TM641

Display alarms, diagrams and data in mapp View

PERFECTION IN AUTOMATION
A MEMBER OF THE ABB GROUP

Prerequisites and requirements	
Training modules	TM611 – Working with mapp View TM671 – Creating efficient mapp View HMI applications
Software	Automation Studio 4.3.3 Automation Runtime 4.34 mapp View 5.0 technology package mapp Services 5.0 technology package
Hardware	ARsim

Table of contents

1 Introduction

With mapp View, automation engineers have all the tools they need to create powerful, intuitive HMI applications. There is no need to deal with the underlying web technology.

This technology is encapsulated in widgets, which are simply dragged and dropped into place where they can be easily configured.

Figure 1: Widgets can be dragged and dropped onto a page.

This training module covers exercises and solutions related to configuring and representing complex process information. We will be using both mapp Services components and mapp View widgets. The mapp View help documentation will be used as the main source of information for configuring the widgets and completing the various exercises.

1.1 Learning objectives

This training module uses selected examples illustrating typical application tasks to help you learn how various functions in mapp View are structured and configured.

- Participants will learn to configure the alarm system and manage alarms in the application.
- Participants will be able to display alarms in the HMI application and integrate them with the alarm system.
- Participants will be familiar with the relationship between the text system and the alarm system.
- Participants will be familiar with the possibilities of the expanded alarm configuration for using condition alarms and reactions.
- Participants will be able to display a live signal in the OnlineChart widget.
- Participants will be able to display historical data in the LineChart widget.

1.2 Symbols and safety notices

Unless otherwise specified, the descriptions of symbols and safety notices listed in "TM210 – Working with Automation Studio" apply.

2 Configuring and displaying alarms

Alarms are used to monitor and display specific machine states so that the users of those machines can respond appropriately and make the necessary decisions. It is important that the alarm information used here is comprehensive and meaningful. This includes using operator-appropriate language and providing the necessary additional information.

When an alarm is configured, the behavior of the alarm is defined along with the required user interaction.

In the mapp Services technology package, the alarm system is represented by the MpAlarmX component. MpAlarmX collects and manages both mapp alarms and user alarms. The alarms are configured using Automation Studio. They are managed in the application and then displayed in a given HMI application or exported as a file.

The alarm system runs on the controller on a standalone basis. The mapp View AlarmList widget facilitates the process of creating a HMI application for alarms.

Figure 2: MpAlarmX

Features of the alarm system:

- Collecting mapp alarms and user alarms
- Recording active and historic alarm states
- Managing alarm texts and properties via the mapp configuration
- Binding the alarm system and text system via the configuration
- Consolidating different alarms based on "reactions"
- Exporting historic alarms to the file system
- Binding the alarm system to the AlarmList widget via mapp Link binding quickly and easily

Services \ mapp Services \ MpAlarmX: Alarm management \

- Concept
- Configuration
- Use cases
- MpAlarmX library
- Guides

2.1 Alarm system terminology

Errors, user-defined alarms and mapp alarms

- Error
  Errors are indicated in the output of a function or function block via the "StatusID." When developing the application, the "StatusID" can be used for troubleshooting. Errors are displayed in the Logger.
- User alarms
  The application developer can define specific alarms via the MpAlarmXCore configuration. The behavior of each alarm is customized here also. User alarms are triggered via the application.
- mapp alarms
  mapp alarms can be found in the configuration for the respective components. In the configuration, the user must specify whether they want the alarms to be transferred to the alarm system.

Basic alarm state

An alarm can have the following states:

- Active and Not Acknowledged
  The alarm is active. mapp alarms are enabled by the respective component, whereas user-defined alarms are set using MpAlarmXSet. If an alarm has not been acknowledged yet, it will also have the state "Not Acknowledged."
- Active and Acknowledged
  The alarm is active. An active alarm can also be already acknowledged. Alarms can be acknowledged in the application via MpAlarmXAcknowledge or in the HMI application.
- Inactive and Not Acknowledged
  The alarm is not active and has not been acknowledged.
- None
  The alarm has been disabled.

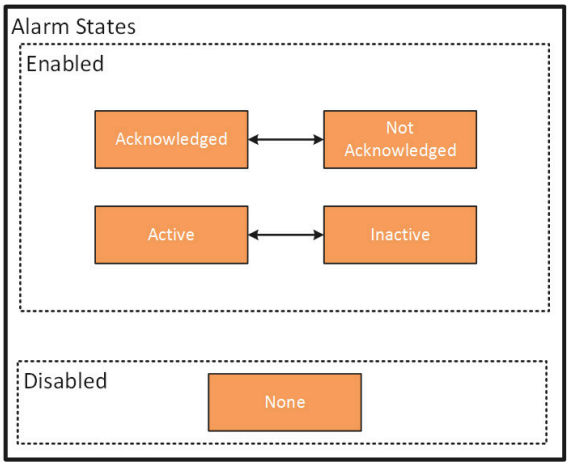

Figure 3: Possible alarm states

Services \ mapp Services \ MpAlarmX: Alarm management \ Concept \ Basic alarm concept

2.2 Alarm behavior

The behavior of an alarm is defined in the alarm configuration. The following section provides a brief overview of configurable alarm behavior.

See also:"Configuring an alarm system" on page 10

Edge alarms

An edge alarm is an alarm that is triggered for a short period only. It is set via the application.

Examples:

- "Unable to load recipe"
- "Unable to send SMS text message to *Shift supervisor*"
- "Data storage device not found"
- "Operation not permitted"

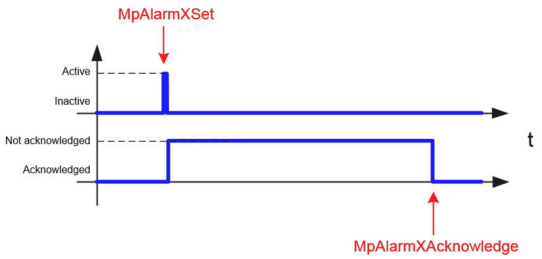

Figure 4: Edge alarm behavior

Persistent alarms

A persistent alarm is an alarm that is typically triggered for a longer period of time. It is set and reset via the application.

