Electric Drives and Controls | Hydraulics | Linear Motion and Assembly Technologies | Pneumatics | Service

Rexroth
Bosch Group

Rexroth PLCopen Function Blocks For Field Bus Drives

R911315058
Edition 01

Application Manual

Title
Rexroth PLCopen
Function Blocks
For Field Bus Drives

Type of Documentation
Application Manual

Document Typecode
DOK-CONTRL-PLCOPENFB*D-AW01-EN-P

Internal File Reference
Document Number, 120-0401-B327-01/EN

Purpose of Documentation
This documentation describes the PLCopen function blocks for field bus drives.

Record of Revisions

Description	Release Date	Notes
120-0401-B327-01/EN	04/06	First Edition

Published by
Bosch Rexroth AG
Bgm.-Dr.-Nebel-Str. 2 • D-97816 Lohr a. Main
Telephone +49 (0)93 52/40-0 • Tx 68 94 21 • Fax +49 (0)93 52/40-48 85
http://www.boschrexroth.com/
Dept. BRC/EPY (NH)

Note
This document has been printed on chlorine-free bleached paper.

Contents

1 MotionControl Libraries

1.1 General Information

The description of the MotionLogic function blocks starts with a brief introduction to the principal options of the MotionControl according to the PLCopen standard.

Chapter 1.5 and the following chapters contain a description of the data types which are included in the RIL_CommonTypes.lib and MP_PLCopenTypes.lib libraries.

The MP_PLCOpen.lib and PLCopenFieldBus.lib libraries contain the MotionControl modules.

An RIL_CommonTypes.lib library is contained in the targets. As an alternative, an MP_PLCOpenTypes.lib library (which is independent of the firmware) can be used instead of the RIL_CommonTypes.lib library.

Meaning of the Function Block Prefixes

Prefix MC_

This prefix means that this function block complies completely with the PLCopen standard.

Prefix MB_

This prefix means that this function block, although complying with the system structure of PLCopen, is different from or is a supplement to this system structure or is not defined at all.

Prefix MP_

This prefix means that this function block complies completely with the PLCopen standard.

Prefix MBP_

This prefix means that this function block, although complying with the system structure of PLCopen, is different from or is a supplement to this system structure or is not defined at all and is applicable to field bus drives only.

1.2 State Diagram

The diagram following below normatively defines the behavior of MotionControl axes from the user's viewpoint, when several MotionControl function blocks are activated at the same time. This combination of motion profiles is useful in the generation of complicated profiles and in the reaction to exceptional situations in programs.

In principle, motion commands are given and processed sequentially, even if the PLC is capable of parallel processing. These commands follow the state diagram of the MotionControl axes.

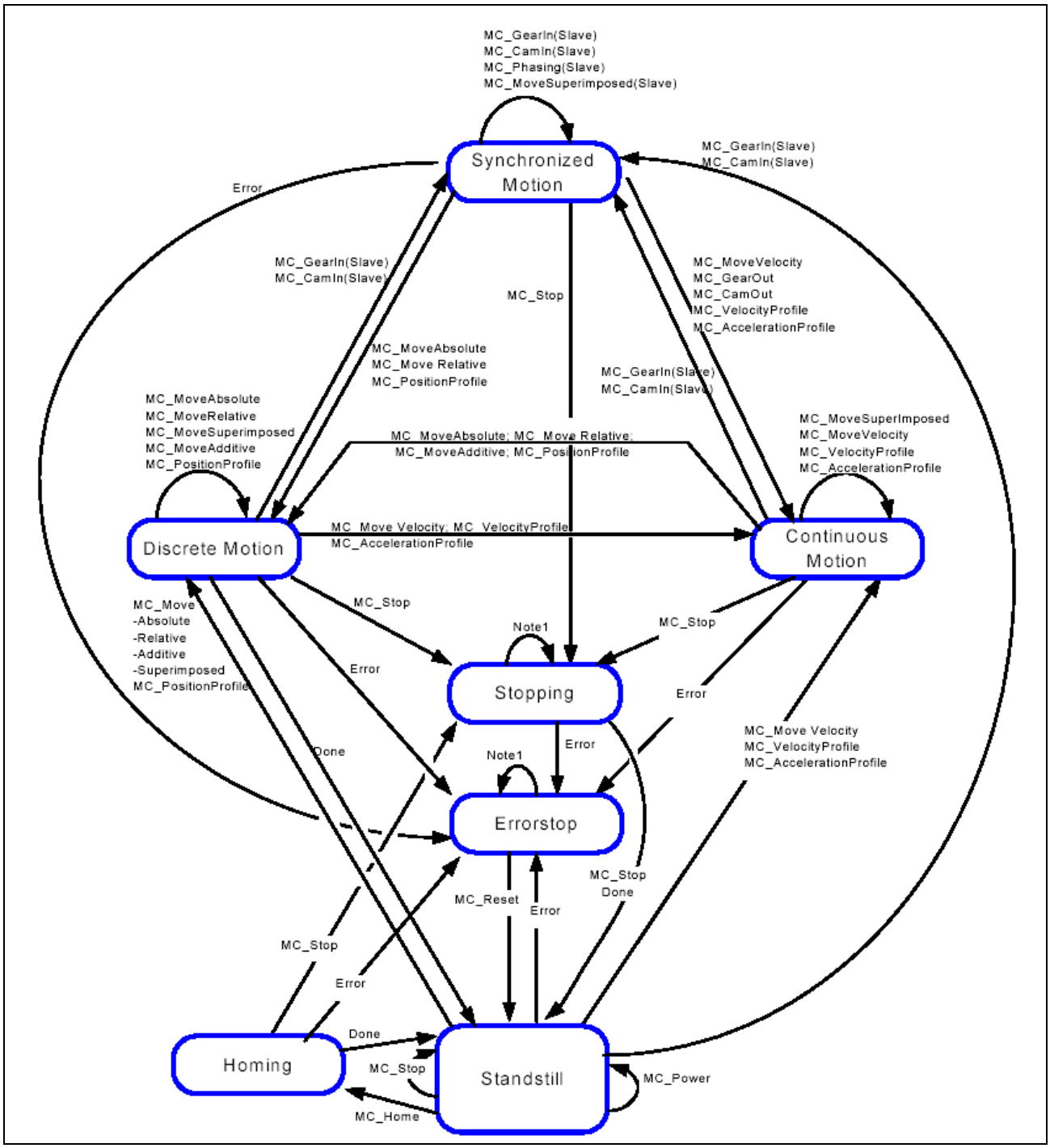
Each axis is in exactly one of the defined states (see the state diagram below). Each motion command represents a state transition for the axis concerned. The sequence of these transitions describes the entire behavior of the axis.
The diagram focuses on the individual axis. Multi-axis function blocks, such as MC_CamIn, MC_GearIn and MC_Phasing, can be considered such that each axis, as seen from the user's view, is in its specific state of the state diagram, e. g. the CAM master may be in the 'continuous motion' state, while the corresponding slave is in the 'synchronized motion' state.

Note: Single-axis function blocks are the only ones functioning for field bus drives!

When a slave axis is connected to follow a master axis, the master axis is not affected.

The following function blocks which have a "managing function" do not affect the state diagram:

- MC_ReadStatus, MC_ReadActualPosition
- MC_ReadAxisError



(1): All function blocks can be called without being processed, except for MC_Reset and Error, which generate the transition to StandStill or ErrorStop respectively.

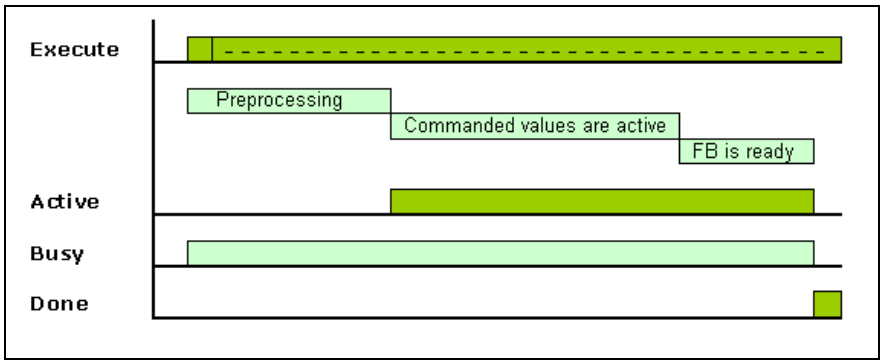
(2): If in a state other than Standstill, the output of MB_Home results in ErrorStop, except for Homing itself.

(3): The transition relates to errors which are caused by axes and axis control, but not to errors which are caused by FB instances. These axis errors are only reflected at the error outputs of the function block instances.

Fig. 1-1: MotionControl state diagram of a real axis

1.3 Command Processing in a PLCopen FB

The PLCopen function blocks use input and output signals with a defined function (also refer to Signal Time Diagrams of Motion Modules).



Execute: at least 1 PLC cycle; no more than as many cycles as desired

Fig. 1-2: Signal interplay of a PLCopen function block

"Execute" Input Signal

With its **rising edge,** the "Execute" signal initiates processing of the FB instance . The duration of the signal must correspond to at least one PLC cycle.

The "Execute" signal triggers the three phases of processing of the FB instance:

- **Preprocessing:** FB inputs are checked; values are precalculated; values are initialized; command values are sent; the operation mode is changed; acknowledgement from the target object is expected.
- **Command values are active:** The command values are available for processing at the target location (e. g. drive). Processing is in progress (communication channel active with communication modules; operation mode and command values active with motion modules).
- **FB is ready:** Processing at the target location is completed. The result is returned and made available at the outputs of the FB.

The three phases become evident at the outputs of the FB:

"Active" Output Signal

"Active" comprises the phases of "Command values are active" and "FB is ready". New information can be applied to a module not earlier than after the "Active" signal has appeared.

"Done" Output Signal

Processing of the order has been completed successfully (without errors). The "Done" signal is output as long as the "Execute" signal is TRUE. If the "Execute" signal was deactivated already beforehand, the "Done" signal is active for one PLC cycle only.

Note: The "Busy" signal that is recommended by PLCopen is not used. If necessary, however, it can be determined.

Alternatives:

"Enable" Input Signal

The order is executed as long as the signal is present, e. g. reading of data (MB_ReadParameter), power on (MC_Power).

"InVelocity", "InGear", "InSync" Output Signals

Contrary to the "Done" signal, these signals indicate that an order is currently executed (without errors). This indication is preserved as long as the "Execute" signal is active.

Note: The "Done", "Active", "InVelocity", "InGear" and "InSync" output signals are active only if the behavior of both the function block instance **and** the target object is without errors; in this case, the "Error" output is always FALSE.

TRUE at the "Error" output, however, excludes TRUE at one of the other outputs.

1.4 Signal Time Diagrams of Motion Modules

The signal time diagrams are binding when the following input and output signals are used:

Inputs Execute (alternatively Enable)

Outputs Done (alternatively InVelocity or InGear or InSync), Error

Edge-Triggered Function Blocks

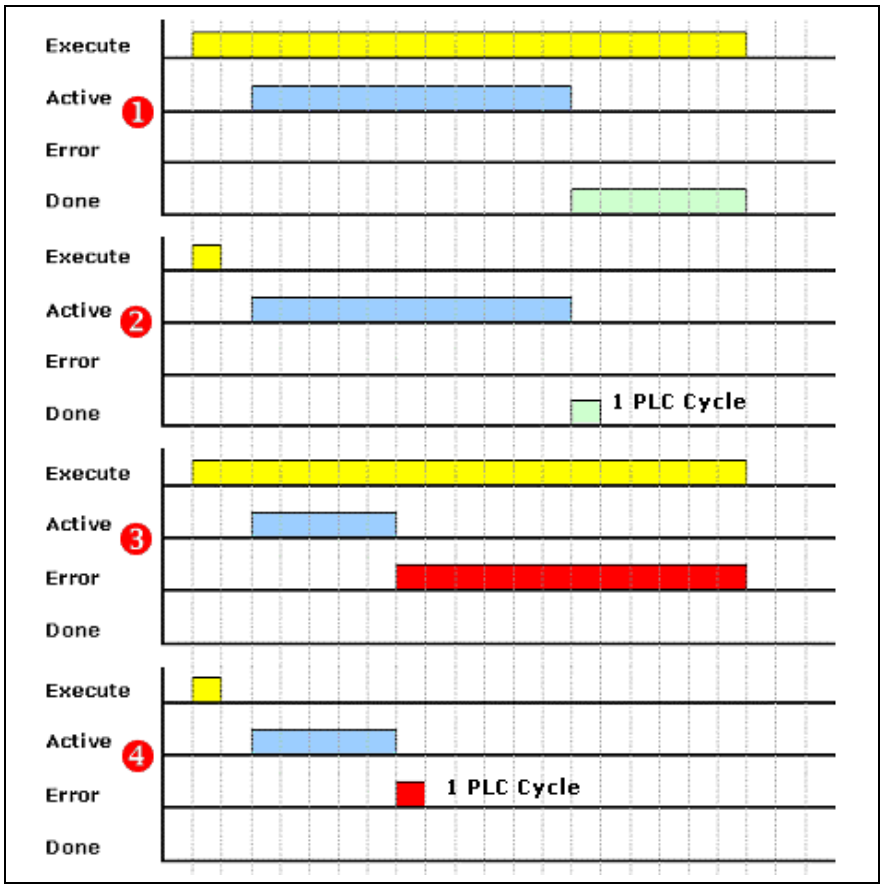


Fig. 1-3: Signal time behavior of edge-triggered function blocks

The signal time behavior is typical of the function blocks listed below:

- MB_WriteParameter
- MC_MoveAbsolute
- MC_MoveRelative
- MC_MoveAdditive
- MC_MoveVelocity

Note: If, in case 2, the "Execute" signal is applied as a pulse only (see Fig. 1-3), the "Done" signal is active for **one** PLC cycle only (InGear, InSync, InVelocity accordingly)!

If, in case 4, the "Execute" signal is applied as a pulse only (see Fig. 1-3), the "Error" signal is active for **one** PLC cycle only!

Status-Controlled Function Blocks

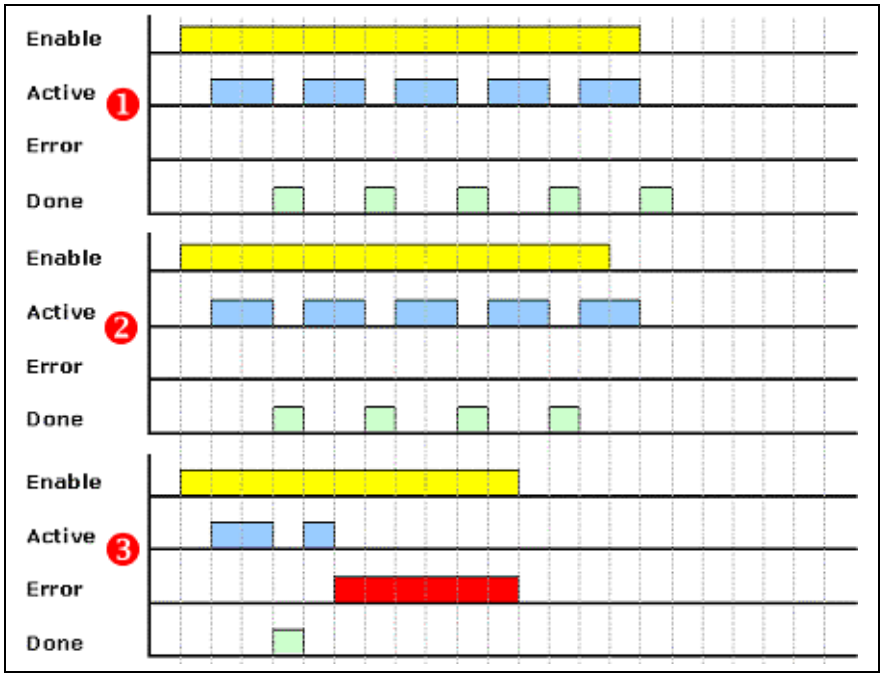


Fig. 1-4: Signal time behavior of status-controlled function blocks

The following function blocks are typical of the time behavior shown above:

- MC_ReadActualPosition
- MB_ReadParameter
- MC_ReadStatus
- MC_ReadAxisError

1.5 Data Types at the Interface between MotionControl Function Blocks (RIL_CommonTypes.lib and MP_PLCOpenTypes.lib)

General Information on Data Types

This section describes data types which are required for the input and output variables of the MotionControl function blocks.

Data types are included in the "RIL_CommonTypes.lib" library. The version of the library is identified by the "Version_RIL_CommonTypes_01V02" function. The "MP_PLCOpenTypes.lib" library meets the same purpose as the "RIL_CommonTypes.lib" library, with the only difference that it is independent of any target since there are no time data types. The structure of the library is as follows:

Function blocks				
	Version			
		Version_RIL_CommonTypes_xxVxx (FUN)		
Data types				
	Data types of logical addresses			
		CONTROLS (ENUM)		
		OBJECTS (ENUM)		
	Data types of PLCopen – AXIS_REF			
		AXIS_REF (STRUCT)		
		MC_CAM_ID (ENUM)		
		MC_START_MODE (ENUM)		
		MC_SYNC_MODE (ENUM)		
	Data types of POU diagnosis			
		Includes		
			ERROR_CODE (ENUM)	
			ERROR_TABLE (ENUM)	
		Fig. 1-10: Assignment of error tables in "ERROR_TABLE" ERROR_STRUCT (STRUCT)		
	Data types of times (NOT IN MP_PLCOpenTypes.lib)			
		SYS_TIME64 (STRUCT)		
		SYS_TIME_DATE (STRUCT)		
		EXT_SYS_TIME_DATE (STRUCT)		

Fig. 1-5: Structure of the "RIL_CommonTypes.lib" library

AXIS_REF

Brief Description AXIS_REF is a structure providing information on the corresponding axis.

	Name	Type	Initial value	Comment
TYPE	AXIS_REF			
STRUCT				
	CntrlNo	CONTROLS	LOCAL_CNTRL	Control number; default: local control
	AxisNo	OBJECTS	AXIS_1	Axis reference number
END_STRUCT				
END_TYPE				

Fig. 1-6: AXIS_REF (STRUCT)

Functional Description Information is transferred via VAR_IN_OUT to all function blocks which are defined in "MP_PLCopen.lib" or "PLCopenFieldBus.lib".

The AXIS_REF structure contains two 16-bit address elements:

- the control address (element of "CONTROLS") and
- the object address (element of "OBJECTS").

The elements can be addressed via the logical or the physical address. In case of a logical address, the element points to a parameter structure with a physical address (e. g. Ethernet address of the control).

CONTROLS

Brief Description The enumeration of connected controls contains all controls that can be addressed via MotionControl function blocks in the MotionControl network:

	Name		Value	Comment
TYPE	CONTROLS			Enumeration of valid logical control addresses
(				
	LOCAL_CNTRL	:=	0	Local control (default)
)				
END_TYPE				

Fig. 1-7: CONTROLS (ENUM) – example

Functional Description This enumeration of connected controls contains all controls that can be addressed via MotionControl function blocks in the MotionControl network. Field bus axes are always local (also refer to "Communicating Field Bus Axes to the PLC" on page 2-1).

- Number 0000 is enabled for the local control (MotionControl).
- Numbers 0001 … 0999 are reserved for all connected controls.

Note: In the event of physical addressing, there are two options of reaching the "local control": Number "0", local, and via its own address, e. g. CTRL_3.

Note: Field bus axes are always local, i. e. *CONTROLS = LOCAL_CNTRL*, or number 0000.

OBJECTS

Brief Description

This enumeration of available axes and axis groups contains all objects of a MotionControl network which can be addressed by means of MotionControl function blocks.

	Name		Value	Comment
TYPE	OBJECTS			Enumeration of valid axes and axis groups
(				
	NO_OBJECT	:=	0	No object defined
	AXIS_1	:=	1	Axis 1
	AXIS_2	:=	2	Axis 2
	...			
	FieldBusAxis_1	:=	6001	Field bus axis 1
	...			
	FieldBusAxis_99	:=	6099	Field bus axis 99
)				
END_TYPE				

Fig. 1-8: OBJECTS (ENUM) – example

Functional Description

This enumeration of available axes and axis groups contains all objects of a MotionControl network which can be addressed by means of MotionControl modules (also refer to "Communicating Field Bus Axes to the PLC" on page 2-1).

- Number 0000 is invalid (no object defined)
- 0001 .. 0999 are reserved for axes.
- 1001 .. 1999 are reserved for real master axes.
- 2001 .. 2999 are reserved for virtual master axes.
- 3001 .. 3999 are reserved for the ELS group local master.
- 4001 .. 4008 are reserved for ELS groups.
- 5001 .. 5006 are reserved for ELS system masters.
- 6001 .. 6099 are reserved for field bus axes.

ERROR_CODE

Brief Description This enumeration contains all error types characterizing an error at the function block.

The error code provides quick information on the cause of the error.

	Name	Value	Comment
TYPE	ERROR_CODE		Enumeration of valid error types
(			
	NONE_ERROR	:= 0	No error code available
	INPUT_INVALID_ERROR	:= 1	Invalid input
	COMMUNICATION_ERROR	:= 2	Communication error
	RESOURCE_ERROR	:= 3	Resource error
	ACCESS_ERROR	:= 4	Access error
	STATE_MACHINE_ERROR	:= 5	Error in the state machine
	INPUT_RANGE_ERROR	:= 6	Range exceeded by input
	CALCULATION_ERROR	:= 7	Calculation error
	DEVICE_ERROR	:= 8	Device error
	OTHER_ERROR	:= 254	General error
	SYSTEM_ERROR	:= 255	System error
)			
END_TYPE			

Fig. 1-9: Meaning of the error codes in "ERROR_CODE"

ERROR_TABLE

Brief Description The error table provides detailed information on system-specific errors. In addition, "Additional1" and "Additional2" can be evaluated in "ERROR_STRUCT" in some of the systems, for example in order to determine the error code.

	Name	Value	Comment
STRUCT			
	NO_TABLE_USED	16#0000	Detailed evaluation not possible
	SERCOS_TABLE	16#0010	SERCOS
	MLDS_TABLE	16#0020	Rexroth IndraMotion MLD (drive)
	MLC_TABLE	16#0030	Rexroth IndraMotion MLC (controller-based)
	MTX_TABLE	16#0040	Rexroth IndraMotion MTX
	MLP_TABLE	16#0050	Rexroth IndraMotion MLP (PC-based)
	PLC_TABLE	16#0060	Rexroth PLC
	INDRV_TABLE	16#0070	Rexroth IndraDrive
	DIAX_TABLE	16#0080	Rexroth DIAX
	ECO_TABLE	16#0090	Rexroth EcoDrive
	PB_DP_TABLE	16#0130	Profibus
	DEVICENET_TABLE	16#0140	DeviceNet
	ETHERNET_TABLE	16#0150	Ethernet
	INTERBUS_TABLE	16#0160	Interbus
	F_RELATED_TABLE	16#0170	Function-related
	USER1_TABLE	16#1000	Free user table
	USER2_TABLE	16#1001	Free user table
	USER3_TABLE	16#1002	Free user table
	USER4_TABLE	16#1003	Free user table
	USER5_TABLE	16#1004	Free user table
	USER6_TABLE	16#1005	Free user table
	USER7_TABLE	16#1006	Free user table
	USER8_TABLE	16#1007	Free user table
	USER9_TABLE	16#1008	Free user table
	USER10_TABLE	16#1009	Free user table
END_STRUCT			

Fig. 1-10: Assignment of error tables in "ERROR_TABLE"

ERROR_STRUCT

Brief Description The error structure provides any detailed information on an actual error. This information can be evaluated and read directly at the particular instance of an IndraMotion function block.

	Name	Type	Comment
STRUCT			
	Code	ERROR_TABLE	Table to which Additional1 and Additional2 refer
	Additional1	DWORD	For example, IndraDrive (ERROR_TABLE = INDRV_TABLE) signals: 16#000F6044 => "Negative assignment of travel range limit switch"
	Additional1	DWORD	
END_STRUCT			

Fig. 1-11: Data type: ERROR_STRUCT (STRUCT)

SYS_TIME64

The time stamp is based on the following structure. It contains the number of milliseconds elapsed since January 1, 1970 (also refer to SYS_TIME_DATE).

Note: Not included in MP_PLCOpenTypes.lib!

```
SYS_TIME64
0001 (* Structure of system time - Microseconds since 1970      *)
0002 TYPE SYS_TIME64 :
0003 STRUCT
0004     LowValue:    DWORD;        (* Low value [us]     *)
0005     HighValue:   DWORD;        (* High value [us]    *)
0006 END_STRUCT
0007 END_TYPE
```

Fig. 1-12: Time stamp in the form of "SYS_TIME64"

SYS_TIME_DATE

The structure reflects the system time or a time stamp in a readable form (also refer to SYS_TIME64).

Note: Not included in MP_PLCOpenTypes.lib!

```
SYS_TIME_DATE
0001 (* Structure of system time                              *)
0002 TYPE SYS_TIME_DATE :
0003 STRUCT
0004     Year : UINT;              (* Year                   *)
0005     Month : UINT;             (* Month                  *)
0006     Day : UINT;               (* Day                    *)
0007     Hour : UINT;              (* Hour                   *)
0008     Minute : UINT;            (* Minute                 *)
0009     Second : UINT;            (* Second                 *)
0010     Milliseconds : UINT;      (* Milliseconds           *)
0011     DayOfWeek: UINT;          (* Day of week            *)
0012 END_STRUCT
0013 END_TYPE
```

Fig. 1-13: System time in the form of "SYS_TIME_DATE"

EXT_SYS_TIME_DATE

This structure results from a combination of SYS_TIME64 and SYS_TIME_DATE.

Note: Not included in MP_PLCOpenTypes.lib!

```
EXT_SYS_TIME_DATE
0001 (* Extended structure of system time                    *)
0002 TYPE EXT_SYS_TIME_DATE :
0003 STRUCT
0004     LowValue: DWORD;          (* Low value [us]        *)
0005     HighValue: DWORD;         (* High value [us]       *)
0006     Year : UINT;              (* Year                  *)
0007     Month : UINT;             (* Month                 *)
0008     Day : UINT;               (* Day                   *)
0009     Hour : UINT;              (* Hour                  *)
0010     Minute : UINT;            (* Minute                *)
0011     Second : UINT;            (* Second                *)
0012     Milliseconds : UINT;      (* Milliseconds          *)
0013     DayOfWeek: UINT;          (* Day of week           *)
0014 END_STRUCT
0015 END_TYPE
```

Fig. 1-14: Extended system time in the form of "EXT_SYS_TIME_DATE"

1.6 Function Blocks for Parameter Access

Access to Data of a Drive or a Control

Structure of and Access to Parameters from the PLC Program

The data of a drive or the control involving these drives can be reached by accessing parameters in a reading or, in part, writing manner. The declarations of control parameters A and C are not included in MP_PLCOpen.lib nor in PLCOpenFieldBus.lib.

S and P parameters can be accessed through the field bus.

The structure of all MotionControl parameters is the same:

Example: S-0-0001, where S(, A, C, P) is the parameter type qualifier which is followed by the group number and the four-digit parameter number.

To this end, declaration files for global constants are available in MP_PLCOpen.lib for P and S parameters.

