

# SIEMENS

SIMATIC

S7-1500

Getting Started

Welcome

Automation task

1

Hardware section

2

Software section

3

Security

4

S7-1500 Motion Control

5




SIMATIC Safety V17

6

## Legal information

### Warning notice system

This manual contains notices you have to observe in order to ensure your personal safety, as well as to prevent damage to property. The notices referring to your personal safety are highlighted in the manual by a safety alert symbol, notices referring only to property damage have no safety alert symbol. These notices shown below are graded according to the degree of danger.

 <b>DANGER</b>
indicates that death or severe personal injury <b>will</b> result if proper precautions are not taken.
 <b>WARNING</b>
indicates that death or severe personal injury <b>may</b> result if proper precautions are not taken.
 <b>CAUTION</b>
indicates that minor personal injury can result if proper precautions are not taken.
<b>NOTICE</b>
indicates that property damage can result if proper precautions are not taken.


If more than one degree of danger is present, the warning notice representing the highest degree of danger will be used. A notice warning of injury to persons with a safety alert symbol may also include a warning relating to property damage.

### Qualified Personnel

The product/system described in this documentation may be operated only by **personnel qualified** for the specific task in accordance with the relevant documentation, in particular its warning notices and safety instructions. Qualified personnel are those who, based on their training and experience, are capable of identifying risks and avoiding potential hazards when working with these products/systems.

### Proper use of Siemens products

Note the following:

 <b>WARNING</b>
Siemens products may only be used for the applications described in the catalog and in the relevant technical documentation. If products and components from other manufacturers are used, these must be recommended or approved by Siemens. Proper transport, storage, installation, assembly, commissioning, operation and maintenance are required to ensure that the products operate safely and without any problems. The permissible ambient conditions must be complied with. The information in the relevant documentation must be observed.

### Trademarks

All names identified by ® are registered trademarks of Siemens AG. The remaining trademarks in this publication may be trademarks whose use by third parties for their own purposes could violate the rights of the owner.

### Disclaimer of Liability

We have reviewed the contents of this publication to ensure consistency with the hardware and software described. Since variance cannot be precluded entirely, we cannot guarantee full consistency. However, the information in this publication is reviewed regularly and any necessary corrections are included in subsequent editions.

# Welcome

## Welcome

Welcome to the Getting Started "TIA Portal V17".

In this Getting Started, we show you an example of how to use the CPU SIMATIC S7-1500 with the TIA Portal to create an automation solution for a "color mixing plant". Video clips will illustrate the approach for creating a solution for the automation task.

In the first part, you assemble the hardware and prepare your configuration PC.

In the second part, you configure the CPU and HMI visualization using the example of a color mixing plant.

In addition, you can find options and extensions for your automation solutions.

## Security information

Siemens provides products and solutions with industrial security functions that support the secure operation of plants, systems, machines and networks.

In order to protect plants, systems, machines and networks against cyber threats, it is necessary to implement – and continuously maintain – a holistic, state-of-the-art industrial security concept. Siemens' products and solutions constitute one element of such a concept.

Customers are responsible for preventing unauthorized access to their plants, systems, machines and networks. Such systems, machines and components should only be connected to an enterprise network or the internet if and to the extent such a connection is necessary and only when appropriate security measures (e.g. firewalls and/or network segmentation) are in place.

For additional information on industrial security measures that may be implemented, please visit (<https://www.siemens.com/industrialsecurity>).

Siemens' products and solutions undergo continuous development to make them more secure. Siemens strongly recommends that product updates are applied as soon as they are available and that the latest product versions are used. Use of product versions that are no longer supported, and failure to apply the latest updates may increase customers' exposure to cyber threats.

To stay informed about product updates, subscribe to the Siemens Industrial Security RSS Feed visit (<https://www.siemens.com/industrialsecurity>).

# Table of contents

	<b>Welcome .....</b>	<b>3</b>
<b>1</b>	<b>Automation task .....</b>	<b>11</b>
1.1	Introduction .....	11
1.2	Application .....	12
1.3	Hardware configuration .....	13
1.4	Sample project.....	14
1.4.1	Selecting the recipe .....	15
1.4.2	Displaying the CMYK and RGB values.....	16
1.4.3	Filling the recipe .....	17
1.4.4	Starting the mixing process.....	18
1.4.5	Filling the color mixture .....	19
1.4.6	Resetting the fill level.....	20
<b>2</b>	<b>Hardware section.....</b>	<b>21</b>
2.1	Introduction .....	21
2.1.1	Requirements .....	21
2.1.2	Additional information.....	23
2.2	Installing the assembly.....	23
2.2.1	Overview .....	23
2.2.2	Installing the assembly.....	24
2.3	Wiring .....	28
2.3.1	Overview .....	28
2.3.2	Wiring rules .....	29
2.3.3	Wiring the mains connection plug .....	30
2.3.4	Wiring the load current supply (PM) to the CPU .....	32
2.3.5	Potential bridge circuits.....	33
2.3.6	Wiring the digital input module.....	34
2.3.7	Wiring the digital output module.....	36
2.3.8	Wiring front connectors .....	38
2.4	Power on .....	40
2.4.1	Overview .....	40
2.4.2	Power on .....	41
2.4.3	Assign IP address via the display.....	43



<b>3</b>	<b>Software section</b>	<b>45</b>
3.1	Creating the project and hardware	45
3.1.1	Introduction to the TIA Portal	45
3.1.2	Creating a project	46
3.1.3	Creating an S7-1500 CPU	49
3.1.4	Running the hardware detection	52
3.1.5	Creating ET 200 interface modules	54
3.1.6	Networking ET 200 interface modules	56
3.1.7	Creating input and output modules and a server module for ET 200SP	57
3.1.8	Creating input and output modules for ET 200MP	59
3.1.9	Assigning names for ET 200	60
3.2	Creating the program	61
3.2.1	Loading the block library	61
3.2.2	Deleting program block Main [OB1]	64
3.2.3	Copying program blocks	66
3.2.4	Cyclic interrupt OB	67
3.2.4.1	Cyclic interrupt OB – Cycle time and phase	67
3.2.4.2	Changing the cycle time	68
3.2.5	Copying tag tables	69
3.2.6	Compiling a project	70
3.2.7	Load project into the CPU	72
3.2.8	Optimized block access	74
3.2.8.1	Introduction	74
3.2.8.2	Expanding and reloading the optimized "Filling" data block	75
3.2.9	Versioning a block	79
3.2.10	Setting retentivity	82
3.2.11	Activating the EN/ENO mechanism	85
3.2.12	Using the comment function	87
3.2.13	Local error handling	88
3.2.13.1	Handle errors within block	88
3.2.13.2	Loading blocks for local error handling	90
3.2.13.3	Generating errors without local error handling	92
3.2.13.4	Generating errors with local error handling	93
3.3	Configure visualization	95
3.3.1	Present sample project	95
3.3.2	HMI configuration	95
3.3.2.1	Overview	95
3.3.2.2	SIMATIC HMI Comfort Panels	96
3.3.2.3	HMI screens	97
3.3.2.4	Additional control elements	98
3.3.2.5	Recipes	99
3.3.2.6	Archives	100
3.3.2.7	User-defined functions	101
3.3.2.8	User Management	103
3.3.2.9	Multilingualism	105
3.3.2.10	Reports	107

3.3.3	Insert HMI device from libraries .....	109
3.3.3.1	Insert HMI device from library .....	109
3.3.4	Configuring HMI connection .....	110
3.3.4.1	Communication between devices.....	110
3.3.4.2	Configuring HMI connection .....	111
3.3.4.3	Connecting HMI tags .....	113
3.3.5	Configuring system diagnostics.....	115
3.3.5.1	System diagnostics basics .....	115
3.3.5.2	System diagnostics views.....	116
3.3.5.3	Configuring the system diagnostic view .....	119
3.3.6	Simulating an HMI device.....	122
3.3.6.1	Simulation basics .....	122
3.3.6.2	Operating the panel in simulation .....	123
3.4	Loading the project into the programming device .....	127
3.4.1	Load CPU to project .....	127
3.5	Team engineering via Inter Project Engineering .....	129
3.5.1	Basics of "Inter Project Engineering" .....	129
3.5.2	Creating an IPE file.....	130
3.5.3	Importing an IPE file.....	131
<b>4</b>	<b>Security.....</b>	<b>135</b>
4.1	Overview of the protective functions of the CPU.....	135
4.2	Using the display to configure additional access protection .....	136
4.3	Know-how protection .....	137
4.4	Copy protection .....	140
4.5	Protection by locking the CPU .....	141
4.6	Configuring access protection for the CPU.....	142
4.7	Configuring protection of the HMI connection.....	145
<b>5</b>	<b>S7-1500 Motion Control.....</b>	<b>147</b>
5.1	Introduction .....	147
5.2	Preparations .....	148
5.2.1	Requirements .....	148
5.2.2	Basic procedure .....	150
5.2.3	Opening a project.....	150
5.2.4	Swapping the start screen.....	151
5.2.5	Pasting blocks.....	154
5.3	Configuring drives .....	155
5.3.1	Configuring SINAMICS S120.....	155
5.3.2	Configuring SINAMICS G120 .....	163
5.3.3	Using prepared drives .....	166
5.4	Creating technology objects.....	168
5.4.1	Configuring the positioning axis.....	168
5.4.2	Configuring a speed axis.....	172
5.4.3	Using prepared technology objects .....	175

5.5	Programming the PLC .....	176
5.5.1	Structure of the user program .....	176
5.5.2	Adding Motion Control instructions.....	176
5.5.2.1	Adding Motion Control instructions for a positioning axis .....	176
5.5.2.2	Adding Motion Control instructions for a speed axis .....	181
5.5.2.3	Using prepared function blocks.....	185
5.5.3	Integrating Motion Control instructions in the cyclic user program .....	186
5.5.3.1	Integrating Motion Control instructions for a positioning axis .....	186
5.5.3.2	Integrating Motion Control instructions for a speed axis .....	191
5.5.3.3	Using the prepared OB35.....	195
5.5.4	Compiling and loading the project .....	196
5.6	Testing a function.....	196
5.6.1	"Positioning axis" axis control panel.....	196
5.6.2	"Speed axis" axis control panel .....	205
5.6.3	Testing the function in the start screen .....	208
5.7	Additional information.....	210
<b>6</b>	<b>SIMATIC Safety V17 .....</b>	<b>211</b>
0	Preface .....	211
6.2	Introduction to example .....	212
6.2.1	Example Structure and Task Definition .....	212
6.2.2	Procedure.....	214
6.2.3	Requirements for configuring and programming .....	217
6.2.4	Additional information.....	218
6.3	Configuring .....	219
6.3.1	Introduction .....	219
6.3.2	Step 1: Configuring the CPU 1516F-3 PN/DP .....	220
6.3.3	Step 2: Configuring an ET 200SP distributed I/O system on a PROFINET subnet .....	222
6.3.4	Step 3: Configuring an F-DI module for connecting an emergency stop switch, position switches, and the laser scanner.....	223
6.3.5	Step 4: Configuring an F-DQ module for connecting a motor .....	227
6.3.6	Step 5: Configuring an F-DI module for user acknowledgement, feedback circuit and start .....	228
6.3.7	Step 6: Download hardware configuration .....	231
6.3.8	Step 7: Assign device name.....	232
6.3.9	Step 8: Assign PROFIsafe addresses .....	234
6.3.10	Summary: Configuring the Hardware .....	236

---

6.4	Programming.....	237
6.4.1	Introduction .....	237
6.4.2	The structure of the safety program .....	237
6.4.3	The Safety Administration Editor .....	239
6.4.4	Step 9: Specifying the centralized settings for the safety program.....	240
6.4.5	Step 10: Create PLC tag table .....	241
6.4.6	Step 11: Creating an F-FB.....	242
6.4.7	Step 12: Programming the safety door function .....	243
6.4.8	Step 13: Programming the emergency stop function .....	244
6.4.9	Step 14: Programming the feedback monitoring .....	245
6.4.10	Step 15: Programming the User Acknowledgment for Reintegration of the F-I/O .....	246
6.4.11	Step 16: Programming of the main safety block.....	247
6.4.12	Step 17: Compiling the safety program .....	248
6.4.13	Step 18: Download the safety program to the F-CPU.....	249
6.5	Setting up access protection .....	251
6.6	Typical configuring and programming errors and their causes .....	253
	<b>Index.....</b>	<b>255</b>

# Automation task

## 1.1 Introduction

### Introduction

In the following section, you will become familiar with the automation task.

You can find out more about the application example, the hardware configuration and the components of the sample project.

### Multimedia-based Getting Started

This is the link (<https://support.industry.siemens.com/cs/ww/en/view/78027451>) to find the Getting Started in multimedia form. Descriptive videos will familiarize you, through an automation task, with the engineering, programming and visualization of the S7-1500 with the TIA Portal.

### See also

Application (Page 12)

Hardware configuration (Page 13)

Sample project (Page 14)

## 1.2 Application

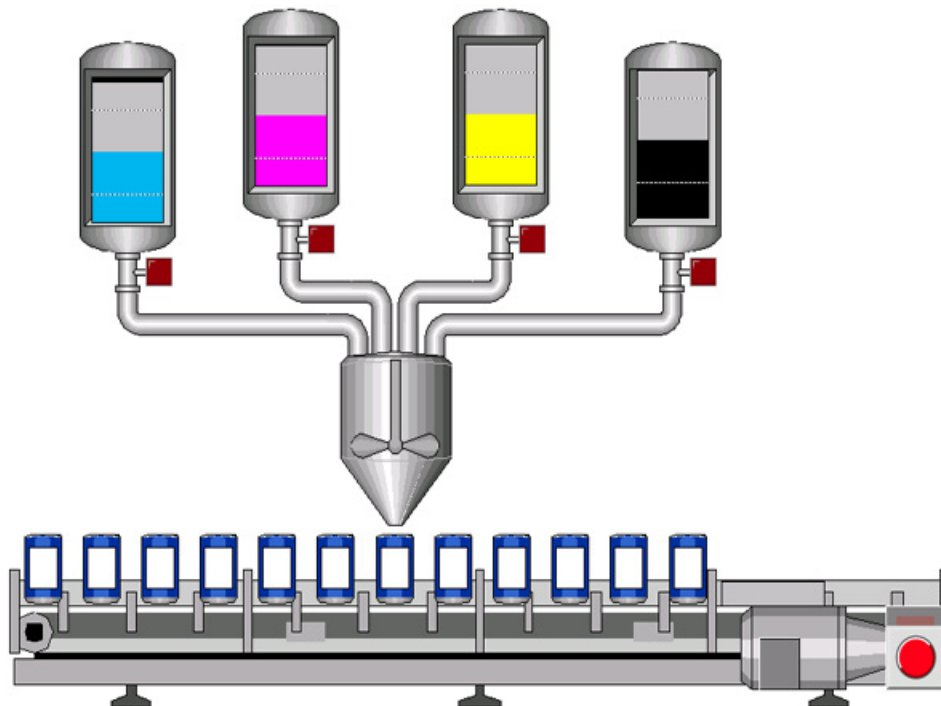
### Application example

The application example for this Getting Started is a color mixing plant for mixing and filling a previously selected color recipe.

There are four color components for the recipe, cyan, magenta, yellow and black, which means the colors of the CMYK color space.

Filling takes place in four steps.

- Selection of the color mixture using the HMI recipe function.
- Filling the recipe components or the four basic colors by opening the respective tank valves.
- Mixing the colors.
- Filling the finished color mixture into tins and transportation by a conveyor belt.

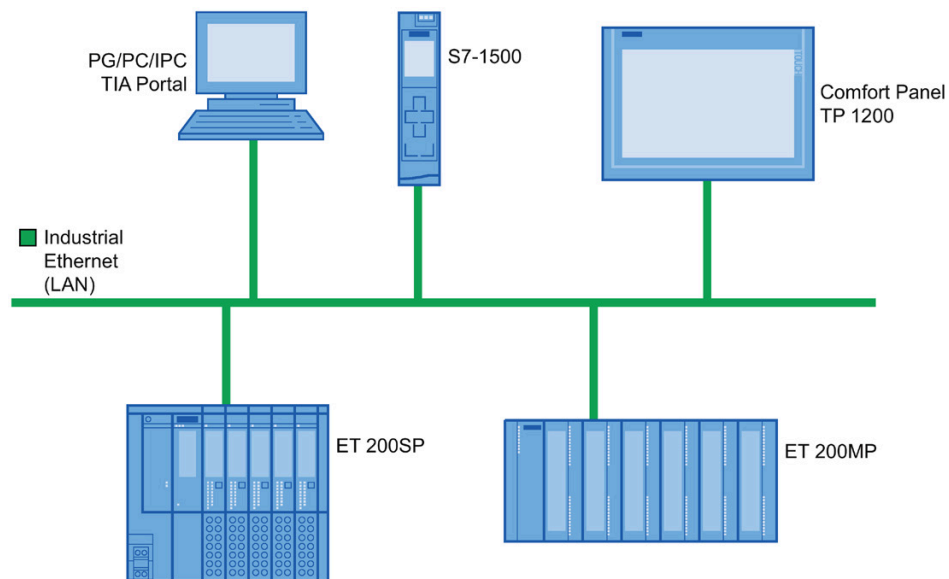


## 1.3 Hardware configuration

### Design of the hardware configuration

The hardware configuration consists of the following devices:

- The CPU 1511-1 PN with an S7-1500 load current supply, a digital input module and a digital output module.
- HMI Panel TP1200 Comfort that can also be simulated with the TIA Portal.
- Distributed ET 200MP I/O system with IM 155-5 PN ST interface module and digital input and digital output modules.
- Distributed ET 200SP I/O system with IM 155-6 PN ST interface module, digital input modules, digital output modules and server module.



## 1.4 Sample project

### Sample project for the application

To configure the color mixing system with the TIA Portal, create the sample project "Color\_Filling\_Station".

The following project components already exist for the sample project:

- The program blocks of the CPU
- The visualization of the HMI on a Comfort Panel

In this section, we will explain the relationships between the individual project components of the sample project.

You will carry out the necessary configuration steps yourself at a later point in time.

### See also

Selecting the recipe (Page 15)

Displaying the CMYK and RGB values (Page 16)

Filling the recipe (Page 17)

Starting the mixing process (Page 18)

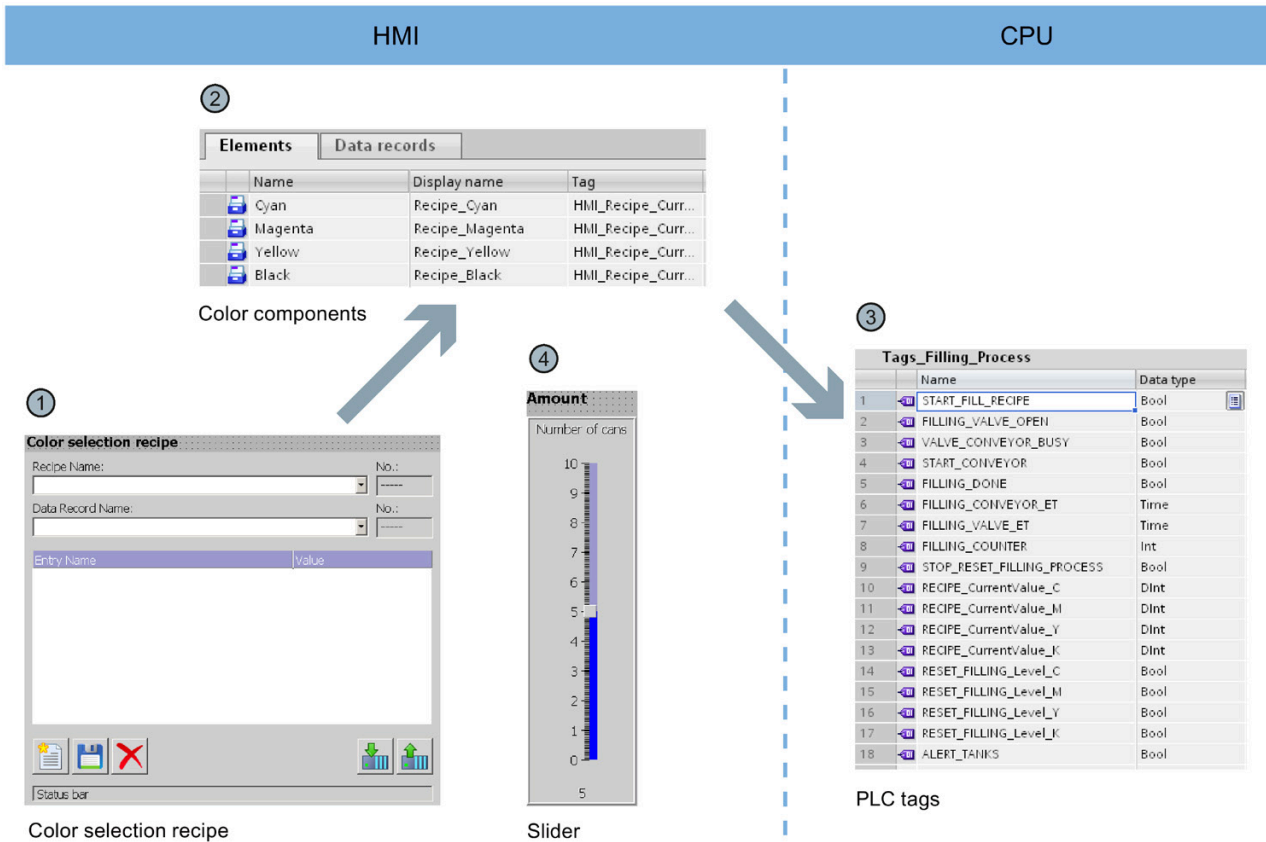
Filling the color mixture (Page 19)

Resetting the fill level (Page 20)



## 1.4.1 Selecting the recipe

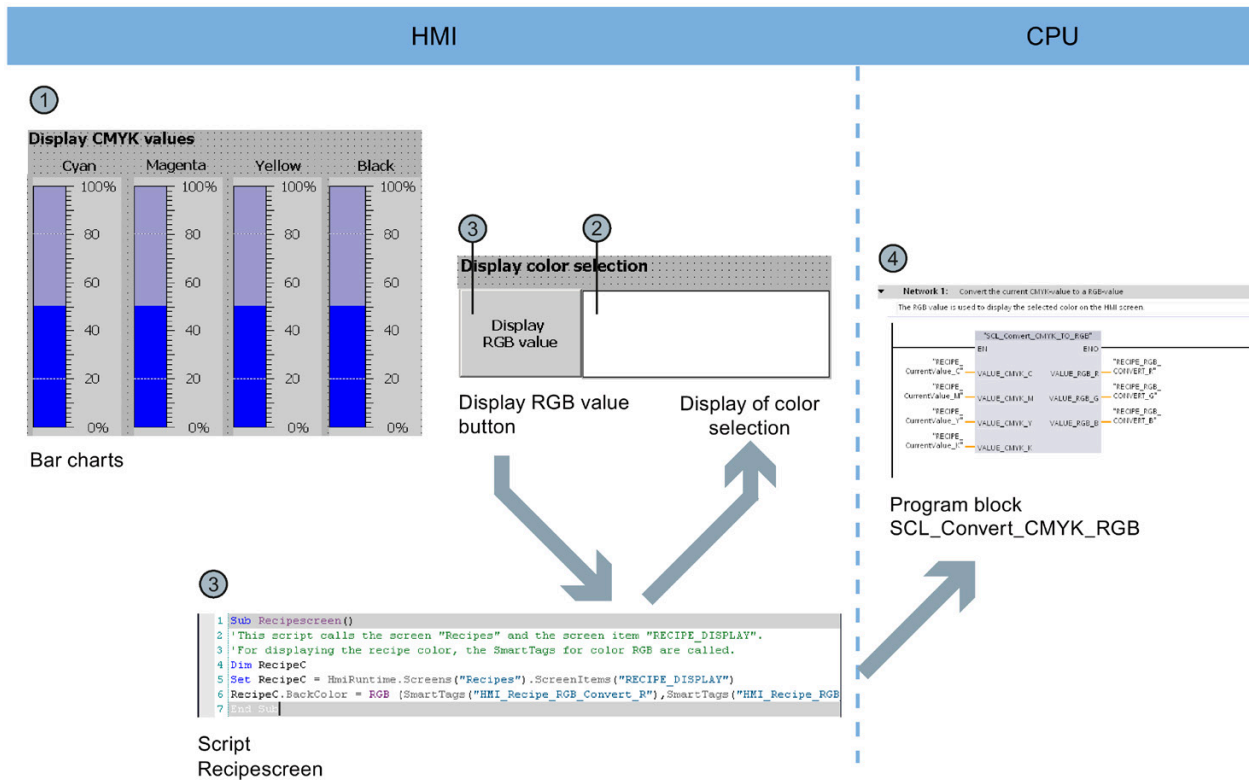
### Selecting the recipe



- ① The HMI screen "Recipes" includes the "Color selection recipe". This is a prefabricated object from the library of the TIA Portal. You can use this object to select data records and to create new data records.
- ② The data records (color mixtures) and elements (color components) are stored in the "Recipes" HMI editor. Each color mixture consists of the four color components cyan (C), magenta (M), yellow (Y), black (K). The proportion of each of the four color components in a color mixture is stored in the "Recipes" editor.
- ③ The values for the respective color components are written in PLC tags when you load a color mixture. The PLC tags are stored in the "Tags\_Filling\_Process" tag table of the CPU.
- ④ The HMI screen "Recipes" also includes a slider. You use this slider to specify the number of tins to be filled.