Examples:

- "Temperature (123°C) not in normal range (100°C – 120°C)"
- "Tank water level too high"
- "Emergency switch-off activated"
- "X20DI8371 not inserted"

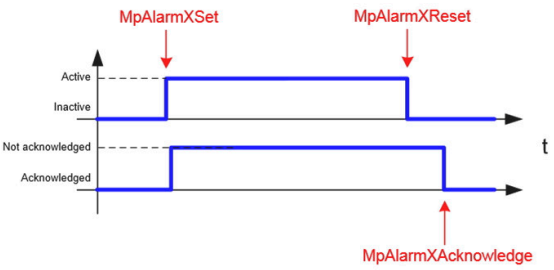

Figure 5: Persistent alarm behavior

Level monitoring alarm

This alarm monitors the level of a process variable (PV). The PV is defined in the configuration. It is possible to define two lower limit values and two upper limit values here. If an upper or lower limit value is exceeded, the alarm is triggered.

Figure 6: Level monitoring alarm

Deviation monitoring alarm

This alarm monitors deviation from a defined level. In this case, a PV is specified with the current value and a second PV defines the setpoint. If the current value deviates from the setpoint by a defined tolerance, an alarm is generated.

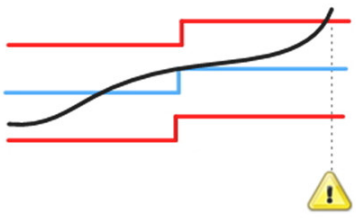

Figure 7: Deviation monitoring alarm

Rate of change monitoring

This alarm monitors the rate of change of a PV. The PV is defined in the configuration. It is possible to define two lower limit values and two upper limit values here. If an upper or lower limit value is exceeded, the alarm is triggered.

Figure 8: Rate-of-change monitoring alarm

Discrete value monitoring alarm

This alarm monitors certain PV values. The PV is defined in the configuration. If a PV changes to a specific value, an alarm is triggered.

Figure 9: Discrete value monitoring alarm

Services \ mapp Services \ MpAlarmX: Alarm management \ Concept

- Alarm properties
- Alarm monitoring

2.3 Configuring an alarm system

The complete MpAlarmX alarm system consists of a mapp Services configuration and a corresponding function block.

In the first step, a new "MpAlarmXCore" mapp configuration is added to Configuration View from the Toolbox in the "mapp Services" package.

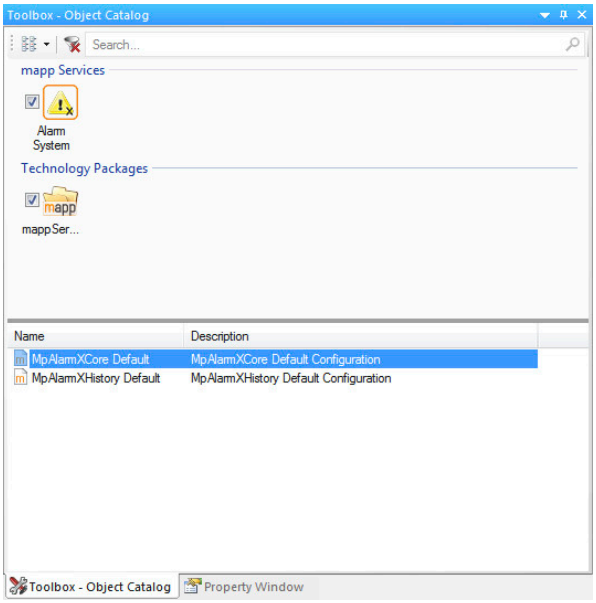

Figure 10: Selecting the "MpAlarmXCore" configuration from the Toolbox

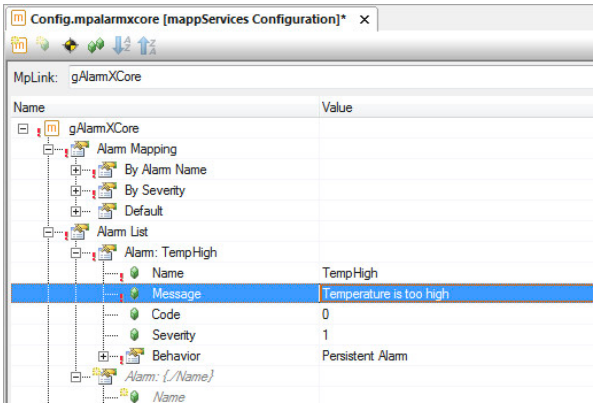

Figure 11: Configuration for "TempHigh" alarm

Configuring a user alarm

The next step is to open the configuration and configure the first alarm. In the figure here, the "TempHigh" alarm is configured as a "Persistent alarm."

At runtime, user alarms are set and reset using the "MpAlarmXSet" and "MpAlarmXReset" functions in the MpAlarmX library.

Calling the alarm system

The "MpAlarmXCore" function block is used to enable the alarm system in the controller. This function block manages all mapp alarms and user alarms at runtime. The previously created configuration is assigned to the function block via mapp Link.

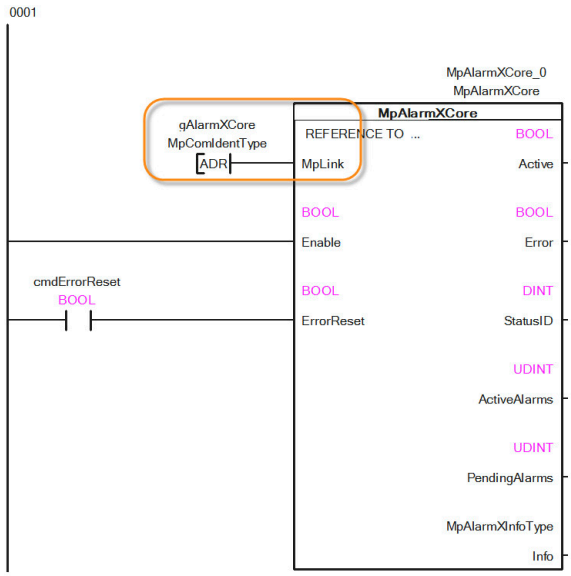

Figure 12: Calling MpAlarmXCore

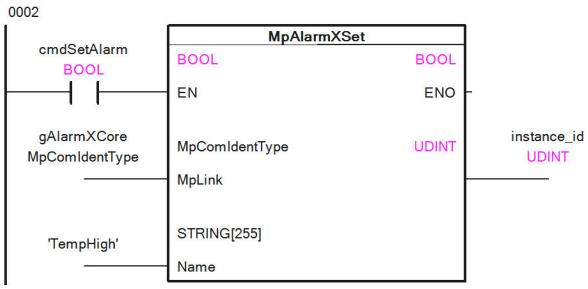

Figure 13: Calling the MpAlarmXSet function

Setting and resetting user alarms

User alarms are set and reset using the "MpAlarmXSet" and "MpAlarmXReset" functions respectively. User alarms are acknowledged using the "MpAlarmXAcknowledge" function. When the functions are called, the mapp Link is transferred also.

