TM1110

Integrated Motion Control – Axis Groups

PERFECTION IN AUTOMATION
www.br-automation.com

Prerequisites and requirements

Training modules	TM410 - Working with Integrated Motion Control TM440 - Motion Control: Basic Functions
Software	Automation Studio 4.2 Automation Runtime 4.08 mapp Technology 1.00.0
Hardware	-

Table of contents

1 Introduction

Axis groups allow you to group multiple axes and control them in the application program using a single function block. This includes preparatory actions, such as switching axes on, as well as performing actual movements and retrieving information from the individual axes in a group.

Path-controlled movements involving multiple axes can be coordinated centrally from the controller using setpoint generation or performed independently on the individual drives.

Axis groups allow you to group multiple axes and control them in the application program using a single function block.

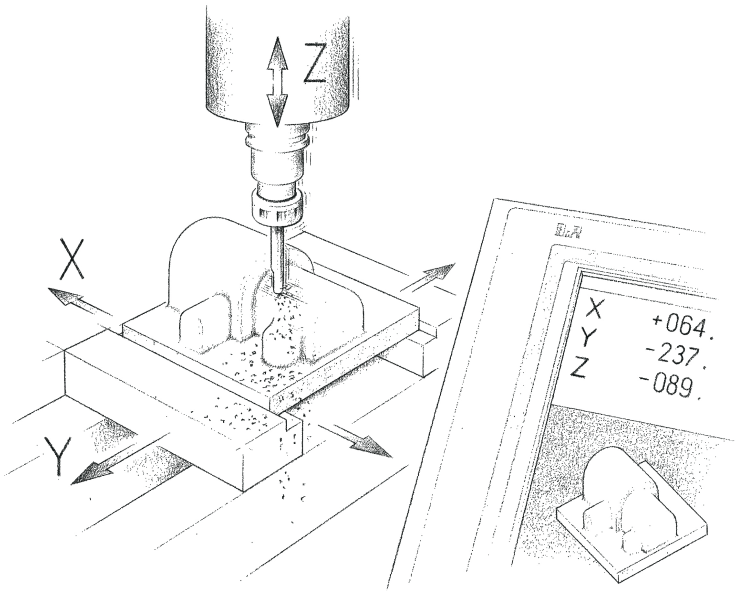


Figure 1: CNC milling machine

1.1 Learning objectives

With the aid of selected exercises, this training module will guide you through the process of creating, operating and programming an axis group in Automation Studio.

- You will become familiar with the concept of Generic Motion Control and understand how the respective libraries work together.
- You will know where axis groups can be used and what benefits they offer.
- You will learn how to set up an axis group in Automation Studio and create your own project using an axis group.
- You will learn the procedure for commissioning an axis group and how to diagnose the group's behavior.
- You will learn axis group programming through using mapp technology and you will be able to create an axis group application.
- You will get to know the behavior of the PLCopen function blocks and will be able to expand axis group applications with it.
- You will learn how to add axes to an existing axis group.

2 The Generic Motion Control concept

2.1 The B&R drive solution

The range of B&R products spans all of the common types of drive technology. Depending on the manufacturing process and technical requirements at hand, many different types of drive concepts can be used for automation.

All components belonging to the axis group are networked with the controller that runs the machine application via fieldbuses.

The controller, the process visualization and the various drive technologies can be mixed homogeneously.

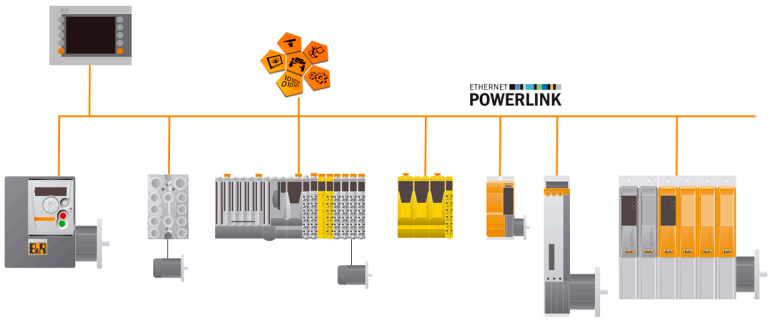


Figure 2: B&R hardware topology

Regardless of whether a stepper or servo motor is needed, the drive can always be connected to the other control components via POWERLINK and be used in an axis group.

2.2 Automation Studio

Automation Studio combines the planning and design of the logic, HMI, safety and motion applications. Built-in diagnostics and commissioning interface facilitates project planning design and helps when commissioning the motion components. All settings and parameters are stored in the Automation Studio project.

Figure 3: Automation Studio

2.3 Generic Motion Control

A look at Generic Motion Control makes it clear how flexible the options are for combining and implementing motion components. In addition, a comprehensive software tool can be used to develop all of the different types of motion applications.

The fundamental requirements placed on a CNC machine include the various electrical properties of the drive system, as well as requirements such as simple speed parameters, position parameters and axis couplings, as well as the ability to perform path-controlled movements via a uniform software interface.

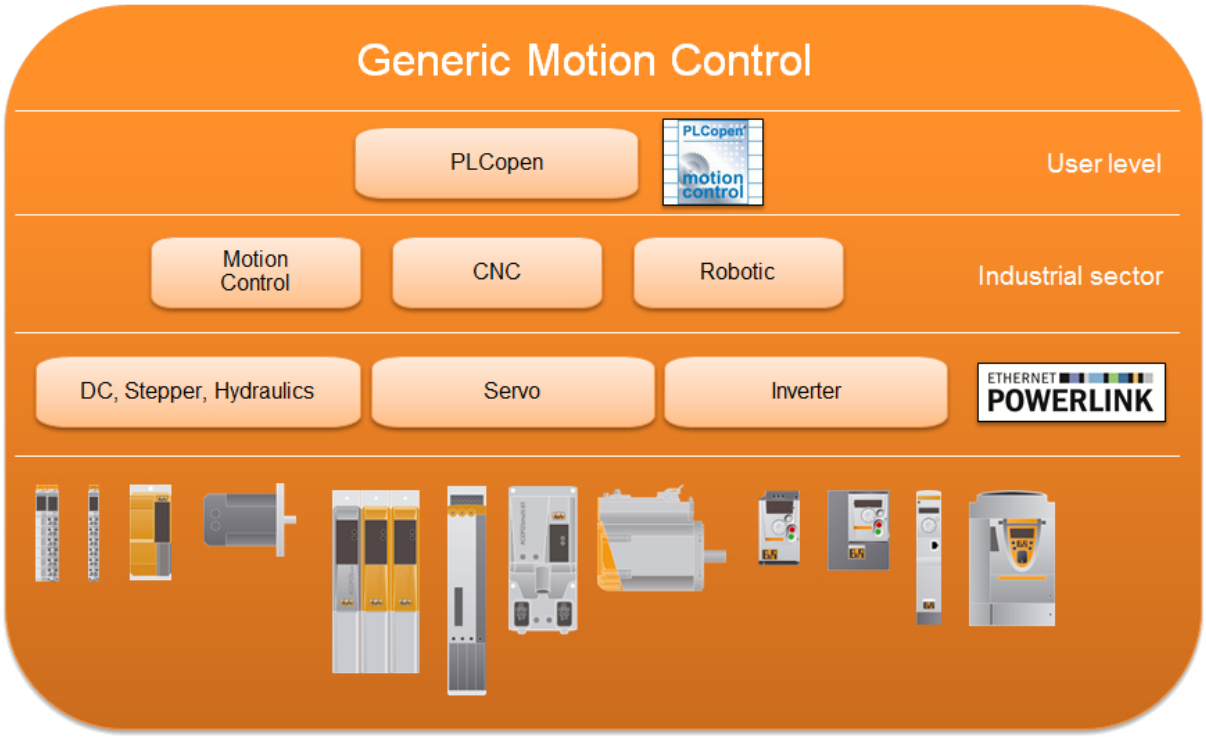


Figure 4: Generic Motion Control

All individual axes and axis groups are controlled using PLCopen-compliant function blocks. It is therefore possible to use the same application program for recurring position sequences, irrespective of the drive hardware used.

Axis groups which control a robot or a CNC machine and individual axes which control converters for step or servo motors, frequency inverters and even hydraulic axes can thus be operated in the same way and combined in any combination.

2.4 Libraries

The GMC libraries are what allow axis group functionality and the user interface to be created independently of one another. There are separate libraries for group implementation and the user interface. A manager library provides the connections between them.