## 1.4.2 Displaying the CMYK and RGB values

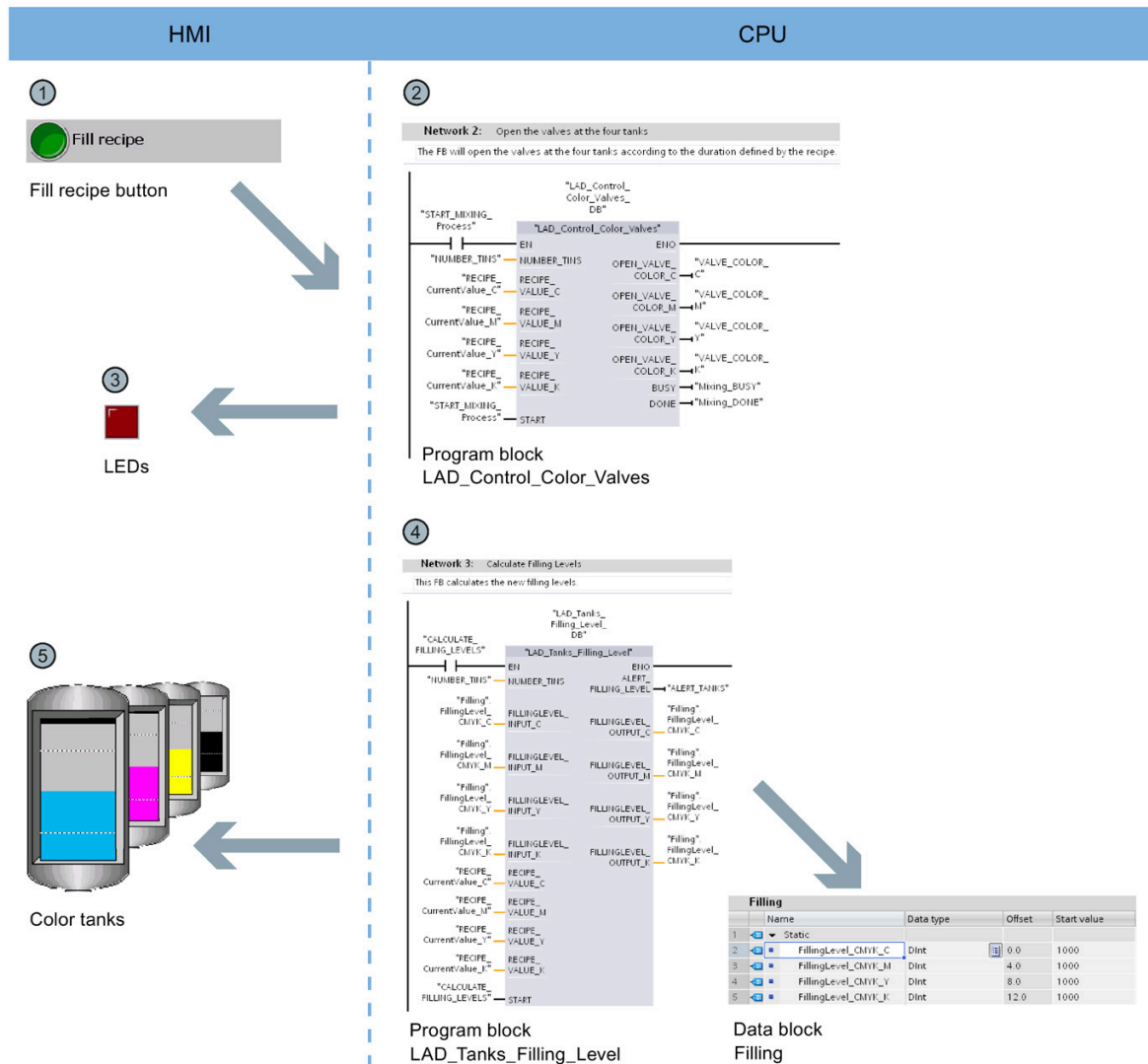
### Displaying the CMYK and RGB values



- ① When the required color mixture is selected in the HMI screen "Recipes", the values are displayed in the CMYK color space by means of a bar diagram.
- ② The color mixture can be shown with an additional display. This requires the "Recipescreen" script to be run.
- ③ You run the "Recipescreen" script by clicking the "Display RGB Value" button. The script assigns the RGB value assigned to the CMYK value to the display, because CMYK values cannot be output directly on screens.
- ④ The required RGB value is calculated by the "SCL\_Convert\_CMYK\_TO\_RGB" program block.

### 1.4.3 Filling the recipe

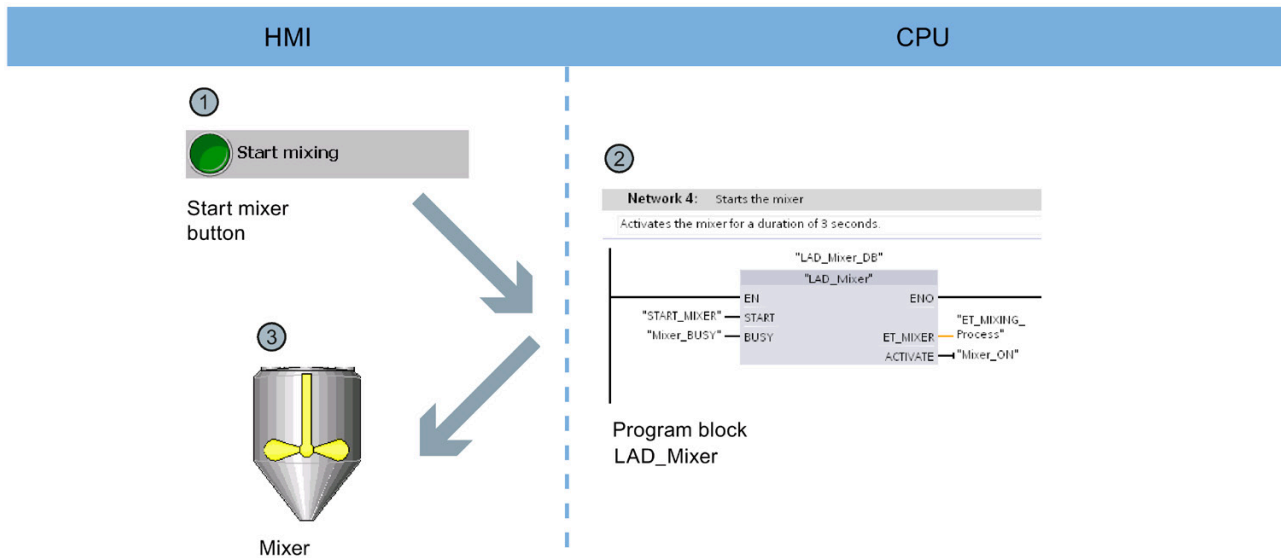
#### Filling the recipe



- ① The "Fill recipe" button starts the filling of the color components in the HMI screen "Start screen". The button activates the "LAD\_Control\_Color\_Valves" program block.
- ② The program block calculates how long each of the four valves needs to stay open for the color mixture based on the specified recipe and the number of tins that have to be filled.
- ③ LEDs below the tanks indicate that the valves are opened.
- ④ The "LAD\_Tanks\_Filling\_Level" program block is executed at the same time as the filling. The program block calculates the quantity remaining in the tank for the tank fill level. The fill levels of the tanks are stored in the global data block "Filling".
- ⑤ The fill level indicators in the HMI screen are directly linked with the global data block and are updated with each runtime acquisition cycle.

### 1.4.4 Starting the mixing process

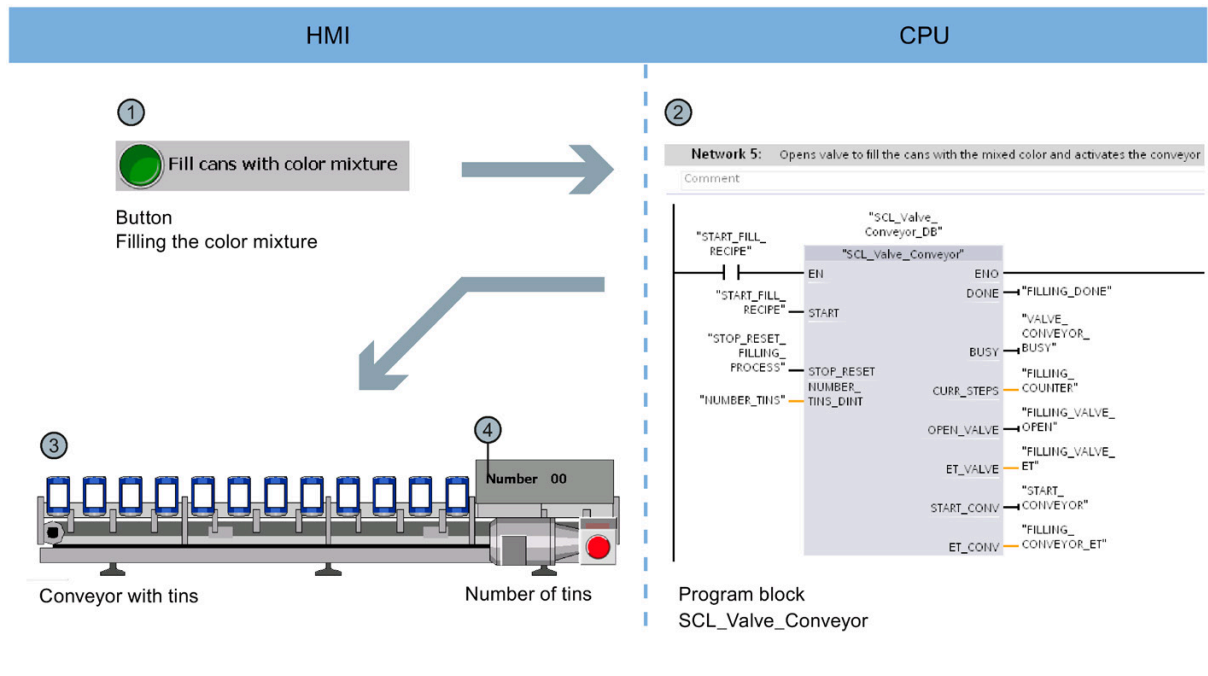
#### Starting the mixing process



- ① The "Start mixing process" button starts the mixer of the color mixing plant in the HMI screen "Start screen".
- ② The "LAD\_Mixer" program block is called for this purpose at the CPU end. It activates the mixer for three seconds.
- ③ The activation of the mixer is indicated by flashing in the HMI screen.

## 1.4.5 Filling the color mixture

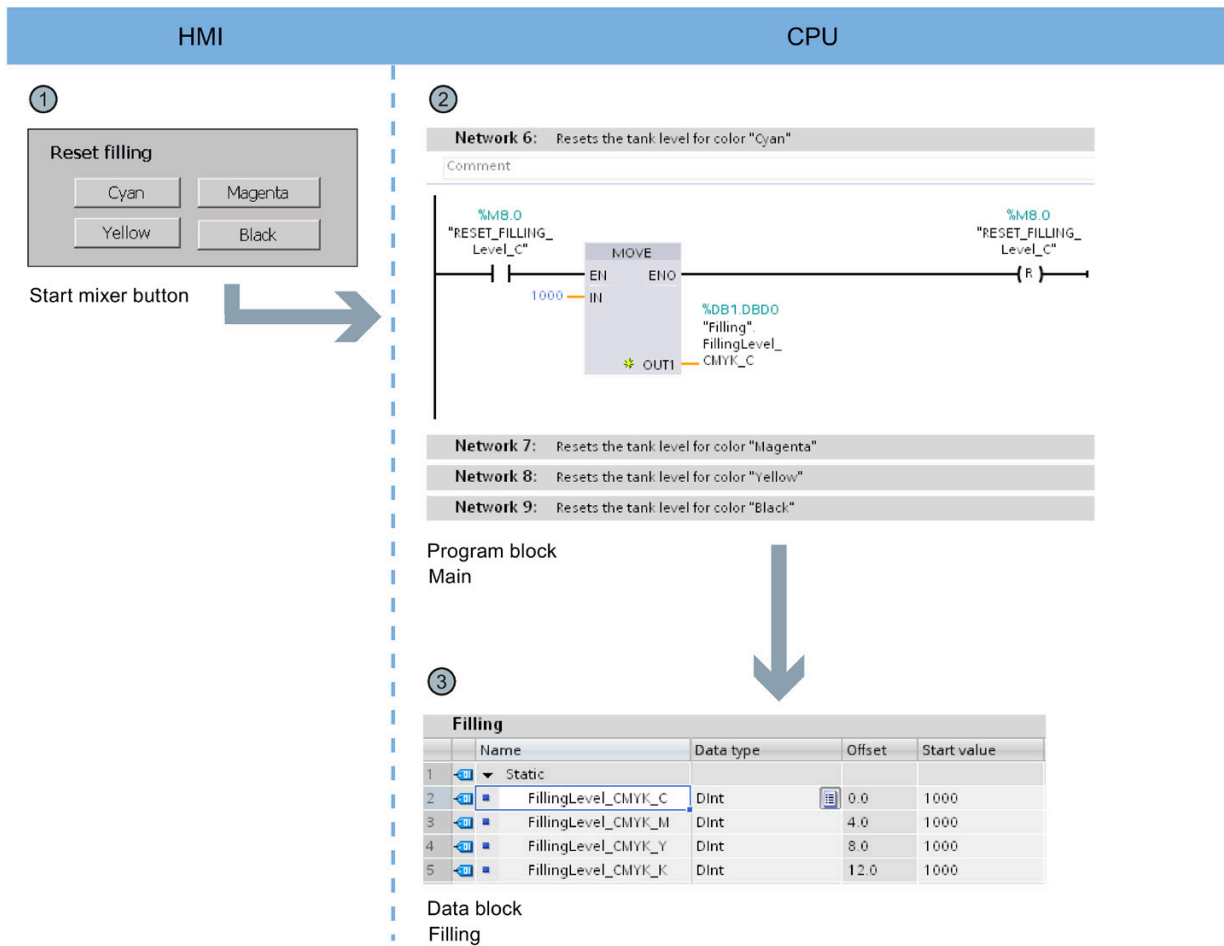
### Filling the color mixture



- ① The "Fill Color Mixture" button starts the filling of the tins in the HMI screen "Start screen".
- ② The "SCL\_Valve\_Conveyor" program block is activated for this purpose at the CPU end; it controls the valves and the conveyor belt.
- ③ The tins are animated according to the movement of the conveyor belt in the HMI screen.
- ④ A counter indicates the number of tins that have already been filled.

## 1.4.6 Resetting the fill level

### Resetting the fill level



- ① There is a button to reset the fill level of each of the four color tanks in the HMI screen "Start screen".
- ② The reset of the respective fill level has been implemented in the "Main" program block in networks 6 to 9.
- ③ Networks 6 to 9 reset the values to the Start value in the global data block "Filling".

## Hardware section

### 2.1 Introduction

The controller family SIMATIC S7-1500 with the Totally Integrated Automation Portal (TIA Portal) offers you numerous options to increase the productivity of your machines and to make the engineering process even more efficient. Explore the options in this Getting Started.

In the first basic steps, you will get to know the hardware better. We will also show you how to configure and program the SIMATIC S7-1500 with SIMATIC STEP 7 V17 (TIA Portal). The connection of a SIMATIC HMI Comfort Panel with SIMATIC WinCC Advanced V17 (TIA Portal) or SIMATIC WinCC Professional V17 (TIA Portal) completes the basic steps.

#### 2.1.1 Requirements

##### Hardware requirements

To implement the hardware section of this Getting Started, you will need:

- 1 × CPU 1511-1 PN (6ES7511-1AK02-0AB0)
- 1 × S7-1500 load current supply PM 70W 120/230VAC (6EP1332-4BA00)
- 1 × Mounting rail (6ES7590-1AB60-0AA0)
- 1 × digital input module DI 16x24VDC SRC BA (6ES7521-1BH50-0AA0)
- 1 × digital output module DQ 16x24V DC/0.5A HF (6ES7522-1BH01-0AB0)
- 2 × Front connectors (6ES7592-1AM00-0XB0)
- 1 × SIMATIC Memory Card with at least 4 MB (e.g. 6ES7954-8LC03-0AA0)
- 1 × Ethernet cable

## Software requirements

To implement the software section of the Getting Started, you will need:

- SIMATIC STEP 7 Professional V17
- SIMATIC WinCC Advanced V17 or SIMATIC WinCC Professional V17



### **WARNING**

#### **Severe personal injury may result**

The S7-1500 automation system in plants or systems is governed by specific standards and regulations, based on the relevant field of application. Observe the applicable safety and accident prevention regulations such as IEC 60204-1 (general machine safety requirements).

Failure to observe these regulations can result in serious injuries and damages to machinery and facilities.



## 2.1.2 Additional information

Detailed information on the hardware used is available here:

- CPU 1511-1 PN (6ES7511-1AK02-0AB0)  
(<https://support.industry.siemens.com/cs/ww/en/view/109752841>)
- S7-1500 load current supply PM 70W 120/230VAC (6EP1332-4BA00)  
(<https://support.industry.siemens.com/cs/ww/en/view/68036174>)
- Digital input module DI 16x24VDC SRC BA (6ES7521-1BH50-0AA0)  
(<https://support.industry.siemens.com/cs/ww/en/view/59191844>)
- Digital output module DQ 16x24V DC/0.5A HF (6ES7522-1BH01-0AB0)  
(<https://support.industry.siemens.com/cs/ww/en/view/109480717>)

## 2.2 Installing the assembly

### 2.2.1 Overview

#### Mounting the assembly

You mount the structure in this section.

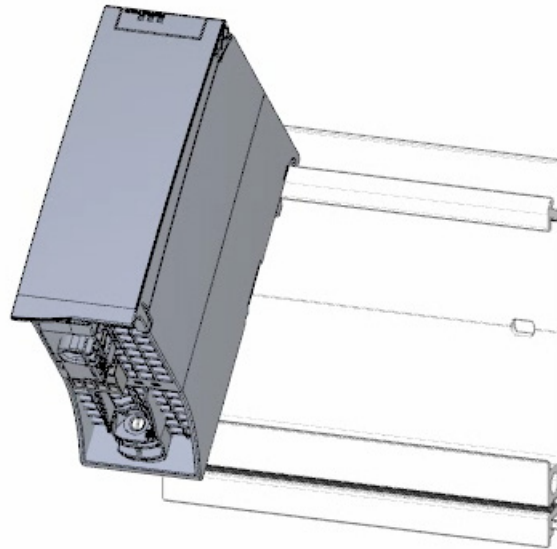
#### See also

Installing the assembly (Page 24)

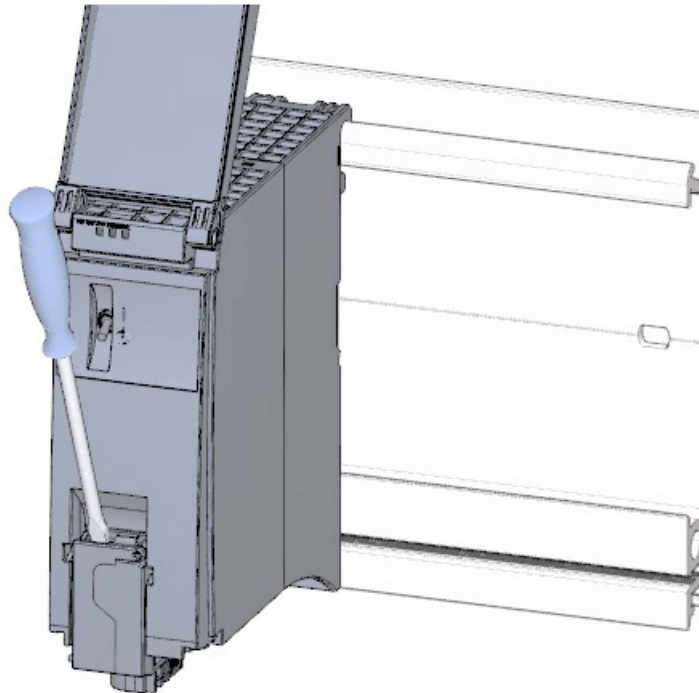
## 2.2.2 Installing the assembly

### Procedure

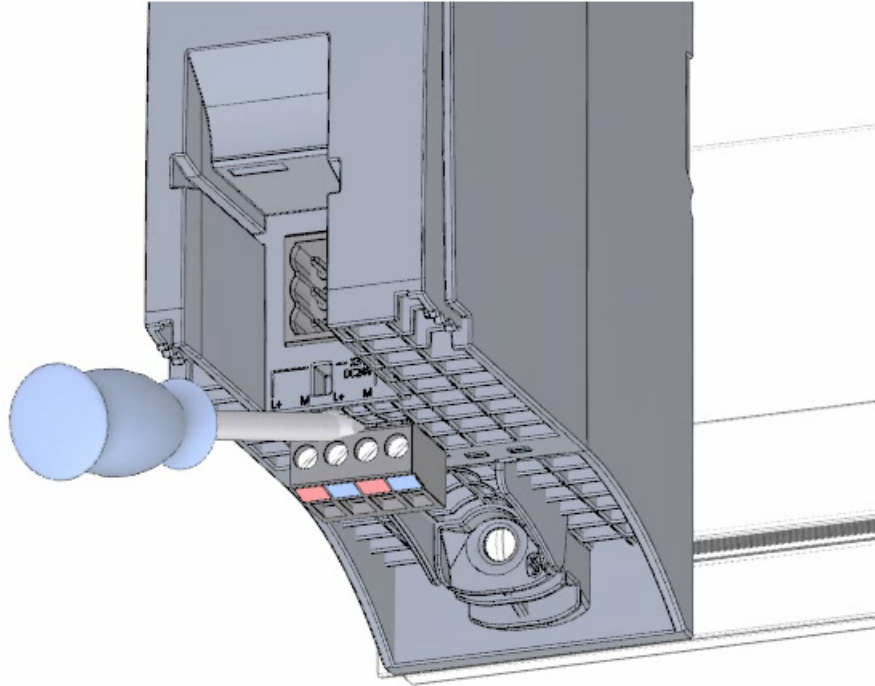
1. Mount the load current supply (PM) on the mounting rail.



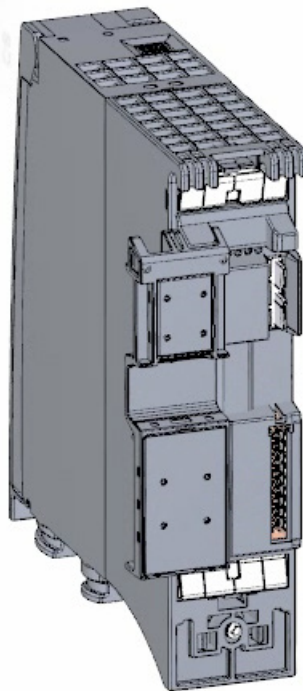
2. Open the front cover and pull out the mains connection plug.



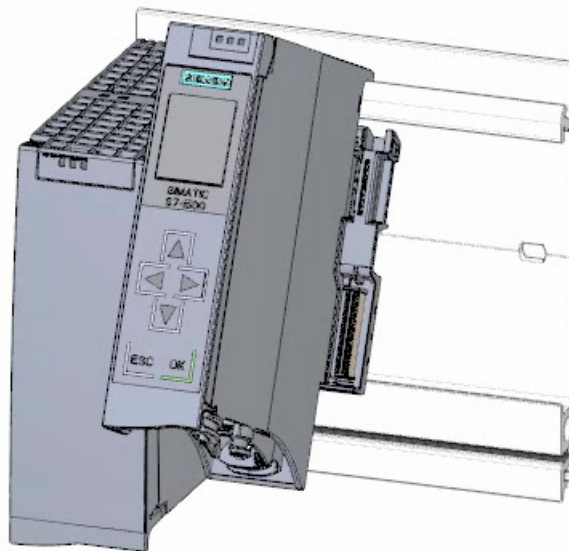
3. Remove the 4-pole connection plug and screw the load current supply (PM) tight.