Handling of mapp alarms

The mapp concept is responsible for handling communication between mapp components. In some cases, other mapp components may have their own standalone alarms and need to transfer these to the alarm system. If applicable, this is enabled in the corresponding configuration. At runtime, mapp alarms are transferred to the alarm system automatically.

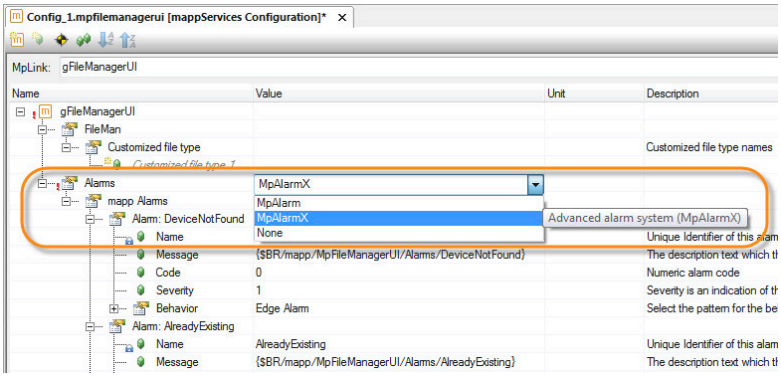

Figure 14: Alarms are sent to the MpAlarmX alarm system automatically.

Services \ mapp Services \ MpAlarmX: Alarm management \

- Configuration \ MpAlarmXCore configuration
  See instructions on YouTube
- MpAlarmX library \ Function blocks

Exercise: Configure alarm system and call in ST

MpAlarmX is used for alarm management in the application. First, the alarm system is configured and called in the Structured Text. Then, alarms are triggered and acknowledged in the program code.

1) Add the configuration for MpAlarmX
2) Add two alarms

   In each case, a name, alarm text and behavior have to be determined.

Name	Text	Severity	Behavior
TempHigh	"Temperature is too high"	20	Persistent
SendFailure	"Message could not send"	10	Edge

Table 1: Configuration of the alarms in the alarm configuration

3) Calling MpAlarmXCore
4) Calling MpAlarmXSet, MpAlarmXReset and MpAlarmAcknowledge for "TempHigh" and "SendFailure"

   Observation of the outputs "ActiveAlarms" and "PendingAlarms" of MpAlarmX

2.4 Displaying alarms in the HMI application

The AlarmList widget is used to display alarms in the HMI application. It is a container widget to which AlarmListItem widgets are added. AlarmItem widgets represent the individual columns of the alarm list.

In the properties for the AlarmList widget, the "itemsPerPage" parameter is used to define the number of entries displayed on the list. mapp binding is used to bind the AlarmList widget to the alarm system instance.

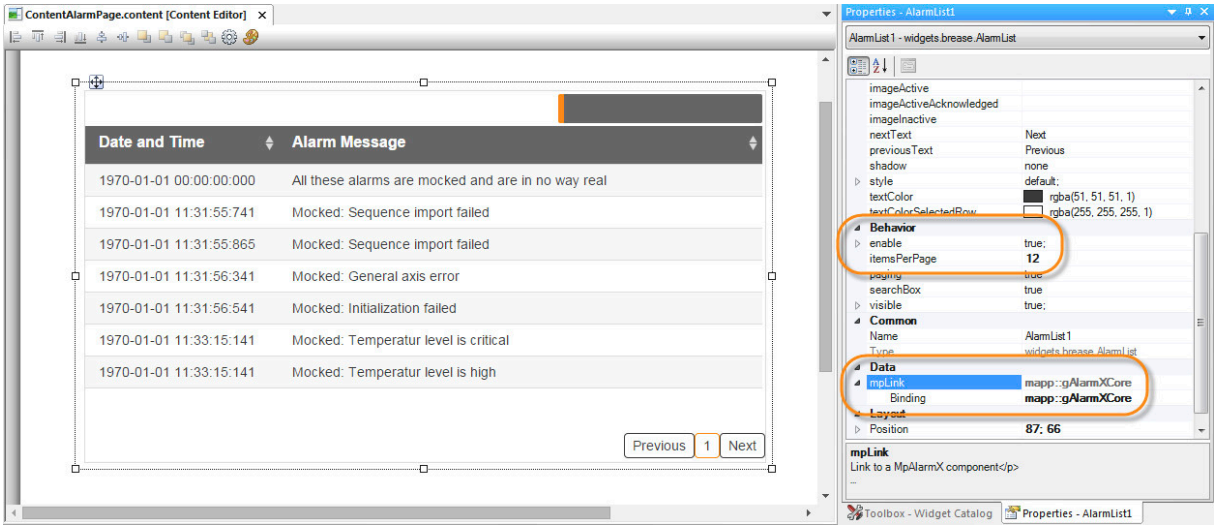

Figure 15: Properties of the AlarmList widget – mapp binding and number of list elements

To use this widget, the "MpServer" library must be transferred to the controller.

The alarm information that is to be displayed is defined in the AlarmItem widget. The column width is also defined in the AlarmItem widget.

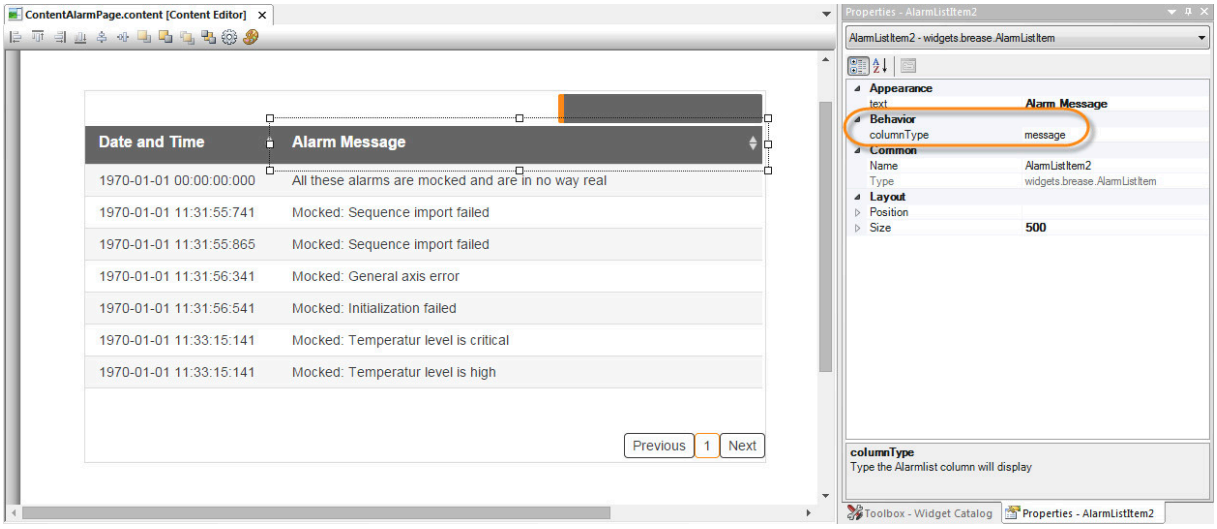

Figure 16: AlarmItem widget properties used for displaying message text

Visualization \ mapp View \ Widgets \ Data \ AlarmList

Exercise: Display current alarm list in mapp View

The alarms are now displayed in the mapp View HMI application.

