	Data type	Description
MX_fReadParamDINT	DINT	Provides the value of the element specified with "Ident" and "ListElem". Note: The value "0" is returned if the parameter or list element cannot be read

Tab. 10-154: Return Value of the Firmware Function "MX_fReadParamDINT"

10.4.22 MX_fReadStringParam

With the firmware function "MX_fReadStringParam" it is possible to read a parameter the operating data of which has the ASCII format (string).



In addition to "MaxLen" a closing 0-character is copied. The return value returns the characters actually read (without 0).

When the function is called, the complete text is immediately read. Afterwards, control goes back to the program.



The reading process can take some time, depending on the text length. If necessary, a separate task should be used.

MX_PLCOpen

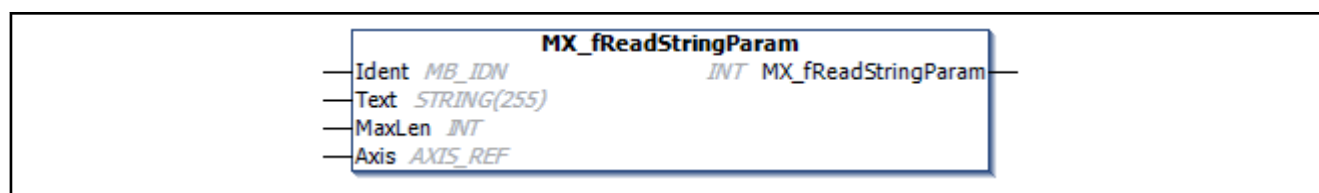


Fig. 10-63: Firmware Function "MX_fReadStringParam"

Input variable	Data type	Description
Ident	MB_IDN	IDN of the parameter
Text	STRING (255)	Text buffer 255 characters
MaxLen	INT	Number of characters to be read (without closing 0)
Axis	AXIS_REF	Axis number (only the global variable "Axis1" can be transmitted here)

Tab. 10-155: Input Variables of the Firmware Function "MX_fReadStringParam"

Return value	Data type	Description
MX_fReadStringParam	INT	Returns the characters actually read (without 0)

Tab. 10-156: Return Value of the Firmware Function "MX_fReadStringParam"

10.4.23 MX_fSetParamLimits

With the firmware function "MX_fSetParamLimits" it is possible to set the limit values of the Global Register PLC parameters.



By switching off or by "load defaults procedure for PLC" [C0730 Load def. proc. com. (load defaults procedure for PLC)], the limit values are set to their default values again.



This function mustn't be called from the system event "PositionLoop". When the function is called from "PositionLoop", a PLC exception occurs and the function returns the value "0".



Fig. 10-64: Firmware Function "MX_fSetParamLimits"

Input variable	Data type	Description
Ident	MB_IDN	A constant from the global variables can be transmitted here
MinValue	DINT	Minimum value to be set
MaxValue	DINT	Maximum value to be set

Tab. 10-157: Input Variables of the Firmware Function "MX_fSetParamLimits"

Return value	Data type	Description
MX_fSetParamLimits	UINT	The function can return the following values: 0: Limit values were successfully set 1: Error, inadmissible IDN was given

Tab. 10-158: Return Value of the Firmware Function "MX_fSetParamLimits"

10.4.24 MX_fSetParamName

With the firmware function "MX_fSetParamName" it is possible to set the names of the Global Register PLC parameters and of the PLC I/O parameters.



By switching off, the names of the Global Register PLC parameters and of the PLC I/O parameters are set to their default values again.

When the names of the Global Register PLC parameters or of the PLC I/O parameters were changed, they are set to the English names by "load defaults procedure for PLC" [C0730 Load def. proc. com. (load defaults procedure for PLC)]. This behavior is due to technical reasons.



This function mustn't be called from the system event "PositionLoop". When the function is called from "PositionLoop", a PLC exception occurs and the function returns the value "0".



Fig. 10-65: Firmware Function "MX_fSetParamName"

Input variable	Data type	Description
Ident	MB_IDN	A constant from the global variables can be transmitted here
NewName	STRING (60)	Name to be set (maximum length: 60 characters)

Tab. 10-159: Input Variables of the Firmware Function "MX_fSetParamName"

Return value	Data type	Description
MX_fSetParamName	UINT	The function can return the following values: 0: Name was successfully set 1: Error, inadmissible IDN was given

Tab. 10-160: Return Value of the Firmware Function "MX_fSetParamName"

10.4.25 MX_fSetParamUnit

With the firmware function "MX_fSetParamUnit" it is possible to set the units of the Global Register PLC parameters.



By switching off or by "load defaults procedure for PLC" [C0730 Load def. proc. com. (load defaults procedure for PLC)], the units are set to their default values again.



This function mustn't be called from the system event "PositionLoop". When the function is called from "PositionLoop", a PLC exception occurs and the function returns the value "0".

MX_PLCOpen



Fig.10-66: Firmware Function "MX_fSetParamUnit"

Input variable	Data type	Description
Ident	MB_IDN	A constant from the global variables can be transmitted here
NewUnit	STRING (20)	Unit to be set (maximum length: 20 characters)

Tab.10-161: Input Variables of the Firmware Function "MX_fSetParamUnit"

Return value	Data type	Description
MX_fSetParamUnit	UINT	The function can return the following values: 0: Unit was successfully set 1: Error, inadmissible IDN was given

Tab.10-162: Return Value of the Firmware Function "MX_fSetParamUnit"

10.4.26 MX_fWriteParamDINT

The firmware function "MX_fWriteParamDINT" is used to write data to a parameter.

It is possible to write in unbuffered or buffered form.

In the case of buffered writing, the parameter values every time are transferred to the non-volatile memory. As this takes much calculating time, data should only be written in buffered form if necessary.

For list parameters only the last write access should take place with "WriteBuffered"=TRUE, as the entire list is buffered at each access.

While a parameter is buffered, another access with buffering will fail (ACCESS_ERROR).

NOTICE

Destruction of the device-internal, non-volatile memory ("Flash") by constant buffered writing of the parameter values!

When the function block is frequently executed, write parameter values in buffered form ("WriteBuffered"=TRUE) only if necessary. In the case of list parameters, for example, only the last write access should be carried out with "WriteBuffered"=TRUE.

With this function it is at first impossible to recognize whether the writing of the parameter was successful or not. The function block "MX_WriteParamDINT" is suited for this purpose.