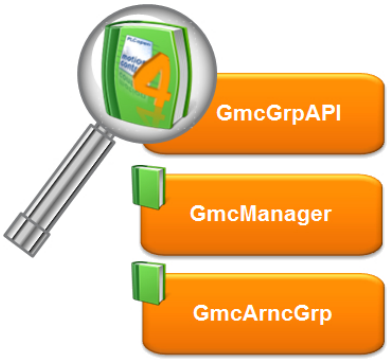


Figure 5: GMC libraries

Library	Description
GmcManager	Serves as the basis for all other GMC libraries, providing the connections between the implementation and the user interface.
GmcGrpAPI	Contains PLCopen function blocks for controlling axis groups
GmcArncGrp	Axis group implementation based on ARNC0

Table 1: GMC libraries

GMC Libraries \ GmcArncGrp \ Technical information \ Dependencies of individual libraries

The latest version of the GMC libraries can be downloaded using the upgrade dialog box in Automation Studio. They then need to be added to the project as a library.

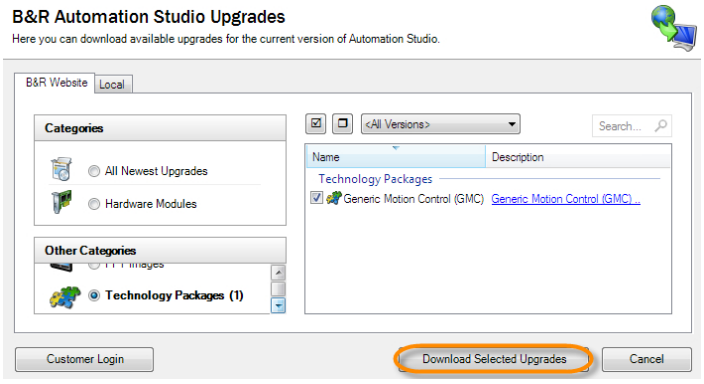


Figure 6: Generic Motion Control technology package

3 Axis group

An axis group brings together multiple axes to simplify the application program. A path generator can be used to perform a path-controlled movement with the axes in a group. Path-controlled movements are required in order to implement CNC and robotics applications.

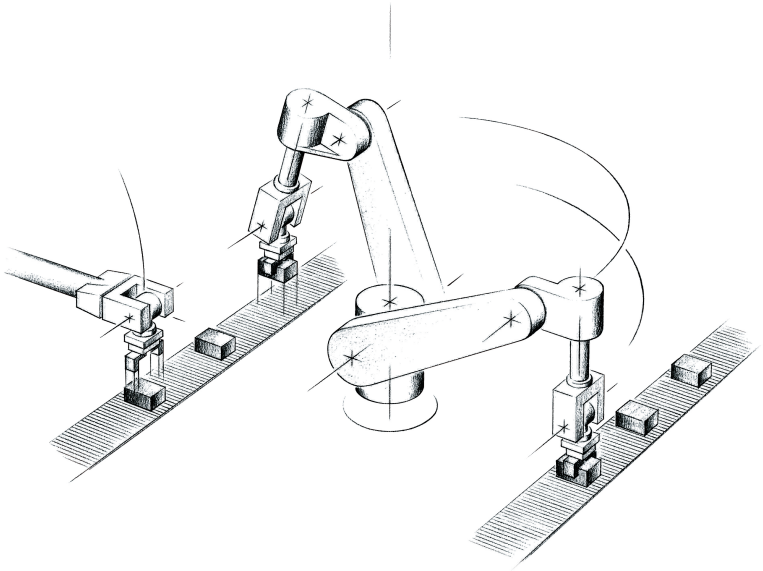


Figure 7: Robotics application

The path generator is described in greater depth in training module "TM1111 - Integrated Motion Control - Path-Controlled Movements."

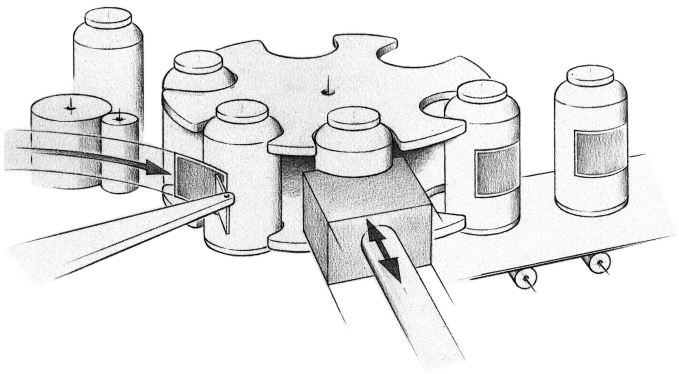


Figure 8: Labeling machines

For many applications it is necessary to assemble different axes in one axis group so that, for example, they can be switched on with just one command or to enable simple error handling for all axes within the axis group.
However, in most cases, only selected axes in the axis group need to be operated using the central setpoint generation so that they are regulated using path-control.
Axes within the axis group that are not included in the path planning can be moved with single axis commands independently of path-controlled movements.

Axis types

An axis group generally consists of various axes which can be subdivided into different types of axes.

Axis type	Path-controlled	Description
Path axis		Path axes are controlled by the path generator. The combination of all the path axes produces the path in space.
Slave axis		Auxiliary axes are also controlled by the path generator. However, they do not contribute to the path in space.
Independent axis		Independent axes are not controlled by the path generator and do not receive a setpoint from the controller. Only administrative function blocks act on these axes.

Table 2: Axis types

A more detailed description of axis groups and axis types, along with sample applications, can be found in the Automation Studio help system:

Motion \ Reference manual \ ARNC0 \ GMC libraries \ GmcGrpAPI \ Technical information \ What is an axis group?

4 Axis group configuration

The characteristics of the individual axes within the axis group can be configured using the axis group configuration. The following points illustrate how an axis group is created and configured in an Automation Studio project.

4.1 Your first axis group project

An X/Y table serves as a first axis group example of a machine tool for which it is possible to control the X and Y axes with path-control.

The ARS is used as the target system in this example.

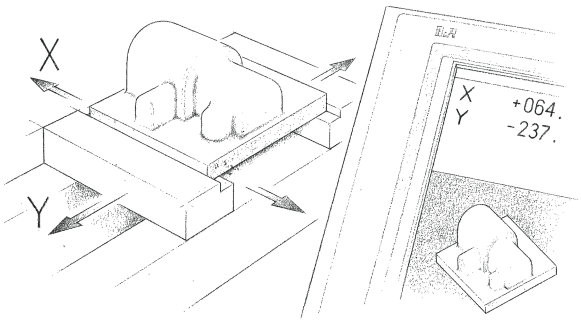
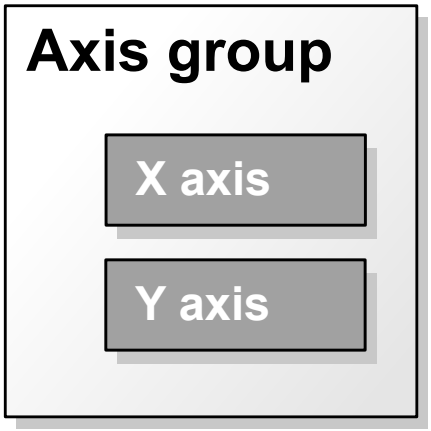


Figure 9: CNC axis group application: X/Y table of a machine tool

What will we need?



We'll need the following components:

- 2 axes (X axis & Y axis)
- 1 axis group

Before a movement can be started, the axes must be activated and homed.

Figure 10: X, Y axis group

Exercise:

1) Create an Automation Studio project
2) Insert motion components
3) Create a CNC program
4) Operate in NC Test

Motion \ Reference manual \ ARNC0 \ GMC libraries \ GmcArncGrp \ Technical information \ Creating an axis group

"Commissioning and diagnostics" provides more detail on how to operate the axis group using NC Test (Section 5.1 "NC Test" on page 13).

Always operate the NC Test of the axes in parallel mode!
If NC Test is enabled in exclusive mode, access to the axis will be blocked by the application program. In parallel mode, parallel control of an axis is possible via NC Test and the application program.

4.2 Axis group reference

Each axis group needs a unique group reference in order to establish the relationships between the axis group created in Automation Studio and the function blocks for programming.

Motion \ Reference manual \ ARNC0 \ GMC libraries \ GmcArncGrp \ Technical information \ Axis group reference

4.3 Axis configuration

The behavior of an axis within a group must be defined. The CNC init parameter table is used for this purpose.

Motion \ Reference manual \ ARNC0 \ GMC libraries \ GmcArncGrp \ Technical information \ Axis configuration

4.4 Unit system

The system of units for the axis group is mostly configured in millimeters or inches, even if the accuracy of the axis is substantially higher (for example, the axis resolution may be 1/1000mm). The units of the axes in an axis group correspond to the PLCopen units.
The PLCopen units factor is used to convert the axis units into PLCopen units.

