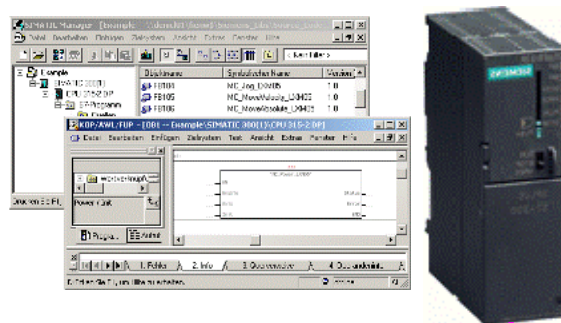


Technical documentation

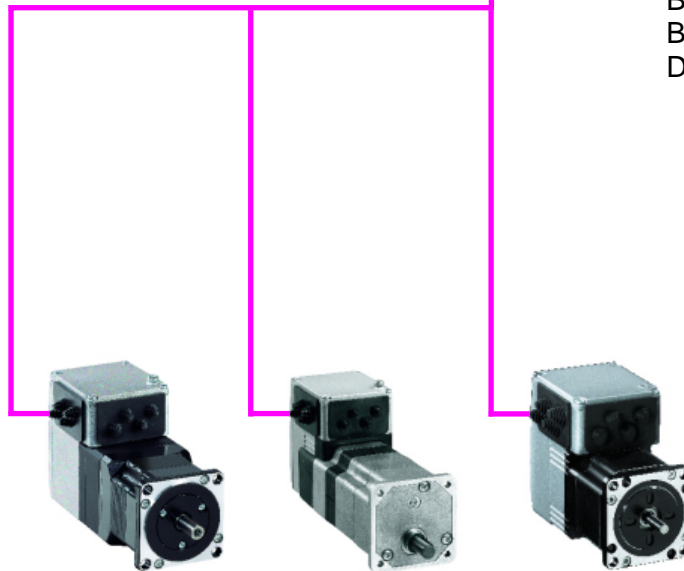
Manual

**Step 7 library for
IFS, IFE, IFA**

Edition: V1.02, 01.2008



Berger Lahr GmbH & Co. KG
Breslauer Str. 7
D-77933 Lahr



Contents

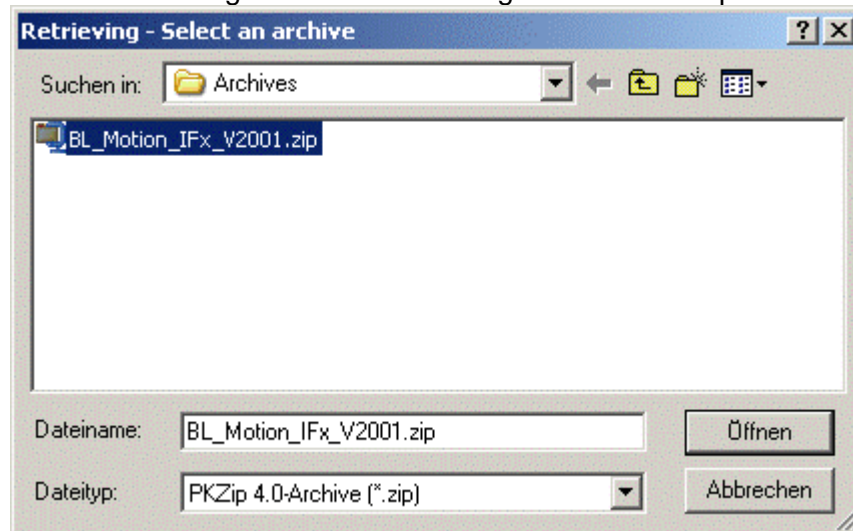
1	Extracting the library	3
2	Starting a new project	5
3	Configuring the hardware	6
4	Installing the GSD	7
5	Linking the drive into the PB network	9
6	Assigning the I/O addresses	11
6.1	Parameter data channel	11
6.2	Process data channel	12
7	Description of the library blocks	13
7.1	Copying the axis structure into the project	13
7.2	Creating an axis reference	14
8	Library blocks	16
8.1	Basic calling procedures	16
8.2	Explanation of common parameters	17
8.3	Phasing diagrams	18
8.4	Initialisation	19
8.5	Jog	22
8.6	Homing	24
8.7	Profile position mode	27
8.8	Velocity mode	30
8.9	Electronic gear	31
8.10	Stopping	34
8.11	Fast position capture	35
8.12	Read parameter	38
8.13	Write parameter	48
8.14	Inputs/outputs	52
8.15	Error handling	58
9	Glossary	60
10	List of error numbers	62
11	Parameter list for Up- and Download function	65

1 Extracting the library

In order to use the library blocks, you must first unpack the archive “BL_Motion_IFx_Vxxxx.zip” with the Step7 software.

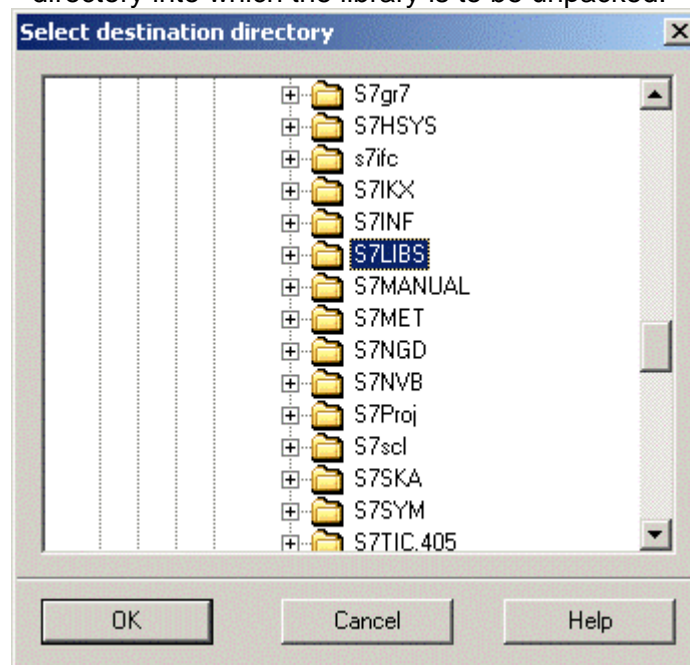
This is done with the menu item **Retrieve** in the menu **File**.

The following window for selecting the archive is opened:



Browse to the directory of the library archive, and mark the library. Confirm your selection with “Open”.

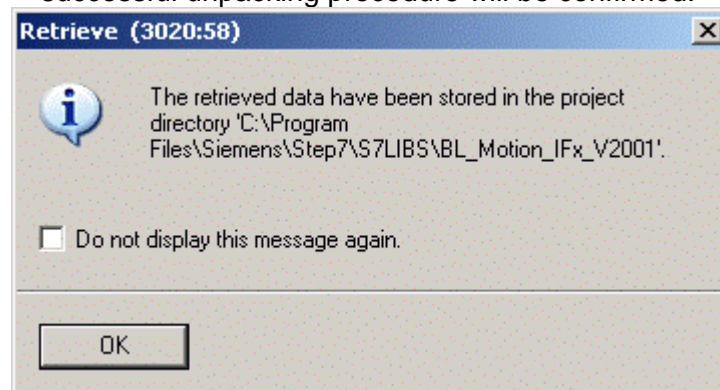
In the window shown below, you select the target directory into which the library is to be unpacked.



Mark the required directory, and confirm your selection with “OK”.

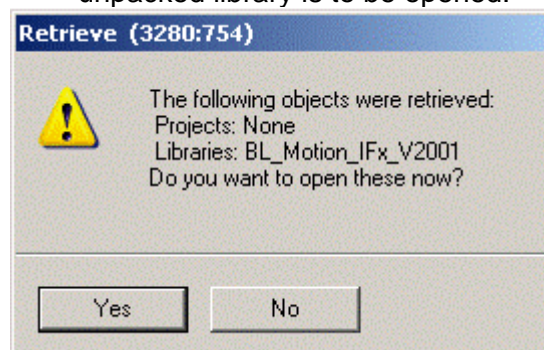
Recommendation: <Siemens directory>\Step7\S7TMP
Example: C:\Programs\Siemens\Step7\S7LIBS

Depending on the configuration of your Step7 software, the successful unpacking procedure will be confirmed.



Confirm with "OK".

In a further confirmation window, you are asked whether the unpacked library is to be opened.



Deny the request with "No".

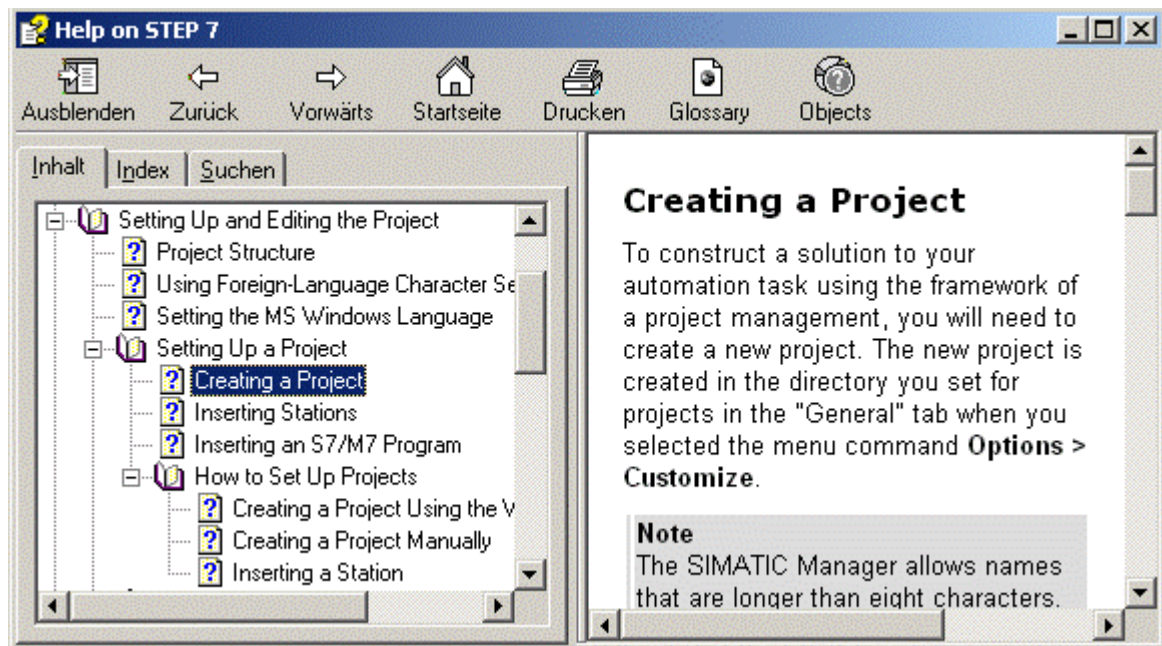
Note: Of course, you can open the library, and manually copy the relevant blocks into your application by means of the copying function of the Step 7 software.

You have now successfully unpacked the library, and can therefore access the blocks with the Step7 editors in order to use them in your application as described below.

For the library's function, it is essential that you use the associated Device Master File (GSD) **BLS70746.GSD**. The GSD supports all three drive types. But first, the corresponding GSD must be installed so that it is available in the Hardware Manager. For this purpose, you must start a new project and start the Hardware Manager.

2 Starting a new project

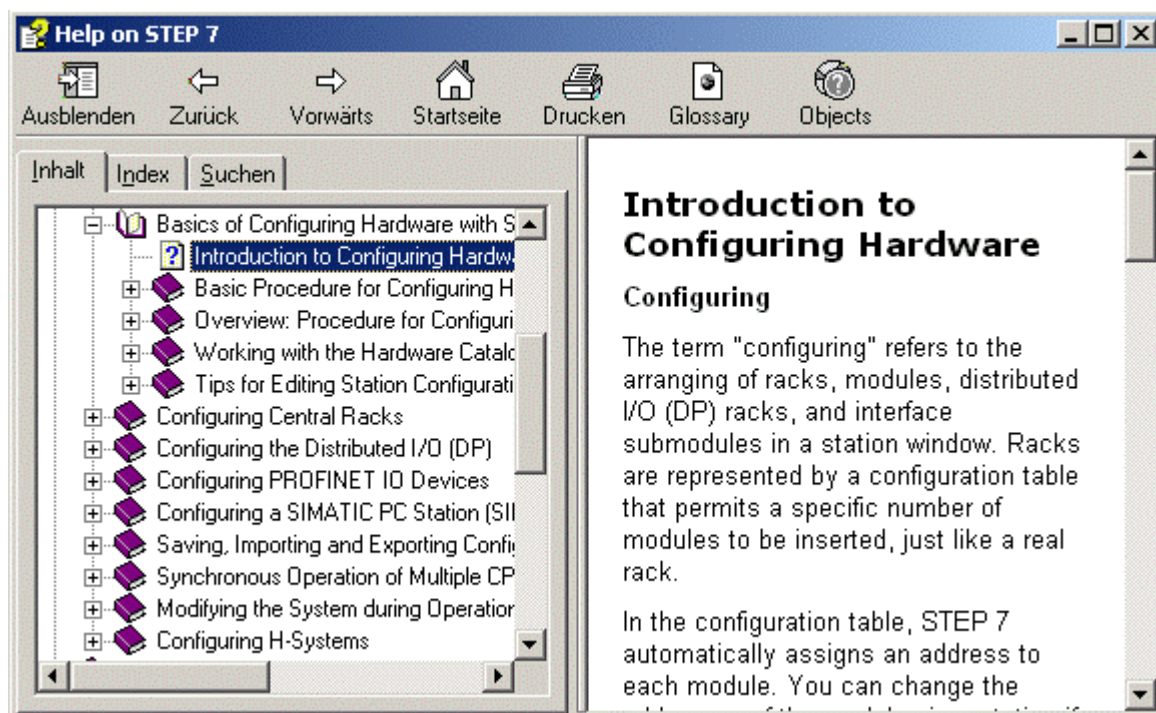
Create a new project. For this, you open the menu **File**, and select the menu item **New** or **Wizard 'New project'**. Hereby, it is assumed that you know how to create a new project, so that reference is made here to the online Help and to the documentation of Step7 and Siemens.



3 Configuring the hardware

When you have created a new project, you must define the hardware that is to be used. For this, you select the menu **Insert**, and insert a station by means of the menu item **Station**. Subsequently, you mark the inserted station, and start the hardware configurator via the menu **Edit** and the sub-menu item **Open Object**.

Hereby, it is assumed that you know how to configure the hardware, so that reference is made here to the online Help and to the documentation of Step7 and Siemens.



In order to link the drive into the Profibus network, you must first install the GSD associated to the corresponding drive, as described in the following section. If this has already been done, you can proceed with the Chapter [Linking the drive into the PB network](#).

4 Installing the GSD

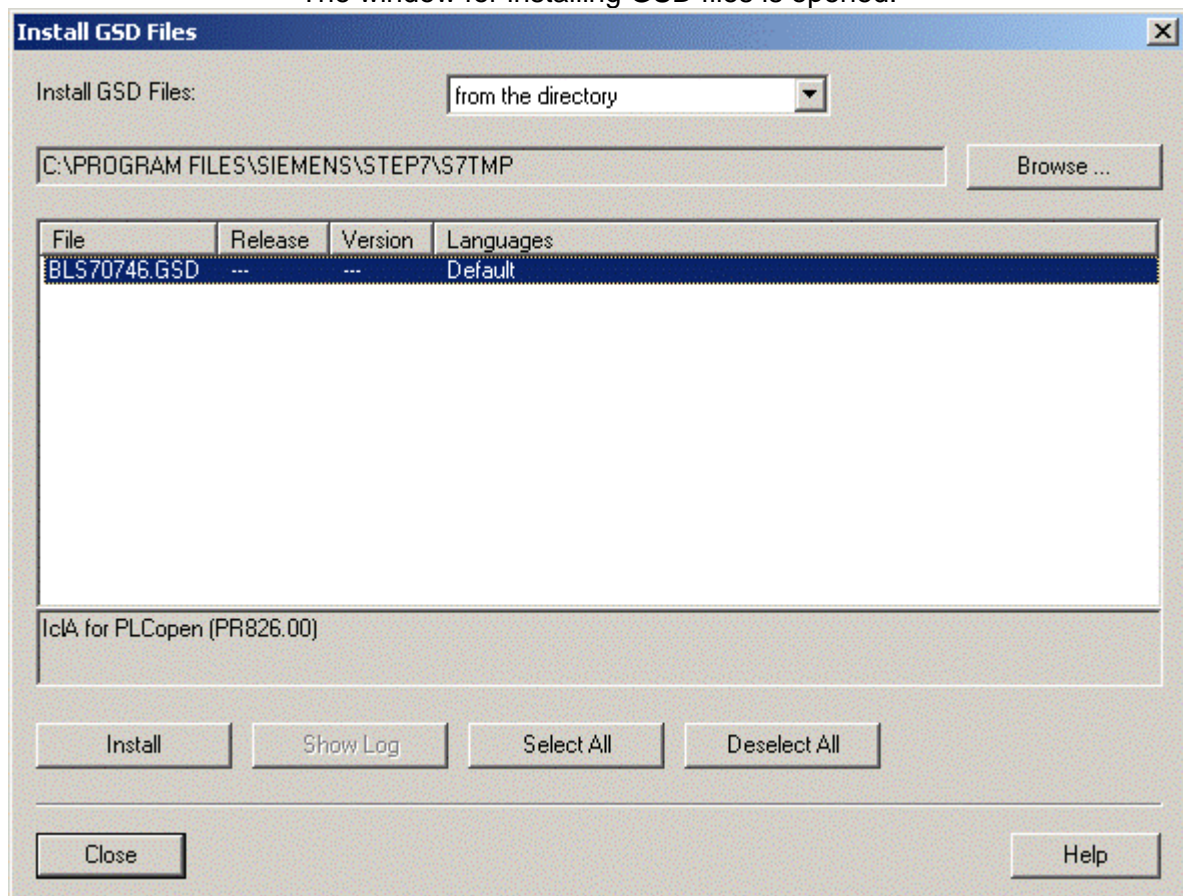
Note: The library may only be used with the associated GSD (data master file) **BLS70746.GSD**, which supports all three drive types IFS, IFE, and IFA.

