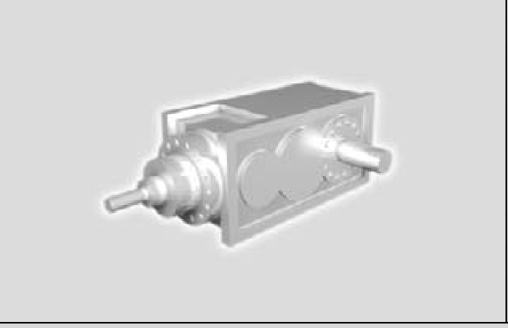
Gearmotors \ Industrial Gear Units \ Drive Electronics \ Drive Automation \ Services



MPLCMotion_MDX Library for MOVI-PLC®

FE330000

Edition 07/2005

11350415 / EN

Manual

SEW-EURODRIVE – Driving the world

1 Important Notes

⚡

- **This manual does not replace the detailed operating instructions!**
- **The MOVI-PLC® controller and the drives it controls may only be installed and started up by trained personnel observing the applicable accident prevention regulations, the MOVI-PLC® controller manual and the MOVIDRIVE® MDX60B/61B operating instructions.**

Documentation

- Read through this manual carefully before you start to install and startup drive inverters controlled by the MOVI-PLC® controller.
- This manual assumes that the user has access to and is familiar with the MOVIDRIVE® documentation, in particular the MOVIDRIVE® MDX60B/61B system manual.
- In this manual, cross references are marked with "→." For example, (→ Sec. X.X) means: Further information can be found in section X.X of this manual.
- A requirement of fault-free operation and fulfillment of any rights to claim under guarantee is that you observe the information in the documentation.

Bus systems

General safety notes for bus systems:

This communication system allows you to match the MOVIDRIVE® drive inverter to the specifics of your application. As with all bus systems, there is a danger of invisible, external (as far as the inverter is concerned) modifications to the parameters which give rise to changes in the inverter behavior. This may result in unexpected (not uncontrolled) system behavior.

Safety and warning notes

Always observe the safety and warning notes contained in this publication!

⚡	**Electrical hazard** Possible consequences: Severe or fatal injuries.
⚠	**Hazard** Possible consequences: Severe or fatal injuries.
!	**Hazardous situation** Possible consequences: Slight or minor injuries.
STOP	**Harmful situation** Possible consequences: Damage to the unit and the environment.
i	Tips and useful information.

2 Introduction

Content of the manual

This user manual describes the function modules of the `MPLCMotion_MDX` library and their application.

Description

MOVI-PLC® is a programmable logic controller designed in accordance with IEC 61131-3. One feature of the MOVI-PLC® controller is, for example, the DHP11B control card.

You can use the MOVI-PLC® controller, for example, as the control unit of a machine module. In this way, MOVI-PLC® controls all the drives within the machine module and in doing so takes off load from the master controller (e.g. machine or system PLC). In conjunction with a DOP operator terminal, MOVI-PLC® can also be used as a controller for entire machines.

The `MPLCMotion_MDX` library of the MOVI-PLC® controller described in this manual allows you to program the axis movements of connected MOVIDRIVE® MDX60B/61B drives simply and centrally.

Functions

The `MPLCMotion_MDX` library provides the following functions for each connected MOVIDRIVE® MDX60B/61B drive:

- Administrative functions
- Inverter operation (speed specification)
- Reference travel
- Positioning
- etc.

These functions are carried out decentrally in the drive inverters. The `MPLCMotion_MDX` library ensures fast communication with the drive inverters. It allows the motor axis movements to be programmed simply and centrally in the MOVI-PLC® controller.

Additional documentation

For simple and effective use of the `MPLCMotion_MDX` library, you should also order the following documentation:

- MOVI-PLC® programming manual
- MPLCInterface Library for MOVI-PLC® manual
- MOVIDRIVE® MDX61B Control Card MOVI-PLC® DHP11B manual
- MOVIDRIVE® MDX60/61B system manual

You must follow the instructions and safety notes published in these manuals when working with the drive system.

2.1 Application areas

The MPLCMotion_MDX.lib library is suitable for all application areas in which the MOVI-PLC® controller controls one or more drive inverters centrally.

Application examples

Typical application examples of the MPLCMotion_MDX.lib library:

- Machine modules
- Small machines
- System modules
- Storage and retrieval systems
- Hoist stations

Characteristics

The MPLCMotion_MDX.lib library has the following characteristics:

- The MOVI-PLC® controller can control up to twelve drives using the MPLCMotion_MDX.lib library.
- Users do not have to work with communication interfaces. Instead, they can operate the MOVI-PLC® controller using only motion and administration commands. This means that users do not have to be familiar with system bus communication and only require basic knowledge of the parameter settings of the drive inverter (e.g. for startup or setting the system bus address).
- The system bus enables fast communication between the MOVI-PLC® controller and the drive inverters.
- The MPLCMotion_MDX.lib library contains numerous function modules. Users can use these modules to program their own applications quickly and flexibly.
- PLCopen-compliant commands allow users to familiarize themselves with the program quickly and easily.

2.2 Overview of the MPLCMotion_MDX library

Required libraries

You require the following libraries to program the MOVI-PLC® controller in conjunction with MOVIDRIVE® MDX60B/61B drive inverters:

- MPLCDatatypes
- MPLCIntern
- MPLCSystem_ErrorHandling_Intern
- MPLCSystem_ConfigLibDataEx_Intern
- MPLCInterface_CAN

These libraries are inserted automatically in the PLC Editor of the MOVITOOLS® MotionStudio software when you install the MPLCMotion_MDX library.

The `MPLCMotion_MDX` library contains the following function modules, which are divided into several directories according to their functions:

MDX_Main

Directory `MDX_Main`:

- – MC_ConnectAxis_MDX
- – MC_Power_MDX
- – MC_Reset_MDX

MDX_SingleAxis

Directory `MDX_SingleAxis`:

- – MC_Home_MDX
- – MC_AxisStop_MDX
- – MC_Stop_MDX

- • Continuous motion function module:
  - – MC_MoveVelocity_MDX
- • Discrete motion function modules:
  - – MC_MoveAbsolute_MDX
  - – MC_MoveAbsoluteModulo_MDX
  - – MC_MoveRelative_MDX

MDX_SingleAxis Sensorless

Directory `MDX_SingleAxisSensorless`:

- – MC_StopSensorless_MDX

- • Continuous motion function module:
  - – MC_MoveVelocitySensorless_MDX

MDX_Inverter Parameters

Directory `MDX_InverterParameters`:

- – MC_ReadParameter_MDX
- – MC_WriteParameter_MDX
- – MC_SetDynamics_MDX
- – MC_SetLimiter_MDX
- – MC_SetJerk_MDX
- – MC_SetHomeParameters_MDX

MDX_Supplements

Directory `MDX_Supplements`:

- – MC_TouchProbe1_MDX
- – MC_TouchProbe2_MDX
- – MC_ReadActualPosition_MDX
- – MC_ReadAxisError_MDX
- – MC_ReadStatus_MDX

Note:

The `MPLCMotion_MDX.lib` library can be used with all the other libraries for the MOVI-PLC® controller at the same time.

However, fault-free operation can only be ensured when you operate all drive inverters, which are controlled using the function modules of the `MPLCMotion_MDX.lib` library, on one or more system CAN buses on which no other CAN objects (e.g. I/O modules) are set up.

2.3 Overview of additional libraries for the MOVI-PLC® controller

In addition to the `MPLCMotion_MDX` library, you can install a number of other libraries in the PLC Editor of the MOVITOOLS® MotionStudio software to optimize the control of the drive and frequency inverters provided by SEW as well as other periphery modules.

The following is an overview of the basic libraries available for controlling units that are connected to the MOVI-PLC® controller. In addition to these libraries, further application-specific libraries are available e.g. for handling, cams, synchronous operation, winder applications, etc. depending on the inverter type.

MPLCProcessdata	**MPLCMotion_MDX**	**MPLCMotion_MC07**	**MPLCMotion_MX**	**MPLCMotion_MM**	**MPLCUtilities**
MOVI-PLC® can be used as a conventional controller • Controls all SEW inverters via process data • Uses process data profiles, application modules or your own IPOS® programs	MOVI-PLC® as motion controller for MOVIDRIVE® B • Single-axis motion commands • Uses MOVIDRIVE® B interfaces	MOVI-PLC® as motion controller for MOVIDRIVE® 07 • Speed commands • Uses MOVITRAC® 07 interfaces	MOVI-PLC® as motion controller for MOVIAXIS® • Single-axis motion commands • Uses MOVIAXIS® interfaces	MOVI-PLC® as motion controller for MOVIMOT® • Speed commands • Uses MOVIMOT® interfaces	To connect e.g. CANopen I/O modules
↓	↓	↓	↓	↓	↓
SEW process data modules	MOVIDRIVE® MDX	MOVITRAC® 07	MOVIAXIS®	MOVIMOT®	CANopen IO modules
Elements that have to be installed in the control configuration of the PLC Editor to be able to use the libraries					
Overview of the inverter/unit-specific motion libraries and input/output libraries					

3 Project Planning and Startup

This section describes the prerequisites for using the `MPLCMotion_MDX.lib` library and contains important information for project planning and startup.

3.1 Prerequisites

PC and software

An engineering PC and the MOVITOOLS® Motion Studio software are both required to program the MOVI-PLC® controller using the `MPLCMotion_MDX.lib` library. For more information on the PC and software requirements, refer to the MOVI-PLC® programming manual.

Drive inverters

Note:

The MOVI-PLC® controller can only be used to control the MOVIDRIVE® MDX60B/61B drive inverters from firmware version 824 854 0.15 of the drive inverter basic unit.

The standard version of the MOVIDRIVE® MDX60B/61B drive inverters is sufficient to use the continuous motion function modules (MC_MoveVelocity_MDX, MC_MoveVelocitySensorless_MDX).

The application version of the MOVIDRIVE® MDX60B/61B drive inverters is required for function modules that position the motor axis (discrete motion function modules, MC_Home_MDX).

Control topology

You can connect twelve of the following drive/frequency inverters to the MOVI-PLC® controller:

- MOVIDRIVE® MDX60B/61B
- MOVITRAC® 07
- MOVIMOT®

Comply with the following prerequisites:

- Connect a maximum of six drive/frequency inverters to one system CAN bus.
  - When connecting between one and three drive inverters to one system CAN bus: Set the baud rate of the system CAN bus to ≥ 500 kBit/s.
  - When connecting between four and six drive inverters to one system CAN bus: Set the baud rate of the system CAN bus to 1000 kBit/s.

This manual contains information on controlling MOVIDRIVE® MDX60B/61B drive inverters.

For information on controlling frequency inverters, refer to the following manual:

- MPLCMotion_MMc / MPLCMotion_MC07 libraries for MOVI-PLC®

Important:

The technical characteristics given in this section only apply when no other CAN bus stations are active on the system CAN bus used to control the drive inverters.

Do not connect any other CAN bus stations to the system CAN bus on which drive inverters controlled by the `MPLCMotion_MDX.lib` library are connected.

3.2 Communication times

All MOVIDRIVE® MDX60B/61B drive inverters connected to the MOVI-PLC® controller send their current actual values to the MOVI-PLC® controller. The cycle time for transferring the actual values is dependent on the data profile and the encoder used, which are set in the control configuration of the module parameters of the drive inverter. Take the update times of the actual values of the MOVIDRIVE® MDX60B/61B drive inverters into account during project planning.

For a detailed description of the data profiles and the corresponding communication times, refer to the section "Function module MC_ConnectAxis_MDX" from page 20.

Drive inverters - SSI encoders

Using an SSI encoder slows access to all the parameters by a factor of five. Therefore, you should avoid using SSI encoder and use a Hiperface® encoder instead.

These characteristics affect the response time of the MOVI-PLC® controller in conjunction with the drive inverters. Take these characteristics into account during project planning.

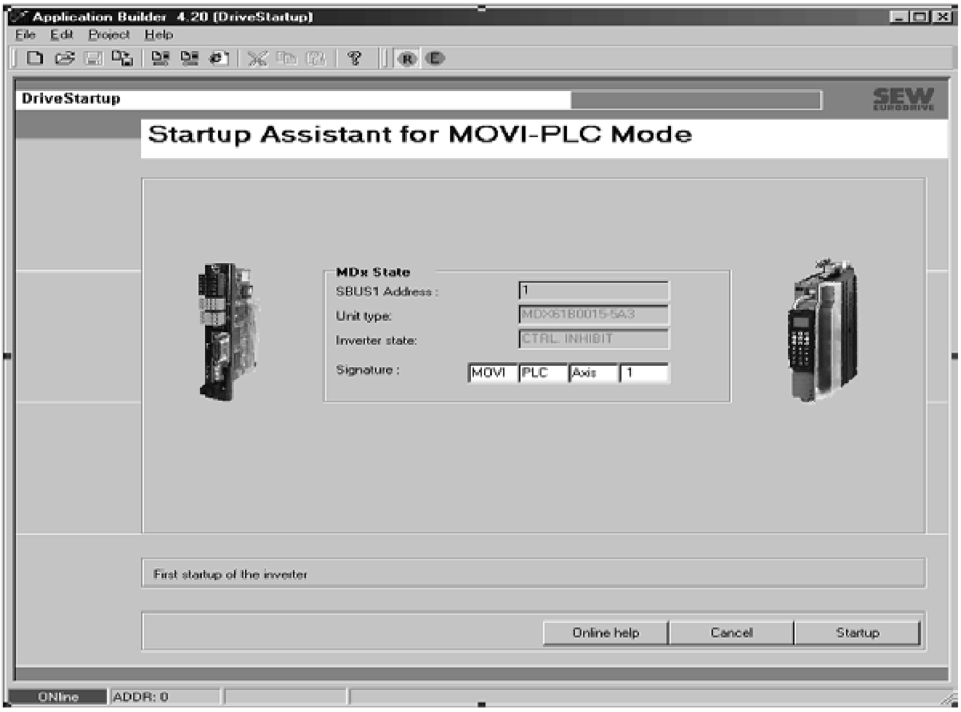
3.3 Startup

This section describes the startup process for the drive inverter when it is to be controlled by the MOVI-PLC® controller.

Warning:

Only start up the drive inverter using the startup assistant described in this section.

You must not change the drive inverter’s parameter settings manually or start up the drive inverter directly using the motor startup assistant either during initial startup or during subsequent restarts or optimizations. Manual changes could lead to unforeseeable operating states that could cause severe or fatal injuries to personnel.



MDx State

SBUS1 Address : 1

Unit type: MDX61B0015-5A3

Inverter state: CTRL. INHIBIT

Signature : MOVI PLC Axis 1

First startup of the inverter

20094AEN

To start the [DriveStartup MOVI-PLC Mode] startup assistant, open the context menu for the entry [MDX ...] in the unit tree of the MOVITOOLS® MotionStudio software.

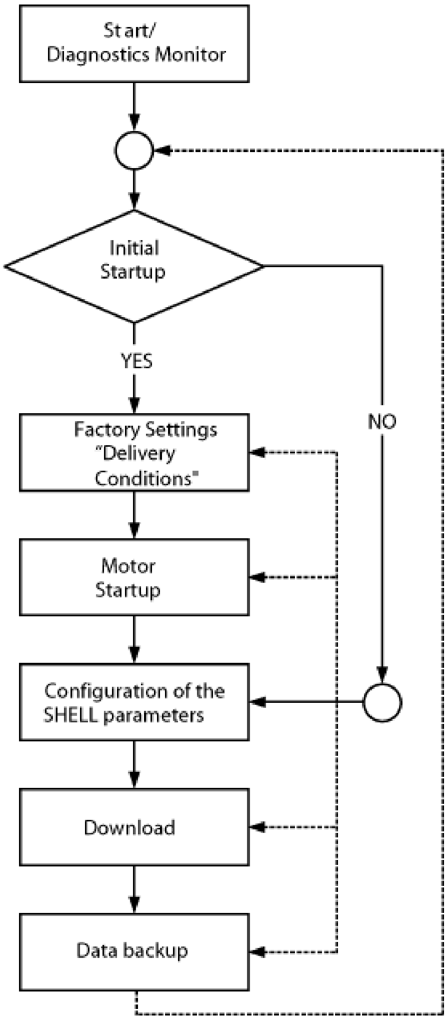
The startup assistant guides you step-by-step through the startup procedure:

1. Load the delivery status
2. Startup the drive inverter
3. Configure the Shell parameters
4. Download the input values
5. Save the inverter data

The startup assistant detects automatically whether you want to perform initial startup or a restart.

For the initial startup process, you must perform all five steps one after the other.

For a restart, the startup assistant goes directly to step 3. However, you can start with step 1 or 2 by clicking on the entries manually. This procedure allows you to use the startup assistant to optimize the data, for example for motor startup, at a later date.



20029AEN

Only perform startup using the startup assistant. Proceed as follows:

Step 1

Load the factory settings

The delivery status is loaded.
When the delivery status is loaded

- The startup data is reset
- All Shell parameters are reset to the default values
- All IPOS® variables are deleted
- Any IPOS® program code is deleted

↓

Step 2

Startup the drive inverter

Follow the instructions of the startup assistant. For details on motor startup, refer to the MOVIDRIVE® MDX60B/61B system manual.

Note:
In the operating mode groups V/f and VFC, only function modules of the `MDX_SingleAxisSensorless` directory can be executed. One of the following operating mode groups has to be set for function modules in the `MDX_SingleAxis` directory:

- VFC n-control
- CFC control
- SERVO control

Within an operating mode group, the MOVI-PLC® controller sets the operating mode required for continuous or discrete motion function modules automatically (see also the detailed description on page 18).

↓

Step 3

Configure the Shell parameters

Follow the instructions of the startup assistant. You can either accept the default values by clicking [Apply proposals] or change the values as required.

Notes:

1. The SBus address set here must match the SBus address used for the MC_ConnectAxis_MDX function module and the address set in the control configuration of the PLC Editor (module parameter of the entry MOVIDRIVE MDX B). The baud rate here must match the baud rate set in the control configuration of the PLC Editor (module parameter of the entry CAN 1/2, default value 500 kBaud).
2. The MOVI-PLC® controller can read and use the digital inputs of the drive inverter basic unit or rather its option in the control program, independent of the parameter settings in the groups *P60x* or *P61x*. However, to prevent the digital input terminals from executing additional functions, the relevant parameters must be set to *IPOS Input* or *No Function*.
   To be able to use the digital outputs of the drive inverter basic unit or rather its option in the control program of the MOVI-PLC® controller, the corresponding parameters in the groups *P62x* or *P63x* have to be set to *IPOS Output*. If these parameters are not set to *IPOS Output*, the digital outputs will be written in the program, but the physical output signals will not be changed. The difference between the output variables and the physical output signal is not displayed in the control program.
   Some of these parameters are already set to the correct value on delivery.

↓

Step 4

Download the input values

You can use this function to load the relevant SHELL user data into the drive inverter.

↓

Step 5

Save the inverter data

A complete set of the drive inverter data is saved in the file [*.vd0].

Important:

During and after startup, you must not change the parameters in Shell. Changing parameters manually can lead to unforeseeable operating states.

Drive inverter - IPOS®

If the drive system is controlled using the `MPLCMotion_MDX.lib` library, users cannot program the IPOS® software in the connected drive inverters themselves.

3.4 Units and ranges of values

Units

The function modules of the `MPLCMotion_MDX` library used the following units for their input and output signals:

- Positions in increments [incr] (4096 increments correspond to a 360° revolution of the motor axis)
- Speed in revolutions / minute [1/min]
- Accelerations as ramp times in milliseconds to achieve a speed [ms] changed by 3000 1/min
- Jerk as time in milliseconds for the duration of the torque build-up [ms]

Ranges of values

For the function modules of the `MPLCMotion_MDX` library, the following maximum ranges of values are permitted. If the values of the input signals are outside of the setting ranges, the function modules issue error messages.

However, the drive inverter adjusts the travel tasks to these limit values automatically depending on the motor connected and the limit values set in the inverter parameters. These limit values can be lower than the maximum values that can be set in the function modules. In this case, the function modules do not issue error messages. As a result, lag errors can occur during positioning tasks.

- Positions: -(2^{31}) ... 2^{31} [incr]
- Speeds for positioning tasks: 0 ... 6000 [1/min]
- Speeds for speed-controlled travel tasks: -6000 ... 6000 [1/min]
- Accelerations for positioning tasks: 10 ... 20000 [ms]
- Accelerations for speed-controlled travel tasks: 0 ... 2000000 [ms]
- Jerk (not used for speed-controlled travel tasks): 5 ... 2000 [ms]

4 Description of the Function Modules

This section describes the functions and the behavior of the function modules in the `MPLCMotion_MDX` library.

4.1 General functionality of the function modules

This section describes the basic functionality of the inputs and outputs of the function modules and other general features of the MOVI-PLC® controller and drive inverters. For specific examples of the interaction and sequence of several function modules, including time diagrams, refer to Sec. 5 "Programming Examples" (→ from page 70).

Input signal Enable

There are different methods to activate two types of function modules:

- Function modules that are activated by the input signal *Enable*.
- Function modules that are activated by the input signal *Execute*.

Function modules that are activated by the input signal *Enable* typically perform cyclical actions (e.g. MC_ReadActualPosition_MDX).

- When the input signal *Enable* is set to *TRUE*,
  - the function module is active.
  - the function module recalculates the output signals in each cycle.
- When the input signal *Enable* is set to *FALSE*,
  - the function module does not recalculate the output signals.
  - all the output signals remain at the value that was last calculated. (Exception: *Done*, *Busy* and *Error* are reset to *FALSE*.)

Therefore, the function module must verify the value of the output signals by setting the output signal *Done* to *TRUE*.

(The input *Enable* of the MC_ConnectAxis_MDX function modules differs from the behavior described here. For details, refer to the description of this function module.)

Input signal Execute

Function modules that are activated by the input signal *Execute* typically perform an action once (e.g. MC_ReadParameter_MDX).

A positive edge change at the *Execute* input starts the action.

The output signals remain in effect until the input signal *Execute* is reset to *FALSE* (falling edge). However, if the input signal *Execute* is reset to *FALSE* before the action is completed, the output signals remain in effect for at least one more control cycle after the action has been concluded.

If a rising edge occurs at the *Execute* input, the values of the input signals are adopted for the action. Changing the input signals during the action has no effect. Another rising edge signal is required at the *Execute* input to adopt the modified values.

Output signal Done

The function module sets the output signal *Done* or *InVelocity* to *TRUE* once the action of the function module has been executed successfully.

Some of the function modules display the conditions *"In Position"* (discrete motion tasks) or *"Speed reached"* (continuous motion tasks) of the drive inverter at the *Done* output signal. These function modules check the conditions until a falling edge occurs at the *Execute* input.

As long as the input signal *Execute* is set to *TRUE*, the function module resets the output signal *Done* to *FALSE* when the condition "In Position" or "Speed reached" is no longer fulfilled. Accordingly, the output signal *Done* is set to *TRUE* when the condition is fulfilled again. Consequently, when the input signal *Execute* is set to *TRUE*, the output signal *Done* can change between *TRUE* and *FALSE* several times.

Output signal Active

The output signal *Active* only exists for function modules that control a movement of the motor axis.

The function module sets the output signal *Active* to *TRUE* when the motor axis controlled by the function module turns to reach the target (target position or target speed).

The function module usually sets the output signal *Active* to *TRUE* shortly after a rising edge occurs at the input *Execute*.

If the motor axis is prevented from rotating due to the terminal assignment on the drive inverter (e.g. controller inhibit), the function module is executed, but the output signal *Active* is reset to *FALSE*.

The function module sets the output signal *Active* to *TRUE*, when:

- the terminal assignment on the drive inverter enables the motor axis to turn when the function module is executed
- and the motor axis moves until it reaches the target.

The function module resets the output signal *Active* to *FALSE* when one of the output signals *Done* or *InVelocity*, *Error* or *CommandAborted* is set to *TRUE*.