For unsigned parameters with 32 bits the input value "Value" can be converted with the function "UDINT_TO_DINT".

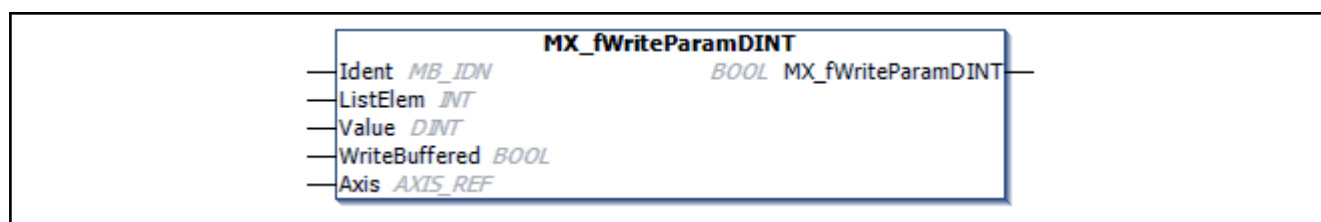


Fig. 10-67: Firmware Function "MX_fWriteParamDINT"

Input variable	Data type	Description
Ident	MB_IDN	IDN of the specified parameter
ListElem	INT	List element number: For list parameters starting with "0" and for single parameter always "0" "-1" for writing the actual length (only allowed for list parameters)
Value	INT	Value which is to be written to the selected element
WriteBuffered	BOOL	"WriteBuffered"=TRUE: Value is buffered; "WriteBuffered"=FALSE: Value is not buffered
Axis	AXIS_REF	Axis number (only the global variable "Axis1" can be transmitted here)

Tab. 10-163: Input Variables of the Firmware Function "MX_fWriteParamDINT"

Return value	Data type	Description
MX_fWriteParamDINT	BOOL	Dummy output without significance

Tab. 10-164: Return Value of the Firmware Function "MX_fWriteParamDINT"

10.4.27 MX_fWriteStringParam

The firmware function "MX_fWriteStringParam" is used to write data to a text parameter.

It is possible to write in unbuffered or buffered form.



In the case of buffered writing, the parameter values every time are transferred to the non-volatile memory. As this takes much calculating time, data should only be written in buffered form if necessary.

While a parameter is buffered, another access with buffering will fail (ACCESS_ERROR).

When the function is called, the complete text is immediately written. Afterwards, control goes back to the program.



The writing process can take some time, depending on the text length. If necessary, a separate task should be used.

NOTICE

Destruction of the device-internal, non-volatile memory ("Flash") by constant buffered writing of the parameter values!

When the function block is frequently executed, write parameter values in buffered form ("WriteBuffered"=TRUE) only if necessary. In the case of list parameters, for example, only the last write access should be carried out with "WriteBuffered"=TRUE.

MX_PLCOpen

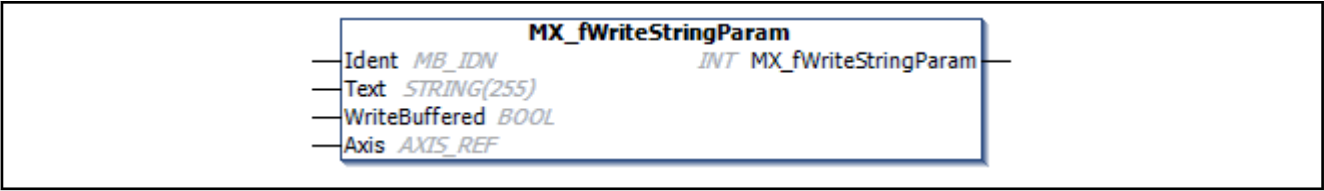


Fig. 10-68: Firmware Function "MX_fWriteStringParam"

Input variable	Data type	Description
Ident	MB_IDN	IDN of the specified parameter
Text	STRING (255)	Text string
WriteBuffered	BOOL	"WriteBuffered"=TRUE: Value is buffered "WriteBuffered"=FALSE: Value is not buffered
Axis	AXIS_REF	Axis number (only the global variable "Axis1" can be transmitted here)

Tab. 10-165: Input Variables of the Firmware Function "MX_fWriteStringParam"

Return value	Data type	Description
MX_fWriteStringParam	INT	The return value returns the characters actually written (without 0)

Tab. 10-166: Return Value of the Firmware Function "MX_fWriteStringParam"

10.4.28 MX_ReadParamDINT

The function block "MX_ReadParamDINT" is used to read a parameter value and return it as DINT.

Among other things, the function block gives a feedback as to whether the reading was successful.

For unsigned parameters with 32 bits the return value can be converted with the function "DINT_TO_UDINT".

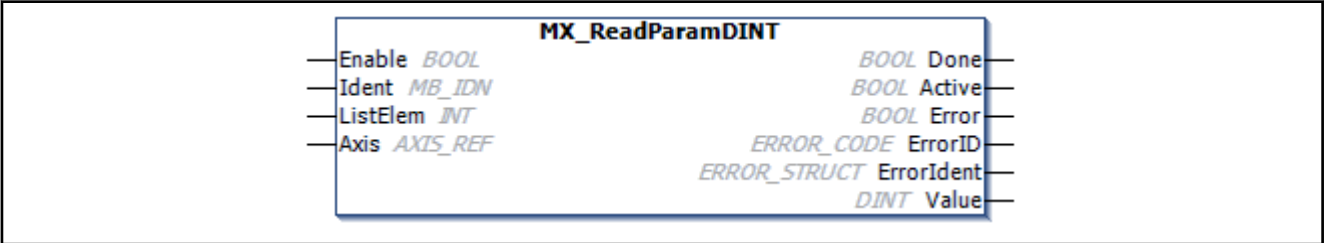


Fig. 10-69: Firmware Function Block "MX_ReadParamDINT"

Input variable	Data type	Description
Enable	BOOL	The function block is activated with "Enable"=TRUE. All outputs are reset with "Enable"=FALSE
Ident	DINT	IDN of the parameter
ListElem	INT	List element number: For list parameters starting with "0" and for single parameter always "0" "-1" for reading the actual length (only allowed for list parameters) "-2" for reading the maximum length (only allowed for list parameters)

Tab. 10-167: Input Variables of the Firmware Function Block "MX_ReadParam-DINT"