The library will not work with the standard GSD.

Copy the GSD into any directory on your hard disk.
 Recommendation: <Siemens directory>\Step7\S7TMP
 Example: C:\Programs\Siemens\Step7\S7LIBS

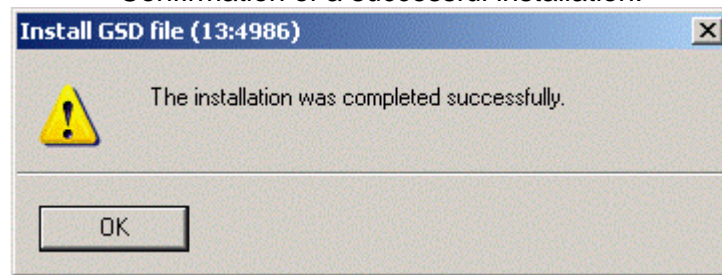
Next, you open the dialogue box in the hardware configurator for installing GSD files. This is done via the menu **Options** and the menu item **Install GSD files ...**

The window for installing GSD files is opened.



Browse to the directory with the GSD, and mark it. Confirm your selection with "Install".

Confirmation of a successful installation.

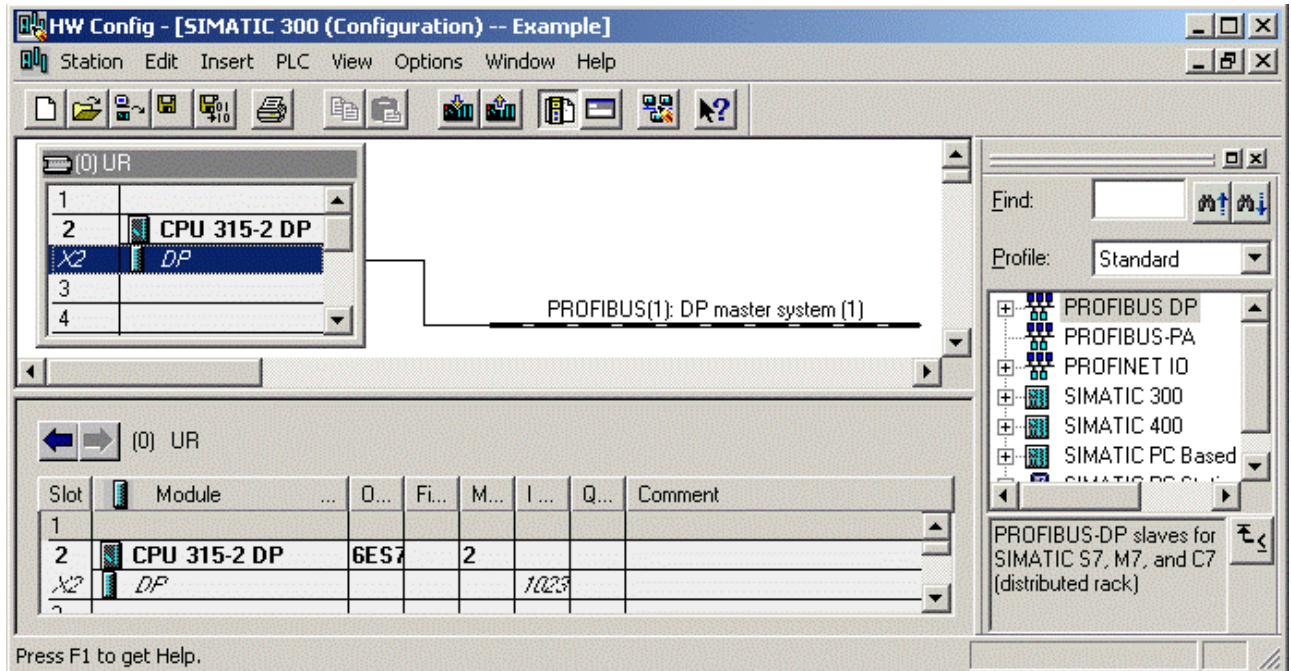


Close the confirmation message with "OK".

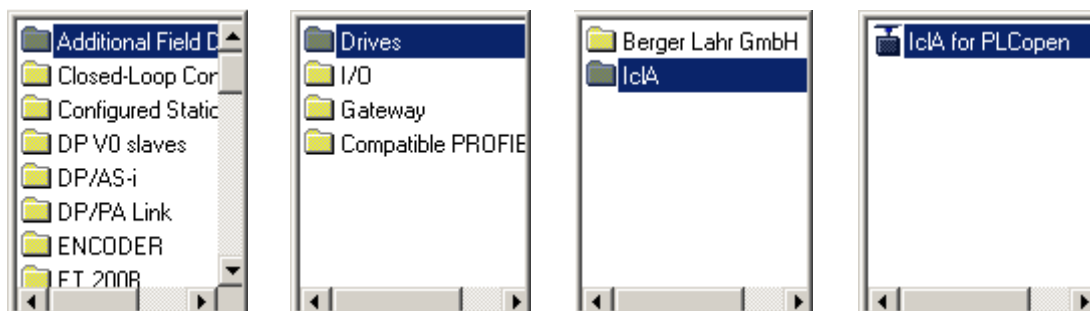
Now also close the window for installing GSDs with "Close", which returns you to the hardware configurator for the remaining hardware installation steps.

5 Linking the drive into the PB network

A prerequisite for linking the drives into the network is that you have included a module rack, a CPU, and a DP master system in your hardware configuration.

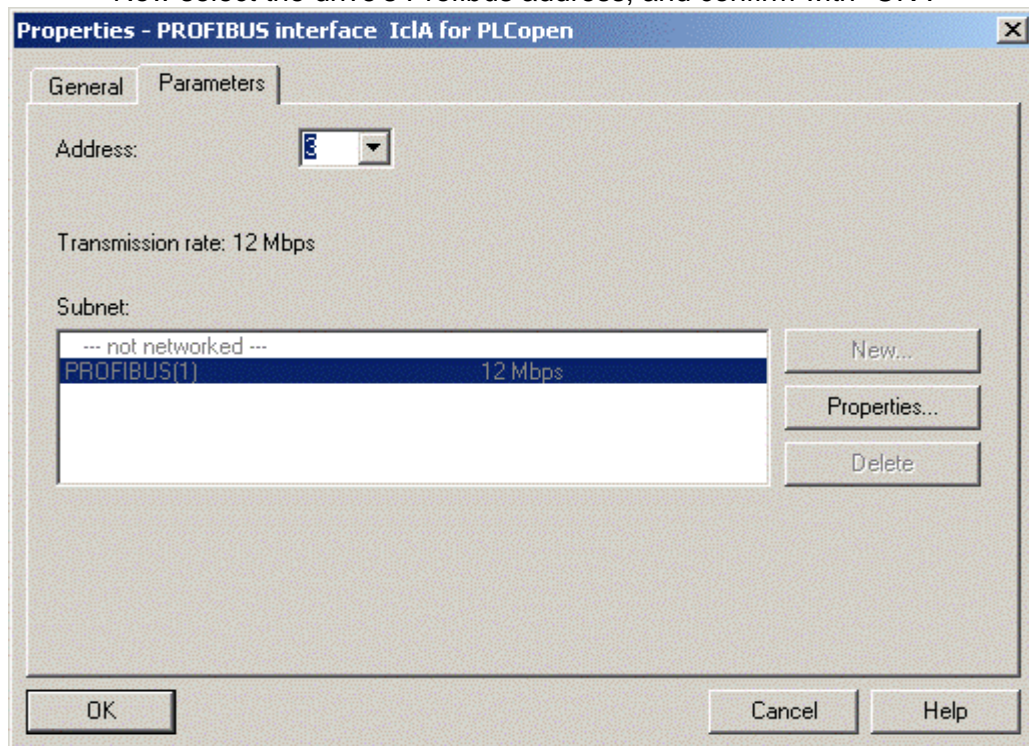


Now mark the master system in the hardware configurator, and via the menu **Insert/Insert Object...** you select the item **IclA for PLCopen** after clicking through the sub-menus **Additional Field Devices**, **Drives**, and **IclA**.

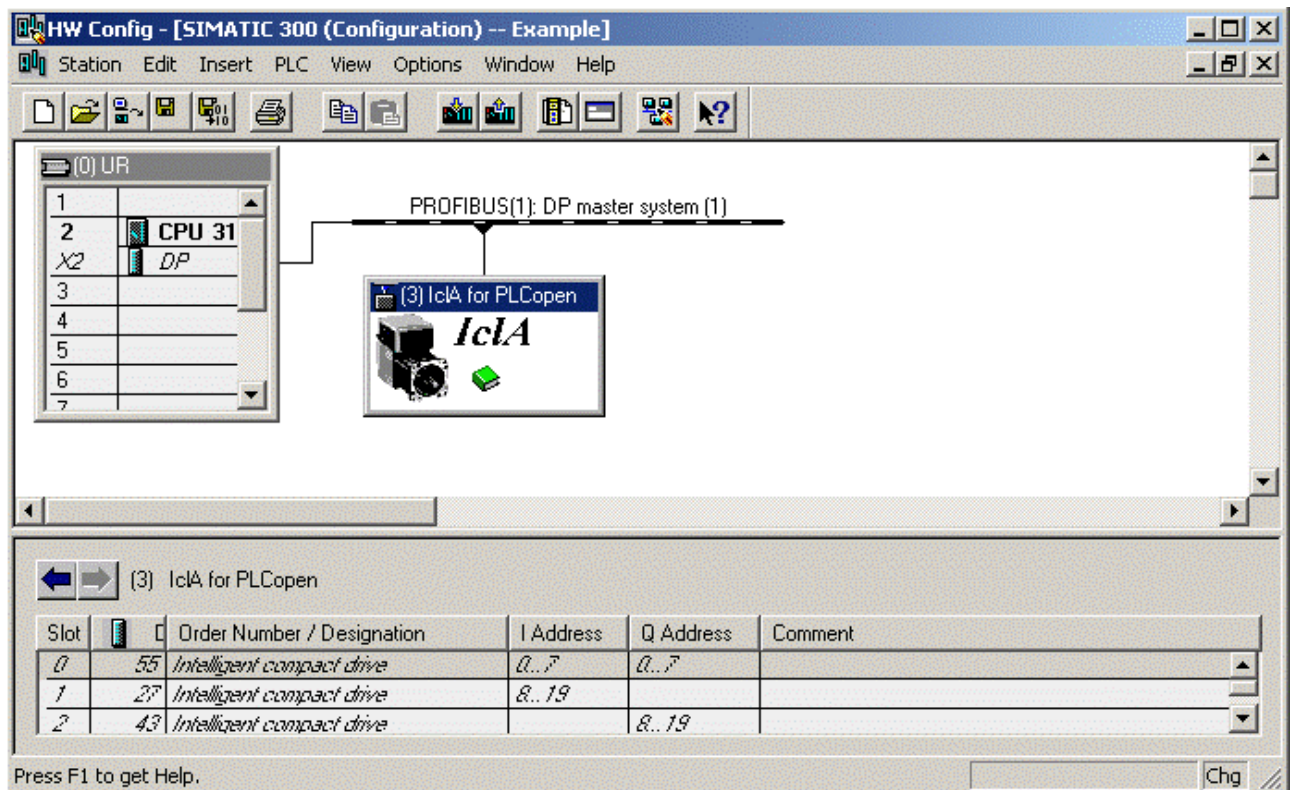


Step 7 Motion library for IFS, IFE, and IFA

Now select the drive's Profibus address, and confirm with "OK".



You have now entered the drive as a Profibus Slave, and linked it to the DP master system. The library uses two communication channels for communication with the drive: The parameter data channel (8 bytes) in Slot 1, and the process data channel (12 bytes) in Slots 2 and 3.



Finally, the I/O addresses for the communication channels must be defined, as described in the next Chapter.

6 Assigning the I/O addresses

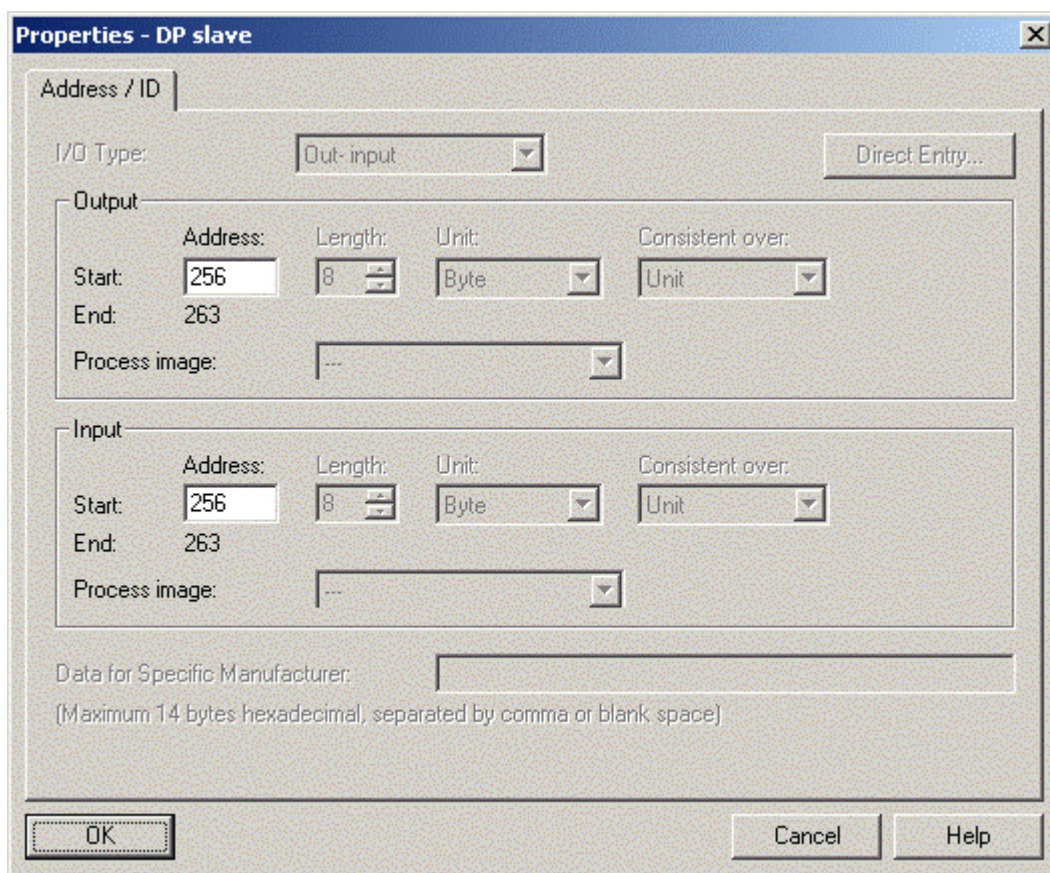
The last adjustment in the hardware configurator involves assigning the input and output addresses of the communication channels. These addresses depend on the projected CPU and on the PLC's configuration.

6.1 Parameter data channel

In order to assign the addresses for the parameter data channel, you must mark Slot 1, and select the menu item **Object Properties...** in the menu **Edit**.

Now assign a free address space for output and input data respectively, as described below.

Note: These addresses must be made known to the library.
The starting address of the output range must be transferred to the function [MC_Init_IFx](#) at the input "AdrParameterOut", and the starting address of the input range at the input "AdrParameterIn".



Properties - DP slave

Address / ID

I/O Type: Out-input Direct Entry...