Output signal Busy

This output signal only exists for function modules that require several control cycles for execution and that do not control any movement of the motor axis.

The function module sets the output signal *Busy* to *TRUE* as long as the function module is executed.

The function module resets the output signal *Busy* to *FALSE* when one of the output signals *Done* or *Error* is set to *TRUE*.

Output signal Command-Aborted

This output signal only exists for function modules that control a movement of the motor axis.

The function module sets the output signal *CommandAborted* to *TRUE*, when its execution is cancelled

- By another function module
- Or by another instance of the same function module,

when both control the same motor axis.

This means that the cancelled task of the function module is no longer executed afterwards.

When the function module that was cancelled sets the output signal *CommandAborted* to *TRUE*, it resets the output signals *Done* or *InVelocity* and *Active* to *FALSE*. The function module resets the output signal *CommandAborted* to *FALSE* by setting a falling edge at the input signal *Execute*.

The following function modules can cancel motion function modules:

- MC_Stop_MDX / MC_AxisStop_MDX / MC_StopSensorless_MDX.
- MC_ConnectAxis_MDX (when the MOVI-PLC® controller detects an inverter error, communication error or 24 V operation of the drive inverter in this function module).
- MC_Power_MDX when *Enable* = *FALSE* (only for discrete motion tasks that have already reached the target position window once; other function modules are only interrupted, see below).
- Discrete motion function modules can only cancel discrete motion function modules.
- Continuous motion function modules can only cancel continuous motion function modules.

The function module resets the output signal *Done* to *FALSE* on cancellation, even if the goal of the function module has already been achieved and the specified position or speed window of the cancelled function module is still maintained.

Output signal Error

If an error occurs in the MOVI-PLC® controller during the execution of a function module, the function module sets the output signal *Error* to *TRUE*. In this case, the relevant error is displayed at the output signal *ErrorID*.

Errors in the drive inverter do not cause the output signal *Error* to be set, but are detected in the MC_ConnectAxis_MDX function module and cause the cancellation of the motion function module.

Response to Controller Inhibit, No Enable or Safe Stop

When one or more of the following conditions applies, the function module interrupts the active travel task (discrete motion, continuous motion, homing) of the motor axis:

- Controller inhibit
- No enable
- Safe stop

However, the function module does not cancel the travel task.

When the active travel task is interrupted, the function module resets the output signal *Active* to *FALSE*. However, it does not set the output signal *CommandAborted* to *TRUE*.

As soon as the above conditions no longer apply, the function module continues the interrupted travel task.

Interrupting the action of a motion function module does not lead to an error. If the action of a motion function module is to be cancelled without executing another function module, the task of one of the function modules MC_Stop_MDX, MC_AxisStop_MDX or MC_StopSensorless_MDX must be executed.

Exception:

The reference travel triggered by the function module MC_Home_MDX is cancelled by setting a controller inhibit or triggering a safe stop. At the same time, the function module MC_Home_MDX sets the output signal *CommandAborted* to *TRUE*.

When the enable is revoked, reference travel is only interrupted. When the process is enabled again, the motor continues the reference travel.

When a braking movement, triggered by one of the function modules MC_Stop_MDX, MC_AxisStop_MDX or MC_StopSensorless_MDX, is interrupted by a controller inhibit, no enable or safe stop, this braking movement does not continue at the speed it had pri-

or to the interruption once the interruption condition has been removed. The motor axis continues to decelerate during the interruption.

Response to 24 V operation

When 24 V operation is activated, the function module currently controlling the movement of the motor axis cancels the travel task. The function module sets the output signal *CommandAborted* to *TRUE*.

Changing between discrete motion / continuous motion

The MOVIDRIVE® drive inverter implements continuous motion function blocks in one of the following operating modes:

- VFC+n control
- CFC control
- SERVO control

The MOVIDRIVE® drive inverter implements discrete motion function blocks in one of the following operating modes:

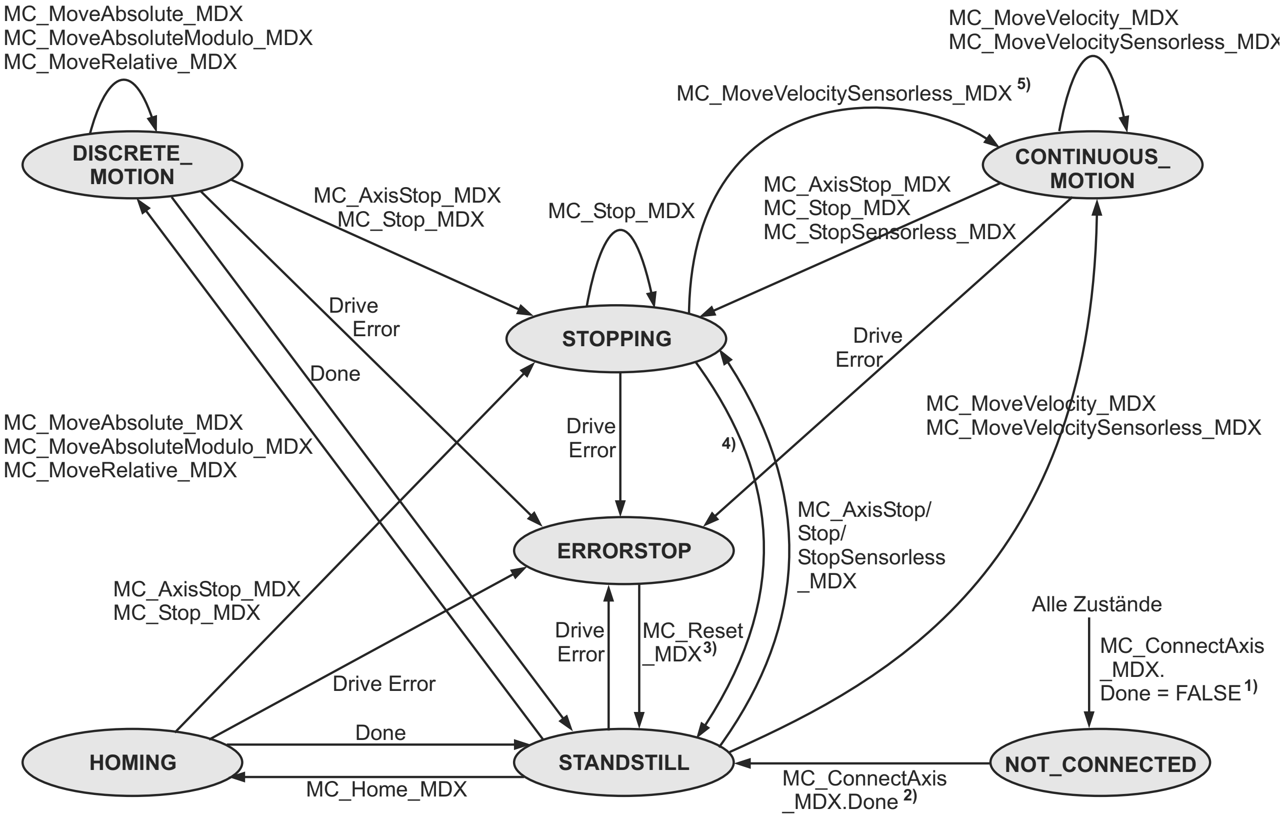
- VFC+n control + IPOS® positioning
- CFC control + IPOS® positioning
- SERVO control + IPOS® positioning

The MOVIDRIVE® drive inverter does not allow an on-the-fly changeover (that is, without a controller inhibit) in all operating modes. However, standard behavior for all operating modes is required.

Therefore, changeover is not possible when the motor axis is rotating. In this case, a corresponding error message is issued at the function module that would require the changeover to execute its travel task.

4.2 State diagram

In accordance with the execution of the function modules in the MPLCMotion_MDX library, the MOVI-PLC® controller is always in a defined state. The current state can be determined at any time at the output *PLCopenState* of the function module MC_ConnectAxis_MDX or at the outputs of the function module MC_ReadStatus_MDX. The following diagram shows which function modules can be executed in the various states and the state transitions that they cause.



20095AEN

1) MC_ConnectAxis_MDX.*Done* = *FALSE* when there is a communication error between the MOVI-PLC® controller and the drive inverter.

2) MC_ConnectAxis_MDX must be called in each control cycle and, therefore, in each state.

3) MC_Reset_MDX can be called in each state, but it only has an effect in the state ERRORSTOP.

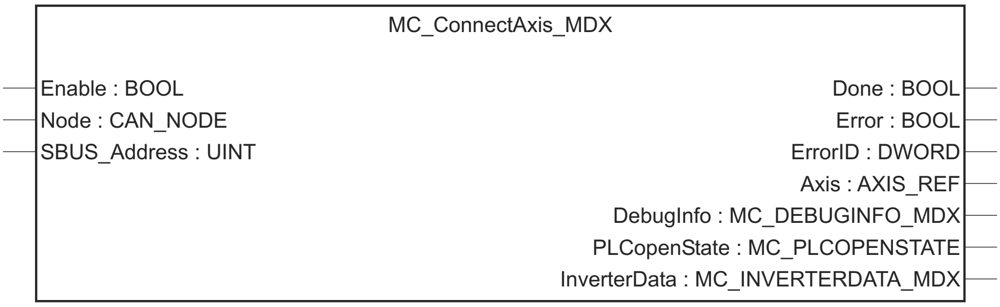
4) Prerequisite: MC_AxisStop_MDX.*Execute* / MC_Stop_MDX.*Execute* / MC_StopSensorless_MDX.Execute = *FALSE*

5) Prerequisite: MC_StopSensorless_MDX.*Execute* = *FALSE*

4.3 Directory MDX_Main

The directory `MDX_Main` includes function modules for managing communication between the MOVI-PLC® controller and the MOVIDRIVE® MDX60B/61B drive inverters. This directory also includes function modules for managing the drive inverter (e.g. enable, reset).

4.3.1 Function module MC_ConnectAxis_MDX



20001AXX

Description

The function module MC_ConnectAxis_MDX establishes the connection with the controlled motor axis. It is called cyclically in the program and creates a process image of the motor axes at the place where it is called.

Note:

The process input or process output image is not created at the start and end of the control cycle.

Instead, the process image for the inputs and outputs is created when the function module MC_ConnectAxis_MDX is called.

Application

The function module MC_ConnectAxis_MDX can be used on all motor axes.

To ensure that communication between the MOVI-PLC® controller and the MOVIDRIVE® MDX60B/61B drive inverters does not breakup, a different instance must be called cyclically for each motor axis.

Note:

If the function module MC_ConnectAxis_MDX is not called in one cycle of the control program, the drive inverter issues the communication error F46 and changes to the state "No enable".

Settings

The function module MC_ConnectAxis_MDX specifies which interface the MOVI-PLC® controller (system bus CAN 1/2) is connected to and the system bus address that is connected to the controlled drive inverter via this function module.

Prerequisites

The following prerequisites must be fulfilled to ensure that communication can be established between the MOVI-PLC® controller and a MOVIDRIVE® MDX60B/61B drive inverter via the function module MC_ConnectAxis_MDX.

- The system bus address set in the function module MC_ConnectAxis_MDX must match the address set in the inverter parameters:
  - P881 for SBUS 1, connector X12
  - P891 for SBUS 2, DFC11B
- The baud rate set in the control configuration must match the baud rate set in the inverter parameters.
  - P884 for SBUS 1, connector X12
  - P894 for SBUS 2, DFC11B

The startup assistant helps you to set these parameters.

Return values

The function module MC_ConnectAxis_MDX provides the user with various pieces of information, including (for details, see the "Outputs" table):

- Reference of the motor axis (logical address of the motor axis). This address is used to access other function modules on the motor axis.
- Debugging information
  - State of the system bus
  - Capacity utilization of the parameter channel
  - etc.
- Current state (→ Sec. 4.2, page 19)
- Inverter data
  - Inverter state
  - Actual position of the motor axis
  - Actual speed of the motor axis
  - etc.

Cancellation

When either an inverter error, 24V operation or a communication error occurs, the motion function module that is currently controlling the motor axis cancels the travel task.

The motion function module resets the output signal *Active* or *Done* / *InVelocity* to *FALSE*. As long as the input signal *Execute* is set to *TRUE*, the motion function module sets *CommandAborted* to *TRUE*.

Effect

The following parameters of the MOVIDRIVE® drive inverter are changed once when the MOVI-PLC® controller is connected to the MOVIDRIVE® drive inverter via the function module MC_ConnectAxis_MDX.

Parameter	Description	Value Entered
P630 ... P637	Binary outputs of the option DIO11B or DIP11B	IPOS® output signal
P885	Sync ID	128
P888	5 ms synchronization	On
P876	PO data	Position high or Position low

After these parameters have been downloaded, the MOVI-PLC® controller loads an IPOS® program required for communication once.

The following parameters of the MOVIDRIVE® drive inverter are changed each time the power supply is switched on or when the MOVI-PLC® controller is reset during the initialization of the function module MC_ConnectAxis_MDX (first call with *Enable* = *TRUE*):

Parameter	Description	
P941	Source actual position	Dependent on data profile
P938 ... P939	IPOS execution speed tasks	Optimized for the process

Inputs

The function module MC_ConnectAxis_MDX has the following inputs:

Input	Type	Description
Enable	BOOL	This input is used to activate the function module. It acts differently to the general description of the input *Enable* given in section 4.1 (from page 15) . • *TRUE*: When the function module is called for the first time by setting *Enable* = *TRUE* , the function module is activated and the applied input signals are adopted. In all the following cycles, the MOVI-PLC® controller has to continue to call the function module to prevent communication errors between the MOVI-PLC® controller and the MOVIDRIVE® drive inverters in *OperationMode NORMAL* (setting in the control configuration). However, regardless of the status of the *Enable* input, changes to the input signals are no longer adopted in the following cycles. The function module can only be reactivated to use the modified input signals after the MOVI-PLC® controller has been reset. • *FALSE*: The task of the function module will not be executed after the MOVI-PLC® controller has been switched on or reset as long as the *Enable* input is set to *FALSE*. However, if the *Enable* input has been set to *TRUE* once, the *Enable* input no longer has any function.
Node	CAN_NODE	This input is used to set the CAN bus node to which the drive inverter is connected. • SBUS_NODE_1: CAN 1 (connector X33 at DHP11B) • SBUS_NODE_2: CAN 2 (connector X32 at DHP11B)
SBUS_Address	UINT	This input is used to enter the system bus address of the connected drive inverter.

Note:

- To transfer the input signals *Node* or *SBUS_Address*, you have to trigger a "Reset" on the MOVI-PLC® controller ([PLC Editor], menu item [Online] / [Reset]).

Outputs

The function module MC_ConnectAxis_MDX has the following outputs:

Output	Type	Description
Done	BOOL	This output shows whether the initialization of the function module has been completed. • *TRUE*: The function module MC_ConnectAxis_MDX has completed initialization and set up communication with the drive inverter. • *FALSE: Initialization has not yet been completed.*
Error	BOOL	This output shows whether an error as occurred. • *TRUE*: An error has occured during the execution of the function module. • *FALSE: No error has occurred.*
*Error*ID	DWORD	This output displays the error code of the error (→ page 68).
Axis	AXIS_REF	This output displays the axis reference. The output signal must be transmitted to all function modules that refer to this physical motor axis.
DebugInfo	MC_DEBUGINFO_MDX	This output is used to locate any errors in the Debugging operation mode (setting in the control configuration in the module parameters of the drive inverter). • *ParameterChannelIdleTime: This value displays the availability of the parameter channel in %. A maximum of one parameter telegram is processed in each control cycle per motor axis.* 0 % → One parameter telegram in each MOVI-PLC® cycle 100 % → No parameter telegram • *PDOxxTransmitCount: Number of PDOxx sent from the MOVI-PLC® controller to the MOVIDRIVE® MDX60B/61B drive inverter.* • *PDOxxTransmitTimeStamp: Time stamp of the PDOxx sent from the MOVI-PLC® controller.* • *PDOxxReceiveCount: Number of PDOxx sent from the MOVIDRIVE® MDX60B/61B drive inverter to the MOVI-PLC® controller.* • *PDOxxReceiveTimeStamp: Time stamp of the PDOxx received by the MOVI-PLC® controller.* • *NumberOfReceiveErrors: Number of errors that occurred when receiving data.* • *NumberOfTransmitErrors: Number of errors that occurred when sending data.*
PLCopenState	MC_ PLCOPEN-STATE	This output displays information on the operating status of the motor axis (→ Sec. 4.2, page 19). • *NotConnected*: There is no communication betwee the MOVI-PLC® controller and the drive inverter. • *Standstill*: The target position window of a discrete motion task has been achieved or the task of the function module MC_AxisStop_MDX, MC_Stop_MDX or MC_StopSensorless_MDX has been completed. • *Errorstop:* There is an error in the drive inverter. The motor axis has been stopped using the emergency stop ramp. Exit the error state by triggering a reset. • *Stopping*: The motor axis is being decelerated because of: – a task of one of the function modules MC_AxisStop_MDX, MC_Stop_MDX or MC_StopSensorless_MDX or – the cancellation of the motion function module that was last active as an error was detected in the function module MC_ConnectAxis_MDX. The motor axis is assigned the state *Stopping* when: – a braking movement has already been completed and – the *Execute* input of the function module MC_Stop_MDX, MC_AxisStop_MDX or MC_StopSensorless_MDX is still set to *TRUE*. • *ContinuousMotion: The operating mode Speed control is active. The movement was started by the function module MC_MoveVelocity_MDX or MC_MoveVelocitySensorless_MDX.* • *Homing: The motor axis performs reference travel.* • *DiscreteMotion: The motor axis performs a positioning task.*

Output	Type	Description
InverterData	MC_INVERTERDATA_MDX	This output displays information on the operating status of the motor axis. • *Error:* An error has occurred in the drive inverter. • *Referenced*: The drive inverter is referenced. • *InverterStatus*: The status of the drive inverter corresponds to the 7-segment display on the drive inverter. • *FaultStatus*: Error number of the drive inverter in case of error • *ActualPosition: Current actual position of the motor axis in the drive inverter. The encoder used is set in the controller configuration*. (Unit: increments) • *ActualModuloPosition*: Current actual Modulo position of the drive inverter. The encoder used is set in the controller configuration. (Unit: increments) • *SetpointSpeed: Setpoint speed of the motor axis in the drive inverter* (Unit: 1/min) • *ActualSpeed: Actual speed of the motor axis in the drive inverter* (Unit: 1/min) • *ActualCurrent: Actual active current of the drive inverter* (Unit: % I_N) • *SoftwareLimitSwitchCW: Software limit switch CW (for clockwise rotation)* • *SoftwareLimitSwitchCCW: Software limit switch CCW (for counterclockwise rotation)*

Transferring MOVIDRIVE® actual values to the MOVI-PLC® controller

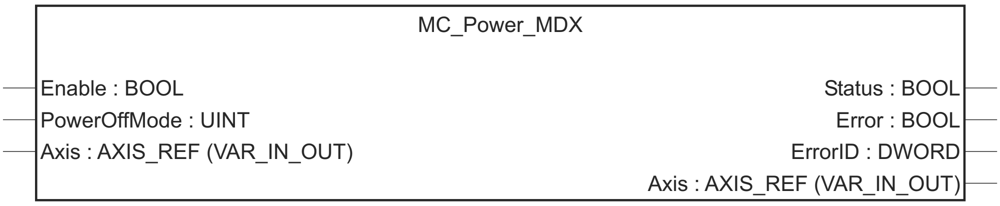
Various cyclic and acyclic process data objects are transferred between the MOVI-PLC® controller and the MOVIDRIVE® MDX60B/61B drive inverter via the CAN bus. Each connected MOVIDRIVE® MDX60B/61B drive inverter sends its actual values to the MOVI-PLC® controller.The type of actual values transferred and the cycle time it takes to transfer the actual values depends on the data profile and the encoder used. These settings are made in the controller configuration of the module parameters on the drive inverter. The following table gives an overview of the MOVIDRIVE® actual values transferred and the cycle times required to send the actual values to the MOVI-PLC® controller.

	"Data profile" →	**1**	**1**	**2**	**3**	**3**
	"Encoder type" →	**X15, X14**	**SSI**	**X15, X14 SSI**	**X15, X14**	**SSI**
MDX actual values ↓	**"Optimized refresh time"[1]** ↓					
Status word	During change	X	X	X	X	X
MDX inputs	During change	X	X	X	X	X
Touch probe positions	During change	X	X	X	X	X
Set speed	2 ms	X	-	-	-	-
	3 ms	-	X	X	-	-
	10 ms	-	-	-	-	-
Actual speed	2 ms	X	-	-	X	-
	3 ms	-	X	X	-	X
	10 ms	-	-	-	-	-
Actual position	2 ms	X	-	-	X	-
	3 ms	-	X	X	-	X
	10 ms	-	-	-	-	-
Modulo Actual position	2 ms	-	-	-	-	-
	3 ms	-	-	X	-	-
	10 ms	-	-	-	-	-
Active current	2 ms	-	-	-	X	-
	3 ms	-	-	X	-	X
	10 ms	-	-	-	-	-
Analog inputs	2 ms	-	-	-	-	-
	3 ms	-	-	-	-	-
	10 ms	-	-	X[2]	-	-

[1] In addition to the "optimized refresh time" you can also set transmission rates of 5 ms, 10 ms, 20 ms, 30 ms, 40 ms, 50 ms and 100 ms.

[2] If you set the "Refresh time" to 5 ms, the analog inputs will still be transferred in a 10 ms cycle.

4.3.2 Function module MC_Power_MDX



20002AXX

Description

The function module MC_Power_MDX switches a drive inverter

- on (inverter status A or 5, depending on the selected operating mode)
- or off (inverter status 1 or 2, depending on the PowerOffMode set)

Application

The function module MC_Power_MDX can only be used on motor axes with encoders.

Note:

A power function module does not have to be called up for drives without encoders.

Interruption

If the *Enable* input is reset to *FALSE*, the function module MC_Power_MDX interrupts the motion function module that is currently controlling the motor axis. The output signal *Active* of the motion function module is reset to *FALSE*.

When the *Enable* input signal of the function module MC_Power_MDX is set to *TRUE* again, the motion function module continues its previous motion task.

Exception: If a discrete motion task has already been assigned the status *Standstill* when the input signal *Enable* is reset to *FALSE* , the function module cancels the discrete motion task and sets the output signal *CommandAborted* to *TRUE*.

Inputs

The function module MC_Power_MDX has the following inputs:

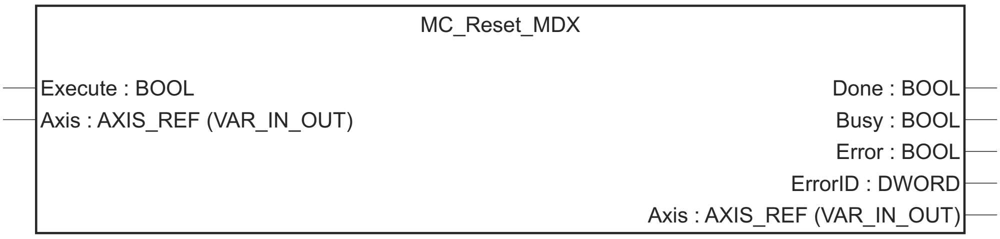
Input	Type	Description
Enable	BOOL	This input is used to switch the drive inverter on and off. The function module is executed with the current input values, even if the *Enable* input is set to *FALSE*. • *TRUE*: If the *Enable* input is set to *TRUE*, the drive inverter is switched on (inverter status 5 or A depending on the operating mode set). The first time that the function module MC_Power_MDX is used to switch on the drive inverter, the inverter is assigned the status "Position control" if a continuous motion task has not already been started. • *FALSE*: If the *Enable* input is set to *FALSE*, the drive inverter is switched off (inverter status 1 or 2 depending on the input *PowerOffMode*).
PowerOffMode	UINT	This input is used to set the status of drive inverter when it is switched off using the function module MC_Power_MDX. • *CTRL_INHIBIT*: The drive inverter changes to the status 'controller inhibit.' The motor axis is decelerated when the drive inverter is switched off (-> inverter status 1). • *NO_ENABLE*: The drive inverter changes to the status "No enable " (-> inverter status 2). Whether or not the brake is activated in inverter status 2 depends on the setting of the drive parameter *P730*. The brake function is switched on as the default setting, which means that the brake will be activated. In inverter status 1, the brake is always activated. This input is also used to switch between "Controller inhibit" and "No enable" when the drive inverter is switched off (*Enable* = *FALSE*)
Axis	AXIS_REF	This input specifies the motor axis on which the actions of the function module are to be executed.