Output variable	Data type	Description
Done	BOOL	With "Done"=TRUE the function block signals that the processing of the task has been successfully completed
Active	BOOL	For remote axes, "Active"=TRUE signals that the function block has accepted the processing command, for single axes this output is not set!
Error	BOOL	With "Error"=TRUE the function block signals that an error is present
ErrorID *	ERROR_CODE	"ErrorID" gives a rough error description.
ErrorIdent *	ERROR_STRUCT	Via the elements: "Table", "Additional1" and "Additional2", "ErrorIdent" shows the exact cause of the error.
Value	DINT	If the output "Done"=TRUE, "Value" signals the value of the selected element

* See note below

Tab. 10-168: *Output Variables of the Firmware Function Block "MX_ReadParamDINT"*



NOTE: For the list of all possible errors, see: "Error Reference Lists"

Input / output variable (VAR_IN_OUT)	Data type	Description
Axis	AXIS_REF	Axis number (only the global variable "Axis1" can be transmitted here)

Tab. 10-169: *Input / Output Variable of the Firmware Function Block "MX_ReadParamDINT"*

10.4.29 MX_WriteParamDINT

The firmware function block "MX_WriteParamDINT" is used to write a parameter. It is possible to write in unbuffered or buffered form.



In the case of buffered writing, the parameter values every time are transferred to the non-volatile memory. As this takes much calculating time, data should only be written in buffered form if necessary.

For list parameters only the last write access should take place with "WriteBuffered"=TRUE, as the entire list is buffered at each access.

While a parameter is buffered, another access with buffering will fail (ACCESS_ERROR).

NOTICE

Destruction of the device-internal, non-volatile memory ("Flash") by constant buffered writing of the parameter values!

When the function block is frequently executed, write parameter values in buffered form ("WriteBuffered"=TRUE) only if necessary. In the case of list parameters, for example, only the last write access should be carried out with "WriteBuffered"=TRUE.

For unsigned parameters with 32 bits the input value "Value" can simply be converted with the function "UDINT_TO_DINT".

MX_PLCOpen

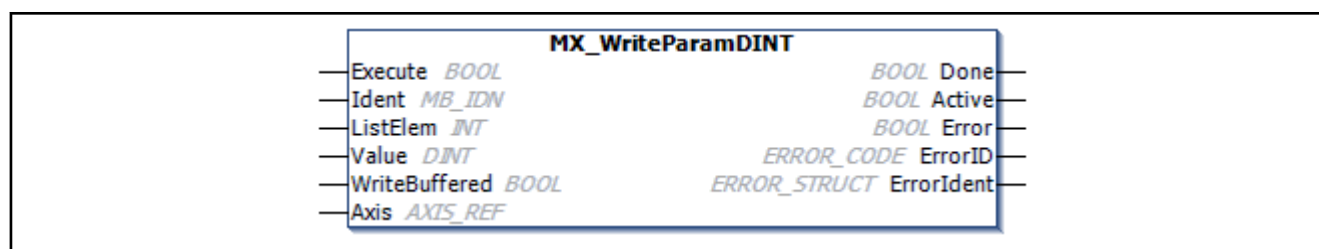


Fig.10-70: Firmware Function Block "MX_WriteParamDINT"

Input variable	Data type	Description
Execute	BOOL	See "Signal Time Diagrams"
Ident	MB_IDN	IDN of the parameter
ListElem	INT	With the "ListElem" input you have to specify the element which is to be written. List element number: For list parameters starting with "0" and for single parameter always "0". "-1" for writing the actual length (only allowed for list parameters)
Value	DINT	At the "Value" input, apply the value which is to be written to the parameter. Take the scaling of the selected parameter into account. According to the scaling, the parameter has decimal places which have to be taken into account and used for calculation
WriteBuffered	BOOL	With "WriteBuffered"=TRUE the value is written in buffered form, if the parameter has been designed for this purpose. With FALSE the parameter is always written in unbuffered form

Tab.10-170: Input Variables of the Firmware Function Block "MX_WriteParamDINT"

Output variable	Data type	Description
Done	BOOL	With "Done"=TRUE the function block signals that the processing of the task has been successfully completed
Active	BOOL	For remote axes, "Active"=TRUE signals that the function block has accepted the processing command, for single axes this output is not set!
Error	BOOL	With "Error"=TRUE the function block signals that an error is present
ErrorID *	ERROR_CODE	"ErrorID" gives a rough error description.
ErrorIdent *	ERROR_STRUCT	Via the elements: "Table", "Additional1" and "Additional2", "ErrorIdent" shows the exact cause of the error.

* See note below

Tab.10-171: Output Variables of the Firmware Function Block "MX_WriteParamDINT"



NOTE: For the list of all possible errors, see: "Error Reference Lists"

Input / output variable (VAR_IN_OUT)	Data type	Description
Axis	AXIS_REF	Axis number (only the global variable "Axis1" can be transmitted here)

Tab.10-172: Input / Output Variable of the Firmware Function Block "MX_WriteParamDINT"

10.5 "Scaling" - Functions for Scaling

10.5.1 MX_fDINT_AccTo_REAL

The firmware function "MX_fDINT_AccTo_REAL" converts an acceleration value in the fixed point format of the DINT data type into a REAL value according to the scaling which has been set.

If the axis reference is invalid, 0.0 is returned.

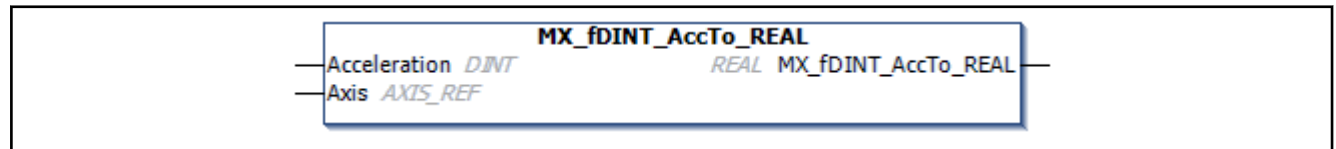


Fig. 10-71: Firmware Function "MX_fDINT_AccTo_REAL"

Input variable	Data type	Description
Acceleration	DINT	Acceleration value in fixed-point DINT format
Axis	AXIS_REF	Axis address: The corresponding axis, e.g. "Axis1" from the global axis variables "AxisAddressing" of "MX_Base.library (IndraDrive)" or "MY_Base.library (HydraulicDrive)" can be transmitted here.