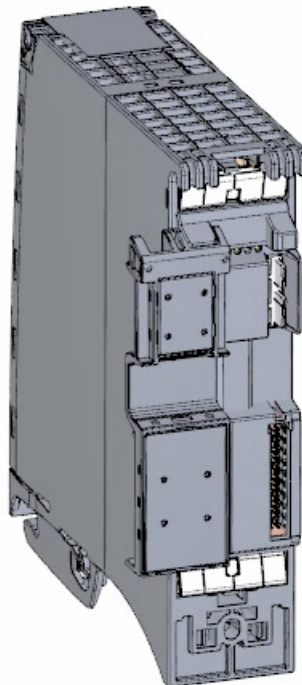
4. Insert the U-connector into the back of the CPU.



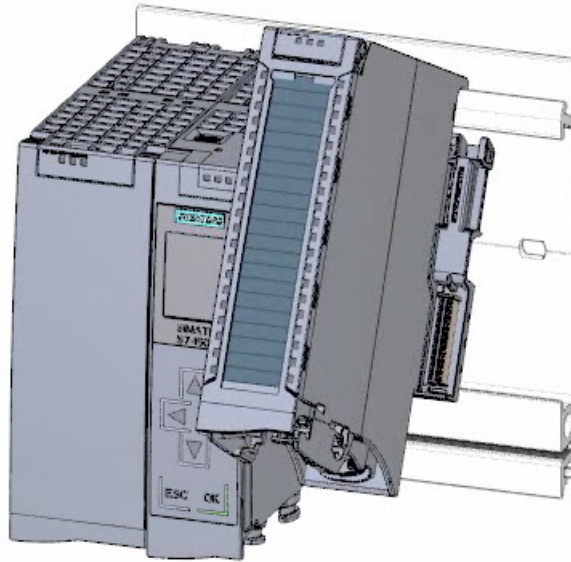
5. Mount the CPU on the mounting rail and screw tight.



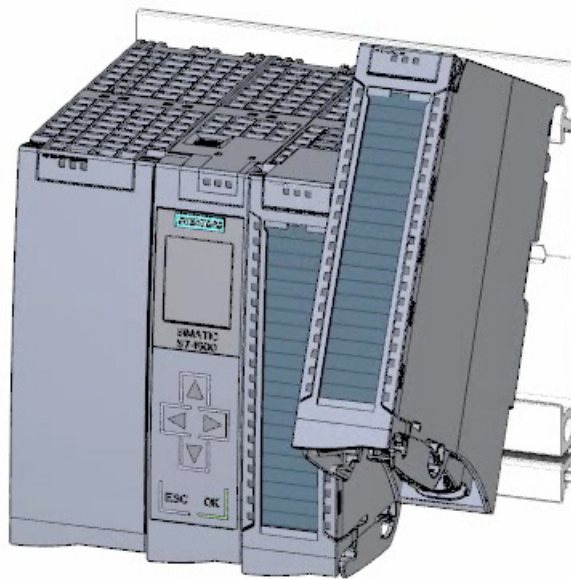
6. Insert the U-connector into the back of the digital input module.



7. Attach the digital input module to the mounting rail and screw tight.



8. Attach the digital output module to the mounting rail and screw tight.



## Result

The assembly has been mounted.

## 2.3 Wiring

### 2.3.1 Overview

#### Wiring the assembly

You mount the assembly in this section.



The mains cable for the load current supply must not be connected to the power supply during wiring.

## 2.3.2 Wiring rules

Operation of an S7-1500 CPU in plants or systems is defined by special set of rules and regulations, based on the relevant field of application.

You can find the general rules and regulations for operating the S7-1500 in the S7-1500 system description (<https://support.industry.siemens.com/cs/ww/en/view/59191792>).

### Wiring rules for the CPU

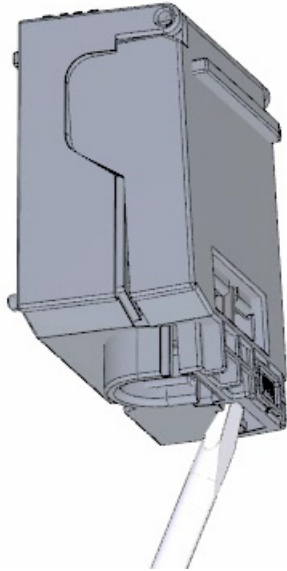
Wiring rules...		CPU	40-pin front connector (screw-type connection)	Load power supply
Connectible wire cross-sections for solid wires		—	up to 0.25 mm <sup>2</sup>	—
		—	AWG*: 24	—
Connectible wire cross-sections for stranded wires	Without wire end ferrule	0.25 to 2.5 mm <sup>2</sup>	0.25 to 1.5 mm <sup>2</sup>	1.5 mm <sup>2</sup>
		AWG*: 24 to 16	AWG*: 24 to 16	AWG*: 16
	With wire end ferrule	0.25 to 2.5 mm <sup>2</sup>	0.25 to 1.5 mm <sup>2</sup>	1.5 mm <sup>2</sup>
		AWG*: 24 to 16	AWG*: 24 to 16	AWG*: 16
Number of wires per connection		1	1 or a combination of 2 cables up to 1.5 mm <sup>2</sup> (total) in the same wire end ferrule	1
Length of stripped wires		10 to 11 mm	10 to 11 mm	7 to 8 mm
End sleeves according to DIN 46228	without plastic sleeve	Design A, 10 mm long	Design A, 10 mm and 12 mm long	Design A, 7 mm long
	with plastic sleeve 0.25 to 1.5 mm <sup>2</sup>	Design E, 10 mm long	Design E, 10 mm and 12 mm long	Design A, 7 mm long
Sheath diameter		—	—	8.5 mm
Tool		3 to 3.5 mm Phillips screwdriver, conic design	3 to 3.5 mm Phillips screwdriver, conic design	3 to 3.5 mm Phillips screwdriver, conic design
Connection system		Push-in terminal	Screw terminal	Screw terminal
Tightening torque		—	from 0.4 Nm to 0.7 Nm	from 0.5 Nm to 0.6 Nm

\* AWG: American Wire Gauge

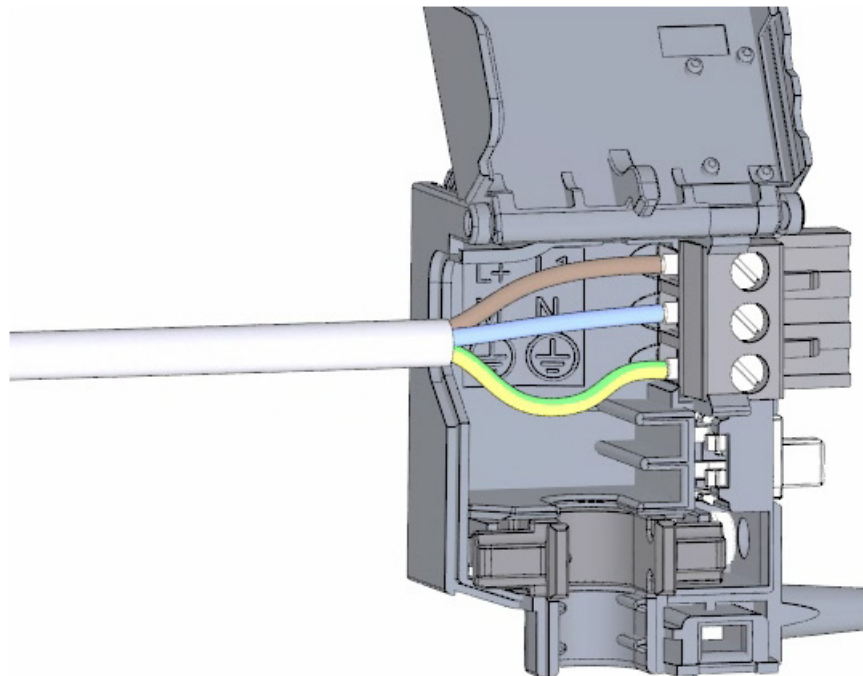
### 2.3.3 Wiring the mains connection plug

#### Procedure

1. Pry off the connector cover using a suitable tool.

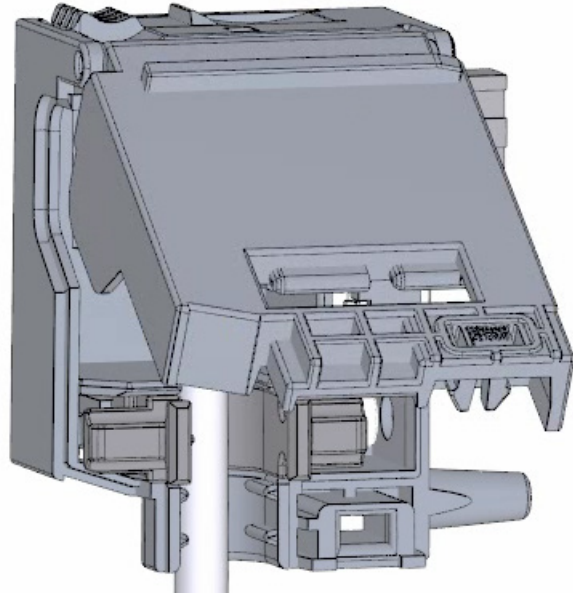


2. Connect the mains cable in the plug according to the connection diagram. You will find information on which voltage the plug is approved for on the side of the plug. You select the voltage by inserting the coding element accordingly on the back of the plug.

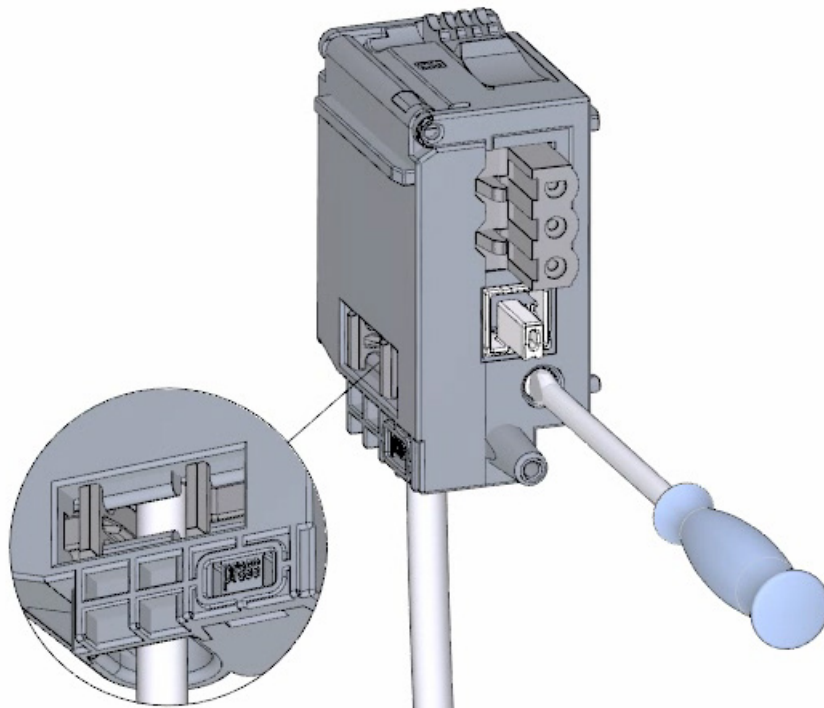




3. Close the cover.



4. Tighten the screw on the front of the mains connection plug.



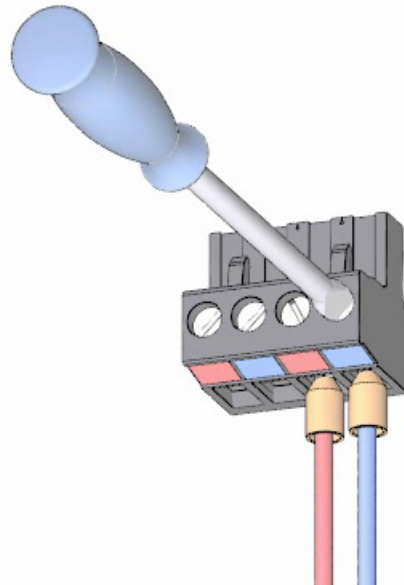
## Result

The mains connection plug is now wired.

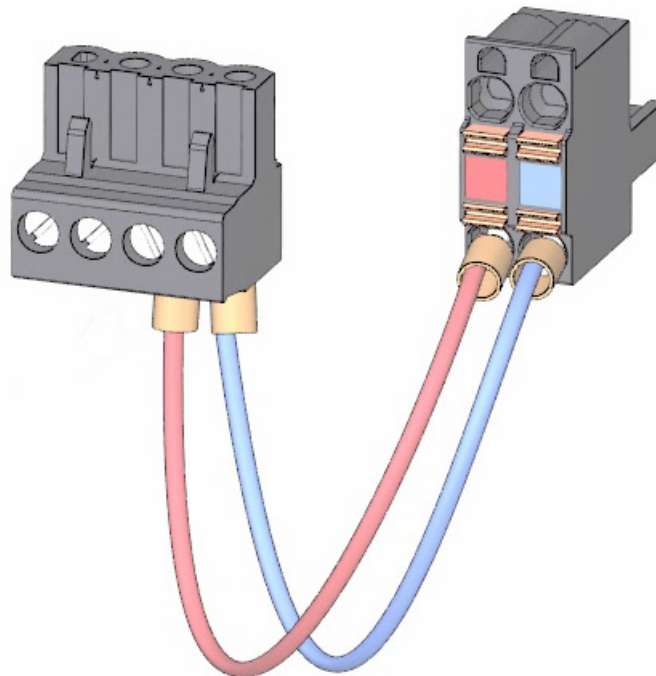
### 2.3.4 Wiring the load current supply (PM) to the CPU

#### Procedure

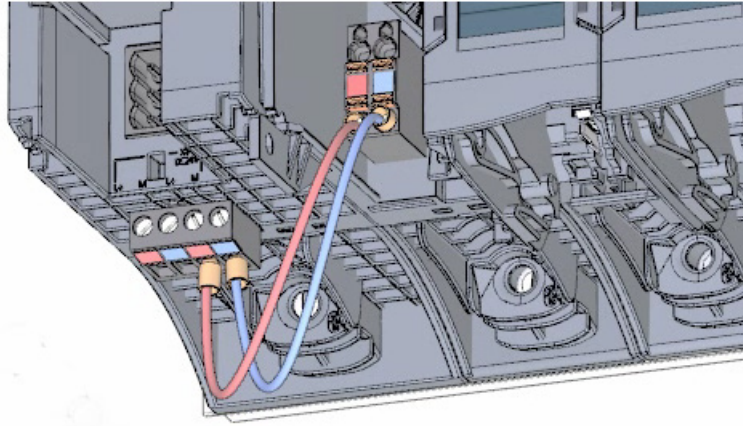
1. Wire the 4-pin connector plug of the load current supply (PM).



2. Wire the 4-pin connector plug with the 4-pin mains connection plug of the CPU.



3. Connect the load current supply (PM) to the CPU.



### Result

The load current supply is now wired to the CPU.

## 2.3.5 Potential bridge circuits

### Application of the potential bridge circuits

If you want to supply the load groups with the same potential (non-isolated), use the potential circuit bridges supplied for the front connector. This means that you avoid having to wire a clamping unit with two wires.

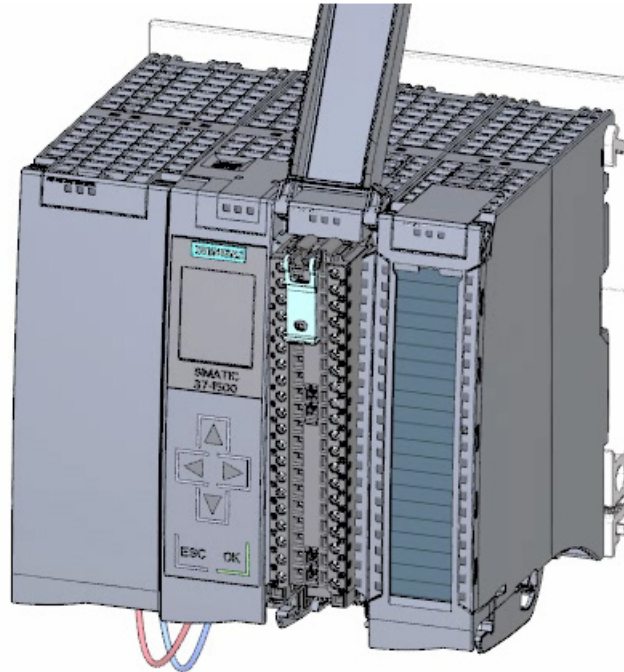
### Tip

Use the terminals 40 (M) and 39 (L+) on the front connector to loop the potential to the next module.

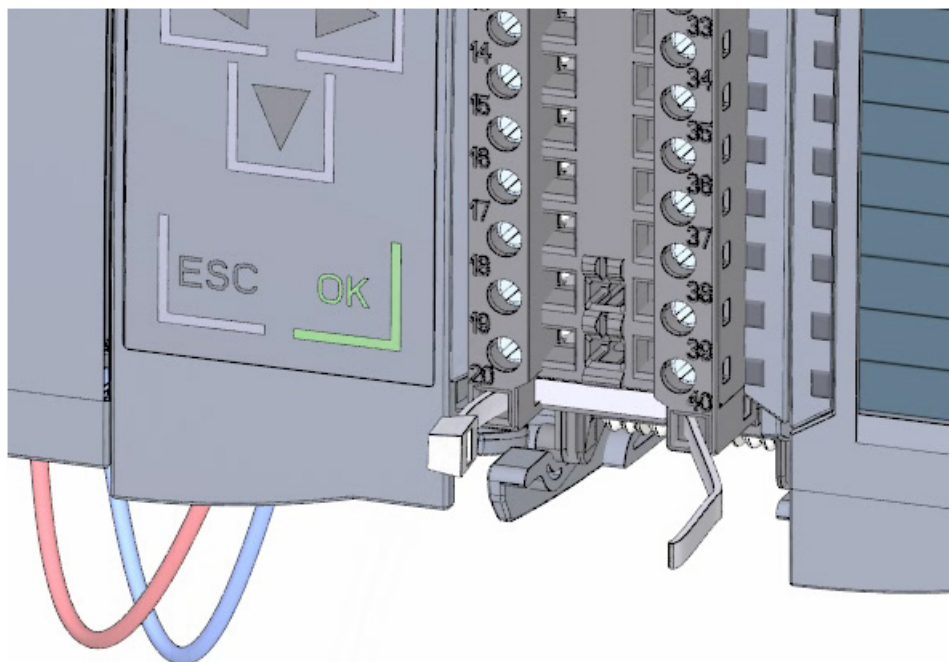
## 2.3.6 Wiring the digital input module

### Procedure

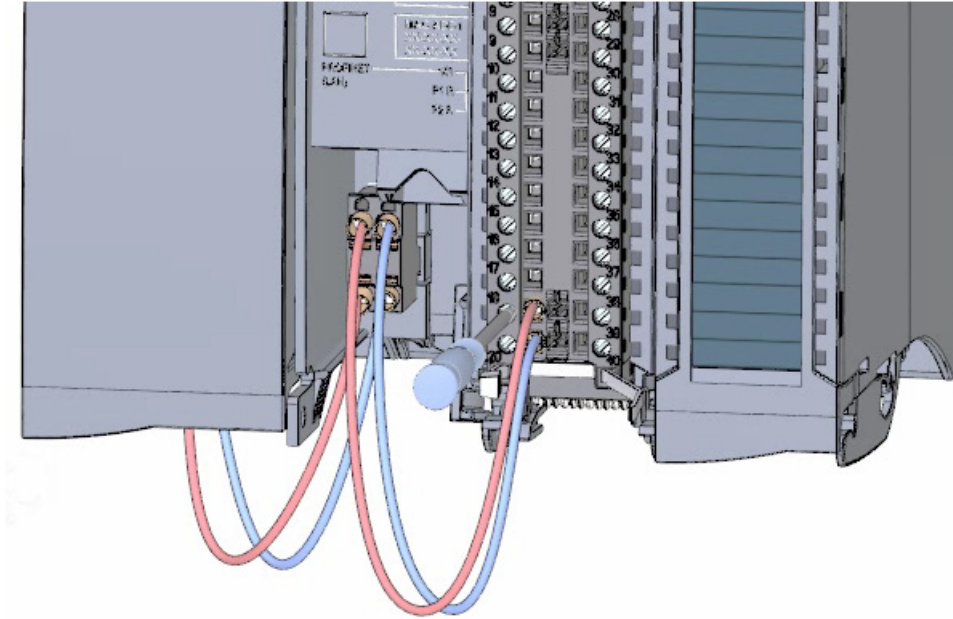
1. Insert the front connector into the pre-wiring position. There is no electrical connection between the front connector and the module in the pre-wiring position.



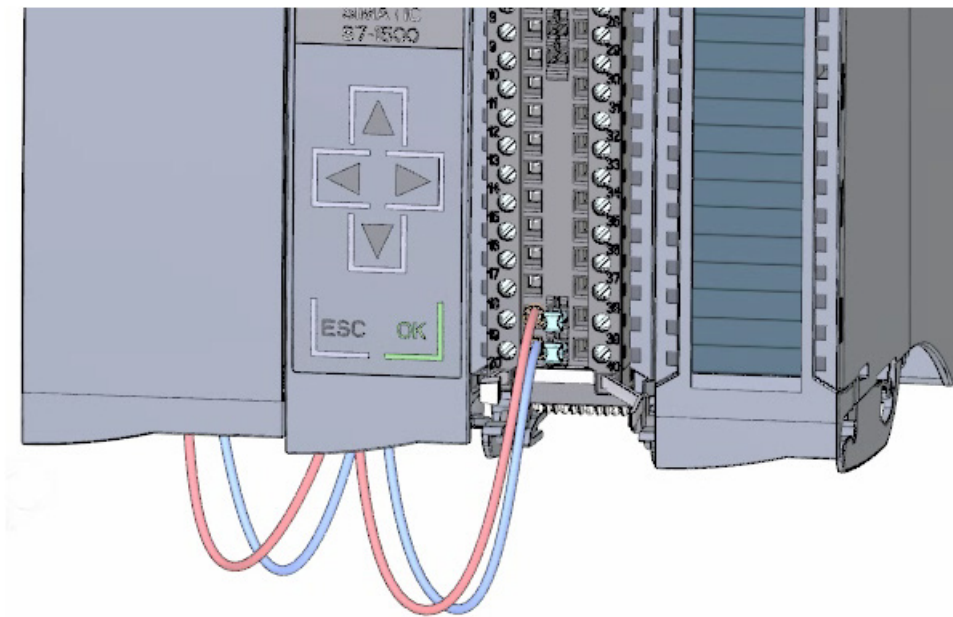
2. Thread in the cable tie.



3. Connect the supply voltage 24 V DC to the terminals 20 (M) and 19 (L+).



4. Insert the potential circuit bridges between the two bottom terminals.



## Result

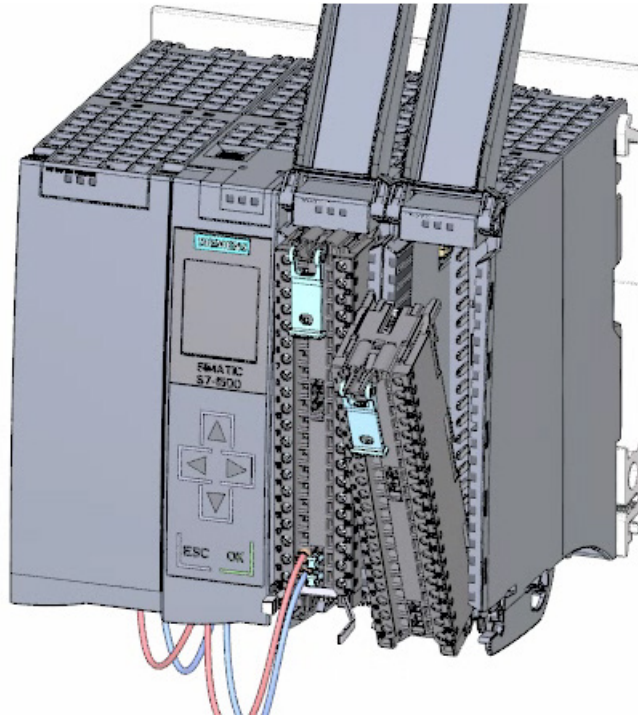
The digital input module is now wired.