Outputs

The function module MC_Power_MDX has the following outputs:

Output	Type	Description
Status	BOOL	This output shows whether the drive inverter is switched on or off. • *TRUE*: Drive inverter is switched on (inverter status 5 or A depending on the operating mode set) • *FALSE:* Drive inverter is switched off (inverter status 1 or 2 depending on the input PowerOffMode)
Error	BOOL	This output shows whether an error as occurred in the function module. • *TRUE*: An error has occured during the execution of the function module. • *FALSE:* No error has occurred in the function module.
*Error*ID	DWORD	This output displays the error code of the error (→ page 68).
Axis	AXIS_REF	This output displays the axis reference.

4.3.3 Function module MC_Reset_MDX



20003AXX

Description

The function module MC_Reset_MDX is used to acknowledge errors on the MOVIDRIVE® MDX60B/61B drive inverter.

Errors that occur in a function module of the `MPLCMotion_MDX` library are not acknowledged by this function module. These errors reset themselves automatically once the cause of the error has been removed.

Application

The function module MC_Reset_MDX can be used on all motor axes.

Inputs

The function module MC_Reset_MDX has the following inputs:

Input	Type	Description
Execute	BOOL	This input is used to acknowledge inverter errors. When a rising edge occurs at this input, the function module resets the inverter error.
Axis	AXIS_REF	This input specifies the motor axis on which the actions of the function module are to be executed.

Outputs

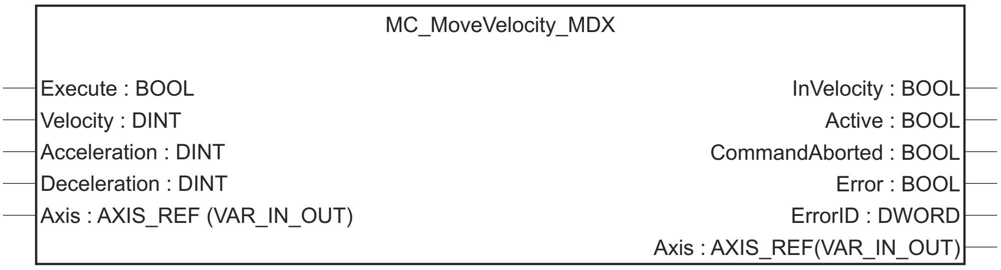
The function module MC_Reset_MDX has the following outputs:

Output	Type	Description
Done	BOOL	This output shows whether the reset has been performed. • *TRUE*: Reset successful. • *FALSE:* Reset unsuccessful.
Busy	BOOL	This output shows whether the inverter error has been reset. • *TRUE*: The function module is currently resetting the inverter error. • *FALSE:* The function module is not resetting the inverter error at present.
Error	BOOL	This output shows whether an error as occurred in the function module. • *TRUE*: An error has occurred during the execution of the function module. • *FALSE:* No error has occurred in the function module.
*Error*ID	DWORD	This output displays the error code of the error (→ page 68).
Axis	AXIS_REF	This output displays the axis reference.

4.4 Directory MDX_SingleAxis

The `MDX_SingleAxis` directory, contains function modules that enable motion tasks for a motor axis with encoder.

4.4.1 Function module MC_MoveVelocity_MDX



20010AXX

Description

The function module MC_MoveVelocity_MDX triggers continuous rotation of a motor axis.

- The inputs *Velocity*, *Acceleration* and *Deceleration* determine the dynamic behavior of the rotation.
- The drive inverter controls the speed of the motor axis specified by the input signal *Velocity* until the task of the function module MC_MoveVelocity_MDX is cancelled or interrupted (→ Sec. "General functionality of the function modules," page 15).

Application

The function module MC_MoveVelocity_MDX can only be used on motor axes with encoders.

Prerequisites

For the function module MC_MoveVelocity_MDX to be executed, the motor axis must be assigned the status *Standstill* or *ContinuousMotion* in *PLCopenState* (output signal of the function module MC_ConnectAxis_MDX).

Inputs

The function module MC_MoveVelocity_MDX has the following inputs:

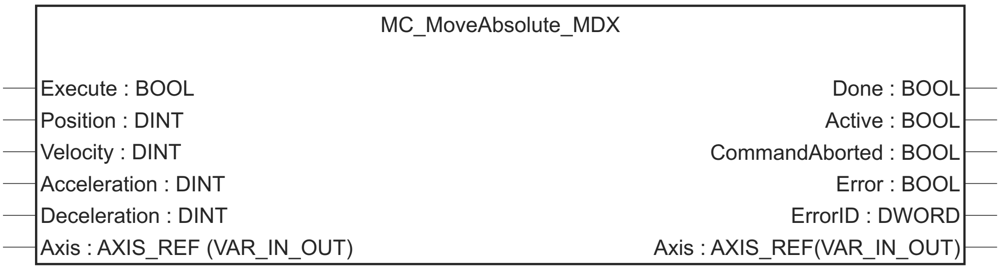
Input	Type	Description
Execute	BOOL	This input starts the task of the function module. When a rising edge occurs at this input, the function module starts the speed control.
Velocity	DINT	This input specifies the target setpoint speed of the motor rotation. (Unit: 1/min, maximum setting range -6000 ...6000)
Acceleration	DINT	This input specifies the ramp time for acceleration to a speed changed by 3000 1/min (increase of kinetic energy in the motor). (Unit: ms, maximum setting range: 0 ... 2000000)
Deceleration	DINT	This input specifies the ramp time for deceleration to a speed changed by 3000 1/min (reduction of kinetic energy in the motor). (Unit: ms, maximum setting range: 0 ... 2000000).
Axis	AXIS_REF	This input specifies the motor axis on which the actions of the function module are to be executed.

Outputs

The function module MC_MoveVelocity_MDX has the following outputs:

Output	Type	Description
InVelocity	BOOL	This output shows whether the motor axis turns at the target setpoint speed. • *TRUE*: The motor axis turns at the speed specified by the *Velocity* input. • *FALSE:* The motor axis is accelerated or decelerated to reach the target setpoint speed.
Active	BOOL	This output shows whether the motor axis is accelerated to reach the setpoint speed. • *TRUE*: The motor axis is accelerated or decelerated. • *FALSE:* The motor axis is not accelerated or decelerated.
Command Aborted	BOOL	This output shows whether the task of the function module has been cancelled. • *TRUE*: The task of the function module has been cancelled. • *FALSE:* The task of the function module has not been cancelled.
Error	BOOL	This output shows whether an error as occurred in the function module. • *TRUE*: An error has occurred in the function module. • *FALSE:* No error has occurred in the function module.
*Error*ID	DWORD	This output displays the error code of the error (→ page 68).
Axis	AXIS_REF	This output displays the axis reference.

4.4.2 Function module MC_MoveAbsolute_MDX



20005AXX

Description

The function module MC_MoveAbsolute_MDX starts the movement of a motor axis to an absolute axis position.

- The inputs *Velocity*, *Acceleration* and *Deceleration* determine the dynamic behavior of the rotation.
- The motor axis remains at the target position subject to position control.

Application

The function module MC_MoveAbsolute_MDX can only be used on motor axes with encoders.

Prerequisite

For the function module MC_MoveAbsolute_MDX to be executed, the motor axis must be assigned the status *Standstill* or *DiscreteMotion* in *PLCopenState* (output signal of the function module MC_ConnectAxis_MDX).

Effect

The following parameters of the MOVIDRIVE® MDX60B/61B drive inverter have a direct effect on the execution of the function module MC_MoveAbsolute_MDX.

- You can use parameters *P916 Ramp type* and *P933 Jerk time* to define positioning ramps with jerk limitation. Use function module MC_SetJerk_MDX to set these parameters.
- Use parameter *P922 Positioning window* to set the positioning window to the target position, within which the function module sets the output signal *Done* to *TRUE* "Target position reached."

The startup assistant helps you to set these parameters.

Inputs

The function module MC_MoveAbsolute_MDX has the following inputs:

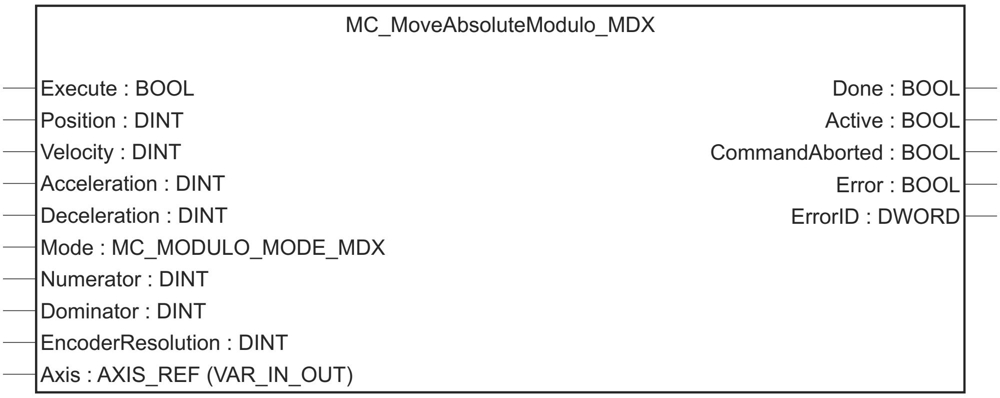
Input	Type	Description
Execute	BOOL	This input is used to start the rotation of the motor axis. When a rising edge occurs at this input, the function module starts the motor axis rotation.
Position	DINT	This input specifies the target position of the motor axis rotation. 4096 increments correspond to one motor axis rotation of 360°. (Unit: increments, maximum setting range: -(2^{31}) ... 2^{31})
Velocity	DINT	This input specifies the motor speed during the constant speed phase. (Unit: 1/ms, maximum setting range: 0 ... 6000)
Acceleration	DINT	This input specifies the ramp time for acceleration to a speed increased by 3000 1/min (increase of kinetic energy in the motor). (Unit: ms, maximum setting range: 10 .. 20000)
Deceleration	DINT	This input specifies the ramp time for deceleration to a speed lowered by 3000 1/min (reduction of kinetic energy in the motor). (Unit: ms, maximum setting range: 10 .. 20000)
Axis	AXIS_REF	This input specifies the motor axis on which the actions of the function module are to be executed.

Outputs

The function module MC_MoveAbsolute_MDX has the following outputs:

Output	Type	Description
Done	BOOL	This output shows whether the motor axis has been positioned. • *TRUE*: The motor axis has performed the rotation. The motor axis has reached the target position window. • *FALSE:* The motor axis has not completed the rotation. The motor axis has not yet reached the target position window.
Active	BOOL	This output shows whether the motor axis is turning to reach the target position window. • *TRUE*: The motor axis is rotating. • *FALSE:* The motor axis is not rotating.
Command Aborted	BOOL	This output shows whether another function module has cancelled the task of the function module. • *TRUE*: Another function module has cancelled the task of the function module. • *FALSE:* The task of the function module has not been cancelled.
Error	BOOL	This output shows whether an error as occurred in the function module. • *TRUE*: An error has occurred in the function module. • *FALSE:* No error has occurred in the function module.
*Error*ID	DWORD	This output displays the error code of the error (→ page 68).
Axis	AXIS_REF	This output displays the axis reference.

4.4.3 Function module MC_MoveAbsoluteModulo_MDX



20022AXX

Description

The function module MC_MoveAbsoluteModulo_MDX starts the movement of a motor axis to an absolute axis position. The function module specifies the position as a modulo position. A rotation of the motor axis of 360° corresponds to 2^{16}.

- The inputs *Velocity*, *Acceleration* and *Deceleration* determine the dynamic behavior of the rotation.
- The *Mode* input specifies the strategy of the task (short distance, clockwise, counter-clockwise).
- The inputs *Numerator*, *Denominator* and *EncoderResolution* are parameters of the connected drive.
- The motor axis remains at the target position subject to position control.

For more information on position specification and position management, refer to the IPOS® manual.

Note:

The CalcLCD function (Calculate Least Common Denominator) is available in the `MPLCUtlilities` library for calculating the least common denominator of two natural numbers.

Application

The function module MC_MoveAbsoluteModulo_MDX can only be used on motor axes with encoders.

Prerequisite

For the function module MC_MoveAbsoluteModulo_MDX to be executed, the motor axis must be assigned the status *Standstill* or *DiscreteMotion* in *PLCopenState* (output signal of the function module MC_ConnectAxis_MDX).

Effect

The following parameters of the MOVIDRIVE® MDX60B/61B drive inverter have a direct effect on the execution of the function module MC_MoveAbsoluteModulo_MDX.

- You can use parameters *P916 Ramp type* and *P933 Jerk time* to define positioning ramps with jerk limitation. Use function module MC_SetJerk_MDX to set these parameters.
- Use parameter *P922 Positioning window* to set the positioning window to the target position, within which the function module sets the output signal *Done* to *TRUE* "Target position reached."

The startup assistant helps you to set these parameters.

Inputs

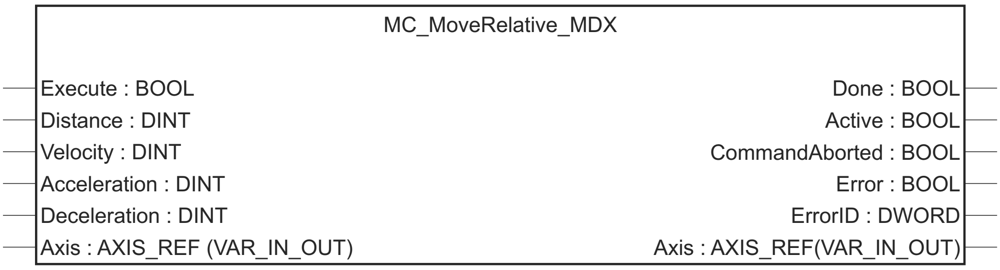
The function module MC_MoveAbsoluteModulo_MDX has the following inputs:

Input	Type	Description
Execute	BOOL	This input is used to start the rotation of the motor axis. When a rising edge occurs at this input, the function module starts the motor axis rotation.
Position	DINT	This input specifies the target position of the motor axis rotation. 2^{16} increments correspond to a motor axis rotation of 360°. (Unit: modulo increments - the number of complete rotations is given in the high word; the target angle between 0° and 360° in the low word). (Maximum setting range: dependent on the modulo numerator, denominator and encoder resolution, see the IPOS® manual).
Velocity	DINT	This input specifies the motor speed during the constant speed phase. (Unit: 1/ms, maximum setting range: 0 ... 6000)
Acceleration	DINT	This input specifies the ramp time for acceleration to a speed increased by 3000 1/min (increase of kinetic energy in the motor). (Unit: ms, maximum setting range: 10 .. 20000)
Deceleration	DINT	This input specifies the ramp time for deceleration to a speed lowered by 3000 1/min (reduction of kinetic energy in the motor). (Unit: ms, maximum setting range: 10 .. 20000)
mode	MC_ MODULO_ MODE_MDX	This input specifies the travel strategy. • SHORT: Short distance • CW: Clockwise rotation • CCW: Counterclockwise rotation
Numerator	DINT	Simulation of the gear unit by entering the number of teeth (Unit: number of teeth, maximum setting range: 1 ... 2^{31}).
Denominator	DINT	Simulation of the gear unit by entering the number of teeth (Unit: number of teeth, maximum setting range: 1 ... 2^{31}).
Encoder Resolution	DINT	This input specifies the resolution of the connected encoder system. (Unit: increments, maximum setting range: 1 ... 20000).
Axis	AXIS_REF	This input specifies the motor axis on which the actions of the function module are to be executed.

Outputs The function module MC_MoveAbsoluteModulo_MDX has the following outputs:

Output	Type	Description
Done	BOOL	This output shows whether the motor axis has been positioned. • *TRUE*: The motor axis has performed the rotation. The motor axis has reached the target position window. • *FALSE:* The motor axis has not completed the rotation. The motor axis has not yet reached the target position window.
Active	BOOL	This output shows whether the motor axis is turning to reach the target position window. • *TRUE*: The motor axis is rotating. • *FALSE:* The motor axis is not rotating.
Command Aborted	BOOL	This output shows whether another function module has cancelled the task of the function module. • *TRUE*: Another function module has cancelled the task of the function module. • *FALSE:* The task of the function module has not been cancelled.
Error	BOOL	This output shows whether an error as occurred in the function module. • *TRUE*: An error has occurred in the function module. • *FALSE:* No error has occurred in the function module.
*Error*ID	DWORD	This output displays the error code of the error (→ page 68).

4.4.4 Function module MC_MoveRelative_MDX



20006AXX

Description

This function module MC_MoveRelative_MDX is used to move the motor axis a specified distance.

- The inputs *Velocity*, *Acceleration* and *Deceleration* determine the dynamic behavior of the rotation.
- The motor axis remains at the target position subject to position control.

Application

The function module MC_MoveRelative_MDX can only be used on motor axes with encoders.

Prerequisite

For the function module MC_MoveRelative_MDX to be executed, the motor axis must be assigned the status *Standstill* or *DiscreteMotion* in *PLCopenState* (output signal of the function module MC_ConnectAxis_MDX).

Effect

The following parameters of the MOVIDRIVE® MDX60B/61B drive inverter have a direct effect on the execution of the function module MC_MoveRelative_MDX.

- You can use parameters *P916 Ramp type* and *P933 Jerk time* to define positioning ramps with jerk limitation. Use function module MC_SetJerk_MDX to set these parameters.
- Use parameter *P922 Positioning window* to set the positioning window to the target position, within which the function module sets the output signal *Done* to *TRUE* "Target position reached."

The startup assistant helps you to set these parameters.

Inputs

The function module MC_MoveRelative_MDX has the following inputs:

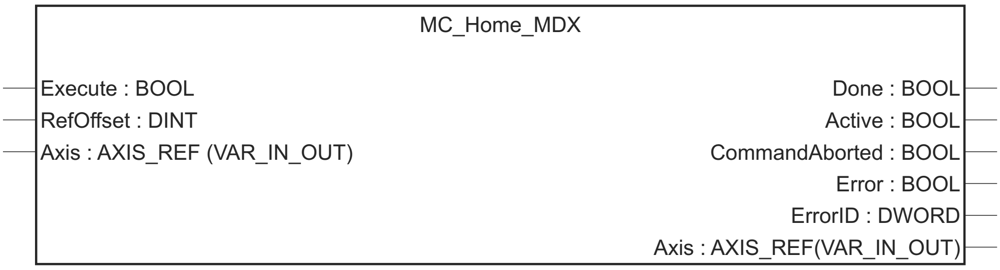
Input	Type	Description
Execute	BOOL	This input is used to start the rotation of the motor axis. When a rising edge occurs at this input, the function module starts the motor axis rotation.
Distance	DINT	This input specifies the target position of the motor axis rotation as a position relative to the current motor axis position when the function module starts. 4096 increments correspond to one motor axis rotation of 360°. (Unit: increments, maximum setting range: -(2^{31}) ... 2^{31})
Velocity	DINT	This input specifies the motor speed during the constant velocity phase. (Unit: 1/ms, maximum setting range: 0... 6000).
Acceleration	DINT	This input specifies the ramp time for acceleration to a speed increased by 3000 1/min (increase of kinetic energy in the motor). (Unit: ms, maximum setting range: 10 ... 20000).
Deceleration	DINT	This input specifies the ramp time for deceleration to a speed lowered by 3000 1/min (reduction of kinetic energy in the motor). (Unit: ms, maximum setting range: 10 ... 20000).
Axis	AXIS_REF	This input specifies the motor axis on which the actions of the function module are to be executed.

Outputs

The function module MC_MoveRelative_MDX has the following outputs:

Output	Type	Description
Done	BOOL	This output shows whether the motor axis has been positioned. • *TRUE*: The motor axis has performed the rotation. The motor axis has reached the target position window. • *FALSE:* The motor axis has not completed the rotation. The motor axis has not yet reached the target position window.
Active	BOOL	This output shows whether the motor axis is turning to reach the target position window. • *TRUE*: The motor axis is rotating. • *FALSE:* The motor axis is not rotating.
Command Aborted	BOOL	This output shows whether another function module has interrupted the task of the function module. • *TRUE*: Another function module has interrupted the task of the function module. • *FALSE:* The task of the function module has not been interrupted.
Error	BOOL	This output shows whether an error as occurred in the function module. • *TRUE*: An error has occurred in the function module. • *FALSE:* No error has occurred in the function module.
*Error*ID	DWORD	This output displays the error code of the error (→ page 68).
Axis	AXIS_REF	This output displays the axis reference.

4.4.5 Function module MC_Home_MDX



20007AXX

Description

The function module MC_Home_MDX triggers the reference travel of a motor axis.

After reference travel, the function module sets the position of the motor axis to the value specified by the *RefOffset* input.

Application

The function module MC_Home_MDX can only be used on motor axes with encoders.

Prerequisite

For the function module MC_Home_MDX to be executed, the motor axis must be assigned the status *Standstill* in *PLCopenState* (output signal of the function module MC_ConnectAxis_MDX).

Effect

In controller configuration, set the encoder to be used for reference travel in the module parameters of the drive inverter. Parameters *P900* to *P904* of the MOVIDRIVE® MDX60B/61B drive inverter specify the type of reference travel. You can change these parameters using the function module MC_SetHomeParameters_MDX.

Inputs

The function module MC_Home_MDX has the following inputs:

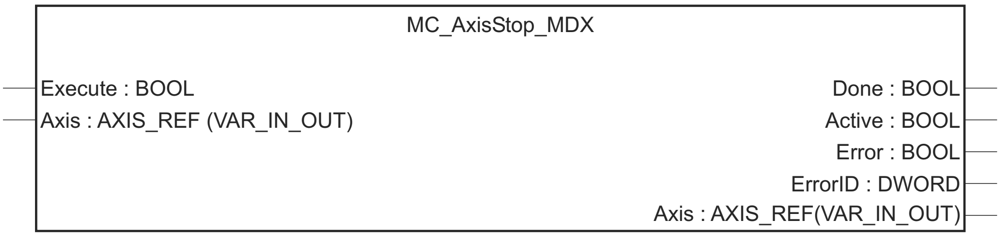
Input	Type	Description
Execute	BOOL	This input is used to start reference travel. When a rising edge occurs at this input, the function module starts reference travel.
RefOffset	DINT	This input specifies the actual position of the motor axis that is to be set after reference travel.
Axis	AXIS_REF	This input specifies the motor axis on which the actions of the function module are to be executed.

Outputs

The function module MC_Home_MDX has the following outputs:

Output	Type	Description
Done	BOOL	This output shows whether the reference travel of the motor axis has been completed. • *TRUE*: Reference travel of the motor axis is completed. The motor axis is at the position specified by the RefOffset input. • *FALSE:* Reference travel of the motor axis is not yet complete.
Active	BOOL	This output shows whether the motor axis is performing reference travel. • *TRUE*: The motor axis is performing reference travel. • *FALSE:* The motor axis is not performing reference travel.
Command Aborted	BOOL	This output shows whether another function module has interrupted reference travel. • *TRUE*: Another function module has interrupted reference travel. • *FALSE:* Reference travel has not been interrupted.
Error	BOOL	This output shows whether an error as occurred in the function module. • *TRUE*: An error has occurred in the function module. • *FALSE:* No error has occurred in the function module.
*Error*ID	DWORD	This output displays the error code of the error (→ page 68).
Axis	AXIS_REF	This output displays the axis reference.