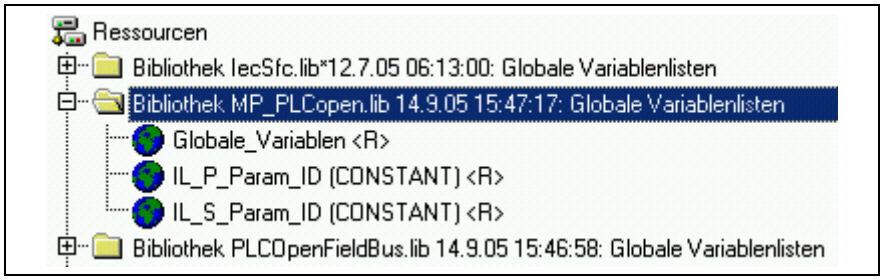


Fig. 1-15: Global constant declarations for parameter accesses

Parameter Number:

```
VAR_GLOBAL CONSTANT
  FP_S_0_0001: DINT := 1; (*NC_cycle_time_TNcyc_*)
  FP_S_0_0002: DINT := 2; (*SERCOS_cycle_time_TScyc_*)
  FP_P_0_0009: DINT := 32787; (*Initial_position_value*)
....
END_VAR
```

As an alternative of the constant itself, the number can be applied to the "ParameterNumber" input of the particular function block.

Parameter Offsets:

Parameter group	Decimal	Hexadecimal
S-0-0000	0	16#0
P-0-0000	32768	16#8000
A-0-0000	65536	16#10000
C-0-0000	131072	16#20000

Fig. 1-16: Parameter offsets

MC_ReadActualPosition

Brief Description

This function block reads the current position of the drive ("Actual position value" S-0-0051) or the parameter specified in the drive on the field bus process data input description.

Library	Range
MP_PLCOpen.lib	Parameters

Fig. 1-17: Library assignment

Interface Description

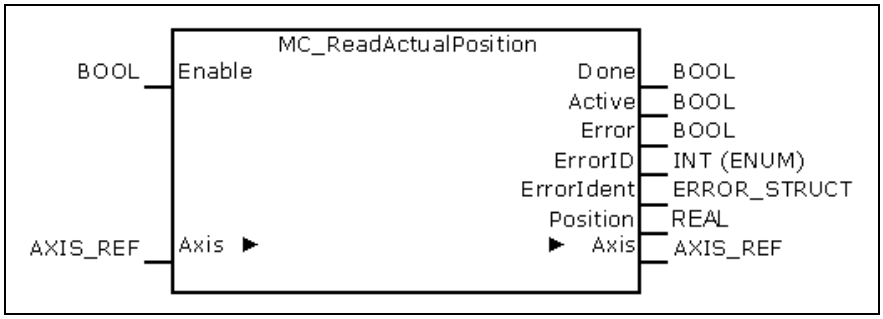


Fig. 1-18: FB MC_ReadActualPosition

	Name	Type	Comment
VAR_IN_OUT	Axis	AXIS_REF	Provides information on the actual axis.
VAR_INPUT	Enable	BOOL	Continuously reads the position value as long as "Enable" is TRUE.
VAR_OUTPUT	Done	BOOL	A valid value is available.
	Active	BOOL	Processing of data is active after preprocessing is completed.
	Error	BOOL	Indicates that an error has occurred in the FB instance.
	ErrorID	ENUM ERROR_CODE	Indicates the basic error cause in the event of an error, using the enumeration values of ERROR_CODE.
	ErrorIdent	STRUCT ERROR_STRUCT	Provides detailed information on the error; this structure consists of ERROR_TABLE, Additional1 and Additional2.
	Position	REAL	Current absolute position of the drive (in the axis unit [u]).

Fig. 1-19: Interface of FB MC_ReadActualPosition

Signal Time Diagram

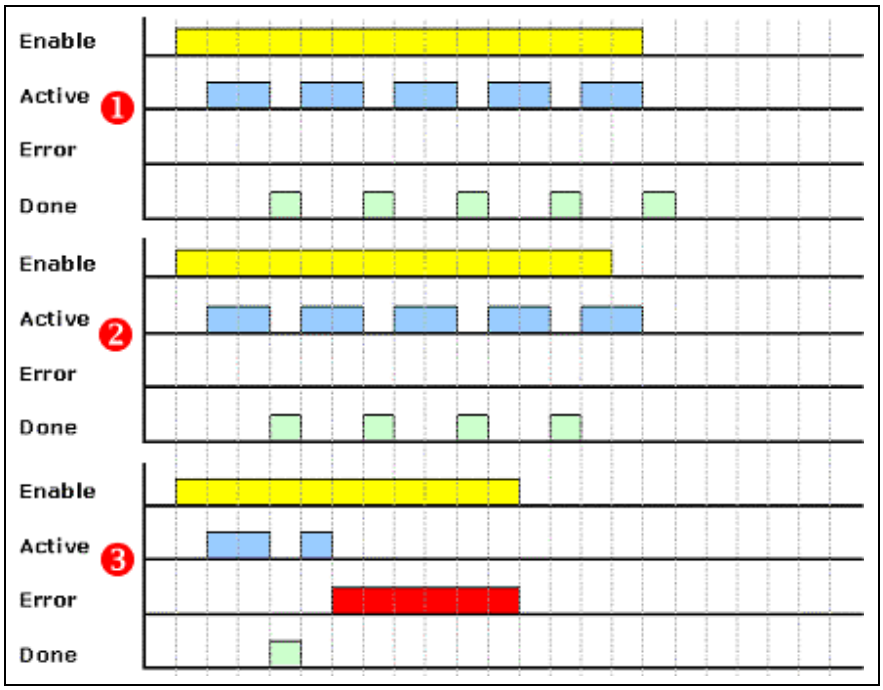


Fig. 1-20: Default signal time diagram

Error Handling

The MC_ReadActualPosition function block generates error messages only if there is a positive edge or TRUE at the "Enable" input.

MC_ReadStatus

Brief Description

This function block outputs the current status of the drive.

Library	Range
MP_PLCOpen.lib	Parameters

Fig. 1-21: Library assignment

Interface Description

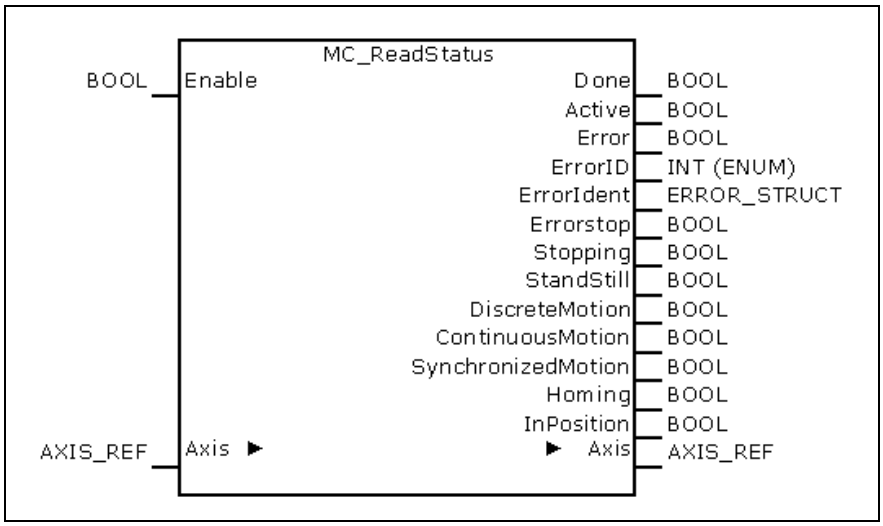


Fig. 1-22: FB MC_ReadStatus

	Name	Type	Comment
VAR_IN_OUT	Axis	AXIS_REF	Provides information on the actual axis.
VAR_INPUT	Enable	BOOL	Continuously reads the axis status as long as "Enable" is TRUE.
VAR_OUTPUT	Done	BOOL	A valid value is available.
	Active	BOOL	Processing of data is active after preprocessing is completed.
	Error	BOOL	Indicates that an error has occurred in the FB instance.
	ErrorID	ENUM ERROR_CODE	Indicates the basic error cause in the event of an error, using the enumeration values of ERROR_CODE.
	ErrorIdent	STRUCT ERROR_STRUCT	Detailed information on the error. This structure consists of ERROR_TABLE, Additional1 and Additional2
	Errorstop	BOOL	Drive in "Errorstop" state
	Stopping	BOOL	Drive in "Stopping" state
	StandStill	BOOL	Drive in "StandStill" state
	DiscreteMotion	BOOL	Drive in "DiscreteMotion" state
	ContinuousMotion	BOOL	Drive in "ContinuousMotion" state
	SynchronizedMotion	BOOL	Drive in "SynchronizedMotion" state
	Homing	BOOL	Drive in "Homing" state
	InPosition	BOOL	Drive in "InPosition" state

Fig. 1-23: Interface of FB MC_ReadStatus

Signal Time Diagram

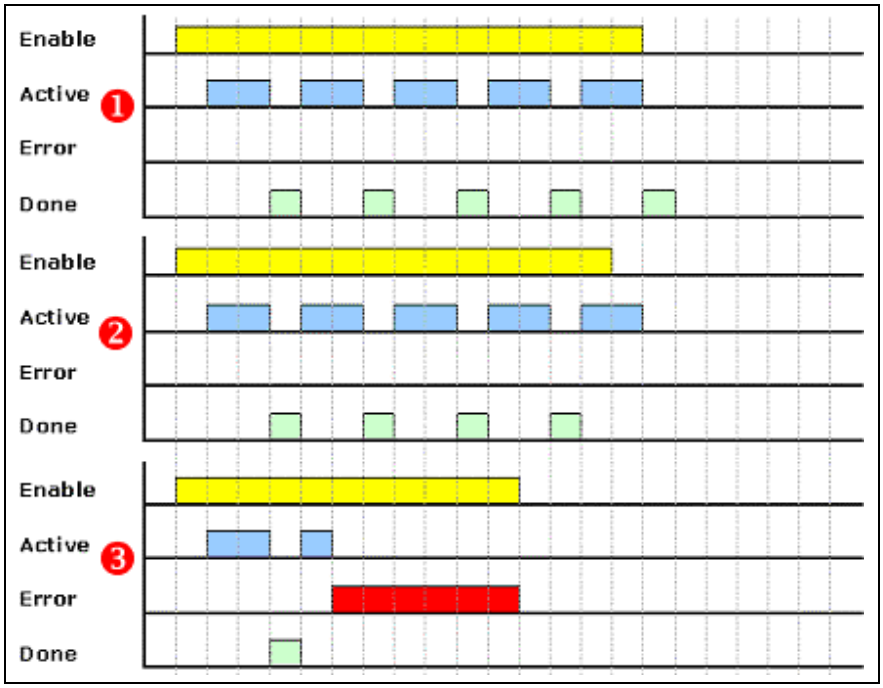


Fig. 1-24: Default signal time diagram

Error Handling

The MC_ReadActualPosition function block generates error messages only if there is a positive edge or TRUE at the "Enable" input.

MC_ReadAxisError

Brief Description

This function block retrieves the error status ("Axis diagnostic message" (A-0-0020)) of the axis addressed. One instance of the function block is to be used for each axis.

Library	Range
MP_PLCOpen.lib	Parameters

Fig. 1-25: Library assignment

Interface Description

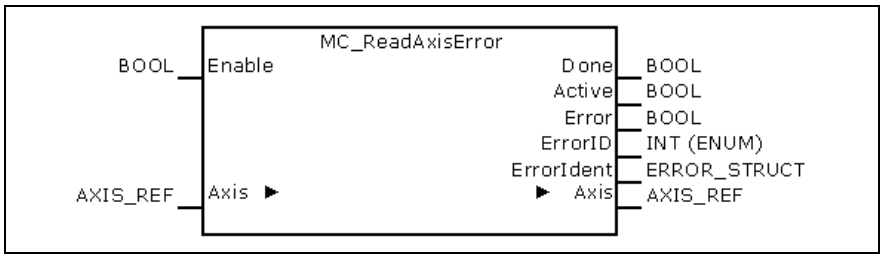


Fig. 1-26: FB MC_ReadAxisError

	Name	Type	Comment
VAR_IN_OUT	Axis	AXIS_REF	Provides information on the actual axis.
VAR_INPUT	Enable	BOOL	Continuously reads the error information as long as "Enable" is TRUE.
VAR_OUTPUT	Done	BOOL	Value available.
	Active	BOOL	Processing of data is active after preprocessing is completed.
	Error	BOOL	Indicates that an error has occurred when the FB instance was called.
	ErrorID	ENUM ERROR_CODE	Indicates the basic error cause in the event of an error, using the enumeration values of ERROR_CODE.
	ErrorIdent	STRUCT ERROR_STRUCT	Provides detailed information on the error; this structure consists of ERROR_TABLE, Additional1 and Additional2.

Fig. 1-27: Interface of FB MC_ReadActualPosition

Signal Time Diagram

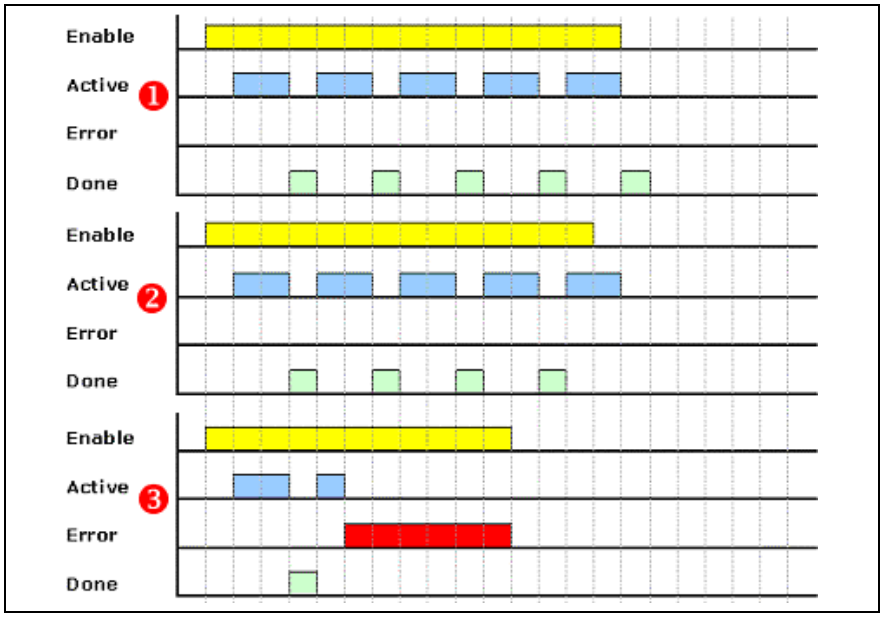


Fig. 1-28: Default signal time diagram

Error Handling

This function block generates error messages only if there is a positive edge or TRUE at the "Enable" input.

MB_ReadParameter

Brief Description

This function block returns the value of a user-specific parameter. The return value (4 bytes) is represented as a DINT number.

Library	Range
MP_PLCOpen.lib	Parameters

Fig. 1-29: Library assignment

Interface Description

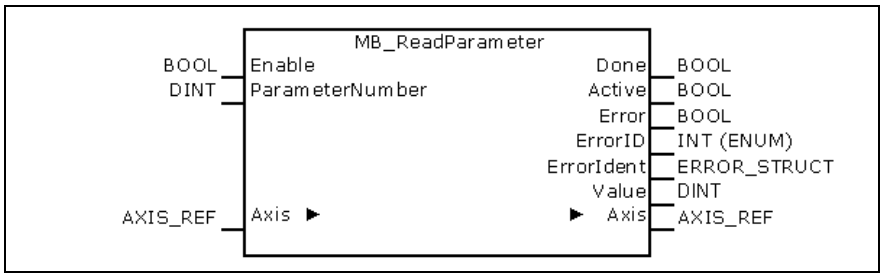


Fig. 1-30: FB MB_ReadParameter

	Name	Type	Comment
VAR_IN_OUT	Axis	AXIS_REF	Provides information on the actual axis.
VAR_INPUT	Enable	BOOL	Continuously reads the parameter as long as "Enable" is TRUE.
	Parameter number	DINT	Number of the parameter desired; refer to Structure of and Access to Parameters
VAR_OUTPUT	Done	BOOL	A valid value is available.
	Active	BOOL	Processing of data is active after preprocessing is completed.
	Error	BOOL	Indicates that an error has occurred when the FB instance was called.
	ErrorID	ENUM ERROR_CODE	Indicates the basic error cause in the event of an error, using the enumeration values of ERROR_CODE.
	ErrorIdent	STRUCT ERROR_STRUCT	Provides detailed information on the error; this structure consists of ERROR_TABLE, Additional1 and Additional2.
	Value	DINT	4-byte value represented as DINT

Fig. 1-31: Interface of FB MB_ReadParameter

Signal Time Diagram

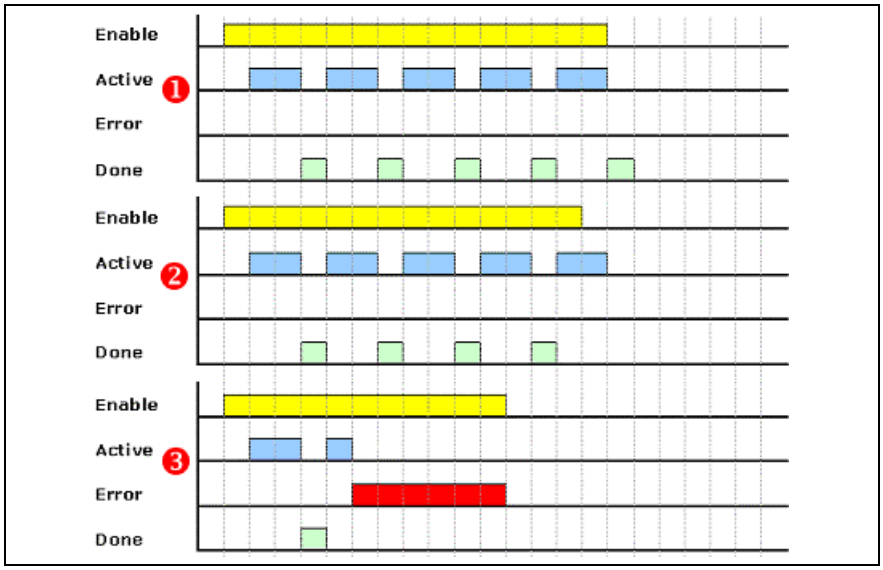


Fig. 1-32: Default signal time diagram

Error Handling This function block generates error messages only if there is a positive edge or TRUE at the "Enable" input.

Note: The MC_ReadParameter function block is working only in connection with IndraDrive drives.

MB_ReadRealParameter

Brief Description This function block returns the value of a user-specific parameter. The return value (4 bytes) is represented as a REAL number.

Library	Range
MP_PLCOpen.lib	Parameters

Fig. 1-33: Library assignment

Interface Description

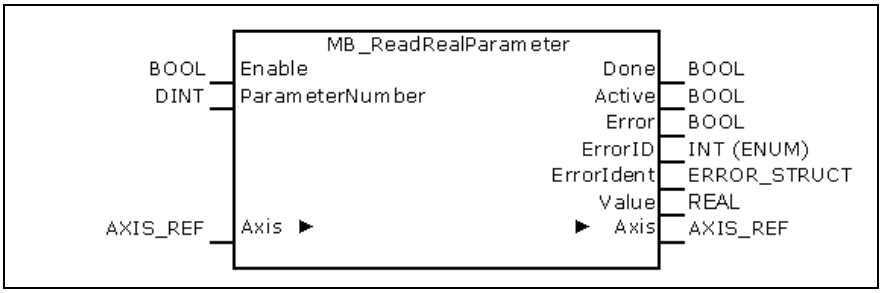


Fig. 1-34: FB MB_ReadRealParameter

	Name	Type	Comment
VAR_IN_OUT	Axis	AXIS_REF	Provides information on the actual axis.
VAR_INPUT	Enable	BOOL	Continuously reads the parameter as long as "Enable" is TRUE.
	ParameterNumber	DINT	Number of the parameter desired; refer to Structure of and Access to Parameters.
VAR_OUTPUT	Done	BOOL	A valid value is available.
	Active	BOOL	Processing of data is active after preprocessing is completed.
	Error	BOOL	Indicates that an error has occurred when the FB instance was called.
	ErrorID	ENUM ERROR_CODE	Indicates the basic error cause in the event of an error, using the enumeration values of ERROR_CODE.
	ErrorIdent	STRUCT ERROR_STRUCT	Provides detailed information on the error; this structure consists of ERROR_TABLE, Additional1 and Additional2.
	Value	REAL	4-byte values represented as REAL number

Fig. 1-35: Interface of FB MB_ReadRealParameter

Signal Time Diagram

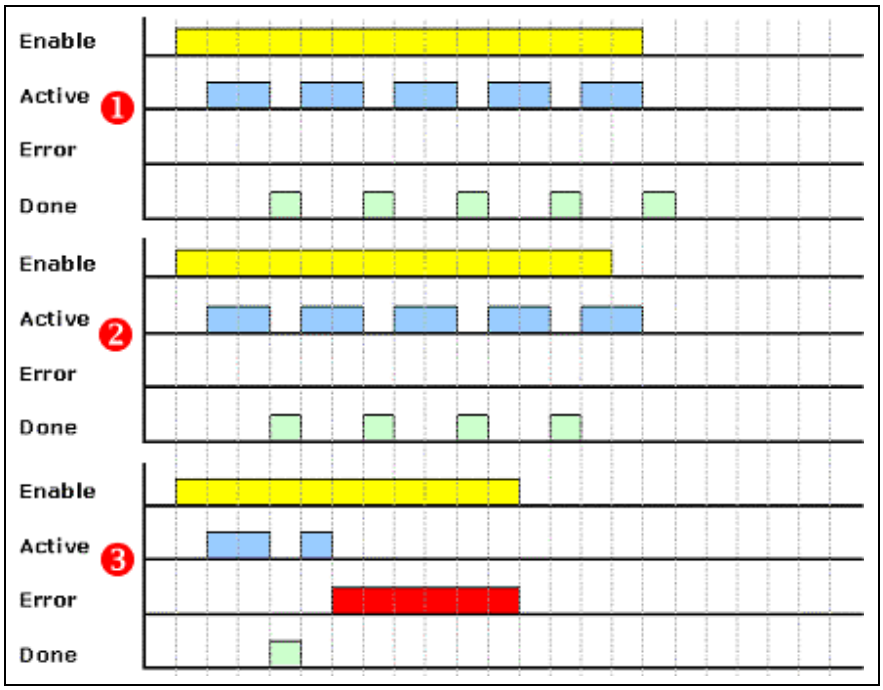


Fig. 1-36: Default signal time diagram

Error Handling

This function block generates error messages only if there is a positive edge or TRUE at the "Enable" input.

Note: The MC_ReadRealParameter function block is working only in connection with IndraDrive drives.

MB_WriteParameter

Brief Description

This function block writes a value to a user-specific parameter.

Library	Range
MP_PLCOpen.lib	Parameters

Fig. 1-37: Library assignment

Interface Description

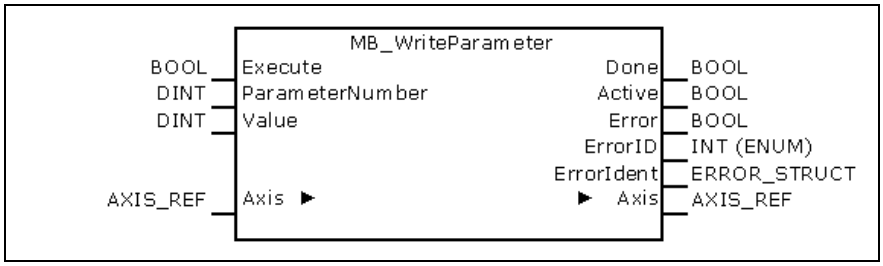


Fig. 1-38: FB MB_WriteParameter

	Name	Type	Comment
VAR_IN_OUT	Axis	AXIS_REF	Provides information on the actual axis.
VAR_INPUT	Execute	BOOL	Starts writing of the value which is present with rising edge to the parameter.
	ParameterNumber	DINT	Number of the parameter desired; refer to Structure of and Access to Parameters
	Value	DINT	Value to be written to the parameter.
VAR_OUTPUT	Done	BOOL	The value has been written successfully to the control / the drive.
	Active	BOOL	Processing of data is active after preprocessing is completed.
	Error	BOOL	Indicates that an error has occurred when the FB instance was called.
	ErrorID	ENUM ERROR_CODE	Indicates the basic error cause in the event of an error, using the enumeration values of ERROR_CODE.
	ErrorIdent	STRUCT ERROR_STRUCT	Provides detailed information on the error; this structure consists of ERROR_TABLE, Additional1 and Additional2.

Fig. 1-39: Interface of FB MB_WriteParameter

Signal Time Diagram

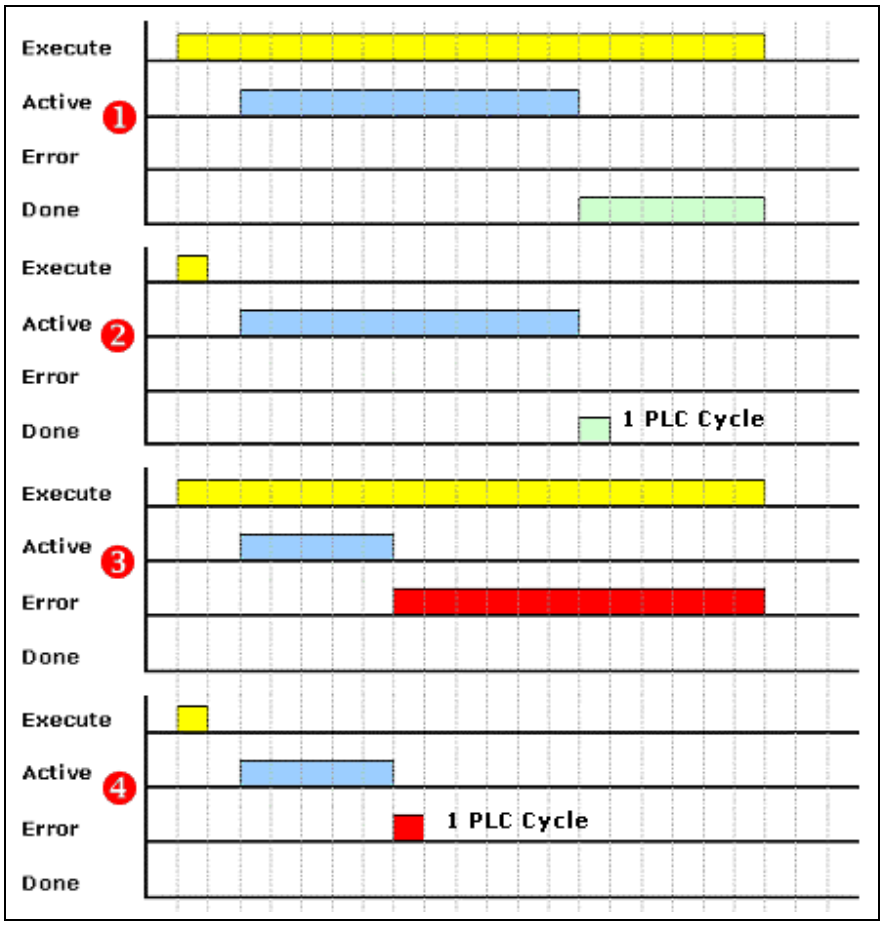


Fig. 1-40: Default signal time diagram

Error Handling

This function block generates error messages only if there is a positive edge or TRUE at the "Execute" input.

Note: The MB_WriteParameter function block is working only in connection with IndraDrive drives.

MB_WriteRealParameter

Brief Description This function block writes a value to a user-specific parameter.