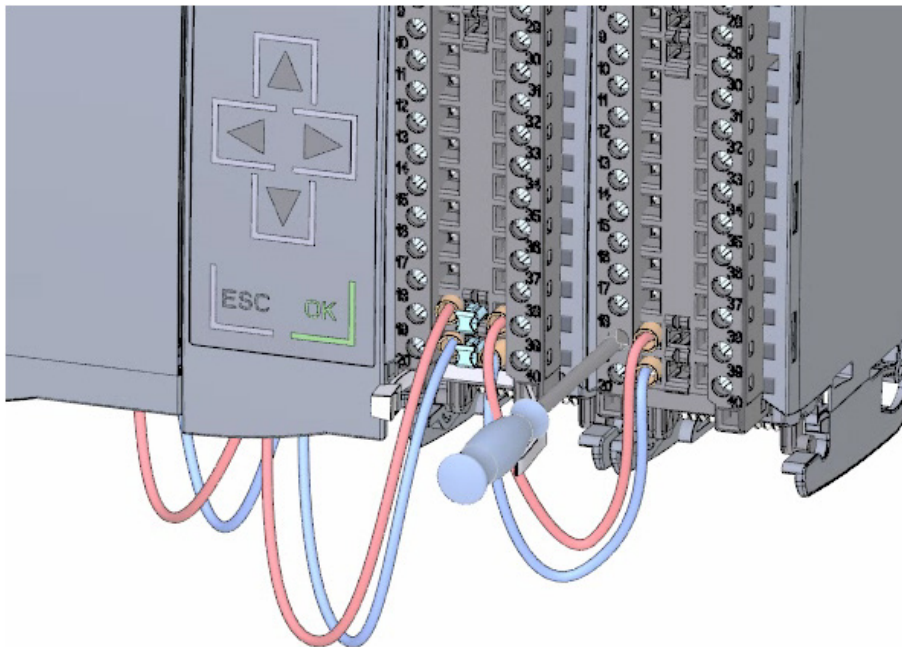
### 2.3.7 Wiring the digital output module

#### Procedure

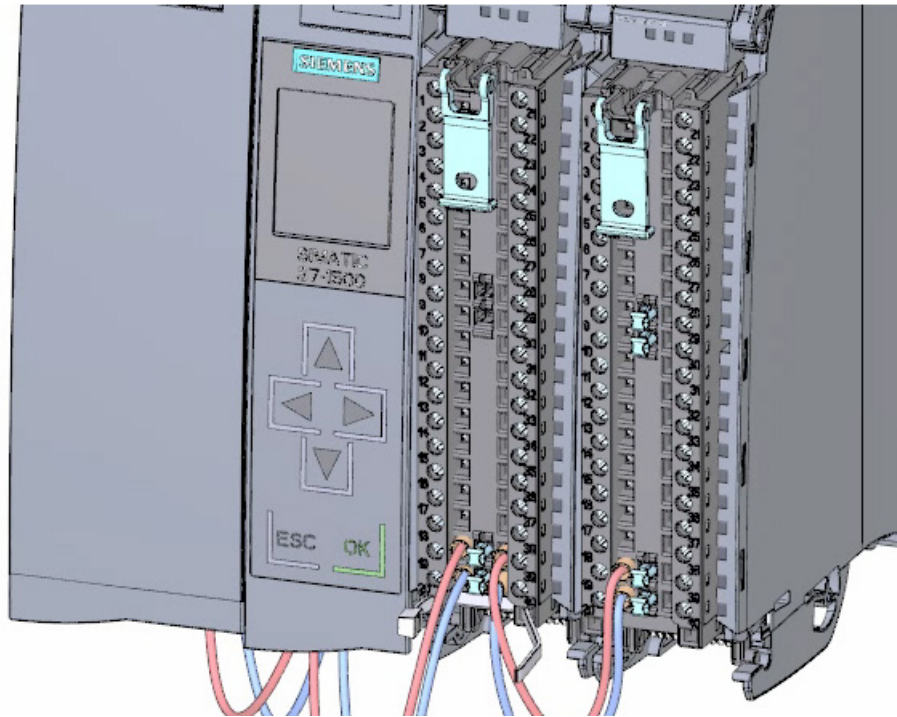
1. Insert the front connector into the pre-wiring position.



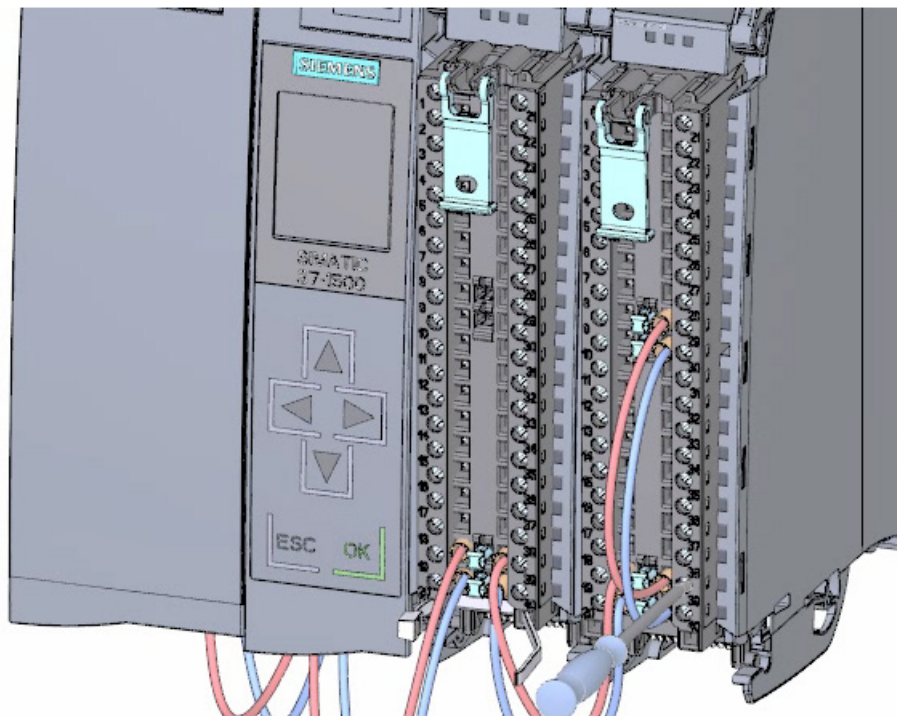
2. Use terminals 40 (M) and 39 (L+) from the digital input module to feed the supply voltage DC 24 V from the digital input module to terminals 20 (M) and 19 (L+).



3. Connect the four potential circuit bridges.



4. Connect the terminals 30 and 40, as well as 29 and 39 to each other.



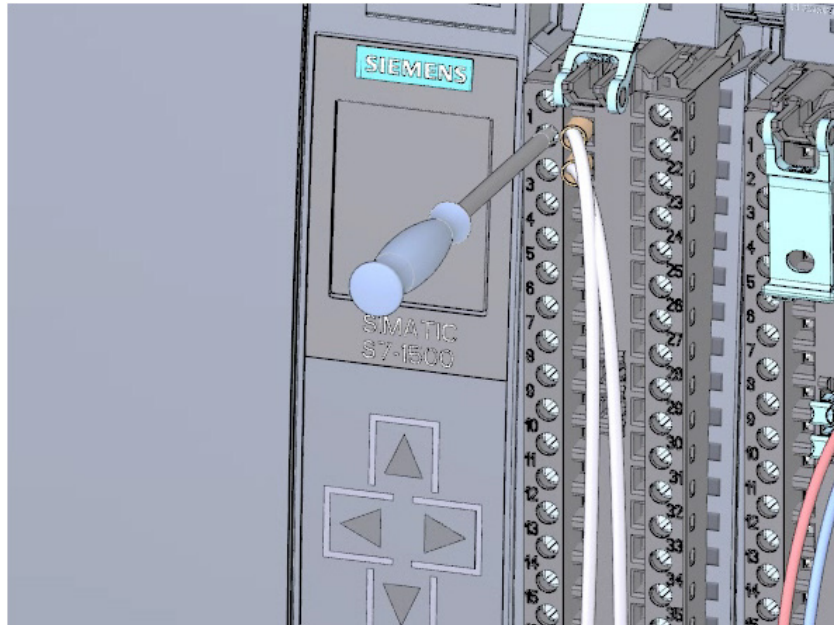
## Result

The digital output module is now wired.

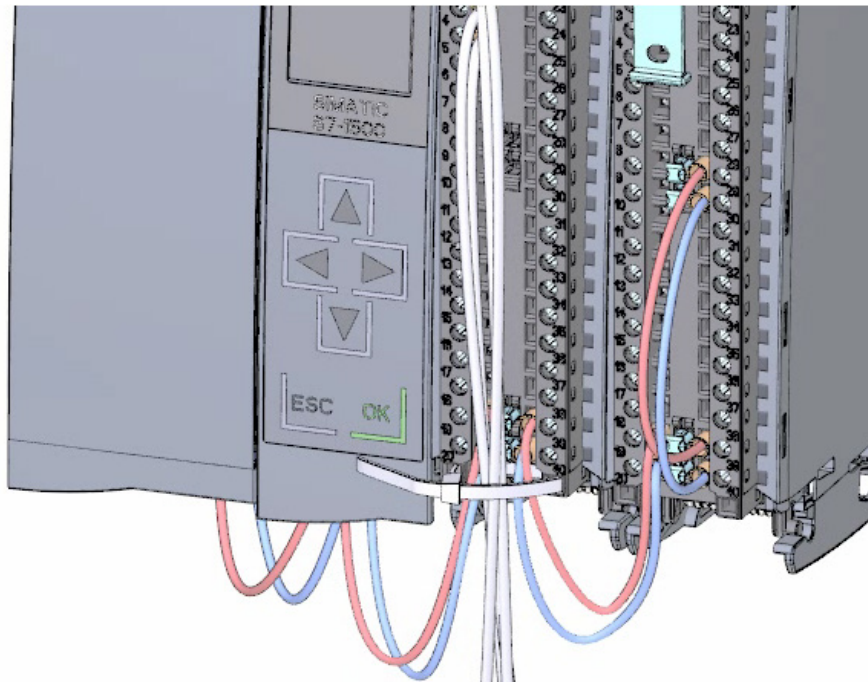
## 2.3.8 Wiring front connectors

### Procedure

1. Connect the individual wires according to the connection diagram on the inner side of the front cover in the terminal and screw tight.

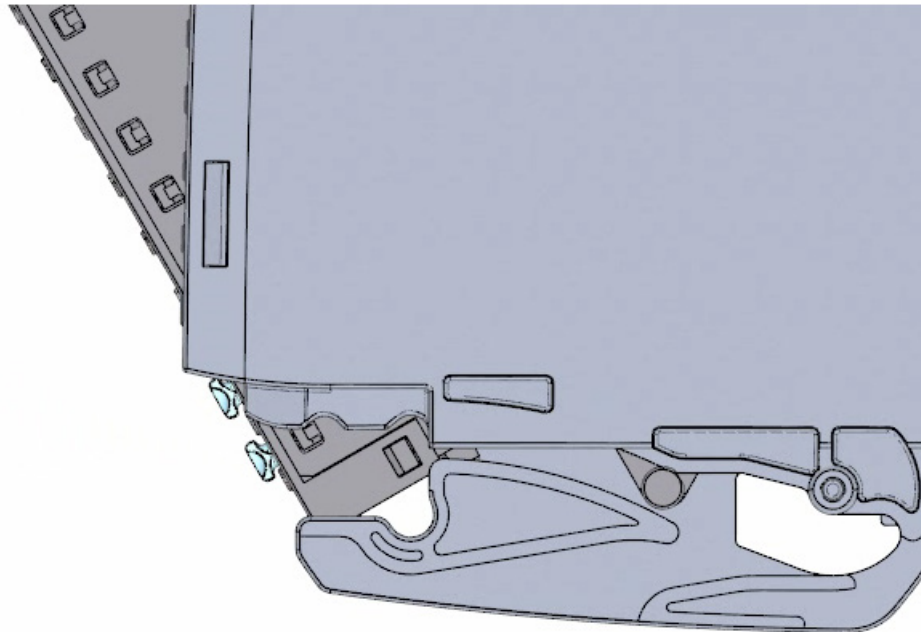


2. For strain relief, run the cable tie around the cable harness and pull tight.

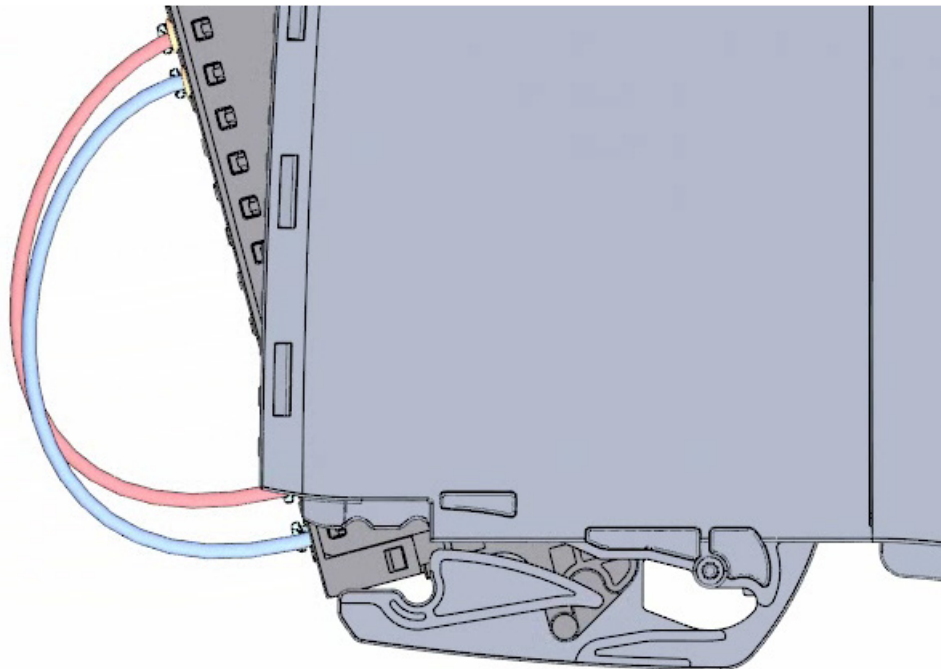




3. Move the front connector from the pre-wiring position to its final position. By doing this, you create an electrical connection between the front connector and the module.



4. **Tip:** Pre-wired front connectors, e.g. for replacing modules, can be inserted directly.



## Result

The front connectors are now wired.

## 2.4 Power on

### 2.4.1 Overview

#### Turning on the CPU for the first time

You turn on the CPU for the first time in this section.

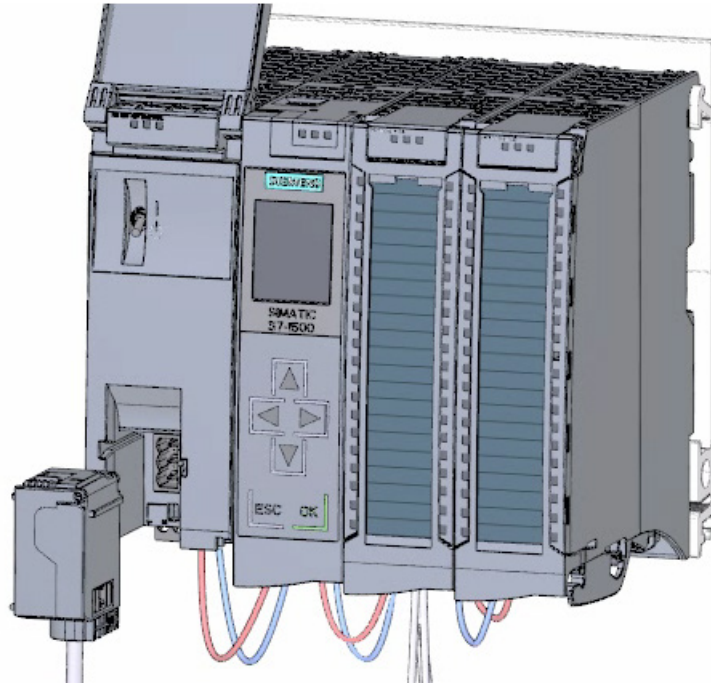
#### See also

Power on (Page 41)

## 2.4.2 Power on

### Procedure

1. Insert mains connection plug of the load current supply (PM).



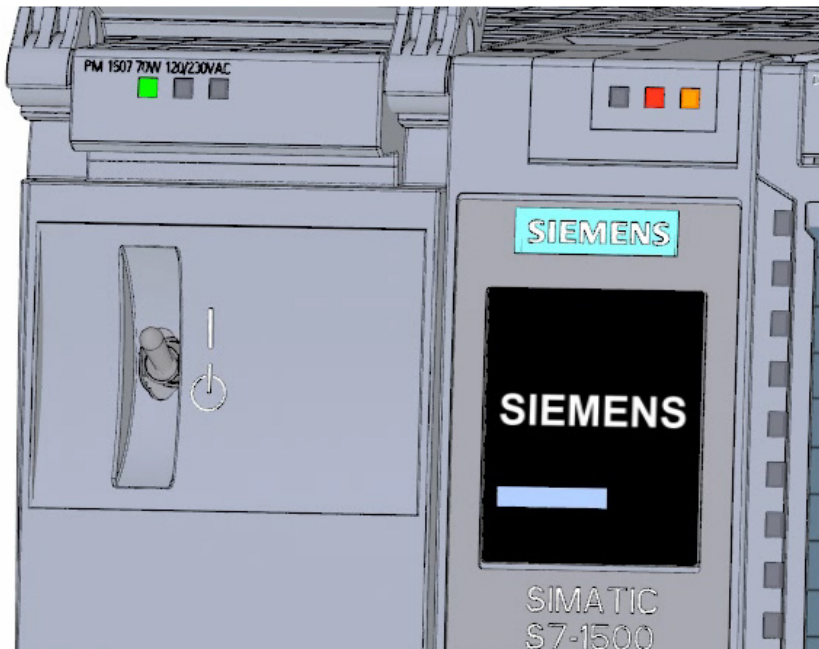
2. Connect the mains connection plug to the power supply.



3. Insert a blank SIMATIC memory card into the CPU.



4. Move the switch for the load current supply (PM) to the position RUN.  
The CPU starts up.



## Result

The CPU starts up and is in STOP mode.

### 2.4.3 Assign IP address via the display

In this step you set the IP address and the subnet mask for the CPU.

#### Procedure

1. Navigate to "Settings".
2. Select "Addresses".
3. Select the interface "X1 (IE/PN)".
4. Select the menu item "IP Addresses".
5. Set the IP address 192.168.0.10.
6. Press the "right" arrow key on the module.
7. Set the subnet mask 255.255.255.0.
8. Press the "down" arrow key on the module to select the menu item "Apply" and confirm the setting with "OK"

#### Result

You have now assigned an IP address and the subnet mask for the interface "X1 (IE/PN)".

## Software section

### 3.1 Creating the project and hardware

#### 3.1.1 Introduction to the TIA Portal

##### Introduction

The Totally Integrated Automation Portal, referred to as TIA Portal in the following, offers all the functions you need for implementing your automation task assembled in a single, cross-software platform.

The TIA Portal is the first shared working environment for integrated engineering with the various SIMATIC systems made available within a single framework. The TIA Portal therefore also enables reliable, convenient cross-system collaboration for the first time.

All required software packages, from hardware configuration and programming to visualization of the process are integrated in a comprehensive engineering framework.



### Advantages of working with the TIA Portal

The following features provide efficient support during the realization of your automation solution when working with the TIA Portal:

- **Integrated engineering with a uniform operating concept**  
Process automation and process visualization go "hand-in-hand".
- **Consistent, centralized data management with powerful editors and universal symbols**  
Data created once is available in all editors. Changes and corrections are automatically applied and updated within the entire project.
- **Comprehensive library concept**  
Use the ready-made instructions and pre-existing parts of the project again and again.
- **Multiple programming languages**  
Five different programming languages are available for implementing your automation task.

### 3.1.2 Creating a project

#### Introduction

In the following step, you will create a new project.

All data which is generated during the creation of an automation solution is saved in the project file. The data is stored in the form of objects. Within the project, the objects are arranged in a tree structure (project hierarchy).

The project hierarchy is based on the devices and stations along with the configuration data and programs belonging to them.

#### Requirement

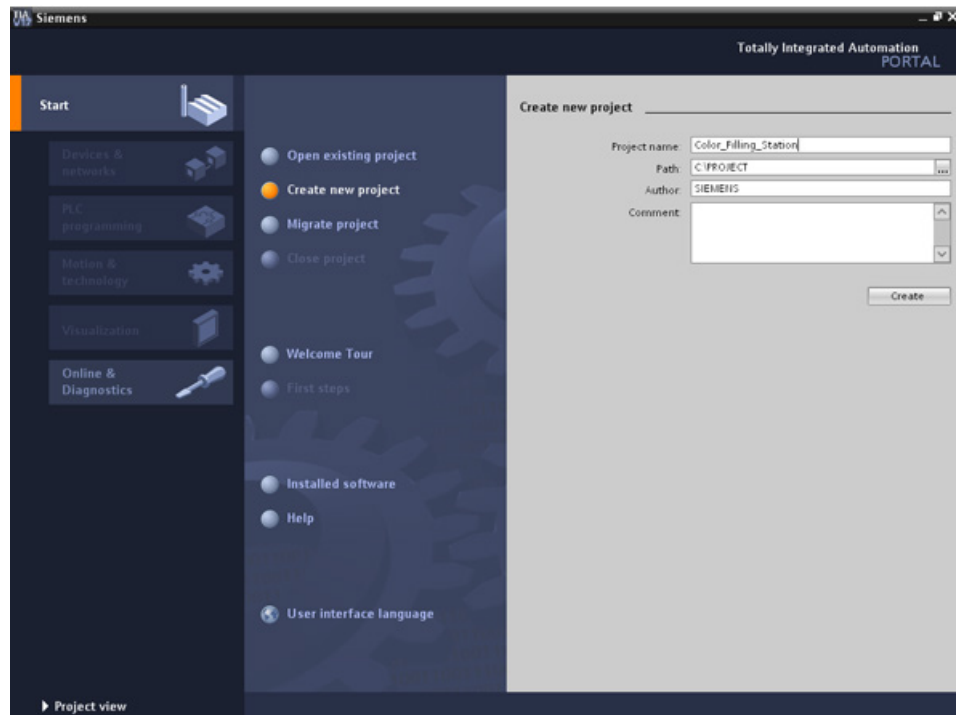
You need the following hardware and software equipment to create the project:

- Hardware:
  - The CPU 1511-1 PN that was installed and wired in the hardware section of the Getting Started.
  - An Ethernet connection to your programming device/PC.
- Software:  
The following software packages must be installed and executable on your programming device/PC:
  - SIMATIC STEP 7 Professional V17
  - SIMATIC WinCC Advanced V17 or SIMATIC WinCC Professional V17

## Creating a new project

To create a new project, follow these steps:

1. Click "Create new project".
2. Enter a name for your project.

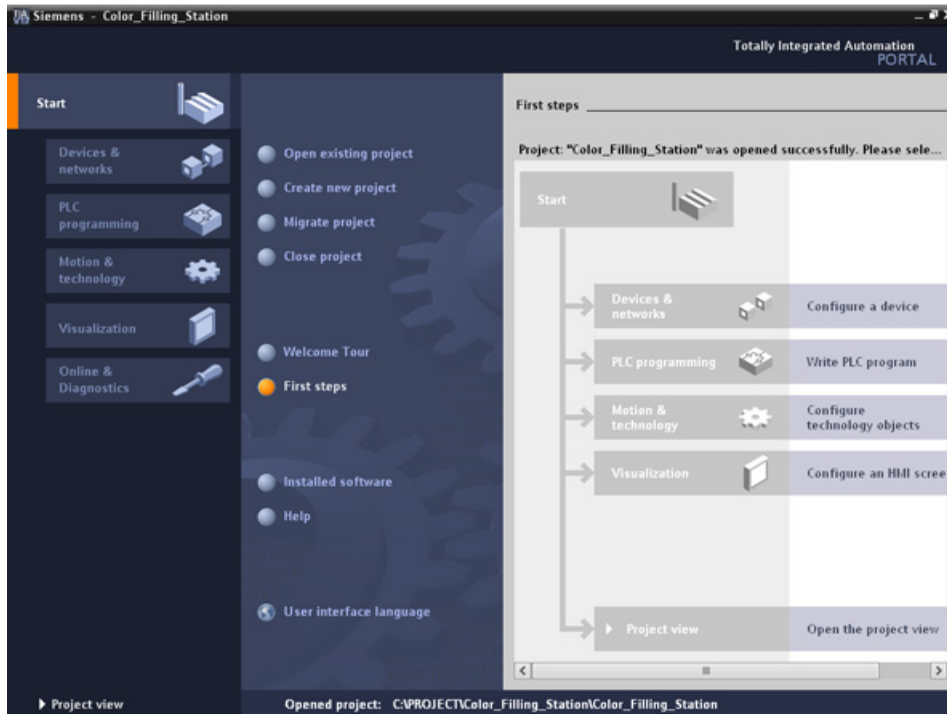


3. Click "Create" to create the new project.



## Result

The project has been created. All data, such as the hardware configuration, the CPU programming and the visualization in HMI, is saved in the project.