Axis position on drive = PLCopen factor * PLCopen position

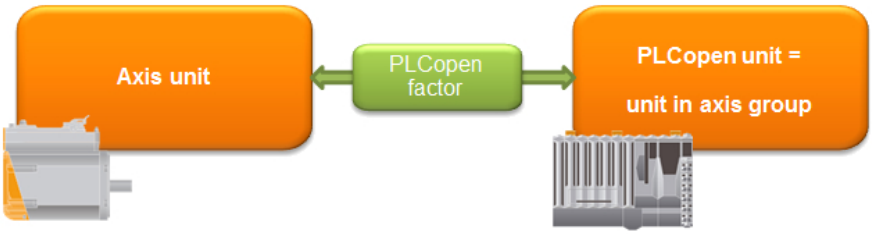


Figure 11: Unit system

The axis group units used in the application program are converted to the high-resolution axis unit using the PLCopen unit factor.

Motion \ Reference manual \ ARNC0 \ GMC libraries \ GmcGrpAPI \ Technical information \ Unit system

5 Commissioning and diagnostics

This section sets out possibilities for how an axis group can be put into operation, step by step, with the aid of functions made available by the Automation Studio, and how these functions can be used for diagnostics.

5.1 NC Test

NC Test can be used to operate the axis group without an application program. Path-controlled movements of the axis group can be performed and diagnosed in this window.

NC Test functions

- Command execution
- Parameter modification
- Watch function
- Trace function
  - Cyclic trace (see 5.2.1 "Cyclic trace" on page 14)
  - Network command trace (see 5.2.2 "Network command trace" on page 14)

Figure 12: Commissioning and diagnostics

NC Test can be operated in parallel mode and in exclusive mode. Exclusive mode does not allow axis group movements to be started from application programs. This prevents interference between NC Test and the application program.

After selecting the axis group in the mapping table, NC Test can be called in the shortcut menu under <Open / Test>. A request for the mode that NC test should be started in occurs.

Motion \ Diagnostics \ NC Test

5.2 NC Trace

After selecting the axis group in the mapping table, NC Trace can be called in the shortcut menu under <Open / Trace>. This opens a window showing the cyclic trace data. Right-click on the NC Trace window to switch to the trace network command.

In Automation Studio, NC Trace is shown as a subsection of the NC Test window, but it can also be opened separately as a window of its own.

5.2.1 Cyclic trace

The cyclic trace can be used to record and analyze the movements (e.g. positions and speeds) of the path and the axes.

The data is recorded on the controller. Once it has been recorded, the data can then be loaded into Automation Studio and displayed there.

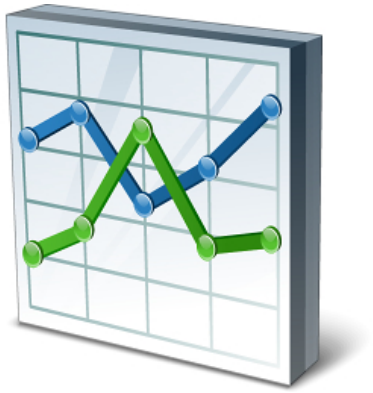


Figure 13: Cyclic trace

Motion \ Diagnostics \ NC Trace

5.2.2 Network command trace

The network command trace records axis communication. It also records commands sent to axes and axis groups, PLCopen state changes as well as errors. This function allows you to analyze the order in which commands are issued by the code. It is therefore a quick and easy way to locate errors.

The data is recorded continuously and can be loaded into Automation Studio from the controller and displayed for diagnostic purposes.

Figure 14: Command trace

Motion \ Diagnostics \ Network command trace

6 mapp technology

With mapp technology[1], we offer users an easy-to-use interface for implementing comprehensive functionality. Many complex operations, such as loading and saving recipe data, controlling a drive axis and recording process values, are carried out using easy-to-use mapp technology components.

Figure 15: mapp technology logo

mapp technology unites configuration and programming. Functionality is implemented in the application program using standard libraries. In addition, mapp provides configuration interfaces that allow the functionality of mapp components to be influenced without having to configure their implementation in the application software.

Application layer - mapp technology

- Concept
- Getting started
- Components

6.1 Instructions for using mapp technology components

The following steps have to be carried out when using a mapp component for the first time.

- Go to the Configuration View
- Add the "mapp" technology package from the toolbox
- Add the standard configuration for the mapp component being used from the toolbox
- Rename the MpLink in the Configuration View as needed
  The Configuration View should look like this:

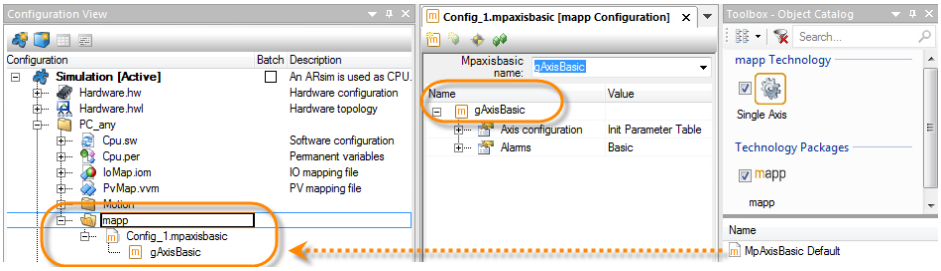


Figure 16: Representation of the Configuration View with mapp technology package and the standard configuration for the MpAxisBasic function.

The MpLink from the Configuration View is transferred to the function block in the program with the ADR() function. This establishes the connection from the configuration to the programming code of the mapp component.

[1] mapp technology stands for "Modular APPlication technology".

Application layer - mapp technology \ Concept \ Component design \ Adding mapp components

Calling mapp components

The mapp component function blocks should be called in every controller cycle. When using high-level programming (Ansi C, ST, etc.), it is advisable to call all mapp components at the end of the program.

Every mapp function block has an "Enable" input. This input is used to enable the mapp component, which causes the configuration for the respective mapp component to be loaded automatically. Successful initialization of the mapp component is displayed on the "Active = TRUE" output.

Application layer - mapp technology \ Concept \ Component design \ Using mapp components

Download behavior

The controller has to be restarted using the standard settings in Automation Studio each time the program has been transferred. It is advisable to use the "Copy mode" transfer method during the implementation phase of the drive application. By doing so, it is no longer required to restart the controller a few of the times. The configuration for the type of transfer is opened using the shortcut menu of the active configuration in Configuration View. The transfer method is set in the Transfer tab using the Advanced button.

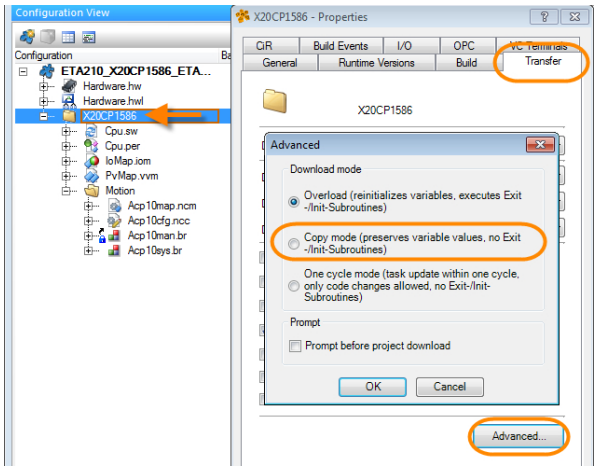


Figure 17: Configuration of the transfer method in the Properties window in Configuration View

Application layer - mapp technology \ Concept \ Component design \ Using mapp components

Real-time operating system \ Target systems \ SG4 \ Download

Configuration Files

There is a configuration available for every mapp component. The configuration is created and modified in the Configuration View in Automation Studio, the WebXs web-based interface or the application program. Additional information about mapp configuration can be found in the Automation Studio help system.

Application layer - mapp technology \ mapp \ Concept

- Component design \ Adding mapp components
- Configuring components

6.2 Diagnostic options for mapp technology components

mapp technology components can be monitored and diagnosed via several different methods. The following is a list of the diagnostic options in Automation Studio, in web-based diagnostics and in the visualization application.

Programming languages in monitor mode

In many cases, the first access is monitor mode during application software programming. The values of the process variables are visible in context directly with the program code. All mapp technology components have the "Error" and "StatusID" outputs that can be used to perform initial diagnostics.