Library	Range
MP_PLCOpen.lib	Parameters

Fig. 1-41: Library assignment

Interface Description

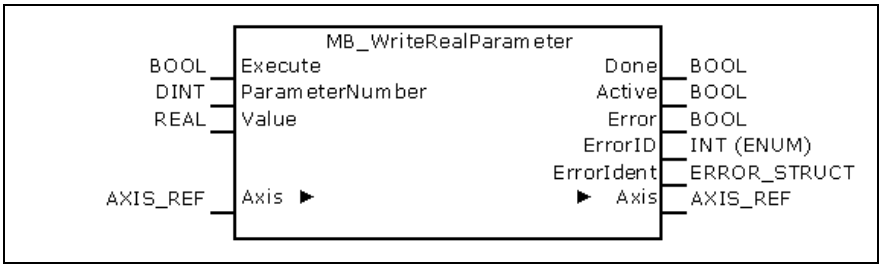


Fig. 1-42: FB MB_MB_WriteRealParameter

	Name	Type	Comment
VAR_IN_OUT	Axis	AXIS_REF	Provides information on the actual axis.
VAR_INPUT	Execute	BOOL	Starts writing of the value which is present with rising edge to the parameter.
	ParameterNumber	REAL	Number of the parameter desired; refer to Structure of and Access to Parameters
	Value	REAL	Value to be written to the parameter.
VAR_OUTPUT	Done	BOOL	The value has been written successfully to the control / the drive.
	Active	BOOL	Processing of data is active after preprocessing is completed.
	Error	BOOL	Indicates that an error has occurred when the FB instance was called.
	ErrorID	ENUM ERROR_CODE	Indicates the basic error cause in the event of an error, using the enumeration values of ERROR_CODE.
	ErrorIdent	STRUCT ERROR_STRUCT	Provides detailed information on the error; this structure consists of ERROR_TABLE, Additional1 and Additional2.

Fig. 1-43: Interface of FB MB_WriteRealParameter

Signal Time Diagram

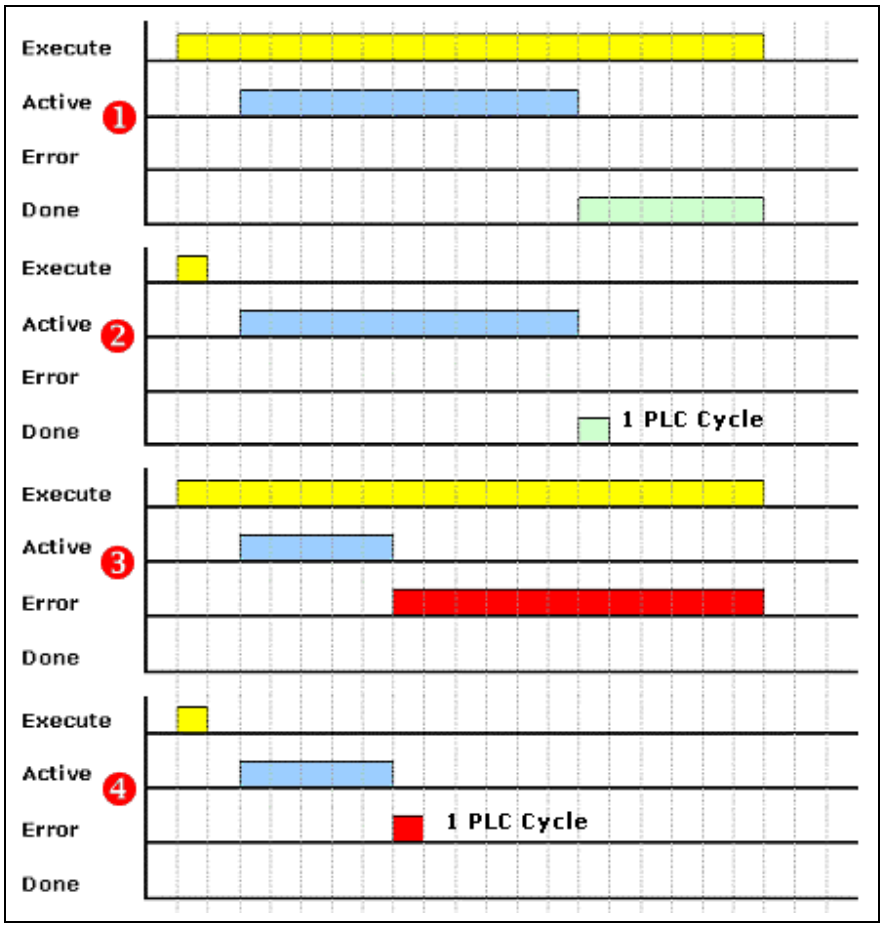


Fig. 1-44: Default signal time diagram

Error Handling

This function block generates error messages only if there is a positive edge or TRUE at the "Execute" input.

Note: The MB_WriteRealParameter function block is working only in connection with IndraDrive drives.

1.7 Function Blocks for Single-Axis Control

MC_Power

Brief Description

This function block controls the power connection (power on or off). **A separate instance** of this function block is required **for each real axis**.

Each motion, acceleration or deceleration requires that this function block is activated.

The MC_Stop function block causes a special situation. With this block, the position is kept active by the drive. That means that the MC_Power function block cannot be deactivated while MC_Stop is active.

Note: This function block must not be used in association with a virtual axis.

Library	Range
MP_PLCOpen.lib	Motion

Fig. 1-45: Library assignment

Interface

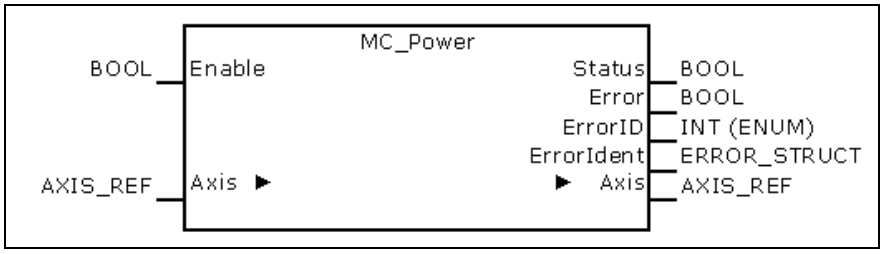


Fig. 1-46: FB MC_Power

	Name	Type	Comment
VAR_IN_OUT	Axis	AXIS_REF	Provides information on the actual axis.
VAR_INPUT	Enable	BOOL	Power is connected as long as "Enable" is TRUE.
VAR_OUTPUT	Status	BOOL	Actual state of power connection.
	Error	BOOL	Indicates that an error has occurred in the FB instance.
	ErrorID	ENUM ERROR_CODE	Indicates the basic error cause in the event of an error, using the enumeration values of ERROR_CODE.
	ErrorIdent	STRUCT ERROR_STRUCT	Provides detailed information on the error; this structure consists of ERROR_TABLE, Additional1 and Additional2.

Fig. 1-47: Interface of FB MC_Power

Error Handling

An error message is initiated if the "status" is still FALSE after a waiting time has elapsed, with "Enable" being set. This indicates that a hardware problem has occurred when power was connected.

An error message is also initiated if the attempt is made of disconnecting the power in an axis state where this is not permitted.

MC_MoveAbsolute

Brief Description

This function block copies the absolutely predefined "position" to the "target position" and moves to the latter.

Note: If this function block is used, power must have been connected beforehand via an instance of MC_Power (for each real axis).

Note: In the modulo mode, reaching of the target position is influenced by the "Moving direction" parameters (A-0-0203 / A-0-2707).

Library	Range
MP_PLCOpen.lib	Motion

Fig. 1-48: Library assignment

Interface Description

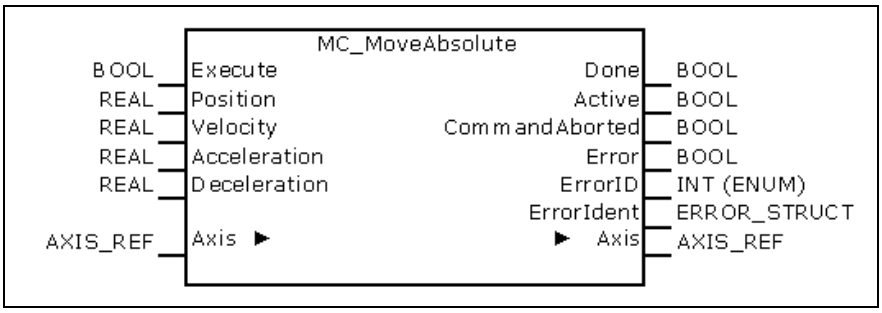


Fig. 1-49: FB MC_MoveAbsolute

	Name	Type	Comment
VAR_IN_OUT	Axis	AXIS_REF	Provides information on the actual axis.
VAR_INPUT	Execute	BOOL	Starts the motion with rising edge.
	Position	REAL	Target position for the motion (in technical unit [u]) (+/-).
	Velocity	REAL	Maximum velocity value (always +) (does not necessarily have to be reached) [u/s].
	Acceleration	REAL	Acceleration (always +) (increased energy of the motor) [u/s²]
	Deceleration	REAL	Deceleration (always +) (reduced energy of the motor) [u/s²]
VAR_OUTPUT	Done	BOOL	The desired position has been reached.
	Active	BOOL	Processing of data is active after preprocessing is completed.
	Command Aborted	BOOL	Command aborted by the following command.
	Error	BOOL	Indicates that an error has occurred in the FB instance.
	ErrorID	ENUM ERROR_CODE	Indicates the basic error cause in the event of an error, using the enumeration values of ERROR_CODE.
	ErrorIdent	STRUCT ERROR_STRUCT	Provides detailed information on the error; this structure consists of ERROR_TABLE, Additional1 and Additional2.

Fig. 1-50: Interface of FB MC_MoveAbsolute

Error Handling

This function block generates error messages only if there is a positive edge or TRUE at the "Execute" input.

Implementation Example

The following example shows how two MC_MoveAbsolute FB instances are combined:

1. The left-hand part of the time diagram shows a case where Instance2 is called after Instance1 has been processed. After Instance1 has implemented the desired position 6000 (velocity 0), its "Done" output activates Instance2 with the target position 10000.
2. The right-hand part of the time diagram shows how Instance2 is already activated while Instance1 is still running. In this case, the motion of Instance1 is stopped and aborted by the "Test" signal. Instance2 moves directly to its target position 10000, although the position 6000 has not been reached yet.

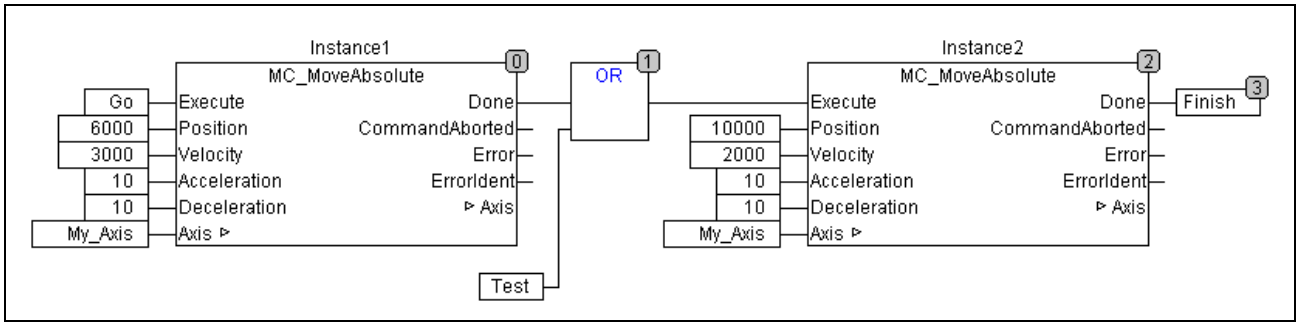


Fig. 1-51: Two instances of MC_MoveAbsolute

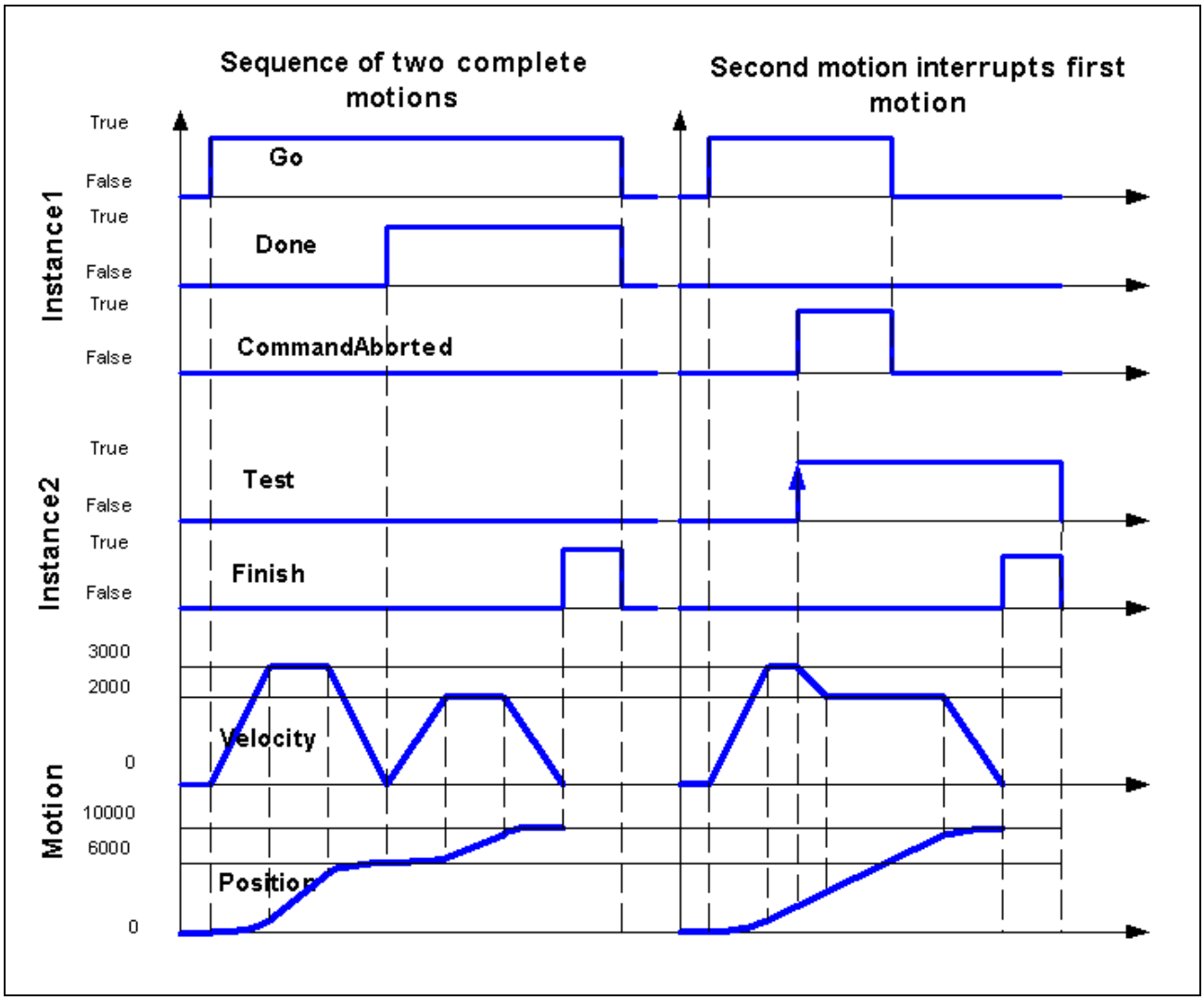


Fig. 1-52: Time diagram: two instances of MC_MoveAbsolute

MC_MoveRelative

Brief Description

This function block determines the new "target position" by adding the "distance" and the "actual position" and moves to that position.

Note: If this function block is used, power must have been connected beforehand via an instance of MC_Power (for each real axis).

Library	Range
MP_PLCOpen.lib	Motion

Fig. 1-53: Library assignment

Interface Description

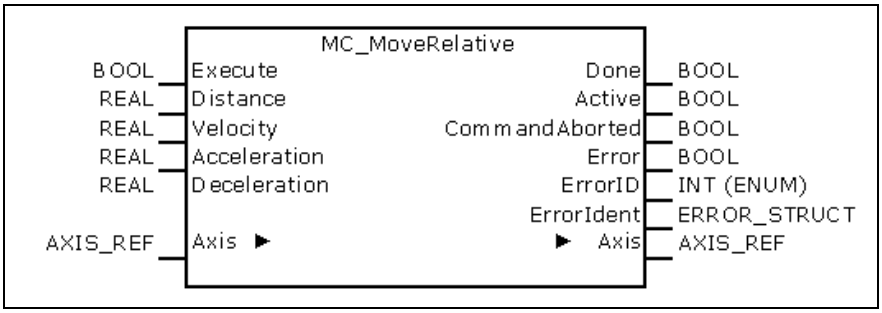


Fig. 1-54: FB MC_MoveRelative

	Name	Type	Comment
VAR_IN_OUT	Axis	AXIS_REF	Provides information on the actual axis.
VAR_INPUT	Execute	BOOL	Starts the motion with rising edge.
	Distance	REAL	Relative distance for the motion (+/-) (in technical unit [u])
	Velocity	REAL	Maximum velocity value (always +) (does not necessarily have to be reached) [u/s].
	Acceleration	REAL	Acceleration (always +) (increased energy of the motor) [u/s²]
	Deceleration	REAL	Deceleration (always +) (reduced energy of the motor) [u/s²]
VAR_OUTPUT	Done	BOOL	The desired position has been reached.
	Active	BOOL	Processing of data is active after preprocessing is completed.
	CommandAborted	BOOL	Command aborted by the following command.
	Error	BOOL	Indicates that an error has occurred in the FB instance.
	ErrorID	ENUM ERROR_CODE	Indicates the basic error cause in the event of an error, using the enumeration values of ERROR_CODE.
	ErrorIdent	STRUCT ERROR_STRUCT	Provides detailed information on the error; this structure consists of ERROR_TABLE, Additional1 and Additional2.

Fig. 1-55: Interface of FB MC_MoveRelative

Signal Time Diagram

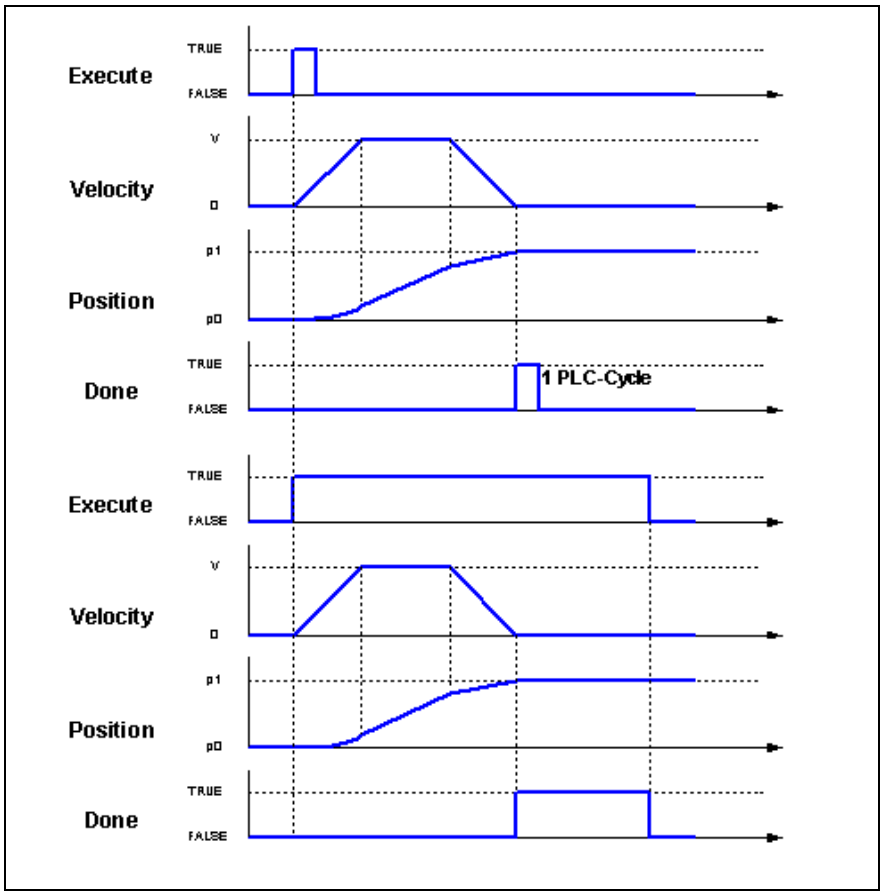


Fig. 1-56: MC_MoveRelative time diagrams

Error Handling

This function block generates error messages only if there is a positive edge or TRUE at the "Enable" input.

Implementation Example

The following example shows how two MC_MoveRelative FB instances are combined.

3. The left-hand part of the time diagram shows a case where Instance2 is called after Instance1. After Instance1 has reached the desired distance 6000 (velocity 0), its "Done" output uses Instance2 to cause a further motion by 4000 to reach the total distance of 10000.
4. The right-hand part shows how Instance2 is already activated by the "Test" signal while Instance1 is still running. In this case, the first motion is aborted during the constant velocity of Instance1. Instance1 adds its distance of 4000 to the current position (3250). The axis moves to the resulting position of 7250.

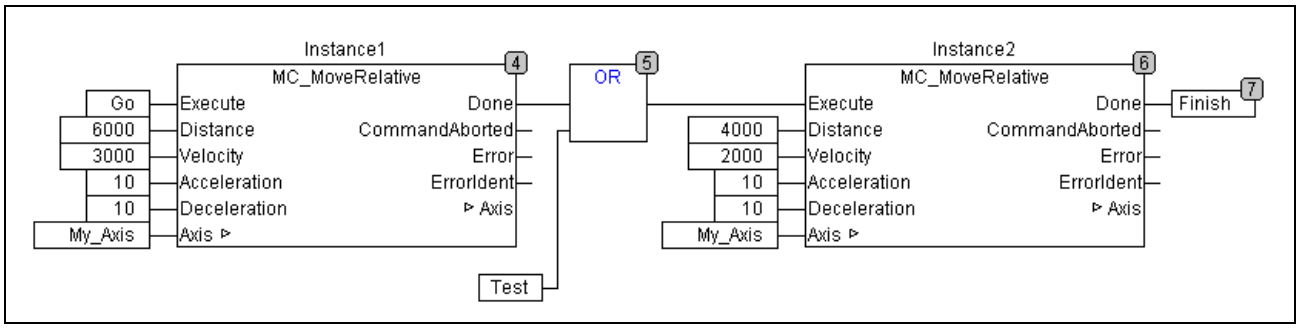


Fig. 1-57: Two instances of MC_MoveRelative

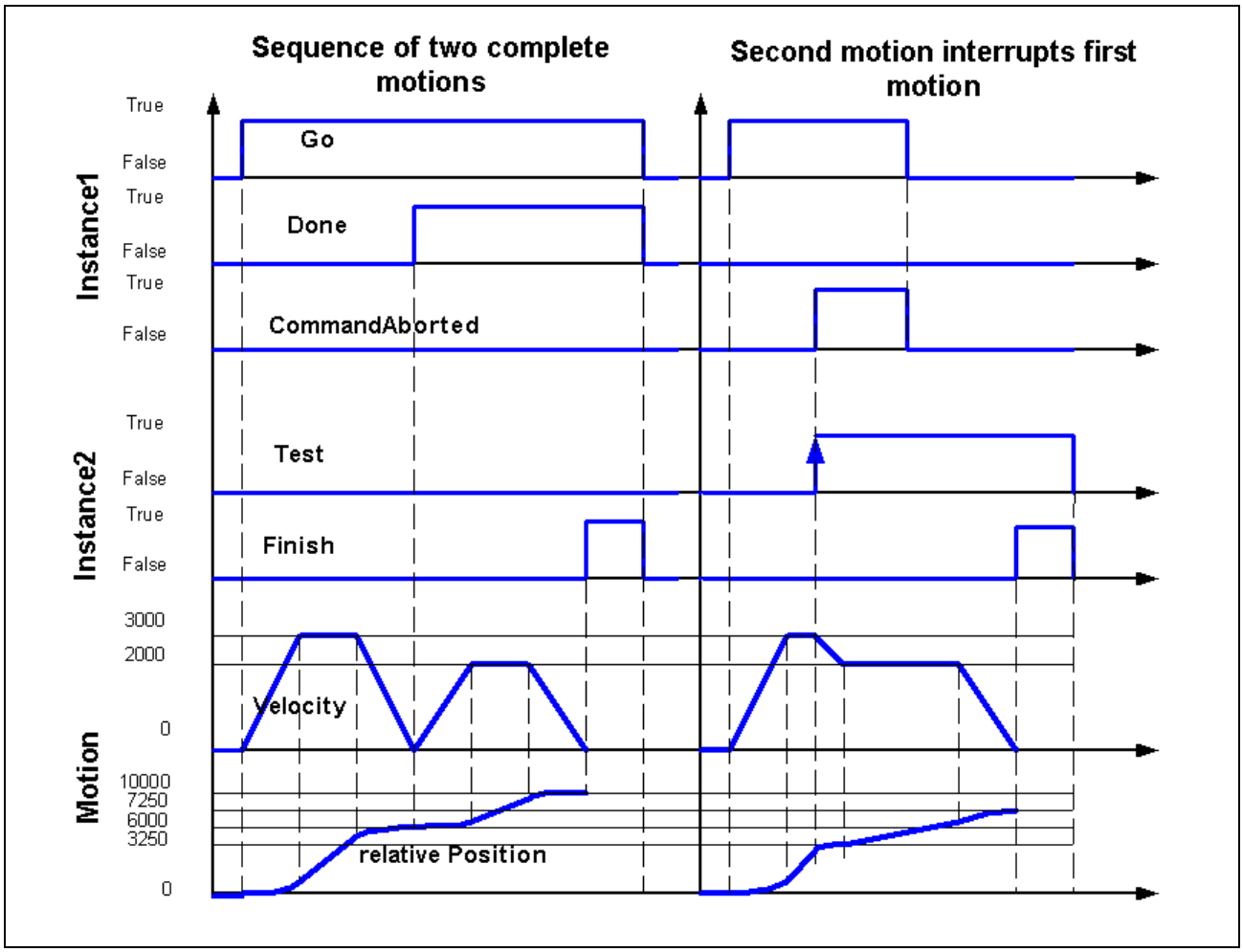


Fig. 1-58: Time diagram: two instances of MC_MoveRelative

MC_MoveAdditive

Brief Description

This function block adds the "distance" to the "target position" and moves to the latter.

Note: If this function block is used, power must have been connected beforehand via an instance of MC_Power (for each real axis).

Library	Range
MP_PLCOpen.lib	Motion

Fig. 1-59: Library assignment

Interface Description

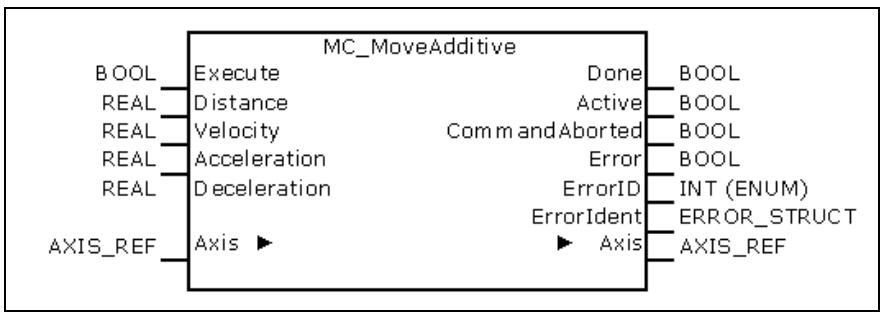


Fig. 1-60: FB MC_MoveAdditive

	Name	Type	Comment
VAR_IN_OUT	Axis	AXIS_REF	Provides information on the actual axis.
VAR_INPUT	Execute	BOOL	Starts the motion with rising edge.
	Distance	REAL	Relative distance for the motion (+/-) (in technical unit [u])
	Velocity	REAL	Maximum velocity value (always +) (does not necessarily have to be reached) [u/s].
	Acceleration	REAL	Acceleration (always +) (increased energy of the motor) [u/s²]
	Deceleration	REAL	Deceleration (always +) (reduced energy of the motor) [u/s²]
VAR_OUTPUT	Done	BOOL	The desired position has been reached.
	Active	BOOL	Processing of data is active after preprocessing is completed.
	CommandAborted	BOOL	Command aborted by the following command.
	Error	BOOL	Indicates that an error has occurred in the FB instance.
	ErrorID	ENUM ERROR_CODE	Indicates the basic error cause in the event of an error, using the enumeration values of ERROR_CODE.
	ErrorIdent	STRUCT ERROR_STRUCT	Provides detailed information on the error; this structure consists of ERROR_TABLE, Additional1 and Additional2.