Output

	Address:	Length:	Unit:	Consistent over:
Start:	256	8	Byte	Unit
End:	263			
Process image:	---			

Input

	Address:	Length:	Unit:	Consistent over:
Start:	256	8	Byte	Unit
End:	263			
Process image:	---			

Data for Specific Manufacturer:

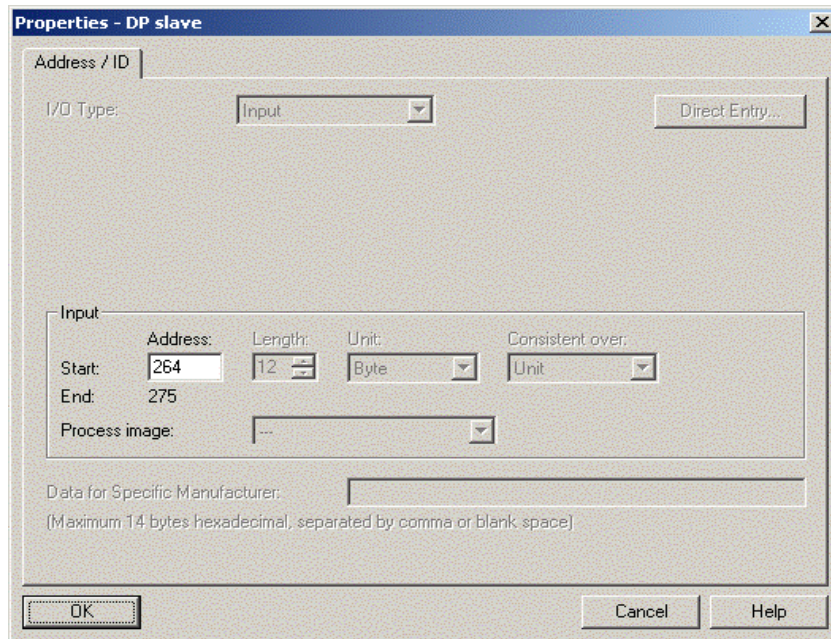
(Maximum 14 bytes hexadecimal, separated by comma or blank space)

OK Cancel Help

6.2 Process data channel

The addresses of the process data channel are assigned in the same way as the addresses of the parameter data channel. The only difference is that the input and output ranges are assigned to other slots.

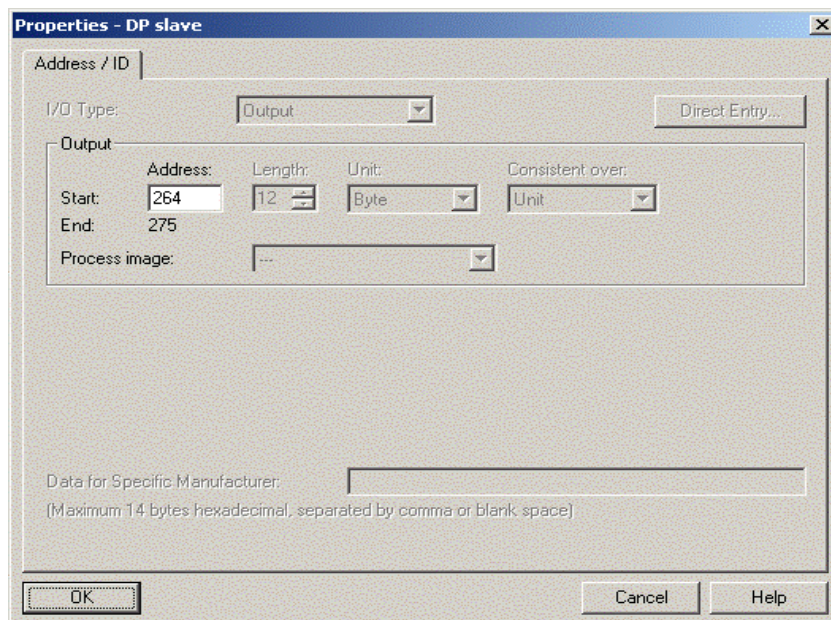
Note: These addresses must be made known to the library.
The starting address of the output range must be transferred to the function [MC_Init_IFx](#) at the input “AdrProcessdataOut”, and the starting address of the input range at the input “AdrProcessdataIn”.



The dialog box titled "Properties - DP slave" shows the configuration for an input. The "I/O Type" is set to "Input". The "Input" section contains the following fields:

Field	Value
Start	264
End	275
Length	12
Unit	Byte
Consistent over:	Unit
Process image:	---

Below the input section is a text field for "Data for Specific Manufacturer:" with a note: "(Maximum 14 bytes hexadecimal, separated by comma or blank space)". At the bottom are "OK", "Cancel", and "Help" buttons.



The dialog box titled "Properties - DP slave" shows the configuration for an output. The "I/O Type" is set to "Output". The "Output" section contains the following fields:

Field	Value
Start	264
End	275
Length	12
Unit	Byte
Consistent over:	Unit
Process image:	---

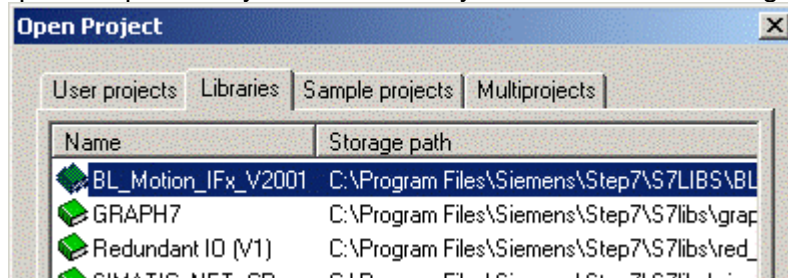
Below the output section is a text field for "Data for Specific Manufacturer:" with a note: "(Maximum 14 bytes hexadecimal, separated by comma or blank space)". At the bottom are "OK", "Cancel", and "Help" buttons.

To conclude the configuration, save and compile the settings by means of the menu item **Save and Compile** in the menu **Station** of the hardware configurator.

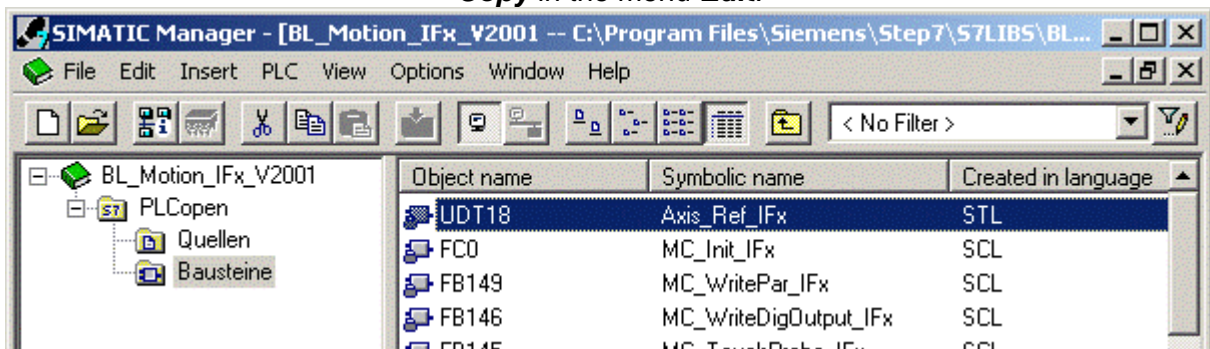
7 Description of the library blocks

7.1 Copying the axis structure into the project

Open the previously extracted library in the SIMATIC Manager.

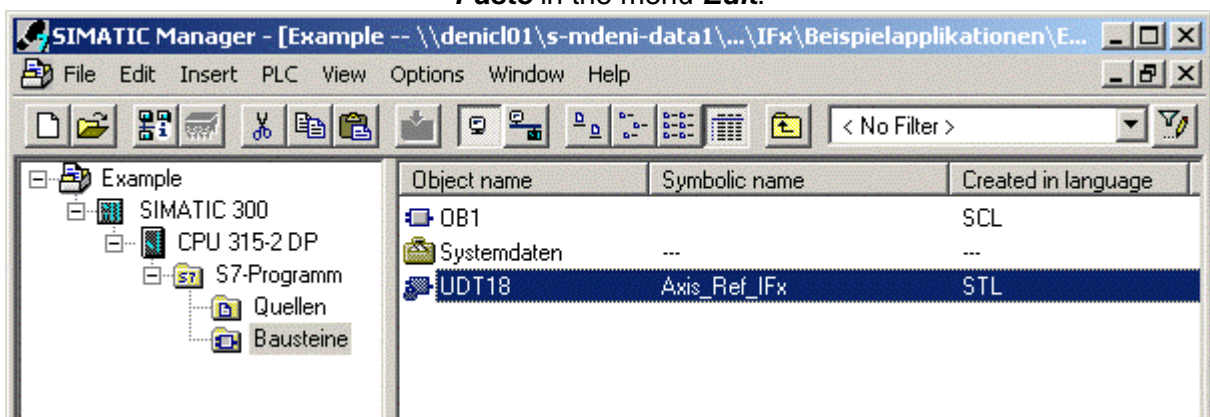


Next, mark the block UDT18, and copy it into the clipboard with the function **Copy** in the menu **Edit**.



Close the library, and mark the block folder in your project.

Now insert the block into your project from the clipboard by means of the function **Paste** in the menu **Edit**.



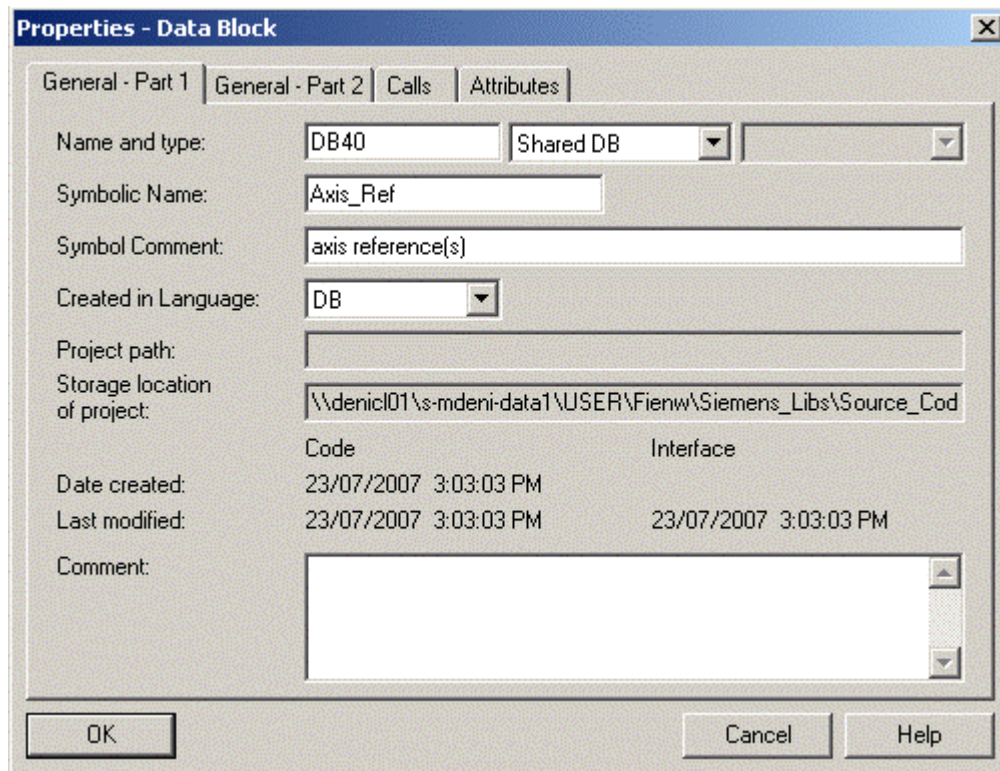
Note: Of course, you can also change the number of the UDT. This is done with the function **Rename** in the menu **Edit**.

7.2 Creating an axis reference

Create a global data block (Axis DB) in the block folder as follows:

Select **Insert / S7-Block / Data Block** in the Simatic Manager.

Edit the block's properties according to your requirements. You must keep in mind that the block is a global DB.



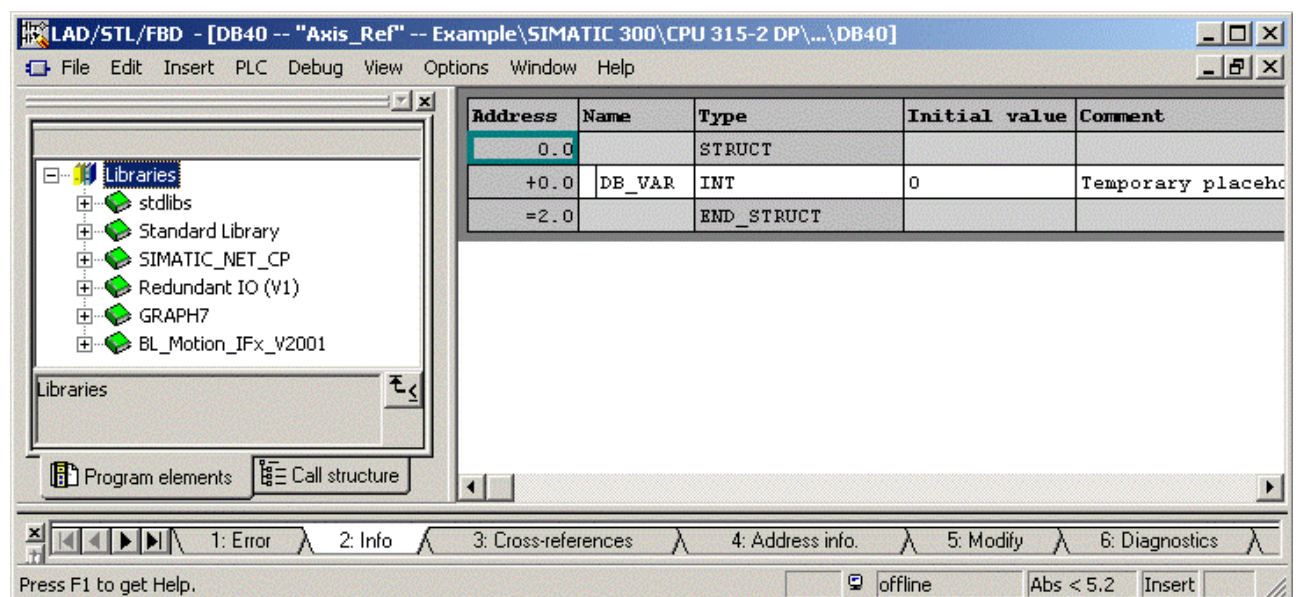
The 'Properties - Data Block' dialog box is shown with the following fields and values:

- Name and type:** DB40, Shared DB
- Symbolic Name:** Axis_Ref
- Symbol Comment:** axis reference(s)
- Created in Language:** DB
- Project path:** (empty)
- Storage location of project:** \\denic01\s-mdeni-data1\USER\Fienw\Siemens_Libs\Source_Cod
- Date created:** 23/07/2007 3:03:03 PM
- Last modified:** 23/07/2007 3:03:03 PM
- Comment:** (empty text area)

Buttons at the bottom: OK, Cancel, Help.

Confirm your entries with "OK".

Next, open the block by marking it and selecting the menu item **Open Object** in the menu **Edit**. This starts the LAD/STL/FBD editor, with which you can edit the block.



The LAD/STL/FBD editor window is shown with the title bar: LAD/STL/FBD - [DB40 -- "Axis_Ref" -- Example\SIMATIC 300\CPU 315-2 DP\...\DB40]. The menu bar includes File, Edit, Insert, PLC, Debug, View, Options, Window, Help.

The left pane shows a tree view of libraries:

- Libraries
 - stdlibs
 - Standard Library
 - SIMATIC_NET_CP
 - Redundant IO (V1)
 - GRAPH7
 - BL_Motion_IFx_V2001

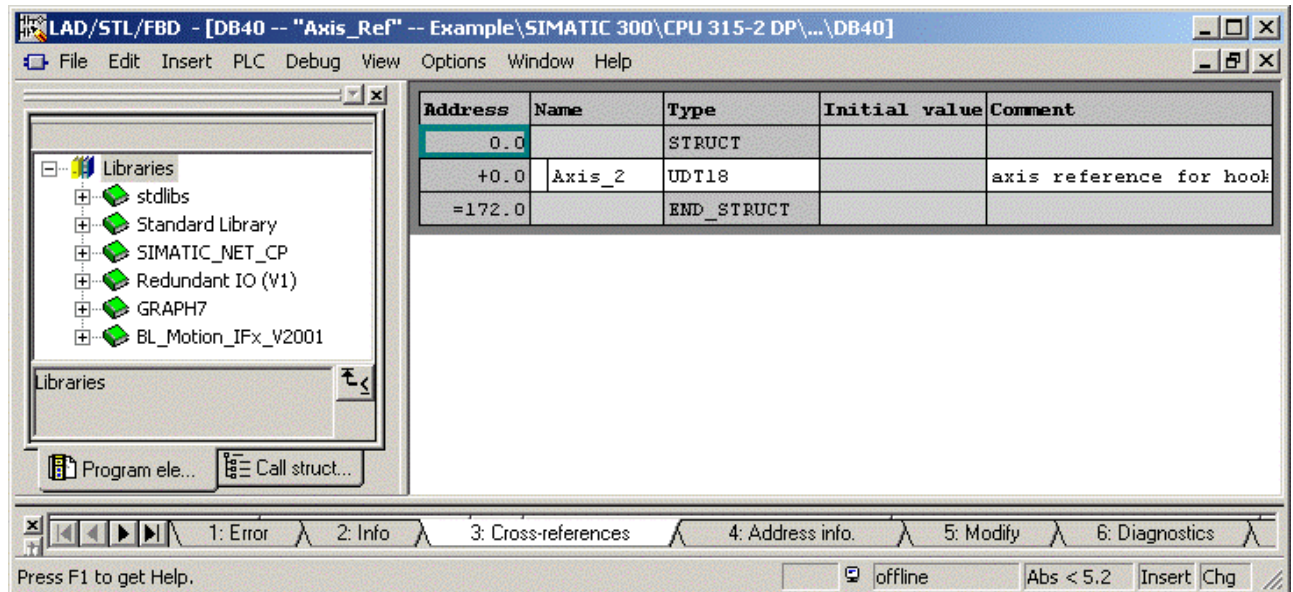
The right pane shows a table with the following data:

Address	Name	Type	Initial value	Comment
0.0		STRUCT		
+0.0	DB_VAR	INT	0	Temporary placeh
=2.0		END_STRUCT		

The bottom status bar shows: Press F1 to get Help. offline Abs < 5.2 Insert

Step 7 Motion library for IFS, IFE, and IFA

You can now create the axis reference by defining a variable of the type UDT18. In case you have renamed the UDT, you must also rename the type.