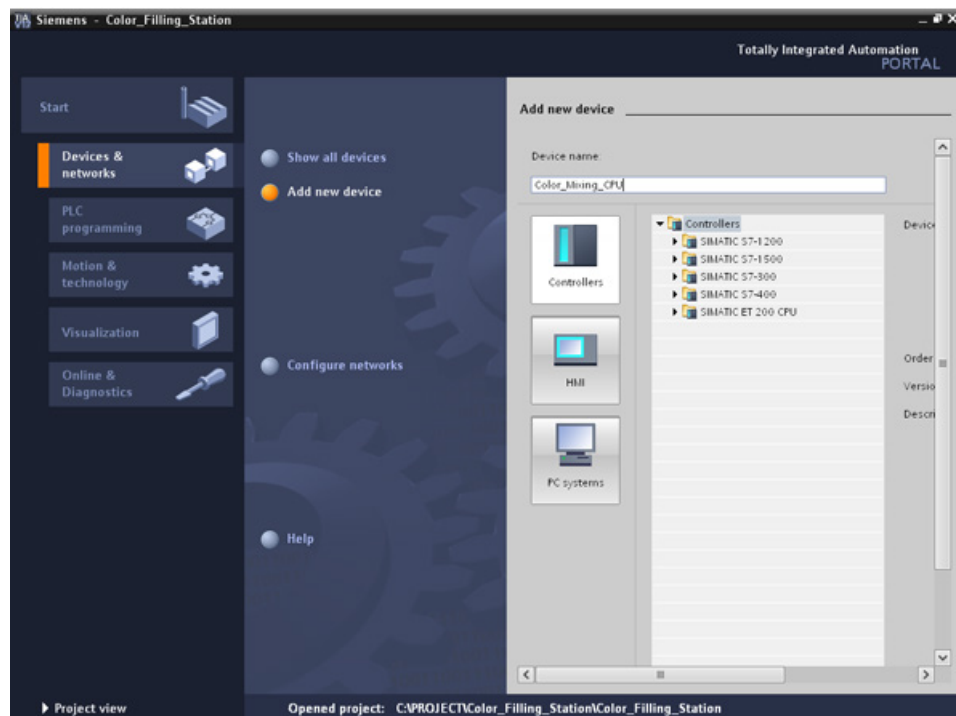
### 3.1.3 Creating an S7-1500 CPU

#### Introduction

In the following step, you will create an unspecified CPU. Unspecified CPUs are placeholders for specific CPUs from the hardware catalog which will be defined later.

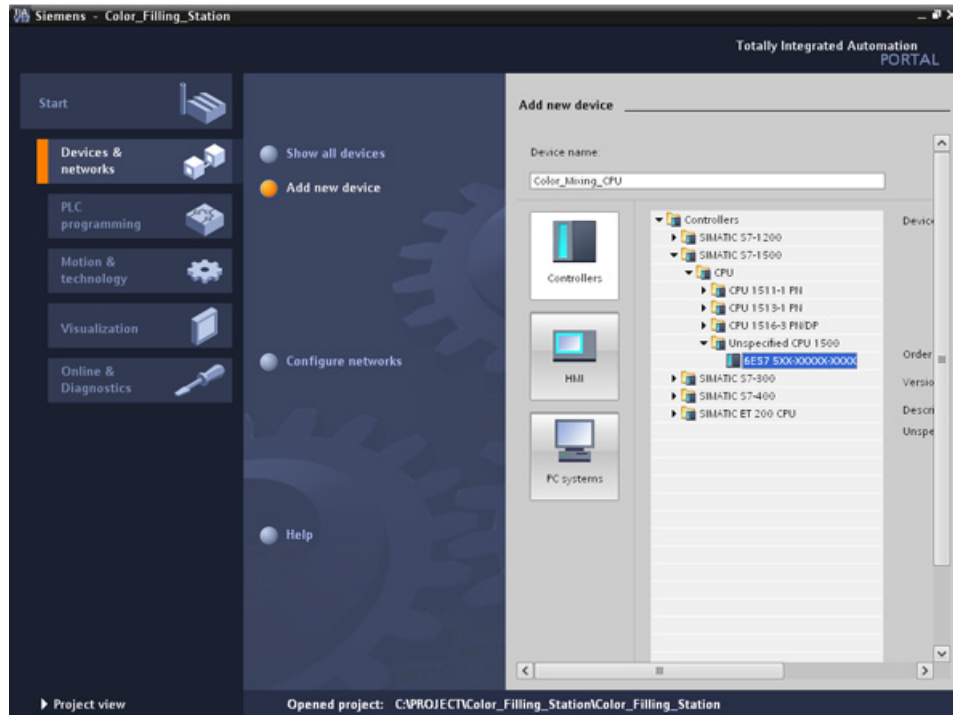
#### Procedure

1. Open the "Devices & Networks" portal.
2. Insert a new device.
3. Enter "Color\_Mixing\_CPU" as the name for the CPU.



4. Open the "SIMATIC S7-1500" folder.

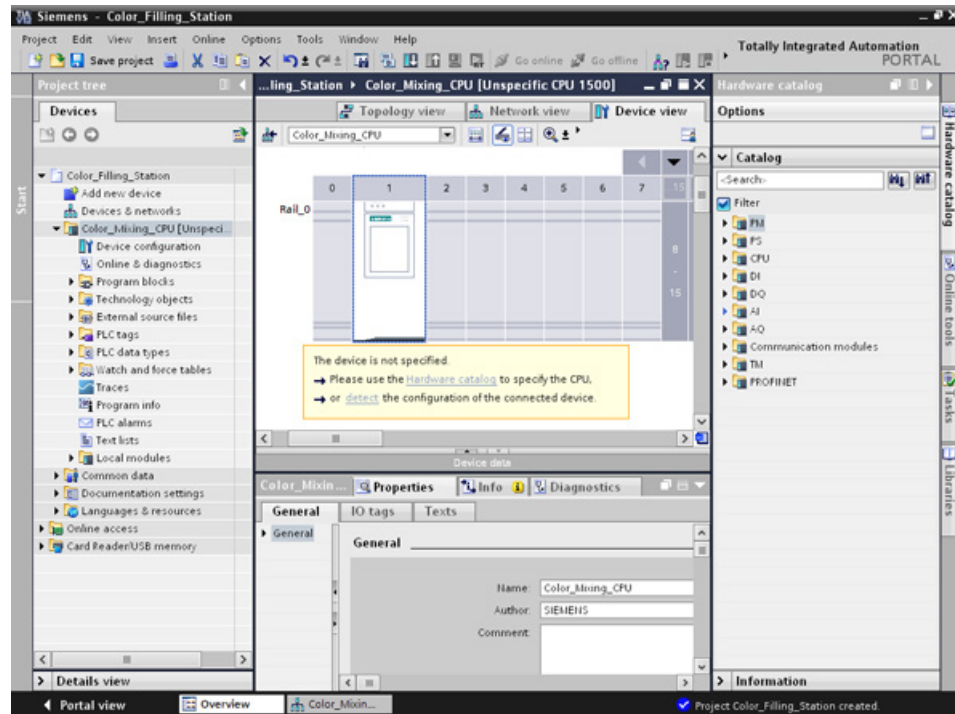
5. Select the CPU which has not yet been specified.



6. Create the CPU with a double-click.

## Result

The unspecified CPU is created in the project file. Contents of the user program can already be created at this point for this CPU.



### 3.1.4 Running the hardware detection

#### Introduction

In the following section, you will use the hardware detection function to read the CPU type. Run an LED flashing test during hardware detection. The LED flashing test activates the LEDs on a detected device. You may also use this function to verify that the correct device was selected in a hardware configuration consisting of several devices.

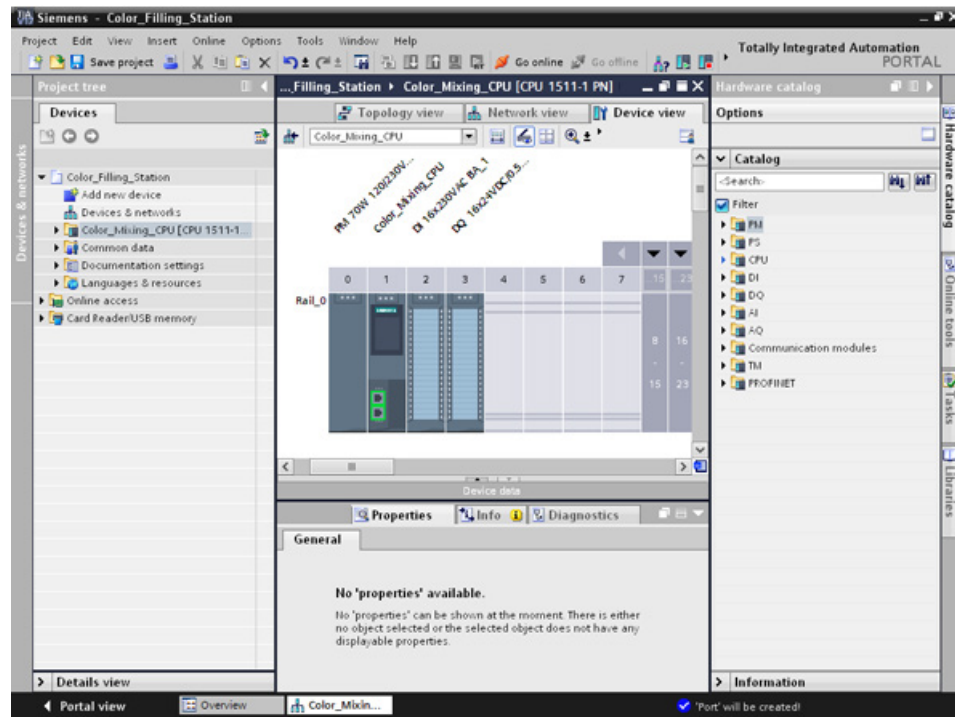
#### Procedure

1. Select the unspecified CPU in the project tree.
2. Select the "Hardware detection" function from the "Online" menu.  
Option 2: Click on the yellow framed alarm in the device view.
3. Select the "PN/IE" entry as the type of PG/PC interface.
4. Select the PG/PC interface.
5. Click the "Show all compatible devices" option.
6. Select the CPU from the compatible devices in the subnet.
7. Select the "Flash LED" check box to run a flashing test.
8. Click "Detect" to replace the unspecified CPU with the necessary CPU type.

## Result

The CPU type is read out. The correct device name is appended in brackets to your CPU name in the project tree.

The CPU and modules used are displayed in the hardware configuration.



### 3.1.5 Creating ET 200 interface modules

#### Introduction

In the following section, you will create two distributed I/O systems in the hardware configuration:

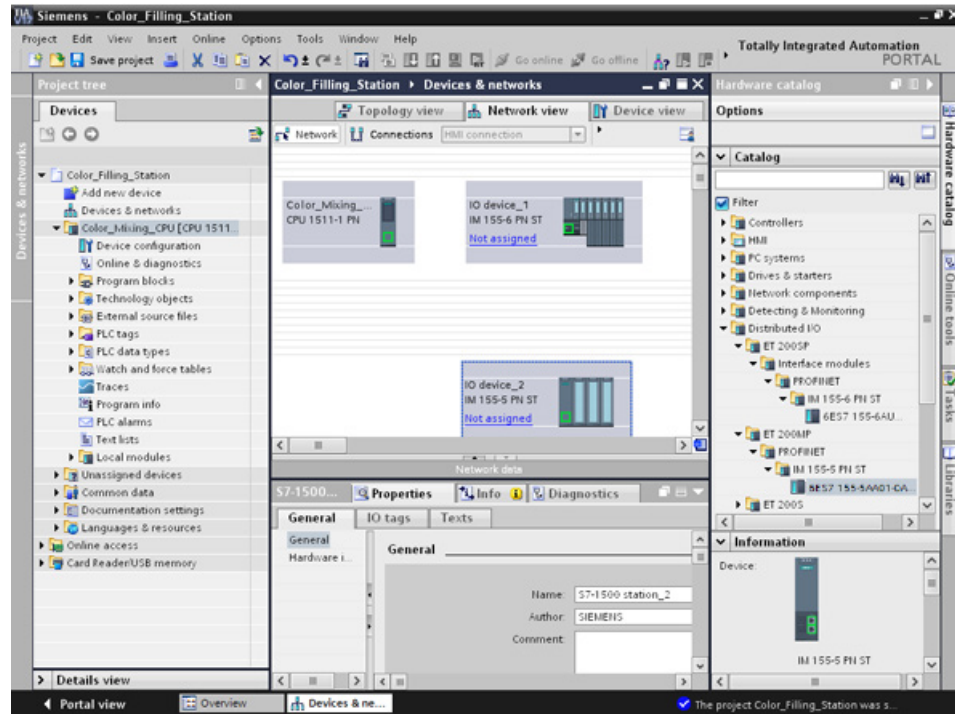
- An ET 200SP distributed I/O system, which basically consists of the following components:
  - An interface module for communication with the CPU.
  - Up to 32 modules that can be inserted in any combination.
  - A server module that completes the configuration.
- An ET 200MP distributed I/O system, which consists of the following components:
  - The interface module for communication with the CPU.
  - Up to 30 modules, each one providing up to 32 channels.

#### Procedure

1. Open the "Hardware catalog".
2. Change to the "Network view".
3. Open the "Distributed I/O" and "ET 200SP" folders.
4. Open the "IM 155-6 PN ST" folder.
5. Drag-and-drop the "6ES7155-6AU01-0BN0" interface module to the network view.
6. Open the "ET 200MP" folder.
7. Open the "IM 155-5 PN ST" folder.
8. Drag-and-drop the "6ES7155-5AA01-0AB0" interface module to the network view.

## Result

The I/O systems have been created in the hardware configuration, but not yet assigned to the CPU 1511-1 PN. They are both displayed under "Unassigned devices" in the project view.



## Additional information

The SIMATIC ET 200 product family offers different scalable I/O systems to suit your specific application.

You will find more information about the SIMATIC ET 200 distributed I/O on the Internet at (<https://new.siemens.com/global/en/products/automation/systems/industrial/io-systems.html>).



### 3.1.6 Networking ET 200 interface modules

#### Introduction

In the following section, you will create a PROFINET I/O system.

A PROFINET I/O system consists of the PROFINET IO controller and its assigned PROFINET IO devices:

- The CPU 1511-1 PN you already created is used as PROFINET IO controller.
- The two distributed I/O systems are used as PROFINET IO devices.

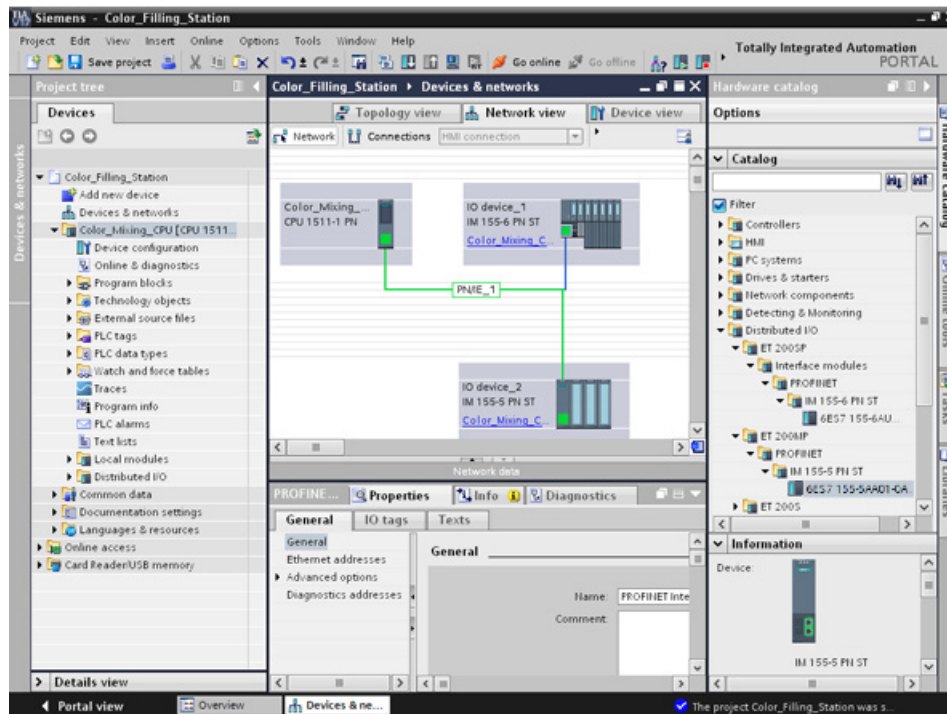
#### Procedure

1. Drag-and-drop a connection from the interface of the IM 155-5 PN ST interface module to the CPU interface.
2. Create a second connection between the IM 155-6 PN ST interface module and the CPU.

#### Result

The interface modules are assigned to the CPU as IO devices. The two distributed I/O systems are displayed in the project navigation in the "Distributed I/O" folder under the CPU.

A PROFINET I/O system was created automatically in the networking process and its properties are displayed in the network view.



### 3.1.7 Creating input and output modules and a server module for ET 200SP

#### Introduction

In the following section, you will create input and output modules for the ET 200SP.

#### Note

You need the server module to operate the input and output modules. These modules will fail if the server module is missing.

#### Maximum configuration per potential group

The number of I/O modules that can be used per potential group depends on the following factors:

1. Total power requirement of all I/O modules operated on this potential group
2. Total power requirement of all loads connected externally to this potential group

The sum of the total current calculated according to 1 and 2 must not exceed the current carrying capacity of the BaseUnit that is used and the load current supply.

Set the "Potential group" parameter for a module as follows:

Parameters	Value range	Usage
Potential group	Use potential group of the left module (default setting)	if the total power consumption of all modules from the left + power consumption of the module is less than the current carrying capacity of the BaseUnit
	Enable new potential group	if the total power consumption of all modules from the left + power consumption of the module is greater than the current carrying capacity of the BaseUnit

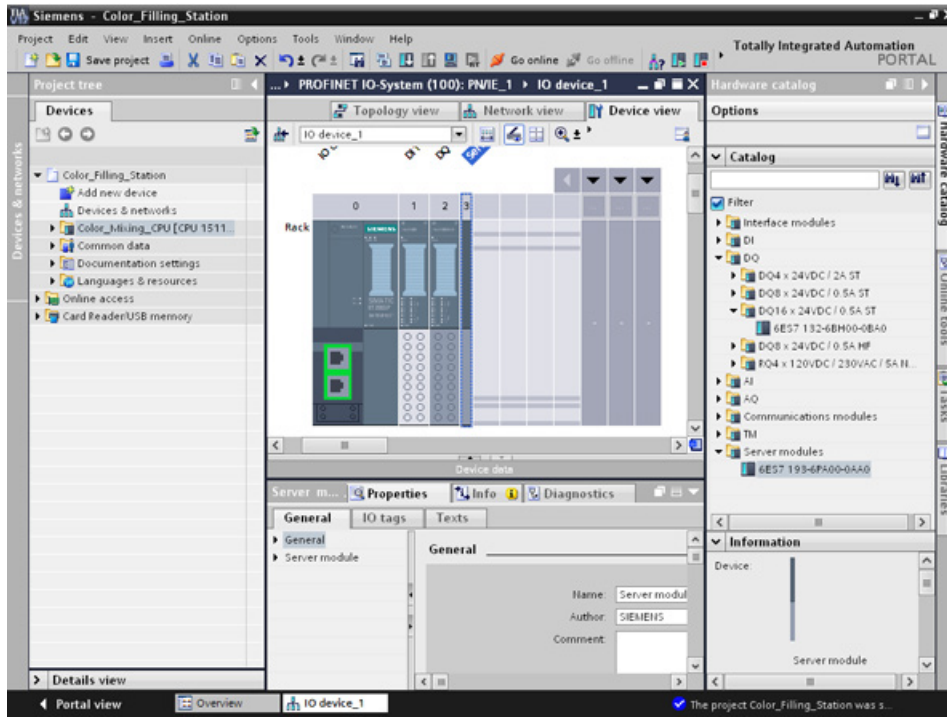
You can find additional information on potential groups in the module manuals such as SIMATIC ET 200SP DI 8x24VDC HF digital input module (<https://support.industry.siemens.com/cs/de/en/view/66912542>).

#### Procedure

1. Open the device view of ET 200SP.
2. Open the "DI" and "DI 16x24VDC ST" folders in the hardware catalog.
3. Drag-and-drop the input module "6ES7131-6BH01-0BA0" to slot 1 of the mounting rail.
4. Open the "DQ" and "DQ 16x24V DC/0,5A ST" folders.
5. Drag-and-drop the output module "6ES7132-6BH01-0BA0" to slot 2 of the mounting rail.
6. Open the "Server modules" folder.
7. Drag-and-drop the server module "6ES7 193-6PA00-0AA0" to slot 3 of the rail.

## Result

You have created the input and output modules and the server module.



### 3.1.8 Creating input and output modules for ET 200MP

#### Introduction

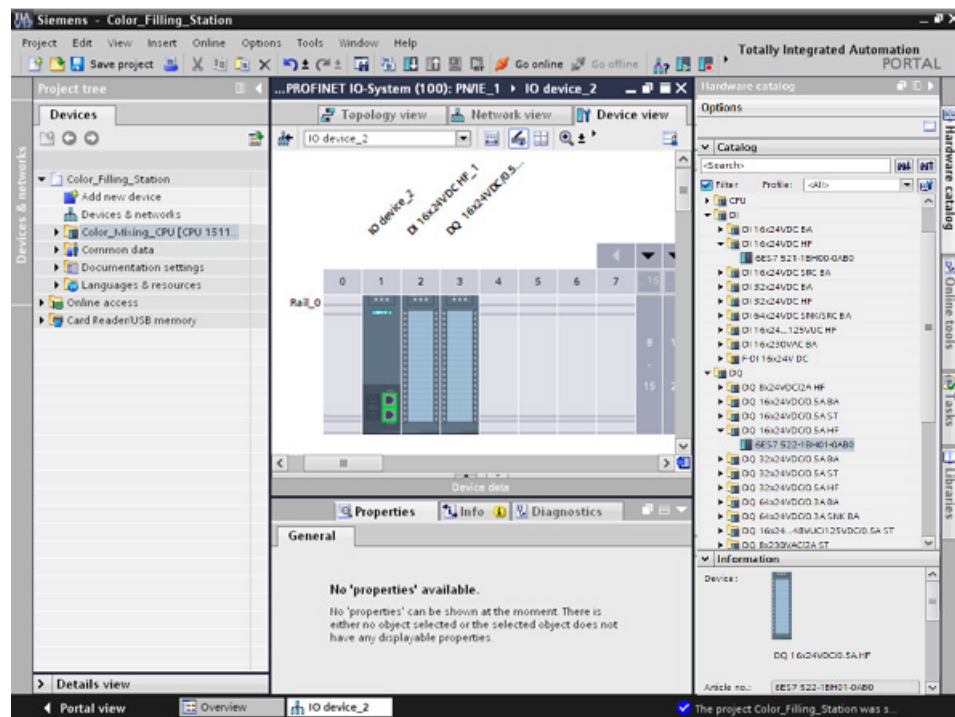
In the following section, you will create the input and output modules for ET 200MP.

#### Procedure

1. Open the device view of ET 200MP.
2. Open the "DI" and "DI 16x24VDC HF" folders in the hardware catalog.
3. Drag-and-drop input module "6ES7 521-1BH00-0AB0" to slot 2 of the rail.
4. Open the "DQ" and "DQ 16x24VDC/0.5A HF" folders.
5. Drag-and-drop the output module "6ES7522-1BH01-0AB0" to slot 3 of the mounting rail.

#### Result

You have created the input and output modules.