1) Add MpServer library to Logical View
2) Add AlarmList widget to "AlarmPage" page
3) Bind AlarmList widget to alarm configuration via mapp Link
4) Add three AlarmListItem widgets

   What should be displayed:

   - Alarm time
   - Alarm text
   - Alarm state

5) Customize AlarmList widget for display

3 Relationship between alarm system and text system

Previously, alarm text was specified in the alarm configuration directly. However, the disadvantage here was that text entered in this way was not taken into account during localization.

To ensure that all alarm system text can be localized, the integrated text system is now used. In addition, alarm text snippets can be used to integrate application data into alarm text directly.

Localizable texts are referenced globally in the project using the fully specified name, which consists of Text ID+Namespace. Namespaces define a logical hierarchy (similar to file paths) that is used to manage texts within a project independently of where the text modules are located.

Services \ mapp Services \ MpAlarmX: Alarm management \ Concept \ Alarm texts

Programming \ Text system \ Managing localizable texts

3.1 Localizing alarm text via the text system

The Automation Studio text system is used for the centralized management of localizable texts for the target system. In addition to the direct use of alarm texts in the application and the display of these texts in the HMI application and in localized Logger entries, it is also possible for alarm texts to be localized via the text system.

Enabling the text system

The first step is to add a language configuration file to Logical View. This is where the languages that are available for localization are managed. The configuration for the text system is then added to Configuration View. All project languages and text data that are required at runtime must be added to the text system configuration.

Programming \ Text system

- Managing project languages
- Text system configuration

Adding text files

For localizable alarm texts, a localizable text file is added to Logical View from the Toolbox.

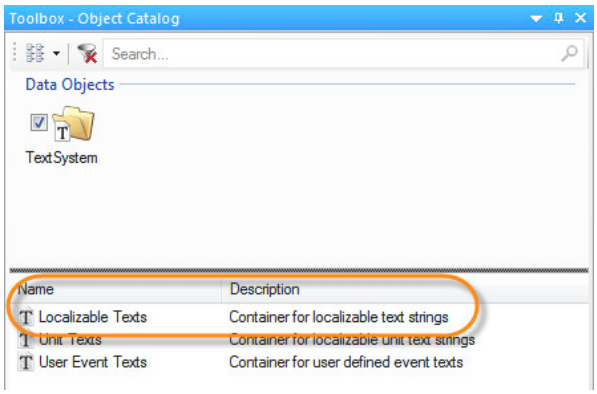

Figure 17: Adding a localizable text file

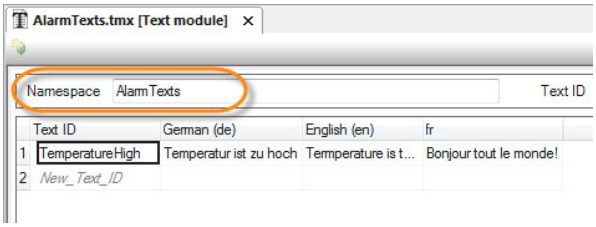

Figure 18: Entering the namespace in the text file

Using texts in the alarm configuration

In this figure, the alarm text is not directly specified in the Message parameter, instead it is linked to the text system itself. The text source is a text file containing the localizable alarm texts that have been added to Logical View.

Syntax: {$Namespace/TextID}

Namespaces and texts

The next step is to configure the namespace, text IDs and texts in the text file. If the texts are located in the "IAT" namespace, they will be localized later via the mapp View client. If they are located in a different namespace, they will be translated by the alarm system.

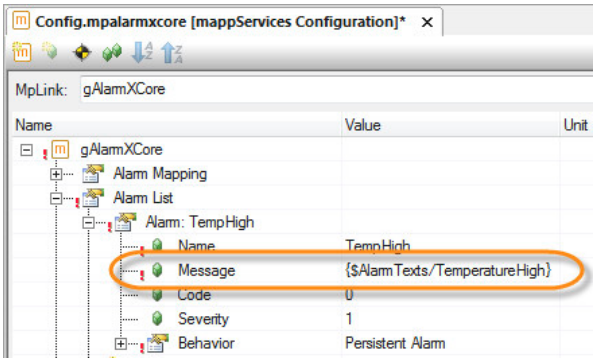

Figure 19: Referencing a text from the text system

Services \ mapp Services \ MpAlarmX: Alarm management \ Concept \ Alarm texts

Exercise: Localize alarm texts via text system

The alarm texts previously entered statically into the alarm configuration are not yet localizable. The objective of these exercises is to use the text system to create localizable alarm texts.

1) Add new TMX file
2) Define namespace
3) Enter alarm texts in the TMX file in different languages

   A unique Text ID needs to be configured for each text
4) Add text file to text system configuration (in Configuration View)
5) Replace alarm configuration with texts from the text system

   The texts must be localized using MpAlarmX.

3.2 Integrating application data

You can use alarm text snippets to integrate application data into your alarm texts. These alarm text snippets can be defined in the alarm configuration.

Alarm text snippets can be used for different alarm texts. This means that a single alarm text snippet can be used for one or more alarms. For each alarm text snippet that is used, a key is defined in order to identify the alarm text snippet in the alarm text.