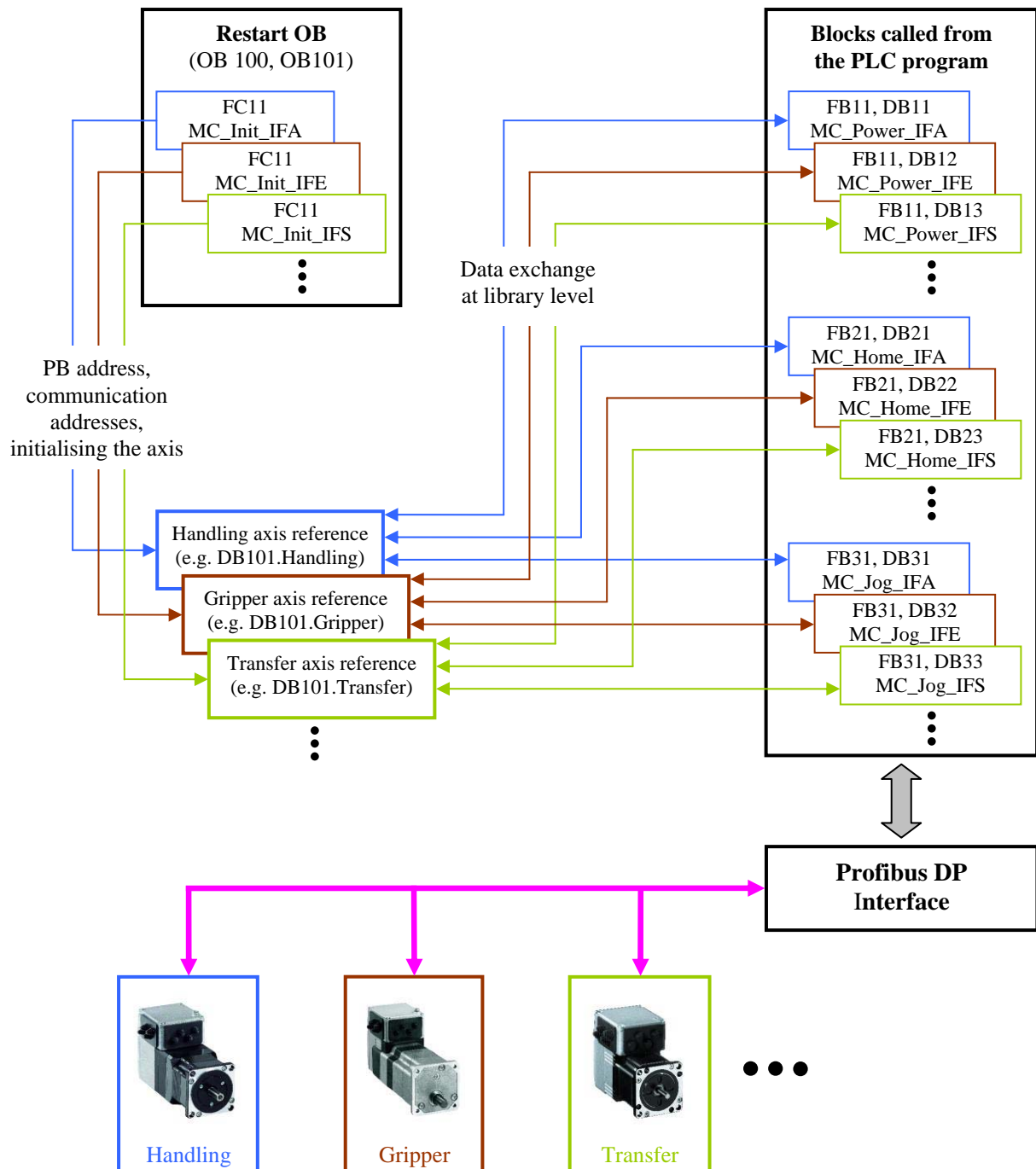


Note: If you are using several drives, you can create all the axis references in one block. This method of defining the axis reference only represents one of several possibilities. Of course, other concepts can be applied, whereby it must only be ensured that all the blocks of an axis use the same structure.

You have now established the basic conditions to start with the actual programming of your application.

8 Library blocks

8.1 Basic calling procedures

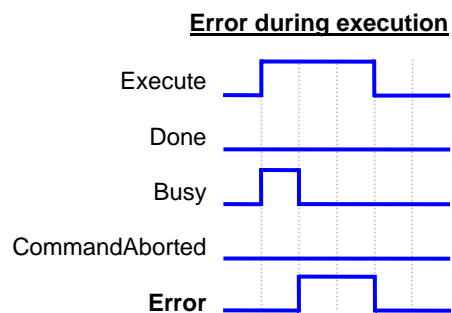
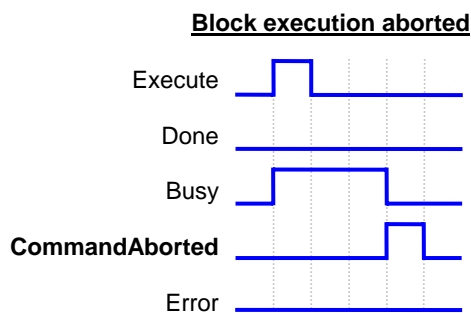
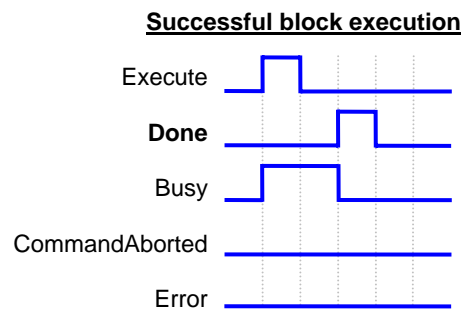
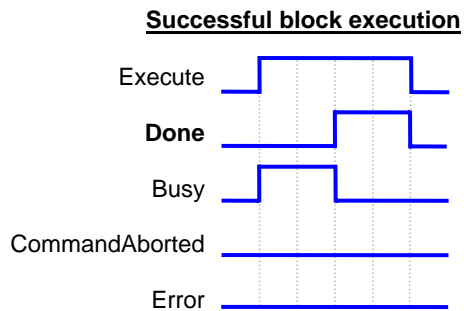


8.2 Explanation of common parameters

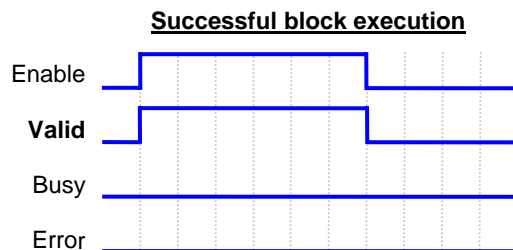
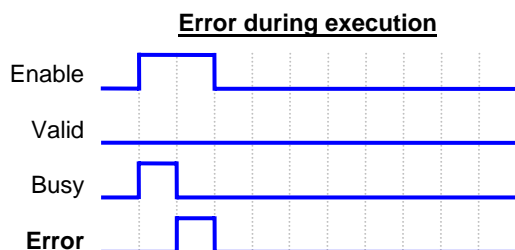
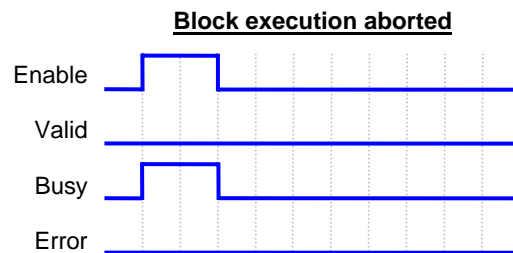
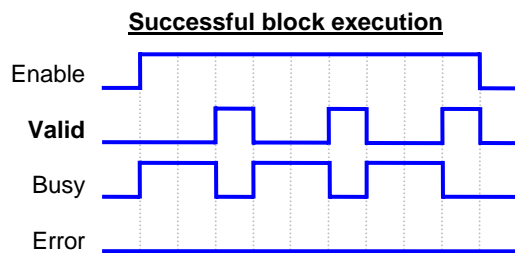
Par. type	Parameter	Data type	Description
IN	Enable	BOOL	Starts (=TRUE) and stops (=FALSE) the block's execution. The block continues to be executed as long as TRUE is returned (level-sensitive).
	Execute	BOOL	<p>The block is executed once by a rising edge. With all motion blocks (except MC_Home) the input parameters are read after a rising edge during execution, and the movement is then continued with the new parameters.</p> <p>When the block has been executed (Busy = FALSE), the output parameters are maintained until FALSE is returned. The falling edge deletes the output parameters. If the input is already FALSE when execution is completed, the output parameters are issued during precisely one block call, and then deleted (edge-sensitive).</p>
OUT	Valid	BOOL	TRUE: The value to be read is available.
	Done	BOOL	TRUE: Block execution was completed successfully.
	Busy	BOOL	TRUE: Block is being executed.
	Error	BOOL	TRUE: An error has occurred during execution.
	CommandAborted	BOOL	TRUE: Block execution has been aborted.
INOUT	Axis	STRUCT	This parameter will be transferred to the AxisDB. Example: Axis := DBname.Axisname
	Init	BOOL	<p>An initialising bit in the axis DB, which is not used by any other FB, is transferred to this parameter ("Init.I0" .. "Init.I63", see bit field for the initialising function). With Init (=TRUE), the block executes its initialisation and subsequently resets the bit.</p> <p>Example: Init := DBname.Axisname.Init.Ix with Ix = I0 .. I63.</p>

8.3 Phasing diagrams

Phasing diagrams with Execute



Phasing diagrams with Enable



8.4 Initialisation

After every restart (hot or cold) of the PLC, it is necessary for the library blocks to be re-initialised in order to set the block's local data into a defined original state (initial value).

For this purpose, the bit field "Init.Ix" is defined in the axis DB, and every block has the in/out parameter "Init". The blocks execute their initialisation routine once, if their parameter "Init" is set, and subsequently reset the transferred initialisation bit automatically. Hereby, the function MC_Init_IFx also handles initialisation of the axis DB, as well as setting the initialisation bits in the axis DB for initialising the library blocks.

Therefore, every library block called in the user program must be given an initialisation bit from the axis DB via the in/out parameter "Init".

Important: Every initialisation bit may only be used by one library block. For this purpose, a total of 64 initialisation bits (AxisDB.Init.I0.. AxisDB.Init.I63) is available per axis.

Initialisation ensures that no erratic functions and dangerous, unexpected motor movements are caused by old, invalid data.

Note:

By using the library specific GSD, the PZD5 and PZD6 in the process data channel are automatically mapped during the profibus initialisation for the send direction.

It is not allowed to change this mapping, otherwise the functionality of the library is not guaranteed !

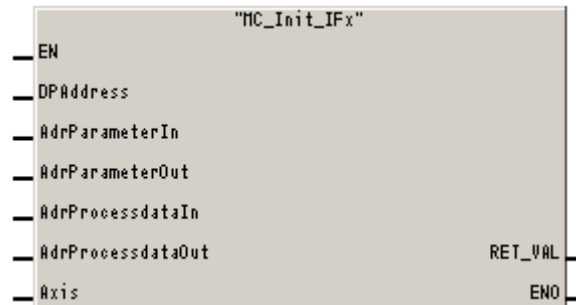
For the receive direction the PZD5 and PZD6 are mappable, following the description in the technical documentation of the device.

8.4.1 MC_Init_IFx

Task:

Initialisation of an axis.

Calling:



Call the block MC_Init_IFx once for every axis after every start of the CPU (OB100 and OB101). Depending on your program structure, you can also call the block directly in the corresponding restart OB.

Parameter description:

Par. type	Parameter	Data type	Description
IN	DPAddress	INT	Profibus address of the axis.
	AdrParameterIn	INT	Input address of the Parameter data channel .
	AdrParameterOut	INT	Output address of the Parameter data channel .
	AdrProcessdataIn	INT	Input address of the Process data channel .
	AdrProcessdataOut	INT	Output address of the Process data channel .
IN_OUT	Axis	STRUCT	Axis reference [AxisDB.AxisReference].
OUT	Ret_Val	INT	Error number (value <>0 = error).

Operating principle:

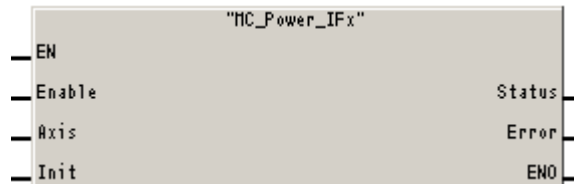
As far as possible, the transferred addresses are checked for plausibility, and entered into the axis structure of the parameter *Axis*. In addition, the initialisation bits are set, in order to prepare for initialisation of the library blocks.

8.4.2 MC_Power_IFx

Task:

Switching the motor current on/off.

Calling:



Parameter description:

Par. type	Parameter	Data type	Description
IN	Enable	BOOL	FALSE: Switches the motor current off. TRUE: Switches the motor current on.
IN_OUT	Axis	STRUCT	Axis reference [AxisDB.AxisReference].
	Init	BOOL	Initialisation bit [Init.I0 .. Init.I63].
OUT	Status	BOOL	Indicates the status of the motor current. FALSE: Motor current is 'off'. TRUE: Motor current is 'on'.
	Error	BOOL	TRUE: An error has occurred during execution.

Operating principle:

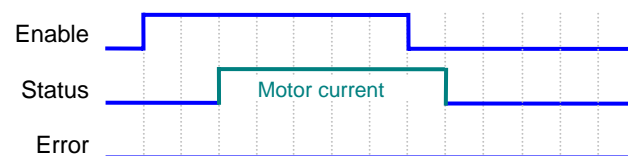
With TRUE at input *Enable*, the motor current is switched 'on'. As soon as the motor current is switched on, the output *Status* is set.

With FALSE at input *Enable*, the motor current is switched 'off'. As soon as the motor current is switched off, the output *Status* is reset.

If an error occurs during execution, the output *Error* is set.

The motor current can be switched off from any status. Any motion block that is active at this point, will be aborted.

Phase diagram:



8.5 Jog

8.5.1 MC_Jog_IFx

Task:

Jog is carried out in the "classical manual mode", i.e. with the inputs *Forward* or *Backward* active for a longer period, the motor changes to continuous operation.

Calling:



Parameter description:

Par. type	Parameter	Data type	Description
IN	Forward	BOOL	FALSE: Stops the movement. TRUE: The axis moves in the clockwise direction.
	Backward	BOOL	FALSE: Stops the movement. TRUE: The axis moves in the counter-clockwise direction.
	Fast	BOOL	Speed switchover is also possible during operation: FALSE: Speed <i>VeloSlow</i> is selected. TRUE: Speed <i>VeloFast</i> is selected.
	TipPos	INT	0: Infinite, i.e. the motor switches to continuous operation immediately. >0: Distance [Inc] travelled by the motor after start before it switches to continuous operation after the delay time (<i>WaitTime</i>) has elapsed. Value range: IFS, IFA: 0..65535, initial value: 20. IFE: 0..65535, initial value: 2.
	WaitTime	INT	Delay time [ms], which starts when the motor has travelled a defined distance (<i>TipPos</i>), and after which the motor switches to continuous operation. Value range: 1..10000, initial value: 500.

Par. type	Parameter	Data type	Description
IN	VeloSlow	INT	Speed [rpm] for movement if Fast = FALSE. Value range: IFS: 1..3000, initial value: 300. IFE: 300..5000, initial value: 300. IFA: 1..6000, initial value: 300.
	VeloFast	INT	Speed [rpm] for movement if Fast = TRUE. Value range: IFS: 1..3000, initial value: 1000. IFE: 300..5000, initial value: 1000. IFA: 1..6000, initial value: 1000.
	Acceleration	DINT	Value for the acceleration ramp gradient [(rpm/s ²)] Value range: IFS: 1..765000, initial value: 2500. IFE: 1000..10000, initial value: 2500. IFA: 1..250000, initial value: 600.
IN_OUT	Axis	STRUCT	Axis reference [AxisDB.AxisReference].
	Init	BOOL	Initialisation bit [Init.I0 .. Init.I63].
OUT	Done	BOOL	TRUE: Block execution was completed successfully.
	Busy	BOOL	TRUE: Block is being executed.
	CommandAborted	BOOL	TRUE: Block execution has been aborted.
	Error	BOOL	TRUE: An error has occurred during execution.