Fig. 1-61: Interface of FBs MC_MoveAdditive

Error Handling

This function block generates error messages only if there is a positive edge or TRUE at the "Execute" input.

Implementation Example

The figure below shows two combined FB instances while the axis is in the "discrete motion state".

Note: Here, use is made of two different FB types:

- Instance1: MC_MoveAbsolute
- Instance2: MC_MoveAdditive

5. The left-hand part of the time diagram shows a case where the second FB is started after the first FB. The first FB reaches the desired distance of 6000 (at a velocity of 0). Then the "Done" output causes the second FB to reach the distance of 10000.
6. The right-hand part of the time diagram shows a case where the second FB is already started while the first FB is still active. In this case, the first motion is stopped and aborted by the "Test" signal at constant velocity. The second FB adds its distance of 4000 to the originally commanded position 5000 and moves the axis to the resulting position 10000, at the velocity specified at the second FB.

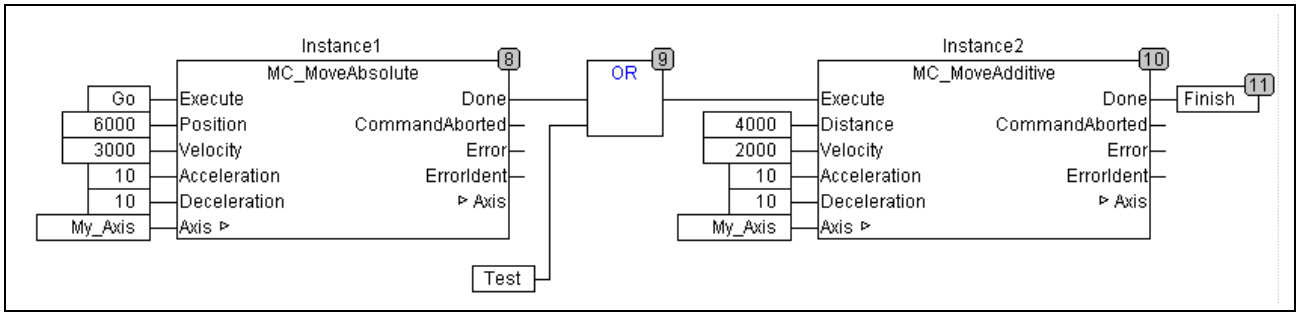


Fig. 1-62: Instances of MC_MoveAbsolute and MC_MoveAdditive

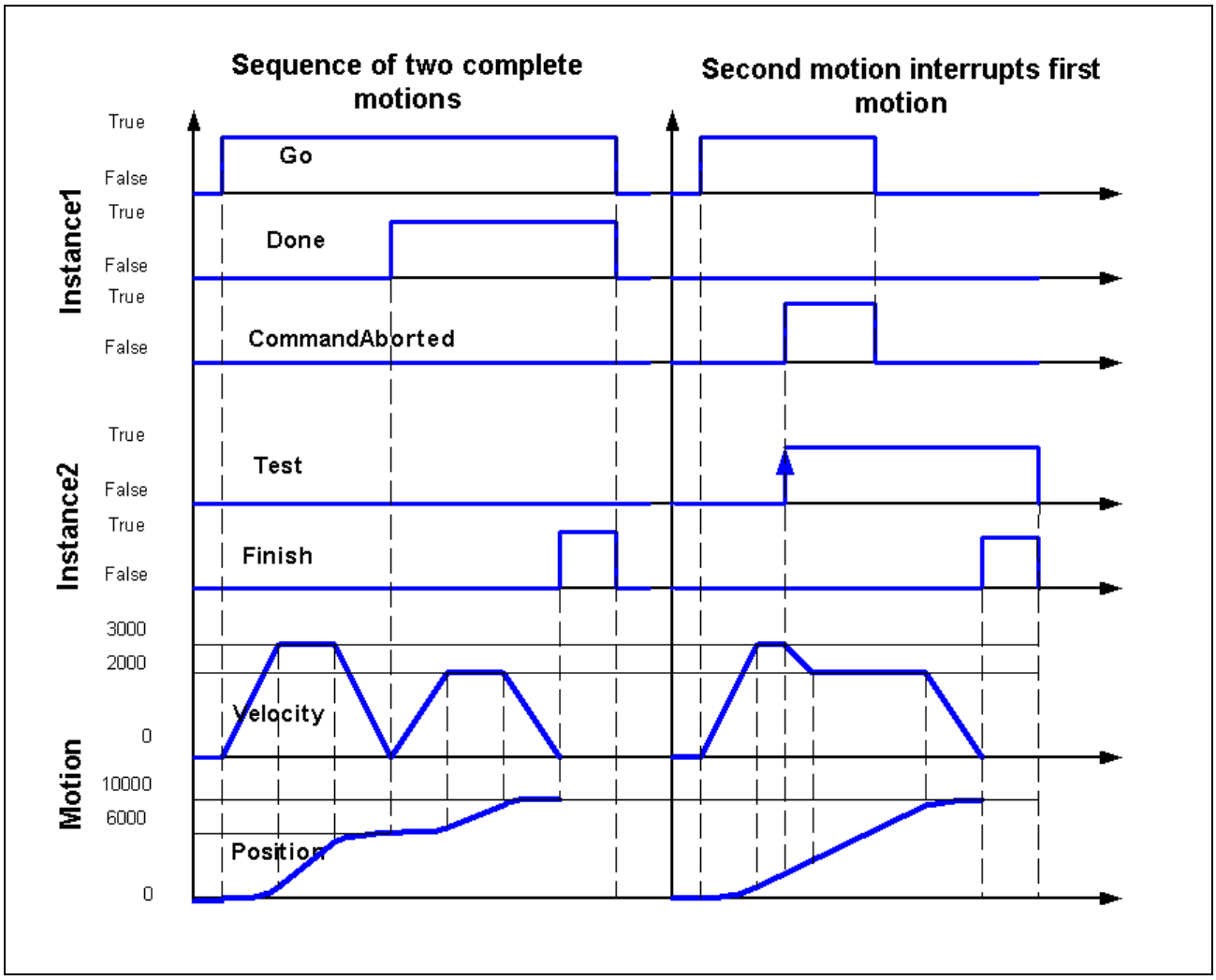


Fig. 1-63: Time diagram: Instances of MC_MoveAbsolute and MC_MoveAdditive

MC_MoveVelocity

Brief Description

This function block controls a "never-ending" motion at the particularly defined velocity. To terminate the motion, the FB instance must be interrupted by another FB instance which issues a new command.

Note: If this function block is used, power must have been connected beforehand via an instance of MC_Power (for each real axis).

The "InVelocity" signal is reset if the FB instance is interrupted by another FB instance (CommandAborted) or if the edge of its "Execute" input is falling.

Note: If "Execute" is deactivated, the drive continues to rotate at the same velocity.

Library	Range
MP_PLCOpen.lib	Motion

Fig. 1-64: Library assignment

Interface Description

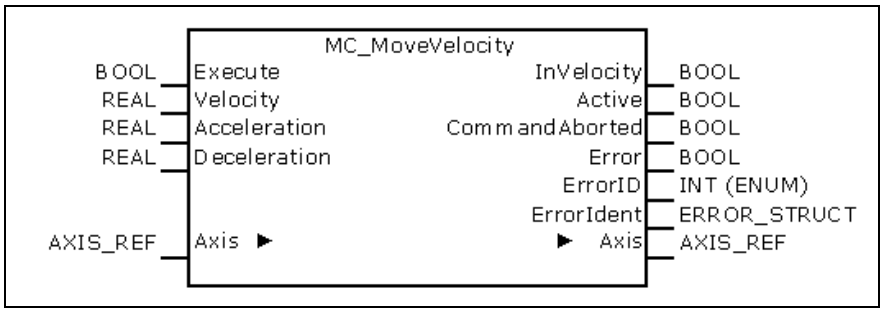


Fig. 1-65: FB MC_MoveVelocity

	Name	Type	Comment
VAR_IN_OUT	Axis	AXIS_REF	Provides information on the actual axis.
VAR_INPUT	Execute	BOOL	Starts the motion with rising edge.
	Velocity	REAL	Maximum velocity value (does not necessarily have to be reached) [u/s].
	Acceleration	REAL	Acceleration (always +) (increased energy of the motor) [u/s²]
	Deceleration	REAL	Deceleration (always +) (reduced energy of the motor) [u/s²]
VAR_OUTPUT	InVelocity	BOOL	Velocity reached (for the first time)
	Active	BOOL	Processing of data is active after preprocessing is completed.
	CommandAborted	BOOL	Command aborted by the following command.
	Error	BOOL	Indicates that an error has occurred in the FB instance.
	ErrorID	ENUM ERROR_CODE	Indicates the basic error cause in the event of an error, using the enumeration values of ERROR_CODE.
	ErrorIdent	STRUCT ERROR_STRUCT	Provides detailed information on the error; this structure consists of ERROR_TABLE, Additional1 and Additional2.

Fig. 1-66: Interface of FB MC_MoveVelocity

Signal Time Diagram

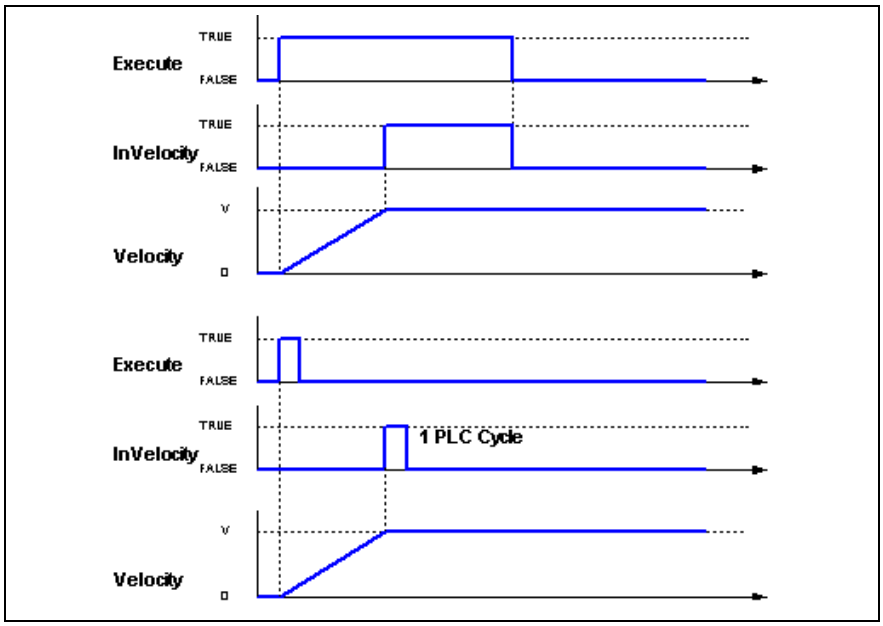


Fig. 1-67: Time diagram of MC_MoveVelocity (without aborted command)

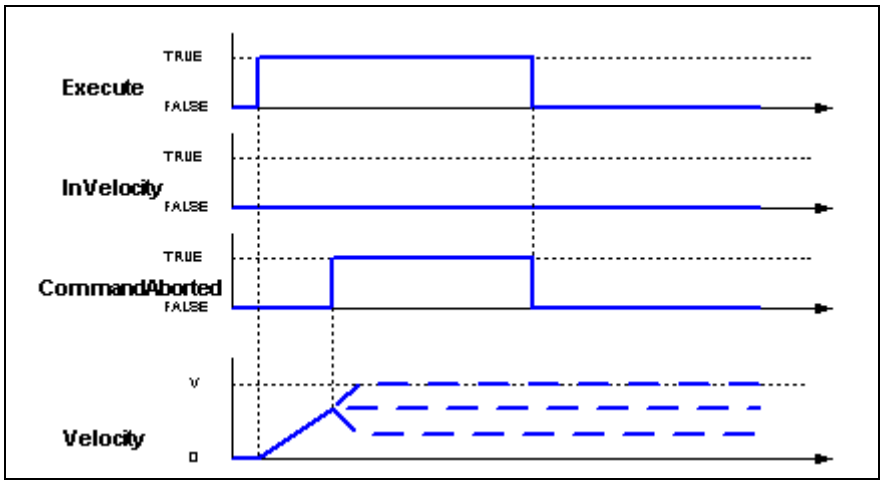


Fig. 1-68: Time diagram of MC_MoveVelocity (with aborted command)

Error Handling

This function block generates error messages only if there is a positive edge or TRUE at the "Enable" input.

Implementation Example

The following example shows how two MC_MoveVelocity FB instances are combined.

7. The left-hand part of the time diagram shows a case where the second instance is called after the job of the first instance is completed. Once the first function block has reached the required velocity of 3000, the "Instance1.InVelocity" AND output causes the "Next" signal; as a result, the second function block continues the motion at a velocity of 2000.
8. The right-hand part of the time diagram shows a case where the first FB has not reached the desired velocity yet, before the second FB is started. Thereafter, the following is shown: The first motion is restarted by "Go" at the "Instance1.Execute" input. While it is still accelerating to achieve the velocity of 3000, the first FB is interrupted by the "Test" signal which starts the second FB. Now, the second FB is running and reduces the velocity to 2000.

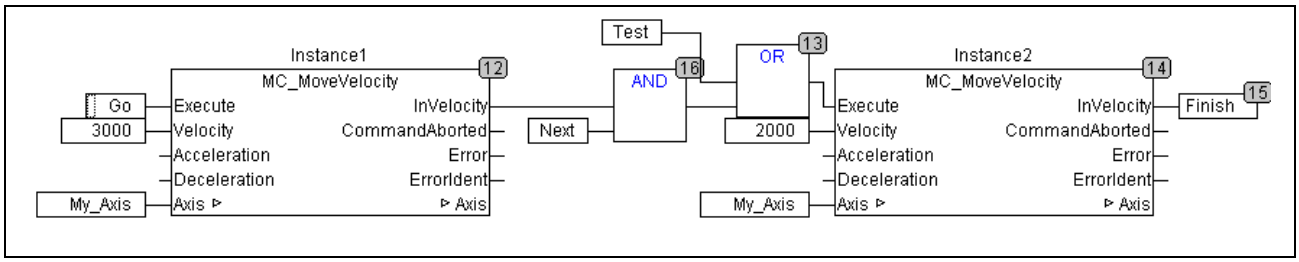


Fig. 1-69: Two instances of MC_MoveVelocity

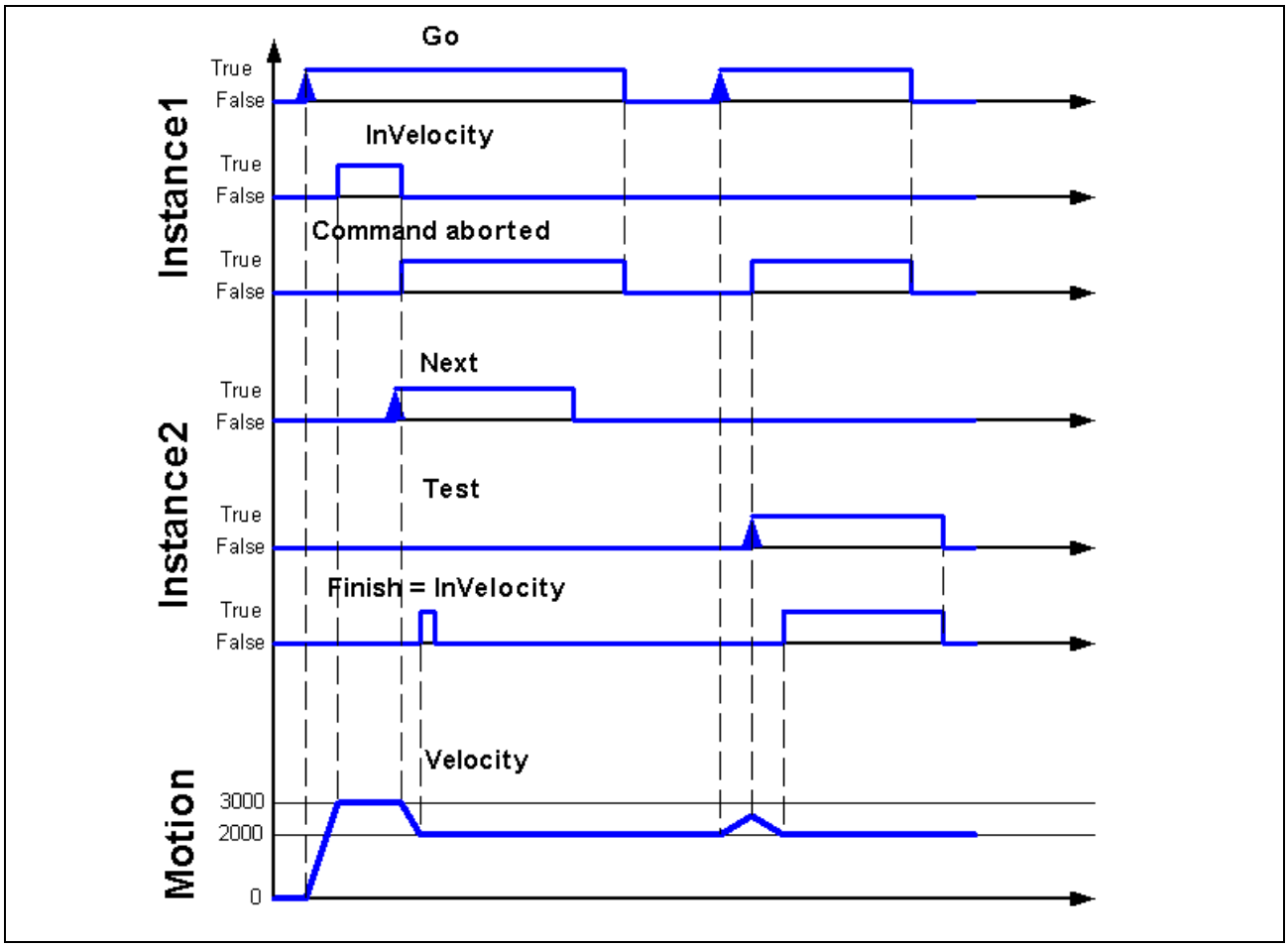


Fig. 1-70: Time diagram: two instances of MC_MoveVelocity

MB_Home

Brief Description

This function block controls **axes with incremental encoder** when executing the "search home" sequence. The details of the sequence must be set by means of axis parameters.

Library	Range
MP_PLCOpen.lib	Motion

Fig. 1-71: Library assignment

Interface Description

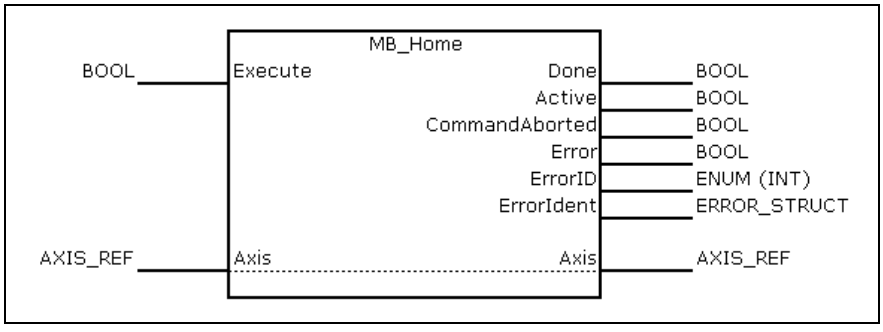


Fig. 1-72: FB MB_Home

	Name	Type	Comment
VAR_IN_OUT	Axis	AXIS_REF	Provides information on the actual axis.
VAR_INPUT	Execute	BOOL	Starts the motion with rising edge.
VAR_OUTPUT	Done	BOOL	<<Homing>> completed; velocity 0
	Active	BOOL	Processing of data is active after preprocessing is completed.
	CommandAborted	BOOL	Command aborted by the following command.
	Error	BOOL	Indicates that an error has occurred in the FB instance.
	ErrorID	ENUM ERROR_CODE	Indicates the basic error cause in the event of an error, using the enumeration values of ERROR_CODE.
	ErrorIdent	STRUCT ERROR_STRUCT	Provides detailed information on the error; this structure consists of ERROR_TABLE, Additional1 and Additional2.

Fig. 1-73: Interface of FB MB_Home

Signal Time Diagram

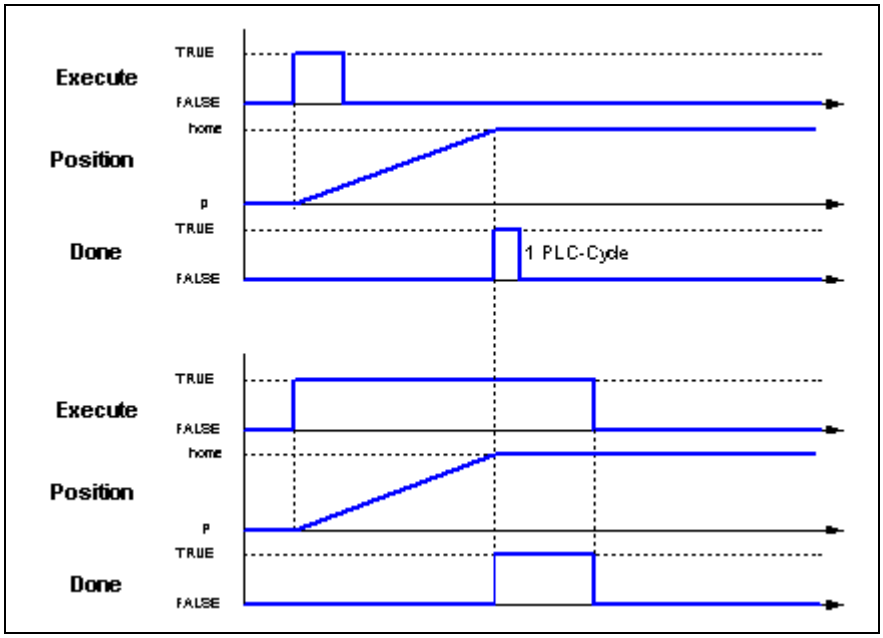


Fig. 1-74: Time diagram of MB_Home

Error Handling

This function block generates error messages only if there is a positive edge or TRUE at the "Execute" input.

MC_Stop

Brief Description

This function block implements a controlled motion stop and transfers the axis to the "Stopping" state. As a result, any function block processing is stopped. When standstill is reached, the "Done" output is set.

The position is kept active.

Note: The axis remains in the "Stopping" state as long as "Execute" is TRUE. Another command cannot be started before the axis has entered the "Stopping" state.

In the "Stopping" state, it is not possible to disconnect the power (MC_Power) of the axis. If the "Enable" signal of the MC_Power instance is deactivated, its "Status" nevertheless remains activated until the "Execute" signal of the MC_Stop instance is also deactivated.

Library	Range
MP_PLCOpen.lib	Motion

Fig. 1-75: Library assignment

Interface Description

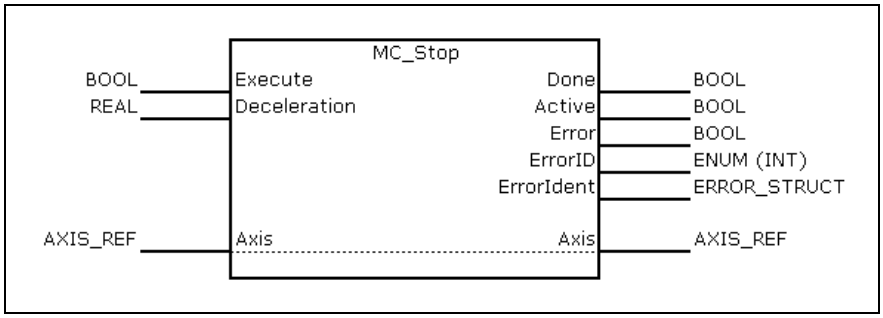


Fig. 1-76: FB MC_Stop

	Name	Type	Comment
VAR_IN_OUT	Axis	AXIS_REF	Provides information on the actual axis.
VAR_INPUT	Execute	BOOL	Starts the action with rising edge. The function block cannot be interrupted before the "Stopping" state is reached.
	Deceleration	REAL	Deceleration (always +) (reduced energy of the motor) [u/s²]
VAR_OUTPUT	Done	BOOL	Standstill reached
	Active	BOOL	Processing of data is active after preprocessing is completed.
	Error	BOOL	Indicates that an error has occurred in the FB instance.
	ErrorID	ENUM ERROR_CODE	Indicates the basic error cause in the event of an error, using the enumeration values of ERROR_CODE.
	ErrorIdent	STRUCT ERROR_STRUCT	Provides detailed information on the error; this structure consists of ERROR_TABLE, Additional1 and Additional2.

Fig. 1-77: Interface of FB MC_Stop

Signal Time Diagram

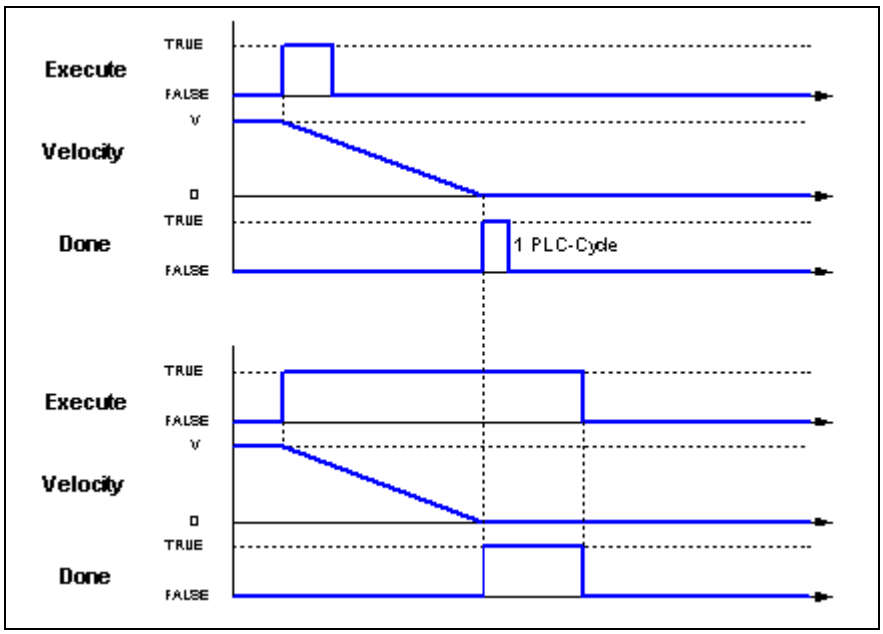


Fig. 1-78: Time diagram of MC_Stop

Error Handling

This function block generates error messages only if there is a positive edge or TRUE at the "Execute" input.

MB_Stop

Brief Description

This function block implements a controlled motion stop and transfers the axis to the "Stopping" state. As a result, any function block processing is stopped. When standstill is reached, the "Done" output is set.

The position is kept active.

Note: The axis remains in the "Stopping" state as long as "ExecuteLock" is TRUE.

Contrary to "MC_Stop", the function block can be interrupted by another block as soon as ExecuteLock is FALSE.

In the "Stopping" state, it is not possible to disconnect the power (MC_Power) of the axis. If the "Enable" signal of the MC_Power instance is deactivated, its "Status" nevertheless remains activated until the "Execute" signal of the MB_Stop instance is also deactivated.