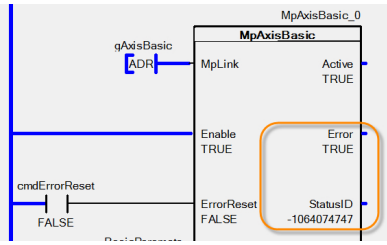


Figure 18: Ladder Diagram program in monitor mode

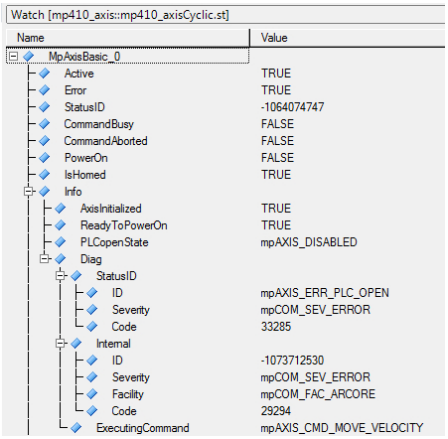


Figure 19: Instance variable of the MpAxisBasic function block in the Watch window

Watch window

The Watch window is opened in Logical View using the program shortcut menu or in the software configuration by selecting **<Watch>** . The instance variables of the function blocks used are added using the toolbar or shortcut menu. For example, the Error, StatusID, CommandBusy outputs and the information structure can help to diagnose the current state. A description of these parameters as well as error numbers is available to read in the description of the respective function block.

Logger

In the case of an error, additional information from the mapp technology component is added into the logger file with the name "$mapp". The error number can be searched for directly in the Automation Studio help system or called by pressing the **<F1> key** . Further information is available in the details section of the highlighted logger entry. For example, if there is a PLCopen error, the affected function and the cause of the error are described clearly in the details section of the logger entry.

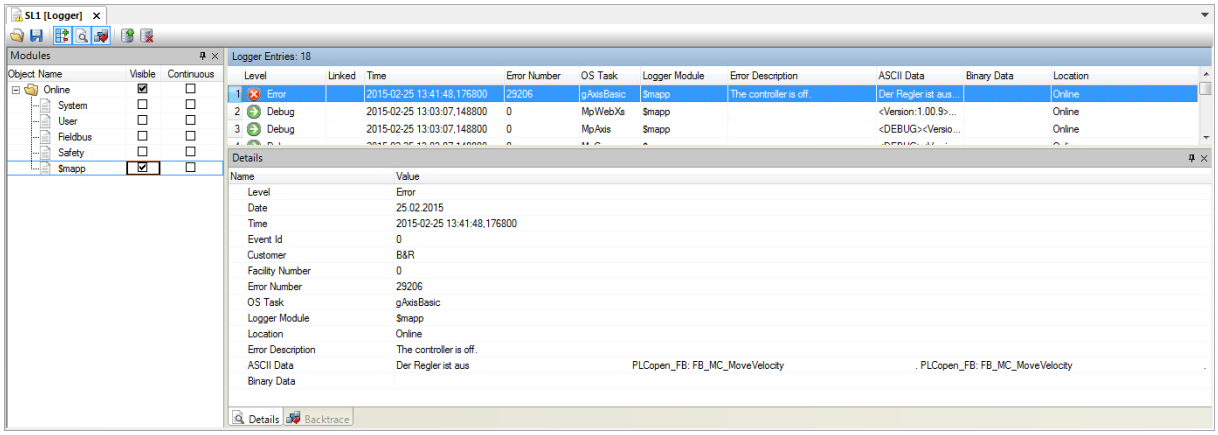


Figure 20: PLCopen error in the logger window

Trace

With the Automation Studio trace function, values of process variables are recorded in real-time and saved. Above all, the timing of input parameters and status variables can be visualized well.

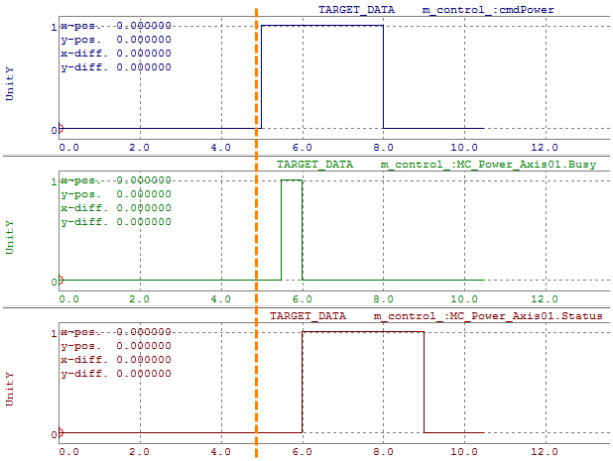


Figure 21: Switching on the controller with cmdPower: the time relationship between commands and status information

System Diagnostics Manager

The "Application Status" button can be called using the System Diagnostics Manager. This directly opens WebXs for mapp technology. Furthermore, saving the logger data for the mapp components in SDM is enabled. There are also basic diagnostics for drive axes available. The SDM can be embedded directly into a Visual Components application using the HTML control.

mapp WebXs

Using mapp technology WebXs, all mapp components being used are shown on a web-based interface. Configuration of the mapp components and alarms is also offered in addition to diagnostics via the instance variables for the components[2].

Integration of Visual Components using the MpAlarm component

The mapp technology components have predefined alarms. User-specific alarms can also be configured. The output of mapp alarms in the Visual Components alarm system is enabled using the MpAlarm component.

2 Whether a web-based configuration can be carried out depends on the component used.

Integration of Visual Components using the MpComLoggerUI component

Event management features of mapp technology store all events in the logger. These logger entries can be easily integrated in Visual Components using the MpComLoggerUI component. Filter functions make it possible to search for individual mapp components, certain error numbers or event types. Additional programming is therefore not required to filter out mapp logger entries.

Diagnostics and service \ Diagnostic tool \

- Logger
- Watch window
- Monitors \ programming languages in monitor mode
- Trace
- System Diagnostics Manager

Application layer - mapp technology \

- WebXs
- Components \ Infrastructure \
  - MpAlarm - Support for alarm management
  - MpCom - mapp management \ function blocks \ MpComLoggerUI
- Diagnostics \ Logger window

7 Integrating an axis group in the control program

There are different components for controlling axis groups with mapp depending on the machine type. These range from CNC systems with different numbers of axes to robots of different types.

Figure 22: mapp components for controlling axis groups

The following chapter shows, step by step, how an application program for controlling an axis group is created.

The MpCnc2Axis mapp component is used for the 2-axis CNC machine. Two axes which are operated with the MpAxisBasic mapp function block are required as the basis for this component.

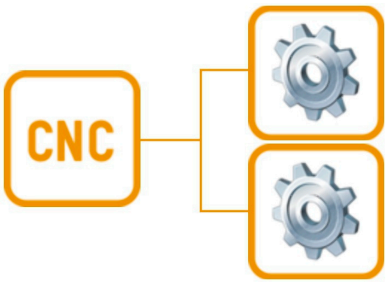


Figure 23: mapp 2-axis CNC

Task: Creating mapp-control of the axes of an X/Y-CNC machine using MpAxisBasic

1. Go to the Configuration View
2. Add mapp technology MpAxisBasic component from the toolbox for the X and Y axes
3. Add the new "cnc_ctrl" Ladder Diagram program
4. Add MpAxisBasic function blocks
5. Connect the "MpLink", "Enable", "Axis" and "Parameters" inputs
6. Transfer project and test axes by switching on "Power"

7.1 The MpCnc2Axis component

The MpCnc2Axis mapp technology component offers standard functions for controlling, configuring and diagnosing 2-axis CNC machines. The following basic functions of a CNC machine are covered by the MpCnc2Axis function block:

- Group preparation
- CNC movements
- Error handling
- Jog mode
- Single-step operation

Figure 24: mapp CNC

Figure 25: mapp components are based on open standards, technology functions and libraries

The MpCnc library is based on the function blocks of the GmcGrpAPI library, just as the MpDelta, MpRoboArm and MpScara libraries are. Both libraries are therefore compatible with one another and can be jointly implemented in applications.

All mapp components are based on open standards, technology functions and libraries that can be directly used by the user in the application.

Details regarding the used function blocks and functionality can be taken from the corresponding section in the Automation Studio help system.

7.2 Creating a program and adding MpCnc2Axis

Now it is necessary to expand the Automation Studio project by calling of the MpCnc2Axis function block. In a preparatory step, the project in the Configuration View is expanded by adding the mapp technology package. Next, the standard configuration for the MpCnc2Axis component is added to the Configuration View. The MpCnc library is added to the Logical View.

At the end, the MpCnc2Axis function block is added to a program and connected to the MpLink which was already set up in the Configuration View.