### 3.1.9 Assigning names for ET 200

#### Introduction

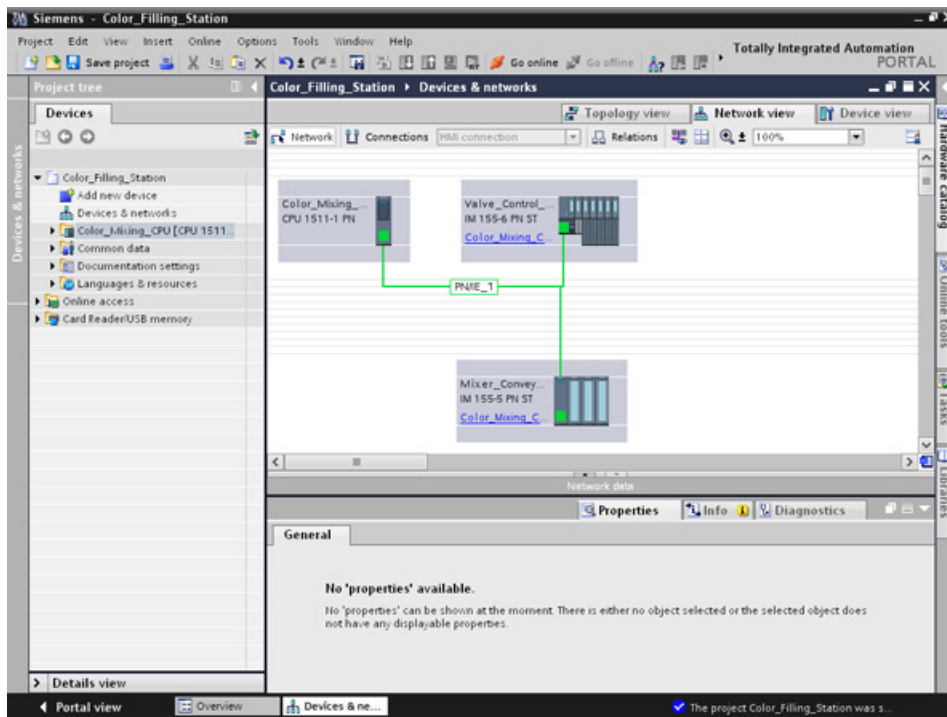
In the following section, you will assign project-specific names to the distributed I/O.

#### Procedure

1. Select ET 200SP.
2. Under **Properties > General** in the inspector window, enter the name "Valve\_Control\_Unit" in the "Name" field.
3. Select ET 200MP and enter the new name "Mixer\_Conveyor\_Control\_Unit".

#### Result

You have assigned the project-specific names.



## 3.2 Creating the program

### 3.2.1 Loading the block library

#### Introduction

In the following section, you will load the global library "ProgLib\_ColorFillingStation". This library contains the blocks and tag tables that you need for the example project. You can find the library as a ZIP file on the Internet (<https://support.industry.siemens.com/cs/ww/en/view/78027451>) in the section "Before you start ..." > "You can find the project for STEP 7 and WinCC here" > "Download". You need to unzip this library before you import it to your project.

#### Global libraries

Global libraries are used to store elements that you want to reuse in other projects. You must create global libraries explicitly.

The following libraries are provided in the standard package:

- "Buttons and Switches"

They offer a large selection of switches and buttons. The folders organize switches and buttons into categories. You can find the "System diagnostics indicator" object in the "DiagnosticsButtons" folder, for example. You use the "System diagnostics indicator" object for system diagnostics in your plant.

- "Monitoring and Control objects"

This provides complex operator control and display objects in several designs as well as suitable control lights, buttons and switches.

---

#### Note

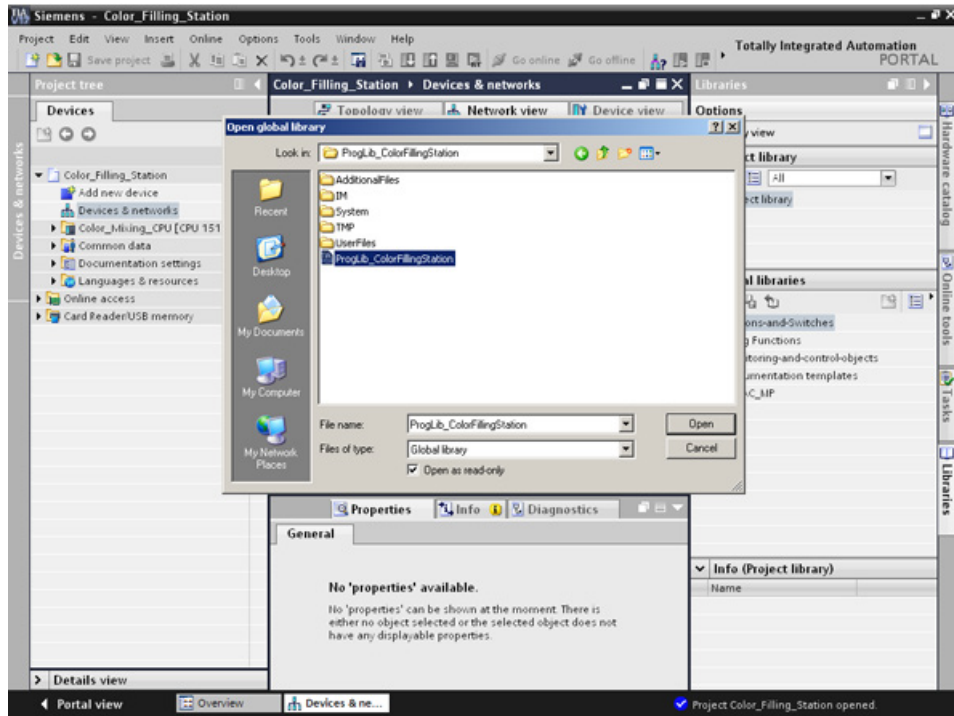
##### Library is write-protected

The "Open read-only" option is activated by default in the "Open global library" dialog. Click in the check box to open the library without write protection.

---

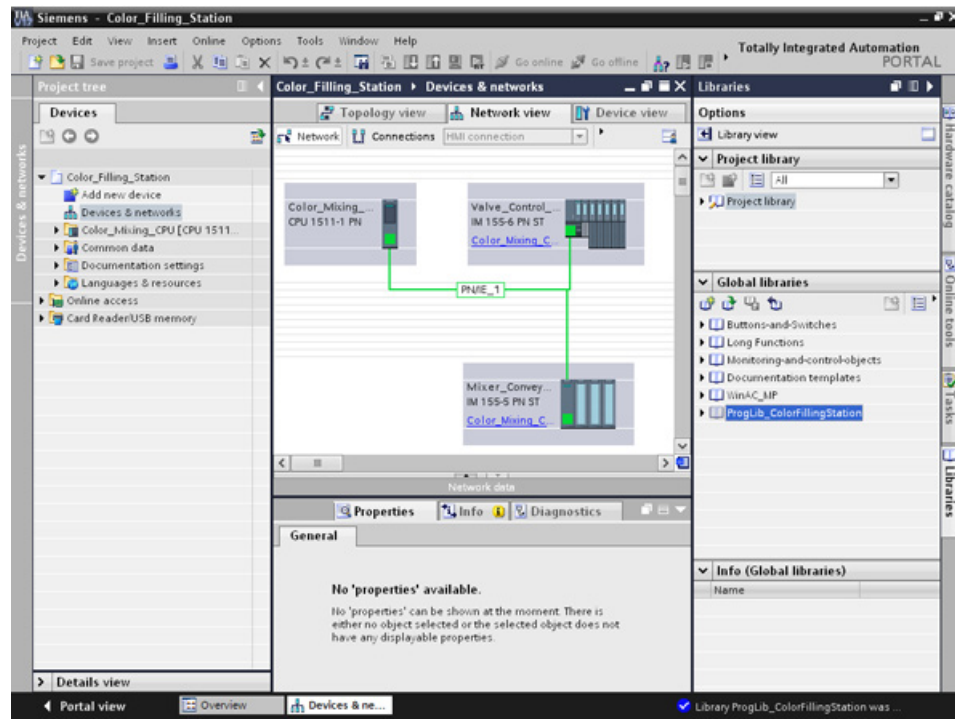
### Procedure

1. Click on the "Libraries" tab.
2. Click "Open global library".
3. Select the "ProgLib\_ColorFillingStation" file from the directory that contains the unzipped library folder and click "Open".



## Result

The "ProgLib\_ColorFillingStation" global library is open.



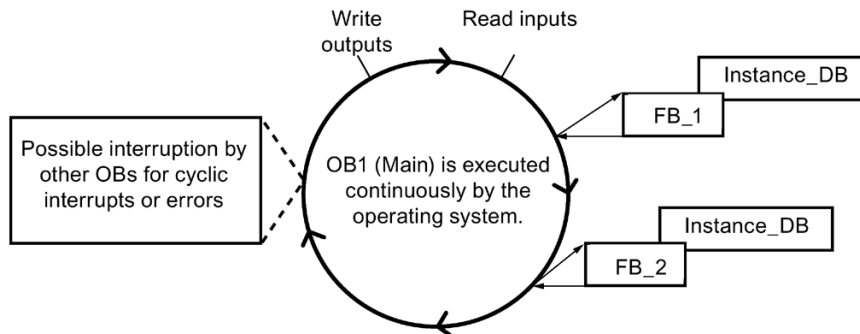


### 3.2.2 Deleting program block Main [OB1]

#### Introduction

In the following section, you will delete the automatically generated "Main [OB1]" program block from the project folder. A "Main [OB1]" program block is included in the program blocks of the example project.

Organization blocks (OBs) form the interface between the CPU operating system and the user program. These blocks are called by the operating system. At least one cycle OB must be available in an automation project.

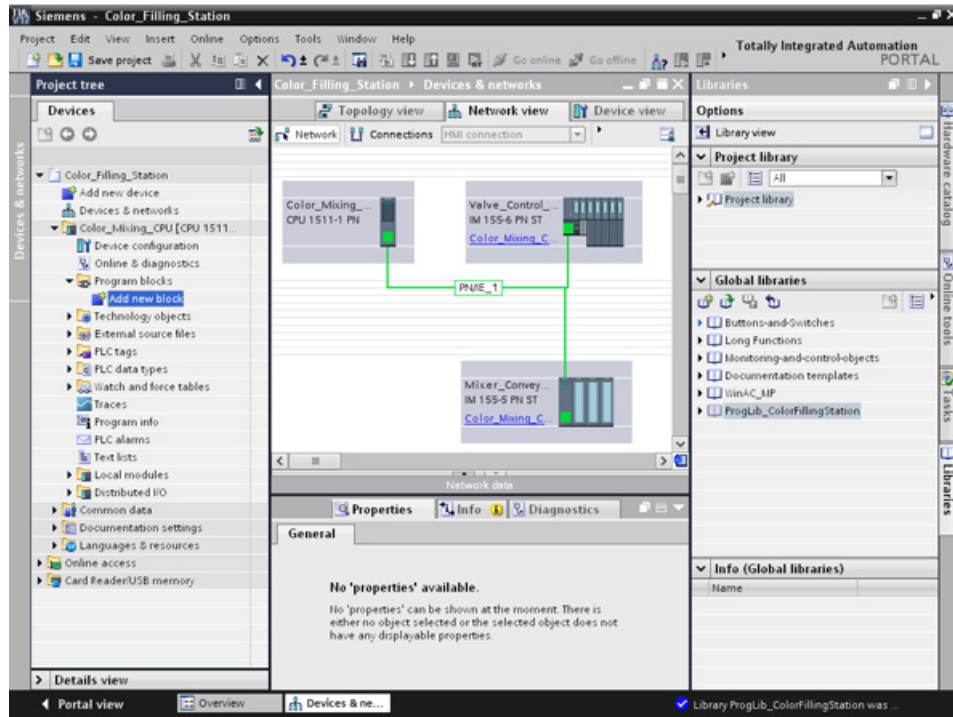


#### Procedure

1. Open the "Program blocks" folder in the project tree and then click the "Main [OB1]" program block.
2. Right-click to open the shortcut menu and then click "Delete".
3. Click "Yes" to confirm deletion of the block.

## Result

The automatically generated "Main [OB1]" program block is deleted.



### 3.2.3 Copying program blocks

#### Introduction

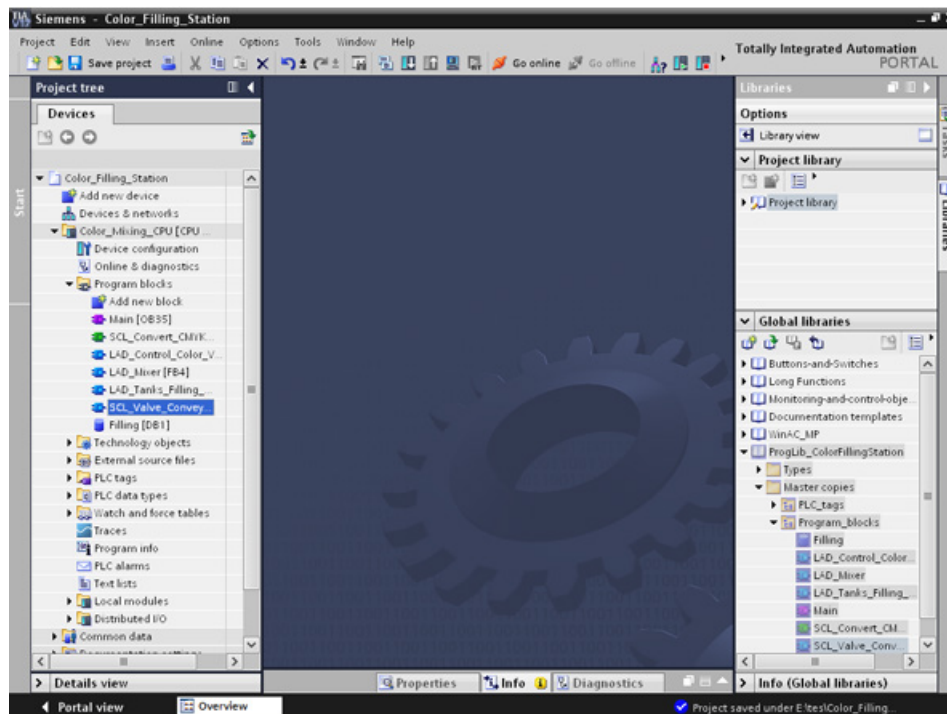
In the following section, you will insert the program blocks from the "ProgLib\_ColorFillingStation" global library into your project.

#### Procedure

1. Click on global library "ProgLib\_ColorFillingStation".
2. Click the "Master copies" folder and then on "Programm\_blocks".
3. Drag-and-drop the program block to be imported from the global library to the "Program blocks" folder.
4. Proceed as described in steps 2 and 3 for the other blocks.

#### Result

The program blocks are inserted in the project folder of the same name.



## 3.2.4 Cyclic interrupt OB

### 3.2.4.1 Cyclic interrupt OB – Cycle time and phase

#### Cycle time and phase offset can be changed

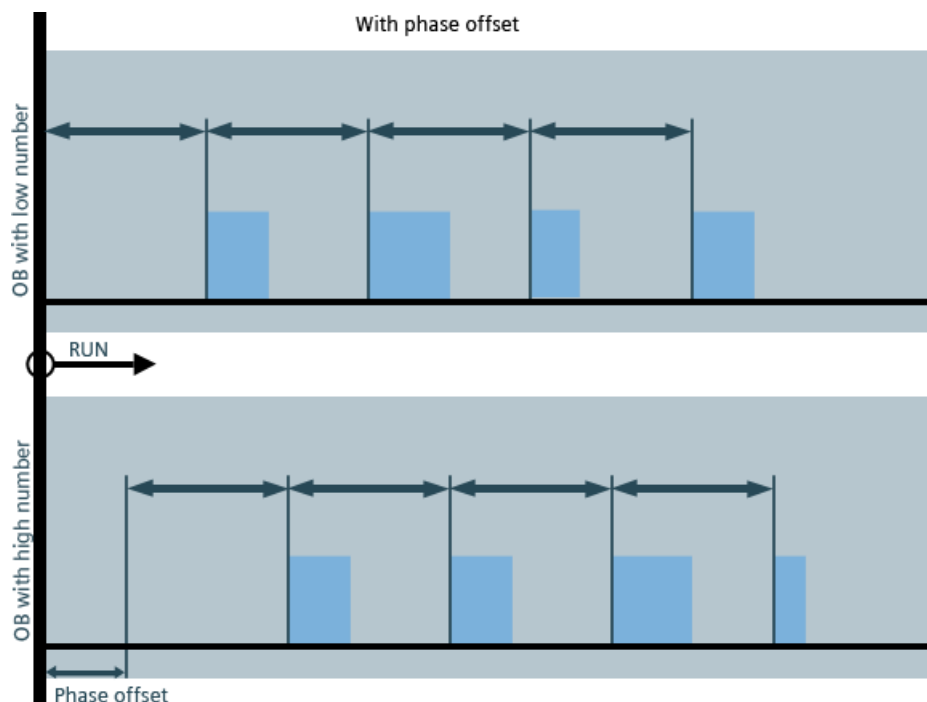
Main [OB35] is located below the program blocks inserted into the project. Main [OB35] is a cyclic interrupt organization block (cyclic interrupt OB). Cyclic interrupt OBs serve to start programs in periodic time intervals independently of the cyclic program execution. The start times of a cyclic interrupt OB are specified using the cycle time and the phase offset.

#### Cycle time

The cycle time determines the interval at which an OB is called. The cyclic interrupt OB has a cycle time of 100000  $\mu$ s by default.

#### Phase offset

The phase offset is used to increase the accuracy of the processing intervals of cyclic interrupt programs. If an OB has the same or a common multiple clock pulse of another OB, both can be operated at a precise interval by a phase offset.



### 3.2.4.2 Changing the cycle time

#### Introduction

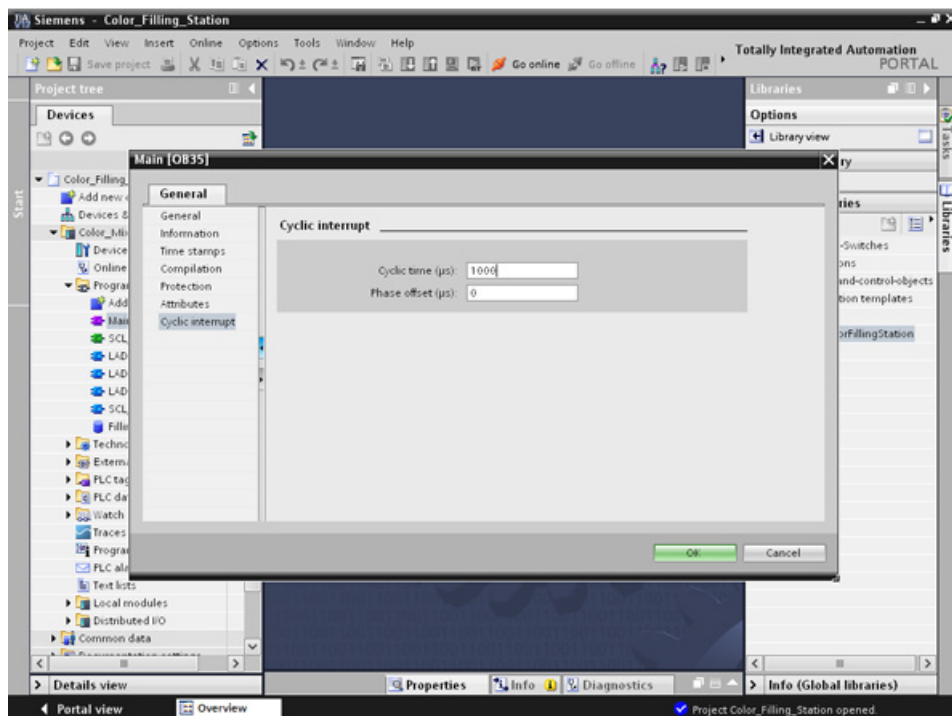
In the following section, you will change the cycle time for the "Main" program block.

#### Requirement

- The program block "Main" [OB35] is contained in the library
- The FB/FC calls exist

#### Procedure

1. Open the properties of the "Main" program block.
2. Select the "Cyclic interrupt" option under "General".
3. Enter the new value for the "Cycle time" and click "OK".



#### Result

The cycle time is changed.

## 3.2.5 Copying tag tables

### Introduction

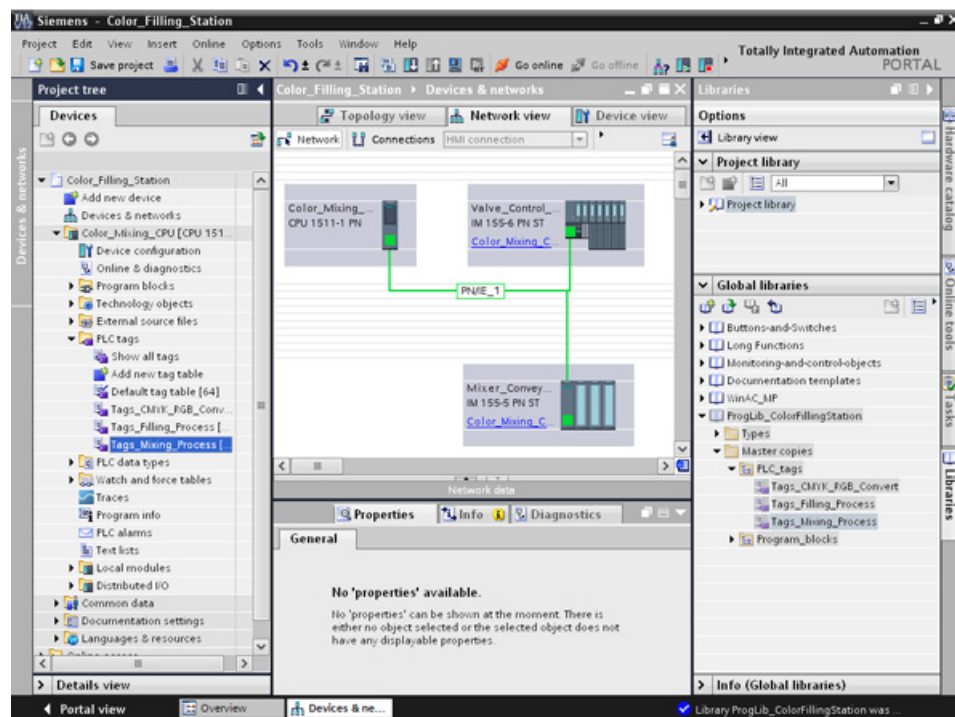
In the following section, you will insert the tag tables from the "ProgLib\_ColorFillingStation" global library into your project.

### Procedure

1. Open the "PLC tags" folder in the project navigation.
2. Open the "PLC\_tags" folder.
3. Drag-and-drop the tag table to be imported from the global library to the "PLC tags" folder.
4. Proceed as described in step 3 for the other tag tables.

### Result

The tag tables are inserted in the project folder of the same name.



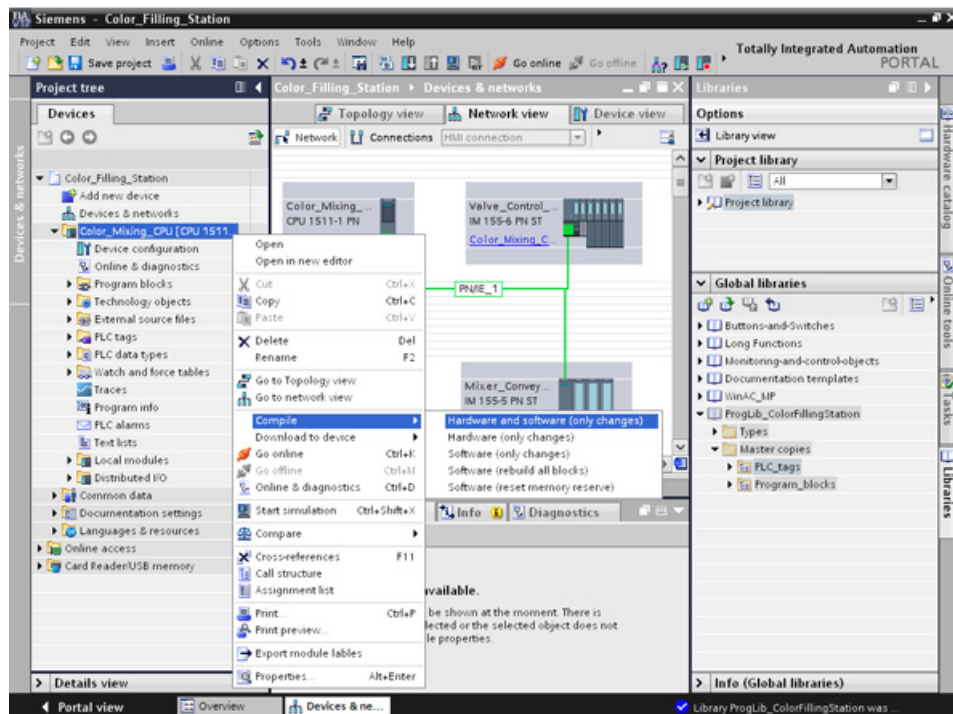
### 3.2.6 Compiling a project

#### Introduction

In the next section, you will compile the "Color\_Filling\_Station" project.

#### Procedure

1. Select the "Color\_Mixing\_CPU" CPU in the project tree.
2. Right-click to open the shortcut menu and then select "Compile" > "Hardware and software (only changes)".