Library	Range
MP_PLCOpen.lib	Motion

Fig. 1-79: Library assignment

Interface Description

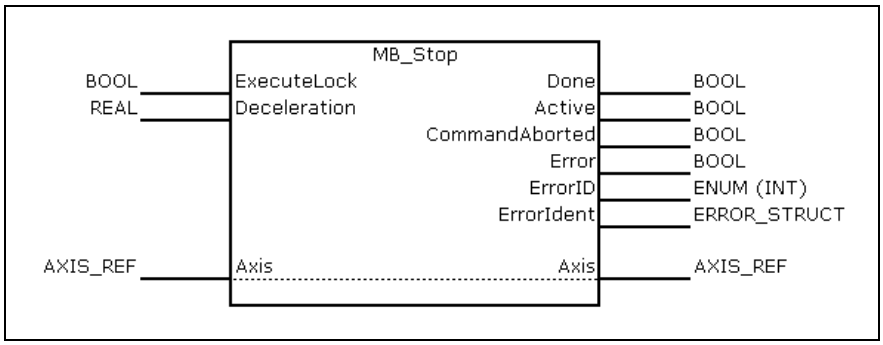


Fig. 1-80: FB MB_Stop

	Name	Type	Comment
VAR_IN_OUT	Axis	AXIS_REF	Provides information on the actual axis.
VAR_INPUT	ExecuteLock	BOOL	Starts the action with rising edge. Cannot be interrupted by other function blocks, as long as TRUE is applied to this input.
	Deceleration	REAL	Deceleration (always +) (reduced energy of the motor) [u/s²]
VAR_OUTPUT	Done	BOOL	Standstill reached
	Active	BOOL	Processing of data is active after preprocessing is completed.
	CommandAborted	BOOL	Command aborted by the following command.
	Error	BOOL	Indicates that an error has occurred in the FB instance.
	ErrorID	ENUM ERROR_CODE	Indicates the basic error cause in the event of an error, using the enumeration values of ERROR_CODE.
	ErrorIdent	STRUCT ERROR_STRUCT	Provides detailed information on the error; this structure consists of ERROR_TABLE, Additional1 and Additional2.

Fig. 1-81: Interface of FB MB_Stop

Signal Time Diagram

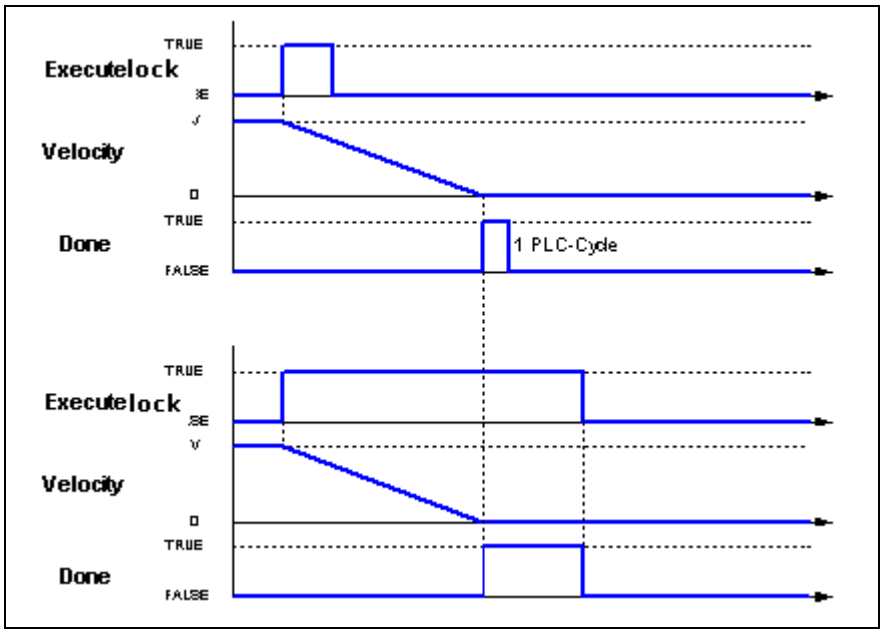


Fig. 1-82: Time diagram of MC_Stop

Error Handling

This function block generates error messages only if there is a positive edge or TRUE at the "ExecuteLock" input.

MC_Reset

Brief Description

This function block implements the transition from the "ErrorStop" state to the "StandStill" state by resetting all internal axis errors.

Note: The outputs of the function block instances having indicated errors are not affected.

If there is no error, triggering of MC_Reset will not have any effect. However, an error-clear command "C5" is always sent to the drive.

Library	Range
MP_PLCOpen.lib	Motion

Fig. 1-83: Library assignment

Interface Description

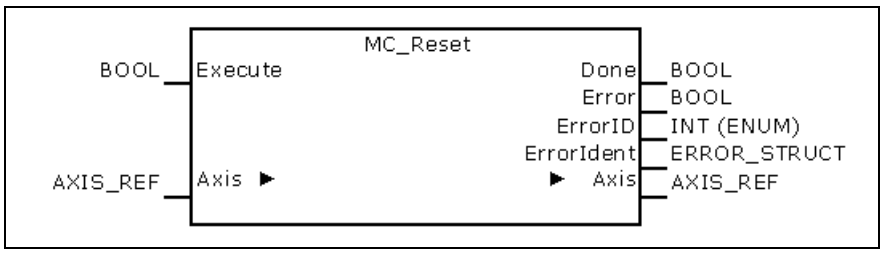


Fig. 1-84: FB MC_Reset

	Name	Type	Comment
VAR_IN_OUT	Axis	AXIS_REF	Provides information on the actual axis.
VAR_INPUT	Execute	BOOL	Starts the action with rising edge.
VAR_OUTPUT	Done	BOOL	Standstill reached
	Error	BOOL	Indicates that an error has occurred in the FB instance.
	ErrorID	ENUM ERROR_CODE	Indicates the basic error cause in the event of an error, using the enumeration values of ERROR_CODE.
	ErrorIdent	STRUCT ERROR_STRUCT	Provides detailed information on the error; this structure consists of ERROR_TABLE, Additional1 and Additional2.

Fig. 1-85: Interface of FB MC_Reset

Signal Time Diagram

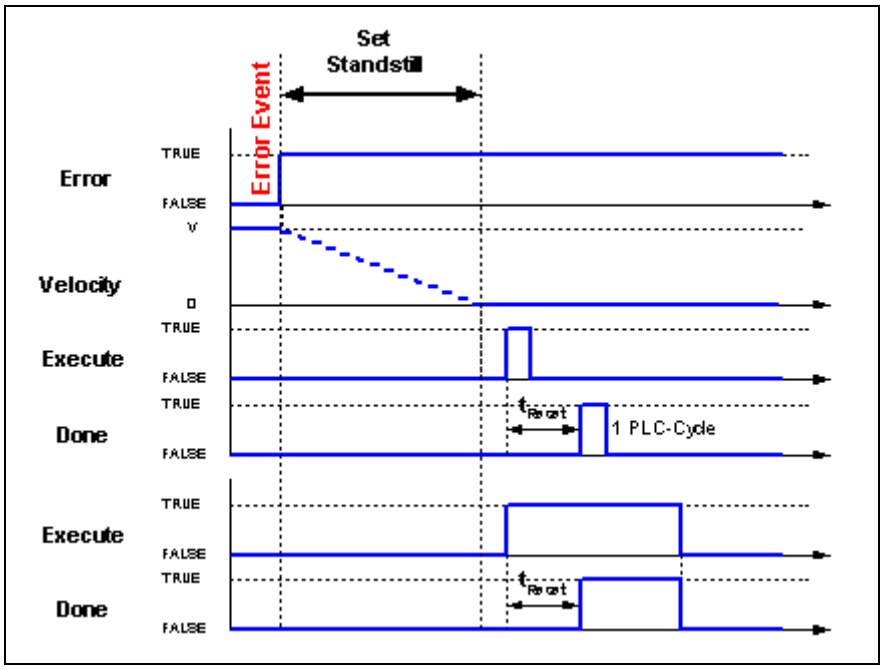


Fig. 1-86: Time diagram of MC_Reset

Error Handling

This function block generates error messages only if there is a positive edge or TRUE at the "Execute" input.

1.8 Function Blocks for Multi-Axis Control

Note: Functions of the multi-axis control are not enabled for field bus drives! Single-axis function blocks are the only ones functioning for field bus drives!

The following parameters and/or function blocks are provided for multi-axis control:

- MC_START_MODE
- MC_SYNC_MODE
- MC_CAM_ID
- MC_CamIn
- MC_CamOut
- MC_GearIn
- MC_GearOut
- MB_GearInPos
- ML_PhasingSlave

2 Communicating Field Bus Axes to the PLC

This chapter describes how to operate field bus axes with the MotionLogic modules.

2.1 Basic PLC Settings

To ensure proper communication of the drives with the PLC, system and drive settings must be made.

Profibus Interface Settings

The Profibus interface must be set to the default parameter "Byteorder Wordmodules Motorola". Select this setting in the IndraWorks project navigator, Profibus/M (master), "Parameters" tab.

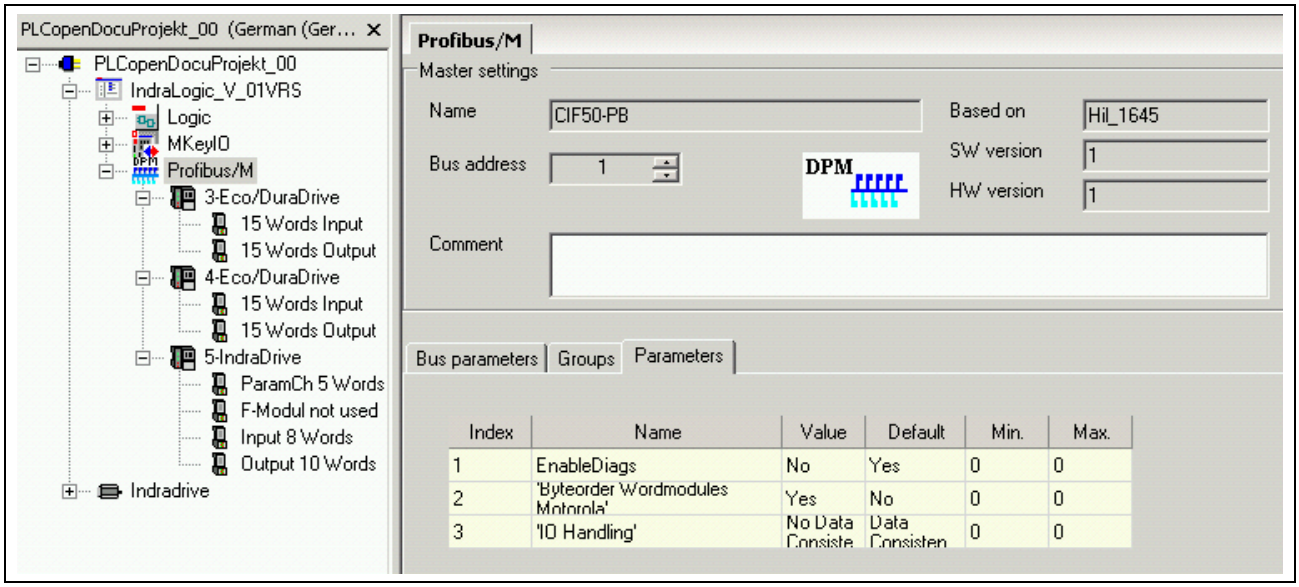


Fig. 2-1: Profibus interface settings

Target Settings

The "Replace constants" check box must be activated in the target settings. If this check box is unchecked, an error message stating that there is no variable declaration will be repeatedly emitted during translation. Make this setting in the IndraWorks project navigator, "Logic" properties, "Further settings" tab.

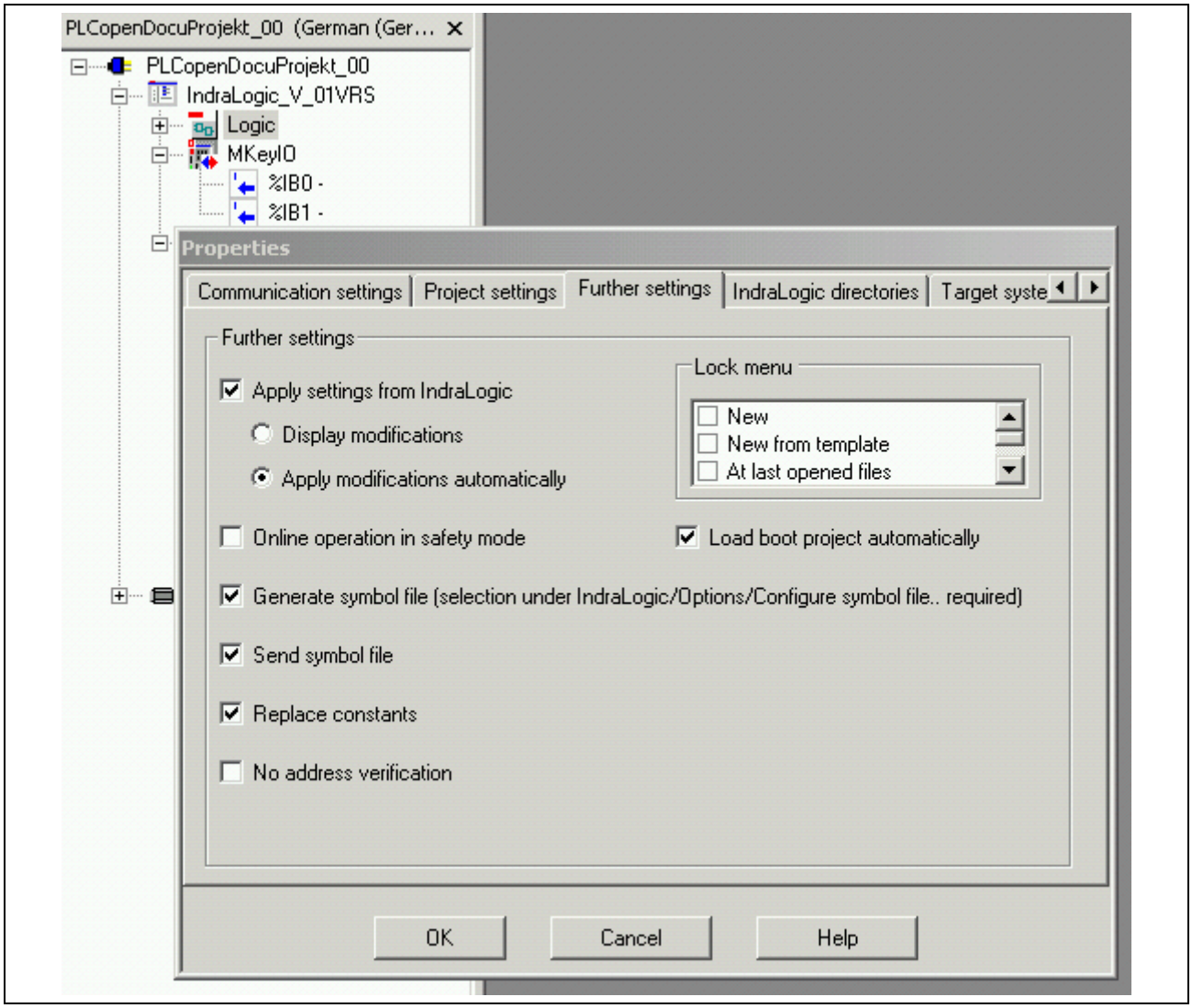


Fig. 2-2: "Replace constants" check box in the target settings

Communicating Field Bus Axes to the PLC

Field bus axes are normal field bus users of Profibus. GSD files for IndraDrive and EcoDrive, Dura and EcoDrive CS are included in the IndraWorks installation:

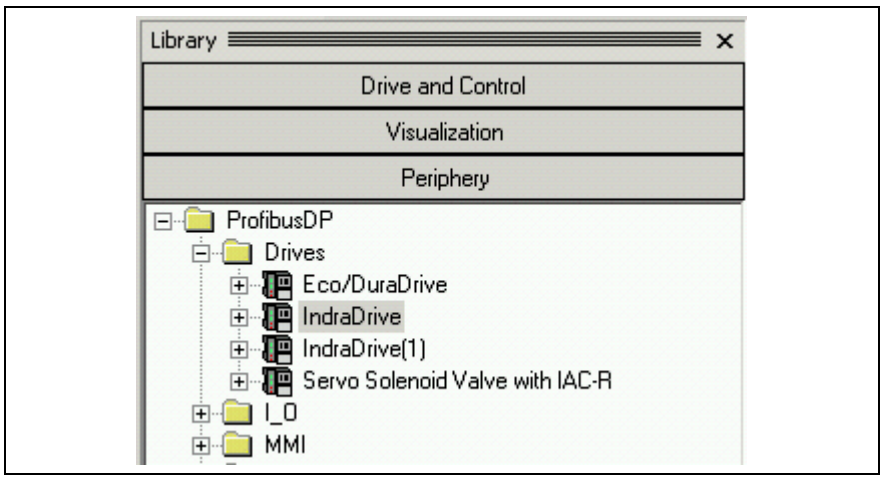


Fig. 2-3: GSD files of the drives in IndraWorks

Move the desired axis from the IndraWorks library to the PLC project by drag-and-drop and place it below the Profibus master interface in the project navigator:

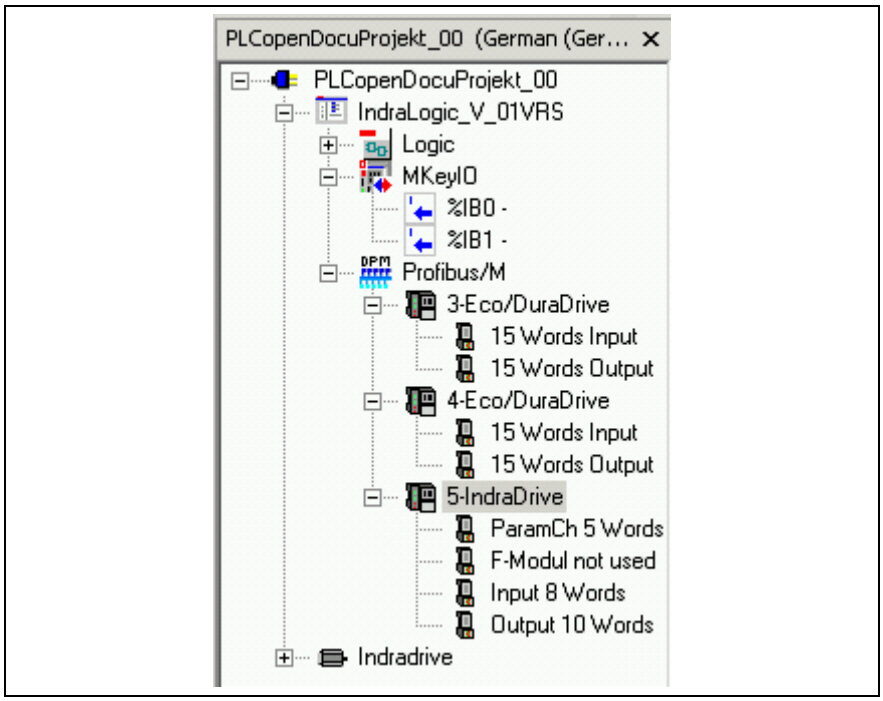


Fig. 2-4: Entering IndraDrive as Profibus user

Note: As can be seen from Fig. 2-4, the modules of IndraDrive and EcoDrive or EcoDrive CS must be entered.

- IndraDrive requires the following mandatory order: ParameterChannel with 5 Words, F-Module **(even if the module is not used!)**, Input 8 Words, and Output 10 Words.
- EcoDrive, EcoDrive CS and DuraDrive require the following order: 15 Words Input, 15 Words Output.

In their original state, the drives fail to have the settings required for proper communication with the PLC via the Profibus. To achieve this, the settings described in the chapter below must be made in the drives.

2.2 Drive Settings Required for Communication

Activation of the field bus drives with PLCopen function blocks requires some basic settings referring to the communication of the PLCopen modules with the field bus drives.

Note: The settings described in this section of the document only refer to the communication of the drive with the PLC. They do **not** describe a general start-up of the drive.

The settings required can be made with the IndraWorks Engineering Desktop for IndraDrive and EcoDrive CS. Likewise, the settings described below can be made with the "DriveTop" tool for the normal EcoDrive and for all drive types. The dialogs described here differ from the DriveTop dialogs to a minor degree only.

Note: The communication required for parameterization is **not** established by means of the Profibus interface of the PLC. Using the PC where the IndraWorks Engineering Desktop has been started, the communication required for parameterization can be established either serially via the RS232 interface or via a separate Profibus master provided for this purpose.

The sections below describe parameterization via serial communication.

Establishing the Communication for Parameterization

Establish the communication with the drive that is required for parameterization using the "Scan for devices" function on the IndraWorks Engineering Desktop.

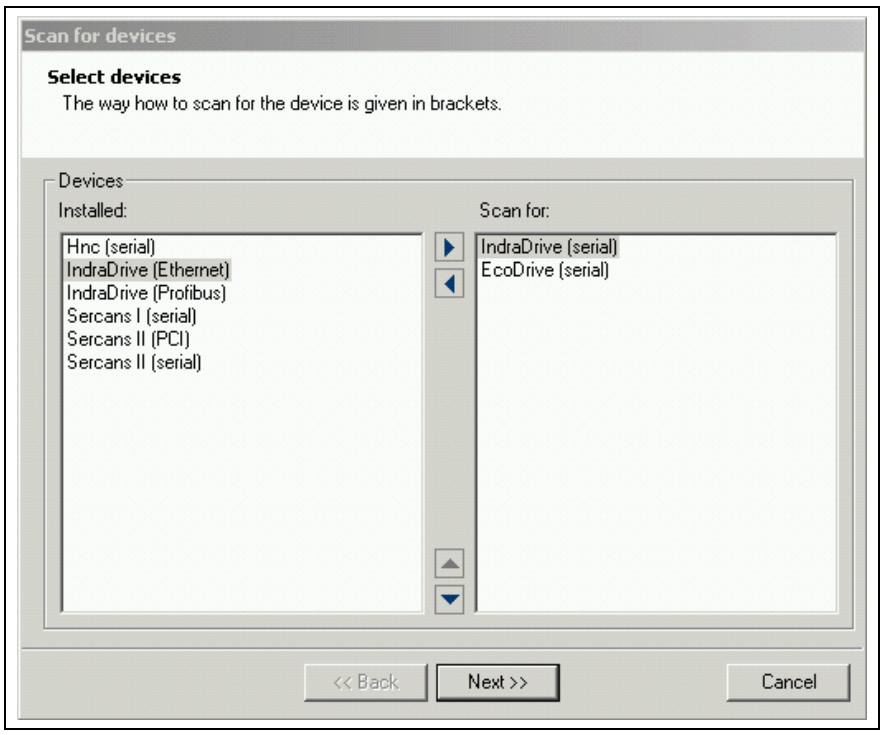


Fig. 2-5: Establishing the communication with the drive required for parameterization in IndraWorks

Once you have established the connection to the drive, you can make the required settings.

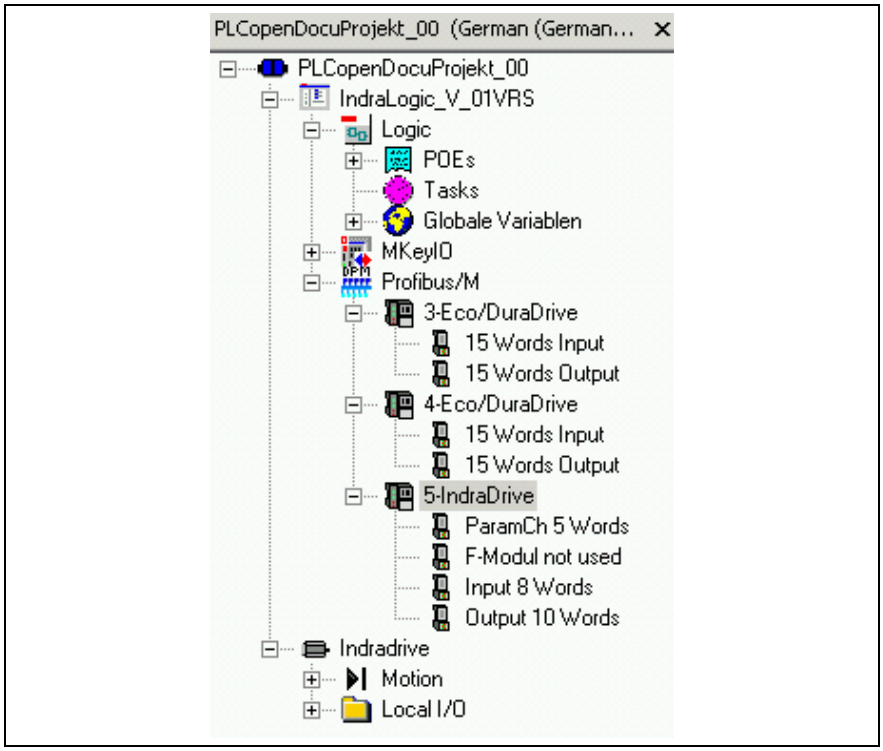


Fig. 2-6: Example of an IW dialog window for parameterization of an IndraDrive

IndraDrive Communication Settings

The sections below describe the settings an IndraDrive requires for communication.

Profile Settings

Select the "Free configurable mode" profile type. Also use this menu to specify the slave address in relation to the Profibus.