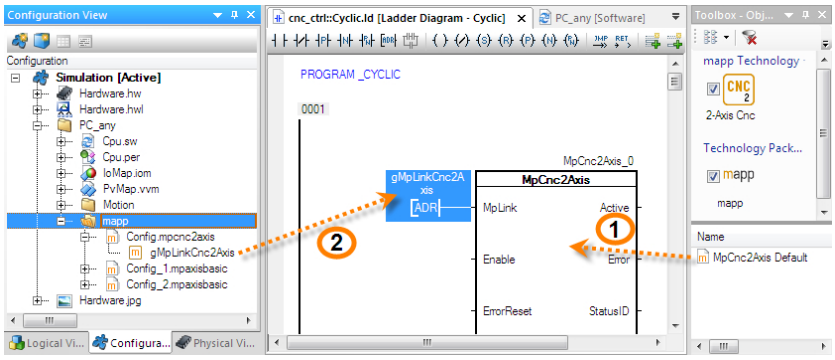


Figure 26: (1) Add MpCnc2Axis from the toolbox and (2) transfer the MpLink address from the Configuration View

Application layer - mapp technology \ Concept \ Component design \ Adding mapp components

Exercise: Adding the mapp technology configuration for MpCnc2Axis

Now expand the Automation Studio project by adding the mapp technology package and then insert the standard configuration for the MpCnc2Axis component. Import the MpCnc library, next add the MpCnc2Axis function block into the cnc_ctrl program and then connect the MpLink from the Configuration View using the ADR() function.

1) Go to the Configuration View.
2) Add the mapp technology package from the toolbox.
3) Add the MpCnc2Axis standard configuration in the Configuration View.
4) Add the MpCnc library in the Logical View.
5) Add MpCnc function block from the toolbox into the "cnc_ctrl" into the Ladder Diagram program.
6) Assign MpLink from the Configuration View to the MpCnc2Axis function block using the ADR() function.

7.3 Connecting the axis group reference and the movement parameters

It is necessary to transfer the axis group reference and the movement parameters so that an axis group can be accessed and movements can be performed.

Using the axis group reference

A global process variable of type MCAxesGroupType_Arnc is set up. The name of the process variable must match the name of the group in the NC mapping table. In this way, the connection between the axis group and the application program can be created.

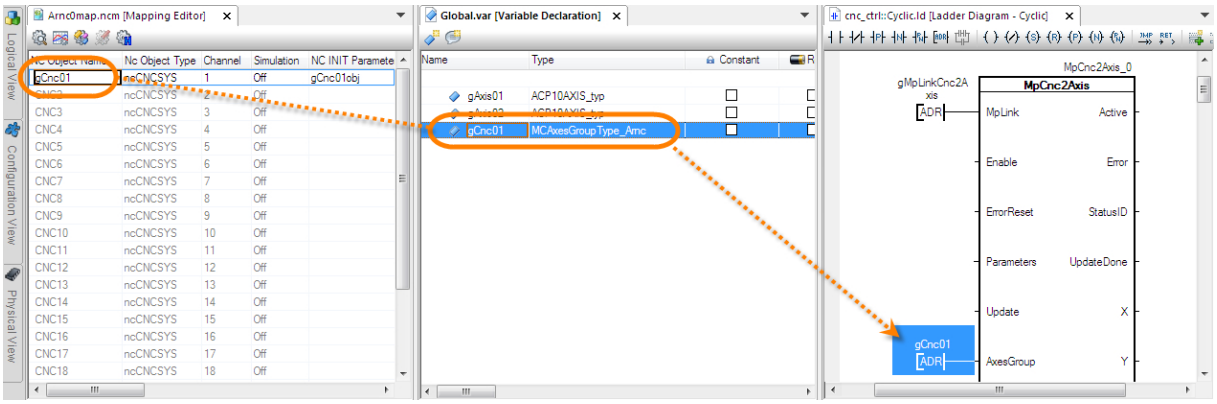


Figure 27: Connecting the axis group reference

The address of the axis reference is specified for all function blocks using the "AxesGroup" parameter. In Ladder Diagram, either the address function (ADR) or an address contact can be used. Access to the axis group happens in the same way for all other function blocks.

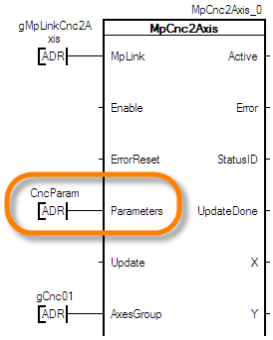


Figure 28: Parameter data structure transfer (MpCnc2AxisParType)

Transferring CNC parameters

For the MpCnc2Axis function block, it is necessary to transfer a data structure with the parameters. The data structure is preinstalled with the standard values. The following parameters can be transferred:

- CNC program name
- Initialization subroutine
- CNC single-step
- Jog parameter
- Tool parameters

What you should be taken into consideration is that an initialization value may not be entered in order to receive the standard values of the data structure when creating data structures in the variable declaration.

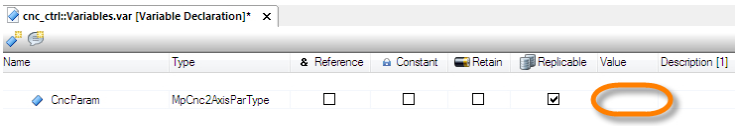


Figure 29: Empty "Value" field to initialize the structure with standard values

Application layer - mapp technology \ Components \ Mechatronics \ MpCnc - CNC system controller \ Function blocks \ MpCnc2Axis

Motion \ Reference manual \ ARNC0 \ GMC libraries \ GmcArncGrp \ Technical information \ Axis group reference

Exercise: Assigning axis group reference and assign parameters

The existing program is now expanded so that the axis group reference and the data structure with the parameters can be transferred to the MpCnc2Axis function block.

1) Assign axis group reference to the "AxesGroup" input using the address function.

2) Connect the "CncParam" structure with the "Parameters" input.

The "CncParam" structure is based on the MpCnc2AxisParType data type. It is preinitialized with default values.

7.4 mapp hierarchy

In order to successfully put the axis group into operation, a hierarchy has to be created between the MpCnc2Axis and MpAxisBasic mapp components. This linking of the axes with the CNC is generated using the mapp MpComLinkToParent function block from the MpCom library.

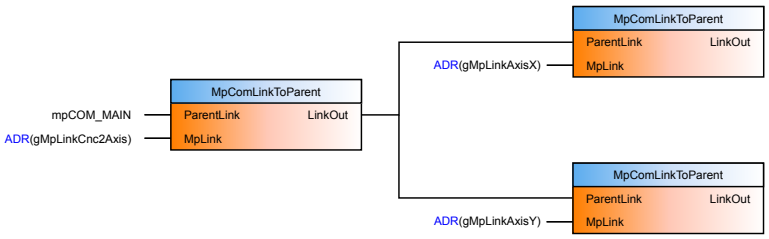


Figure 30: mapp hierarchy with MpComLinkToParent

The MpComLinkToParent function block must be called during the initialization of the application program.

Application layer - mapp technology \ Components \ Infrastructure \ MpCom \ mapp management \ MpComLinkToParent

Application layer - mapp technology \ Components \ Mechatronics \ MpCnc - controller of a CNC system \ Technical information \ CNC hierarchy concept

Create mapp link hierarchy and switch on axis group with MpCnc2Axis

1. Add mapp hierarchy to the initialization part of the "cnc_ctl" program using the MpComLinkToParent function block.
2. Transfer the program to the controller
3. Set the "Enable" input of the MpAxisBasic mapp function blocks to TRUE.
4. Wait until the "Active" output and the "Info.ReadyToPowerOn" output are TRUE.
5. Set the "Enable" input of MpCnc2Axis to TRUE.
6. Switch on the axis group via "Power" on the MpCnc2Axis function block.
7. Observe the status outputs of MpCnc2Axis and MpAxisBasic.

7.5 Function block operation and status evaluation

All mapp function blocks are accessed using uniform operating parameters and return uniform status parameters. This simplifies the application and adds clarity during programming.

Application layer - mapp technology \ Concept \ Component design \ Inputs and outputs

Timing diagrams

In the Automation Studio help system, the timing diagrams show how the function blocks behave in different input states. They explain the functioning of the components and provide assistance with the correct implementation of the application program.

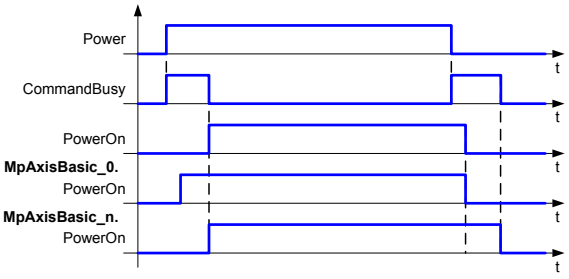


Figure 31: Input "Power" timing diagram at MpCnc2Axis

Application layer - mapp technology \ Components \ Mechatronics \ MpCnc - controller of a CNC system \ Function blocks \ MpCnc2Axis \ Timing diagrams

Status information

In the event of an error, the value of the "Error" output becomes = TRUE. The "StatusID" output contains numerical information that can be searched for in the Automation Studio help system. Additional information about the current status is made available in the "info" output structure. This can be displayed when using WebXs in the browser. Errors are also entered in the logger. This can optionally be shown in Automation Studio or in the System Diagnostics Manager.