### 3.2.7 Load project into the CPU

#### Introduction

In the next section, you will download the "Color\_Filling\_Station" project to the CPU.

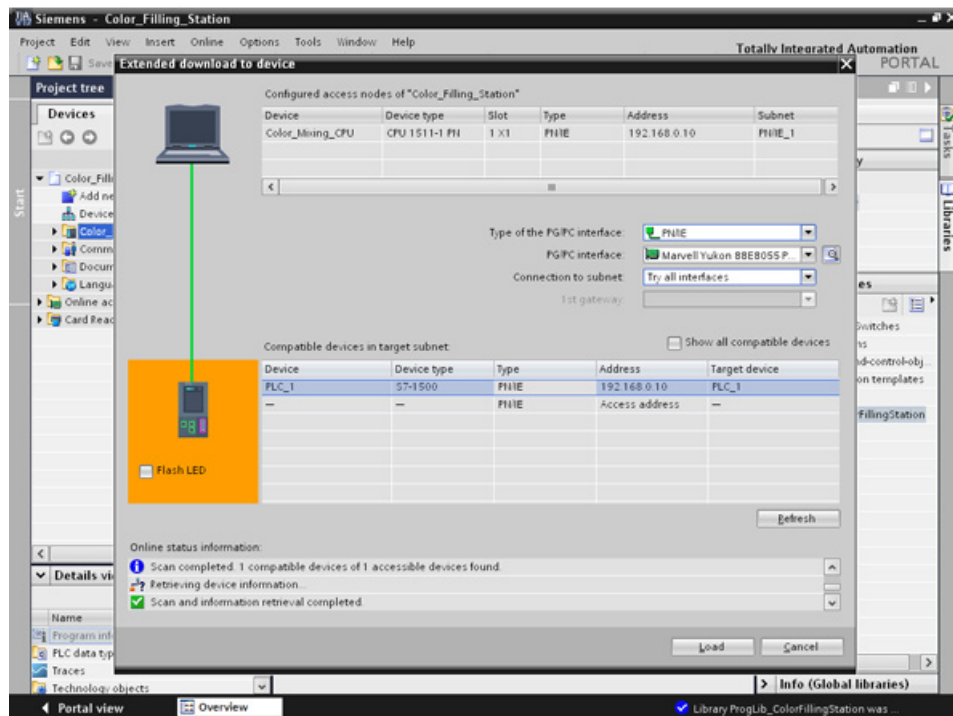
#### Note

##### Displaying all compatible devices

If the desired CPU is not displayed after you have made the settings in the "Extended Download" dialog, click the option "Show all compatible devices".

#### Procedure

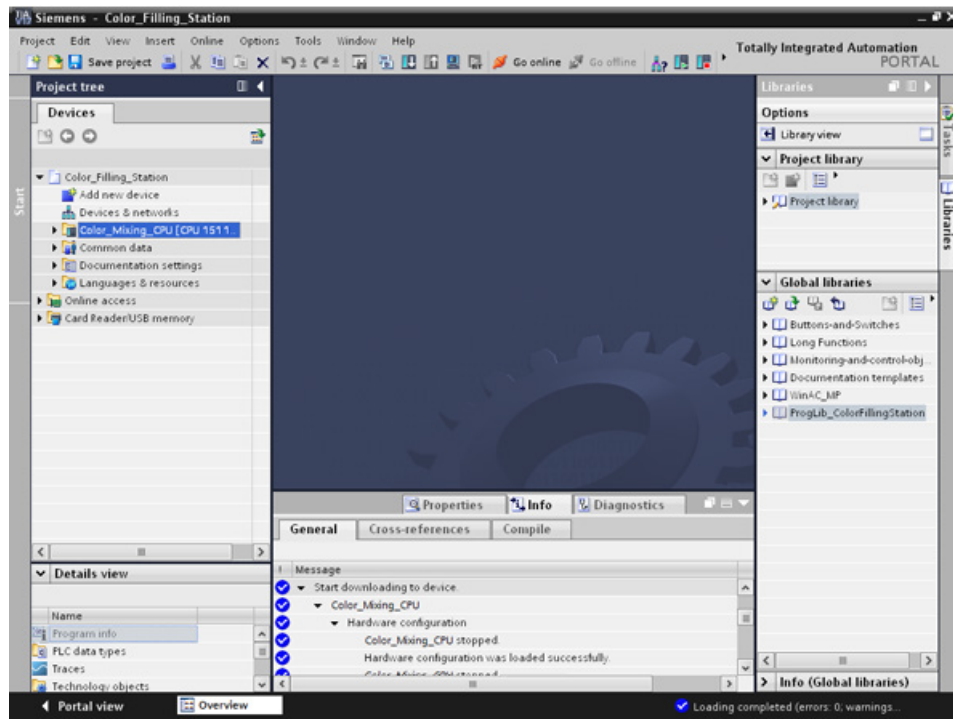
1. Open the CPU shortcut menu and select "Download to device" > "Hardware and software (only changes)".
2. From the drop-down lists, select the PG/PC interface type, the interface and the connection with the subnet.
3. Select the CPU from the compatible devices in the subnet and click "Load".



4. Confirm the two "Assign IP address" dialogs with "Yes" and "OK".
5. In the "Load preview" dialog, select the alternative entry for all entries set to "No action" in the drop-down list and confirm open options.
6. Click "Load".
7. Confirm the "Start all" option and click "Finish".

## Result

The project is downloaded to the CPU.



## 3.2.8 Optimized block access

### 3.2.8.1 Introduction

#### Operating principle

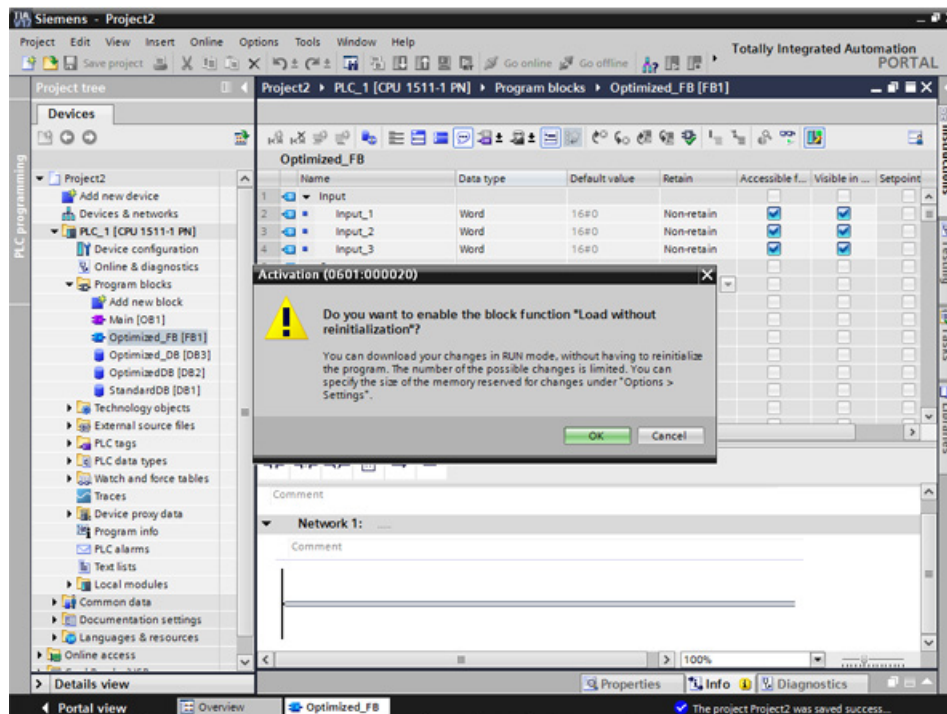
The "optimized data blocks" of the CPUs of the S7-1500 series are optimized for performance and are only programmed symbolically. By using the optimized data blocks, you make your program more efficient, because the declared tags are given symbolic names and no longer a fixed address.

You can create data blocks with any structure without paying attention to the physical arrangement of the individual tags. Quick access to the optimized data is always available because the data storage is optimized and managed by the system.

Changing data types increases the risk of error in the standard block. In the optimized block, changes lead to a reorganization of the data storage. Addressing remains unique.

To enable the subsequent editing of user programs that are already running in a CPU, the S7-1500 CPUs support the option of extending the interfaces of function or data blocks during runtime. You can download the modified blocks without setting the CPU to STOP and without affecting the actual values of tags already loaded.

In addition: You can define in the data block itself, which the values in the CPU are read-only for an HMI device ("Visible in HMI") or which can be written ("Accessible from HMI").



### 3.2.8.2 Expanding and reloading the optimized "Filling" data block

#### Introduction

In the following section, you will supplement the "Filling" data block with the date and time of the last filling and reload the data block. To do this, create a block for recording the date and time and enable the function "Download without reinitialization".

**Note:** The "Download without reinitialization" function protects the actual parameters of the data block from being overwritten during download to the CPU.

**Advantages of symbolic addressing:** The use of universally applied and meaningful symbols in the entire project makes the program code easier to read and understand. This gives you the following advantages:

- You do not have to write detailed comments.
- Data access is faster.
- No errors occur when accessing data.
- You no longer have to work with absolute addresses.
- The assignment of the symbol to the memory address is monitored by STEP 7, which means that all points of use are automatically updated when the name or the address of a tag changes.

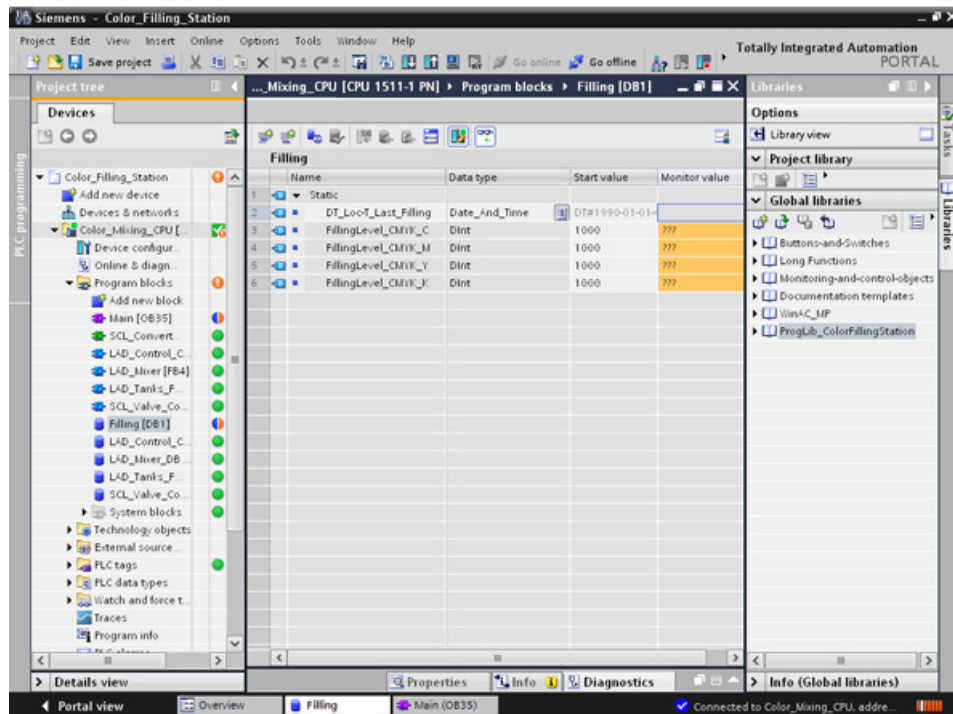
#### Requirement

- The library has been loaded
- The project has been compiled and loaded into the CPU

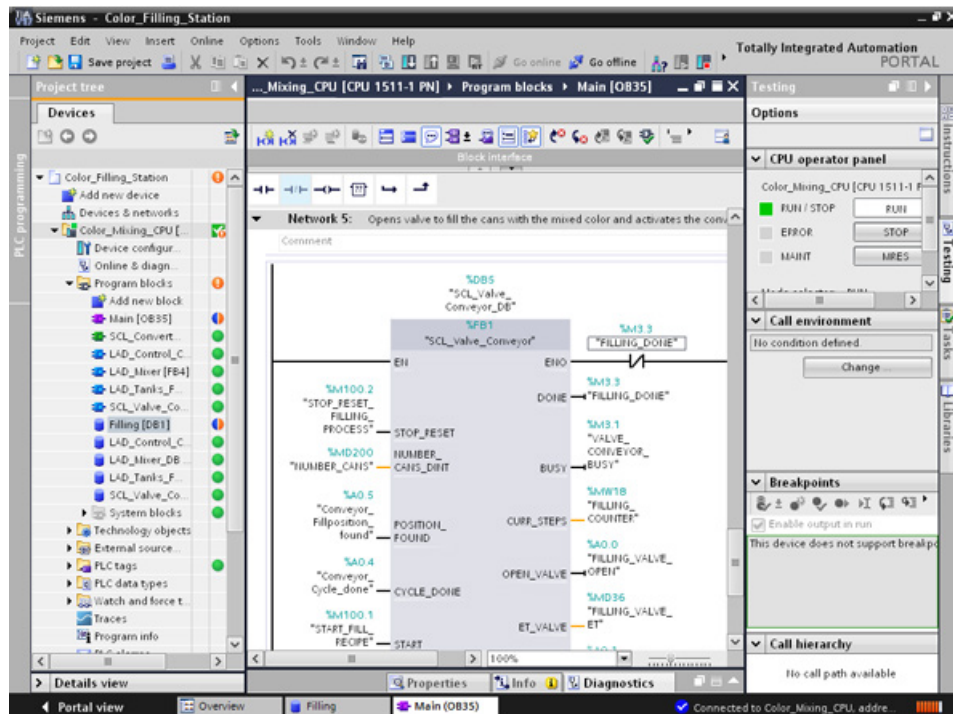
#### Procedure

1. Open the "Filling" data block and the "Main" program block.
2. Enable the "Monitoring on/off" function for the "Main" program block.
3. In the "Main" program block, open the shortcut menu of the "'FILLING' FillingLevel\_CMYK\_C" I/O in the 3 network with a right-click and select "Modify > Modify operand".
4. Enter a new value and click "OK".
5. Enable the "Download without reinitialization" function and the "Monitor all" function in the "Filling" data block.

6. Create a new parameter named "DT\_Loc-T\_Last\_Filling" and select "Date\_And\_Time" as the data type.

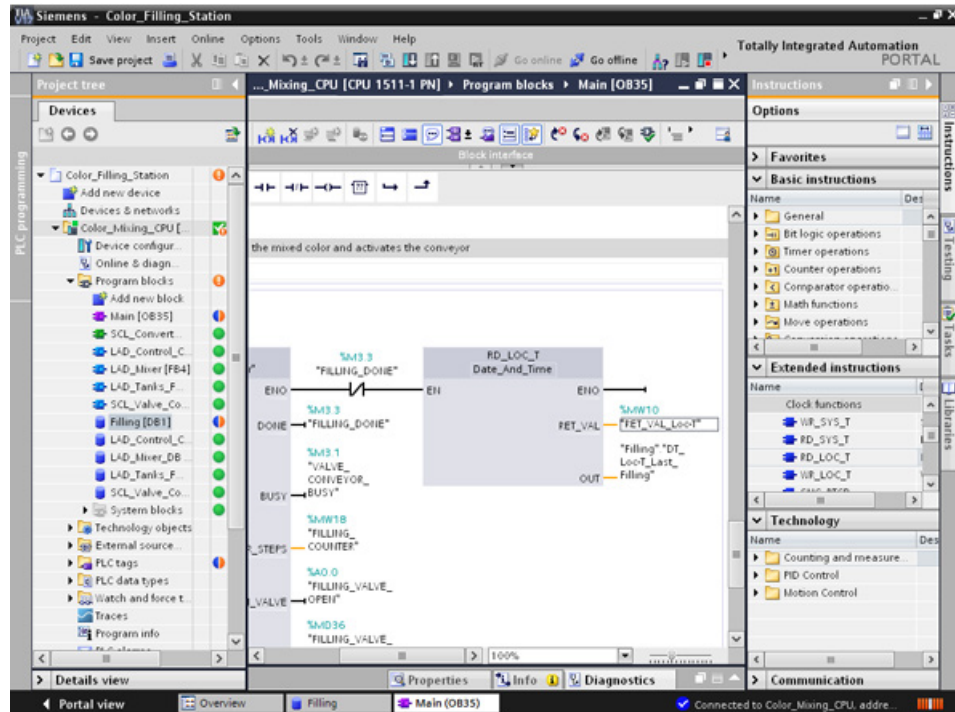


7. Insert a normally closed contact into the "Main" program block in the 5 network, and interconnect it with the "FILLING\_DONE" parameter.



8. Open the "Date & time" folder from the "Instructions" tab and insert the "RD\_Loc\_T" block in the "Main" program block.

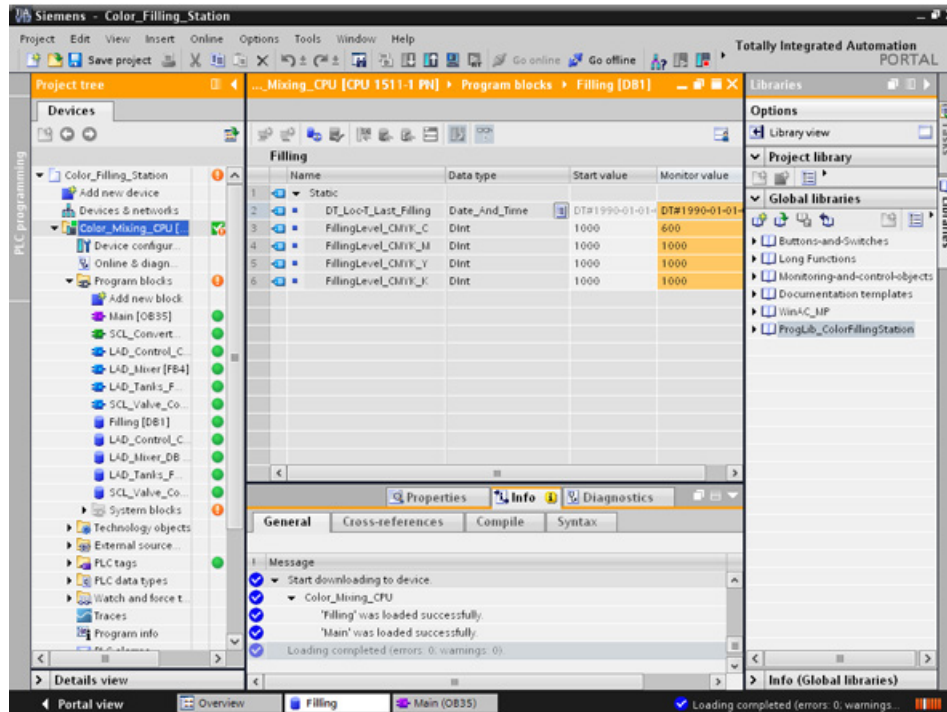
- Interconnect the "OUT" output with the "DT\_Loc-T\_Last\_Filling" parameter and the "RED\_VAL" output with the newly created "RED\_VAL\_Loc-T" parameter. Use the "LAD\_Tanks\_Filling\_Process" data block as the storage location for the "RED\_VAL\_Loc-T" parameter.



- Compile and download the project.

## Result

The date and time of the last filling are reloaded. The actual parameters of the "Filling" data block are not overwritten.





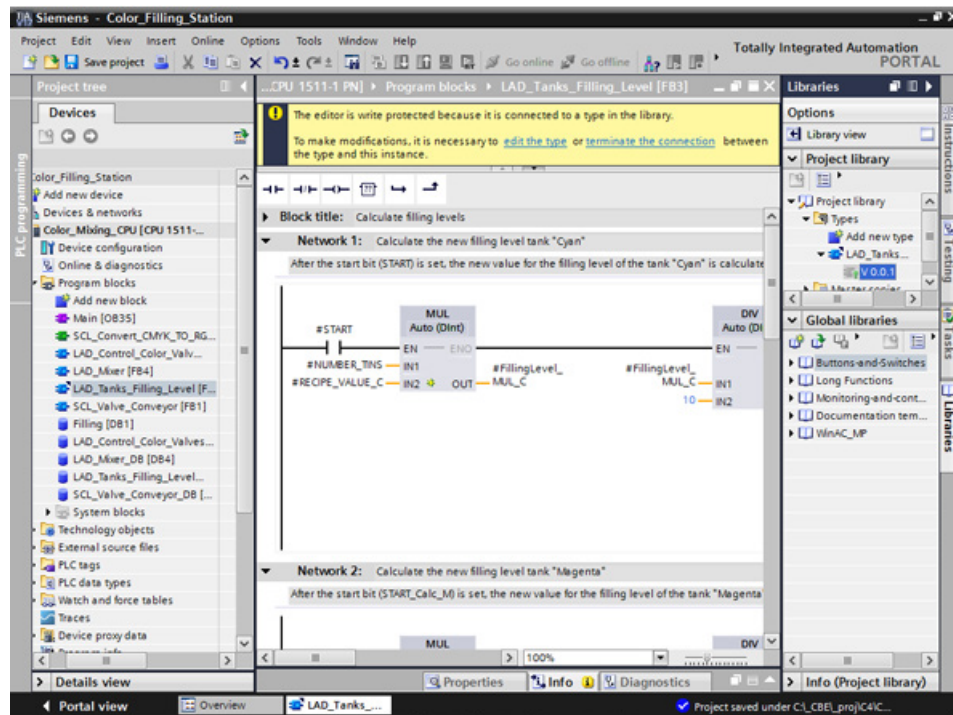
## 3.2.9 Versioning a block

### Introduction

The use of block types ensures a high degree of standardization in your projects. You can easily integrate function extensions to the block type into existing projects. Change tracking is ensured by versioning. In this example, you create a "LAD\_Tanks\_Filling" block as a type in the project library. As a function extension, replace the three instructions for the level calculation with CalculateBox, which performs all arithmetic functions. This optimization means that fewer temporary tags are required and that the switch between blocks with various programming languages is no longer necessary.

### Procedure

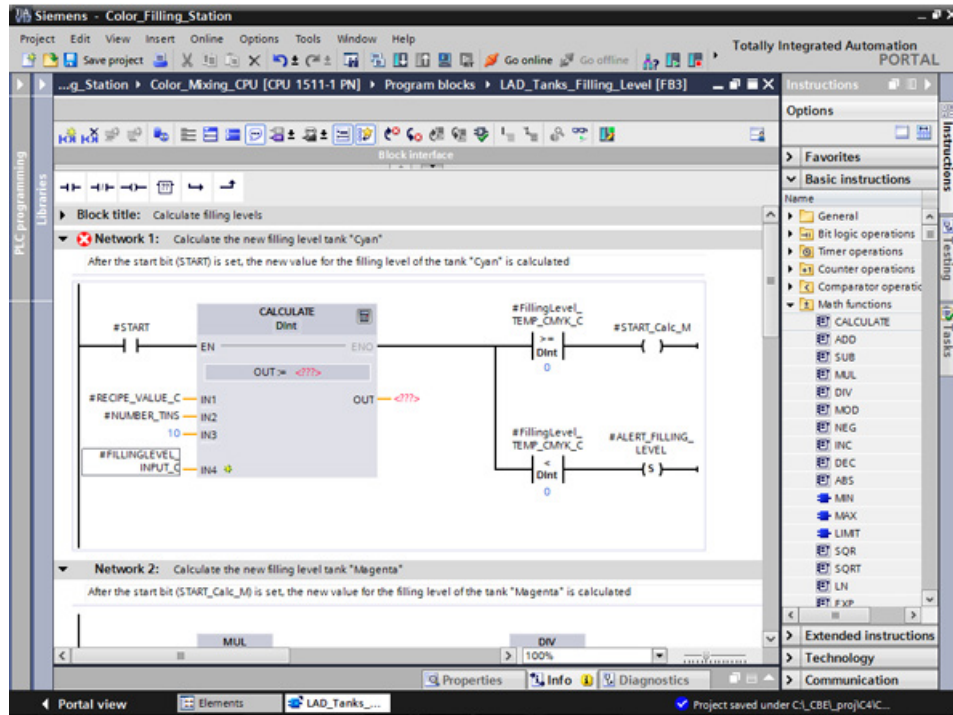
1. Compile the "LAD\_Tanks\_Filling" block and then insert it in the project library under "Types".
2. Create a new block version with "Edit type".