Tab. 10-173: Input Variables of the Firmware Function "MX_fDINT_AccTo_REAL"

Return value	Data type	Description
MX_fDINT_AccTo_REAL	REAL	Returns an acceleration value in the REAL format according to the scaling which has been set

Tab. 10-174: Return Value of the Firmware Function "MX_fDINT_AccTo_REAL"

10.5.2 MX_fDINT_DistTo_REAL

The firmware function "MX_fDINT_DistTo_REAL" converts a position value in the fixed point format of the DINT data type into a REAL value according to the scaling which has been set.

If the axis reference is invalid, 0.0 is returned.

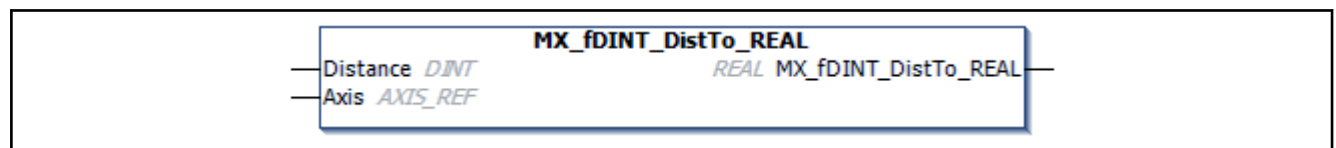


Fig. 10-72: Firmware Function "MX_fDINT_DistTo_REAL"

Input variable	Data type	Description
Distance	DINT	Position value in fixed-point DINT format
Axis	AXIS_REF	Axis address: The corresponding axis, e.g. "Axis1" from the global axis variables "AxisAddressing" of "MX_Base.library (IndraDrive)" or "MY_Base.library (HydraulicDrive)" can be transmitted here.

Tab. 10-175: Input Variables of the Firmware Function "MX_fDINT_DistTo_REAL"

Return value	Data type	Description
MX_fDINT_DistTo_REAL	REAL	Returns a position value in the REAL format according to the scaling which has been set

Tab. 10-176: Return Value of the Firmware Function "MX_fDINT_DistTo_REAL"

MX_PLCOpen

10.5.3 MX_fDINT_VelTo_REAL

The firmware function "MX_fDINT_VelTo_REAL" converts a velocity value in the fixed point format of the DINT data type into a REAL value according to the scaling which has been set.

If the axis reference is invalid, 0 is returned.



Fig. 10-73: Firmware Function "MX_fDINT_VelTo_REAL"

Input variable	Data type	Description
Velocity	DINT	Velocity value in fixed-point DINT format
Axis	AXIS_REF	Axis address: The corresponding axis, e.g. "Axis1" from the global axis variables "AxisAddressing" of "MX_Base.library (IndraDrive)" or "MY_Base.library (HydraulicDrive)" can be transmitted here.

Tab. 10-177: Input Variables of the Firmware Function "MX_fDINT_VelTo_REAL"

Return value	Data type	Description
MX_fDINT_VelTo_REAL	REAL	Returns a velocity value in the REAL format according to the scaling which has been set

Tab. 10-178: Return Value of the Firmware Function "MX_fDINT_VelTo_REAL"

10.5.4 MX_fREAL_AccTo_DINT

The firmware function "MX_fREAL_AccTo_DINT" converts an acceleration value in the REAL format into a fixed-point DINT value according to the scaling which has been set.

If the axis reference is invalid, 0.0 is returned.

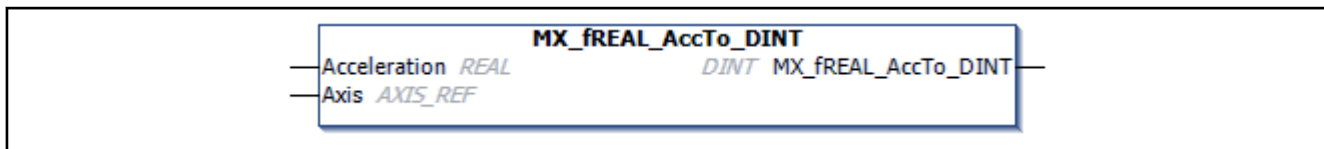


Fig. 10-74: Firmware Function "MX_fREAL_AccTo_DINT"

Input variable	Data type	Description
Acceleration	REAL	Acceleration value in the REAL format
Axis	AXIS_REF	Axis address: The corresponding axis, e.g. "Axis1" from the global axis variables "AxisAddressing" of "MX_Base.library (IndraDrive)" or "MY_Base.library (HydraulicDrive)" can be transmitted here.

Tab. 10-179: Input Variables of the Firmware Function "MX_fREAL_AccTo_DINT"

Return value	Data type	Description
MX_fREAL_AccTo_DINT	DINT	Returns an acceleration value in the fixed-point DINT format according to the scaling which has been set

Tab. 10-180: Return Value of the Firmware Function "MX_fREAL_AccTo_DINT"

10.5.5 MX_fREAL_DistTo_DINT

The firmware function "MX_fREAL_DistTo_DINT" converts a position value in the REAL format into a fixed-point DINT value according to the scaling which has been set.

If the axis reference is invalid, 0 is returned.

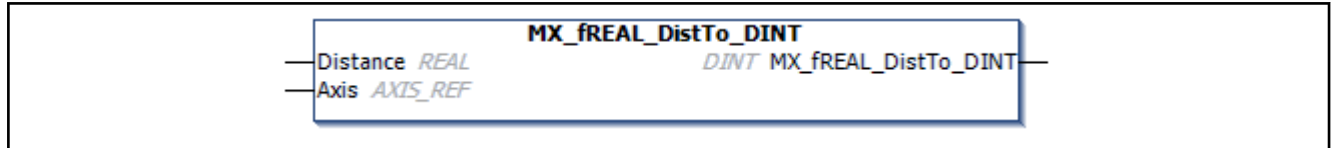


Fig. 10-75: Firmware Function "MX_fREAL_DistTo_DINT"

Input variable	Data type	Description
Distance	REAL	Position value in the REAL format
Axis	AXIS_REF	Axis address: The corresponding axis, e.g. "Axis1" from the global axis variables "AxisAddressing" of "MX_Base.lib", can be transmitted here

Tab. 10-181: Input Variables of the Firmware Function "MX_fREAL_DistTo_DINT"

Return value	Data type	Description
MX_fDINT_DistTo_REAL	DINT	Returns a position value in the fixed-point DINT format according to the scaling which has been set

Tab. 10-182: Return Value of the Firmware Function "MX_fREAL_DistTo_DINT"

10.5.6 MX_fREAL_TorqueRampTo_DINT

The firmware function "MX_fREAL_TorqueRampTo_DINT" converts a torque/force ramp value in the REAL format into a fixed-point DINT value according to the scaling which has been set.