Operating principle:

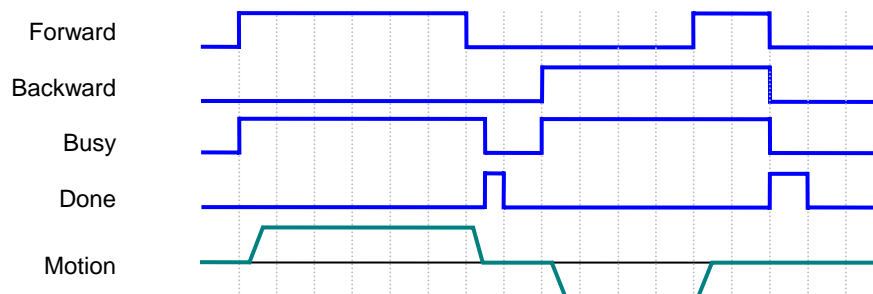
With TRUE at the *Forward* or *Backward* input, jog is started.

Depending on the parameter *Fast*, operation is either with the slow (*VeloSlow*) or with the fast (*VeloFast*) speed. The speed can also be changed during active jog. The parameters *TipPos* and *WaitTime* are used to determine the conditions for switching from the jogging mode to continuous operation.

If *Forward* and *Backward* = FALSE, the operating mode is terminated, and *Done* is set.

If *Forward* and *Backward* = TRUE, the operating mode remains active, the jogging mode is stopped, and *Busy* remains set.

Phase diagram:



8.6 Homing

In homing mode, an absolute scale reference of the motor position at a defined axis position is established. Homing can be executed by means of the two blocks described below.

8.6.1 MC_SetPosition_IFx

Task:

Absolute and relative set dimensions.

Calling:

Set dimensions can only be carried out while the drive is at standstill.



Parameter description:

Par. type	Parameter	Data type	Description
IN	Execute	BOOL	FALSE: Deletes the output parameter when block has been executed. TRUE: Rising edge starts block execution.
	Position	DINT	Dimension setting position [Inc] Value range: - 2147483648..2147483647, initial value: 0.
	Mode	BOOL	FALSE: Set current motor position as <i>Position</i> . TRUE: Add <i>Position</i> to current motor position.
IN_OUT	Axis	STRUCT	Axis reference [AxisDB.AxisReference].
	Init	BOOL	Initialisation bit [Init.I0 .. Init.I63].
OUT	Done	BOOL	TRUE: Block execution was completed successfully.
	Busy	BOOL	TRUE: Block is being executed.
	Error	BOOL	TRUE: An error has occurred during execution.

Operating principle:

Specifying the dimension reference relative to the current motor position Set dimensions allows the current motor position to be defined as the new axis reference point to which all subsequent position data relate. Set dimensions shifts the reference point for setpoint positions to the new dimension setting position.

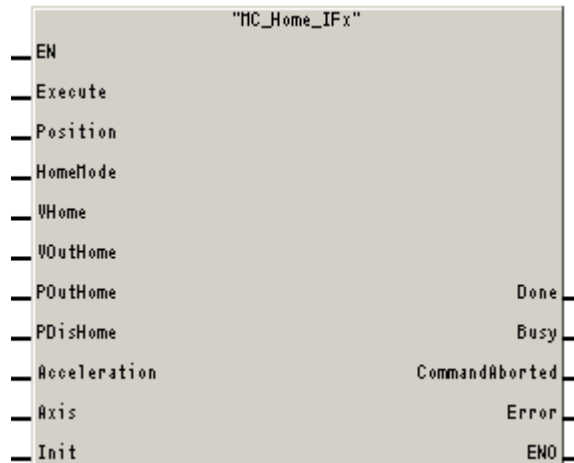
Set dimensions can be used to carry out a continuous absolute positioning without exceeding the positioning limits.

8.6.2 MC_Home_IFx

Task:

Executing the reference movement.

Calling:



Parameter description:

Par. type	Parameter	Data type	Description
IN	Execute	BOOL	FALSE: Deletes the output parameter when block has been executed. TRUE: Rising edge starts block execution.
	Position	DINT	Position is set as current motor position after successful reference movement [Inc]. Value range: - 2147483648..2147483647, initial value: 0.
	HomeMode	INT	1 = LIMP Reference movement up to positive limit switch 2 = LIMN Reference movement up to the negative limit switch 3 = REFZ Reference movement up to reference switch using counter-clockwise rotation 4 = REFZ Reference movement up to reference switch using clockwise rotation 5 = Reference movement up to index pulse using counter-clockwise rotation 6 = Reference movement up to index pulse using clockwise rotation 7 = Block movement using counter-clockwise rotation 8 = Block movement using clockwise rotation 9 = set dimensions
	VHome	INT	Speed for searching the limit or reference switch [rpm]. Drive stops when switching edge has been detected. IFS: 1..3000, initial value: 60. IFE: 300..5000, initial value: 1000. IFA: 1..6000, initial value: 60.
	VOutHome	INT	Speed for clearance movement back to the switching edge [rpm]. The max. travel distance when searching for the switching edge can be restricted with the parameter <i>POutHome</i> . IFS: 1..3000, initial value: 6. IFE: 300..5000, initial value: 500. IFA: 1..6000, initial value: 6.

Par. type	Parameter	Data type	Description
	POutHome	DINT	Run-off [Inc], i.e. max. travel distance when searching for the switching edge. If the switching edge is not found in this distance, the reference movement is interrupted with an error. Value range: 1..2147483647 [Inc], initial value: 200000 [Inc].
IN	PDisHome	DINT	Distance between switching edge and reference point [Inc]. At end of movement, the drive moves back towards switching edge until the distance has been reached. Value range: 1..2147483647, initial value: 200.
	Acceleration	DINT	Value for the acceleration ramp gradient [(rpm/s ²)] Value range: IFS: 1..765000, initial value: 2500. IFE: 1000..10000, initial value: 2500. IFA: 1..250000, initial value: 600.
IN_OUT	Axis	STRUCT	Axis reference [AxisDB.AxisReference].
	Init	BOOL	Initialisation bit [Init.I0 .. Init.I63].
OUT	Done	BOOL	TRUE: Block execution was completed successfully.
	Busy	BOOL	TRUE: Block is being executed.
	CommandAborted	BOOL	TRUE: Block execution has been aborted.
	Error	BOOL	TRUE: An error has occurred during execution.

Operating principle:

With the reference movement a defined position on the axis is approached. The defined position is specified by a mechanical switch: Limit switch, reference switch.

There are six standard reference movements:

1. Movement to positive limit switch LIMP.
2. Movement to negative limit switch LIMN.
3. Search for the reference switch REF using clockwise rotation.
4. Search for the reference switch REF using counterclockwise rotation.
5. Block (IFE) or index pulse (IFS with index pulse and IFA) using clockwise rotation.
6. Block (IFE) or index pulse (IFS with index pulse and IFA) using counterclockwise rotation.

A reference movement can be executed with or without an index pulse.

- Reference movement without index pulse
Movement from switching edge to a parametrisable distance from the switching edge.
- Reference movement with index pulse
Movement from the switching edge to the next index pulse of the encoder.

For the reference movement, search speed (V_{Home}), clearance speed ($V_{OutHome}$), safety distance ($P_{DisHome}$) and the clearance path ($P_{OutHome}$) are adjustable. A reference movement must be completed for the new reference point to be valid. If a reference movement has been aborted, it must be restarted.

The motor moves as a function of these parameters until it reaches its target or the operating mode is interrupted by the execution of another block (e.g. MC_Stop).

Any attempt to accept changed parameters by means of a rising edge at *Execute* during a homing movement is not allowed, and results in an error.

After successful completion of the reference movement, a reference position is created automatically. In this way, the reached position is defined as the absolute reference position, and is set as the value of *Position*.

8.7 Profile position mode

In the profile position operating mode, the motor is positioned from a point A to a point B by means of a positioning block.

8.7.1 MC_MoveAbsolute_IFx

Task:

Starting and monitoring the profile position operating mode with an absolute target position.

Calling:



Parameter description:

Par. type	Parameter	Data type	Description
IN	Execute	BOOL	FALSE: Deletes the output parameter when block has been executed. TRUE: Rising edge starts block execution.
	Position	DINT	Value for the absolute target position [Inc]. Value range: - 2147483648..2147483647, initial value: 0.
	Velocity	INT	Value for the set speed for movement [rpm]. IFS: 0..3000, initial value: 60. IFE: 0..5000, initial value: 1000. IFA: 0..n_maxDrv ¹ , initial value: 60.
	Acceleration	DINT	Value for the acceleration ramp gradient [(rpm/s ²)] Value range: IFS: 1..765000, initial value: 2500. IFE: 1000..10000, initial value: 2500. IFA: 1..250000, initial value: 600.
IN_OUT	Axis	STRUCT	Axis reference [AxisDB.AxisReference].
	Init	BOOL	Initialisation bit [Init.I0 .. Init.I63].
OUT	Done	BOOL	TRUE: Block execution was completed successfully.
	Busy	BOOL	TRUE: Block is being executed.
	CommandAborted	BOOL	TRUE: Block execution has been aborted.
	Error	BOOL	TRUE: An error has occurred during execution.

¹ With the IFA, the max. value of the set speed depends on the motor type, and can be determined with the help of the device documentation or with the block MC_ReadParameter_IFx via the Object (300Fh:12h).

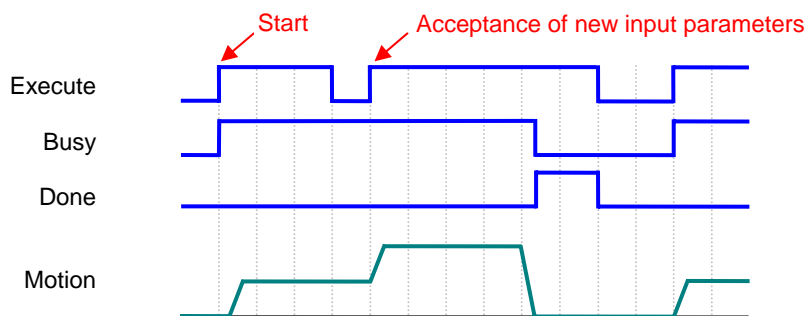
Operating principle:

Positioning on the target position *Position* at speed *Velocity*, and with absolute reference to the axis zero point. The motor moves as a function of these parameters until it reaches its target, a new reference value is entered, or the operating mode is interrupted by the execution of another block (e.g. MC_Stop).

Note:

Before an absolute positioning, the reference point must be defined by homing.

Phase diagram:

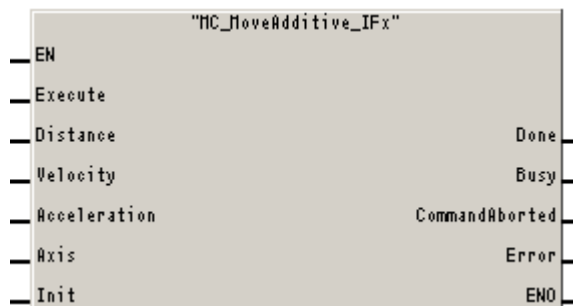


8.7.2 MC_MoveAdditive_IFx

Task:

Starting and monitoring the profile position operating mode with a target position relative to the current target position.

Calling:



Parameter description:

Par. type	Parameter	Data type	Description
IN	Execute	BOOL	FALSE: Deletes the output parameter when block has been executed. TRUE: Rising edge starts block execution.
	Distance	DINT	Value for the travel distance that is added to the current target position, and thus determines the new target position [Inc]. Value range: - 2147483648..2147483647, initial value: 0.
	Velocity	INT	Value for the set speed for movement [rpm]. IFS: 0..3000, initial value: 60. IFE: 0..5000, initial value: 1000. IFA: 0..n_maxDrv ¹ , initial value: 60.
	Acceleration	DINT	Value for the acceleration ramp gradient [(rpm/s ²)] Value range: IFS: 1..765000, initial value: 2500. IFE: 1000..10000, initial value: 2500. IFA: 1..250000, initial value: 600.
IN_OUT	Axis	STRUCT	Axis reference [AxisDB.AxisReference].
	Init	BOOL	Initialisation bit [Init.I0 .. Init.I63].
OUT	Done	BOOL	TRUE: Block execution was completed successfully.
	Busy	BOOL	TRUE: Block is being executed.
	CommandAborted	BOOL	TRUE: Block execution has been aborted.
	Error	BOOL	TRUE: An error has occurred during execution.

¹ With the IFA, the max. value of the set speed depends on the motor type, and can be determined with the help of the device documentation or with the block MC_ReadParameter_IFx via the Object (300Fh:12h).

Operating principle:

Positioning with a travel distance *Distance* referred to the current target position at speed *Velocity*. The motor moves as a function of these parameters until it reaches its target, a new reference value is entered, or the operating mode is interrupted by the execution of another block (e.g. MC_Stop).

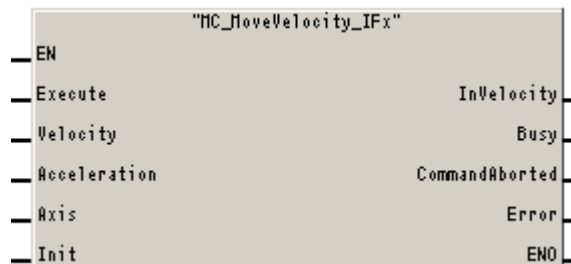
8.8 Velocity mode

8.8.1 MC_MoveVelocity_IFx

Task:

Starting and monitoring the velocity mode.

Calling:



Parameter description:

Par. type	Parameter	Data type	Description
IN	Execute	BOOL	FALSE: Deletes the output parameter when block has been executed. TRUE: Rising edge starts block execution.
	Velocity	INT	Value for the set speed for movement [rpm]. Value range: IFS: -3000..3000, initial value: 0. IFE: -5000..-300, 300..5000 ¹ , initial value: 0. IFA: -n_maxDrv..n_maxDrv ² , initial value: 0.
	Acceleration	DINT	Value for the acceleration ramp gradient [(rpm/s ²)] Value range: IFS: 1..765000, initial value: 2500. IFE: 1000..10000, initial value: 2500. IFA: 1..250000, initial value: 600.
IN_OUT	Axis	STRUCT	Axis reference [AxisDB.AxisReference].
	Init	BOOL	Initialisation bit [Init.I0 .. Init.I63].
OUT	InVelocity	BOOL	TRUE: Set speed reached
	Busy	BOOL	TRUE: Block is being executed.
	CommandAborted	BOOL	TRUE: Block execution has been aborted.
	Error	BOOL	TRUE: An error has occurred during execution.

¹ With the IFE, set speeds of -300 to 300 rpm are not permitted (except for 0).

² With the IFA, the max. value of the set speed depends on the motor type, and can be determined with the help of the device documentation or with the block MC_ReadParameter_IFx via the Object (300Fh:12h).

Operating principle:

In velocity mode, the motor receives a set speed via the parameter *Velocity*, and a movement without a target position is started.

The motor moves as a function of this setpoint until a new reference value is entered, or the operating mode is interrupted by the execution of another block (e.g. MC_Stop).

8.9 Electronic gear

In the electronic gear mode the positioning controller calculates a new position setpoint for the motor movement from a position preset and an adjustable gear ratio. This mode is used if one or more motors is to follow the reference signal from a NC controller or an encoder.

For the electronic gear operating mode, the reference signals must be applied to the optional slot (CN2). If reference pulses are applied, the positioning controller offsets them with the gear ratio, and positions the motor at the new setpoint position.

Position values are given in internal increments. If the values change, the positioning controller follows immediately. The electronic gear mode is not limited by the positioning area boundaries.

The gear ratio is the relationship between the motor increments and the externally applied reference pulses for motor movement. The gear ratio is determined with numerator and denominator. A negative numerator reverses the direction of rotation. The resulting positioning path is dependent upon the current motor resolution.

Notes

This function is only available in drive IFA.

Synchronisation: Before the electronic gear operating mode is started, there is no synchronisation between reference pulses and motor.

In case of a compensation movement (*SyncMode* = TRUE), motor movement is only limited by the max. current (device parameter *Settings.I_max*, see manual) and the drive's max. speed.

As soon as gear processing is enabled, the control deviation due to the accumulated pulses may not be greater than the tracking error limit (device parameter "*Settings.p_maxDif2*", see manual). Otherwise, the drive will signal a tracking error.

In case of immediate synchronisation (*SyncMode* = FALSE) the motor will follow the reference pulses, starting from the time when gear processing is enabled in the drive.