4.4.6 Function module MC_AxisStop_MDX



20008AXX

Description

The function module MC_AxisStop_MDX terminates the current rotation of the motor axis. The motor axis is slowed with the deceleration specified at the start of the rotational movement (input *Deceleration* or the reference travel parameter).

Use the function module MC_AxisStop_MDX to interrupt motor axis rotations during standard processes.

Note:

The *PLCopenState* of the motor axis remains in the status *Stopping* as long as the *Execute* input is set to *TRUE*. This means a motion task for a subsequent movement cannot be performed.

To be able to leave the PLCopenState *Stopping*, the *Execute* input of the function module must be reset to *FALSE* in at least one call of the function module.

Application

The function module MC_AxisStop_MDX can only be used on motor axes with encoders. It cannot be used during the execution of interpolated movements using the function module MC_Interpolation_MDX.

Cancellation

Braking movements triggered by the function module MC_AxisStop_MDX can only be cancelled by:

- The function module MC_Stop_MDX (the input signal *Execute* must have already been reset to *FALSE*)
- The status "Controller inhibit"
- The status "No enable"
- The status "Safe stop"

When one of these statuses occurs, the motor axis is decelerated with the highest possible braking effect.

Inputs

The function module MC_AxisStop_MDX has the following inputs:

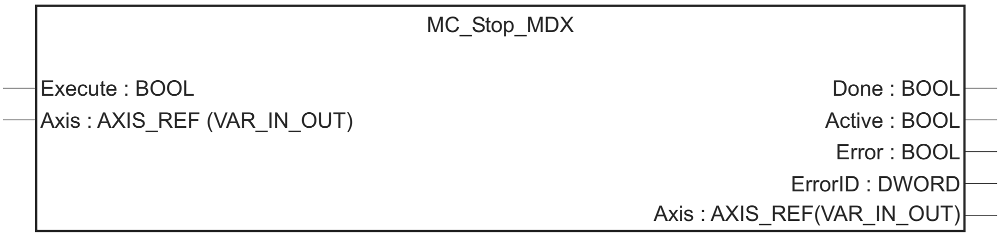
Input	Type	Description
Execute	BOOL	This input is used to start the braking process for the motor axis. When a rising edge occurs at this input, the function module starts to slow the motor axis.
Axis	AXIS_REF	This input specifies the motor axis on which the actions of the function module are to be executed.

Outputs

The function module MC_AxisStop_MDX has the following outputs:

Output	Type	Description
Done	BOOL	This output shows whether the braking process has been completed. • *TRUE*: The braking process of the motor axis is completed. The motor axis is at a standstill (position control). • *FALSE:* The braking process of the motor axis is not yet complete.
Active	BOOL	This output shows whether the motor axis is rotating. • *TRUE*: The motor axis is rotating. • *FALSE:* The motor axis is not rotating.
Error	BOOL	This output shows whether an error as occurred in the function module. • *TRUE*: An error has occurred in the function module. • *FALSE:* No error has occurred in the function module.
*Error*ID	DWORD	This output displays the error code of the error (→ page 68).
Axis	AXIS_REF	This output displays the axis reference.

4.4.7 Function module MC_Stop_MDX



20009AXX

Description

The function module MC_Stop_MDX slows the current rotational movement of the motor axis using the ramp set in the drive inverter.

Use function module MC_Stop_MDX to brake the motor axis with the highest possible deceleration.

Note:

The *PLCopenState* of the motor axis remains in the status *Stopping* as long as the *Execute* input is set to *TRUE*. This means a motion task for a subsequent movement is not possible.

To be able to leave the PLCopenState *Stopping*, the *Execute* input of the function module must be reset to *FALSE* in at least one call of the function module.

Application

The function module MC_Stop_MDX can only be used on motor axes with encoders. It cannot be used during the execution of interpolated movements using the function module MC_Interpolation_MDX.

Cancellation

The task of the function module MC_Stop_MDX is **not** cancelled by a task of different function modules.

Effect

The motor axis is slowed using the stop ramp set in parameter *P136* in the drive inverter.

The startup assistant helps you to set this parameter.

Inputs

The function module MC_Stop_MDX has the following inputs:

Input	Type	Description
Execute	BOOL	This input is used to start the braking process for the motor axis. When a rising edge occurs at this input, the function module starts to slow the motor axis.
Axis	AXIS_REF	This input specifies the motor axis on which the actions of the function module are to be executed.

Outputs

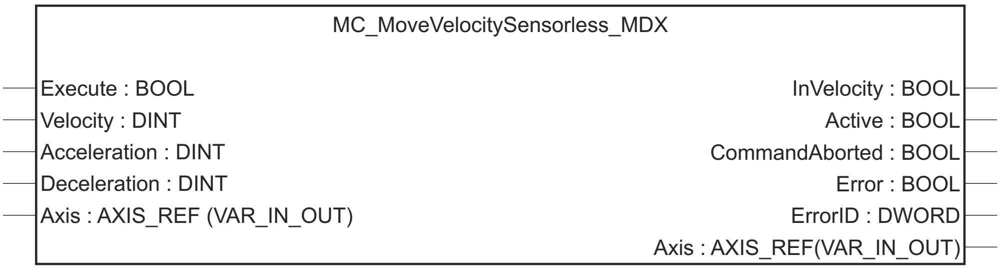
The function module MC_Stop_MDX has the following outputs:

Output	Type	Description
Done	BOOL	This output shows whether the braking process has been completed. • *TRUE*: The braking process of the motor axis is completed. The motor axis is at a standstill (position control). • *FALSE:* The braking process of the motor axis is not yet complete.
Active	BOOL	This output shows whether the motor axis is rotating. • *TRUE*: The motor axis is rotating. • *FALSE:* The motor axis is not rotating.
Error	BOOL	This output shows whether an error as occurred in the function module. • *TRUE*: An error has occurred in the function module. • *FALSE:* No error has occurred in the function module.
*Error*ID	DWORD	This output displays the error code of the error (→ page 68).
Axis	AXIS_REF	This output displays the axis reference.

4.5 Directory MDX_SingleAxisSensorless

The MDX_SingleAxisSensorless directory contains function modules that enable motion tasks for a motor axis without encoder.

4.5.1 Function module MC_MoveVelocitySensorless_MDX



20011AXX

Description

The function module MC_MoveVelocitySensorless_MDX triggers continuous rotation of a motor axis without encoder.

- The inputs *Velocity*, *Acceleration* and *Deceleration* determine the dynamic behavior of the rotation.
- The drive inverter controls the speed of the motor axis specified by the input signal *Velocity* until the task of the function module MC_MoveVelocitySensorless_MDX is cancelled or interrupted (→ Sec. "General functionality of the function modules,"page 15).

Application

The function module MC_MoveVelocitySensorless_MDX can only be used on motor axes without encoders.

Note:

In the speed control status, motor axes without encoders can only be operated at speeds higher than the minimum speed (drive parameter *P301*; however, not less than 15 1/min). In contrast to the process of function module MC_MoveVelocity_MDX, the function module MC_MoveVelocitySensorless_MDX rotates the motor axis at the minimum speed even when a speed of, for example, 0 is specified.

Prerequisite

For the function module MC_MoveVelocitySensorless_MDX to be executed, the motor axis must be assigned the status *Standstill*, *ContinuousMotion* or *Stopping* in *PLCopenState* (output signal of the function module MC_ConnectAxis_MDX).

Cancellation

The speed control triggered by the function module MC_MoveVelocitySensorless_MDX is cancelled by the function module MC_StopSensorless_MDX.

Note:

A rising edge at the input of either function module MC_Stop_MDX or MC_AxisStop_MDX does not cancel the task of the function module MC_MoveVelocitySensorless_MDX.

Inputs

The function module MC_MoveVelocitySensorless_MDX has the following inputs:

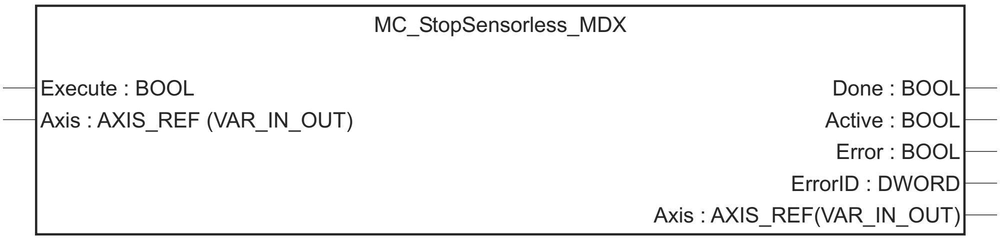
Input	Type	Description
Execute	BOOL	This input starts the task of the function module. When a rising edge occurs at this input, the function module starts the speed control.
Velocity	DINT	This input specifies the target setpoint speed of the motor axis rotation. (Unit: 1/ms, maximum setting range: -6000 ... 6000)
Acceleration	DINT	This input specifies the ramp time for acceleration to a speed increased by 3000 1/min (increase of kinetic energy in the motor). (Unit: ms, maximum setting range: 0 ... 2000000)
Deceleration	DINT	This input specifies the ramp time for deceleration to a speed lowered by 3000 1/min (reduction of kinetic energy in the motor). (Unit: ms, maximum setting range: 0 ... 2000000)
Axis	AXIS_REF	This input specifies the motor axis on which the actions of the function module are to be executed.

Outputs

The function module MC_MoveVelocitySensorless_MDX has the following outputs:

Output	Type	Description
InVelocity	BOOL	This output shows whether the motor axis turns at the target setpoint speed. • *TRUE*: The motor axis turns at the speed specified by the *Velocity* input. • *FALSE:* The motor axis is accelerated to reach the target setpoint speed.
Active	BOOL	This output shows whether the motor axis is accelerated to reach the setpoint speed. • *TRUE*: The motor axis is accelerated or decelerated. • *FALSE:* The motor axis is not accelerated or decelerated.
Command Aborted	BOOL	This output shows whether the task of the function module has been cancelled. • *TRUE*: The task of the function module has been cancelled. • *FALSE:* The task of the function module has not been cancelled.
Error	BOOL	This output shows whether an error as occurred in the function module. • *TRUE*: An error has occurred in the function module. • *FALSE:* No error has occurred in the function module.
*Error*ID	DWORD	This output displays the error code of the error (→ page 68).
Axis	AXIS_REF	This output displays the axis reference.

4.5.2 Function module MC_StopSensorless_MDX



20012AXX

Description

The function module MC_StopSensorless_MDX slows the current rotational movement of the motor axis using the ramp set in the drive inverter.

Note:

The *PLCopenState* of the motor axis remains in the status *Stopping* as long as the *Execute* input is set to *TRUE*. This means a motion task for a subsequent movement is not possible.

To be able to leave the PLCopenState *Stopping*, the *Execute* input of the function module must be reset to *FALSE* in at least one call of the function module.

Application

The function module MC_StopSensorless_MDX can only be used on motor axes without encoders.

Note:

In the speed control status, motor axes without encoders can only be operated at speeds higher than the minimum speed (drive parameter *P301*; however, not at less than 15 1/min). In contrast to the process of function module MC_AxisStop_MDX, the function module MC_StopSensorless_MDX applies the motor brake shortly before the motor axis comes to a standstill so that the motor axis is decelerated until it comes to a complete stop. Furthermore, the drive inverter switches to the status "No enable".

Cancellation

Braking movements triggered by the function module MC_StopSensorless_MDX can only be cancelled by:

- The status "Controller inhibit"
- The status "No enable"
- The status "Safe stop"

When one of these statuses occurs, the motor axis is decelerated with the highest possible braking effect.

Effect

The motor axis is slowed using the stop ramp set in parameter *P136* in the drive inverter.

The startup assistant helps you to set these parameters.

Note:

If a rising edge occurs at the *Execute* input of an instance of the function module MC_MoveVelocitySensorless_MDX, the task of the function module MC_StopSensorless_MDX is cancelled when the *Execute* input of the function module MC_StopSensorless_MDX is no longer set to *TRUE*.

The difference in behavior compared to the function modules MC_Stop_MDX and MC_AxisStop_MDX for motor axes with encoder is due to the fact that:

For motor axes without an encoder you have to use the function module MC_StopSensorless_MDX to slow the motor axis until it comes to a standstill. For motor axes with encoders, you can use the function module MC_MoveVelocity_MDX.

However, it should be possible to change the speed smoothly from zero to a different speed during a controlled speed change for motor axes both with and without encoders.

Inputs

The function module MC_StopSensorless_MDX has the following inputs:

Input	Type	Description
Execute	BOOL	This input is used to start the braking process for the motor axis. When a rising edge occurs at this input, the function module starts to slow the motor axis.
Axis	AXIS_REF	This input specifies the motor axis on which the actions of the function module are to be executed.

Outputs

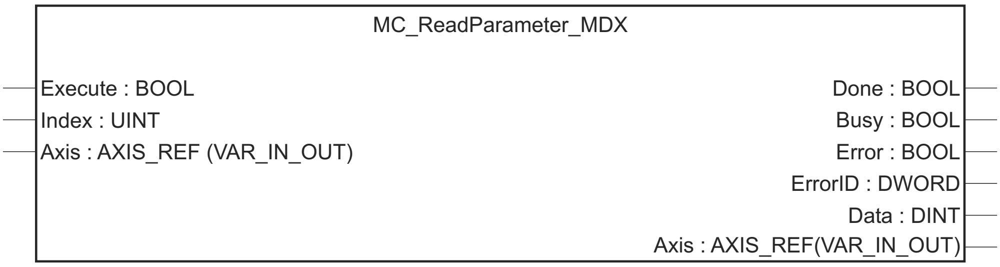
The function module MC_StopSensorless_MDX has the following outputs:

Output	Type	Description
Done	BOOL	This output shows whether the braking process has been completed. • *TRUE*: The braking process of the motor axis is completed. The motor axis is at a standstill (status of motor axis: "No enable"). • *FALSE:* The braking process of the motor axis is not yet complete.
Active	BOOL	This output shows whether the motor axis is rotating. • *TRUE*: The motor axis is rotating. • *FALSE:* The motor axis is not rotating.
Error	BOOL	This output shows whether an error as occurred in the function module. • *TRUE*: An error has occurred in the function module. • *FALSE:* No error has occurred in the function module.
*Error*ID	DWORD	This output displays the error code of the error (→ page 68).
Axis	AXIS_REF	This output displays the axis reference.

4.6 Directory MDX_InverterParameters

The `MDX_InverterParameters` directory contains function modules required to write and read parameters of the MOVIDRIVE® drive inverter.

4.6.1 Function module MC_ReadParameter_MDX



20016AXX

Description

The function module MC_ReadParameter_MDX is used to read parameters (indices) of the drive inverter into the MOVI-PLC® controller.

Application

The function module MC_ReadParameter_MDX can be used on all motor axes.

Inputs

The function module MC_ReadParameter_MDX has the following inputs:

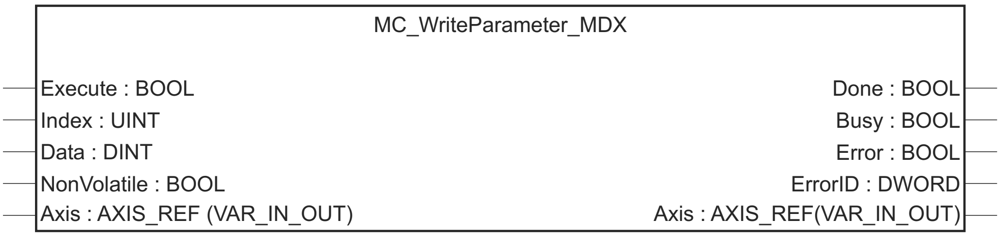
Input	Type	Description
Execute	BOOL	This input is used to start the read process for the parameter. When a rising edge occurs at this input, the function module starts to transfer the parameter from the drive inverter to the MOVI-PLC® controller.
Index	UINT	This input specifies which parameter is to be transferred from the drive inverter to the MOVI-PLC® controller. The index number of the parameter can be displayed in Shell of the MOVITOOLS® MotionStudio software. To do so, place your cursor in the relevant paramter field and press CTRL + F1.
Axis	AXIS_REF	This input specifies the motor axis on which the actions of the function module are to be executed.

Outputs

The function module MC_ReadParameter_MDX has the following outputs:

Output	Type	Description
Done	BOOL	This output shows whether the parameter has been transferred. • *TRUE*: The value of the parameter at the *Data* output is valid. • *FALSE:* The parameter has not been transferred.
Busy	BOOL	This output shows whether the parameter is being transferred. • *TRUE*: The parameter is being transferred. • *FALSE:* The parameter is not being transferred at present.
Error	BOOL	This output shows whether an error as occurred in the function module. • *TRUE*: An error has occurred during parameter transfer. • *FALSE:* No error has occurred.
*Error*ID	DWORD	This output displays the error code of the error (→ page 68).
Data	DINT	This output contains the transferred parameter value.
Axis	AXIS_REF	This output displays the axis reference.

4.6.2 Function module MC_WriteParameter_MDX



20017AXX

Description

The function module MC_WriteParameter_MDX is used to transfer parameters (indices) from the MOVI-PLC® controller to the drive inverter.

Application

The function module MC_WriteParameter_MDX can be used on all motor axes.

Important:

Changing some parameters, which require certain settings to ensure fault-free control of the drive inverter by the MOVI-PLC® controller, can lead to unforeseeable operating states. Therefore, you should only use this function module after consultation with SEW-EURODRIVE or after testing the required functionality thoroughly, ensuring the protection of personnel and machinery.

Inputs

The function module MC_WriteParameter_MDX has the following inputs:

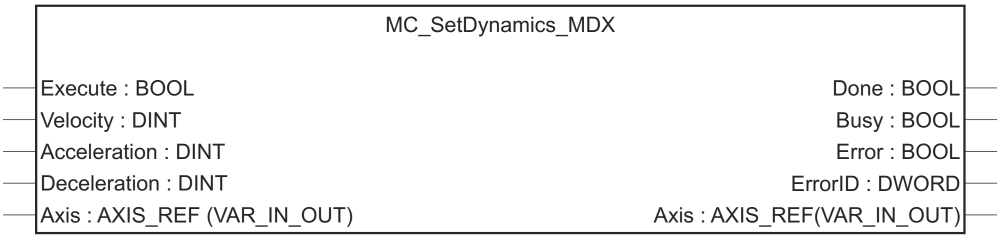
Input	Type	Description
Execute	BOOL	This input is used to start the parameter transfer process. When a rising edge occurs at this input, the function module starts to transfer the parameter from the MOVI-PLC® controller to the drive inverter.
Index	UINT	This input specifies which parameter is to be transferred from the MOVI-PLC® controller to the drive inverter. The index number of the parameter can be displayed in Shell of the MOVITOOLS® MotionStudio software. To do so, place your cursor in the relevant paramter field and press CTRL + F1.
Data	DINT	This input contains the transferred parameter value.
NonVolatile	BOOL	This input specifies whether the parameter should be copied to the non-volatile memory. • *TRUE*: The parameter is transferred to the **non-volatile** memory of the drive inverter. The parameter value is still stored once the drive inverter is switched off and then on again. • *FALSE:* The parameter is transferred to the volatile memory of the drive inverter. When the drive inverter is switched off and then back on again, the parameter is reset to its original value.
Axis	AXIS_REF	This input specifies the motor axis on which the actions of the function module are to be executed.

Outputs

The function module MC_WriteParameter_MDX has the following outputs:

Output	Type	Description
Done	BOOL	This output shows whether the parameter has been transferred. • *TRUE*: The parameter has been transferred from the MOVI-PLC® controller to the drive inverter. • *FALSE:* The parameter has not been transferred.
Busy	BOOL	This output shows whether the parameter is being transferred. • *TRUE*: The parameter is being transferred. • *FALSE:* The parameter is not being transferred at present.
Error	BOOL	This output shows whether an error as occurred in the function module. • *TRUE*: An error has occurred during parameter transfer. • *FALSE:* No error has occurred.
*Error*ID	DWORD	This output displays the error code of the error (→ page 68).
Axis	AXIS_REF	This output displays the axis reference.

4.6.3 Function module MC_SetDynamics_MDX



20023AXX

Description

The function module MC_SetDynamics_MDX is used to change the dynamic parameters while a task of one of the function modules MC_MoveAbsolute_MDX or MC_MoveRelative_MDX is being executed.

- The task of the function module MC_SetDynamics_MDX should only be started when the drive is executing a task of one of the function modules MC_MoveAbsolute_MDX or MC_MoveRelative_MDX.
- The inputs *Velocity*, *Acceleration* and *Deceleration* determine the dynamic behavior of the rotation.

Application

The function module MC_SetDynamics_MDX can be used on all motor axes.

Inputs

The function module MC_SetDynamics_MDX has the following inputs:

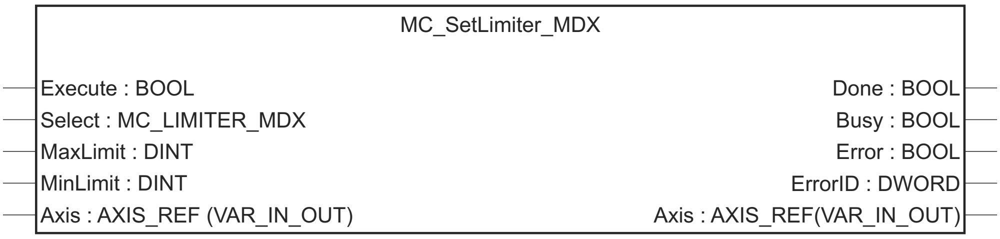
Input	Type	Description
Execute	BOOL	This input starts the task of the function module. When a rising edge occurs at this input, the function module changes the dynamic parameters of the motor rotation according to the inputs *Velocity*, *Acceleration* and *Deceleration*.
Velocity	DINT	This input specifies the new target setpoint speed of the motor rotation. (Unit: 1/min, maximum setting range -6000 ... 6000)
Acceleration	DINT	This input specifies the new ramp time for acceleration to a speed increased by 3000 1/min (increase of kinetic energy in the motor). (Unit: ms, maximum setting range: 0 ... 2000000)
Deceleration	DINT	This input specifies the new ramp time for deceleration to a speed lowered by 3000 1/min (reduction of kinetic energy in the motor). (Unit: ms, maximum setting range: 0 ... 2000000).
Axis	AXIS_REF	This input specifies the motor axis on which the actions of the function module are to be executed.

Outputs

The function module MC_SetDynamics_MDX has the following outputs:

Output	Type	Description
Done	BOOL	This output shows whether the motor axis rotates according to the new dynamic parameters *Velocity*, *Acceleration* and *Deceleration* . • *TRUE*: Motor axis rotates according to the new dynamic parameters. • *FALSE:* Motor axis rotates according to the previous dynamic parameters.
Active	BOOL	This output shows whether the new dynamic parameters are transferred to the drive inverter. • *TRUE*: New dynamic parameters are transferred to the drive inverter. • *FALSE:* No new dynamic parameters are transferred to the drive inverter.
Error	BOOL	This output shows whether an error as occurred in the function module. • *TRUE*: An error has occurred in the function module. • *FALSE:* No error has occurred in the function module.
*Error*ID	DWORD	This output displays the error code of the error (→ page 68).
Axis	AXIS_REF	This output displays the axis reference.

4.6.4 Function module MC_SetLimiter_MDX



20024AXX

Description

The function module MC_SetLimiter_MDX is used to transfer one of the following parameters from the MOVI-PLC® controller to the drive inverter (*P301* to *P303* parameter set 1):

- Minimum speed
- Maximum speed
- Current limit
- Torque limit

Application

The function module MC_SetLimiter_MDX can be used on all motor axes.