The function can be used for the real axes "Axis1", "Axis2", etc.

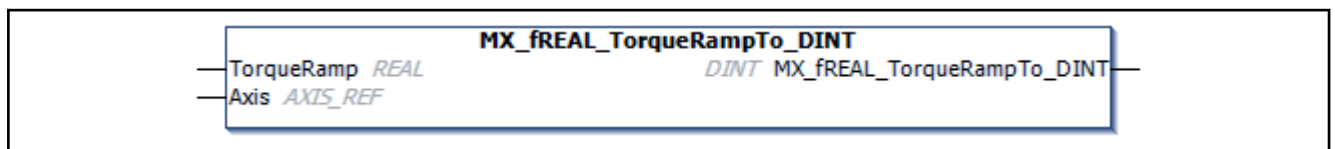


Fig. 10-76: Firmware Function "MX_fREAL_TorqueRampTo_DINT"

Input variable	Data type	Description
TorqueRamp	REAL	Torque/force ramp value in the REAL format
Axis	AXIS_REF	Axis address: The corresponding axis, e.g. "Axis1" from the global axis variables "AxisAddressing" of "MX_Base.library (IndraDrive)" or "MY_Base.library (HydraulicDrive)" can be transmitted here.

Tab. 10-183: Input Variables of the Firmware Function "MX_fREAL_TorqueRampTo_DINT"

Return value	Data type	Description
MX_fREAL_TorqueRampTo_DINT	DINT	Returns a torque/force ramp value in the fixed-point DINT format according to the scaling which has been set

Tab. 10-184: Return Value of the Firmware Function "MX_fREAL_TorqueRampTo_DINT"

MX_PLCOpen

10.5.7 MX_fREAL_TorqueTo_DINT

The firmware function "MX_fREAL_TorqueTo_DINT" converts a torque/force value in the REAL format into a fixed-point DINT value according to the scaling which has been set.

The function can be used for the real axes "Axis1", "Axis2", etc.

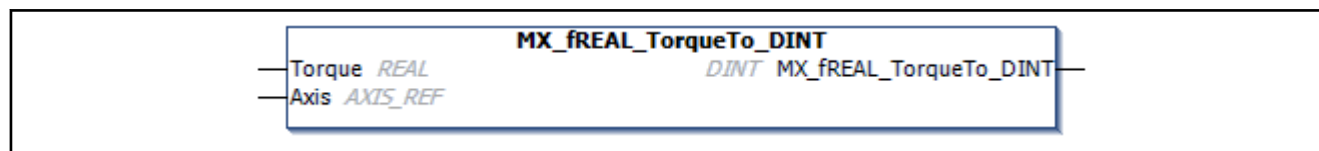


Fig. 10-77: Firmware Function "MX_fREAL_TorqueTo_DINT"

Input variable	Data type	Description
Torque	REAL	Torque/force value in the REAL format
Axis	AXIS_REF	Axis address: The corresponding axis, e.g. "Axis1" from the global axis variables "AxisAddressing" of "MX_Base.library (IndraDrive)" or "MY_Base.library (HydraulicDrive)" can be transmitted here.

Tab. 10-185: Input Variables of the Firmware Function "MX_fREAL_TorqueTo_DINT"

Return value	Data type	Description
MX_fREAL_TorqueTo_DINT	DINT	Returns a torque/force value in the fixed-point DINT format according to the scaling which has been set

Tab. 10-186: Return Value of the Firmware Function "MX_fREAL_TorqueTo_DINT"

10.5.8 MX_fREAL_VelTo_DINT

The firmware function "MX_fREAL_VelTo_DINT" converts a velocity value in the REAL format into a fixed-point DINT value according to the scaling which has been set.

If the axis reference is invalid, 0 is returned.



Fig. 10-78: Firmware Function "MX_fREAL_VelTo_DINT"

Input variable	Data type	Description
Velocity	REAL	Velocity value in the REAL format
Axis	AXIS_REF	Axis address: The corresponding axis, e.g. "Axis1" from the global axis variables "AxisAddressing" of "MX_Base.library (IndraDrive)" or "MY_Base.library (HydraulicDrive)" can be transmitted here.

Tab. 10-187: Input Variables of the Firmware Function "MX_fREAL_VelTo_DINT"

Return value	Data type	Description
MX_fREAL_VelTo_DINT	DINT	Returns a velocity value in the fixed-point DINT format according to the scaling which has been set

Tab. 10-188: Return Value of the Firmware Function "MX_fREAL_VelTo_DINT"

10.6 "Tools" - General Functions

10.6.1 MX_fGetFreeTicks

The firmware function "MX_fGetFreeTicks" is used to determine the time of the last PLC time slice still available to the PLC. This PLC time slice is that part of a millisecond which remains after the drive controller and control tasks have been processed.

Timer ticks are returned that can be converted into microseconds with the function "MX_fHighResTimerTicks_to_us".



- When a freewheeling task is active, "0" is always returned in this case because this task works for the whole time slice.
- If "MX_fGetFreeTicks" is called in a task which clocked more slowly than 1 ms, only the available time of the last ms time slice is displayed.
- When a cyclic task takes more time to be processed than the cycle time which was set, "MX_fGetFreeTicks" always returns the value "0".



Fig. 10-79: Firmware Function "MX_fGetFreeTicks"

Return value	Data type	Description
MX_fGetFreeTicks	DINT	Returns timer ticks. The timer ticks can be converted into μ s with the function "MX_fHighResTimerTicks_to_us"

Tab. 10-189: Return Value of the Firmware Function "MX_fGetFreeTicks"

10.6.2 MX_fGetHighResTime

The firmware function "MX_fGetHighResTime" returns a growing timer value with high resolution.

Timer ticks are returned that can be converted into microseconds with the function "MX_fHighResTimerTicks_to_us".



Fig. 10-80: Firmware Function "MX_fGetHighResTime"

Return value	Data type	Description
MX_fGetHighResTime	UDINT	Returns timer ticks. The timer ticks can be converted into μ s with the function "MX_fHighResTimerTicks_to_us"

Tab. 10-190: Return Value of the Firmware Function "MX_fGetHighResTime"

10.6.3 MX_fHighResTimerTicks_to_us

The firmware function "MX_fHighResTimerTicks_to_us" is used to convert timer ticks into microseconds and return the value as REAL.