The image below shows the MpCnc2Axis component with an expanded "info" structure in WebXs. The "Error" output = TRUE and a value is given on the "StatusID" output. The "info" structure indicates that an axis group error has occurred. The error code is 15131. If you look for this error number in the Automation Studio help system, you will find a description. In this case, the error description points out that an attempt was made to start a nonexistent NC program.

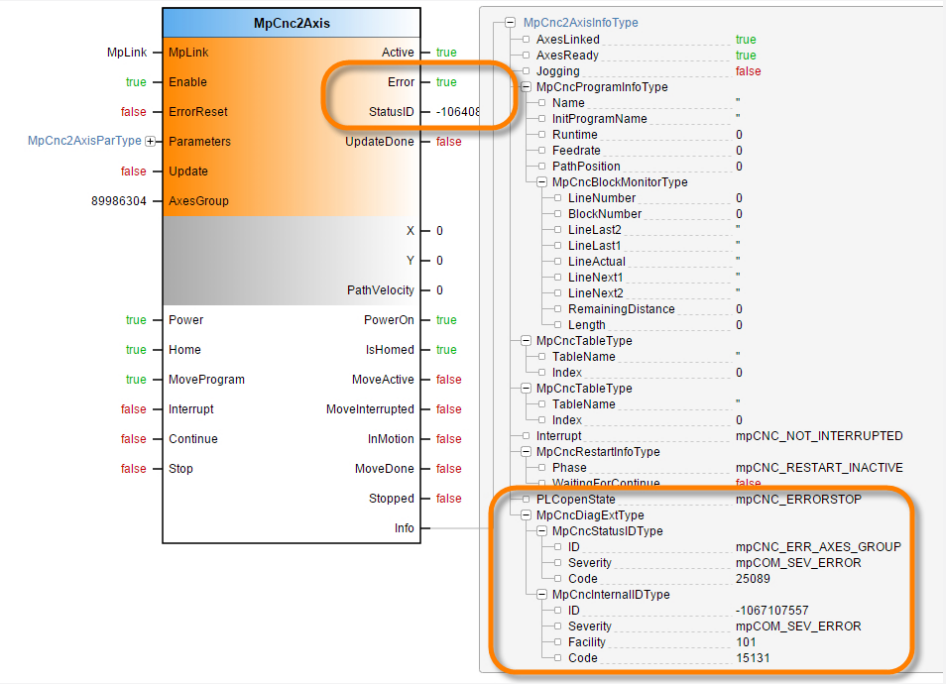


Figure 32: Comprehensive "info" structure - Representation in WebXs

An entry in the Automation Studio logger named "$mapp" and "GMC" is generated as well. Additional details can be read in addition to the error number for identifying the cause of the error.

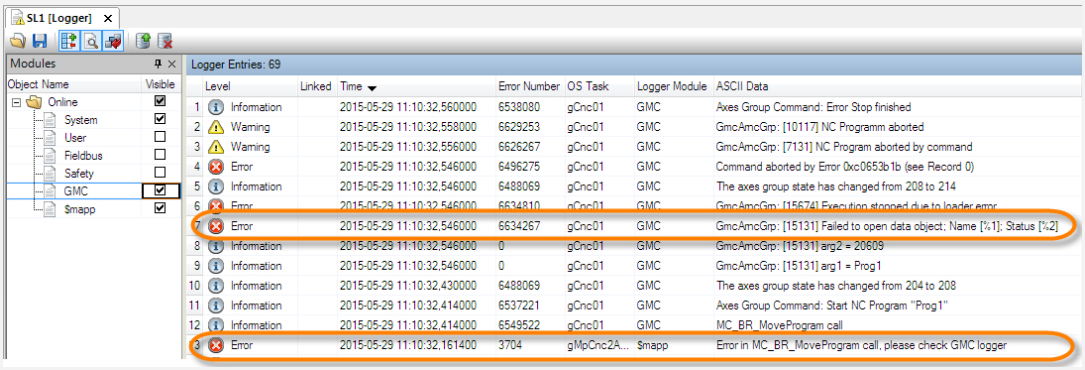


Figure 33: "$mapp" and "GMC" logger modules

The error can be acknowledged via the "ErrorReset" input. After the error has been corrected, the program can be started via a new rising edge on the "MoveProgram" input.

8 Axis group states

The PLCopen states are used for operating an axis group. These states simplify the overview of complex movement procedures and make it easier to handle error situations.

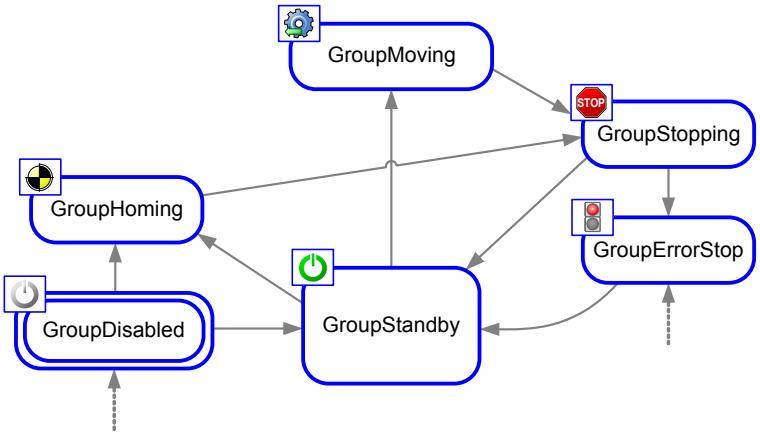


Figure 34: PLCopen axis group state diagram

Transitions between these states can be initiated by calling certain function blocks. The GroupErrorStop state can be entered from any other state when an error occurs in the group on one of the group's axes.

State	Description
GroupDisabled	The group is disabled. No axis group movements can be performed.
GroupStandby	The group is switched on, but no movement is being performed.
GroupHoming	Homing is active for the entire group.
GroupMoving	At least one axis in the group is moving. This movement can be caused by a group movement command or a single-axis movement command.
GroupErrorStop	The group is in an error state. Movements are stopped.
GroupStopping	The group is stopped.

Table 3: PLCopen axis group states

Motion \ Reference manual \ ARNC0 \ GMC libraries \ GmcGrpAPI \ Technical information \ State diagram

The current state of the axis group can read using the "PLCopenState" element of the "info" structure of the MpCnc2Axis function block.

Exercise: Command execution and condition monitoring

Execute different commands and always check the status outputs and the axis group state.

1 Add the MpWebXs library and transfer the project.
2 Check the "info" structure in the Watch window as well as in WebXs.
3 Switch on the axis group and check the status of the axis group.
4 Home the axis group with the "Home" command and check the "IsHomed" status. The command can then be reset.
5 Set the "Override" input to the value 100%.
6 Enter a program name under "ProgramName" in the parameter structure of the MpCnc2Axis function block and start the CNC program using the "MoveProgram" command.
7 Check the status outputs and the axis positions on the MpCnc2Axis function block.
8 Use additional commands such as "Interrupt", "Continue" and "Stop" and observe the behavior of the outputs on the MpCnc2Axis function block.

Exercise: Generating and deleting axis group errors

Attempt to start a nonexistent CNC program and observe the status outputs. Then delete the error via the "ErrorReset" input.

1 Enter an invalid program name under "ProgramName" in the parameter structure of the MpCnc2Axis function block and start the CNC program using the "MoveProgram" command.
2 Check the "Error" output and the "StatusID".
3 Check the "info" structure and the logger entries.
4 Acknowledge the error via the "ErrorReset" input.
5 Recheck the "Error" and "StatusID" outputs as well as the "info" structure.

Exercise: Single-step operation

1 Enable single-step operation using the "SingleStep" input and check the "SingleStepActive" output.
2 Configure the variable trace to plot the "Continue", "MovementActive" and "MoveInterrupted" variables.
3 Start the trace and then the CNC program.
4 After "MovementInterrupted" has switched to TRUE, stop the trace and analyze the record.
5 Observe the "BlockMonitor" and "Interrupt" values of the "info" structure.
6 Start the next movement in the CNC program with "Continue".

9 PLCopen axis group library

The function blocks of the GmcGrpAPI library are used as the basis for controlling the axis groups with mapp technology. The interface for these function blocks of this library is based on the definition of PLCopen. This section addresses what PLCopen is and the advantages brought for the user by using these blocks in conjunction with mapp.