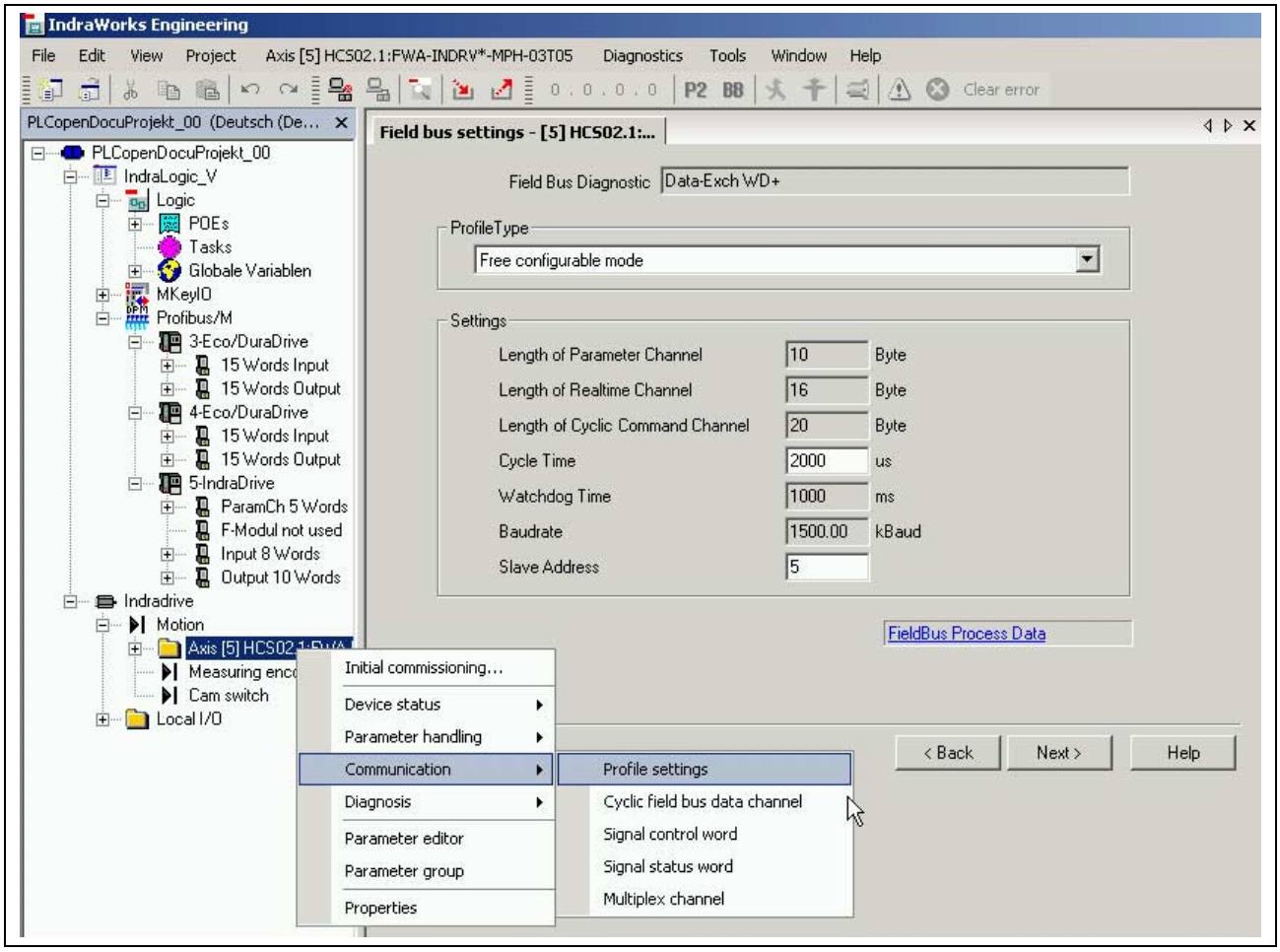


Fig. 2-7: Example of an IW dialog window for profile settings of an IndraDrive

Cyclic Field Bus Data Channel

Use this item to set the parameters of the cyclic data exchange with the PLC:

- Configuration list of the cyclic actual-value data channel (P-0-4078 Field bus status word)
  - P-0-4078 Field bus: status word
  - S-0-0051 Position feedback 1 value
  - S-0-0040 Velocity feedback value
  - S-0-0390 Diagnostic message number
  - S-0-0144 Signal status word

- Configuration list of the cyclic command-value data channel (P-0-4077 Field bus control word)
  - P-0-4077 Field bus: control word
  - S-0-0282 Positioning command value
  - S-0-0259 Positioning velocity
  - S-0-0260 Positioning acceleration
  - S-0-0359 Positioning deceleration
  - S-0-0145 Signal control word

Signal Control Word

Only the first bit is used in the signal control word:

- S-0-0346 Positioning control word
  - Bit 4

To enter these values, the parameters below must be assigned as follows:

- S-0-0027, Configuration list signal control word

  S-0-0346 Positioning control word assigned to element 0
- S-0-0329, Assign list signal control word

  4 (bit 4) assigned to element 0

The other bit positions are available for use outside of the PLCopen function blocks.

Note: Firmware MPH02V18 and higher must be installed to ensure that IndraDrive moves MC_MoveRelative properly. Otherwise, S-0-0346 cannot be configured in the signal control word.

Signal Status Word

Only the first bit is used in the signal status word:

- S-0-0437 Positioning status
  - Bit 3

To enter these values, the parameters below must be assigned as follows:

- S-0-0026, Configuration list signal status word

  S-0-0437 Positioning status assigned to element 0
- S-0-0328, Assign list signal status word

  3 (bit 3) assigned to element 0

Parameter Channel Length

Set the length of the parameter channel to 10 bytes by directly entering the parameter P-0-4083 using the parameter editor.

Mode Selection

Assign "Drive-controlled positioning" as the primary mode and "Velocity control" as the first secondary mode to the drive. The second and third modes are of no relevance, since they are not selected by the PLCopen function blocks.

Note: The MB_Home and MP_Home function blocks execute homing only if the drive has been parameterized for homing.

EcoDrive (CS) Communication Settings

The sections below describe the settings an EcoDrive or EcoDrive CS requires for communication. The necessary parameter settings are the same for both EcoDrive and EcoDrive CS.

Note: At present, IndraWorks supports the startup of IndraDrive and EcoDrive CS only. If you intend to start up a normal EcoDrive, we recommend to enter the settings with DriveTop.

Profile Settings

Select the "Free configurable operating mode" profile type and set the length of the parameter channel to 12 bytes. Set the slave address directly at the drive, using the rotary address switches.

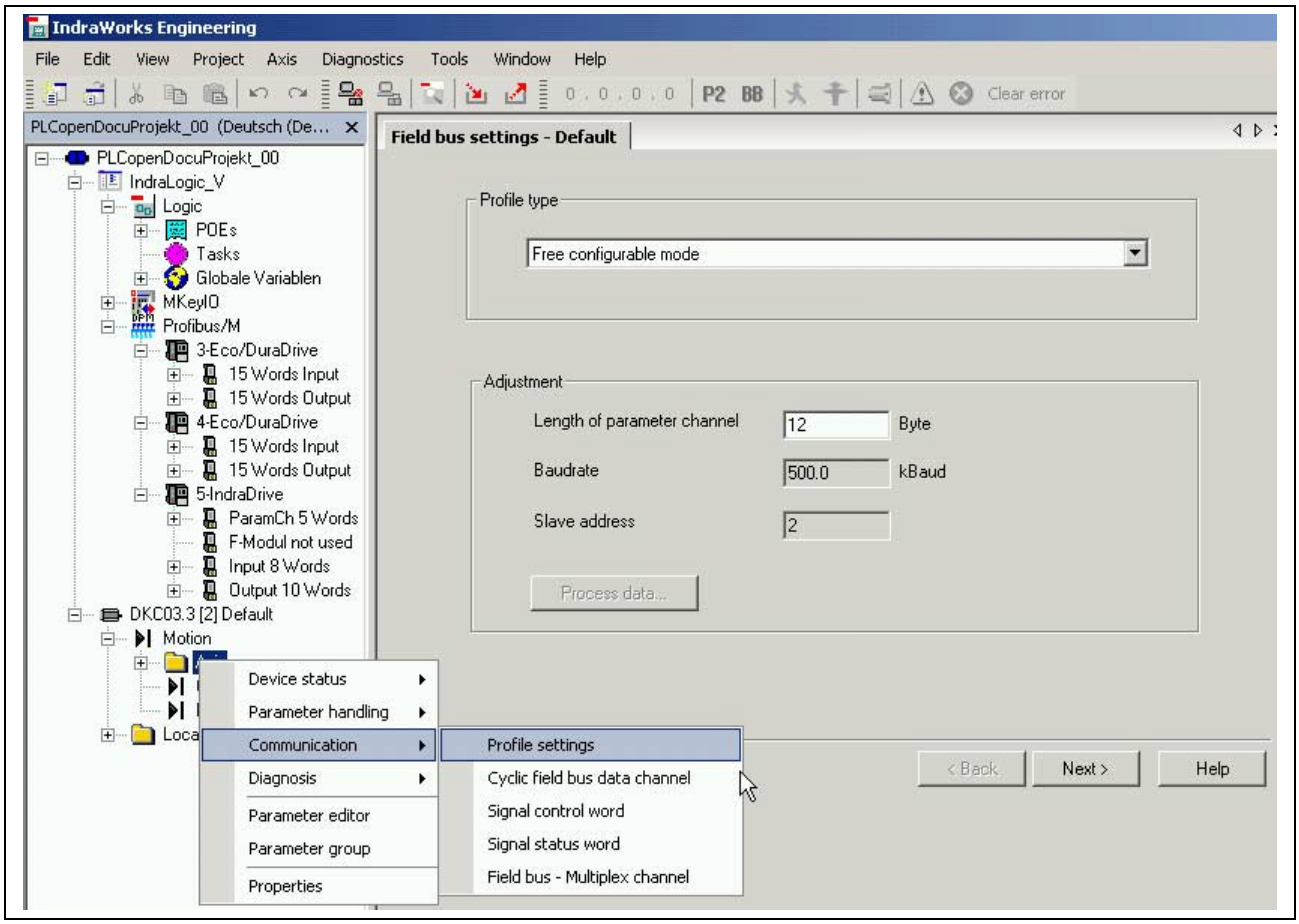


Fig. 2-8: Example of an IW dialog window for profile settings of an EcoDrive

Cyclic Field Bus Data Channel

Use this item to set the parameters of the cyclic data exchange with the PLC:

- Configuration list of the process data input description or of the cyclic actual-value data channel (P-0-4078 Field bus status word)
  - P-0-4078 Field bus: status word
  - S-0-0051 Position feedback 1 value
  - S-0-0040 Velocity feedback value
  - S-0-0390 Diagnostic message number
  - S-0-0368 Addressing for data container A
  - P-0-0144 Signal status word
  - P-0-4076 Field bus container object
- Configuration list of the process data output description or of the cyclic command-value data channel (P-0-4077 Field bus control word)
  - P-0-4077 Field bus: control word
  - S-0-0282 Positioning command value
  - S-0-0259 Positioning velocity
  - S-0-0368 Addressing for data container A
  - S-0-0360 MDT Data container A
  - S-0-0145 Signal control word

Signal Control Word

Only the first bit is used in the signal control word:

- S-0-0393 Command value mode
  - Bit 2

To enter these values, the parameters below must be assigned as follows:

- S-0-0027, Configuration list signal control word

  S-0-0393 Command value mode assigned to element 0
- S-0-0329, Assign list signal control word

  2 (bit 2) assigned to element 0

The other bit positions remain available for use outside of the PLCopen function blocks.

Signal Status Word

The bits of the signal status word remain completely available for use outside of the PLCopen function blocks.

Multiplex Channel Settings

EcoDrive CS also requires the multiplex channel for data transfer.

- Configuration of MDT data container
  - S-0-0260 Positioning acceleration
  - S-0-0359 Positioning deceleration
  - P-0-4030 Jog velocity
- Configuration of AT data container
  - S-0-0260 Positioning acceleration
  - S-0-0353 Positioning deceleration
  - S-0-0040 Velocity feedback value

Mode Selection

The drive needs "Drive-controlled positioning" as the primary mode and "Jog mode" as the first secondary mode. The second and third modes are of no relevance, since they are not selected by the PLCopen function blocks.

Note: The MB_Home and MP_Home function blocks execute homing only if the drive has been parameterized for homing.

2.3 Necessary Libraries

The PLCopen function blocks and the functions required for operation are packed in libraries.

MP_PLCopen.lib

This *.lib file contains the PLCopen function blocks with the prefix MC_ defined in PLCopen, e. g. MC_MoveAbsolute.

This *.lib file functions only in connection with the PLCopenFieldBus.lib library. As a matter of principle, the MP_PLCopen.lib and PLCopenFieldBus.lib libraries can be run on all IndaLogic target systems. For example, an MLC (Motion Logic) also comprises a library with PLCopen function blocks which optimally communicate with the drives through a SERCOS interface. The physics of activation through a SERCOS interface, however, is completely different, so that the same function blocks for SERCOS axes cannot simultaneously activate field bus axes. Here, it is not possible to use the MP_PLCopen.lib library for operating the field bus axes since, otherwise, two function blocks with the same designation would be available. To enable this mixed mode nevertheless, the MP_PLCopen.lib library only contains PLCopen function blocks which call up the actual PLCopen functions from the PLCopenFieldBus.lib library. This *.lib file contains the actual function blocks for field bus drives. The function blocks in the PLCopenFieldBus.lib library have the prefix MP_, e. g. MP_MoveAbsolute.

PLCopenFieldBus.lib

This *.lib file contains the actual function blocks which communicate with the field bus axes. These have the prefix MP_, e. g. MP_MoveAbsolute.

RIL_CommonTypes.lib The RIL_CommonTypes.lib library contain the generally applicable IndraLogic data type declarations, such as AXIS_REF or the structures and enumeration values of error handling.

RIL_Check.lib The PLCopenFieldBus.lib library requires the RIL_Check.lib library. The RIL_Check.lib library provides necessary technical safety system functions, such as the function of monitoring whether array limits are kept.

A PLC project containing field bus axes to be activated with the PLCopen functions of the PLCOpenFieldBus.lib library accordingly requires incorporation of these three *.lib files.

Note: If you intend to do without the safety mechanisms of the RIL_Check.lib library, the global variable CheckExceedingOccurred of type BYTE which is contained in the *.lib file must be declared in the PLC program.

If the RIL_Check.lib library fails to be provided, any incorrect access to non-available axes or any improper programming will not be detected, such causing unforeseen effects, e. g. uncontrolled axis movements in the worst case.

2.4 Configuration in the PLC Program

This chapter describes the steps required for incorporating field bus drives in IndraLogic through the program.

The functions can also be incorporated in an existing PLC program subsequently.

Incorporating the Necessary Libraries

The settings described in Section 2.1, "Basic PLC Settings" and Section 2.2 "Drive Settings Required for Communication" have been made. Now enter the necessary libraries in a new or an existing PLC program:

- RIL_CommonTypes.lib
- RIL_Check.lib
- PLCOpenFieldBus.lib
- MP_PLCOpen.lib, if necessary

Usually, these files are contained in the target installation. In the default installation, the path is as follows:

Installation drive:\Programme\Rexroth\IndraWorks\IndraLogic\Targets\ *Target name*

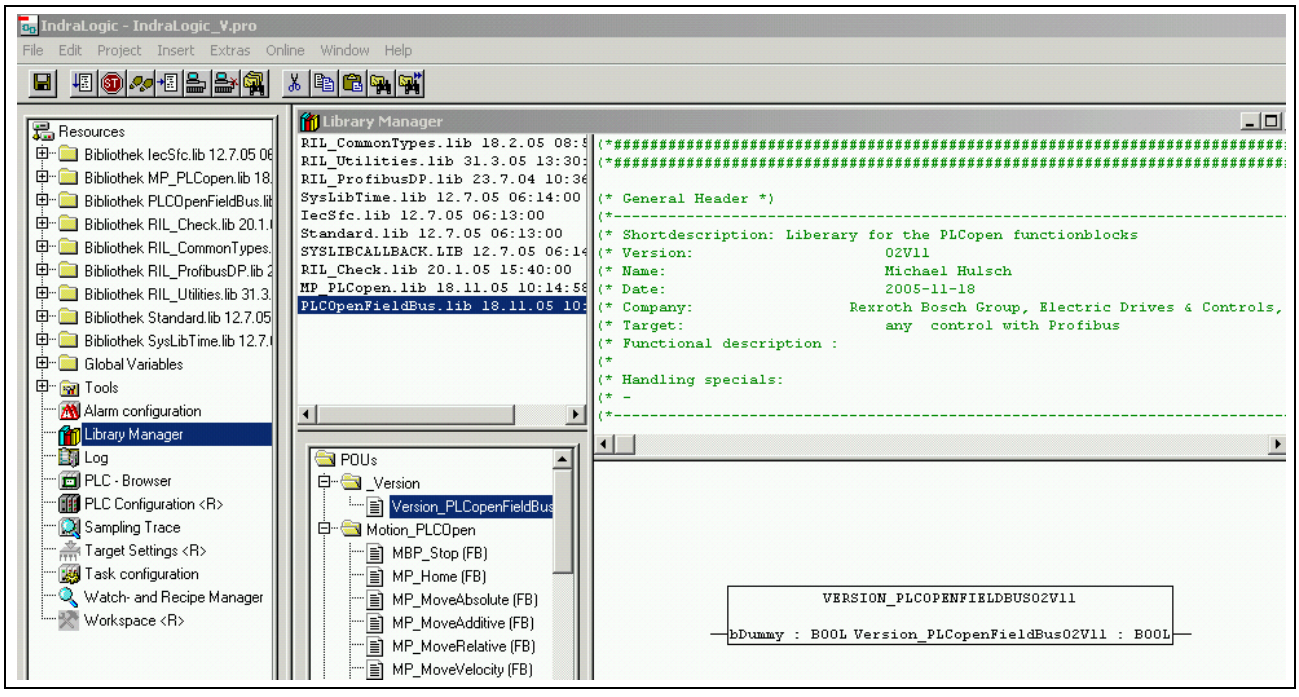


Fig. 2-9: Entering the necessary libraries in a PLC program

Note: If the PLCOpenFieldBus.lib library is incorporated in a PLC program, the declaration of at least one axis

- AXIS_DATA: ARRAY [FieldBusAxis_1..FieldBusAxis_1] OF FieldBusDriveControlData

and two programs

- "PR_FieldBusDrives" and "PR_FieldBusDrivesParameter"

must be included. Otherwise, there will be error messages during translation.

Necessary Declarations

AXIS_REF

Using the structure AXIS_REF, each axis is described with reference to its number and to the controller it belongs to. The assigned axis number does not necessarily refer to the station number set at the drive.

```
Drive_XYZ: AXIS_REF :=
            (CntrlNo:=LOCAL_CNTRL,
            AxisNo:=FieldBusAxis_1);
Drive_EcoDrive: AXIS_REF :=
            (CntrlNo:=LOCAL_CNTRL,
            AxisNo:=FieldBusAxis_2);
Drive_EcoDriveCS: AXIS_REF :=
            (CntrlNo:=LOCAL_CNTRL,
            AxisNo:=FieldBusAxis_3);
Drive_IndraDrive: AXIS_REF :=
            (CntrlNo:=LOCAL_CNTRL,
            AxisNo:=FieldBusAxis_4);
```

Fig. 2-10: AXIS_REF (global) declaration

AXIS_REF Declaration *Drive_XYZ :AXIS_REF :=*

The necessary data types are contained in the RIL_CommonTypes.lib library, "Datatypes of PLCopen" folder, AXIS_REF element.

CntrlNo Assignment *(CntrlNo := LOCAL_CNTRL*

The enumeration value LOCAL_CNTRL is contained in the RIL_CommonTypes.lib library, "Datatypes of logical addresses" folder, CONTROLS element.

The field bus modules always refer to the local PLC, that is LOCAL_CNTRL.

AxisNo Assignment *AxisNo:=FieldBusAxis_1);*

The enumeration value FieldBusAxis_1 is contained in the RIL_CommonTypes.lib library, "Datatypes of logical addresses" folder, OBJECTS element.

In order to avoid ambiguities, a separate range of numbers, that is FieldBusAxis_1 (6001) to FieldBusAxis_99 (6099), is filed here for the field bus axes. It is appropriate, although not mandatory, to assign the numbers consecutively without any gap. This consecutive assignment has the benefit that the declaration of the communication array will not become too large, because this communication array is declared from the lowest axis number to the highest axis number. This will be described in the following sections.

AXIS_DATA

The data of this array is used to enable communication of the PLCopen function blocks (MP_MoveAbsolute, MP_ReadStatus, etc.) with the actual axis-related function blocks (MP_IndraDrive_Command_MPH02VRS, MP_Ecodrive_Command_FGP03VRS).

```
AXIS_DATA: ARRAY [FieldBusAxis_1..
                  FieldBusAxis_4]
                  OF FieldBusDriveControlData;
```

Fig. 2-11: Example of an AXIS_DATA declaration

AXIS_DATA *AXIS_DATA: ARRAY*

The declaration of this array which carries the designation AXIS_DATA is mandatory in the project.

FieldBusAxis_1 (Lowest Axis Number Assigned) *FieldBusAxis_1*

Enter the lowest axis number assigned or the lowest axis enumeration value assigned as the lower dimension of the array.

FieldBusAxis_99 (Highest Axis Number Assigned) *FieldBusAxis_99*

Enter the highest axis number assigned or the highest axis enumeration value assigned as the upper dimension of the array (FieldBusAxis_4 in the example, however not higher than FieldBusAxis_99).

FieldBusDriveControlData *FieldBusDriveControlData*

The data type is contained in the PLCopenFieldBus.lib library, "Datatypes_PLCopen" folder.

Control Function Blocks

The control function blocks MP_Ecodrive_Command_FGP03VRS and/or MP_IndraDrive_Command_MPH02VRS assume the actual control of the axes. For that reason, they relate to the hardware and must be selected depending on the type of the axis, i. e. EcoDrive or IndraDrive. These function blocks do not assume the parameter communication of the MP_Read/WriteParameter and MP_Read/WriteRealParameter function blocks. As a result, resources that are not required can be saved in small control systems.

```
DriveEcoDrive:   MP_Ecodrive_Command_FGP03VRS;
DriveIndraDrive: MP_IndraDrive_Command_MPH02VRS;
```

Fig. 2-12: Example of the declaration of control function blocks

MP_Ecodrive_Command_FGP03VRS *DriveEcoDrive: MP_Ecodrive_Command_FGP03VRS*

This function block controls an EcoDrive, an EcoDrive CS or a DuraDrive. The instance name can be assigned as desired.

MP_IndraDrive_Command_MPH02VRS *DriveIndraDrive: MP_IndraDrive_Command_MPH02VRS*

This function block controls an IndraDrive. The instance name can be assigned as desired.

PR_FieldBusDrives

This program must only be created, i. e. be available in the PLC program, if the PLCOpenFieldBus.lib library has been entered in the PLC program. The name is defined because it is called up by the actual PLCopen functions blocks, such as MP_MoveAbsolute or MP_ReadStatus, etc. The previously declared axis-related control function blocks are programmed to this program.

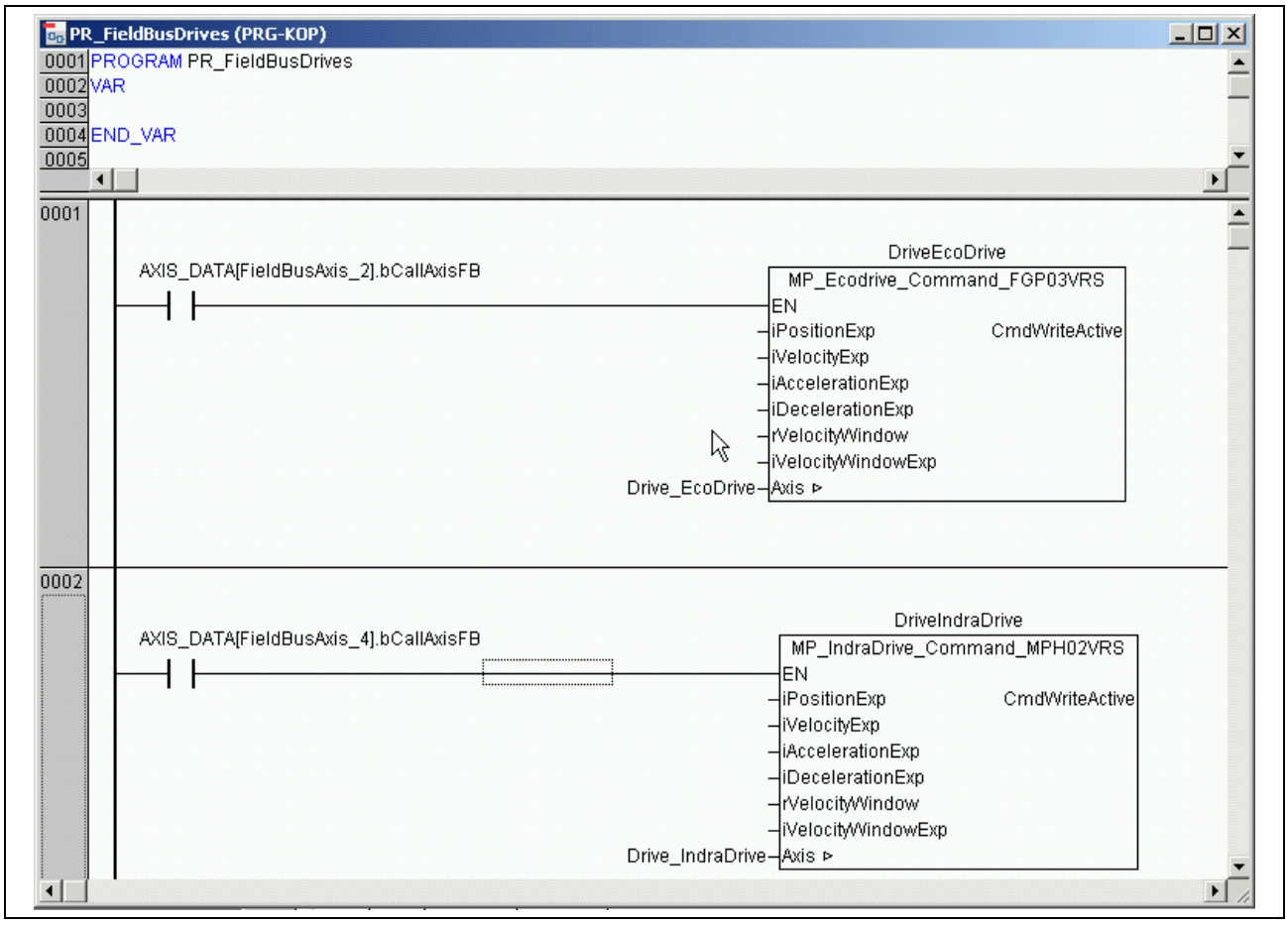


Fig. 2-13: Example of PR_FieldBusDrives

AXIS_DATA[FieldBusAxis_2]bCallAxisFB at the EN input

AXIS_DATA at the Control Function Block

This contact is entered at the EN input of the control function block *DriveEcoDrive: MP_Ecodrive_Command_FGP03VRS* and/or *MP_IndraDrive_Command_MPH02VRS* to optimize the runtime. The array element AXIS_DATA*[FieldBusAxis_xx]* relates to the enumeration value entered at the declaration of AXIS_REF, AxisNo element. The element *bCallAxisFB* is managed by the PLCopen function blocks and is connected to the module here only. Any further use is not permitted; this applies particularly to a writing access in the user PLC program.

Velocity and Acceleration Evaluation

iPositionExp; iVelocityExp; iAccelerationExp; iDecelerationExp

These inputs of the control function blocks are provided for being able to alter the evaluation of the velocity and acceleration values as compared with the default settings of a drive. 4 decimal places are pre-assigned to the position and velocity data; 3 decimal places are pre-assigned to the acceleration data. If the application of a drive is different from these default values, they must be specified here. The numerical value to be specified is the evaluation, i. e. the number of places after the decimal point. A numerical value of 4 means 4 decimal places; a numerical value of 3 means 3 decimal places, etc.

InVelocity Window

rVelocityWindow

This is the window for formation of the **InVelocity** message.

iVelocityWindowExp

The number of places after the decimal point is as described above.

Axis

Axis

The previously globally declared AXIS_REF of the particular axis is created at the Axis input.

PR_FieldBusDrivesParameter

The control function blocks MP_Ecodrive_Command_FGP03VRS and MP_IndraDrive_Command_MPH02VRS are designed for all default communication processes. If, however, parameters must be read or written directly in the drive (with MP_ReadParamter, MP_WriteParameter, MP_ReadRealParamter, MP_WriteRealParameter), then the control function block MP_IndraDrv_AcyclicCommunication must be declared and programmed in analogy with the normal control function block.

```
DriveIndraDrive_Parameter:
                    MP_IndraDrv_AcyclicCommunication;
DriveIndraDrive:
                    MP_IndraDrive_Command_MPH02VRS;
```

Fig. 2-14: Example of a global IndraDrive declaration with parameter communication

The control function block itself is then only called in the PR_FieldBusDrivesParameter program, in analogy to the control function block MP_IndraDrive_Command_MPH02VRS.