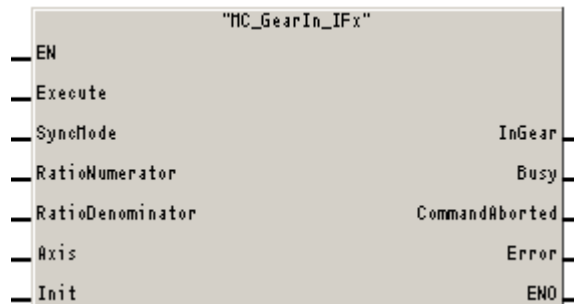
Direction enable: Before enabling gear processing, the direction of a compensation movement can be specified (device parameter "*Gear.dirEnGear*", see manual). For correct enabling of a direction, the direction inversion must be taken into account, which can be determined via the device parameter "*Motion.invertDir*" (see manual).

8.9.1 MC_GearIn_IFx

Task:

Starting and monitoring the electronic gear operating mode with a gear ratio.

Calling:



Parameter description:

Par. type	Parameter	Data type	Description
IN	Execute	BOOL	FALSE: Deletes the output parameter when block has been executed. TRUE: Rising edge starts block execution.
	SyncMode	BOOL	FALSE: Real-time synchronisation. The positioning controller follows the reference pulses from the time at which gear processing is enabled. Reference pulses that occur before the operating mode is started, are ignored. TRUE: Synchronisation with compensation movement. After gear processing has been enabled, the motor attempts to execute the accumulated reference pulses.
	RatioNumerator	INT	Gear ratio numerator. Value range: -32768 .. 32767, initial value: 1.
	RatioDenominator	INT	Gear ratio denominator. Value range: 1 .. 32767, initial value: 1.
IN_OUT	Axis	STRUCT	Axis reference [AxisDB.AxisReference].
	Init	BOOL	Initialisation bit [Init.I0 .. Init.I63].
OUT	InGear	BOOL	FALSE: The electronic gear is disabled. TRUE: The electronic gear is enabled.
	Busy	BOOL	TRUE: Block is being executed.
	CommandAborted	BOOL	TRUE: Block execution has been aborted.
	Error	BOOL	TRUE: An error has occurred during execution.

Operating principle:

In the electronic gear operating mode, the motor is supplied continuously with calculated position presets in the form of reference pulses at the encoder input in combination with the gear ratio (parameters *Numerator* and *Denominator*).

The motor moves as a function of this setpoint until it receives a new gear ratio. Operation is terminated by execution of the block MC_GearOut_IFx or by execution of another block (e.g. MC_Stop).

8.9.2 MC_GearOut_IFx

Task:

Switch-off of the electronic gear operating mode.

Calling:



Parameter description:

Par. type	Parameter	Data type	Description
IN	Execute	BOOL	FALSE: Deletes the output parameter when block has been executed. TRUE: Rising edge starts block execution.
	Axis	STRUCT	Axis reference [AxisDB.AxisReference].
IN_OUT	Init	BOOL	Initialisation bit [Init.I0 .. Init.I63].
	Done	BOOL	TRUE: Block execution was completed successfully.
OUT	Busy	BOOL	TRUE: Block is being executed.
	CommandAborted	BOOL	TRUE: Block execution has been aborted.
	Error	BOOL	TRUE: An error has occurred during execution.

Operating principle:

With enabled electronic gear operating mode, the motor is directly uncoupled from the gear master, and braked to a standstill by means of a torque ramp. This interrupts the active block MC_GearIn_IFx, which in turn signals CommandAborted = TRUE.

8.10 Stopping

8.10.1 MC_Stop_IFx

Task:

Stopping the drive with a torque ramp.

Calling:



Parameter description:

Par. type	Parameter	Data type	Description
IN	Execute	BOOL	FALSE: Deletes the output parameter when block has been executed. TRUE: Rising edge starts block execution.
IN_OUT	Axis	STRUCT	Axis reference [AxisDB.AxisReference].
	Init	BOOL	Initialisation bit [Init.I0 .. Init.I63].
OUT	Done	BOOL	TRUE: Block execution was completed successfully.
	Busy	BOOL	TRUE: Block is being executed.
	Error	BOOL	TRUE: An error has occurred during execution.

Operating principle:

Every operating mode can be terminated by stopping the drive. This does not generate an error. The interrupted movement block terminates its execution with *CommandAborted* = TRUE, and the drive changes to the status "Stopping". This status remains active until the drive is at a standstill and the block's input *Execute* has been reset. The status then changes to "Standstill", and movement blocks can be started again.

Note:

This function cannot be interrupted by other movement blocks. As long as *Execute* = TRUE, no other movement block can be started. Also after standstill the drive remains blocked.

The block brakes the motor with a torque ramp. The parameter LIM_I_maxHalt (see manual) determines the current for the torque ramp. After drive standstill an internal position compensation is executed, the position control is enabled, and the motor is stopped with the power amplifier active.

8.11 Fast position capture

The motor position can be captured with a precision of 10 μ s delay by means of 2 parametrisable channels. Two signal inputs CAP1 and CPA2 are available for capturing the trigger signals.

Notes

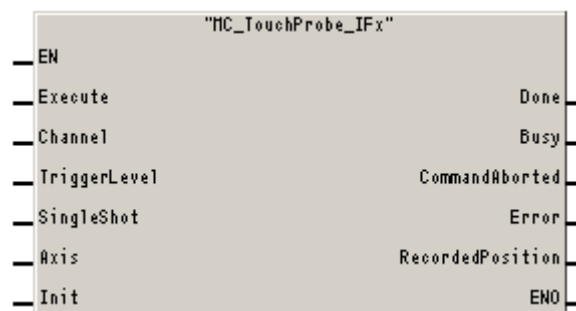
This function is only available in drives IFA and IFS.

8.11.1 MC_TouchProbe_IFx

Task:

Adjusting, starting, and monitoring the fast position capture.

Calling:



Parameter description:

Par. type	Parameter	Data type	Description
IN	Execute	BOOL	FALSE: Deletes the output parameter when block has been executed. TRUE: Rising edge starts block execution.
	Channel	UINT	Channel number: Selection of the channel to which the other parameters are referred (1 = CAP1, 2 = CAP2). Value range: 1 .. 2, initial value: 1.
	TriggerLevel	BOOL	Triggering signal edge. FALSE: falling edge TRUE: rising edge
	SingleShot	BOOL	FALSE: If the triggering event occurs repetitively, the recorded position is overwritten with the most recent position. TRUE: Position capture is disabled after the triggering event has occurred, so that the recorded position cannot be overwritten. Initial value: TRUE.
IN_OUT	Axis	STRUCT	Axis reference [AxisDB.AxisReference].
	Init	BOOL	Initialisation bit [Init.I0 .. Init.I63].

Par. type	Parameter	Data type	Description
OUT	Done	BOOL	TRUE: Block execution was completed successfully.
	Busy	BOOL	TRUE: Block is being executed.
	CommandAborted	BOOL	TRUE: Block execution has been aborted.
	Error	BOOL	TRUE: An error has occurred during execution.
	RecordedPosition	BOOL	Recorded motor position when the triggering event occurs [Inc]. Value range: -2147483648 .. 2147483647, initial value: 0.

Operating principle:

Fast position capture serves to detect the current motor position at the time when a digital 24V signal appears at one of the two capture inputs (IO2 = CAP1, IO3 = CAP2). Hereby, the motor position can be detected by means of a falling edge at the corresponding input (*Channel*).

Moreover, the parameter *SingleShot* can be used to determine whether position capture is to be executed once or continuously.

In case of a single position capture, the block is terminated with *Done* = TRUE as soon as the adjusted edge (*TriggerLevel*) occurs, and signals the detected position (*RecordedPosition*).

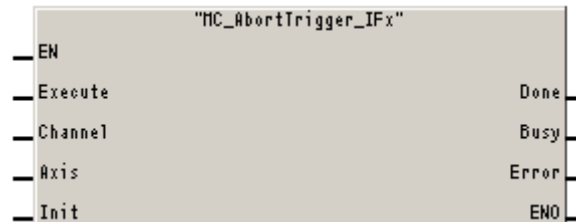
In case of continuous position capture, the block signals the most recently detected position (*RecordedPosition*) with every occurring edge, whereby the previous position is overwritten. The block does not terminate itself – it can only be interrupted with MC_AbortTrigger_IFx.

8.11.2 MC_AbortTrigger_IFx

Task:

Terminating an active position capture.

Calling:



Parameter description:

Par. type	Parameter	Data type	Description
IN	Execute	BOOL	FALSE: Deletes the output parameter when block has been executed. TRUE: Rising edge starts block execution.
	Channel	INT	1: Terminates position capture on channel 1 (CAP1). 2: Terminates position capture on channel 2 (CAP2). Value range: 1 .. 2, initial value: 1.
IN_OUT	Axis	STRUCT	Axis reference [AxisDB.AxisReference].
	Init	BOOL	Initialisation bit [Init.I0 .. Init.I63].
OUT	Done	BOOL	TRUE: Block execution was completed successfully.
	Busy	BOOL	TRUE: Block is being executed.
	Error	BOOL	TRUE: An error has occurred during execution.

Operating principle:

The active position capture is disabled for the corresponding *Channel*.
For this channel, the block MC_TouchProbe_IFx signals *CommandAborted* = TRUE.

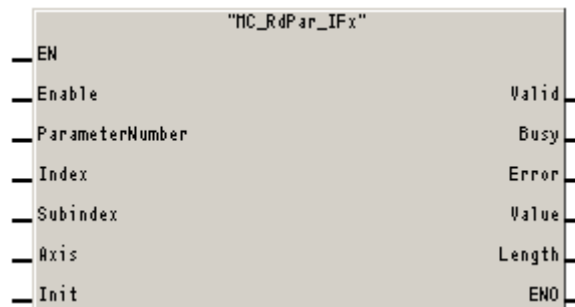
8.12 Read parameter

8.12.1 MC_ReadParameter_IFx

Task:

Reading an Object from the [Deviceparameter list](#).

Calling:



Parameter description:

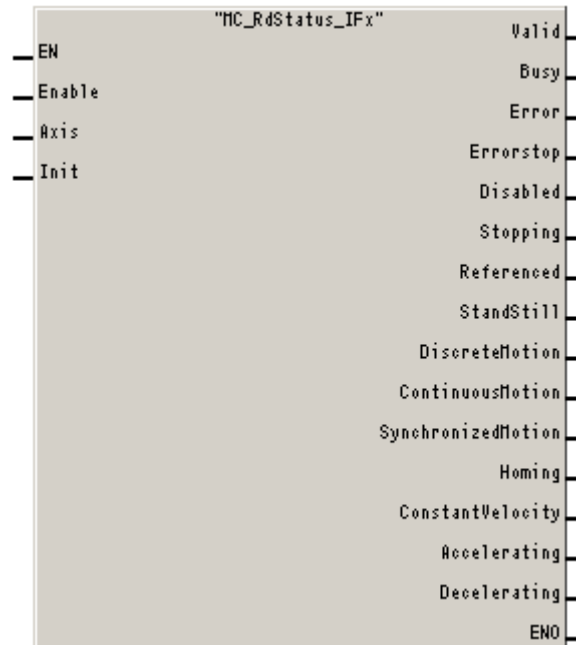
Par. type	Parameter	Data type	Description
IN	Enable	BOOL	FALSE: Terminates block execution. TRUE: Starts block execution.
	ParameterNumber	INT	0: The parameter is selected with <i>Index</i> . 1: Current setpoint position of the profile generator [Inc]. 10: Current actual speed [rpm]. 11: Current set speed [rpm]. Other numbers are not supported. Value range: 0..32767, initial value: 0.
	Index	INT	Index of the Object to be read – the Objects are listed in the manual with their index and sub-index. Only valid with <i>ParameterNumber</i> = 0. Value range: 0..32767, initial value: 0.
	sub-index	INT	Sub-index of the Object to be read – the Objects are listed in the manual with their index and sub-index. Only valid with <i>ParameterNumber</i> = 0. Value range: 0..32767, initial value: 0.
IN_OUT	Axis	STRUCT	Axis reference [AxisDB.AxisReference].
	Init	BOOL	Initialisation bit [Init.I0 .. Init.I63].
OUT	Valid	BOOL	TRUE: The value to be read is available.
	Busy	BOOL	TRUE: Block is being executed.
	Error	BOOL	TRUE: An error has occurred during execution.
	Value	DINT	Value of the parameter to be read. Value range: -2147483648..2147483647, initial value: 0.
	Length	INT	Length of the parameter to be read [bytes]. Value range: 1..4, initial value: 0.

8.12.2 MC_ReadStatus_IFx

Task:

Reading the drive's current status.

Calling:



Parameter description:

Par. type	Parameter	Data type	Description
IN	Enable	BOOL	FALSE: Terminates block execution. TRUE: Starts block execution.
	Axis	STRUCT	Axis reference [AxisDB.AxisReference].
IN_OUT	Init	BOOL	Initialisation bit [Init.I0 .. Init.I63].
	Valid	BOOL	TRUE: The read status is valid. FALSE: The status is not (yet) valid.
OUT	Busy	BOOL	TRUE: Block is being executed.
	Error	BOOL	TRUE: An error has occurred during execution.
	Errorstop	BOOL	TRUE: The axis is in a fault condition.
	Disabled	BOOL	TRUE: Motor current is 'off'.
	Stopping	BOOL	TRUE: The axis has been stopped and is still being blocked by the block MC_Stop_IFx.
	Referenced	BOOL	TRUE: The drive is ready.
	StandStill	BOOL	TRUE: The drive is stopped.
	DiscreteMotion	BOOL	TRUE: The drive is in a profile position operating mode.
	ContinuousMotion	BOOL	TRUE: The drive is in a continuous motion operating mode.

Par. type	Parameter	Data type	Description
OUT	ContinuousMotion	BOOL	TRUE: The drive is in an operating mode without a defined target position (MC_Jog_IFx, MC_MoveVelocity_IFx).
	SynchronizedMotion	BOOL	The drive is in the electronic gear operating mode (only IFA).
	Homing	BOOL	TRUE: The drive is in the homing operating mode.
	ConstantVelocity	BOOL	TRUE: The drive is running at a constant speed.
	Accelerating	BOOL	TRUE: The drive is accelerating.
	Decelerating	BOOL	TRUE: The drive is slowing down.

Operating principle:

The drive's current status information is being read and output. These are only valid in connection with the parameter *Valid*.

Note:

At any time, the drive is in only one of the states *StandStill*, *Homing*, *DiscreteMotion*, *ContinuousMotion*, *SynchronizedMotion*, *Stopping*, *Disabled* or *Errorstop*. The correspondingly named output of the block is then TRUE.

The same applies for the movement conditions *ConstantVelocity*, *Accelerating*, and *Decelerating*.

8.12.3 MC_ReadActualPosition_IFx

Task:

Reading the motor's actual position in increments.

Calling:



Parameter description:

Par. type	Parameter	Data type	Description
IN	Enable	BOOL	FALSE: Terminates block execution. TRUE: Starts block execution.
	Axis	STRUCT	Axis reference [AxisDB.AxisReference].
IN_OUT	Init	BOOL	Initialisation bit [Init.I0 .. Init.I63].
	Valid	BOOL	TRUE: The value to be read is available.
OUT	Busy	BOOL	TRUE: Block is being executed.
	Error	BOOL	TRUE: An error has occurred during execution.
OUT	Position	DINT	Motor's current actual position [Inc]. Value range: -2147483648..2147483647, initial value: 0.

8.12.4 MC_ReadActualVelocity_IFx

Task:

Reading the motor's current speed in rpm.

Calling:



Parameter description:

Par. type	Parameter	Data type	Description
IN	Enable	BOOL	FALSE: Terminates block execution. TRUE: Starts block execution.
IN_OUT	Axis	STRUCT	Axis reference [AxisDB.AxisReference].
	Init	BOOL	Initialisation bit [Init.I0 .. Init.I63].
OUT	Valid	BOOL	TRUE: The value to be read is available.
	Busy	BOOL	TRUE: Block is being executed.
	Error	BOOL	TRUE: An error has occurred during execution.
	Velocity	DINT	Motor's current speed [rpm]. Value range: -2147483648..2147483647, initial value: 0.

8.12.5 MC_ReadRefPosition_IFx

Task:

Reading the movement profile generator's current position in increments.

Calling:



Parameter description:

Par. type	Parameter	Data type	Description
IN	Enable	BOOL	FALSE: Terminates block execution. TRUE: Starts block execution.
IN_OUT	Axis	STRUCT	Axis reference [AxisDB.AxisReference].
	Init	BOOL	Initialisation bit [Init.I0 .. Init.I63].
OUT	Valid	BOOL	TRUE: The value to be read is available.
	Busy	BOOL	TRUE: Block is being executed.
	Error	BOOL	TRUE: An error has occurred during execution.
	Position	DINT	Actual position of movement profile generator [Inc]. Value range: -2147483648..2147483647, initial value: 0.

8.12.6 MC_ReadRefVelocity_IFx

Task:

Reading the movement profile generator's current speed in rpm.