Inputs

The function module MC_SetLimiter_MDX has the following inputs:

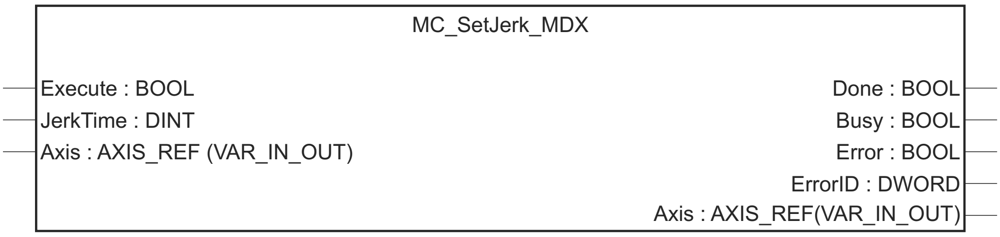
Input	Type	Description
Execute	BOOL	This input starts the task of the function module. When a rising edge occurs at this input, the function module transfers the parameter specified by the *Select* input.
Select	DINT	This input specifies which parameter the function module is to transfer from the MOVI-PLC® controller to the drive inverter. • *P301_P302_SPEED_LIMIT*: The function module transfers the parameters *Minimum speed* and *Maximum speed*. (Unit: 1/min, maximum setting range 0 ... 6100) • *P303_CURRENT_LIMIT*: The function module transfers the parameter *Current limit*. (Unit: % I_N, maximum setting range 0 ... 150 % (BG0 200%)) • *P304_TORQUE_LIMIT*: The function module transfers the parameter *Torque limit*. (Unit: % I_N, maximum setting range 0 ... 150 (BG0 200%)) • *P920_SW_LIMIT_SWITCH_CW*: The function module transfers the parameter *Software limit switch CW*. (Unit: increments, maximum setting range -(2^{31}-1) ... (2^{31}-1)) • *P921_SW_LIMIT_SWITCH_CCW*: The function module transfers the parameter *Software limit switch CCW*. (Unit: increments, maximum setting range -(2^{31}-1) ... (2^{31}-1))
MaxLimit	DINT	This input specifies the transferred parameter value. When the *Select* input is set to *P301_302_SPEED_LIMIT*, this input specifies the value for the parameter *P302* . (Unit and maximum setting range → see input *Select*)
MinLimit	DINT	This input is only used when the input *Select* is set to *P301_302_SPEED_LIMIT*. In this case, it specifies the value for parameter *P301*. (Unit: 1/min, maximum setting range 0 ... 6100)
Axis	AXIS_REF	This input specifies the motor axis on which the actions of the function module are to be executed.

Outputs

The function module MC_SetLimiter_MDX has the following outputs:

Output	Type	Description
Done	BOOL	This output shows whether the parameter has been transferred. • *TRUE*: The parameter has been transferred from the MOVI-PLC® controller to the drive inverter. • *FALSE:* The parameter has not been transferred.
Busy	BOOL	This output shows whether the parameter is being transferred. • *TRUE*: The parameter is being transferred. • *FALSE:* The parameter is not being transferred at present.
Error	BOOL	This output shows whether an error as occurred in the function module. • *TRUE*: An error has occurred during parameter transfer. • *FALSE:* No error has occurred.
*Error*ID	DWORD	This output displays the error code of the error (→ page 68).
Axis	AXIS_REF	This output displays the axis reference.

4.6.5 Function module MC_SetJerk_MDX



20019AXX

Description

The function module MC_SetJerk_MDX is used to activate the jerk limitation for positioning function modules. The function module MC_SetJerk_MDX transfers the jerk time to the drive inverter.

You cannot change the jerk time of the drive inverter while a travel task is being performed.

Application

The function module MC_SetJerk_MDX can only be used on motor axes with encoders.

Prerequisites

For the function module MC_SetJerk_MDX to be executed, the motor axis must be assigned the status *Standstill* in *PLCopenState* (output signal of the function module MC_ConnectAxis_MDX).

Inputs

The function module MC_SetJerk_MDX has the following inputs:

Input	Type	Description
Execute	BOOL	This input is used to start the jerk time transfer. When a rising edge occurs at this input, the function module starts to transfer the jerk time from the MOVI-PLC® controller to the drive inverter.
JerkTime	DINT	This input specifies the jerk time. (Unit: ms, maximum setting range: 0, 5 ... 2000) Note: The positioning process is longer than a positioning process using a linear ramp by approximately the jerk time.
Axis	AXIS_REF	This input specifies the motor axis on which the actions of the function module are to be executed.

Outputs

The function module MC_SetJerk_MDX has the following outputs:

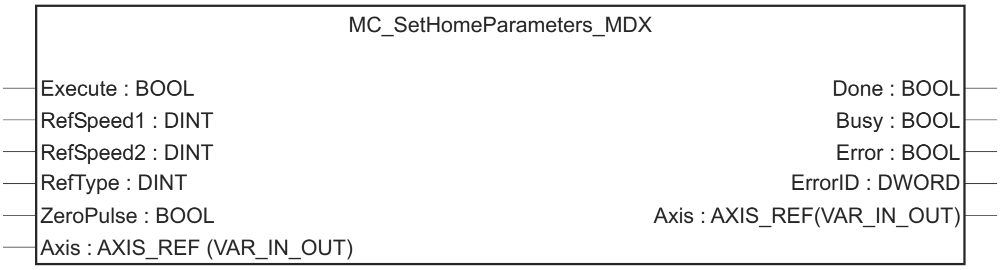
Output	Type	Description
Done	BOOL	This output shows whether the jerk time has been transferred. • *TRUE*: The jerk time has been transferred from the MOVI-PLC® controller to the drive inverter. • *FALSE: The jerk time has not been transferred.*
Busy	BOOL	This output shows whether the jerk time is being transferred. • *TRUE*: The jerk time is being transferred. • *FALSE: The jerk time is not being transferred at present.*
Error	BOOL	This output shows whether an error as occurred in the function module. • *TRUE*: An error has occurred during the transfer of the jerk time. • *FALSE: No error has occurred.*
*Error*ID	DWORD	This output displays the error code of the error (→ page 68).
Axis	AXIS_REF	This output displays the axis reference.

Note:

The function module MC_SetJerk_MDX changes the ramp type of the drive inverter from a linear ramp to a jerk-limited ramp. For speed-controlled travel tasks, the drive inverter automatically uses a linear ramp. When the drive inverter then executes a positioning task, it automatically switches back to the jerk-limited ramp.

If you set the *JerkTime* input to *0*, the function module MC_SetJerk_MDX switches the ramp type back to linear again.

4.6.6 Function module MC_SetHomeParameters_MDX



20018AXX

Description

The function module MC_SetHomeParameters_MDX is used to set the parameters that are required for the motor axis to perform reference travel. The function module MC_SetHomeParameters_MDX transfers parameters required for reference travel from the MOVI-PLC® controller to the drive inverter.

For more information on the reference travel parameters and the reference travel types, refer to the online help of the MOVITOOLS® MotionStudio software.

Application

The function module MC_SetHomeParameters_MDX can only be used on motor axes with encoders.

Inputs

The function module MC_SetHomeParameters_MDX has the following inputs:

Input	Type	Description
Execute	BOOL	This input is used to start the parameter transfer. When a rising edge occurs at this input, the function module starts to transfer the parameters from the MOVI-PLC® controller to the drive inverter.
RefSpeed1	DINT	This input specifies reference speed 1.
RefSpeed2	DINT	This input specifies reference speed 2.
RefType	DINT	This input defines the type of reference travel (0 ... 8).
ZeroPulse	BOOL	This input shows whether the motor axis is referenced to the zero pulse during reference travel. • *TRUE*: The motor axis is referenced to the zero pulse during reference travel. • *FALSE:* The motor axis is not referenced to the zero pulse during reference travel.
Axis	AXIS_REF	This input specifies the motor axis on which the actions of the function module are to be executed.

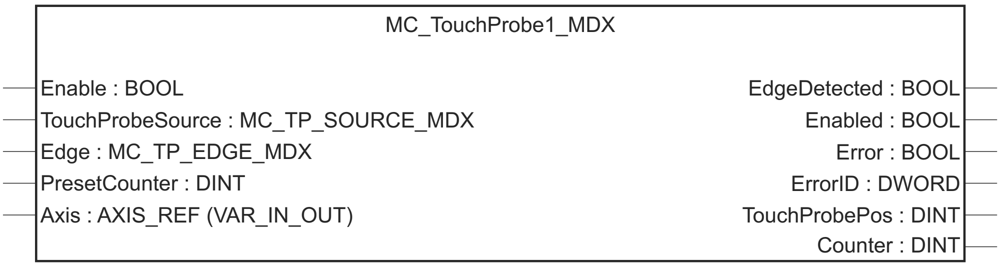
Outputs

The function module MC_SetHomeParameters_MDX has the following outputs:

Output	Type	Description
Done	BOOL	This output shows whether all parameters have been transferred. • *TRUE*: All parameters have been transferred from the MOV-PLC® controller to the drive inverter. • *FALSE:* The parameters have not been transferred.
Busy	BOOL	This output shows whether the parameters are being transferred. • *TRUE*: The parameters are being transferred. • *FALSE:* The parameters are not being transferred at present.
Error	BOOL	This output shows whether an error as occurred in the function module. • *TRUE*: An error has occurred during the transfer of the parameters. • *FALSE:* No error has occurred.
*Error*ID	DWORD	This output displays the error code of the error (→ page 68).
Axis	AXIS_REF	This output displays the axis reference.

4.7 Directory MDX_Supplements

4.7.1 Function module MC_TouchProbe1_MDX



20020AXX

Description

The function module MC_TouchProbe1_MDX controls a maximum of two touch probe interrupt routines.

- The function module copies the interrupt routines in the IPOS® code of the connected drive inverter. The drive inverter initializes a total of three acyclical send objects (control word 2, touch probe 1 position, touch probe 2 position).
- When the *Enable* input signal of the function module is set to *TRUE*, the function module sets up a receive object. When the function module detects the touch probe event, it reads the actual position from the drive inverter.

TouchProbe Routine

1. Calling the function module
   - The function module MC_TouchProbe1_MDX is called by setting the *Enable* input signal to *TRUE*.
   - The drive inverter initializes the IPOS® state automatically and starts a touch probe interrupt routine.
     (Edge change and touch probe selection)
2. Waiting for a touch probe event
   - The function module MC_TouchProbe1_MDX sets the *Enabled* output to *TRUE*.
3. The touch probe event has occurred.
   - The drive inverter signals to the MOVI-PLC® controller that the touch probe event has occurred.
     (CtrlWord2, TouchProbe1 -> Bit 0, TouchProbe2 -> Bit 1)
   - The function module MC_TouchProbe1_MDX reads in the touch probe position of the selected encoder system.
   - The function module MC_TouchProbe1_MDX increases the value of the *Counter* output by one.
   - The function module MC_TouchProbe1_MDX sets the *EdgeDetected* output to *TRUE*.
   - Monitoring of the touch probe input is started again.
4. Evaluating the touch probe event
   - The function module MC_TouchProbe1_MDX transfers the touch probe positions to the outputs *TouchProbePos_X14* to TouchProbePos_X62.
   - The function module resets the output *EdgeDetected* to *FALSE*.
   - The function module activates the touch probe interrupt routine.

Application

The function module MC_TouchProbe1_MDX can be used on all motor axes.

Inputs

The function module MC_TouchProbe1_MDX has the following inputs:

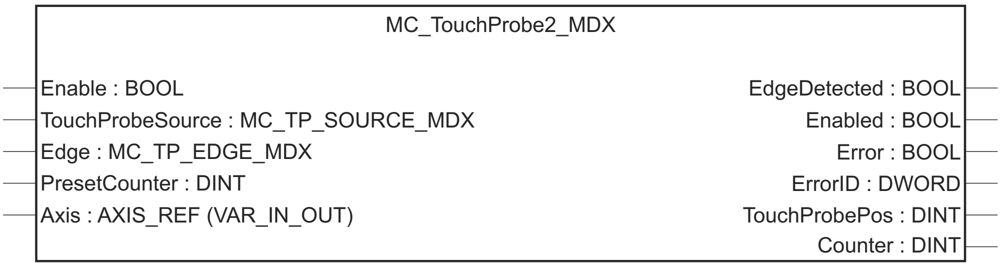
Input	Type	Description
Enable	BOOL	This input is used to activate the function module MC_TouchProbe1_MDX. The function module is only executed when the *Enable* input is set to *TRUE*. The values of the other inputs are only read in when a rising edge occurs at the *Enable* input.
TouchProbe Source	MC_TP_ SOURCE _MDX	This input specifies the encoder system. This setting is made independent of the setting in the controller configuration. • *X15*: A motor encoder measures the position. • *X14:* An external encoder measures the position. • *SSI:* An absolute encoder measures the position (DIP11B).
Edge	MC_TP_ EDGE_MDX	This input defines the edge evaluation of the touch probe input: • *EN*: Both edges • *EN_HI:* Rising edge • *EN_LO:* Falling edge
PresetCounter	DINT	This input specifies the initial value of the *Counter* output.
Axis	AXIS_REF	This input specifies the motor axis on which the actions of the function module are to be executed.

Outputs

The function module MC_TouchProbe1_MDX has the following outputs:

Output	Type	Description
EdgeDetected	BOOL	This output shows whether a touch probe interrupt event has taken place. • *TRUE*: Touch probe event has taken place. • *FALSE*: This output is set automatically to *FALSE* after each control cycle.
Enabled	BOOL	This output shows whether the function module is active. • *TRUE*: The Enable input is set to *TRUE*. • *FALSE:* The Enable input is set to *FALSE*.
Error	BOOL	This output shows whether an error as occurred in the function module. • *TRUE*: An error has occurred in the function module. • *FALSE:* No error has occurred in the function module.
*Error*ID	DWORD	This output displays the error code of the error (→ page 68).
TouchProbePos	DINT	This output shows the touch probe position of the selected encoder system.
Counter	DINT	This output displays the number of touch probe interrupt events.

4.7.2 Function module MC_TouchProbe2_MDX



20028AXX

Description

The function module MC_TouchProbe2_MDX controls a maximum of two touch probe interrupt routines.

- The function module copies the interrupt routines in the IPOS® code of the connected drive inverter. The drive inverter initializes a total of three acyclical send objects (control word 2, touch probe 1 position, touch probe 2 position).
- When the *Enable* input signal of the function module is set to *TRUE*, the function module sets up a receive object. When the function module detects the touch probe event, it reads the actual position from the drive inverter.

TouchProbe Routine

1. Calling the function module
   - The function module MC_TouchProbe2_MDX is called by setting the *Enable* input signal to *TRUE*.
   - The drive inverter initializes the IPOS® state automatically and starts a touch probe interrupt routine.
     (Edge change and touch probe selection)
2. Waiting for a touch probe event
   - The function module MC_TouchProbe2_MDX sets the *Enabled* output to *TRUE*.
3. The touch probe event has occurred.
   - The drive inverter signals to the MOVI-PLC® controller that the touch probe event has occurred.
     (CtrlWord2, TouchProbe2 -> Bit 0, TouchProbe2 -> Bit 1)
   - The function module MC_TouchProbe2_MDX reads in the touch probe position of the selected encoder system.
   - The function module MC_TouchProbe2_MDX increases the value of the *Counter* output by one.
   - The function module MC_TouchProbe2_MDX sets the *EdgeDetected* output to *TRUE*.
   - Monitoring of the touch probe input is started again.
4. Evaluating the touch probe event
   - The function module MC_TouchProbe2_MDX transfers the touch probe positions to the outputs *TouchProbePos_X14* to TouchProbePos_X62.
   - The function module resets the output *EdgeDetected* to *FALSE*.
   - The function module activates the touch probe interrupt routine.

Application

The function module MC_TouchProbe2_MDX can be used on all motor axes.

Inputs

The function module MC_TouchProbe2_MDX has the following inputs:

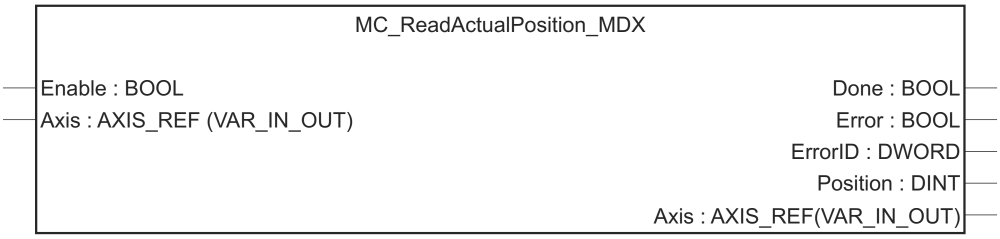
Input	Type	Description
Enable	BOOL	This input is used to activate the function module MC_TouchProbe2_MDX. The function module is only executed when the *Enable* input is set to *TRUE*. The values of the other inputs are only read in when a rising edge occurs at the *Enable* input.
TouchProbe Source	MC_TP_ SOURCE _MDX	This input specifies the encoder system. This setting is made independent of the setting in the controller configuration. • *X15*: A motor encoder measures the position. • *X14:* An external encoder measures the position. • *SSI:* An absolute encoder measures the position (DIP11B).
Edge	MC_TP_ EDGE_MDX	This input defines the edge evaluation of the touch probe input: • *EN*: Both edges • *EN_HI:* Rising edge • *EN_LO:* Falling edge
PresetCounter	DINT	This input specifies the initial value of the *Counter* output.
Axis	AXIS_REF	This input specifies the motor axis on which the actions of the function module are to be executed.

Outputs

The function module MC_TouchProbe2_MDX has the following outputs:

Output	Type	Description
EdgeDetected	BOOL	This output shows whether a touch probe interrupt event has taken place. • *TRUE*: Touch probe event has taken place. • *FALSE*: This output is set automatically to *FALSE* after each control cycle.
Enabled	BOOL	This output shows whether the function module is active. • *TRUE*: The Enable input is set to *TRUE*. • *FALSE:* The Enable input is set to *FALSE*.
Error	BOOL	This output shows whether an error as occurred in the function module. • *TRUE*: An error has occurred in the function module. • *FALSE:* No error has occurred in the function module.
*Error*ID	DWORD	This output displays the error code of the error (→ page 68).
TouchProbePos	DINT	This output shows the touch probe position of the selected encoder system.
Counter	DINT	This output displays the number of touch probe interrupt events.

4.7.3 Function module MC_ReadActualPosition_MDX


MC_ReadActualPosition_MDX
Enable : BOOL
Axis : AXIS_REF (VAR_IN_OUT)
Done : BOOL
Error : BOOL
ErrorID : DWORD
Position : DINT
Axis : AXIS_REF(VAR_IN_OUT)

20025AXX

Description

The function module MC_ReadActualPosition_MDX is used to read in the current position of the motor axis that is detected by the encoder system. The encoder system used is specified in the controller configuration in the module parameters of the drive inverter.

Application

The function module MC_ReadActualPosition_MDX can be used on all motor axes. For motor axes without encoders, the value of the transferred actual position may not be used when a motor encoder is not connected at input *X15* and set as an encoder in the controller configuration *X15* .

Inputs

The function module MC_ReadActualPosition_MDX has the following inputs:

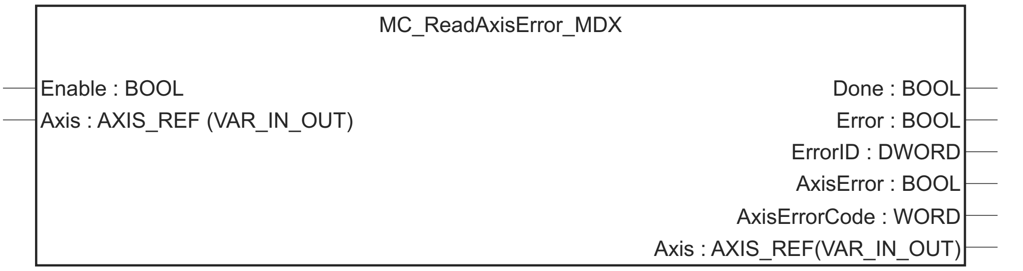
Input	Type	Description
Enable	BOOL	This input is used to activate the function module. This task of the function module is only executed when this input is set to *TRUE*.
Axis	AXIS_REF	This input specifies the motor axis on which the actions of the function module are to be executed.

Outputs

The function module MC_ReadActualPosition_MDX has the following outputs:

Output	Type	Description
Done	BOOL	This output shows whether the motor axis position has been transferred. • *TRUE*: The value of the motor axis position of the *Position* output is in effect. • *FALSE:* The motor axis position has not been transferred.
Error	BOOL	This output shows whether an error as occurred in the function module. • *TRUE*: An error has occurred during the transfer of the motor axis position. • *FALSE:* No error has occurred.
*Error*ID	DWORD	This output displays the error code of the error (→ page 68).
Position	DINT	This output contains the transferred motor axis position.
Axis	AXIS_REF	This output displays the axis reference.

4.7.4 Function module MC_ReadAxisError_MDX



20026AXX

Description The function module MC_ReadAxisError_MDX is used to read in the current error code of the motor axis from the drive inverter to the MOVI-PLC® controller.

Application The function module MC_ReadAxisError_MDX can be used on all motor axes.

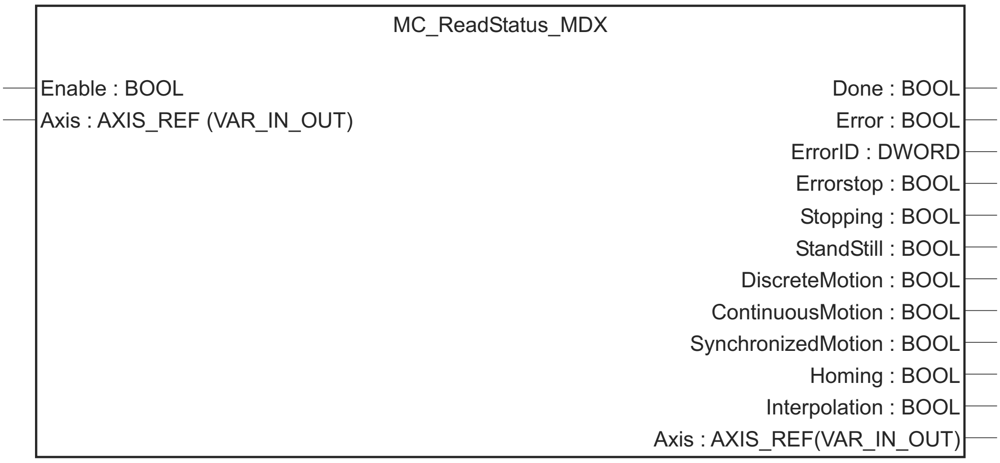
Inputs The function module MC_ReadAxisError_MDX has the following inputs:

Input	Type	Description
Enable	BOOL	This input is used to activate the function module. This task of the function module is only executed when this input is set to *TRUE*.
Axis	AXIS_REF	This input specifies the motor axis on which the actions of the function module are to be executed.

Outputs The function module MC_ReadAxisError_MDX has the following outputs:

Output	Type	Description
Done	BOOL	This output shows whether the error code has been transferred. • *TRUE*: The value of the error code at the output is in effect. • *FALSE:* The error code has not been transferred.
Error	BOOL	This output shows whether an error as occurred in the function module. • *TRUE*: An error has occurred while the error code was being transferred. • *FALSE:* No error has occurred.
*Error*ID	DWORD	This output displays the error code of the error (→ page 68).
AxisError	BOOL	This output shows whether the drive inverter has detected an error. • *TRUE*: The drive inverter has detected an error. • *FALSE:* No error has occurred.
AxisErrorCode	WORD	This output contains the transferred error code.
Axis	AXIS_REF	This output displays the axis reference.

4.7.5 Function module MC_ReadStatus_MDX



20027AXX

Description The function module MC_ReadStatus_MDX is used to read in the PLCopen state of the motor axis from the drive inverter to the MOVI-PLC® controller.

Application The function module MC_ReadStatus_MDX can be used on all motor axes.

Inputs The function module MC_ReadStatus_MDX has the following inputs:

Input	Type	Description
Enable	BOOL	This input is used to activate the function module. This task of the function module is only executed when this input is set to *TRUE*.
Axis	AXIS_REF	This input specifies the motor axis on which the actions of the function module are to be executed.