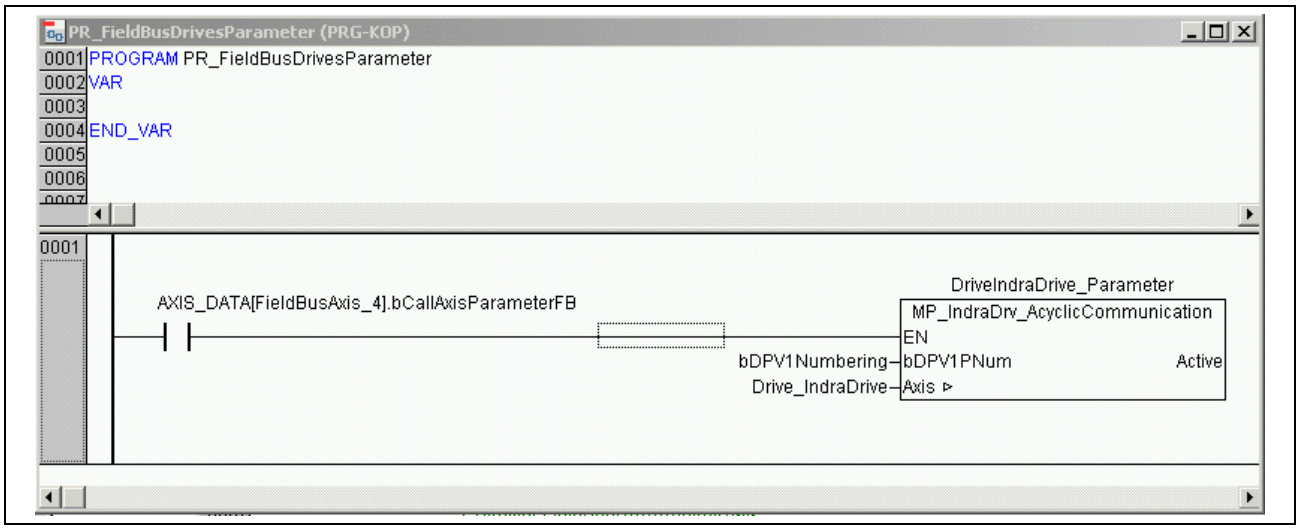


Fig. 2-15: Example of PR_FieldBusDrivesParameter

This program must only be created, i. e. be available in the PLC program, if the PLCOpenFieldBus.lib library has been entered in the PLC program. The name is defined, because it is called by the actual PLCopen function blocks MP_ReadParameter, MP_WriteParameter, MP_ReadRealParameter and MP_WriteRealParameter. The previously declared axis-related control function blocks are programmed to this program.

*AXIS_DATA[FieldBusAxis_4].**bCallAxisParameterFB** at the EN input*

This contact is entered at the EN input of the control function block *MP_IndraDrive_Command_MPH02VRS* to optimize the runtime. The array element AXIS_DATA*[FieldBusAxis_xx]* relates to the enumeration value entered at the declaration of AXIS_REF, AxisNo element. The element *bCallAxisParameterFB* is managed by the PLCopen function blocks and is connected to the module here only. Any further use is not permitted; this applies particularly to a writing access in the user PLC program.

bDPV1PNum

There are two types of numbering the parameters:

- According to DPV1; please refer to Chapter "Function Blocks for Parameter Access" on page 1-13. To achieve this, the SERCOS parameters are filed as constants for S-parameters in MP_PLCopen.lib, global variables, IL_S_Param_ID and for P-parameters in IL_P_Param_ID.
- Alternatively, there are enumeration values for field bus drives only, according to the object directory of Profibus DP (ProfiDrive). These are residing in the PLCopenFieldBus.lib library, under the data types item, in the Datatypes_PLCopen folder, DriveParameter.

Axis *Axis*

The previously globally declared AXIS_REF of the axis to be controlled is created at the Axis input.

I/O Addressing of the Drives

The field bus drives appear as normal I/O modules on Profibus. I/O addresses must be assigned accordingly. To facilitate assignment of I/O addresses, it has been performed as variable configuration within the control function blocks.

As described in Chapter "Control Function Block" on page 2-16, only the I/O start addresses relating to the I/O modules of the drives must be specified after the control function blocks have been declared. To achieve this, open "Variable_Configuration (VAR_CONFIG)" under "Global Variables" in the "Resources" folder.

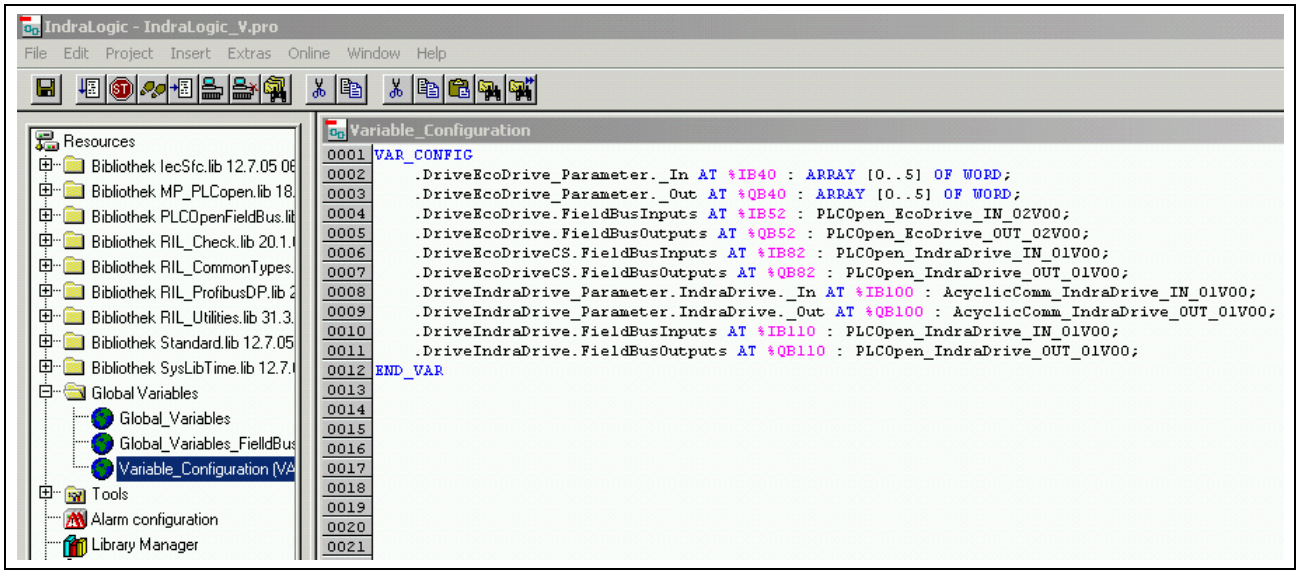


Fig. 2-16: Example of addressing I/O addresses of the field bus drives "Insert – All instance paths"

If the focus is on the "Variable_Configuration" window, you can generate a list containing all instances still missing an I/O address assignment by executing the "All instance paths" under the "Insert" menu item.

This list contains two lines for each control function block:

- Inputs => status signals from the drive
- Outputs => command signals to the drive

Assigning I/O Addresses for EcoDrive, EcoDrive CS and DuraDrive

The data exchange between the drive and the PLC comprises two parts. The first six words stand for acyclic parameter communication; the next nine words stand for cyclic data exchange, so that the total length on Profibus is 15 words.

Example: In IndraWorks, an EcoDrive has been set to the input address %IB40 and to the output address %QB40. These start addresses must comply with the start addresses of the EcoDrive entered under VAR_CONFIG.

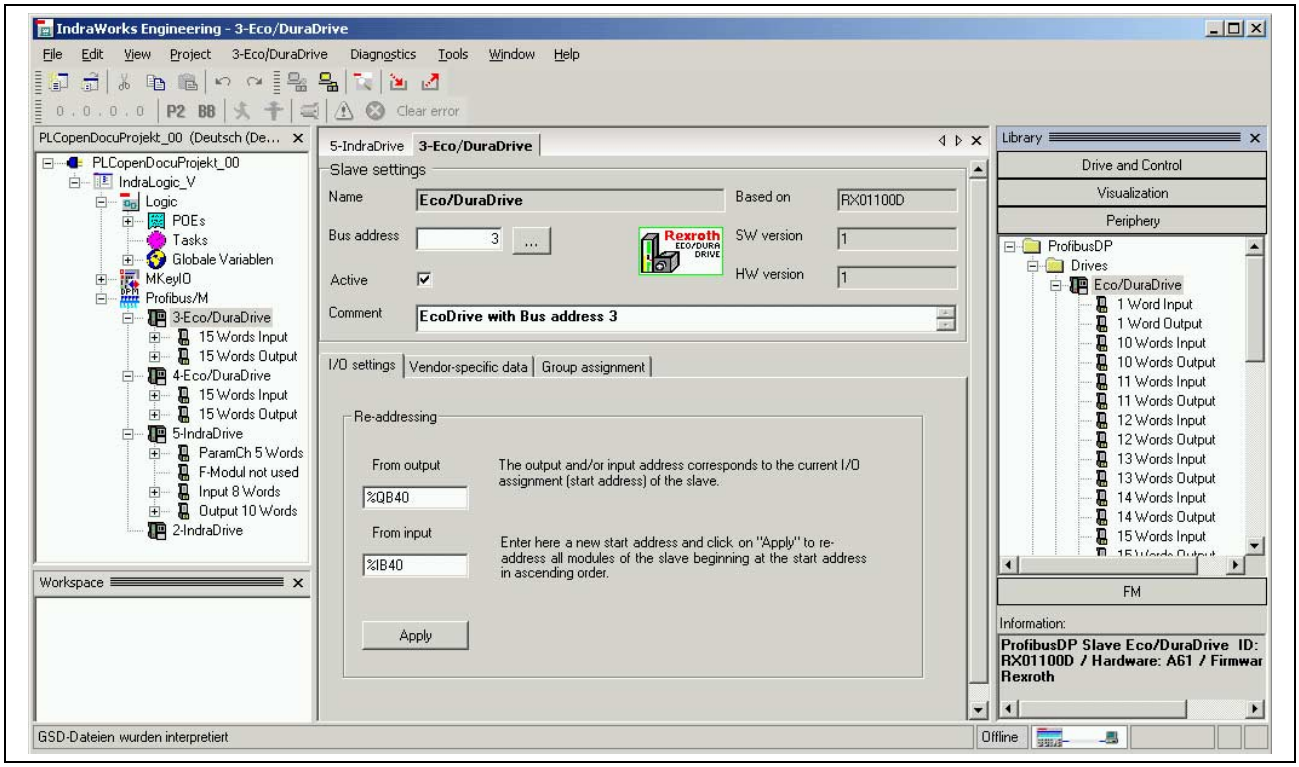


Fig. 2-17: Example of an I/O address for EcoDrive, EcoDrive CS, and DuraDrive in IndraWorks

```
VAR_GLOBAL
   .DriveEcoDrive_Parameter._In AT %IB40 : ARRAY [0..5] OF WORD; (*Parameter communication*)
   .DriveEcoDrive_Parameter._Out AT %QB40 : ARRAY [0..5] OF WORD; (*Parameter communication*)
   .DriveEcoDrive.FieldBusInputs AT %IB52 : PLCOpen_EcoDrive_IN_02V00; (*Cycl. Data exch.*)
   .DriveEcoDrive.FieldBusOutputs AT %QB52 : PLCOpen_EcoDrive_OUT_02V00; (*Cycl. Data exch.*)
END_VAR
```

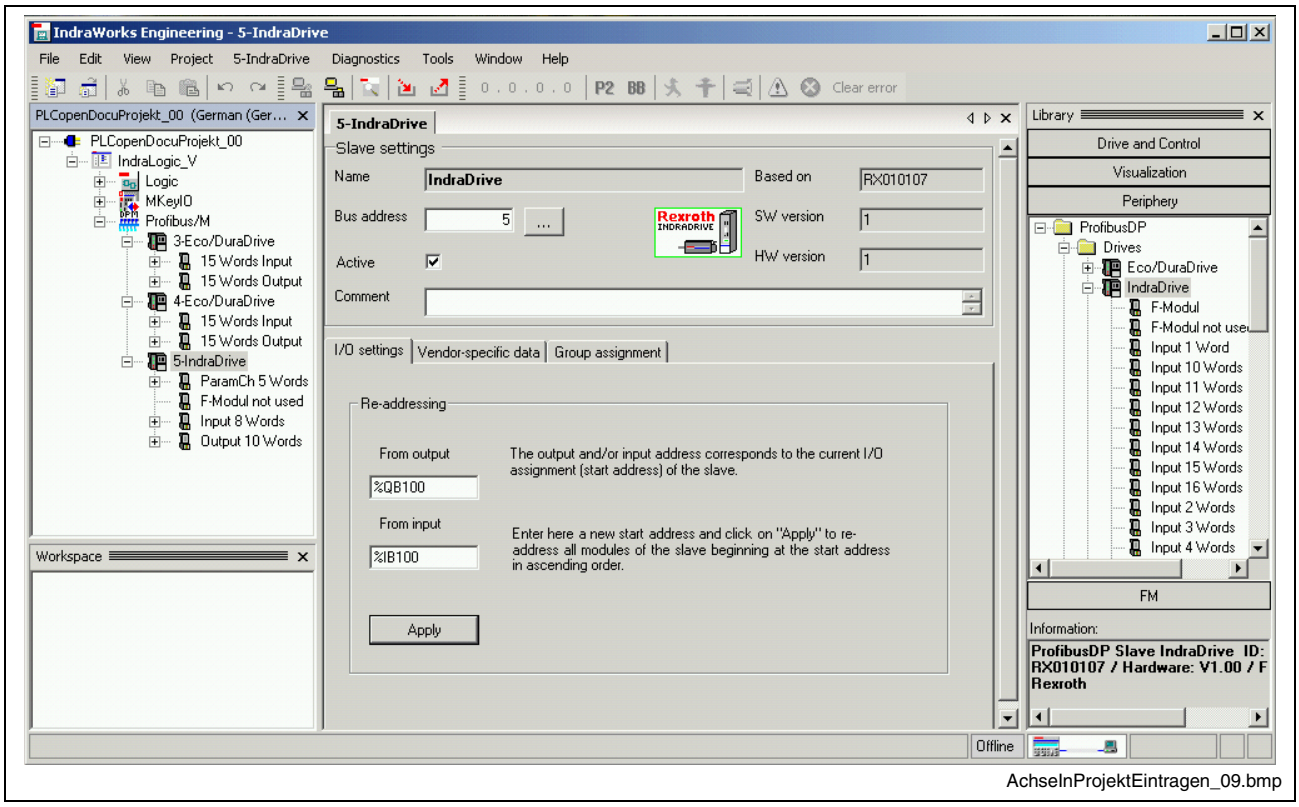
Fig. 2-18: Example of an I/O address assignment for EcoDrive, EcoDrive CS, and DuraDrive

Since, contrary to IndraDrive, the parameter communication does not represent a separate Profibus submodule, this start address is applicable to the parameter communication only. The cyclic data exchange (<Instance name of control function block>.FieldBusInputs and <Instance name of control function block>.FieldBusOutputs) has an offset of 12 bytes as compared with this start address and is accordingly calculated by adding 12 bytes to the start address of the drive.

Assigning I/O Addresses for IndraDrive

The drive-to-PLC communication comprises up to three parts. The first five words stand for acyclic parameter communication; the following addresses are those of the F-module; and the following eight input words or 10 output words stand for cyclic data exchange. That means that the start address of cyclic data exchange is additionally depending on an F-module if any is installed (technical safety). Any non-installed F-module (technical safety) has a length of 0 bytes.

Example: In IndraWorks, an IndraDrive has been set to the input address %IB100 and to the output address %QB100. These start addresses must comply with the start addresses of the IndraDrive entered under VAR_CONFIG.



AchseInProjektEintragen_09.bmp

Fig. 2-19: Example of an IndraDrive I/O address

```
VAR_GLOBAL
  .DriveIndraDrive_Parameter.IndraDrive._In AT %IB100 : AcyclicComm_IndraDrive_IN_01V00;
  .DriveIndraDrive_Parameter.IndraDrive._Out AT %QB100 : AcyclicComm_IndraDrive_OUT_01V00;
  .DriveIndraDrive.FieldBusInputs AT %IB110 : PLCOpen_IndraDrive_IN_01V00; (*Cycl. Data exch.*)
  .DriveIndraDrive.FieldBusOutputs AT %QB110 : PLCOpen_IndraDrive_OUT_01V00; (*Cycl. Data exch.*)
END_VAR
```

Fig. 2-20: Example of assigning an IndraDrive I/O address

As regards IndraDrive, the parameter communication, the F-module (technical safety) and the cyclic communication each represent separate submodules of the device. Separate start addresses can be assigned to each of these submodules. Unless these start addresses have been moved during project planning, the start addresses are added from the module lengths. The start address indicated in the main tab is applicable to the parameter communication here as well. The start address of the cyclic data exchange accordingly consists of the start address of the parameter communication (10 bytes) and that of the F-module (0 bytes without module or the number of bytes of an existing module). In the example, the start address of the cyclic data exchange has an offset of 10 bytes as compared with the address of the parameter communication, because there is no F-module and, thus, 0 bytes are added for the F-module.

2.5 Acyclic Parameter Communication with EcoDrive, EcoDrive CS or DuraDrive

In EcoDrive, EcoDrive CS and DuraDrive, parameters cannot use the standardized PLCopen function blocks (MB_ReadParameter, MB_ReadRealParameter, MB_WriteParameter, MB_WriteRealParameter). However, simple parameter communication with these drives can be achieved by means of the function block MP_Ecodrive_AcyclicCommunication_FGP03VRS_01V00.

MP_Ecodrive_AcyclicCommunication_FGP03VRS_02V00

Brief Description

The function block assumes acyclic communication for reading a parameter attribute as well as reading and writing of parameter contents to the EcoDrive, EcoDrive CS or DuraDrive.

Note: The FB is designed such that one FB instance communicates with one drive. For that reason, only one FB instance may be declared for each drive. Declaration of the FB in the global variable range (VAR_GLOBAL) is appropriate.

Library	Range
PLCOpenFieldbus.lib	SystemFBs_PLCopen

Fig. 2-21: Library assignment

Interface Description

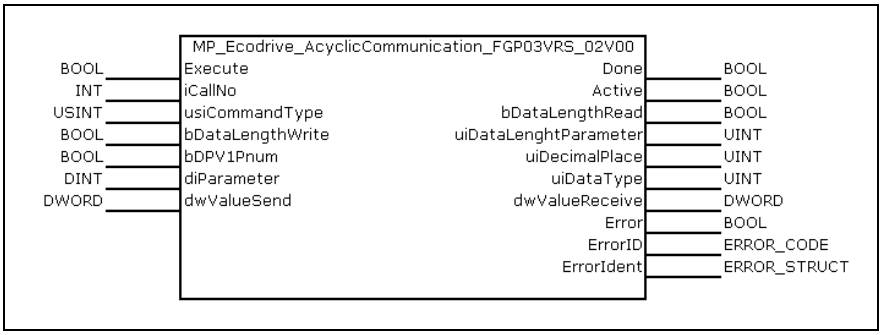


Fig. 2-22: FB MP_Ecodrive_AcyclicCommunication_FGP03VRS_02V00

	Name	Type	Comment
VAR_INPUT	Execute	BOOL	Starts to write or read the value present with rising edge.
	ICallNo	INT	Is not evaluated in the FB. Can be used for locking in the program, if the FB is to be called at more program sites than one. **Only one instance of the FB is permitted for each drive.**
	UsiCommandType	USINT	0 = reading of the attribute; output to uiDecimalPlace, uiDataType and dwValueReseive. 1 = reading of the parameter value; output to "dwValueReseive" 2 = writing of the value at the "dwValueSend" input to the parameter
	BDataLengthWrite	BOOL	FALSE = 2-byte value TRUE = 4-byte value
	bDPV1Pnum	BOOL	TRUE = parameter numbering according to DPV1, FALSE = according to the object directory of Profibus-DP (ProfiDrive)
	DiParameter	DINT	Number of the parameter desired (only S- or P-drive parameters)
	DwValueSend	DWORD	Integer non-evaluated value to be written to the parameter
VAR_OUTPUT	Done	BOOL	The attribute or the value has been successfully read from or written to the drive.
	Active	BOOL	Processing of data is active after preprocessing is completed.
	BDataLengthRead	BOOL	Data length of the read data (output dwValueReceive): FALSE = 2-byte value TRUE = 4-byte value
	UiDataLengthParameter	UINT	While the attribute is read, the data length of the parameter is indicated: 1 = 2-byte value 2 = 4-byte value 4 = 1-byte values, variable length (list parameter) 5 = 2-byte values, variable length (list parameter) 6 = 4-byte values, variable length (list parameter)
	UiDecimalPlace	UINT	While the attribute is read, the number of places after the decimal point of the parameter is indicated: Relevant for uiDataType = 2 or 3
	UiDataType	UINT	Parameter data type 0 = binary number 1 = unsigned integer decimal number 2 = integer decimal number 3 = hexadecimal number 4 = text 5 = ident number (2 bytes) 6 = floating-point number according to ANSI/IEEE 754-1985 7 = parameter number (4 bytes)
	DwValueReceive	DWORD	usiCommandType = 0 (reading of attribute) => the attribute of the parameter is output; usiCommandType = 1 (reading of parameter value) => the read integer non-evaluated content of the parameter is output; usiCommandType = 2 (writing to parameter) => the written value, that is dwValueSend, is mirrored when output.
	Error	BOOL	Indicates that an error has occurred in the FB instance.

	Name	Type	Comment
VAR_OUTPUT	ErrorID	ENUM ERROR_CODE	Indicates the basic error cause in the event of an error, using the enumeration values of ERROR_CODE.
	ErrorIdent	STRUCT ERROR_STRUCT	Provides detailed information on the error; this structure consists of ERROR_TABLE, Additional1 and Additional2.

Fig. 2-23: Interface of the FB MP_Ecodrive_AcyclicCommunication_FGP03VRS_02V00

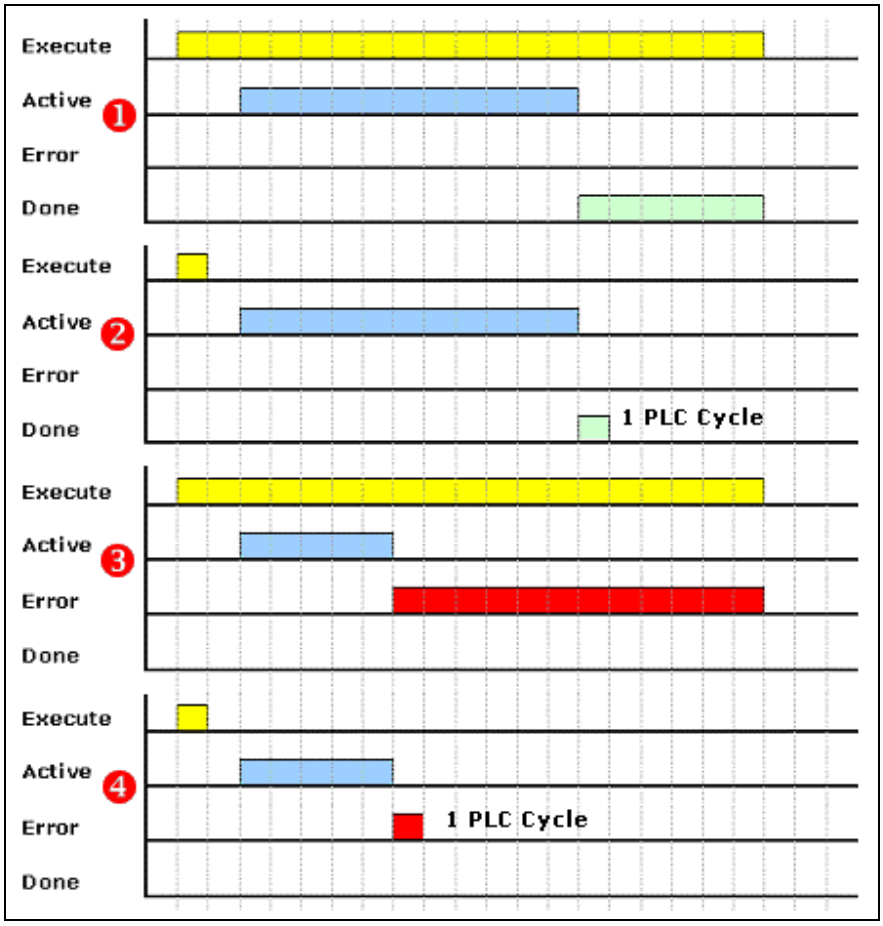


Fig. 2-24: Default signal time diagram

Functional Description

The function block assumes acyclic communication for reading a parameter attribute as well as reading and writing of parameter contents to the EcoDrive, EcoDrive CS or DuraDrive.

This function block does not allow reading of list parameters.

Communication always takes place between one FB instance and one drive. While the FB is declared, the inputs and outputs for acyclic parameter communication must also be assigned at the same time. The I/O start addresses of the drive must be specified after the function block has been declared. To achieve this, open "Variable_Configuration (VAR_CONFIG)" under "Global Variables" in the "Resources" folder. If the focus is on the "Variable_Configuration" window, you can generate a list containing all instances still missing an I/O address assignment by executing the "All instance paths" under the "Insert" menu item.

This list contains two lines for each function block:

- Inputs => status signals from the drive
- Outputs => command signals to the drive

```
VAR_GLOBAL
  fbParamterRW_Ecodrv: MP_Ecodrive_AcyclicCommunication_FGP03VRS_02V00;
  DriveEcoDrive:
MP_Ecodrive_Command_FGP03VRS; (*Field bus drive Y of the IndraDrive type*)
END_VAR

VAR_CONFIG
  .fbParamterRW_Ecodrv._In AT %IB30 : ARRAY [0..5] OF WORD;
  .fbParamterRW_Ecodrv._Out AT %QB30 : ARRAY [0..5] OF WORD;
  .DriveEcoDrive.FieldBusInputs AT %IB42 : PLCOpen_EcoDrive_IN_02V00;
  .DriveEcoDrive.FieldBusOutputs AT %QB42 : PLCOpen_EcoDrive_OUT_02V00;
END_VAR
```

Fig. 2-25: Example of assigning I/O addresses for acyclic communication with EcoDrive

In the example, 30 bytes are assigned to each of the start addresses of EcoDrive, i. e. inputs and outputs. For that reason, the start address for acyclic communication is also %IB30 and %QB30 respectively.

Note: The MP_Ecodrive_AcyclicCommunication_FGP03VRS_02V00 function block works only in connection with EcoDrive, EcoDrive CS and DuraDrive.

Error Handling

This function block generates error messages only if there is a positive edge or TRUE at the "Execute" input. The indicated error numbers originate directly from the drive and can be found in the drive documentation.

3 List of Figures

4 Index

F

I

L

M

N

O

P

R

S

T

V

5 Service & Support

5.1 Helpdesk

Unser Kundendienst-Helpdesk im Hauptwerk Lohr am Main steht Ihnen mit Rat und Tat zur Seite. Sie erreichen uns

Our service helpdesk at our headquarters in Lohr am Main, Germany can assist you in all kinds of inquiries. Contact us

- telefonisch - by phone: **+49 (0) 9352 40 50 60**
  über Service Call Entry Center Mo-Fr 07:00-18:00 Central European Time
  - via Service Call Entry Center Mo-Fr 7:00 am - 6:00 pm CET
- per Fax - by fax: **+49 (0) 9352 40 49 41**
- per e-Mail - by e-mail: service.svc@boschrexroth.de

5.2 Service-Hotline

Außerhalb der Helpdesk-Zeiten ist der Service Deutschland direkt ansprechbar unter

After helpdesk hours, contact the German service experts directly at

+49 (0) 171 333 88 26

oder - or **+49 (0) 172 660 04 06**

Hotline-Rufnummern anderer Länder entnehmen Sie bitte den Adressen in den jeweiligen Regionen.

Hotline numbers of other countries to be seen in the addresses of each region.

5.3 Internet

Unter **www.boschrexroth.com** finden Sie ergänzende Hinweise zu Service, Reparatur und Training sowie die **aktuellen** Adressen *) unserer auf den folgenden Seiten aufgeführten Vertriebs- und Servicebüros.

Verkaufsniederlassungen
Niederlassungen mit Kundendienst

Außerhalb Deutschlands nehmen Sie bitte zuerst Kontakt mit unserem für Sie nächstgelegenen Ansprechpartner auf.

*) Die Angaben in der vorliegenden Dokumentation können seit Drucklegung überholt sein.