Calling:



Parameter description:

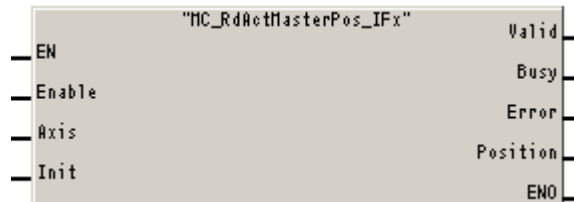
Par. type	Parameter	Data type	Description
IN	Enable	BOOL	FALSE: Terminates block execution. TRUE: Starts block execution.
	Axis	STRUCT	Axis reference [AxisDB.AxisReference].
IN_OUT	Init	BOOL	Initialisation bit [Init.I0 .. Init.I63].
	Valid	BOOL	TRUE: The value to be read is available.
OUT	Busy	BOOL	TRUE: Block is being executed.
	Error	BOOL	TRUE: An error has occurred during execution.
	Velocity	INT	Current speed of the movement profile generator [rpm]. Value range: -32768..32767, initial value: 0.

8.12.7 MC_ReadActualMasterPosition_IFx

Task:

Reading the encoder's current position in increments.

Calling:



Parameter description:

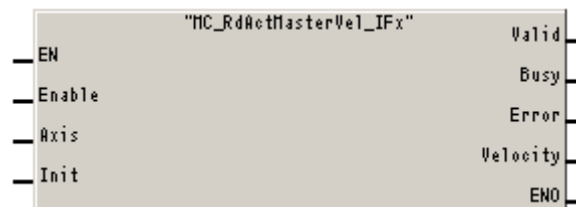
Par. type	Parameter	Data type	Description
IN	Enable	BOOL	FALSE: Terminates block execution. TRUE: Starts block execution.
IN_OUT	Axis	STRUCT	Axis reference [AxisDB.AxisReference].
	Init	BOOL	Initialisation bit [Init.I0 .. Init.I63].
OUT	Valid	BOOL	TRUE: The value to be read is available.
	Busy	BOOL	TRUE: Block is being executed.
	Error	BOOL	TRUE: An error has occurred during execution.
	Position	DINT	Current position of the external encoder [Inc]. Value range: -2147483648..2147483647, initial value: 0.

8.12.8 MC_ReadActualMasterVelocity_IFx

Task:

Reading the external encoder's current speed in increments per second.

Calling:



Parameter description:

Par. type	Parameter	Data type	Description
IN	Enable	BOOL	FALSE: Terminates block execution. TRUE: Starts block execution.
	Axis	STRUCT	Axis reference [AxisDB.AxisReference].
IN_OUT	Init	BOOL	Initialisation bit [Init.I0 .. Init.I63].
	Valid	BOOL	TRUE: The value to be read is available.
OUT	Busy	BOOL	TRUE: Block is being executed.
	Error	BOOL	TRUE: An error has occurred during execution.
	Velocity	DINT	Current speed of the external encoder [Inc/s]. Value range: -2147483648..2147483647, initial value: 0.

8.12.9 MC_UploadParameter_IFx

Task:

Reading all variable parameters and store them in the [parameter list](#).

Calling:



Parameter description:

Par-typ	Parameter	Datentyp	Bedeutung
IN	Execute	BOOL	FALSE: Deletes the output parameter when block has been executed. TRUE: Rising edge starts block execution.
	Data	ANY	Structure for the read data.
IN_OUT	Axis	STRUCT	Axis reference [AxisDB.AxisReference].
	Init	BOOL	Initialisation bit [Init.I0 .. Init.I63].
OUT	Done	BOOL	TRUE: Block execution was completed successfully.
	Busy	BOOL	TRUE: Block is being executed.
	Error	BOOL	TRUE: An error has occurred during execution.
	Errorinfo	DWORD	Additional error information, address of the error. High Word: DB-Number, Low Word: parameter address in the DB Value range: 16#0..16#FFFFFFF, initial value: 16#0.

Operating principle:

The list of the variable parameters is defined in the library in the UDT12 by the structure Data. For uploading the parameters a structure from the UDT12 has to be created in a Shared DB and this structure has to be hand over to the input Data of the function block. The read parameters will be written into this structure.

Note:

With the two blocks MC_UploadParameter_LXM05 and [MC_DownloadParameter_IFx](#), a defective device can be exchanged without a special tool to parameterize the device.

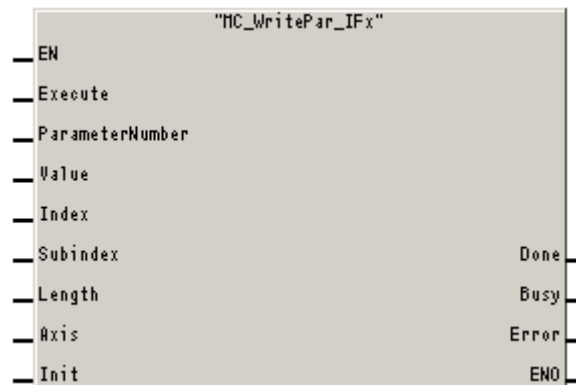
8.13 Write parameter

8.13.1 MC_WriteParameter_IFx

Task:

Writing an Object from the [Deviceparameter list](#).

Calling:



Parameter description:

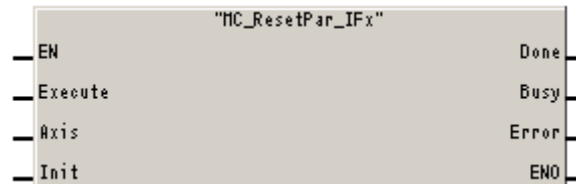
Par. type	Parameter	Data type	Description
IN	Execute	BOOL	FALSE: Deletes the output parameter when block has been executed. TRUE: Rising edge starts block execution.
	ParameterNumber	INT	0: The parameter is selected with <i>Index</i> . Other numbers are not supported. Value range: 0..32767, initial value: 0.
	Value	DINT	Value of the parameter to be written. Value range: -2147483648..2147483647, initial value: 0.
	Index	INT	Index of the Object to be read – the Objects are listed in the manual with their index and sub-index. Only valid with <i>ParameterNumber</i> = 0. Value range: 0..32767, initial value: 0.
	sub-index	INT	Sub-index of the Object to be read – the Objects are listed in the manual with their index and sub-index. Only valid with <i>ParameterNumber</i> = 0. Value range: 0..32767, initial value: 0.
	Length	INT	Length of the parameter to be written [bytes]. Value range: 1..4, initial value: 0.
IN_OUT	Axis	STRUCT	Axis reference [AxisDB.AxisReference].
	Init	BOOL	Initialisation bit [Init.I0 .. Init.I63].
OUT	Done	BOOL	TRUE: Block execution was completed successfully.
	Busy	BOOL	TRUE: Block is being executed.
	Error	BOOL	TRUE: An error has occurred during execution.

8.13.2 MC_ResetParameters_IFx

Task:

Resetting the [User parameters](#) to the factory settings.

Calling:



Parameter description:

Par. type	Parameter	Data type	Description
IN	Execute	BOOL	FALSE: Deletes the output parameter when block has been executed. TRUE: Rising edge starts block execution.
	Axis	STRUCT	Axis reference [AxisDB.AxisReference].
IN_OUT	Init	BOOL	Initialisation bit [Init.I0 .. Init.I63].
	Done	BOOL	TRUE: Block execution was completed successfully.
OUT	Busy	BOOL	TRUE: Block is being executed.
	Error	BOOL	TRUE: An error has occurred during execution.

Operating principle:

All the [User parameters](#) are reset to their default values, and stored in the EEPROM. To enable the default values, the device must be switched off/on.

Note:

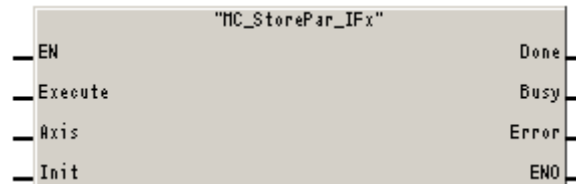
All parameter values adjusted by the user will be lost if a backup has not been previously made with the commissioning software.
Enabling the default settings is only possible with the drive at standstill.

8.13.3 MC_StoreParameters_IFx

Task:

Saving all the [User parameters](#) in a non-volatile memory (EEPROM).

Calling:



Parameter description:

Par. type	Parameter	Data type	Description
IN	Execute	BOOL	FALSE: Deletes the output parameter when block has been executed. TRUE: Rising edge starts block execution.
	Axis	STRUCT	Axis reference [AxisDB.AxisReference].
IN_OUT	Init	BOOL	Initialisation bit [Init.I0 .. Init.I63].
	Done	BOOL	TRUE: Block execution was completed successfully.
OUT	Busy	BOOL	TRUE: Block is being executed.
	Error	BOOL	TRUE: An error has occurred during execution.

Operating principle:

All the [User parameters](#) are stored in a non-volatile memory (EEPROM).

Note:

Saving the [User parameters](#) is only possible with the drive at standstill.

8.13.4 MC_DownloadParameter_IFx

Task:

Writing all variable parameters out of the [parameter list](#) to the drive.

Calling:



Parameter description:

Par-typ	Parameter	Datentyp	Bedeutung
IN	Execute	BOOL	FALSE: Deletes the output parameter when block has been executed. TRUE: Rising edge starts block execution.
	Data	ANY	Structure with the writing data.
IN_OUT	Axis	STRUCT	Axis reference [AxisDB.AxisReference].
	Init	BOOL	Initialisation bit [Init.I0 .. Init.I63].
OUT	Done	BOOL	TRUE: Block execution was completed successfully.
	Busy	BOOL	TRUE: Block is being executed.
	Error	BOOL	TRUE: An error has occurred during execution.
	Errorinfo	DWORD	Additional error information, address of the error. High Word: DB-Number, Low Word: parameter address in the DB Value range: 16#0..16#FFFFFFF, initial value: 16#0.

Operating principle:

The list of the variable parameters is defined in the library by the UDT13. For downloading the parameters a structure from the UDT13 has to be created in a Shared DB and this structure has to be hand over to the input Data of the function block. The parameters to be written will be taken out of this structure.

Note:

It is recommended to make an upload of all parameters with the function block [MC UploadParameters_IFx](#) before downloading the parameters within this function block. The initialisation values of the parameter list may not be equal to the default values of those in the drive. After uploading the parameters it is possible to change the data in the defined structure and then execute the download.

8.14 Inputs/outputs

Apart from the process image, in which the digital inputs and outputs of the target system are displayed, other blocks are available, which provide access to the digital inputs/outputs of every drive in the system. The drive's 24V signal interface provides 4 programmable inputs and/or outputs, which can be assigned with functions.

8.14.1 MC_ReadDigitalInput_IFx

Task:

Reading the drive's current input status.

Calling:



Parameter description:

Par. type	Parameter	Data type	Description
IN	Enable	BOOL	FALSE: Terminates block execution. TRUE: Starts block execution.
	InputNumber	INT	Number of the input that is to be read: 0: IOU 1: IOU 2: IOU 3: IOU Value range: 0..3, initial value: 0.
IN_OUT	Axis	STRUCT	Axis reference [AxisDB.AxisReference].
	Init	BOOL	Initialisation bit [Init.I0 .. Init.I63].
OUT	Valid	BOOL	TRUE: The value to be read is available.
	Busy	BOOL	TRUE: Block is being executed.
	Error	BOOL	TRUE: An error has occurred during execution.
	Value	BOOL	TRUE: The read input (InputNumber) has a 24V signal level. FALSE: The read input (InputNumber) has a 0V signal level.
	Inputs	WORD	Overall input status (regardless of InputNumber): IO0 = Bit 0, IO1 = Bit 1, IO2 = Bit 2, IO3 = Bit 3. Value range: 00h..0Fh, initial value: 00h.

8.14.2 MC_ReadDigitalOutput_IFx

Task:

Reading the drive's current output status.

Calling:



Parameter description:

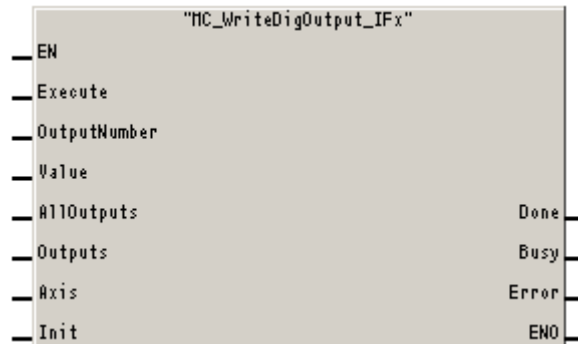
Par. type	Parameter	Data type	Description
IN	Enable	BOOL	FALSE: Terminates block execution. TRUE: Starts block execution.
	OutputNumber	INT	Number of the output that is to be read: 0: IOU 1: IOU 2: IOU 3: IOU Value range: 0..3, initial value: 0.
IN_OUT	Axis	STRUCT	Axis reference [AxisDB.AxisReference].
	Init	BOOL	Initialisation bit [Init.I0 .. Init.I63].
OUT	Valid	BOOL	TRUE: The value to be read is available.
	Busy	BOOL	TRUE: Block is being executed.
	Error	BOOL	TRUE: An error has occurred during execution.
	Value	BOOL	TRUE: The read output (OutputNumber) has a 24V signal level. FALSE: The read output (OutputNumber) has a 0V signal level.
	Outputs	WORD	Overall output status (regardless of OutputNumber): IO0 = Bit 0, IO1 = Bit 1, IO2 = Bit 2, IO3 = Bit 3 Value range: 00h..0Fh, initial value: 00h.

8.14.3 MC_WriteDigitalOutput_IFx

Task:

Writing the drive's output status.

Calling:



Parameter description:

Par. type	Parameter	Data type	Description
IN	Execute	BOOL	FALSE: Deletes the output parameter when block has been executed. TRUE: Rising edge starts block execution.
	OutputNumber	INT	Number of the output that is to be written: 0: IO 0: 1: IO 1: 2: IO 2: 3: IO 3: Value range: 0..3, initial value: 0.
	Value	BOOL	FALSE: Resets the output (0V signal level) TRUE: Sets the output (24V signal level)
	AllOutputs	BOOL	FALSE: Writes one output (OutputNumber, value is valid) TRUE: Write all outputs (outputs valid)
	Outputs	WORD	Overall output status (regardless of OutputNumber): IO0 = Bit 0, IO1 = Bit 1, IO2 = Bit 2, IO3 = Bit 3 Value range: 00h..0Fh, initial value: 00h.
IN_OUT	Axis	STRUCT	Axis reference [AxisDB.AxisReference].
	Init	BOOL	Initialisation bit [Init.IO .. Init.I63].
OUT	Done	BOOL	TRUE: Block execution was completed successfully.
	Busy	BOOL	TRUE: Block is being executed.
	Error	BOOL	TRUE: An error has occurred during execution.

Note:

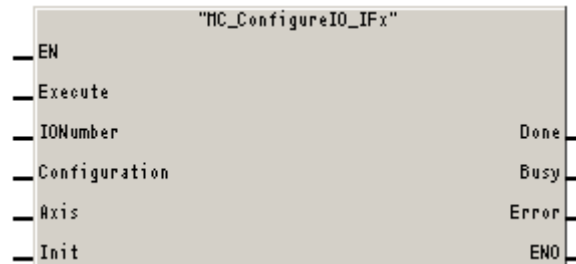
This block can only be used for I/Os that have been configured as outputs (see manual or description in the next section).

8.14.4 MC_ConfigureIO_IFx

Task:

Configuring and enabling the inputs/outputs.

Calling:



Parameter description:

Par. type	Parameter	Data type	Description
IN	Execute	BOOL	FALSE: Deletes the output parameter when block has been executed. TRUE: Rising edge starts block execution.
	IONumber	INT	Number of the input/output that is to be configured: 0: IO 0: 1: IO 1: 2: IO 2: 3: IO 3: Value range: 0..3, initial value: 0.
	Configuration	INT	Configuration: 0: Input; freely usable 1: LIMP; input for positive limit switch (only possible for IO0) 2: LIMN; input for negative limit switch (only possible for IO1) 3: STOP; Stop input 4: REF; input for reference switch 5: Input programmable (see MC_ControlIO_IFx) 128: Output; freely usable 129: Output; index pulse (only possible for IO0) 130: Output programmable (see MC_ControlIO_IFx)
IN_OUT	Axis	STRUCT	Axis reference [AxisDB.AxisReference].
	Init	BOOL	Initialisation bit [Init.IO .. Init.I63].
OUT	Done	BOOL	TRUE: Block execution was completed successfully.
	Busy	BOOL	TRUE: Block is being executed.
	Error	BOOL	TRUE: An error has occurred during execution.

Note:

The functionality of inputs LIMP, LIMN, STOP, and REF is automatically enabled with the corresponding configuration.

8.14.5 MC_ControlIO_IFx

Task:

Programming the configurable inputs and outputs ([MC_ConfigureIO_IFx](#)).