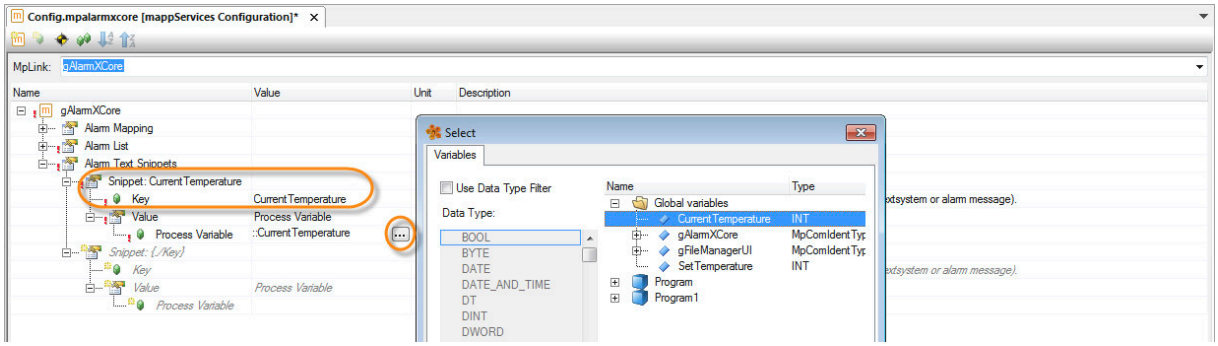

Figure 20: Creating an alarm text snippet

In the next step, the alarm text snippet is integrated into the alarm text.

Alarm text snippets can be integrated into the alarm text regardless of whether the text is retrieved from the alarm configuration or imported from the text system.

The syntax is as follows: {&Key}

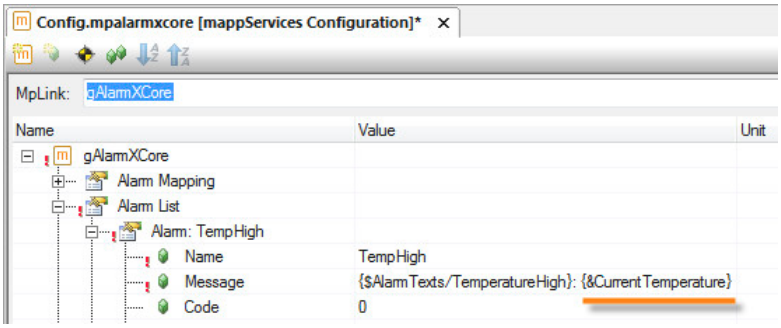

Figure 21: Integrating alarm text snippets into alarm text

For more details about the options available here, see the Format strings section.

Result

Figure 22: Value of alarm text snippet displayed in the alarm text

Services \ mapp Services \ MpAlarmX: Alarm management \ Concept \ Alarm texts

Programming \ Text system \ Managing localizable texts \ Format strings

Relationship between alarm system and text system

Exercise: Add text snippets in alarm texts

When displaying alarms, it's useful to display the current process value at the time the alarm is triggered. This is enabled by using text snippets.

1) Set up text snippet with the current temperature "CurrentTemperature"
2) Add text snippet to the alarm "TempHigh"

4 Expanded alarm configuration

Previously, user alarms were set directly in the application using the "MpAlarmXSet" function. Using alarm text snippets, data from the controller was displayed in the alarm text as additional information.

The alarm configuration contains further configuration options to help tailor the alarm system to users' specific needs.

4.1 Severity level and monitoring

Monitoring

In addition to the display of persistent alarms and edge alarms, monitoring functions are also provided for process variables. This enables the monitoring of automatic limit values for process variables and, if these limits are exceeded, the automatic triggering of alarms.

See: "Alarm behavior" on page 8

Services \ mapp Services \ MpAlarmX: Alarm management \ Concept \ Alarm monitoring

Severity

The severity indicates the severity "level" of an alarm. This is a freely selectable integer that can be defined as needed. For example, non-critical alarms can be assigned a severity of 10, while critical alarms can be given a severity of 100.

The severity level can be used later for filtering alarms or for monitoring alarms with the same severity level.

Services \ mapp Services \ MpAlarmX: Alarm management \ Concept \ Alarm properties

Exercise: Execute expanded alarm configuration

Now additional alarms are added to the configuration that are used for different monitoring purposes.

1) Add two new conditions

   In addition to the alarm, the current process value must be displayed in the alarm text using a text snippet when the alarm goes off.

Name	Text	Severity	Behavior	Monitored variable	Limit
TankLevel	"Water level is too high"	30	LevelMonitoring	"aiTankLevel"	120 cm
SetTemp100	"Input setpoint for boiling temperature performed"	60	DiscreteValueMonitoring	"SetTemperature"	100°C

Table 2: Monitoring alarms in the alarm configuration

2) Monitoring the process values in the Watch window and the alarm indication

Expected result

The following text is shown in the HMI application when the process variable "aiTankLevel" exceeds the value 120 cm:

"Water level is too high: current water level 121 cm".

The following alarm text is shown when entering the value 100°C for the variable "SetTemperature":

"Input setpoint for boiling temperature performed"

4.2 Mapping alarms

In addition to alarm properties, users can also define how an alarm is to behave in the overall system. They can decide what should happen for a particular alarm, or what actions should be carried out for a certain trigger. This is done using alarm mapping.

In each case, a trigger is connected to one or more actions (reactions). A trigger can be a specific alarm or a severity level.

The following diagram illustrates the relationship between the different alarms. These relationships can be assigned in the alarm configuration directly, or assigned to a specific reaction via the severity level. For example, the "NoProduct", "InvalidProduct" and "SawBattered" alarms will always trigger a "Stop" reaction. Using reactions significantly reduces the number of states to be analyzed in the application.