At **www.boschrexroth.com** you can find additional notes about service, repairs and training. The current addresses *) for our sales and service facilities locations around the world are on the following pages.

sales agencies
sales agencies providing service

Please contact our sales / service office in your area first.

*) Data in the present documentation may have become obsolete since printing.

5.4 Vor der Kontaktaufnahme... - Before contacting us...

Wir können Ihnen schnell und effizient helfen wenn Sie folgende Informationen bereithalten:

1. detaillierte Beschreibung der Störung und der Umstände.
2. Angaben auf dem Typenschild der betreffenden Produkte, insbesondere Typenschlüssel und Seriennummern.
3. Tel.-/Faxnummern und e-Mail-Adresse, unter denen Sie für Rückfragen zu erreichen sind.

For quick and efficient help, please have the following information ready:

1. Detailed description of the failure and circumstances.
2. Information on the type plate of the affected products, especially type codes and serial numbers.
3. Your phone/fax numbers and e-mail address, so we can contact you in case of questions.

5.5 Kundenbetreuungsstellen - Sales & Service Facilities

Deutschland – Germany

vom Ausland: (0) nach Landeskennziffer weglassen!
from abroad: don't dial (0) after country code!

Vertriebsgebiet Mitte Germany Centre	SERVICE AUTOMATION	SERVICE AUTOMATION	SERVICE AUTOMATION
Bosch Rexroth Electrice Drives and Controls GmbH Bgm.-Dr.-Nebel-Str. 2 / Postf. 1357 97816 Lohr am Main / 97803 Lohr **Kompetenz-Zentrum Europa** Tel.: +49 (0)9352 40-0 Fax: +49 (0)9352 40-4885	CALL ENTRY CENTER Helpdesk MO – FR von 07:00 - 18:00 Uhr from 7 am – 6 pm **Tel. +49 (0) 9352 40 50 60** **Fax +49 (0) 9352 40 49 41** service.svc@boschrexroth.de	HOTLINE 24 / 7 / 365 außerhalb der Helpdesk-Zeit out of helpdesk hours **Tel.: +49 (0)172 660 04 06** oder / or **Tel.: +49 (0)171 333 88 26**	**ERSATZTEILE / SPARES** verlängerte Ansprechzeit - extended office time - ♦ nur an Werktagen - only on working days - ♦ von 07:00 - 18:00 Uhr - from 7 am - 6 pm - **Tel. +49 (0) 9352 40 42 22**
Vertriebsgebiet Süd Germany South	Vertriebsgebiet West Germany West	Gebiet Südwest Germany South-West	
Bosch Rexroth AG Landshuter Allee 8-10 80637 München Tel.: +49 (0)89 127 14-0 Fax: +49 (0)89 127 14-490	Bosch Rexroth AG Regionalzentrum West Borsigstrasse 15 40880 Ratingen Tel.: +49 (0)2102 409-0 Fax: +49 (0)2102 409-406 +49 (0)2102 409-430	Bosch Rexroth AG Service-Regionalzentrum Süd-West Siemensstr. 1 70736 Fellbach Tel.: +49 (0)711 51046–0 Fax: +49 (0)711 51046–248	
Vertriebsgebiet Nord Germany North	Vertriebsgebiet Mitte Germany Centre	Vertriebsgebiet Ost Germany East	Vertriebsgebiet Ost Germany East
Bosch Rexroth AG Walsroder Str. 93 30853 Langenhagen Tel.: +49 (0) 511 72 66 57-0 Service: +49 (0) 511 72 66 57-256 Fax: +49 (0) 511 72 66 57-93 Service: +49 (0) 511 72 66 57-783	Bosch Rexroth AG Regionalzentrum Mitte Waldecker Straße 13 64546 Mörfelden-Walldorf Tel.: +49 (0) 61 05 702-3 Fax: +49 (0) 61 05 702-444	Bosch Rexroth AG Beckerstraße 31 09120 Chemnitz Tel.: +49 (0)371 35 55-0 Fax: +49 (0)371 35 55-333	Bosch Rexroth AG Regionalzentrum Ost Walter-Köhn-Str. 4d 04356 Leipzig Tel.: +49 (0)341 25 61-0 Fax: +49 (0)341 25 61-111

Europa (West) - Europe (West)

vom Ausland: (0) nach Landeskennziffer weglassen, Italien: 0 nach Landeskennziffer mitwählen
from abroad: don't dial (0) after country code, Italy: dial 0 after country code

Austria - Österreich

Bosch Rexroth GmbH
Electric Drives & Controls
Stachegasse 13
1120 Wien
Tel.: +43 (0) 1 985 25 40
Fax: +43 (0) 1 985 25 40-1459

Austria – Österreich

Bosch Rexroth GmbH
Electric Drives & Controls
Industriepark 18
4061 Pasching
Tel.: +43 (0)7221 605-0
Fax: +43 (0)7221 605-1220

Belgium - Belgien

Bosch Rexroth NV/SA
Henri Genessestraat 1
1070 Bruxelles

Tel: +32 (0) 2 451 26 08
Fax: +32 (0) 2 451 27 90
info@boschrexroth.be
service@boschrexroth.be

Denmark - Dänemark

BEC A/S
Zinkvej 6
8900 Randers

Tel.: +45 87 11 90 60
Fax: +45 87 11 90 61

Denmark - Dänemark

Bosch Rexroth A/S
Engelsholmvej 26
8900 Randers

Tel.: +45 36 77 44 66
Fax: +45 70 10 03 20
tj@boschrexroth.dk

Great Britain – Großbritannien

Bosch Rexroth Ltd.
Electric Drives & Controls
Broadway Lane, South Cerney
Cirencester, Glos GL7 5UH

Tel.: +44 (0)1285 863-000
Fax: +44 (0)1285 863-030
sales@boschrexroth.co.uk
service@boschrexroth.co.uk

Finland - Finnland

Bosch Rexroth Oy
Electric Drives & Controls
Ansatie 6
017 40 Vantaa

Tel.: +358 (0)9 84 91-11
Fax: +358 (0)9 84 91-13 60

France - Frankreich

Bosch Rexroth SAS
Electric Drives & Controls
Avenue de la Trentaine
(BP. 74)
77503 Chelles Cedex
Tel.: +33 (0)164 72-63 22
Fax: +33 (0)164 72-63 20
Hotline: +33 (0)608 33 43 28

France - Frankreich

Bosch Rexroth SAS
Electric Drives & Controls
ZI de Thibaud, 20 bd. Thibaud
(BP. 1751)
31084 Toulouse
Tel.: +33 (0)5 61 43 61 87
Fax: +33 (0)5 61 43 94 12

France – Frankreich

Bosch Rexroth SAS
Electric Drives & Controls
91, Bd. Irène Joliot-Curie
69634 Vénissieux – Cedex
Tel.: +33 (0)4 78 78 53 65
Fax: +33 (0)4 78 78 53 62

France – Frankreich

Tightening & Press-fit:
Globe Techniques Nouvelles
143, Av. du Général de Gaulle
92252 La Garenne Colombes

Tel.: +33 (0)1 41 19 33 33

Italy - Italien

Bosch Rexroth S.p.A.
Strada Statale Padana
Superiore 11, no. 41
20063 Cernusco S/N.MI
Hotline: +39 02 92 365 563
Tel.: +39 02 92 365 1
Service: +39 02 92 365 300
Fax: +39 02 92 365 500
Service: +39 02 92 365 516

Italy - Italien

Bosch Rexroth S.p.A.
Via Paolo Veronesi, 250
10148 Torino

Tel.: +39 011 224 88 11
Fax: +39 011 224 88 30

Italy - Italien

Bosch Rexroth S.p.A.
Via Mascia, 1
80053 Castellamare di Stabia NA

Tel.: +39 081 8 71 57 00
Fax: +39 081 8 71 68 85

Italy - Italien

Bosch Rexroth S.p.A.
Via del Progresso, 16 (Zona Ind.)
35020 Padova

Tel.: +39 049 8 70 13 70
Fax: +39 049 8 70 13 77

Italy - Italien

Bosch Rexroth S.p.A.
Via Isonzo, 61
40033 Casalecchio di Reno (Bo)

Tel.: +39 051 29 86 430
Fax: +39 051 29 86 490

Italy - Italien

Tightening & Press-fit:
TEMA S.p.A. Automazione
Via Juker, 28
20025 Legnano

Tel.: +39 0 331 4671

Netherlands – Niederlande/Holland

Bosch Rexroth B.V.
Kruisbroeksestraat 1
(P.O. Box 32)
5281 RV Boxtel

Tel.: +31 (0) 411 65 16 40
Fax: +31 (0) 411 65 14 83
www.boschrexroth.nl

Netherlands - Niederlande/Holland

Bosch Rexroth Services B.V.
Technical Services
Kruisbroeksestraat 1
(P.O. Box 32)
5281 RV Boxtel
Tel.: +31 (0) 411 65 19 51
Fax: +31 (0) 411 67 78 14
Hotline: +31 (0) 411 65 19 51
services@boschrexroth.nl

Norway - Norwegen

Bosch Rexroth AS
Electric Drives & Controls
Berghagan 1 or: Box 3007
1405 Ski-Langhus 1402 Ski
Tel.: +47 64 86 41 00

Fax: +47 64 86 90 62

Hotline: +47 64 86 94 82
jul.ruud@rexroth.no

Spain – Spanien

Goimendi Automation S.L.
Parque Empresarial Zuatzu
C/ Francisco Grandmontagne no.2
20018 San Sebastian

Tel.: +34 9 43 31 84 21
- service: +34 9 43 31 84 56
Fax: +34 9 43 31 84 27
- service: +34 9 43 31 84 60
sat.indramat@goimendi.es

Spain - Spanien

Bosch Rexroth S.A.
Electric Drives & Controls
Centro Industrial Santiga
Obradors 14-16
08130 Santa Perpetua de Mogoda
Barcelona
Tel.: +34 9 37 47 94-00
Fax: +34 9 37 47 94-01

Spain - Spanien

Bosch Rexroth S.A.
Electric Drives & Controls
c/ Almazara, 9
28760 Tres Cantos (Madrid)

Tel.: +34 91 806 24 79
Fax: +34 91 806 24 72
fernando.bariego@boschrexroth.es

Sweden - Schweden

Bosch Rexroth AB
Electric Drives & Controls
- Varuvägen 7
(Service: Konsumentvägen 4, Älfsjö)
125 81 Stockholm

Tel.: +46 (0) 8 727 92 00
Fax: +46 (0) 8 647 32 77

Sweden - Schweden

Bosch Rexroth AB
Electric Drives & Controls
Ekvändan 7
254 67 Helsingborg
Tel.: +46 (0) 4 238 88 -50
Fax: +46 (0) 4 238 88 -74

Switzerland East - Schweiz Ost

Bosch Rexroth Schweiz AG
Electric Drives & Controls
Hemrietstrasse 2
8863 Buttikon
Tel. +41 (0) 55 46 46 111
Fax +41 (0) 55 46 46 222

Switzerland West - Schweiz West

Bosch Rexroth Suisse SA
Av. Général Guisan 26
1800 Vevey 1

Tel.: +41 (0)21 632 84 20
Fax: +41 (0)21 632 84 21

Europa (Ost) - Europe (East)

vom Ausland: (0) nach Landeskennziffer weglassen
from abroad: don't dial (0) after country code

Czech Republic - Tschechien	Czech Republic - Tschechien	Hungary - Ungarn	Poland – Polen
Bosch -Rexroth, spol.s.r.o. Hviezdoslavova 5 627 00 Brno Tel.: +420 (0)5 48 126 358 Fax: +420 (0)5 48 126 112	**Tightening & Press-fit:** Bosch -Rexroth, spol.s.r.o. Stetkova 18 140 68 Praha 4 Tel.: +420 (0)241 406 675	Bosch Rexroth Kft. Angol utca 34 1149 Budapest Tel.: +36 (1) 422 3200 Fax: +36 (1) 422 3201	Bosch Rexroth Sp.zo.o. ul. Staszica 1 05-800 Pruszków Tel.: +48 (0) 22 738 18 00 – service: +48 (0) 22 738 18 46 Fax: +48 (0) 22 758 87 35 – service: +48 (0) 22 738 18 42
Poland – Polen	**Romania - Rumänien**	**Romania - Rumänien**	**Russia - Russland**
Bosch Rexroth Sp.zo.o. Biuro Poznan ul. Dabrowskiego 81/85 60-529 Poznan Tel.: +48 061 847 64 62 /-63 Fax: +48 061 847 64 02	East Electric S.R.L. Bdul Basarabia no.250, sector 3 73429 Bucuresti Tel./Fax:: +40 (0)21 255 35 07 +40 (0)21 255 77 13 Fax: +40 (0)21 725 61 21 eastel@rdsnet.ro	Bosch Rexroth Sp.zo.o. Str. Drobety nr. 4-10, app. 14 70258 Bucuresti, Sector 2 Tel.: +40 (0)1 210 48 25 +40 (0)1 210 29 50 Fax: +40 (0)1 210 29 52	Bosch Rexroth OOO Tschschjolkowskoje Chaussee 100 105523 Moskau Tel.: +7-495-783 30 60 + 64 Fax: +7-495 783 30 68 + 69 brcschrexroth.ru
Turkey - Türkei	**Turkey - Türkei**	**Slowakia - Slowakei**	**Slowenia - Slowenien**
Bosch Rexroth Otomasyon San & Tic. A..S. Fevzi Cakmak Cad No. 3 34295 Sefaköy Istanbul Tel.: +90 212 411-13 00 Fax: +90 212 411-13 17 www.boschrexroth.com.tr	Servo Kontrol Ltd. Sti. Perpa Ticaret Merkezi B Blok Kat: 11 No: 1609 80270 Okmeydani-Istanbul Tel: +90 212 320 30 80 Fax: +90 212 320 30 81 remzi.sali@servokontrol.com www.servokontrol.com	**Tightening & Press-fit:** MTS, spol .s.r.o. 02755 Kriva 53 Tel.: +421 43 5819 161	DOMEL Otoki 21 64 228 Zelezniki Tel.: +386 5 5117 152 Fax: +386 5 5117 225 brane.ozebek@domel.si

Australien, Süd-Afrika - Australia, South Africa

Australia - Australien	Australia - Australien	South Africa - Südafrika	South Africa - Südafrika
AIMS - Australian Industrial Machinery Services Pty. Ltd. 28 Westside Drive Laverton North Vic 3026 Melbourne Tel.: +61 3 93 14 3321 Fax: +61 3 93 14 3329 **Hotlines: +61 3 93 14 3321** **+61 4 19 369 195** enquires@aimservices.com.au	Bosch Rexroth Pty. Ltd. No. 7, Endeavour Way Braeside Victoria, 31 95 Melbourne Tel.: +61 3 95 80 39 33 Fax: +61 3 95 80 17 33 mel@rexroth.com.au	TECTRA Automation (Pty) Ltd. 100 Newton Road, Meadowdale Edenvale 1609 Tel.: +27 11 971 94 00 Fax: +27 11 971 94 40 **Hotline: +27 82 903 29 23** georgv@tectra.co.za	**Tightening & Press-fit:** Jendamark Automation 76A York Road, North End 6000 Port Elizabeth Tel.: +27 41 391 4735

Asien - Asia (incl. Pacific Rim)

China

Shanghai Bosch Rexroth
Hydraulics & Automation Ltd.
No.122, Fu Te Dong Yi Road
Waigaoqiao, Free Trade Zone
Shanghai 200131 - P.R.China

Tel.: +86 21 58 66 30 30
Fax: +86 21 58 66 55 23
brcservice@boschrexroth.com.cn

China

Shanghai Bosch Rexroth
Hydraulics & Automation Ltd.
4/f, Marine Tower
No.1, Pudong Avenue
Shanghai 200120 - P.R.China

Tel: +86 21 68 86 15 88
Fax: +86 21 68 86 05 99
+86 21 58 40 65 77
y.wu@boschrexroth.com.cn

China

Bosch Rexroth (China) Ltd.
Satellite Service Office Changchun
Rm. 1910, Guangming Building
No.336 Xi'an Rd., Chao Yang Distr.
Changchun 130061 - P.R.China

Tel.+Fax: +86 431 898 1129
Mobile: +86 139 431 92 659
shuhong.wang@boschrexroth.com.cn

China

Bosch Rexroth (China) Ltd.
Satellite Service Office Wuhan
No. 22, Pinglanju, Milanyuan, Golden Harbor
No. 236 Longyang Avenue
Economic & Technology Development Zone
Wuhan 430056 - P.R.China

Tel.+Fax: +86 27 84 23 23 92
Mobile: +86 139 71 58 89 67
ym.zhu@boschrexroth.com.cn

China

Bosch Rexroth (China) Ltd.
Beijing Representative Office
Xi San Qi Dong, De Sheng Mei Wai
Hai Dian District
Beijing 100096, P.R.China

Tel.: +86 10 82 91 22 29
Fax: +86 10 82 91 81 09
yan.zhang@boschrexroth.com.cn

China

Bosch Rexroth (China) Ltd.
Guangzhou Repres. Office
Room 3710-3716, Metro Plaza,
Tian He District, 183 Tian He Bei Rd
Guangzhou 510075, P.R.China

Tel.: +86 20 87 55 00 30
+86 20 87 55 00 11
Fax: +86 20 87 55 23 87
andrew.wang@boschrexroth.com.cn

China

Bosch Rexroth (China) Ltd.
Dalian Representative Office
Room 2005,Pearl River Int. Building
No.99 Xin Kai Rd., Xi Gang District
Dalian, 116011, P.R.China

Tel.: +86 411 83 68 26 02
Fax: +86 411 83 68 27 02
jason.tan@boschrexroth.com.cn

China

Tightening & Press-fit:
C. Melchers GmbH & Co
Shanghai Representation
13 Floor Est Ocean Centre
No.588 Yanan Rd. East
65 Yanan Rd. West
Shanghai 200001
Tel.: +86 21 63 52 88 48
Fax: +86 21 63 51 31 38
shermanxia@sh.melchers.com.cn

Hongkong

Bosch Rexroth (China) Ltd.
6th Floor,
Yeung Yiu Chung No.6 Ind Bldg.
19 Cheung Shun Street
Cheung Sha Wan,
Kowloon, Hongkong

Tel.: +852 27 86 46 32
Fax: +852 27 42 60 57
Paul.li@boschrexroth.com.cn

India - Indien

Bosch Rexroth (India) Ltd.
Electric Drives & Controls
Plot. No.96, Phase III
Peenya Industrial Area
Bangalore – 560058

Tel.: +91 80 41 17 0211
Fax: +91 80 83 97 374
pradeep.rs@boschrexroth.co.in

India - Indien

Bosch Rexroth (India) Ltd.
Electric Drives & Controls
Advance House, II Floor
Ark Industrial Compound
Narol Naka, Makwana Road
Andheri (East), Mumbai - 400 059

Tel.: +91 22 28 56 32 90
+91 22 28 56 33 18
Fax: +91 22 28 56 32 93
singh.op@boschrexroth.co.in

India - Indien

Tightening & Press-fit:

MICO
Hosur Road Adugodi
560 030 Bangalore Karnataki

Tel.: +91 80 22 99 28 86

India - Indien

Bosch Rexroth (India) Ltd.
S-10, Green Park Extension
New Delhi – 110016

Tel.: +91 11 26 56 65 25
+91 11 26 56 65 27
Fax: +91 11 26 56 68 87
koul.rp@boschrexroth.co.in

Indonesia - Indonesien

PT. Bosch Rexroth
Building # 202, Cilandak Commercial Estate
Jl. Cilandak KKO, Jakarta 12560

Tel.: +62 21 7891169 (5 lines)
Fax: +62 21 7891170 - 71
rudy.karimun@boschrexroth.co.id

Japan

Bosch Rexroth Corporation
Service Center Japan
Yutakagaoka 1810, Meito-ku,
NAGOYA 465-0035, Japan

Tel.: +81 52 777 88 41
+81 52 777 88 53
+81 52 777 88 79
Fax: +81 52 777 89 01

Japan

Bosch Rexroth Corporation
Electric Drives & Controls
3-6-7 Shibuya, Shibuya-ku
Tokyo 150-0002, Japan

Tel : +81-3-54 85-7240
Fax: +81-3-54 85-7241

Korea

Bosch Rexroth-Korea Ltd.
Electric Drives & Controls
1515-14 Dadae-Dong, Saha-gu
Pusan Metropolitan City, 604-050

Tel.: +82 51 26 00 741
Fax: +82 51 26 00 747
eunkyong.kim@boschrexroth.co.kr

Korea

Bosch Rexroth-Korea Ltd.
Electric Drives and Controls
Bongwoo Bldg. 7FL, 31-7, 1Ga
Jangchoong-dong, Jung-gu
Seoul, 100-391

Tel.: +82 234 061 813
Fax: +82 222 641 295

Korea

Bosch Rexroth-Korea Ltd.
Electric Drives & Controls
1515-14 Dadae-Dong, Saha-gu
Ulsan, 680-010

Tel.: +82 52 256-0734
Fax: +82 52 256-0738
keonhyun.jeong@boschrexroth.co.kr

Korea

Tightening & Press-fit:
KVT Co., Ltd.
901, Daeryung Techno Town 8
481-11 Gasan-Dong
Geumcheon-Gu
Seoul, 153-775
Tel.: +82 2 2163 0231 9

Malaysia

Bosch Rexroth Sdn.Bhd.
11, Jalan U8/82, Seksyen U8
40150 Shah Alam
Selangor, Malaysia

Tel.: +60 3 78 44 80 00
Fax: +60 3 78 45 48 00
hhlim@boschrexroth.com.my
rexroth1@tm.net.my

Singapore - Singapur

Bosch Rexroth Pte Ltd
15D Tuas Road
Singapore 638520

Tel.: +65 68 61 87 33
Fax: +65 68 61 18 25
lai.ts@boschrexroth.com.sg

Taiwan

Bosch Rexroth Co., Ltd.
Taichung Industrial Area
No.19, 38 Road
Taichung, Taiwan 407, R.O.C.

Tel : +886 - 4 -235 08 383
Fax: +886 - 4 -235 08 586
jim.lin@boschrexroth.com.tw
david.lai@boschrexroth.com.tw

Taiwan

Bosch Rexroth Co., Ltd.
Tainan Branch
No. 17, Alley 24, Lane 737
Chung Cheng N.Rd. Yungkang
Tainan Hsien, Taiwan, R.O.C.

Tel : +886 - 6 –253 6565
Fax: +886 - 6 –253 4754
charlie.chen@boschrexroth.com.tw

Thailand

NC Advance Technology Co. Ltd.
59/76 Moo 9
Ramintra road 34
Tharang, Bangkhen,
Bangkok 10230

Tel.: +66 2 943 70 62
+66 2 943 71 21
Fax: +66 2 509 23 62
Hotline +66 1 984 61 52
sonkawin@hotmail.com

Nordamerika – North America

USA Headquarters - Hauptniederlassung Bosch Rexroth Corporation Electric Drives & Controls 5150 Prairie Stone Parkway Hoffman Estates, IL 60192-3707 Tel.: +1 847 645-3600 Fax: +1 847 645-6201 servicebrc@boschrexroth-us.com repairbrc@boschrexroth-us.com	USA Central Region - Mitte Bosch Rexroth Corporation Electric Drives & Controls 1701 Harmon Road Auburn Hills, MI 48326 Tel.: +1 248 393-3330 Fax: +1 248 393-2906	USA Southeast Region - Südost Bosch Rexroth Corporation Electric Drives & Controls 2810 Premiere Parkway, Suite 500 Duluth, GA 30097 Tel.: +1 678 957-4050 Fax: +1 678 417-6637	**USA SERVICE-HOTLINE** - 7 days week/ 24 hrs day - **+1-800-REXROTH** **+1 800 739 7684**
USA Northeast Region – Nordost Bosch Rexroth Corporation Electric Drives & Controls 99 Rainbow Road East Granby, CT 06026 Tel.: +1 860 844-8377 Fax: +1 860 844-8595	USA West Region – West Bosch Rexroth Corporation Electric Drives & Controls 7901 Stoneridge Drive, Suite 220 Pleasanton, CA 94588 Tel.: +1 925 227-1084 Fax: +1 925 227-1081	Canada East - Kanada Ost Bosch Rexroth Canada Corporation 5345 Outer Drive unit 5 Windsor, Ontario Canada N9A 6J3 Tel.: +1 519 737 7393 Fax.: +1 519 737 9999	Canada East - Kanada Ost Bosch Rexroth Canada Corporation Automation Division 3426 Mainway Drive Burlington, Ontario Canada L7M 1A8 Tel.: +1 905 335 5511 Fax: +1 905 335 4184 (Main) +1 905 335 9803 (Serv.) automation.service@boschrexroth.ca automation.repair@boschrexroth.ca
Canada West - Kanada West Bosch Rexroth Canada Corporation 5345 Goring St. Burnaby, British Columbia Canada V7J 1R1 Tel. +1 604 205 5777 Fax +1 604 205 6944 automation.service@boschrexroth.ca automation.repair@boschrexroth.ca	CANADA SERVICE HOTLINE - 7 days week/ 24 hrs day - **+1 905 335 5511**	Mexico Bosch Rexroth Mexico S.A. de C.V. Calle Neptuno 72 Unidad Ind. Vallejo 07700 Mexico, D.F. Tel.: +52 55 57 54 17 11 Fax: +52 55 57 54 50 73 mario.francioli@boschrexroth.com.mx	Mexico Bosch Rexroth S.A. de C.V. Calle Argentina No 3913 Fracc. las Torres 64930 Monterrey, N.L. Tel.: +52 81 83 49 80 91 +52 81 83 49 80 92 +52 81 83 49 80 93 Fax: +52 81 83 65 52 80

Südamerika – South America

Argentina - Argentinien Bosch Rexroth S.A.I.C. "The Drive & Control Company" Rosario 2302 B1606DLD Carapachay Provincia de Buenos Aires Tel.: +54 11 4756 01 40 +54 11 4756 02 40 +54 11 4756 03 40 +54 11 4756 04 40 Fax: +54 11 4756 01 36 +54 11 4721 91 53 victor.jabif@boschrexroth.com.ar	Argentina - Argentinien NAKASE SRL Servicio Tecnico CNC Calle 49, No. 5764/66 B1653AOX Villa Balester Provincia de Buenos Aires Tel.: +54 11 4768 42 42 Fax: +54 11 4768 42 42 111 **Hotline: +54 11 155 307 6781** nakase@usa.net nakase@nakase.com gerencia@nakase.com (Service)	Brazil - Brasilien Bosch Rexroth Ltda. Av. Tégula, 888 Ponte Alta, Atibaia SP CEP 12942-440 Tel.: +55 11 4414 -56 92 +55 11 4414 -56 84 Fax sales: +55 11 4414 -57 07 Fax serv.: +55 11 4414 -56 86 alexandre.wittwer@rexroth.com.br	Brazil - Brasilien Bosch Rexroth Ltda. R. Dr.Humberto Pinheiro Vieira, 100 Distrito Industrial [Caixa Postal 1273] 89220-390 Joinville - SC Tel./Fax: +55 47 473 58 33 Mobil: +55 47 9974 6645 sergio.prochnow@boschrexroth.com.br
Columbia - Kolumbien Reflutec de Colombia Ltda. Calle 37 No. 22-31 Santafé de Bogotá, D.C. Colombia Tel.: +57 1 208 65 55 Fax: +57 1 269 97 37 reflutec@etb.net.co			

Rexroth
Bosch Group

Bosch Rexroth AG
Electric Drives and Controls
P.O. Box 13 57
97803 Lohr, Germany
Bgm.-Dr.-Nebel-Str. 2
97816 Lohr, Germany
Phone +49 (0)93 52-40-50 60
Fax +49 (0)93 52-40-49 41
service.svc@boschrexroth.de
www.boschrexroth.com

R911315058

Printed in Germany
DOK-CONTRL-PLCOPENFB*D-AW01-EN-P