Calling:



Parameter description:

Par. type	Parameter	Data type	Description
IN	Execute	BOOL	FALSE: Deletes the output parameter when block has been executed. TRUE: Rising edge starts block execution.
	IONumber	INT	Number of the input/output that is to be configured: 0: IO 0: 1: IO 1: 2: IO 2: 3: IO 3: Value range: 0..3, initial value: 0.
	Index	INT	Index of the parameter to be controlled. Programmable input: Index of the parameter to be written. Programmable output: Index of the parameter to be read. Value range: 0..32767, initial value: 0.
	sub-index	INT	Sub-index of the parameter to be controlled. Programmable input: Sub-index of the parameter to be read. Programmable output: Sub-index of the parameter to be read. Value range: 0..32767, initial value: 0.
	BitMask	DWORD	Bitmask for the parameter value. Bitmask for a logical AND operation with the reading value before further processing. Special case: With bitmask 0, the reading value remains unchanged (as for 16#FFFFFFFF). Value range: 0..16#FFFFFFFF, initial value: 0.

Par. type	Parameter	Data type	Description
IN	Switch	INT	Edge detection or comparison operator. Programmable input: Selection of the edges to be detected 0: No response to signal level change 1: Response to rising edge (Value 1) 2: Response to falling edge (Value 2) 3: Response to both edges (Value 1 and Value 2) Programmable output: Selection of the comparison condition 0: Reading value = comparison value (Value 1) 1: Reading value <> comparison value (Value 1) 2: Reading value < comparison value (Value 1) 3: Reading value > comparison value (Value 1) Value range: 0..3, initial value: 0.
	Value1	DINT	Write value for rising edge or comparison value Programmable input: Write value for rising edge Programmable output: Comparison value for condition Value range: -2147483648..2147483647, initial value: 0.
	Value2	DINT	Write value for falling edge Programmable input: Write value for falling edge Programmable output: no meaning Value range: -2147483648..2147483647, initial value: 0.
IN_OUT	Axis	STRUCT	Axis reference [AxisDB.AxisReference].
	Init	BOOL	Initialisation bit [Init.I0 .. Init.I63].
OUT	Done	BOOL	TRUE: Block execution was completed successfully.
	Busy	BOOL	TRUE: Block is being executed.
	Error	BOOL	TRUE: An error has occurred during execution.

Operating principle:

Configuration of "programmable input"

The drive continuously monitors the input *IONo*, and executes parameters accesses independently as a function of bitmask and edge detection.

Parameter access always has the following sequence:

1. Rising or falling edge detected (*Switch*) at the input (*IONo*).
2. Read the parameter (*Index*, *Sub-index*).
3. AND operation of the reading value using the bitmask (*BitMask*).
4. OR operation of the result using the write value of a rising (*Value1*) or falling edge (*Value2*).
5. Write the result to the parameter (*Index*, *Sub-index*).

Configuration of "programmable output"

The drive cyclically reads the parameter and sets the output as a function of bitmask and comparison operator *IONo*.

Parameter access always has the following sequence:

1. Read the parameter (*Index*, *Sub-index*).
2. AND operation of the reading value using the bitmask (*BitMask*).
3. Comparison of the result with the comparison value (*Value1*).
4. Depending of the result, the output is set (*IONo*) HIGH or LOW (*Switch*).

8.15 Error handling

For the purpose of error handling, every block has an output *Error*, which is set if a [synchronous](#) or [asynchronous](#) error occurs. For a more detailed analysis of the error's cause, the block MC_ReadAxisError_IFx is called. By means of MC_Reset_IFx, the error cell is cleared to make it available for future error messages.

8.15.1 MC_ReadAxisError_IFx

Task:

Reading the [error message](#) of a drive.

Calling:



Parameter description:

Par. type	Parameter	Data type	Description
IN	Enable	BOOL	FALSE: Terminates block execution. TRUE: Starts block execution.
IN_OUT	Axis	STRUCT	Axis reference [AxisDB.AxisReference].
	Init	BOOL	Initialisation bit [Init.I0 .. Init.I63].
OUT	Valid	BOOL	TRUE: The value to be read is available.
	Busy	BOOL	TRUE: Block is being executed.
	Error	BOOL	TRUE: An error has occurred during execution.
	ErrorID	INT	0: No error message in the error cell >0: Error number (see list of error numbers). Value range: 0..32767, initial value: 0.

8.15.2 MC_Reset_IFx

Task:

Error acknowledgement.

Calling:



Parameter description:

Par. type	Parameter	Data type	Description
IN	Execute	BOOL	FALSE: Deletes the output parameter when block has been executed. TRUE: Rising edge starts block execution.
	Axis	STRUCT	Axis reference [AxisDB.AxisReference].
IN_OUT	Init	BOOL	Initialisation bit [Init.I0 .. Init.I63].
	Done	BOOL	TRUE: Block execution was completed successfully.
OUT	Busy	BOOL	TRUE: Block is being executed.
	Error	BOOL	TRUE: An error has occurred during execution.

Operating principle:

The error cell is cleared to make it available for future error messages, provided that the cause of the error has been rectified.

If the motor has been stopped by the automatic error response, it will be enabled again, provided that the cause of the error has been rectified when the error message is acknowledged.

Note

Only the first occurred error is entered in the free error cell, in order to permit conclusions to be drawn about the error's cause. As long as the error cell is occupied, previous error numbers are not overwritten (also not if the cause of the error has already been rectified), so that no new errors are entered.

9 Glossary

User parameters

Parameters that can be changed by the user, and remain in the memory after the drive has been switched off. These parameters are always marked with **per** in the manual.

[MC_ResetParameters_IFx](#)

[MC_StoreParameters_IFx](#)

Asynchronous error

Asynchronous errors occur independently of the programme sequence, such as an activated limit switch or motor overtemperature, for example.

[Error handling](#)

Resolution

The positioning resolutions referred to the motor output shaft (without gearing) are:

- For IFA: 16384 increments per revolution
- For IFE: 12 increments per revolution
- For IFS: 20000 increments per revolution

Movement profile generator

From the parameters for acceleration, deceleration, set speed, set and actual position, the movement profile generator calculates a position/timing diagram that indicates the motor's setpoint position at any time of the movement. This profile is processed by the drive control during the movement.

Error class

The device response depends on the severity of an error:

Class	Response	Description
0	Warning	Message only, no interruption of movement mode.
1	Quick stop	Motor stops, power amplifier and control remain active.
2	Switch-off	Motor standstill, power amplifier is switched off when motor is at standstill.
3	Fatal error	Power amplifier is switched off immediately
4	Uncontrolled operation	Power amplifier is switched off immediately, device must be switched off.

Error cell

The error cell contains the error code and the error class of an error that has occurred. A newly occurred error will be entered, provided that the error cell is free (i.e. equal to zero). If the error cell is occupied (i.e. not equal to zero), the previous error message will not be overwritten – instead, the new error message is ignored. The error cell is cleared by executing the block MC_Reset_IFx, provided that the cause of the error has been rectified.

Device parameter list or Object list

List of all the parameters in the device that can be accessed for reading or writing. The parameters are described in the device documentation.

[MC_WriteParameter_IFx](#)

[MC_ReadParameter_IFx](#)

Inc, Inc/s

Stands for "increments" or "increments per second".

Referred to the motor, this represents the resolution of the power amplifier, with which the motor can be positioned (without taking any gearing into account).

Resolution of the drives:

- IFA: 16384 increments per revolution
- IFE: 12 increments per revolution
- IFS: 20000 increments per revolution

Drive speed results from the number of increments per second [Inc/s].

Synchronous error

Synchronous errors occur during writing of parameters or starting of functions, and are related to an action, for example writing an impermissible parameter value or starting a movement with disabled motor current.

[Error handling](#)

10 List of error numbers

ErrorID hex	ErrorID dec	Error class	Drive error messages
0100h	256	2	Undervoltage 1 power supply
0101h	257	3	Undervoltage 2 power supply
0102h	258	3	Overvoltage power supply
0105h	261	3	Motor overload
010Ch	268	2	Power amplifier overtemperature
0110h	272	3	Motor blocked or stalled
0111h	273	3	Tracking error
0112h	274	4	Motor position sensor defective
0115h	277	1	Protocol error fieldbus
0116h	278	2	fieldbus: Nodeguarding/Watchdog or Clear
0117h	279	3	Frequency at pulse/direction input too high
0118h	280	3	Short circuit dig. outputs
0119h	281	3	Safety function "Power Removal" tripped (PWRR_A, PWRR_B)
011Ah	282	4	PWRR_A and PWRR_B inputs at different signal levels for >1 second
011Ch	284	4	Hardware error EEPROM
011Dh	285	4	Start-up error
011Eh	286	4	Internal system error
011Fh	287	4	Watchdog
0120h	288	0	Warning position overrun profile generator
0121h	289	0	Warning overtemperature IGBTs
0128h	296	0	Warning I/O timing
0130h	304	0	Parameter does not exist, invalid index
0131h	305	0	Parameter does not exist, invalid sub-index
0132h	306	0	Communication protocol: unknown service
0133h	307	0	Parameter not writable
0134h	308	0	Parameter out of range
0135h	309	0	Segment service not initialised
0136h	310	0	Error with recording function
0137h	311	0	Status not Operation Enable
0138h	312	0	Processing in current operating status of state machine not possible
0139h	313	0	Setpoint position generation interrupted
013Ah	314	0	Switchover during axis operating mode not possible
013Bh	315	0	Command not allowed during processing (xxxx_end=0)
013Ch	316	0	Error in selection parameter
013Dh	317	0	Position overrun exists/occurred
013Eh	318	0	Actual position is not yet defined
013Fh	319	4	EEPROM not initialised
0140h	320	4	EEPROM not compatible to current software
0141h	321	4	EEPROM read error
0142h	322	4	EEPROM write error
0143h	323	4	Checksum error in EEPROM

Step 7 Motion library for IFS, IFE, and IFA

ErrorID hex	ErrorID dec	Error class	Drive error messages
0144h	324	0	Non-calculable value
0145h	325	0	Function only allowed at standstill
0146h	326	0	Reference movement is active
0147h	327	0	Command not allowed during processing (xxx_end=0)
0148h	328	1	RS 485 interface: overrun error
0149h	329	1	RS 485 interface: framing error
014Ah	330	1	RS 485 interface: parity error
014Bh	331	1	RS 485 interface: receive error
014Ch	332	1	RS 485 interface: buffer overrun
014Dh	333	1	RS 485 interface: protocol error
014Eh	334	1	Node guarding, interface no longer serviced
014Fh	335	0	"Quick Stop" status is enabled
0150h	336	1	Illegal limit switch active
0151h	337	1	Switch was overrun, retraction impossible
0152h	338	1	Switching edge within run-off not found
0153h	339	1	Index pulse not found
0154h	340	1	Reproducibility of the index pulse movement uncertain, index pulse motion too close to the switch
0155h	341	1	Switch still active after retraction, cause possible bouncing of switch
0156h	342	1	Input not configured as LIMP/LIMN/REF
0157h	343	1	Interruption or "Quick Stop" via LIMP
0158h	344	1	Interruption or "Quick Stop" via LIMN
0159h	345	1	Interruption or "Quick Stop" via REF
015Ah	346	1	Interruption or "Quick Stop" via STOP
015Bh	347	1	Limit switch not enabled
015Ch	348	0	Processing not allowed in current axis operating mode
015Dh	349	0	Parameters not available with this device
015Eh	350	0	Function not available with this device
015Fh	351	0	Access denied
0160h	352	4	Production data in EEPROM not compatible with current software
0161h	353	4	Index pulse sensor not compensated
0162h	354	0	Drive is not referenced
0163h	355	0	CAN interface COB-ID not correct
0164h	356	0	CAN interface Error in query
0165h	357	0	CAN interface overrun error
0166h	358	0	CAN interface telegram could not be stored
0167h	359	0	CAN interface general error CAN stack
0168h	360	0	fieldbus: Data type and parameter length do not match
0169h	361	0	Blocking detection is switched off
016Ah	362	0	Connection failure to DSP boot loader
016Bh	363	0	Communication error to DSP boot loader
016Ch	364	0	Error initialising SPC3 memory
016Dh	365	0	Error in calculation of the length of input/output data
016Eh	366	0	Specified Profibus address is outside legal range
016Fh	367	0	Illegal use of DIP switch S1.1

Step 7 Motion library for IFS, IFE, and IFA

ErrorID hex	ErrorID dec	Error class	Drive error messages
0170h	368	0	DSP software not compatible with Profibus software
0171h	369	0	Checksum of Profibus DP interface software incorrect
0172h	370	0	Oscilloscope function: no other data available
0173h	371	0	Oscilloscope function: trigger variable was not defined
0174h	372	0	Oscilloscope function incompletely configured
0175h	373	1	Internal communication
0177h	375	1	Interruption or "Quick Stop" via software limit switch for clockwise rotation
0178h	376	1	Interruption or "Quick Stop" via software limit switch for counterclockwise rotation
Library error messages			
16#FF00	65280		Wrong_AxisRef
16#FF01	65281		Initialization_Failed
16#FF02	65282		Wrong_Data_Length
16#FF03	65283		Timeout
16#FF04	65284		Axis_Busy
16#FF05	65285		Invalid_Parameter_Number
16#FF06	65286		Unknown_State
16#FF07	65287		Capture_Busy
16#FF08	65288		Trigger_Event_Lost
16#FF09	65289		Axis_Not_In_Standstill
16#FF0A	65290		Unknown_Device_Type
16#FF0B	65291		Wrong_Data_Struct
16#FF20	65312		C/S specifier not valid or unknown
16#FF21	65313		Attempt to write a read only Object
16#FF22	65314		Object does not exist in the Object dictionary
16#FF23	65315		Data type does not match, length of service parameter does not match
16#FF24	65316		Sub-index does not exist
16#FF25	65317		Value range of parameter exceeded (only for write access)
16#FFFF	65535		Unknown_Error

11 Parameter list for Up- and Download function

[MC UploadParameter IFx](#)
[MC DownloadParameter IFx](#)

Name of parameter	Profibus address	Drive
timeout	1:11	IFA, IFE, IFS
name1	11:1	IFA, IFE, IFS
name2	11:2	IFA, IFE, IFS
I_still	14:1	IFS
I_acc	14:2	IFS
I_const	14:3	IFS
I_stop	14:4	IFS
monitorM	14:7	IFS
I_max	15:3	IFA, IFE
I_maxStop	15:4	IFA
I_maxBlk	15:5	IFE
p_maxDiff	15:7	IFE
KPn	15:8	IFA, IFE
TNn	15:9	IFA, IFE
KPp	15:10	IFA, IFE
KFPp	15:11	IFA, IFE
T_block	15:12	IFE
p_win	15:15	IFA
p_winTime	15:16	IFA
p_MaxDif2	15:17	IFA
pscDamp	15:20	IFA
pscDelay	15:21	IFA
serBaud	22:1	IFA, IFE, IFS
serAdr	22:2	IFA, IFE, IFS
serFormat	22:3	IFA, IFE, IFS
SafeState	24:5	IFA, IFE, IFS
invertDir	28:6	IFA, IFE, IFS
WarnOvrn	28:11	IFA, IFE, IFS
SignEnabl	28:13	IFA, IFE, IFS
SignLevel	28:14	IFA, IFE, IFS
dec_Stop	28:21	IFA, IFE, IFS
Flt_pDif	28:24	IFA, IFE
v_target0	29:23	IFA, IFE, IFS
acc	29:26	IFA, IFE, IFS
IO0_def	34:1	IFA, IFE, IFS
IO1_def	34:2	IFA, IFE, IFS
IO2_def	34:3	IFA, IFE, IFS
IO3_def	34:4	IFA, IFE, IFS
progDelay	34:7	IFA, IFE, IFS

Step 7 Motion library for IFS, IFE, and IFA

IO0_Index	800:1	IFA, IFE, IFS
IO0_Subindex	800:2	IFA, IFE, IFS
IO0_BitMask	800:3	IFA, IFE, IFS
IO0_Switch	800:4	IFA, IFE, IFS
IO0_Value1	800:5	IFA, IFE, IFS
IO0_Value2	800:6	IFA, IFE, IFS
IO1_Index	801:1	IFA, IFE, IFS
IO1_Subindex	801:2	IFA, IFE, IFS
IO1_BitMask	801:3	IFA, IFE, IFS
IO1_Switch	801:4	IFA, IFE, IFS
IO1_Value1	801:5	IFA, IFE, IFS
IO1_Value2	801:6	IFA, IFE, IFS
IO2_Index	802:1	IFA, IFE, IFS
IO2_Subindex	802:2	IFA, IFE, IFS
IO2_BitMask	802:3	IFA, IFE, IFS
IO2_Switch	802:4	IFA, IFE, IFS
IO2_Value1	802:5	IFA, IFE, IFS
IO2_Value2	802:6	IFA, IFE, IFS
IO3_Index	803:1	IFA, IFE, IFS
IO3_Subindex	803:2	IFA, IFE, IFS
IO3_BitMask	803:3	IFA, IFE, IFS
IO3_Switch	803:4	IFA, IFE, IFS
IO3_Value1	803:5	IFA, IFE, IFS
IO3_Value2	803:6	IFA, IFE, IFS