Outputs

The function module MC_ReadStatus_MDX has the following outputs:

Output	Type	Description
Done	BOOL	This output shows whether the PLCopen state has been transferred. • *TRUE*: The PLCopen state has been transferred. • *FALSE: The PLCopen state has not been transferred.*
Error	BOOL	This output shows whether an error as occurred in the function module. • *TRUE*: An error occurred while the PLCopen state was being transferred. • *FALSE:* No error has occurred.
*Error*ID	DWORD	This output displays the error code of the error (→ page 68).
Errorstop	BOOL	This output shows whether the PLCopen state of the drive inverter is set to *Errorstop*. • *TRUE*: The PLCopen state is set to *Errorstop*. • *FALSE:* The PLCopen state is not set to *Errorstop*.
Stopping	BOOL	This output shows whether the PLCopen state of the drive inverter is set to *Stopping*. • *TRUE*: The PLCopen state is set to *Stopping*. • *FALSE:* The PLCopen state is not set to *Stopping*.
StandStill	BOOL	This output shows whether the PLCopen state of the drive inverter is set to *StandStill*. • *TRUE*: The PLCopen state is set to *StandStill*. • *FALSE:* The PLCopen state is not set to *Standstill*.
Discrete Motion	BOOL	This output shows whether the PLCopen state of the drive inverter is set to *DiscreteMotion*. • *TRUE*: The PLCopen state is set to *DiscreteMotion*. • *FALSE:* The PLCopen state is not set to *DiscreteMotion*.
Continuous Motion	BOOL	This output shows whether the PLCopen state of the drive inverter is set to *ContinuousMotion*. • *TRUE*: The PLCopen state is set to *ContinuousMotion*. • *FALSE:* The PLCopen state is not set to *ContinuousMotion*.
Synchronized Motion	BOOL	This output shows whether the PLCopen state of the drive inverter is set to *SynchronizedMotion*. • *TRUE*: The PLCopen state is set to *SynchronizedMotion*. • *FALSE:* The PLCopen state is not set to *SynchronizedMotion*.
Homing	BOOL	This output shows whether the PLCopen state of the drive inverter is set to *Homing*. • *TRUE*: The PLCopen state is set to *Homing*. • *FALSE:* The PLCopen state is not set to *Homing*.
Interpolation	BOOL	This output shows whether the PLCopen state of the drive inverter is set to *Interpolation*. • *TRUE*: The PLCopen state is set to *Interpolation*. • *FALSE:* The PLCopen state is not set to *Interpolation*.
Axis	AXIS_REF	This output displays the axis reference.

4.8 Error ID

The following table displays the error codes, error desginations and a description of the errors that could occur at the *ErrorID* output of the function modules.

Error code	Error designation	Error description
General IEC error code		
FA0001h	E_IEC_GENERAL_MAX_NUMBER_OF_AXIS	Maximum number of axes exceeded
FA0002h	E_IEC_GENERAL_INTERNAL_ERROR	Error during initialization of a MOVI-PLC® interface
FA0003h	E_IEC_GENERAL_COM_NOT_READY	COM interface is not ready
FA0004h	E_IEC_GENERAL_INVALID_COM_NODE	COM interface is not valid
FA0005h	E_IEC_GENERAL_INVALID_COM_ADR	COM address is not valid
FA0006h	E_IEC_GENERAL_INVALID_TECHNOLOGIE_OPTION	To perform the requested function, a MOVIDRIVE® drive inverter in application version is required.
FA0070h	E_IEC_PARAMETER_VALUE_OUT_OF_RANGE	Value for parameter access is outside the permitted range of values.
FA0071h	E_IEC_PARAMETER_INVALID_SELECTION	Invalid input selection on function module
FA0072h	E_IEC_PARAMETER_INVALID_SERVICE	Invalid service
MPLCMotion_MDX error codes		
FB0030h	E_MDX_CONNECTAXIS_NO_INVERTER_CONNECTED	No MOVIDRIVE® MDX drive inverter detected on CAN bus. Check CAN connection.
FB0031h	E_MDX_CONNECTAXIS_CAN_ID_ERROR	Required CAN ID is occupied
FB0032h	E_MDX_CONNECTAXIS_CYCLIC_COMMUNICATION	Cyclical communication between the MOVI-PLC® controller and the MOVIDRIVE® drive inverter has been interrupted.
FB0033h	E_MDX_CONNECTAXIS_IPOS_DOWNLOAD_ERROR	Error during IPOS® download
FB0034h	E_MDX_CONNECTAXIS_WRONG_DEVICE_CONNECTED	Incorrect unit connected. Check controller configuration.
FB0035h	E_MDX_CONNECTAXIS_SIMULATION_NOT_AVAILABLE	Function cannot be executed in simulation mode.
FB0060h	E_MDX_POWER_INVERTER_NOT_READY	MOVIDRIVE® MDX drive inverter assigned the status "24V operation" or "Safe stop." Device cannot be switched on.
FB0061h	E_MDX_POWER_INVERTER_FAULT_STATE	Error detected on MOVIDRIVE® MDX drive inverter. Device cannot be switched on.
FB0070h	E_MDX_MOTIONBLOCK_INVALID_DATA_PROFIL	Motion function module cannot be executed from the current PLCopen state.
FB0071h	E_MDX_MOTIONBLOCK_LOG_ADR_NOT_INITIALIZED	Motion function module was called up before MC_ConnectAxis_MDX had generated a logical address (AXIS_REF).
FB0072h	E_MDX_MOTIONBLOCK_INVALID_LOG_ADR	motion function module was called up with an invalid logical address (AXIS_REF).
FB0073h	E_MDX_MOTIONBLOCK_INVALID_STATE	Motion function module cannot be executed from the current PLCopen state.
FB0074h	E_MDX_MOTIONBLOCK_INVALID_OPERATING_MODE	Motion function module cannot be executed in the operating mode set on the MOVIDRIVE® MDX inverter.
FB0075h	E_MDX_MOTIONBLOCK_INVALID_INVERTER_STATUS	Motion function module cannot be executed in the state currently active on the MOVIDRIVE® MDX inverter.
FB0076h	E_MDX_MOTIONBLOCK_INVALID_VELOCITY	Specified speed is outside the permitted range of values.
FB0090h	E_MDX_PARAMCHANNEL_SEND_BUFFER_OVERFLOW	Buffer for parameter channel is full.
FB0091h	E_MDX_PARAMCHANNEL_WRITEPARAMETER_BUSY	Trying to access parameter while the parameter channel is busy
FB0092h	E_MDX_PARAMCHANNEL_READPARAMETER_BUSY	Trying to access parameter while the parameter channel is busy

Error code	Error designation	Error description
MPLCUtilities error codes		
FC0001h	E_WAGO_TIME_OUT_SBUS_INIT	Internal timeout during SBUS initialization
FC0002h	E_WAGO_TIME_OUT_WAGO_INIT	Timeout during initialization of the WAGO module
FC0003h	E_WAGO_NO_CONNECTION_DURING_INIT_SEQ	Termination of the connection to WAGO module during initialization
FC0004h	E_WAGO_WATCH_DOG_ERROR	Watchdog error. Communication to WAGO module interrupted.
MPLCSystem error codes (extract)		
CC0070h	E_NO_MATCH_FOR_PARAMSET	No entry found in controller configuration for the axis.
F2000Ah	E_MVLINK_CAN	MOVILINK® error. Possible cause: Value outside the permitted range of values.
F20002h	E_MVLINK_RS485	MOVILINK® error. Possible cause: Value outside the permitted range of values.

5 Programming Examples

This section describes how to program a number of drive tasks using the function modules described in this manual by means of specific examples.

Furthermore, this section introduces you to the basic functionality of the controller configuration, the library manager and the FUP Editor of the MOVITOOLS® MotionStudio software.

For more information, refer to the online help of the MOVITOOLS® MotionStudio software.

5.1 Prerequisites

To be able to test the created programs with a motor axis, the following prerequisites must be fulfilled:

- The MOVI-PLC® controller and a MOVIDRIVE® MDX60B/61B drive inverter with a connected motor[1)]are installed according to the instructions in the relevant manuals.
- A system bus connection is installed correctly between the CAN 1 connection of the MOVI-PLC® controller and the CAN 1 connection of the MOVIDRIVE® MDX60B/61B drive inverter in accordance with the instructions in the relevent manuals[2)].
- The engineering PC is connected correctly to the MOVI-PLC® controller in accordance with the instructions in the relevant manuals. The interface of the engineering PC is configured correctly.
- You used the startup assistant to start the MOVIDRIVE® MDX60/B61B drive inverter to be controlled with a MOVI-PLC® controller (→ Sec. 3.3, page 10).

[1)] A motor with encoder is required for positioning.

[2)] When the MOVI-PLC® controller communicates with the drive inverter via the backplane connector, you do not need the system bus connection via the CAN 1 connection.

5.2 Positioning a motor axis

Task description

When a rising edge of 24V DC voltage occurs at a digital input of the connected MOVIDRIVE® MDX60B/61B drive inverter, the connected motor axis should turn 10 revolutions clockwise and be positioned.

The drive is operated by using only the digital inputs. The MOVI-PLC® controller evaluates the inputs of the drive inverter and controls the positioning of the motor axis.

Sub-tasks

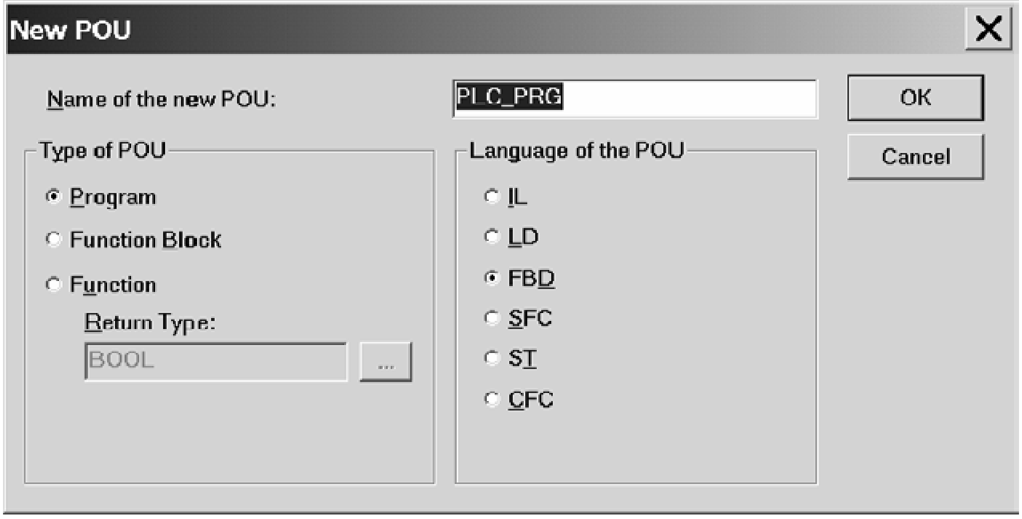
The programming example is divided into the following sub-tasks:

1. Creating a new project
2. Setting the controller configuration
3. Install the required libraries
4. Programming communication with the motor axis
5. Programming the switch on/off procedure for the drive inverter
6. Programming the positioning process for the motor axis
7. Transferring the project to the MOVI-PLC® controller
8. Testing the program

Step 1

Create a new project

1. Switch on the engineering PC and the MOVI-PLC® controller[1].
2. Start the PLC Editor of the MOVITOOLS® MotionStudio software in accordance with the instructions in the section "Starting MOVITOOLS® MotionStudio" in the MOVI-PLC® manual.
3. Create a new project by choosing [File] / [New].
4. The [Target Settings] dialog box appears. Select the configuration of your MOVI-PLC® controller; in this programming example it can be [MOVIPLC basic DHP11B] and click [OK].



Name of the new POU: PLC_PRG

● Program

● FBD

20030AXX

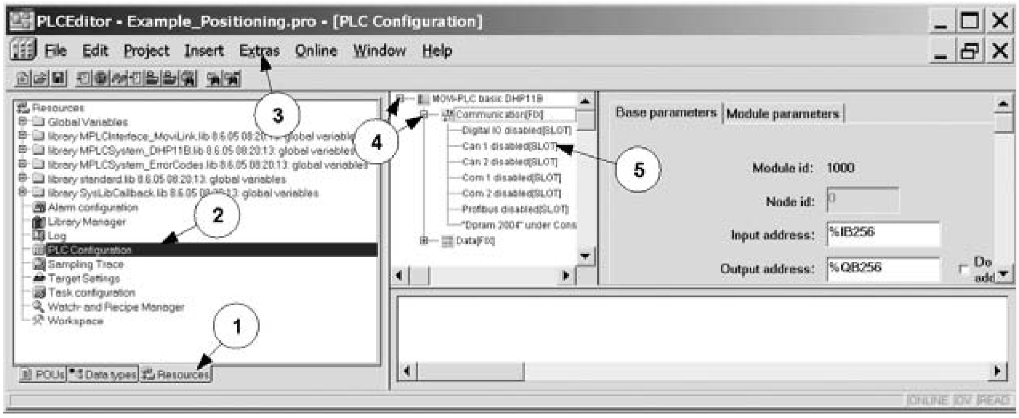
5. In this example, do not change the name of the function module "PLC_PRG" in the [Name of the new POU:] selection field. After you start the program (step 8 in this example), the MOVI-PLC® controller will execute the module "PLC_PRG" automatically.
6. In the [Type of POU] field, choose [Program].
7. In the [Language of the POU] group box, check the radio button [FBD] and confirm your entries by clicking [OK].
8. Save the project by choosing [File] / [Save] and enter a name for the project. Save the project regularly once you have made a number of changes or after you have finished creating the program.

[1] The MOVI-PLC® controller must be powered as your version requires, or the drive inverter that the hardware of the MOVI-PLC® controller is plugged into must be switched on.

Step 2

Set the controller configuration

Set the controller configuration as required to be able to use the interfaces of the MOVI-PLC® controller to connect peripheral equipment and to enable communication with other devices, e.g. inverters, I/O modules or a master controller.



Module id: 1000

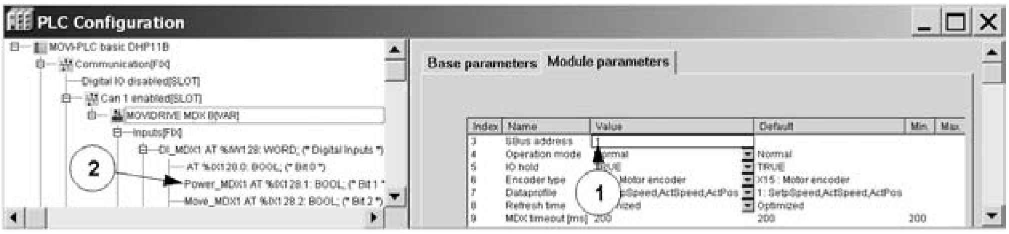
Node id: 0

Input address: %IB256

Output address: %QB256

20031AXX

1. Choose the [Resources] tab page, located at the bottom of the screen [1].
2. Activate [PLC Configuration] by double-clicking the entry [2].
3. From the [Extras] menu, set the [Standard configuration] once. Confirm the query as to whether you want to delete the current configuration and replace it with the standard configuration by selecting [Yes] [3].
4. Open the configuration tree by clicking on the [+] symbol in front of the entries [MOVI-PLC basic DHP11B] and [Communication] [4].
5. Activate the CAN 1 interface of the MOVI-PLC® controller by clicking with the right mouse button on the entry [Can 1 disabled] and choosing [Replace element] / [Can 1 enabled] from the context menu that appears [5].
6. Configure the MOVIDRIVE® MDX60B/61B drive inverter at the CAN 1 interface by clicking on the right mouse button on the entry [Can 1 enabled] and choosing [Append subelement] and [MOVIDRIVE MDX B] from the context menu that appears.



PLC Configuration

```
⊟— MOVI-PLC basic DHP11B
   ⊟— Communication[FIX]
      ——Digital IO disabled[SLOT]
      ⊟— Can 1 enabled[SLOT]
         ⊟— MOVIDRIVE MDX B[VAR]
            ⊟—Inputs[FIX]
               ⊟—DI_MDX1 AT %IW128: WORD; (* Digital Inputs *)
                  ——AT %IX128.0: BOOL; (* Bit 0 *)
                  ——Power_MDX1 AT %IX128.1: BOOL; (* Bit 1 *
                  ——Move_MDX1 AT %IX128.2: BOOL; (* Bit 2 *)
```

Base parameters | Module parameters

Index	Name	Value	Default	Min.	Max.
3	SBus address	1			
4	Operation mode	ormal	Normal		
5	IO hold	RUE	TRUE		
6	Encoder type	Motor encoder	X15 : Motor encoder		
7	Dataprofile	pSpeed,ActSpeed,ActPos	1: SetpSpeed,ActSpeed,ActPos		
8	Refresh time	ized	Optimized		
9	MDX timeout [ms]	200	200	200	

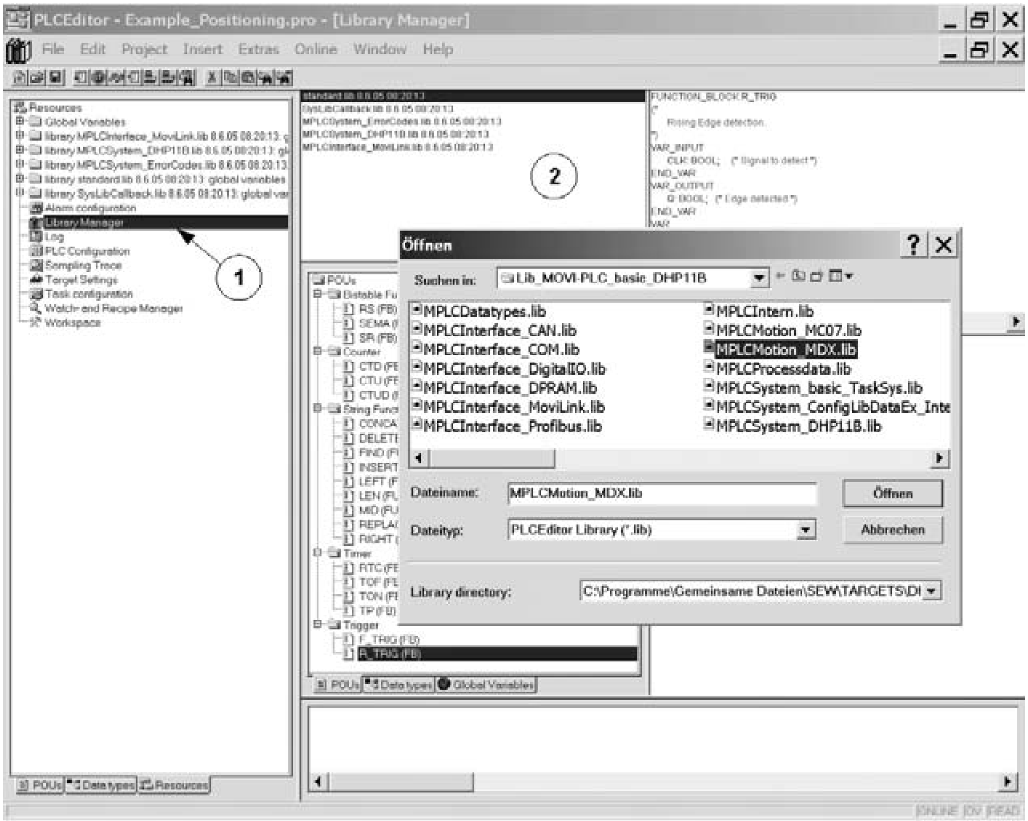
20032AXX

7. Mark the element [MOVIDRIVE MDX B] and in the [Module parameters] tab page, enter the SBus address that you selected when you started the drive inverter [1].
8. In the control program, you can use the inputs and outputs by specifying the address details directly using the syntax "%I" / "%Q".

   However, it is easier to use symbolic identifiers as follows: Click on the [+] symbol in front of the entries [MOVIDRIVE MDX B] and [Inputs]. Click on the corresponding [AT] field in the configuration tree and enter the name as required. In this example, the symbolic identifier for the digital inputs of the drive inverter is [DI_MDX1]. For individual inputs/outputs, the symbolic identifiers are [Power_MDX1] and [Move_MDX1] [2].

Step 3 **Connect the libraries**

To use the function modules of the `MPLCMotion_MDX.lib` library, add the library [MPLCMotion_MDX.lib] to the existing libraries as follows.


PLCEditor - Example_Positioning.pro - [Library Manager]
File Edit Project Insert Extras Online Window Help
Resources
Library Manager
1
2
Öffnen
Suchen in: Lib_MOVI-PLC_basic_DHP11B
MPLCDatatypes.lib
MPLCInterface_CAN.lib
MPLCInterface_COM.lib
MPLCInterface_DigitalIO.lib
MPLCInterface_DPRAM.lib
MPLCInterface_MoviLink.lib
MPLCInterface_Profibus.lib
MPLCIntern.lib
MPLCMotion_MC07.lib
MPLCMotion_MDX.lib
MPLCProcessdata.lib
MPLCSystem_basic_TaskSys.lib
MPLCSystem_ConfigLibDataEx_Inte
MPLCSystem_DHP11B.lib
Öffnen
Abbrechen
POUs
Data types
Global Variables
Resources

Dateiname: MPLCMotion_MDX.lib

Dateityp: PLCEditor Library (*.lib)

Library directory: C:\Programme\Gemeinsame Dateien\SEW\TARGETS\DI

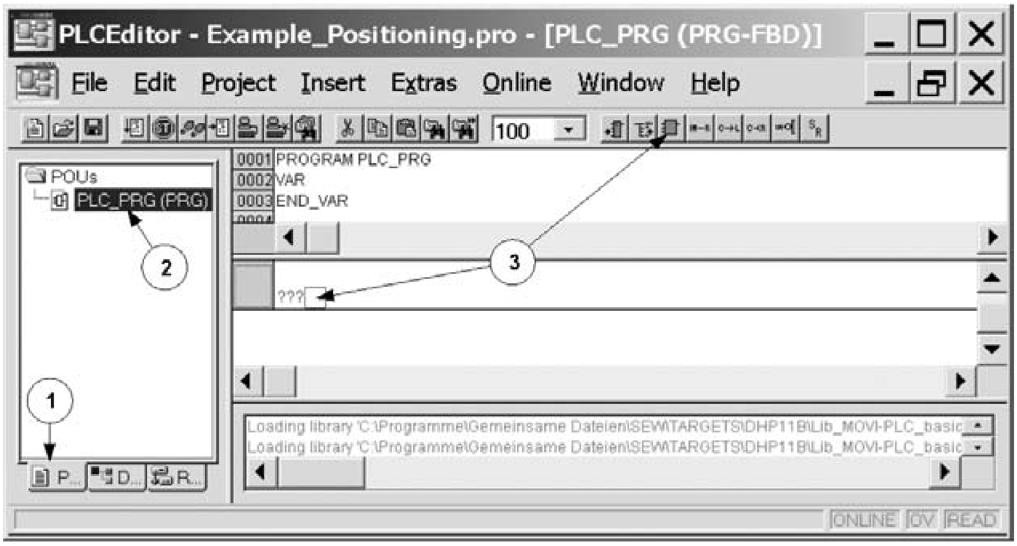
20033AXX

1. Activate the [Library manager] by double-clicking the entry [1].
2. Click the right mouse button in the library field and choose [Additional Library] [2].
3. Select the library [`MPLCMotion_MDX.lib`] and click [Open].

Step 4

Program the communication with the motor axis

To establish and execute communication between the MOVI-PLC® controller and the drive inverter, add an instance of the MC_ConnectAxis_MDX function module as follows.