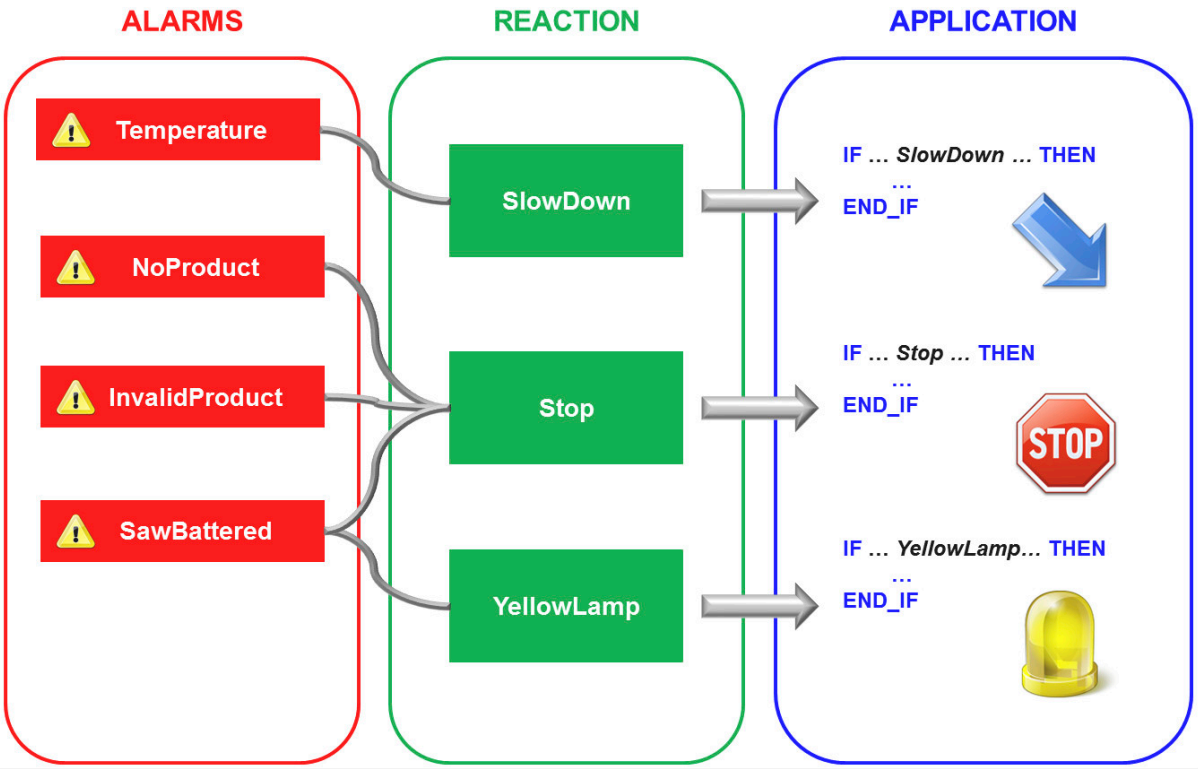

Figure 23: Relationship between alarms, severity and the application program

Alarm

Whenever a specific alarm is triggered, one or more actions are carried out. In the figure below, we can see that the "StopMachine" reaction is initiated when the "TempHigh" alarm is triggered.

A number of different alarms can initiate the same reaction.

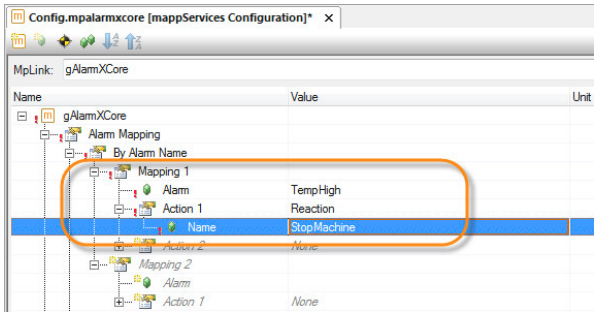

Figure 24: "StopMachine" reaction triggered by "TempHigh" alarm

Severity

The alarm configuration can be used to set reactions for specific severities. Alarms that have the same severity or a severity within a given value range can trigger a specific reaction, for example.

As shown in the figure, a value range can also be entered for the severity.

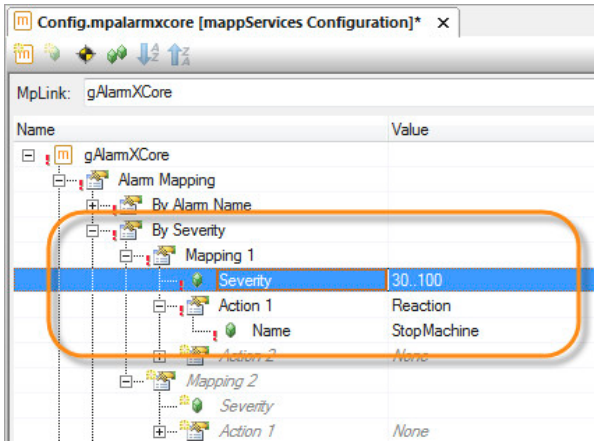

Figure 25: Alarm mapping using a range of severity values

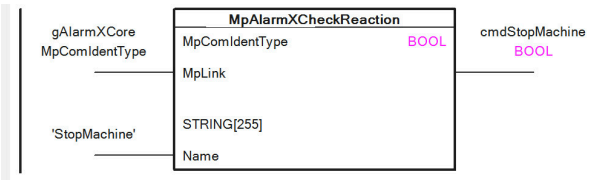

Figure 26: Querying whether the “StopMachine” reaction is active

Evaluating reactions

The MpAlarmXCheckReaction function can be used to check whether a certain reaction is active. The application software can then respond to this reaction appropriately.

Services \ mapp Services \ MpAlarmX: Alarm management \ Concept \ Mapping alarms

Exercise: Consolidate alarm states using reactions

Different alarms, which should trigger identical events on the machine, can be merged using the reaction.

Three reactions should be configured. The first reaction reduces the speed of the machine. The second reaction stops the machine completely. The third reaction activates a red light, which is triggered based on the severity level when the machine is stopped.

1) Configuration of reactions for the configured alarms

Cause of the reaction	Reaction
TempHigh	SlowDown
TankLevel	StopMachine
Severity > 50	StopMachine
Severity > 50	RedLamp

Table 3: Assigning alarms to reactions

2) Evaluation of the reactions in the application program with MpAlarmXCheckReaction

Expected result

Depending on the configuration, a reaction can be triggered by one or more specific alarms. A reaction being triggered by the occurrence of alarms with a certain severity level is also enabled. Several reactions can be triggered simultaneously by evaluating alarms and severities.

5 Displaying diagrams

Diagrams provide a visual representation of one or more values. For example, they can show the curve of a signal over time in a two-dimensional coordinate system – similar to an oscilloscope.

Charts display graphs in a widget where the horizontal x-axis (abscissa) represents time, and the vertical y-axis (ordinate) displays the process values.

Charts not only allow you to observe a specific process value, more importantly they allow you to track how this value changes over time.