MX_PLCOpen



Fig. 10-81: Firmware Function "MX_fHighResTimerTicks_to_us"

Input variable	Data type	Description
HighResTimerTicks	UDINT	Transfer values are timer ticks (see, for example, "MX_fGetHighResTime")

Tab. 10-191: Input Variable of the Firmware Function "MX_fHighResTimerTicks_to_us"

Return value	Data type	Description
MX_fHighResTimerTicks_to_us	REAL	Returns the timer ticks in µs

Tab. 10-192: Return Value of the Firmware Function "MX_fHighResTimerTicks_to_us"

10.6.4 MX_IECTaskGetLoad

The function block "MX_IECTaskGetLoad" is used to activate the extended runtime measurement and displays information on the task load (see also Application Manual on IndraMotion MLD, chapter "Runtime Measurements").

- 

As soon as a function block instance has been created in the project, the extended runtime measurement is activated in the drive.
- 

The function block "MX_IECTaskGetLoad" can be called in the "own" task, but also in another (slower) task to save calculating time. The maximum task load is also measured, if the function block is not called in every cycle.

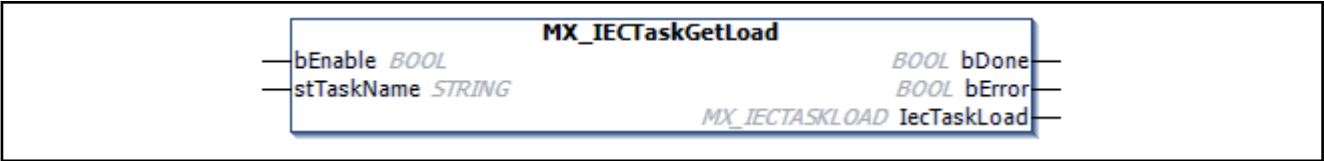


Fig. 10-82: Firmware Function Block "MX_IECTaskGetLoad"

Input variable	Data type	Description
bEnable	BOOL	Upon "bEnable"=TRUE the task information of the task named at the "stTaskName" input is returned.
stTaskName	STRING	At "stTaskName" indicate the name of the task the runtime information of which is to be displayed

Tab. 10-193: Input Variables of the Firmware Function Block "MX_IECTaskGetLoad"

Output variable	Data type	Description
bDone	BOOL	"bDone" returns TRUE, if the task exists and the outputs are valid
bError	BOOL	With "Error"=TRUE the function block signals that an error is present: • The task was not found (wrong task name) or • The extended runtime measurement was activated in the MPB firmware, but this firmware does not support extended runtime measurement
IecTaskLoad	MX_IECTASKLOAD	Runtime information of the task Elements in "IecTaskLoad": • "rLoad" displays the load of the task in the last cycle in percent, with reference to the cycle time • "rLoadMax" displays the maximum load of the task in percent, with reference to the cycle time • "rFreeTime" displays the remaining unused calculating time of the task up to its next start, from the last cycle in µs • "rMinFreeTime" displays the minimum remaining unused calculating time of the task up to its next start in µs

Tab. 10-194: *Output Variables of the Firmware Function Block "MX_IECTaskGetLoad"*

10.6.5 MX_PositionLoopEventInfo



The function block "MX_PositionLoopEventInfo" is for exclusive use by Bosch Rexroth staff.

11 Basic Libraries

11.1 MX_CommonTypes

The "MX_CommonTypes" library contains data types and structures most of which are only used internally.

The sections below only describe the data types and structures relevant to the user.

Structure "ERROR_STRUCT"

"ERROR_STRUCT" is a structure relevant to the user. The structure is used for the "ErrorIdent" output. "ERROR_STRUCT" contains information which describes the occurred error in detail.

Elements of the structure "ERROR_STRUCT":

- **Table**

"Table" is of the ENUM type. The ENUM value refers to the list required for identifying the specific error of "Additional1".

- **Additional1**

"Additional1" is of the DWORD type. "Additional1" contains the specific error number and can be used in conjunction with "Additional2" in different ways, depending on the system.

- **Additional2**

"Additional2" is of the DWORD type. "Additional2" can contain additional information. Its content differs depending on the system.

11.2 RIL_HMI_Uilities

The "RIL_HMI_Uilities" library contains function blocks supporting the HMI devices.

The library is supported by several targets and therefore is described in the documentation "Basic Libraries, IndraLogic 2G".

11.3 RIL_SocketComm

The "RIL_SocketComm" library contains functions and function blocks which allow using the TCP/UDP communication.

The library is supported by several targets and therefore is described in the documentation "Basic Libraries, IndraLogic 2G".

11.4 RIL_Uilities

The "RIL_Uilities" library contains functions and function blocks for converting and influencing different data types.

The library is supported by several targets and therefore is described in the documentation "Basic Libraries, IndraLogic 2G".

12 System Libraries

12.1 General Information

The below-mentioned system libraries are supported by Rexroth IndraMotion MLD as of MPx-18 (IndraDrive) / HDx-18 (HydraulicDrive).

NOTICE

Property damage caused when non-supported libraries are included.

Only the libraries listed below are allowed to be included.

12.2 Standard

As a standard, the "Standard.library" library is made available with the IndraLogic 2G development system.

It contains all functions and function blocks that are required by IEC61131-3 as standard blocks for an IEC programming system. The difference between a standard function and an operator is that the operator is implicitly known to the programming system while the standard blocks must be included in the project as a library.

The library is supported by several targets and therefore is described in the documentation "PLC System Libraries, IndraLogic 2G".

12.3 Util

The "Util.library" library contains function blocks which can be used for BCD conversion, bit/byte functions, mathematical auxiliary functions, as controllers, signal generators, function manipulators and for analog value processing.

The library is supported by several targets and therefore is described in the documentation "PLC System Libraries, IndraLogic 2G".

12.4 SysDir

12.4.1 General Information

With the functions of the "SysDir" library it is possible to synchronously access a file directory system on the target. Directory entries can be read.

The library can be added via the library manager (category "System"- "SysLibs").



Restriction: The "SysDirGetCurrent" and "SysDirSetCurrent" functions are not supported.

The library is supported by several targets and therefore is described in the documentation "PLC System Libraries, IndraLogic 2G".