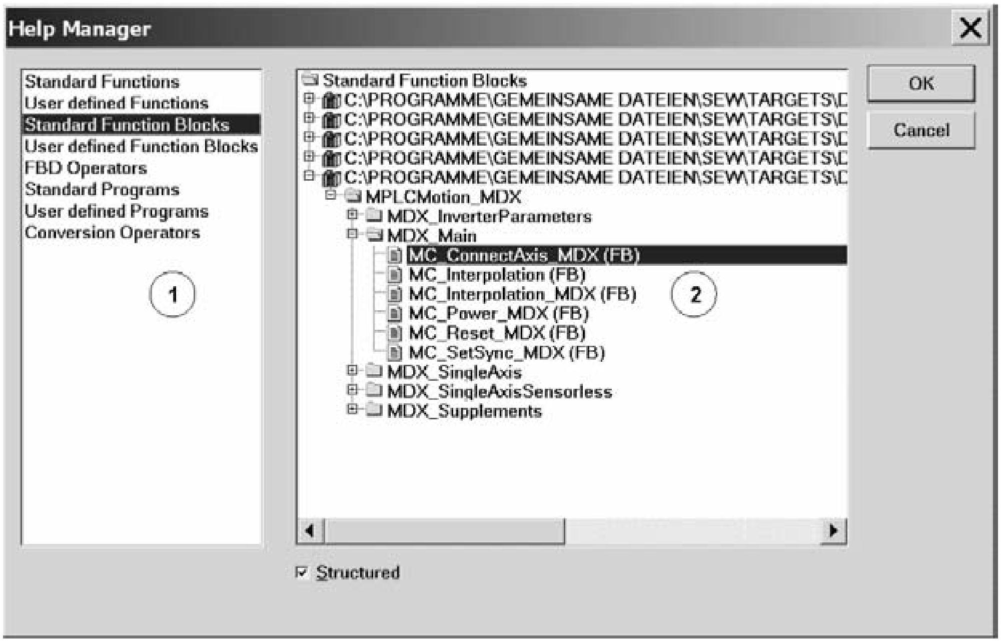

PLCEditor - Example_Positioning.pro - [PLC_PRG (PRG-FBD)]
File Edit Project Insert Extras Online Window Help
100
POUs
PLC_PRG (PRG)
???
P... D... R...
ONLINE OV READ
1
2
3

```
0001 PROGRAM PLC_PRG
0002 VAR
0003 END_VAR
```

```
Loading library 'C:\Programme\Gemeinsame Dateien\SEW\TARGETS\DHP11B\Lib_MOVI-PLC_basic
Loading library 'C:\Programme\Gemeinsame Dateien\SEW\TARGETS\DHP11B\Lib_MOVI-PLC_basic
```

20034AXX

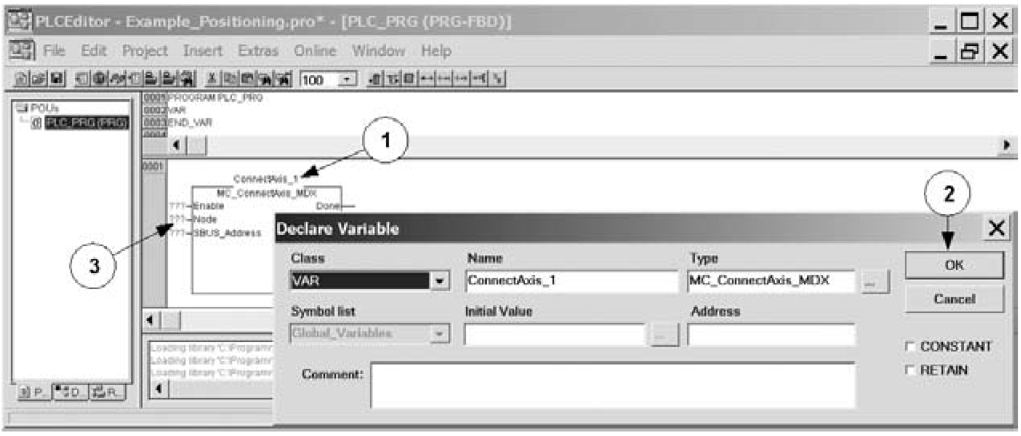
1. Choose the [POUs] tab page, located at the bottom of the screen [1].
2. Open the editor of the [PLC_PRG(PRG)] module by double-clicking on the entry [2].
3. Insert a new function module by first clicking on the box next to the questions marks [???] and then pressing the button [3].



20036AXX

4. Mark the text "AND" in the function module you have just added.
5. Press the <F2> button. The software opens the [Help Manager] dialog box.

6. In the panel on the left of the screen, choose [Standard Function Blocks] [1].
7. In the panel on the right of the screen, choose the function module [MC_ConnectAxis_MDX (FB)] from the directory [`MDX_Main`] of the [`MPLCMotion_MDX`] library. Click [OK] [2].



```
0001 PROGRAM PLC_PRG
0002 VAR
0003 END_VAR
```

Class: VAR

Name: ConnectAxis_1

Type: MC_ConnectAxis_MDX

Symbol list: Global_Variables

Initial Value:

Address:

Comment:

20037AXX

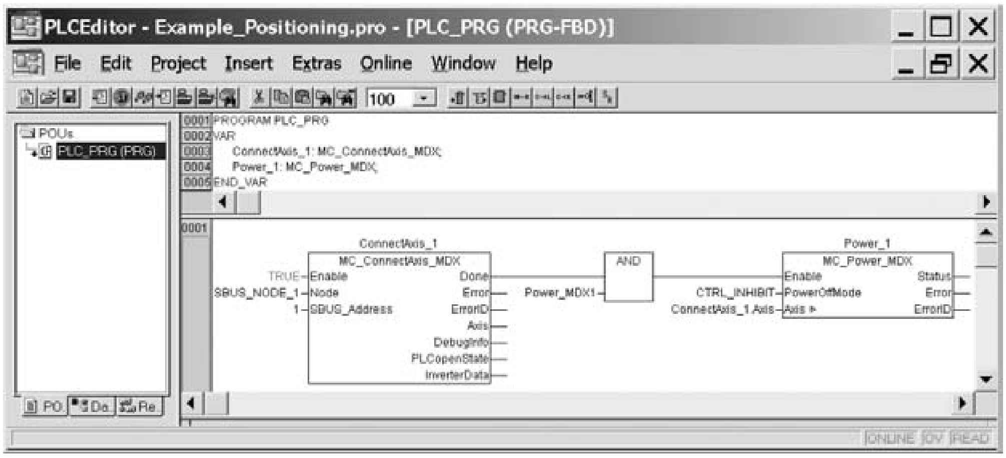
8. Click on the inserted function module MC_ConnectAxis_MDX and then on the question marks "???" above the function module [1].
9. Enter the instance name (e.g. "ConnectAxis_1") of the function module [2] and press the <ENTER> key. Confirm the [Declare Variable] dialog box that appears by pressing the [OK] button.
10. Enter the following values at the inputs of the function module by clicking on the question marks "???" to the left of the input, enter the value and press the <ENTER> key.

Enable	*TRUE*
Node	*SBUS_NODE_1*
SBUS_Address	SBUS 1 address set during startup of the drive inverter

Schritt 5

Program the switch on/off procedure for the drive inverter

Insert a function module that turns the MOVIDRIVE® MDX60B/61B drive inverter on/off. The drive inverter can only be switched on if the MC_ConnectAxis_MDX function module has already been executed for this axis. Furthermore, the drive inverter should only be switched on when 24 VDC voltage is applied at the digital input *DI01* of the drive inverter. Therefore, the corresponding signals are linked with each other using an "AND" module.



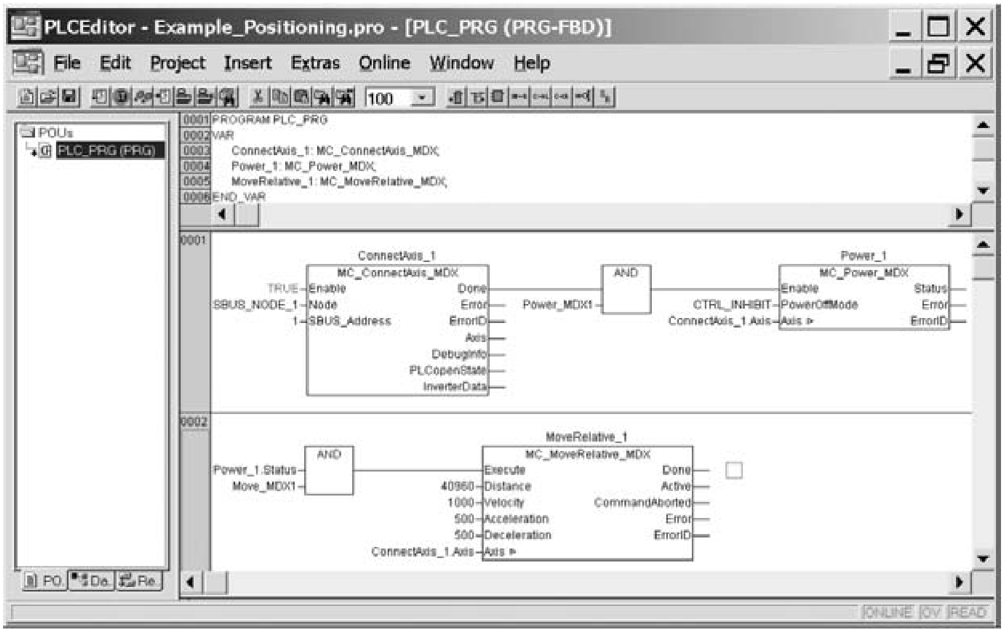
```
PROGRAM PLC_PRG
VAR
    ConnectAxis_1: MC_ConnectAxis_MDX;
    Power_1: MC_Power_MDX;
END_VAR
```

20038AXX

1. Click on the line of the *Done* output of the MC_ConnectAxis_MDX function module.
2. Insert a new "AND" function module by clicking the <⬛> button.
3. At the second input of the new "AND" function module, add the value read in at the digital input *DI01* of the drive inverter. To do so, enter the symbolic name selected in the controller configuration (in this example, "MDX1_DI.1" or directly "Power_MDX1").
4. Insert the function module MC_Power_MDX by clicking to the right of the "AND" function module and then pressing the [⬛] button. Convert the "AND" function module that you have inserted to an MC_Power_MDX function module by following the instructions described in step 4. Enter a name for the function module, e.g. the instance name "Power_1".
5. At the *PowerOffMode* input, enter the constant "CTRL_INHIBIT".
6. At the *Axis* input, enter the axis reference "ConnectAxis_1.Axis" that is output by the module instance *ConnectAxis_1*. To do so, click the field "???" before the *Axis* input and enter "ConnectAxis_1.". Select the entry [Axis] from the dialog box that appears automatically. Confirm the entry by pressing the <ENTER> button.

Step 6 **Program the positioning process for the motor axis**

Insert an MC_MoveRelative_MDX function module that controls a relative positioning movement of the motor axis. In this example, the motor axis should turn ten revolutions (=10 x 4096 increments of the encoder) clockwise each time a rising edge of the 24V DC voltage occurs at the digital input *DI02* of the drive inverter. During constant travel, the motor axis should turn at a speed of 1000 1/min.



```
PROGRAM PLC_PRG
VAR
    ConnectAxis_1: MC_ConnectAxis_MDX;
    Power_1: MC_Power_MDX;
    MoveRelative_1: MC_MoveRelative_MDX;
END_VAR
```

20039AXX

1. Click the right mouse button in a free area of the [0001] network. A context menu appears: Choose [Network (after)].
2. In the new [0002] network, insert an MC_MoveRelative_MDX function module and give it an instance name (e.g. "MoveRelative_1") as described in step 4. The MC_MoveRelative_MDX function module is included in the [`MDX_SingleAxis`] directory of the [`MPLCMotion_MDX`] library under the [Standard Function Blocks] entry.
3. Enter the following values at the inputs of the function module:

Distance	*40960*
Velocity	*1000*
Acceleration	*500*
Deceleration	*500*
Axis	*ConnectAxis_1.Axis*

4. The travel command can only be executed when the MC_Power_MDX function module has been performed beforehand. Therefore, at the *Execute* input of the MC_MoveRelative_MDX function module, enter the result of an "AND" connection between the *Status* output of the MC_Power_MDX function module and the digital input, which is to be used to start the movement.

   To do so, click on the line before the *Execute* input. Insert a new "AND" function module by clicking the [▣] button. Assign the inputs of the "AND" function module as required (see screenshot above).
5. To rebuild the project, choose [Project] / [Rebuild all]. When the programming is error-free, the message window displays the message "0 Error(s), 0 Warnings(s)".

Step 7

Transfer project to the MOVI-PLC® controller

Choose the menu item [Online] / [Communication Parameters]).

In the dialog box that appears, set the communication parameters in accordance with the communication channel to be used. You only have to perform this step once.

Next select the menu item [Online] / [Login].

In the dialog box that appears, choose [Yes] in response to the question whether you want to load a program.

Step 8

Test the program

In the final step, execute the program by activating the digital inputs as required.

Warning:

Depending on the terminal assignment, the drive inverter status and the control program, the motor axis can begin to move immediately after the MOVI-PLC® controller has been started. To prevent risk of injury, keep sufficient distance from all moving parts.

Turn on the MOVIDRIVE® MDX60B/61B drive inverter if it was not already switched on in step 1 in connection with the MOVI-PLC® controller.

Start the MOVI-PLC® controller by clicking on the menu item [Online] / [Run].

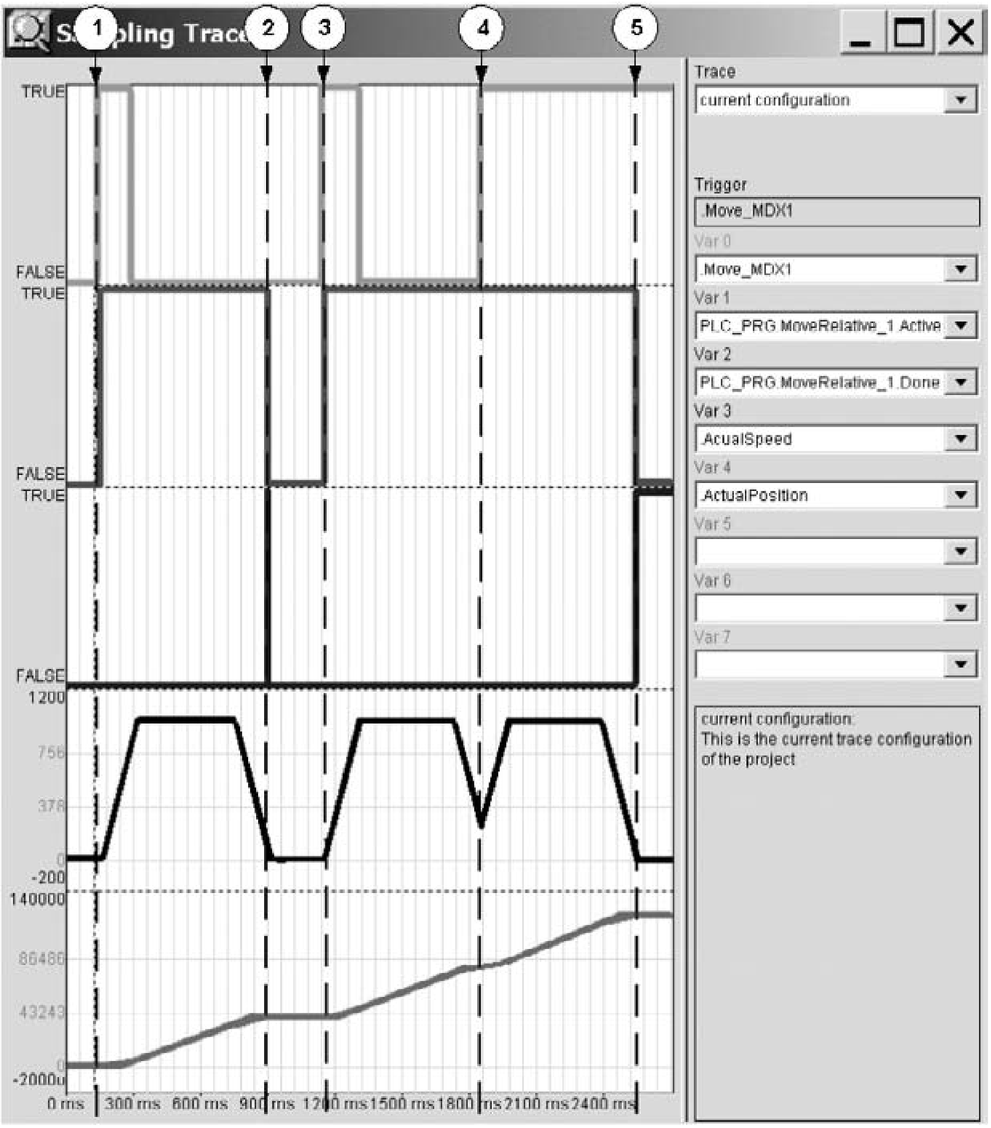
Switch on the drive inverter by applying the 24 V DC voltage first to input *DI00* "/Controller inhibit" and then to input *DI01* (Enable of the MC_Power_MDX function module).

Start positioning of the motor axis by applying 24 V DC voltage at the *DI02* input of the drive inverter.

Programming is correct when the motor axis turns ten revolutions clockwise each time a rising edge of 24 V DC voltage occurs at the *DI02* input.

For more details on the behavior of the MOVI-PLC® controller and the connected MOVIDRIVE® drive inverter in this example, refer to the following section "Trace recording."

Trace recording



Trace
current configuration

Trigger
.Move_MDX1

Var 0
.Move_MDX1

Var 1
PLC_PRG.MoveRelative_1.Active

Var 2
PLC_PRG.MoveRelative_1.Done

Var 3
.AcualSpeed

Var 4
.ActualPosition

Var 5

Var 6

Var 7

current configuration:
This is the current trace configuration of the project

20087AXX

When a rising edge occurs at the *Move_MDX1* signal at the *Execute* input of the MC_MoveRelative_MDX function module, the motor axis starts to move to the target position [1]. The *Active* output shows that this is the case when it is set to *TRUE*.

Once positioning is complete, the function module resets the *Active* output to *FALSE* and sets output *Done* to *TRUE* [2]. In this example, the *Done* output is only set to *TRUE* for one control cycle because the *Execute* input signal was already reset to *FALSE* before the positioning was concluded.

As a rising edge then occurs at the *Execute* input, the motor axis starts the positioning process again [3].

However, the next rising edge occurs at the *Execute* input before positioning has been concluded. Another positioning process, starting from the current position of the motor axis at the time the last rising edge occurred at the *Execute* input, is started without interruption [4]. In this way, the motor axis is not slowed to a standstill, but starts the next movement directly.

Once positioning is complete, the function module resets the *Active* output to *FALSE*. In this case, however, the *Done* output signal remains set to TRUE after positioning is complete because the Execute input signal has not been reset to FALSE [5].

5.3 Speed control of a motor axis with encoder

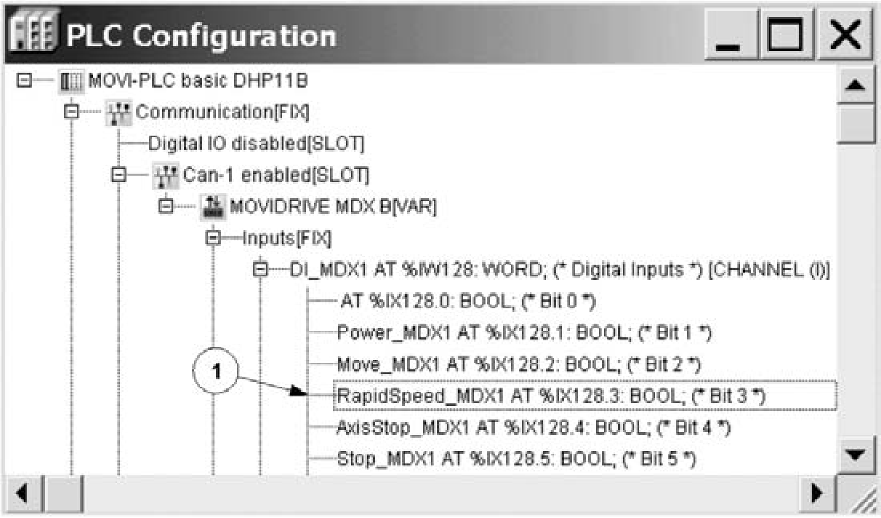
Task description When a rising edge of 24V DC voltage occurs at a digital input of the MOVIDRIVE® MDX60B/61B drive inverter, the connected motor axis should start movement subject to speed control. The speed of the motor axis should be able to be switched between two values using another digital input. Two inputs should be used to start a braking process with the help of the function module MC_Stop_MDX or MC_AxisStop_MDX.

The drive is operated by using only the digital inputs. The MOVI-PLC® controller evaluates the inputs of the drive inverter and controls the speed of the motor axis.

Programming **Unchanged steps**

Program steps 1, 3-5 and 7 as described in the previous programming example "Positioning a motor axis."

Step 2 **Set the controller configuration**



```
MOVI-PLC basic DHP11B
  Communication[FIX]
    Digital IO disabled[SLOT]
    Can-1 enabled[SLOT]
      MOVIDRIVE MDX B[VAR]
        Inputs[FIX]
          DI_MDX1 AT %IW128: WORD; (* Digital Inputs *) [CHANNEL (I)]
            AT %IX128.0: BOOL; (* Bit 0 *)
            Power_MDX1 AT %IX128.1: BOOL; (* Bit 1 *)
            Move_MDX1 AT %IX128.2: BOOL; (* Bit 2 *)
            RapidSpeed_MDX1 AT %IX128.3: BOOL; (* Bit 3 *)
            AxisStop_MDX1 AT %IX128.4: BOOL; (* Bit 4 *)
            Stop_MDX1 AT %IX128.5: BOOL; (* Bit 5 *)
```

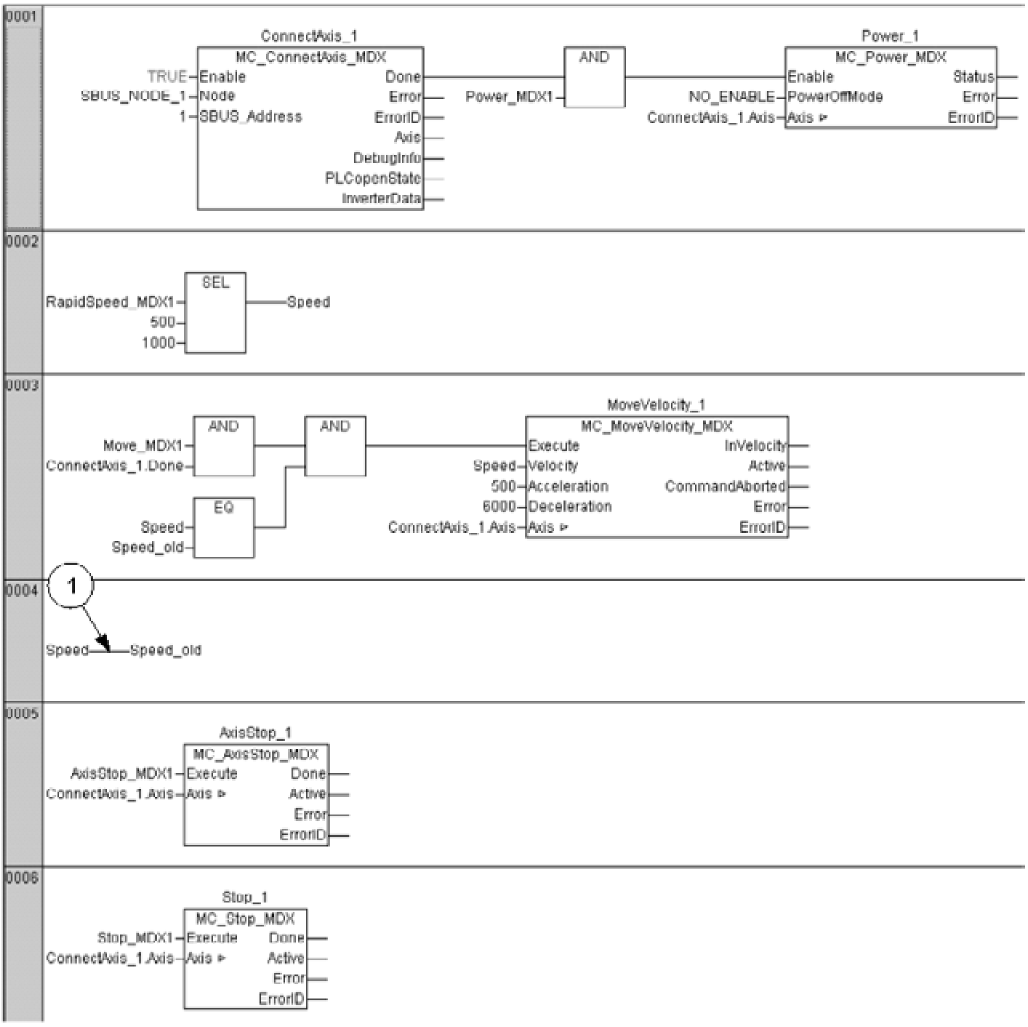
20088AXX

In the controller configuration, in addition to the identifiers [Power_MDX1] and [Move_MDX1], assign the identifiers

- [RapidSpeed_MDX1]
- [AxisStop_MDX1]
- [Stop_MDX1]

to the digital inputs of the MOVIDRIVE® MDX60B/61B drive inverter as shown in the screenshot [1].