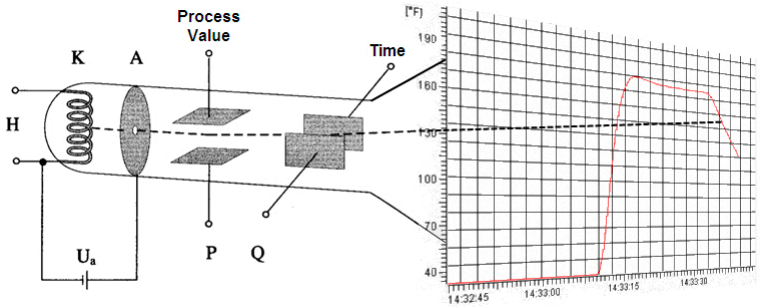

Figure 27: Virtual oscilloscope – Chart widgets

OnlineChart and LineChart features:

- Displaying live process values over time
- Displaying the recorded values from the application program
- Displaying multiple graphs differentiated by color
- Measurement cursor for calculating values at intersections
- Flexible configuration of value and time axes
- Zooming in and scrolling through data in the chart
- Showing and hiding curves

Visualization \ mapp View \ Widgets \ Chart

This training module does not cover additional methods of representation such as Timeline, RadialGauge, LinearGauge, BarChart and ProfileGenerator.

Exercise: Configure additional page

To complete the remaining exercises, you must create a new blank page. The page must then be added to the automatic navigation.

1) Add new "ChartPage" page
2) Add new "ContentChartPage" page content
3) Open page file
4) Assign layout
5) Assign AreaContents folder
6) Reference page in .vis file
7) Update .nav file to include page in automatic navigation

5.1 Configuring the OnlineChart widget

The OnlineChart widget is used to display a live value over time. It is a container widget and therefore requires the addition of further child widgets to ensure the full display of all data.

1) The OnlineChartTimeAxis widget is added to the OnlineChart widget.
2) The OnlineChartYAxis widget is added to the OnlineChart widget.
3) An OnlineChartGraph widget is added to the y-axis.
4) In the configuration for the OnlineChartGraph widget, the x-axis is assigned via the “xAxisRe-fld” property.
5) Bindings to a variable are set in the configuration for the OnlineChartGraph widget.

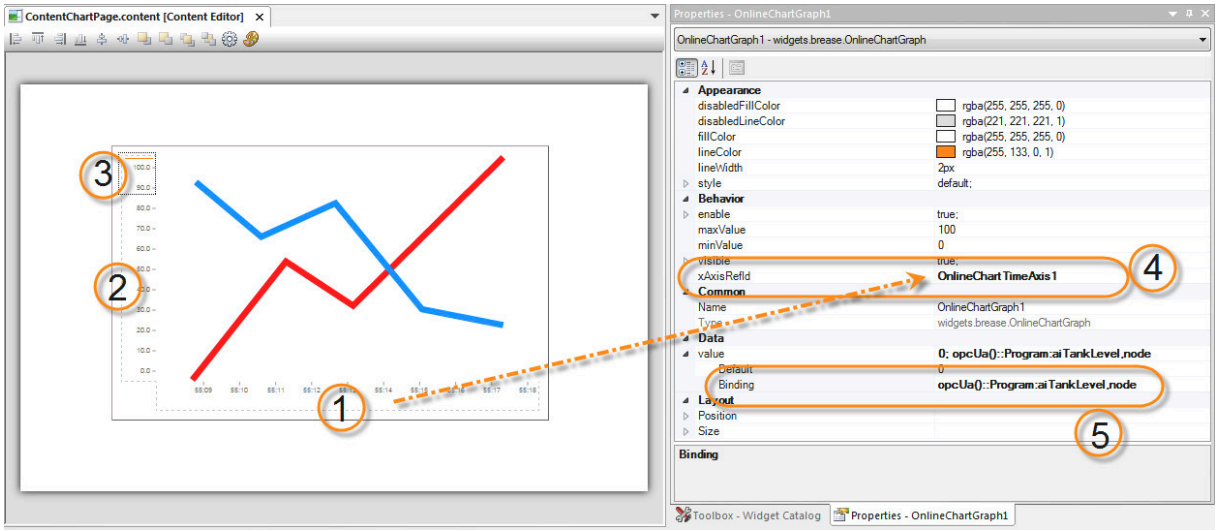

Figure 28: Structure of the OnlineChart widget container

Visualization \ mapp View \ Widgets \ Chart \ OnlineChart \ Concept

Exercise: Display live signal in the OnlineChart widget

In this exercise, the process variable "Program:aiTankLevel" is shown in the mapp View HMI application using the OnlineChart widget. The OnlineChart widget is the Container widget, which allows the creation of an OnlineChart with one or more characteristic curves. Additional widgets are required for the configuration.

1) Add the OnlineChart widget
2) Add the OnlineTrendTimeAxis widget
3) Add the OnlineTrendYAxis widget
4) Add the OnlineChartGraph widget

   A characteristic curve can be displayed using the OnlineChartGraph widget. The OnlineChartGraph widget is retrieved from the Widget Catalog and placed on the y-axis.

   The name of the x-axis referenced by the data is entered on property xAxisRefId of the OnlineChartGraph widget.

5) Bind value property to the process variable "aiTankLevel"

Exercise: Expand OnlineChart with a second live value

The OnlineChart widget allows the management of several characteristic curves as well as of label axes. The objective of this exercise is to add a second characteristic curve with an independent y-axis.

1) Add an additional OnlineTrendYAxis widget
2) Add an additional OnlineChartGraph widget
3) Bind value property to the process variable "CurrentTemperature"

5.2 Configuring the LineChart widget

The LineChart widget is used to display numerical data in a diagram. It is a container widget and therefore requires the addition of further child widgets to ensure the full display of all data.

1) The LineChartIndexAxis widget is added to the LineChart widget.
2) The LineChartYAxis widget is added to the LineChart widget.
3) A LineChartXAxisIndexCursor widget is added to the x-axis.
4) The LineChartGraph widget is added to the y-axis.
5) In the configuration for the LineChartGraph widget, the x-axis is assigned via the “xAxisRefId” property.
6) Bindings to a variable are set in the configuration for the LineChartGraph widget.