9.1 PLCopen library

The PLCopen function blocks provided by B&R for controlling axis groups are based on the ones specified in the PLCopen Motion Control Part 4 standard. To extend functionality to cover all the needs of an axis group, B&R-specific function blocks are provided in addition to the function blocks specified in the standard.

Whether a function block is a PLCopen standard function block or a B&R-specific function block is indicated by its name: Standard function blocks always include the prefix "MC_" at the beginning of the name, e.g. MC_MoveDirectAbsolute(). B&R-specific function block and expansion names begin with "MC_BR_", e.g. MC_BR_GroupPower().

Figure 35: PLCopen @ B&R

The function blocks for controlling axis groups can be found in the GmcGrpAPI library.

Motion \ Reference manual \ ARNC0 \ GMC libraries \ GmcGrpAPI

9.2 Using the function blocks

This section provides some basic information about using PLCopen function blocks. We will look at how to operate the function blocks as well as the options available for monitoring their operation.

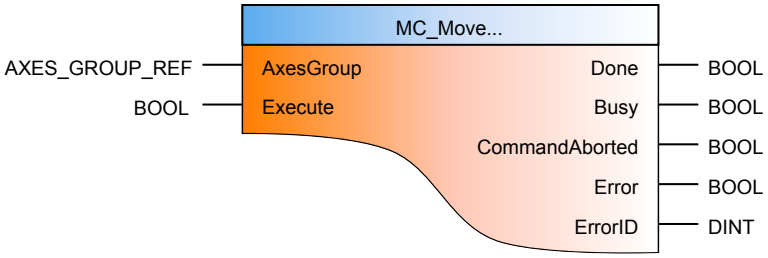


Figure 36: Standard function block parameters

All function blocks are accessed using a uniform set of operating and status parameters. This simplifies the application and adds clarity during programming.

Axis group reference

The axis group reference establishes the link between the axis group and the function block.

Parameter	Description
AxesGroup	Specifies the axis group for which the function block should be used.

Motion \ Reference manual \ ARNC0 \ GMC libraries \ GmcArncGrp \ Technical information \ Axis group reference

Begin execution

There are two basic types of function block, those with an Enable input and those with an Execute input.

Parameter	Description
Enable	The function block is active as long as Enable is set. When Enable is reset, the function block is disabled.
Execute	The function block initiates an action when a rising edge is received at the Execute input. No action is initiated when the Execute input is reset.

Status outputs

The function blocks have the following status outputs. Only one of these outputs can be set at a time.

Parameter	Description
Busy	The function block is active and must continue being called.
Done	The action was completed successfully.
CommandAborted	The action was aborted by another function block call.
Error	The action was aborted due to an error.

Error information

If the Error status output is set, then an error number is provided at the ErrorID output.

Parameter	Description
ErrorID	Information about the cause of the error. A list of possible error numbers can be found in the Automation Studio help system.

Motion \ Reference manual \ ARNC0 \ GMC libraries \ GmcGrpAPI \ *Name of the function block* \ Error numbers

The values of the Done, Command Aborted, Error and ErrorID outputs remain the same until the Execute or Enable input is reset.

If the Execute input is already inactive before these signals arrive, these outputs are set for the duration of one cycle.

Motion \ Reference manual \ ARNC0 \ GMC libraries \ GmcGrpAPI \ Technical information \ Function block interface

9.3 Function blocks

The PLCopen function blocks can be divided into administrative and non-administrative function blocks. Administrative function blocks do not cause a movement when they are called. Non-administrative function blocks start movements when they are called.

The most important function blocks used when working with axis groups are listed in this section.

Figure 37: PLCopen Motion Control logo

A majority of the listed function blocks are used within the MpCnc2Axis mapp component. If a required function is not mapped in the mapp component, then PLCopen function blocks can be easily combined with MpCnc2Axis.

Preparing the group

The following table shows the most important function blocks used to prepare an axis group.

Function block	Description
MC_BR_GroupPower()	Switches on the axis group and the axis controllers
MC_GroupHome()	Homes the axes

Table 4: Important function blocks used for preparation

Executing movements

The following table shows the most important function blocks used to execute movements.

Function block	Description
MC_MoveDirectAbsolute_15()	Performs a direct movement to a defined end position
MC_MoveDirectRelative_15()	Performs a direct movement over a defined distance
MC_MoveLinearAbsolute_15()	Performs a linear movement to a defined end position
MC_MoveLinearRealtive_15()	Performs a linear movement over a defined distance
MC_BR_MoveBlock()	Executes a single interpretable line (block)
MC_BR_MoveProgram()	Executes an interpretable program (CNC program)
MC_GroupStop()	Stops an active movement

Table 5: Important function blocks used to execute movements

Error handling

The following table shows the most important function blocks used to handle errors.

Function block	Description
MC_GroupReadStatus()	Reads the state of the group
MC_GroupReset()	Takes the axis group out of the GroupErrorStop and state and acknowledges all errors
MC_GroupReadError()	Reads the current group error and acknowledges it with a command.

Table 6: Important function blocks used for error handling

An overview of all available function blocks can be found in the Automation Studio help system.

Motion \ Reference manual \ ARNC0 \ GMC libraries \ GmcGrpAPI \ Function blocks

10 Programming

The application program should establish an automatic sequence for controlling the axis group. Sequences are started by actions of the operator. Errors must also be taken into consideration and responded to appropriately.

There are many possibilities for implementing these requirements. On the one hand, you can choose between different programming languages, and on the other, there are different approaches to structuring the software.

Figure 38: A more in-depth look at the application program

10.1 Automating tasks

The MpCnc2Axis function block offers the possibility of controlling the axis group. Processes can be started at a defined point in the program using commands. The function block's status outputs and output parameters provide information about the current status of the axis group.

In order to create a program where certain sequences are executed automatically, the status outputs can be used as an information source to decide which additional commands should be executed. This can be achieved through logical links of the variables or with conditions. Timing diagrams in the Automation Studio help system provide assistance with correct programming.

Application layer - mapp technology \ Components \ Mechatronics \ MpCnc - controller of a CNC system \ Function blocks \ MpCnc2Axis \ Timing diagrams

Exercise: Automatic homing after switching on the axis group

The existing program should now be automated. The MpCnc2Axis and MpAxisBasic components should be enabled immediately after switching on the controller. Additionally, the program must start a homing procedure after successfully being switched on if the group has not yet been homed.

1. After starting up, MpCnc2Axis must be configured automatically and enabled.
2. After the "PowerOn" output is set, a homing procedure must be started.
3. The group must only be homed if it has not yet been done.
4. If the homing procedure has been executed successfully or if an error occurs, then the "Home" command is reset again.

10.2 Uses of control structures

Now a superior machine logic must be created in an additional program. Here actions by the user should trigger certain commands on the axis group and transitions between machine states.

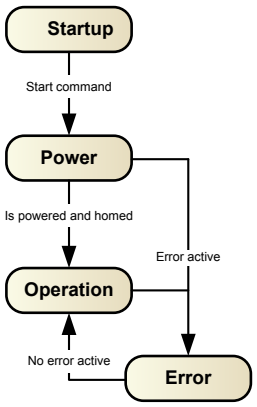


Figure 39: Control structure diagram

The following information from MpCnc2Axis can be used to control the sequences:

- Was the execution of the command successful?
- If not, what error occurred?
- Status of the process:
  - Is the axis group in motion?
  - Has the axis group arrived at its destination?
  - Was homing performed successfully?

This information can be used to control steps in the program sequence. The program will have to respond differently depending on whether or not an error occurs.

The step sequencer is a control structure that is especially well-suited for managing these types of function sequences.

This type of structure allows the implementation of individual steps whose sequence can be determined by the use of a step index.

The necessary commands can be executed in the individual steps of the control structure. Status parameters such as Error, StatusID and the "info" structure can be used to determine the step where the application will continue.

This gives the application a clear structure and opens it up for future expansion.

Exercise: Creating machine logic

Machine logic with operation states must now be created in a new program. This must be declared as a global variable so that the MpCnc2Axis function block can be accessed. The group must be switched on by using the command. After successfully having been switched on and homed, the logic switches to the "Operation" state. Only in this state may additional commands for starting movements be issued. The group is switched off and switched to the "Start" state by resetting the start command.