12.4.2 Notes On Utilization

General Information

Only the "User" directory on the external memory card can be accessed. Any access from the user program to the root directory is redirected to the "User" directory.

System Libraries

When listing the directories, only the content of the "User" directory and possibly contained subdirectories are displayed. The "User" directory itself is not displayed, as if "User" were a separate partition. Access to the root node from the user's point of view can be obtained without indicating the directory, with single "\" and with double "\\" (Windows-compatible).

Runtime Behavior

When synchronous file system access is used in a cyclic MLD task, the time behavior of this task might possibly be changed. Therefore, it is advisable to carry out the file system access from a separate, freewheeling task.

12.5 SysFile

12.5.1 General Information

The functions of the "SysFile" library support **synchronous** file access. "Synchronous file access" means that a function is called and only returns after the output has been completed.

With the functions of the libraries, files can be

- opened,
- closed,
- deleted and
- renamed.
- In addition, it is possible to write data to files or read data from files.
- With other functions it is possible
 - to determine the file size or
 - the point of time of the last access, and
 - to determine or change the current offset.

The library is supported by several targets and therefore is described in the documentation "PLC System Libraries, IndraLogic 2G".

12.5.2 Notes On Utilization

General Information

Only the "User" directory on the external memory card can be accessed. Any access from the user program to the root directory is redirected to the "User" directory.

When listing the directories, only the content of the "User" directory and possibly contained subdirectories are displayed. The "User" directory itself is not displayed, as if "User" were a separate partition. Access to the root node from the user's point of view can be obtained without indicating the directory, with single "\" and with double "\\" (Windows-compatible).

Runtime Behavior

When synchronous file system access is used in a cyclic MLD task, the time behavior of this task might possibly be changed. Therefore, it is advisable to carry out the file system access from a separate, freewheeling task.

12.6 SyslecTasks23

From the "SyslecTasks23" library, only the "SysIECTaskGetConfig" function may be used; all other functions are not supported by IndraMotion MLD 2G and must not be used.

By means of the "SysIECTaskGetConfig" function, the task configuration can be read.

The tasks are listed according to their priorities, the cycle time is displayed in milliseconds.

12.7 SysMem

The "SysMem" library is used for memory management.

The library functions can be used to reserve (allocate) memory space and de-allocate it again, to define memory, to compare two memory ranges, to copy or displace data from one memory to the other, or to swap.

Processing takes place synchronously.

The library is supported by several targets and therefore is described in the documentation "PLC System Libraries, IndraLogic 2G".

12.8 SysCallback23

The "SysCallback23" library contains functions for activating defined callback functions for runtime events.

The library is supported by several targets and therefore is described in the documentation "PLC System Libraries, IndraLogic 2G".

12.9 SysSem

The "SysSem" library provides functions for generating and using semaphores for task synchronization. Semaphores serve as a means to avoid simultaneous access to critical data used by several tasks.

Processing takes place synchronously.



In the case of a PLC reset, there aren't any semaphores unlocked. The user must make sure to unlock the semaphores again.

The library is supported by several targets and therefore is described in the documentation "PLC System Libraries, IndraLogic 2G".

12.10 SysSockets

The "SysSockets" library provides functions supporting the access to sockets for communication via TCP/IP and UDP.

To begin with, please check whether the "RIL_SocketComm" library can fulfill your requirements.

If supported by the target, the corresponding functions of the operating system can be called via the library functions. For detailed information on the operating system functions, please see the relevant Help.

Please note:

- Depending on the target, the behavior of the functions can differ.
- Opening/closing sockets can take a lot of time when many sockets are opened/closed simultaneously.

The library is supported by several targets and therefore is described in the documentation "PLC System Libraries, IndraLogic 2G".

12.11 SysStr23

12.11.1 Overview

Comparing Strings

The "SysStr23" library provides functions for handling strings.

- [SysStrCmp, page 142](#)

System Libraries

- [SysStrCmpl, page 142](#)
- [SysStrCmpN, page 142](#)
- [SysStrCmpNI, page 143](#)

Copying Strings

[SysStrCpy, page 143](#)

Determining the String Length

[SysStrLen, page 143](#)

The functions are processed synchronously.

12.11.2 SysStrCmp

With the "SysStrCmp" function of the DINT type, is it possible to compare two strings lexicographically.

Input variable	Data type	Description
sString1	STRING	First string
sString2	STRING	Second string

The function provides one of the following results:

Return value < 0	String1 smaller than String2
Return value = 0	String1 = String2
Return value > 0	String1 greater than String2

12.11.3 SysStrCmpl

With the "SysStrCmpl" function of the DINT type, is it possible to check whether two strings are identical.

Input variable	Data type	Description
sString1	STRING	First string
sString2	STRING	Second string

The function provides one of the following results:

Return value < 0	String1 smaller than String2
Return value = 0	String1 = String2
Return value > 0	String1 greater than String2

12.11.4 SysStrCmpN

The "SysStrCmpN" function of the DINT type allows comparing the size of two strings; in doing so, a defined character position starting from the beginning of the string is examined.

Input variable	Data type	Description
sString1	STRING	First string
sString2	STRING	Second string
diChars	DINT	Number of characters for which - counting starts at the beginning of the string - the size of the strings is to be compared

The function provides one of the following results:

Return value < 0	String1 smaller than String2
Return value = 0	String1 = String2
Return value > 0	String1 greater than String2

12.11.5 SysStrCmpNI

With the "SysStrCmpNI" function of the DINT type, is it possible to compare whether the starting ranges of two strings are identical. The number of characters for which counting starts at the beginning of the string must be set with "diChars".

Input variable	Data type	Description
sString1	STRING	First string
sString2	STRING	Second string
diChars	DINT	Number of characters for which counting and checking starts at the beginning of the string to find out whether the contents of the two strings are identical.

The function provides one of the following results:

Return value < 0	String1 smaller than String2
Return value = 0	String1 = String2
Return value > 0	String1 greater than String2

12.11.6 SysStrCpy

The "SysStrCpy" function of the DWORD type allows copying one string (sString2) to another string (sString1).

Input variable	Data type	Description
sString1	STRING	String to which the data are to be copied (target)
sString2	STRING	String that is to be copied (source)

The return value obtained is a pointer to the target string (sString1).

12.11.7 SysStrLen

With the "SysStrLen" function of the DINT type, is it possible to determine the length of a string.