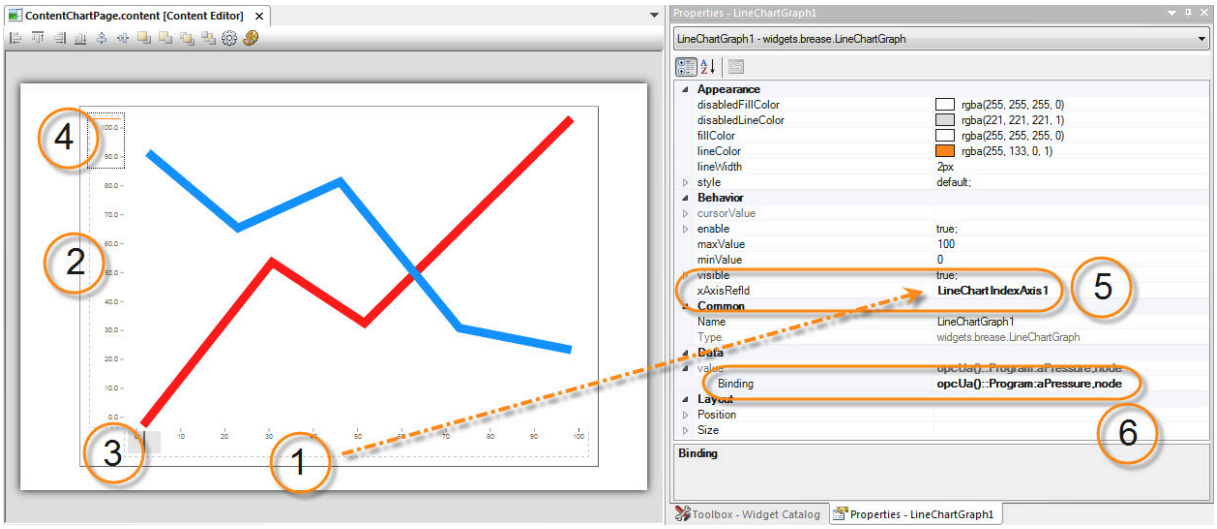

Figure 29: Structure of the LineChart widget container

? Visualization \ mapp View \ Widgets \ Chart \ LineChart \ Concept

Exercise: Display data buffer in LineChart widget

In this exercise, a data buffer from the controller is shown in the mapp View HMI application using the LineChart widget. The LineChart widget is the Container widget, which allows the creation of a LineChart with one or more characteristic curves. Additional widgets are required for the configuration. In the controller, a field variable is generated as the data source. The values are automatically filled using the declaration editor.

1) Generate field variable with data
   - Declaration of field variable "aPressure" INT[0..99]
   - Fill in the field variable in the declaration editor
   - Call the field variable in the control program
   - Enable the field variable in the OPC UA default view
2) Add the LineChart widget
3) Add the LineChartIndexAxis widget

   When using the LineChartIndexAxis widget, the values are numbered in ascending order on the x-axis.
4) Add the LineChartYAxis widget
5) Add LineChartGraph widget and bind it to OPC UA ArrayNode

   A characteristic curve can be displayed using the LineChartGraph widget. The LineChartGraph widget is taken from the Widget Catalog and placed on the y-axis.

   The name of the x-axis referenced by the data is entered on property xAxisRefId of the LineChartGraph widget.

The engineering unit of the connected OPC UA node must apply to the "unit" property of the y-axis. For example, if BAR or hectopascal units are used for printing, the unit string will be as follows:

```
"{'metric':'BAR','imperial':'A97','imperial-us':'A97'}"
```

Visualization \ mapp View \ Widgets \ LineChart \ Concept

6 Displaying additional data

The controller contains a large amount of data in various different formats. The following exercises show how different data can be displayed in mapp View HMI applications.

Visualization \ mapp View \ Widgets \ UseCases

Exercise: Display temperature values in Table widget together with unit

The objective of this exercise is to display a list of temperature values in a table together with the unit. The first step is to create a new "aTemperatures" variable in the controller based on the INT data type and then to initialize it with values. Using the Table widget, the list of values should then be displayed together with the units.

1) Create and initialize "aTemperatures" variable based on INT data type

   The field variable should accommodate 10 elements.

2) Enable new variable in OPC UA default view and include field elements
3) Position Table widget
4) Add TableItem widget to Table widget
5) Create binding for value list
6) Configure unit string for °C and °F conversion

Programming \ Unit system \ Available standard units

7 Summary

With mapp View, automation engineers have all the tools they need to create powerful, intuitive HMI applications. There is no need to deal with the underlying web technology.

This technology is encapsulated in widgets, which are simply dragged and dropped into place where they can be easily configured.

Figure 30: Widgets can be dragged and dropped onto a page.

Using the mapp Services alarm system and the associated mapp View AlarmList widgets, a freely configurable alarm system can easily be integrated into the HMI application. Container widgets enable flexible configuration for the visualization of complex data and can be customized to each user's specific requirements.

The comprehensive mapp View and mapp Services help documentation describes the full range of options as well as common use cases.

8 Other tasks

Exercise: Display alarm history

The goal of this exercise is to display the graph of the alarms and export it as a file.

1) Add configuration for MpAlarmXHistory
2) Configure file device for AlarmExport
3) Call MpAlarmXHistory - Specify FileDevice
4) Call MpAlarmXHistoryUI
5) Enable MpAlarmXHistoryUIConnect in the OPC UA server
6) Bind MpAlrmXHistoryUIConnect to TableView widget and Button widget to the interaction
7) Enable AlarmExport with MpAlarmXHistory.Export

Exercise: Integration of the FileBrowser widget [variant 1]

The content of the FileBrowser should be displayed in the mapp View HMI application.

1) Call MpFileManagerUI component
2) Create file device and initialize device on MpFileManagerUIConnect structure
3) Enable MpFileManagerUIConnect structure in the OPC UA server
4) Add the use of different widgets for displaying the file list.

   What should be displayed:

   - File name
   - File size
   - File type

5) Add in Button widgets for navigating and interacting with the file manager

   The following actions should be possible:

   - Scrolling up and down

Offered by the Automation Academy

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.

Seminars

Quality and relevance are essential components of our seminars. The pace of a specific seminar is based strictly on the experience that course participants bring with them and tailored to the requirements they face. A combination of group work and self-study provides the high level of flexibility needed to maximize the learning experience.
Each seminar is taught by one of our highly skilled and experienced trainers.

Training modules

Our training modules provide the basis for learning both at seminars as well as for self-study. These compact modules rely on a consistent didactic concept. Their bottom-up structure allows complex, interrelated topics to be learned efficiently and effectively. They serve as the best possible companion to our extensive help system. The training modules are available as downloads and can be ordered as printed versions.

Topic categories:

- ⇨ Control technology
- ⇨ Motion control
- ⇨ Safety technology
- ⇨ HMI
- ⇨ Process control
- ⇨ Diagnostics and service
- ⇨ POWERLINK and openSAFETY

ETA system

The ETA system provides realistic constructions for training, education and laboratory use. Two different basic mechanical constructions can be selected. The ETA light system offers a high degree of mobility, saves space and is well-suited for lab work. The ETA standard system has a sturdy mechanical structure and includes pre-wired sensors and actuators.

Find out more!

Do you require additional training? Are you interested in finding out what the B&R Automation Academy has on offer? You've come to the right place.
Detailed information can be found under the following links:
www.br-automation.com/academy