1 Create new "machine" program
2 Create state machine and add the necessary steps (optional when using high-level programming languages)
3 Create the MpCnc2Axis function block of the "cnc_ctrl" program as a global variable
4 The automatic sequence is enabled by the "cmdOn" command if "Info.ReadyToPowerOn" = TRUE
5 Set "Power" and wait until the "PowerOn" output = TRUE and "IsHomed" = TRUE
6 Transfer additional commands in the "Operation" state
7 Switch off the group if "cmdOn" = FALSE and switch to the "Start" state
8 A movement can be started with "cmdMoveProgram" in the "Operation" state

10.3 Error handling

When programming an application, it is always important to include error evaluation in the program sequence. The following errors can occur

- Error calling a function block
- Axis error
- Group error

Error handling can be programmed in the application using the "Error" and "StatusID" status outputs and the "info" structure. If an error occurs, then the execution of the function block is interrupted. The error should then be corrected considering the error message description in the Automation Studio help system. Using the "ErrorReset" input, MpCnc2Axis can be put into the operative state again and additional commands can be issued.

Figure 40: Error handling in the application program

The logger and the Visual Components alarm system can be used to display errors.

Logger

Detailed data about the cause of the error is entered in the "$mapp" and "GMC" logger file automatically. The logger file can be opened using Automation Studio and the System Diagnostics Manager and be saved on the PC. Alternatively, the logger file can be read using the "AsArLog" library.

By using the HMTL control for Visual Components, it is possible to embed the SDM page, which displays the logger, directly into the machine visualization application.

Application layer - mapp technology \ Diagnostics

Diagnostics and service \ Diagnostic tool

- Logger
- System Diagnostics Manager

Visualization \ Visual Components VC4 \ Control reference \ HTML view

Programming \ Libraries \ Configuration, system information, runtime control

- AsArLog
- ArEventLog

MpAlarm component and Visual Components alarm system

Using the MpAlarm component, the predefined alarms of the MpCnc2Axis component can be forwarded to the Visual Components alarm system using the MpAlarmUI function block. The entries displayed there can be filtered by group, time and priority. Language switching can be used for the displayed texts (see training module TM640 – Alarm System, Trends and Diagnostics).

Application layer - mapp technology \ Components \ Infrastructure \ MpAlarm - Support for alarm management

Visualization \ Visual Components VC4 \ Shared Resources \ Alarm System

Exercise: Programming error handling

In the event of an error, you should switch to the "Error" state. All commands are reset so that the axis group can be switched on again once the acknowledgment is complete and so that all commands for starting movements are disabled.

1 Evaluate the "Error" output in order to recognize the error state
2 Reset the command
3 Axis group acknowledgment must occur through "cmdReset" (request from the operator)
4 Switch on again and perform a new homing procedure, if necessary

10.4 Programming PLCopen function blocks

In order to get additional features in the application program, PLCopen function blocks can be added. They are called cyclically in the program just as the mapp components are.

Error evaluation of the PLCopen function blocks in the program also takes place using "Error" output, error numbers are shown on the "ErrorID" output. If an axis error or an axis group error occurs during the execution of a PLCopen function block, then this is signaled via the two outputs. Additionally, the error information is available via MpCnc2Axis and can be acknowledged with the "ErrorReset" input of this component. For this reason the axis group is brought out of the GroupErrorStop state. In order to reset the error on the PLCopen function block, the "Execute" or "Enable" input must be reset.

Errors that don't lead to an error in the axis group can also occur on the PLCopen function block. This can be the case, for example, when using an invalid axis group reference. In this case, the error is available exclusively on the PLCopen function block but not in the mapp component.

Exercise: Programming additional PLCopen functionality

The functionality must now be expanded by the MC_MoveDirectAbsolute_15 PLCopen function block. It is called cyclically and should be able to be enabled via command. Additionally, the function block must be integrated into the error handling of the application program.

1 Insert MC_MoveDirectAbsolute_15 into the "cnc_ctrl" program
2 Starting the movement is possible in the "Operation" step with the "cmdMoveDirectAbsolute" command
3 Switch into the error step if the "Error" output is set
4 Acknowledge the error and reset the command

11 Summary

Using axis groups can considerably simplify the process of creating a motion application containing multiple axes. Whether it is a CNC or robotics application or a simple group of multiple axes, using axis group functions can drastically reduce the number of function blocks required in the application. The result is an application that is not only easier and faster to create, but also easier to read and maintain.

The components of the MpCnc library are available for controlling axis groups. The function blocks contained comply with the PLCopen standard and are set apart by their efficient design and usability.

Integration in the controller application begins as soon as the user is confident with mapp technology (which is quite simple). Elaborate configuration management, web-based diagnostics, the direct integration of error information in the Automation Studio logger window as well as in the System Diagnostics Manager are available through the component architecture. Incorporating drive alarms in the visualization application is enabled by the MpAlarm component.

Figure 41: mapp technology offers a comprehensive portfolio of functions

Figure 42: PLCopen Motion Control logo

The enhancement of the drive application is possible without any problems via the function blocks of the GmcGrpAPI library because of the function compatibility with the PLCopen function blocks. This way the axis group application can be optimally adjusted to the requirements of the machine.

Additional function blocks are provided to complement the ones defined in the standard. These handle functions specific to the B&R axis group solution. This enables the programmer to access the full range of functions to solve any task.

Seminars and training modules

The Automation Academy provides targeted training courses for our customers as well as our own employees.
At the Automation Academy, you'll develop the skills you need in no time!
Our seminars make it possible for you to improve your knowledge in the field of automation engineering.
Once completed, you will be in a position to implement efficient automation solutions using B&R technology. This will make it possible for you to secure a decisive competitive edge by allowing you and your company to react faster to constantly changing market demands.

Automation Studio seminars and training modules

Programming and configuration	Diagnostics and service
SEM210 – Basics SEM246 – IEC 61131-3 programming language ST* SEM250 – Memory management and data storage SEM410 – Integrated motion control* SEM441 – Motion control (multi-axis systems) ** SEM480 – Hydraulics** SEM1110 – Axis groups and path-controlled movements** SEM510 – Integrated safety technology* SEM540 – Safe motion control*** SEM610 – Integrated visualization*	SEM920 – Diagnostics and service for end users SEM920 – Diagnostics and service with Automation Studio SEM950 – POWERLINK configuration and diagnostics* If you don't happen to find a seminar on our website that suits your needs, keep in mind that we also offer customized seminars that we can set up in coordination with your sales representatives: SEM099 – Individual training day Please visit our website for more information****: **www.br-automation.com/academy**

Overview of training modules	
TM210 – Working with Automation Studio TM213 – Automation Runtime TM223 – Automation Studio Diagnostics TM230 – Structured Software Development TM240 – Ladder Diagram (LD) TM241 – Function Block Diagram (FBD) TM242 – Sequential Function Chart (SFC) TM246 – Structured Text (ST) TM250 – Memory Management and Data Storage TM400 – Introduction to Motion Control TM410 – Working with Integrated Motion Control TM440 – Motion Control: Basic Functions TM441 – Motion Control: Multi-axis Functions TM1110 – Integrated Motion Control (Axis Groups) TM1111 – Integrated Motion Control (Path Controlled Movements) TM450 – Motion Control Concept and Configuration TM460 – Initial Commissioning of Motors TM500 – Introduction to Integrated Safety TM510 – Working with SafeDESIGNER TM540 – Integrated Safe Motion Control	TM600 – Introduction to Visualization TM610 – Working with Integrated Visualization TM630 – Visualization Programming Guide TM640 – Alarm System, Trends and Diagnostics TM670 – Advanced Visual Components TM920 – Diagnostics and service TM923 – Diagnostics and Service with Automation Studio TM950 – POWERLINK Configuration and Diagnostics TM261 – Closed-loop Control with LOOPCONR TM280 – Condition Monitoring for Vibration Measurement TM480 – The Basics of Hydraulics TM481 – Valve-based Hydraulic Drives TM482 – Hydraulic Servo Pump Drives TM490 – Printing machine technology In addition to a printed version, our training modules are also available on our website for download as electronic documents (login required): Visit our website for more information: **www.br-automation.com/academy**

Process control seminars and training modules

Process control standard seminars	Process control training modules
SEM841 – Process Control Training: Basic 1 SEM842 – Process Control Training: Basic 2 SEM890 – Advanced Process Control Solutions	TM800 – APROL System Concept TM811 – APROL Runtime System TM812 – APROL Operator Management TM813 – APROL XML Queries and Audit Trail TM830 – APROL Project Engineering TM890 – The Basics of LINUX Visit our website for more information: **www.br-automation.com/academy**

* SEM210 - Basics is a prerequisite for this seminar.
** SEM410 - Integrated motion control is a prerequisite for this seminar.
*** SEM410 - Integrated motion control and SEM510 - Integrated safety technology are prerequisites for this seminar.
**** Our seminars are listed in the Academy\Seminars area of the website. Seminar titles may vary by country. Not all seminars are available in every country.