Input variable	Data type	Description
sString1	STRING	String the length of which is to be checked

The return value obtained is the number of characters, "terminal NULL" not being counted.

13 Free PLC Library by "OSCAT"

The open-source (free) "Basic" PLC library by "OSCAT" ("Open Source Community for Automation Technology") contains many useful functions in the areas of "automation technology" and "building automation".

"OSCAT" libraries are independent of the target and manufacturer.

Among other things, the "Basic" library contains function blocks on the following topics:

- Mathematical calculations
- String, list and buffer processing
- Logic modules
- Date and time functions
- Control technology
- Signal processing
- Network and communication functions
- Metrology and sensor technology
- Device drivers for standard peripherals

To obtain OSCAT libraries, please refer to www.oscat.de. Please observe that these are not Rexroth products. For this reason, we exclude any liability for completeness, functionality and operational safety, and we do not provide any support for these libraries.



On the local system, libraries can be managed in different "repositories" (directories, storage locations). Before a library can be included in a project, it must be installed in such a repository on the local system. In IndraLogic 2G, this is done in the "Library Repository" dialog.

Index

A

Analog value processing..... 139

B

BCD conversion..... 139

Bit/byte functions..... 139

Buffer processing..... 145

C

CheckBounds..... 29

CheckDivLReal..... 29

CheckDivReal..... 29

CheckPointer..... 29

CheckRangeSigned..... 29

CheckRangeUnsigned..... 29

CheckRtv..... 29

Communication via TCP/IP..... 141

Communication via UDP..... 141

Comparing strings..... 141

Copying strings..... 142

D

Date functions..... 145

Determining the string length..... 142

Directory access..... 139

E

Error..... 17

Error handling..... 17

ERROR_CODE..... 17

ERROR_STRUCT..... 17

ErrorID..... 17

ErrorIdent..... 17

F

File access

 Asynchronous..... 140

 Synchronous..... 140

I

INDRV_TABLE..... 22

Input signals "Execute" / "Enable"..... 11

L

Library

 Debug..... 31

 SysStr23..... 141

List processing..... 145

Logic modules..... 145

M

Mathematical calculations..... 145

MB_ChangeCamData..... 37

MB_ChangeProfileSet..... 37

MB_ChangeProfileStep..... 37

MB_ClearAllError..... 38

MB_Command..... 79

MB_GearInPos..... 39

MB_Home..... 39

MB_MotionProfile..... 40

MB_Phasing..... 40

MB_PhasingSlave..... 40

MB_PreSetMode..... 40

MB_ReadListParameter..... 85

MB_ReadMaxRealValue..... 87

MB_ReadMaxValue..... 89

MB_ReadMinRealValue..... 90

MB_ReadMinValue..... 92

MB_ReadName..... 93

MB_ReadParameter..... 95

MB_ReadRealListParameter..... 97

MB_ReadRealParameter..... 99

MB_ReadSercosAttribute..... 101

MB_ReadSercosDataStatus..... 102

MB_ReadStringParameter..... 104

MB_ReadUnit..... 106

MB_SetPositionControlMode..... 41

MB_Stop..... 43

MB_TorqueControl..... 44

MB_ValveControl..... 47

MB_WriteListParameter..... 107

MB_WriteParameter..... 110

MB_WriteRealListParameter..... 112

MB_WriteRealParameter..... 114

MB_WriteStringParameter..... 116

MC_CamIn..... 50

MC_CamOut..... 50

MC_GearIn..... 50

MC_GearOut..... 51

MC_Jog..... 51

MC_MoveAbsolute..... 53

MC_MoveAdditive..... 55

MC_MoveRelative..... 57

MC_MoveVelocity..... 59

MC_Power..... 61

MC_ReadStatus..... 63

MC_Reset..... 64

MC_Stop..... 65

MC_TorqueControl..... 67

Means of representation

 Conventions of notation..... 5

 Notations..... 5

 Notes..... 5

Memory management..... 141

MLD_TABLE..... 18

MX_Command..... 118

MX_CommonTypes..... 137

MX_fCheckCmdRequest..... 33

MX_fDINT_AccTo_REAL..... 129

Index

MX_fDINT_DistTo_REAL.....	129
MX_fDINT_VelTo_REAL.....	130
MX_fGetDriveWarning.....	33
MX_fGetFreeTicks.....	133
MX_fGetHighResTime.....	133
MX_fHighResTimerTicks_to_us.....	133
MX_fReadParamDINT.....	121
MX_fReadStringParam.....	121
MX_fREAL_AccTo_DINT.....	130
MX_fREAL_DistTo_DINT.....	131
MX_fREAL_TorqueRampTo_DINT.....	131
MX_fREAL_TorqueTo_DINT.....	132
MX_fREAL_VelTo_DINT.....	132
MX_fSetDriveError.....	34
MX_fSetDriveErrorText.....	34
MX_fSetDriveWarning.....	35
MX_fSetDriveWarningText.....	35
MX_fSetParamLimits.....	122
MX_fSetParamName.....	123
MX_fSetParamUnit.....	123
MX_fWriteParamDINT.....	124
MX_fWriteStringParam.....	125
MX_IECTaskGetLoad.....	134
MX_MoveAbsolute.....	68
MX_MoveAdditive.....	70
MX_MoveRelative.....	72
MX_PLCOpen.....	33
MX_PositionLoopEventInfo.....	135
MX_ReadParamDINT.....	126
MX_SetCmdState.....	36
MX_SetControl.....	74
MX_SetDeviceMode.....	76
MX_SetOpMode.....	77
MX_WriteParamDINT.....	127

P

Processing

Synchronous.....	142
Processing commands in a PLCOpen function block.....	11

R

Runtime errors

Signaling/checking.....	29
Runtime event.....	141

S

Scaling.....	129
Semaphores.....	141
SERCOS_TABLE.....	21
Signal generators.....	139
Signal time diagrams.....	11
Sockets.....	141
String processing.....	145
Strings.....	141
Synchronous file access.....	140
SysStr23.....	141

SysStrCmp.....	142
SysStrCmpl.....	142
SysStrCmpN.....	142
SysStrCmpNI.....	143
SysStrCpy.....	143
SysStrLen.....	143
System libraries.....	139

T

Time functions.....	145
---------------------	-----

Notes

The Drive & Control Company

Rexroth
Bosch Group

Bosch Rexroth AG
Hydraulics
Zum Eisengießer 1
97816 Lohr, Germany
info@boschrexroth.de
www.boschrexroth.com