Step 6 **Program the speed control function**



20089AXX

1. Create the program shown in the diagram in accordance with the procedure described in the previous programming example.
2. To program the value assignment of the variable *Speed* to the variable *Speed_old*, insert the network [0004] and then click on the box next to the question marks "???." Then press the [] button [1].

   Replace the question marks "???" with the variable names.

Step 8

Test the program

Execute the program by activating the digital inputs as required.

Warning:

Depending on the terminal assignment, the drive inverter status and the control program, the motor axis can begin to move immediately after the MOVI-PLC® controller has been started. To prevent risk of injury, keep sufficient distance from all moving parts.

Turn on the MOVIDRIVE® MDX60B/61B drive inverter if it was not already switched on in step 1 in connection with the MOVI-PLC® controller.

Start the MOVI-PLC® controller by clicking on the menu item [Online] / [Run].

Switch on the drive inverter by applying the 24 V DC voltage first to input *DI00* "/Controller inhibit" and then to input *DI01* (*Enable* of the MC_Power_MDX function module).

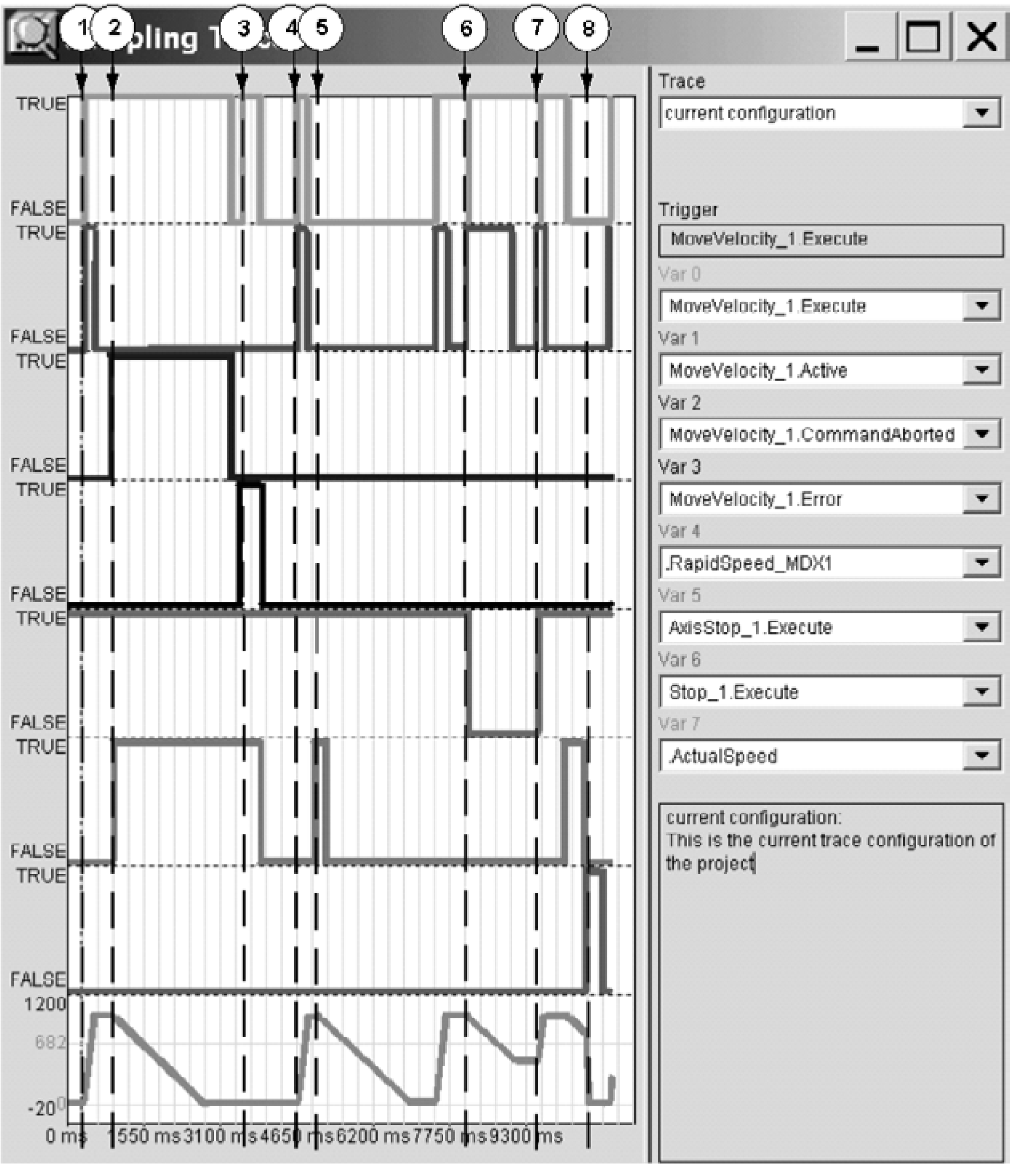
Start speed control of the motor axis by applying 24 V DC voltage at the *DI02* input of the drive inverter.

Programming is correct when

- the motor axis begins to turn when a rising edge of 24 V DC voltage occurs at input DI02
- the speed of the motor axis switches between 500 1/min and 1000 1/min when the voltage is switched between 0V and 24V DC at the *DI03* input.
- the speed of the motor axis is slowed when 24 VDC voltage is applied at the input *DI04* or *DI05*.

For more details on the behavior of the MOVI-PLC® controller and the connected MOVIDRIVE® drive inverter in this example, refer to the following section "Trace recording."

Trace recording



Trace
current configuration

Trigger
MoveVelocity_1.Execute

Var 0
MoveVelocity_1.Execute

Var 1
MoveVelocity_1.Active

Var 2
MoveVelocity_1.CommandAborted

Var 3
MoveVelocity_1.Error

Var 4
.RapidSpeed_MDX1

Var 5
AxisStop_1.Execute

Var 6
Stop_1.Execute

Var 7
.ActualSpeed

current configuration:
This is the current trace configuration of the project

20090AXX

The motor axis starts speed control when a rising edge occurs at the *Execute* input of the MC_MoveVelocity_MDX function module [1]. The function module sets the *Active* output to *TRUE* until the setpoint speed is reached. When the setpoint speed is reached, the *Active* output is reset to *FALSE* and the *InVelocity* output (not shown in the time diagram) is set to *TRUE*.

Due to a rising edge at the *Execute* input of the MC_AxisStop_MDX function module, the motor axis executes a braking process with the deceleration speed defined at the *Deceleration* input of the MC_MoveVelocity_MDX function module [2]. The MC_MoveVelocity_MDX function module displays the cancellation of the speed control task by setting the *CommandAborted* output to *TRUE*.

While the *Execute* input of the MC_AxisStop_MDX function module is still set to *TRUE*, another rising edge occurs at the MC_MoveVelocity_MDX function module. Although the braking process was already complete, the motor axis does not move. To start movement, the status "Stopping" must first be activated by resetting the *Execute* input of the MC_AxisStop_MDX function module to *FALSE*. As soon as the condition is fulfilled, the motor axis starts movement with speed control again when a rising edge occurs at the *Execute* input of the MC_MoveVelocity_MDX function module [4].

A braking process is started when a rising edge occurs at the *Execute* input of the MC_AxisStop_MDX function module [5]. However, in this case the MC_MoveVelocity_MDX function module does not set the *CommandAborted* output to *TRUE* because the *Execute* input has already been reset to *FALSE*.

When the *RapidSpeed_MDX1* signal is changed from *TRUE* to *FALSE*, the *Execute* input of the MC_MoveVelocity_MDX function module is reset to *FALSE* [6]. The reset is caused by the comparison module [EQ] in the control program during a control cycle. The rising edge in the following control cycle activates speed control with the new, slower setpoint speed. Accordingly, the motor axis turns at the higher speed again when the *RapidSpeed_MDX1* signal is changed to *TRUE* [7].

The braking process, started by a rising edge at the *Execute* input of the MC_AxisStop_MDX function module, can be cancelled by a rising edge at the *Execute* input of the MC_Stop_MDX function module. Consequently, the braking process is executed using the brake ramp set in the parameters of the drive inverter [8].

5.4 Jog mode of a motor axis without encoder

Task description In this programming example, jog mode is to be performed with two motor axis speeds.

Two digital inputs of the MOVIDRIVE® MDX60B/61B drive inverter are used for the signals *Jog positive* and *Jog negative* for this purpose. The motor axis should turn in jog mode when 24 V DC voltage is applied at one of the two digital inputs. Otherwise, the drive must be slowed. The speed of the motor axis should be able to be switched between two values using another digital input.

The drive is operated by using only the digital inputs. The MOVI-PLC® controller evaluates the inputs of the drive inverter and controls the speed of the motor axis.

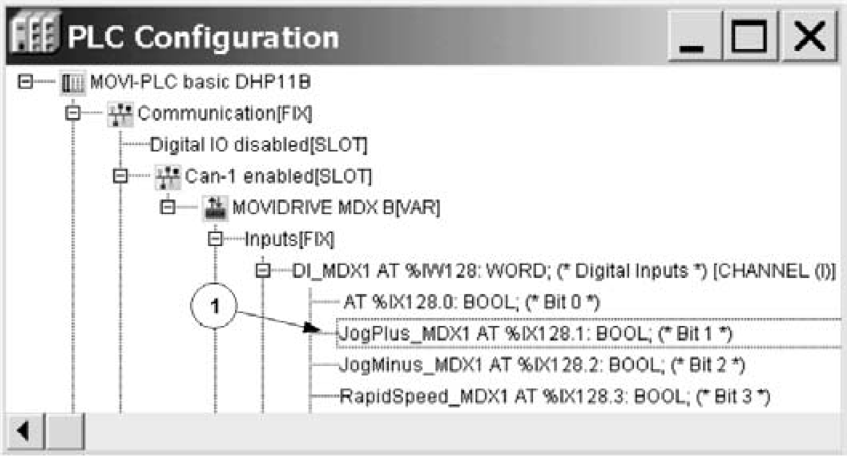
Programming **Unchanged steps**

Program steps 1, 3, 4 and 7 as described in the previous programming examples "Positioning a motor axis" and "Speed control of a motor axis with encoder."

Note:

You do not have to program step 5 because the MC_Power_MDX function module is not used for motor axes without encoders.

Step 2 **Set the controller configuration**



PLC Configuration

```
MOVI-PLC basic DHP11B
    Communication[FIX]
        Digital IO disabled[SLOT]
        Can-1 enabled[SLOT]
            MOVIDRIVE MDX B[VAR]
                Inputs[FIX]
                    DI_MDX1 AT %IW128: WORD; (* Digital Inputs *) [CHANNEL (I)]
                        AT %IX128.0: BOOL; (* Bit 0 *)
                        JogPlus_MDX1 AT %IX128.1: BOOL; (* Bit 1 *)
                        JogMinus_MDX1 AT %IX128.2: BOOL; (* Bit 2 *)
                        RapidSpeed_MDX1 AT %IX128.3: BOOL; (* Bit 3 *)
```

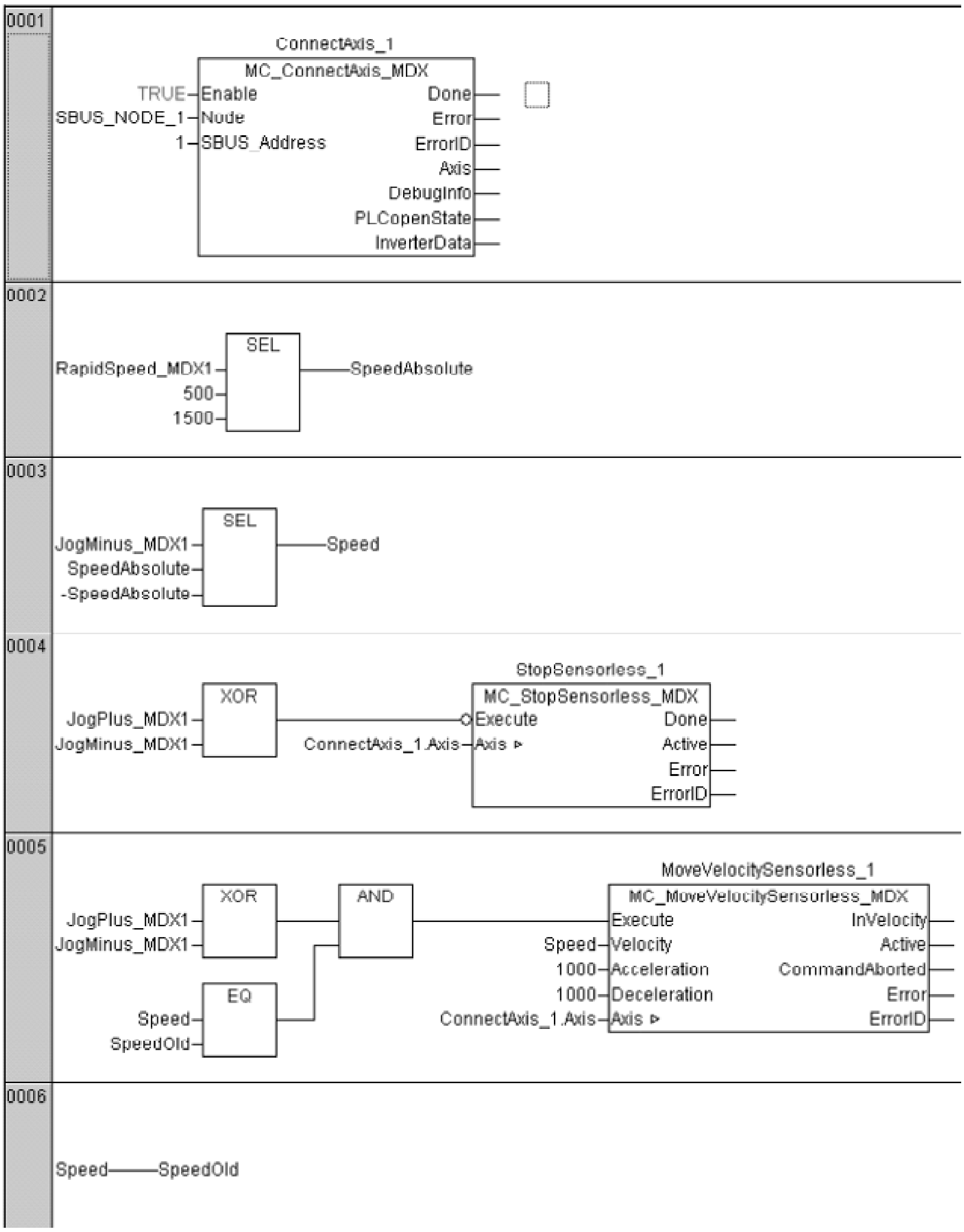
20091AXX

In the controller configuration, assign the identifiers

- [JogPlus_MDX1]
- [JogMinus_MDX1]
- [RapidSpeed_MDX1]

to the digital inputs of the MOVIDRIVE® MDX60B/61B drive inverter as shown in the screenshot [1].

Step 6 **Programming jog mode**



20092AXX

Create the program shown in the diagram in accordance with the procedure described in the previous programming examples.

Step 8

Testing the program

Execute the program by activating the digital inputs as required.

Warning:

Depending on the terminal assignment, the drive inverter status and the control program, the motor axis can begin to move immediately after the MOVI-PLC® controller has been started. To prevent risk of injury, keep sufficient distance from all moving parts.

Turn on the MOVIDRIVE® MDX60B/61B drive inverter if it was not already switched on in step 1 in connection with the MOVI-PLC® controller.

Start the MOVI-PLC® controller by clicking on the menu item [Online] / [Run].

Deactivate the controller inhibit by applying 24V DC voltage at the *DI00* input "/Controller inhibit."

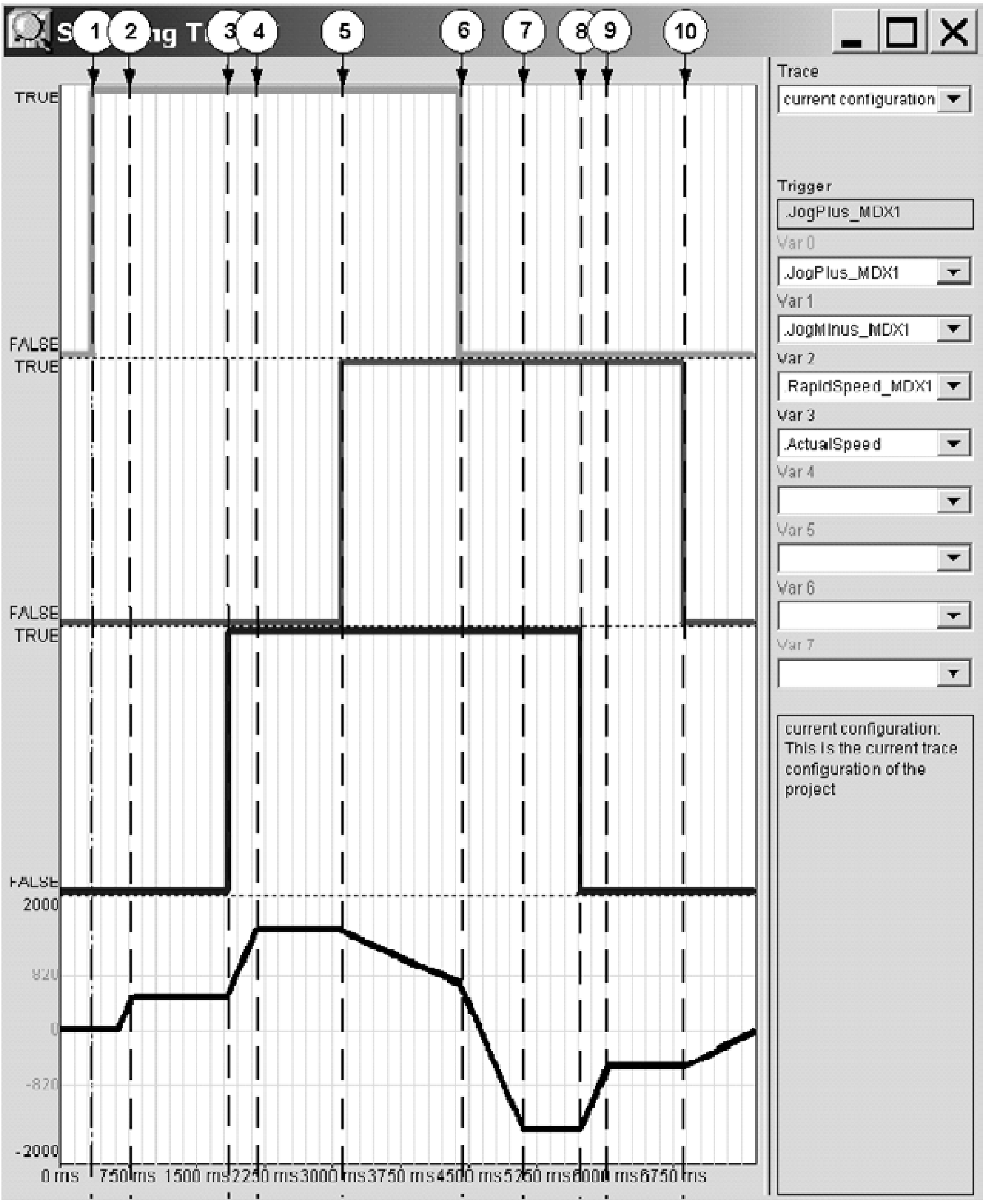
Start jog mode of the motor axis by applying 24V DC voltage at one of the inputs *DI01* or *DI02* of the drive inverter.

Programming is correct when

- the motor axis turns in a positive (clockwise) or negative (counterclockwise) direction when 24V DC is applied at input *DI01* or *DI02*
- the absolute value of the motor speed switches between 500 1/min and 1000 1/min when the voltage is switched between 0V and 24V DC at the *DI03* input.
- the motor axis is decelerated by applying the 24 V DC voltage to both the inputs *DI01* and *DI02* or by removing the voltage from both.

For more details on the behavior of the MOVI-PLC® controller and the connected MOVIDRIVE® drive inverter in this example, refer to the following section "Trace recording."

Trace recording



Trace: current configuration

Trigger: .JogPlus_MDX1

Var 0: .JogPlus_MDX1

Var 1: .JogMinus_MDX1

Var 2: .RapidSpeed_MDX1

Var 3: .ActualSpeed

Var 4:

Var 5:

Var 6:

Var 7:

current configuration: This is the current trace configuration of the project

20093AXX

When a rising edge occurs at the *JogPlus_MDX1* signal, the motor axis starts speed control using the speed specified by the *RapidSpeed_MDX1* signal [1].

When the *RapidSpeed_MDX1* signal changes from *FALSE* to *TRUE*, the motor axis increases the speed to the higher of the two values [3].

If both signals *JogPlus_MDX1* and *JogMinus_MDX1* are set to *TRUE* at the same time, the XOR operation, in conjunction with the negation in the controller program, causes a rising edge at the *Execute* input of the MC_StopSensorless_MDX function module. The braking process is started [5].

For axes without encoders, a braking process triggered by a rising edge at the *Execute* input of the MC_StopSensorless_MDX function module can be cancelled. The cancellation is executed when a speed-controlled movement is called up by a rising edge at the Execute input of the MC_MoveVelocitySensorless_MDX function module [6]. In this example, the required rising edge is generated by the XOR operation in the [0005] network when the *JogPlus_MDX1* signal is switched from *TRUE* to *FALSE*.

When the *RapidSpeed_MDX1* signal switches from *TRUE* to *FALSE*, the motor axis is slowed to the lower of the two speeds [8].

Resetting both the signals *JogPlus_MDX1* and *JogMinus_MDX1* to *FALSE* starts the braking process in the same way as setting both signals to *TRUE* as described above [10].

6 Index

Gearmotors \ Industrial Gear Units \ Drive Electronics \ Drive Automation \ Services

How we’re driving the world

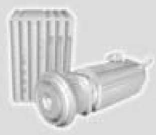
With people who think fast and develop the future with you.

With a worldwide service network that is always close at hand.

With drives and controls that automatically improve your productivity.

With comprehensive knowledge in virtually every branch of industry today.

With uncompromising quality that reduces the cost and complexity of daily operations.



With a global presence that offers responsive and reliable solutions. Anywhere.

With innovative technology that solves tomorrow’s problems today.

With online information and software updates, via the Internet, available around the clock.

SEW-EURODRIVE
Driving the world

SEW
EURODRIVE

SEW-EURODRIVE GmbH & Co KG
P.O. Box 3023 · D-76642 Bruchsal / Germany
Phone +49 7251 75-0 · Fax +49 7251 75-1970
sew@sew-eurodrive.com
→ **www.sew-eurodrive.com**