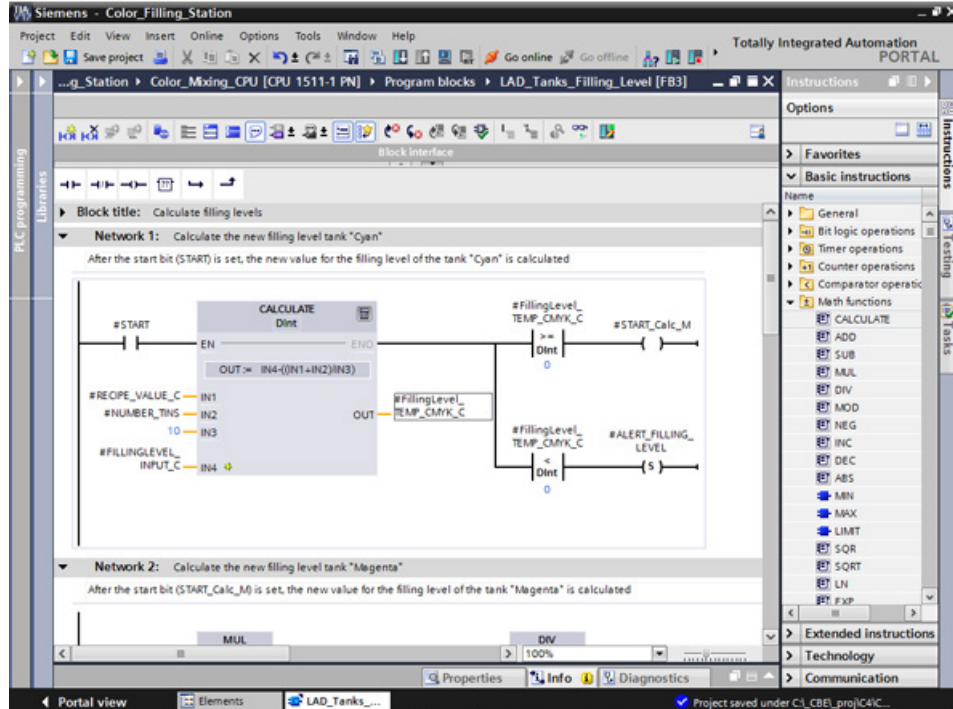
3. Insert the CALCULATE instruction: from the "Basic instructions > Mathematical functions" library.
4. Delete the MUL, DIV and SUB instructions from the block.



5. Insert two inputs into the CALCULATE instruction and interconnect the inputs.



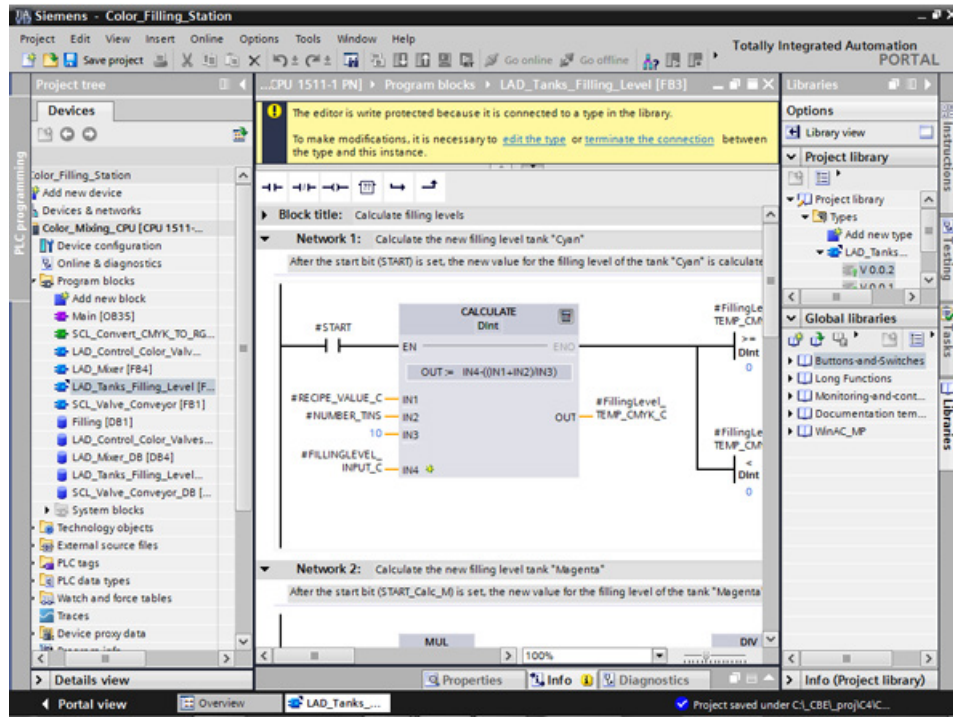
6. Define the calculation formula and then interconnect the output.



7. Release the block version.

## Result

The revised version of the block type is saved in the library with a new version number.



### 3.2.10 Setting retentivity

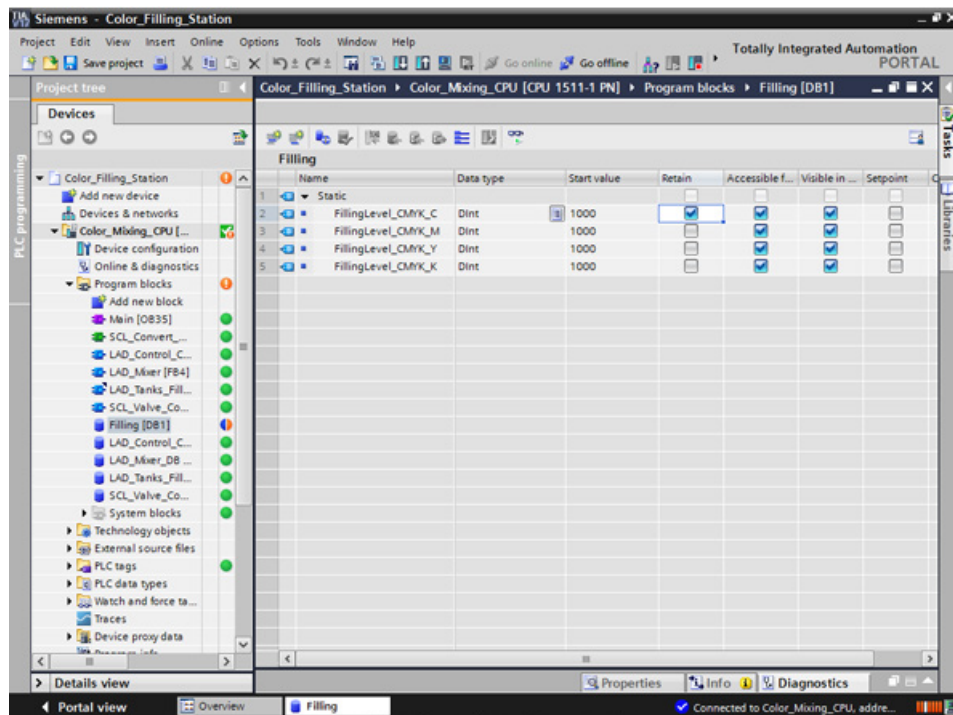
#### Introduction

All tags are initialized with their configured start values during CPU startup, for example, after a power failure. The most recent values the tags had immediately before the interruption are overwritten with the initial values. To prevent this, define the tag as retentive. Retentive tags retain their values even after a restart.

In this example, the levels of paint storage tanks are backed up in the retentive memory area of the CPU.

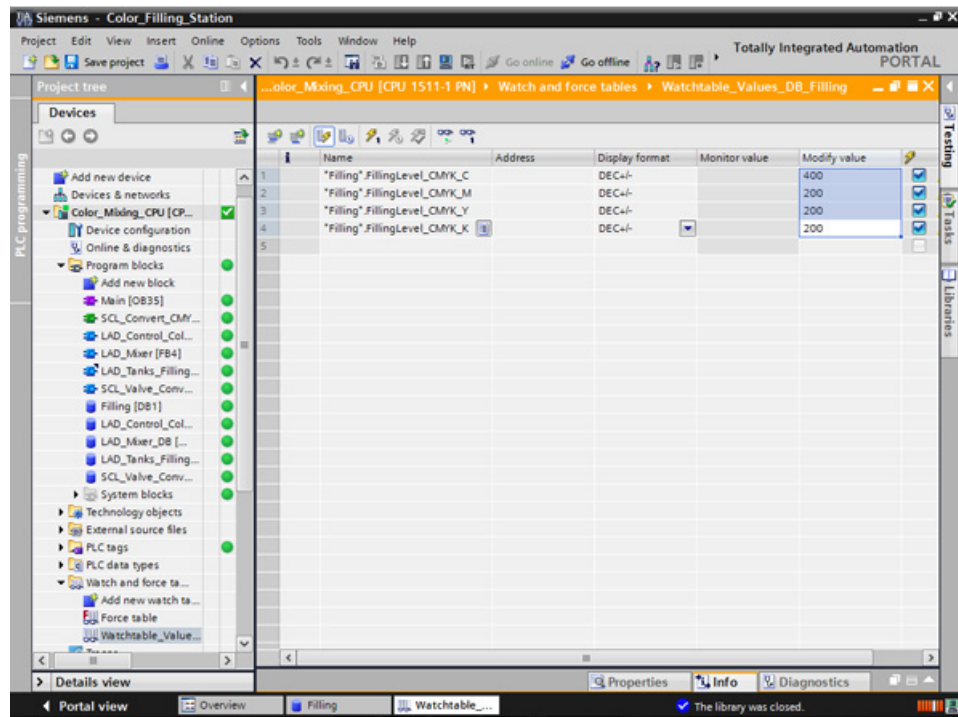
#### Procedure

1. Connect to the CPU online.
2. Enable the retentivity for the "Cyan" entry in the "Filling" data block.



3. Load the change to the CPU.
4. Drag the "Watchtable" object from the library into the project. This object contains the fill level tags included a control value.

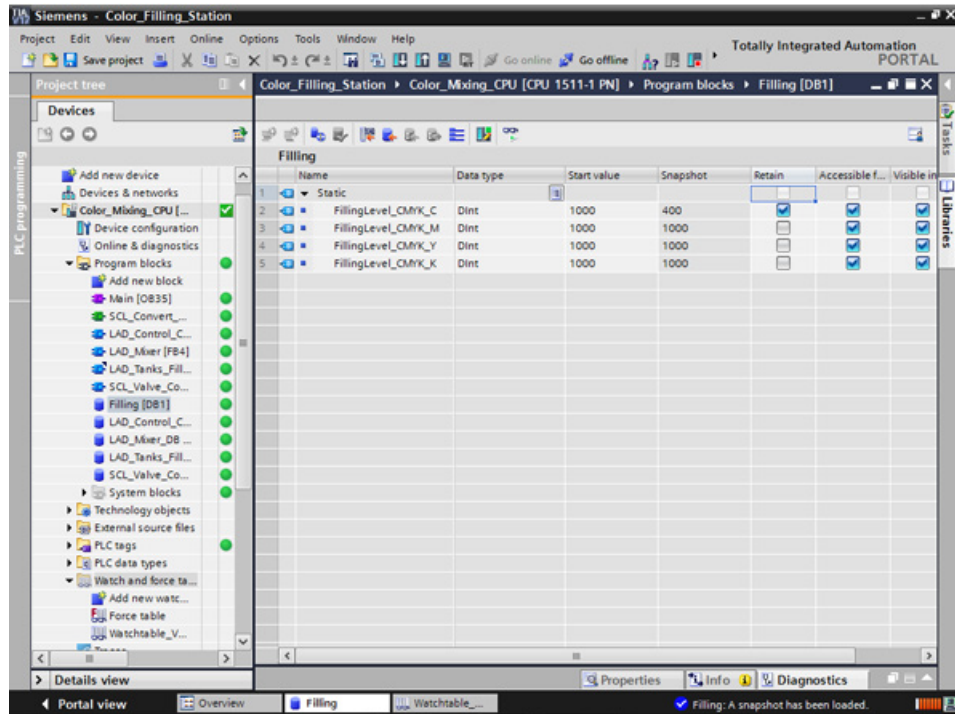
5. Transfer the control values to the CPU with "Modify now".



6. Close the online connection to the CPU. To simulate a power failure, disconnect the power supply to the CPU.
7. Reconnect the power supply and go online to the CPU. Enable "Monitor all" for the "Filling" DB.

## Result

The fill level for "Cyan" is read from the retentive memory area. All other fill levels are re-initialized with their start value.



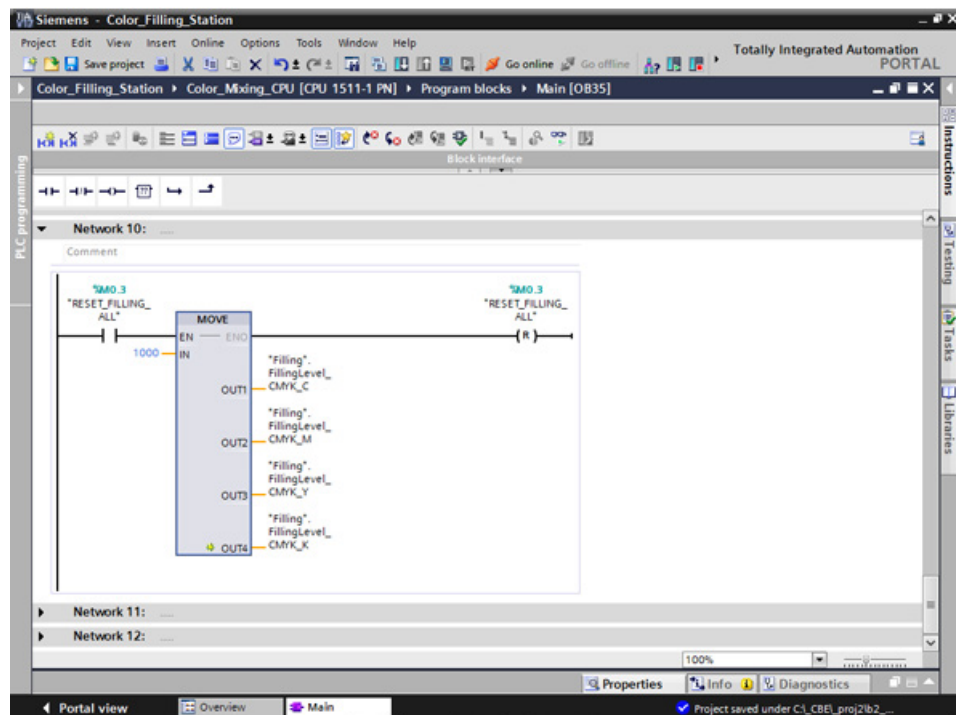
### 3.2.11 Activating the EN/ENO mechanism

#### Introduction

The EN/ENO mechanism in various instructions enables you to detect runtime errors and avoid a program crash. Newly inserted ENO instructions are disabled by default. You can then activate the ENO enable output. You can use this in a new network that has the fill level of all paint storage tanks reset to the start value (1000) at the same time.

#### Procedure

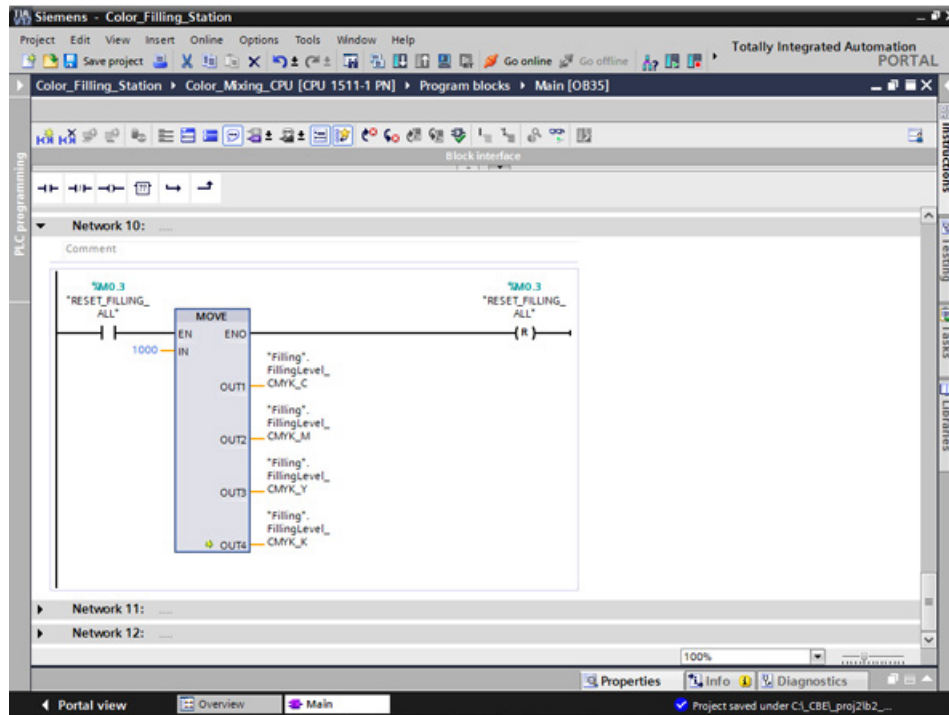
1. Open the Main[OB35] program block and insert the MOVE instruction into network 10.
2. Expand the instruction to a total of four outputs.
3. Insert a normally open contact before the MOVE instruction.
4. Insert a reset coil after the MOVE instruction.
5. Interconnect the inputs and outputs of the MOVE instruction.



6. Generate the instruction with the ENO shortcut menu.

## Result

The EN/ENO mechanism is interconnected for this block. If there are no errors during execution, the ENO enable output has the signal state "1". If there are errors during execution, the ENO enable output has the signal state "0".





### 3.2.12 Using the comment function

#### Introduction

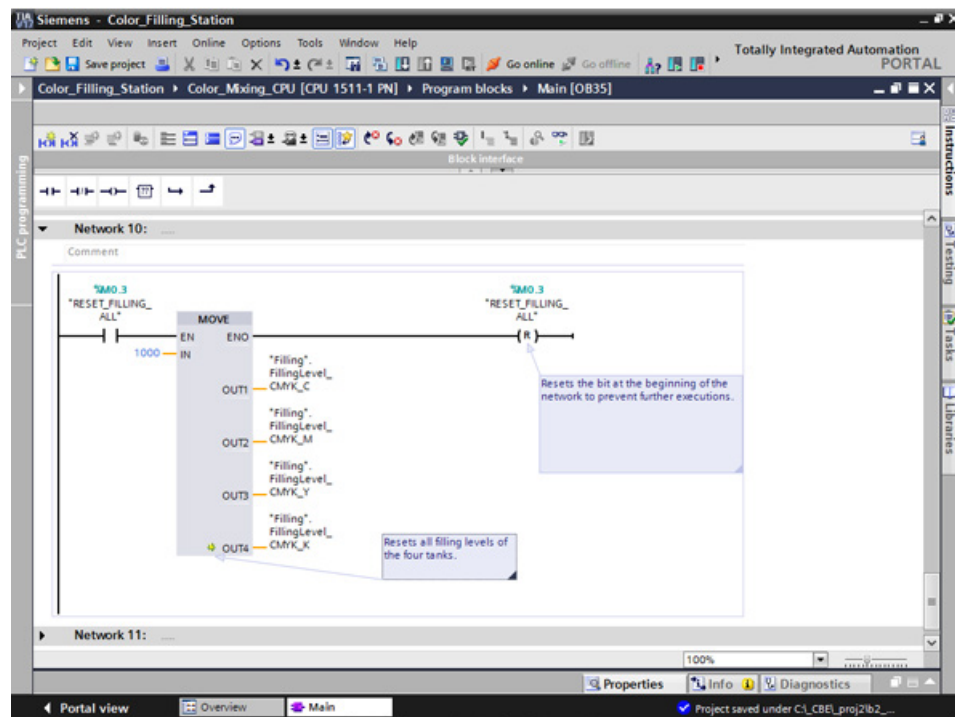
The MOVE and Reset instructions should be expanded with detailed commentary.

#### Procedure

1. Insert a comment using the shortcut menu.
2. Enter the comment text.

#### Result

The comments for the instruction and the coil are entered.



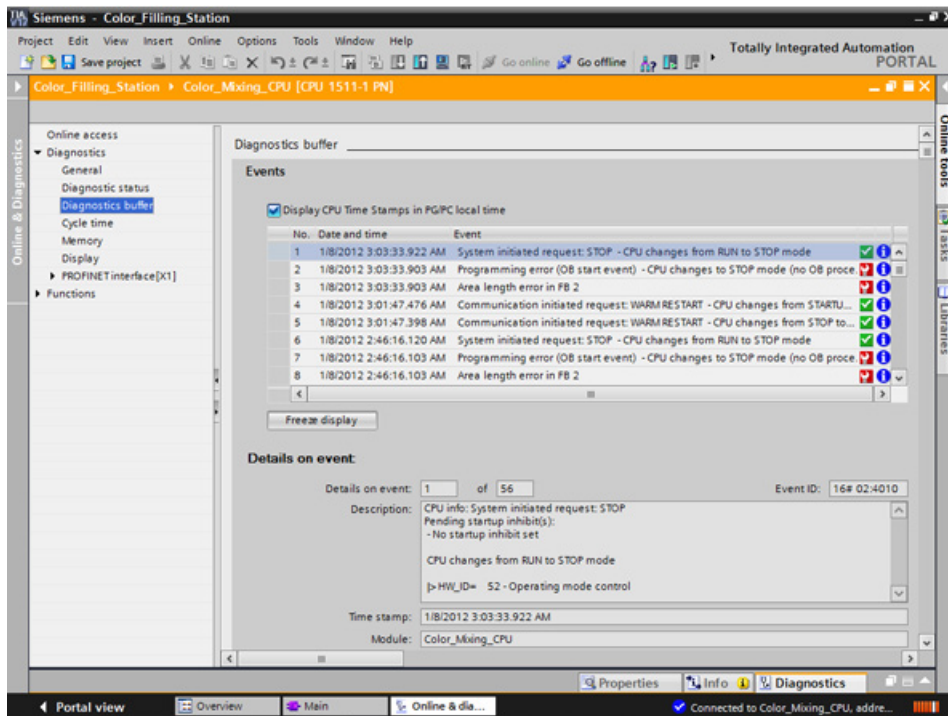


### 3.2.13 Local error handling

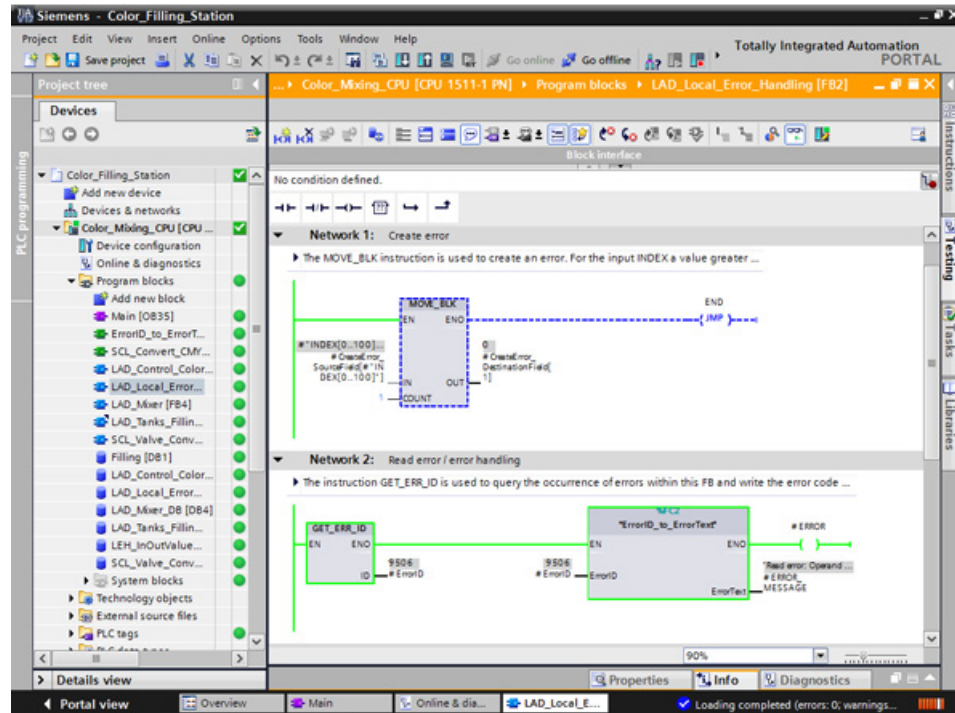
#### 3.2.13.1 Handle errors within block

#### Procedure

Unlike the CPUs of the S7-300/400, CPUs of the S7-1500 go to STOP with errors much less often. If an error occurs, it is entered in the diagnostics buffer of the CPU. You avoid the CPU STOP by using local error handling at each block. You should preferably enable local error handling during development of the user program.



You can precisely evaluate the information and, for example, program the error handling in the block with STL/FBD/LAD and SCL programs. The block generates an error ID that is evaluated by the "GET\_ERROR\_ID" instruction. You can call the "GET\_ERROR\_ID" instruction in both the MAIN block and in the function blocks. The CPU remains in RUN mode.



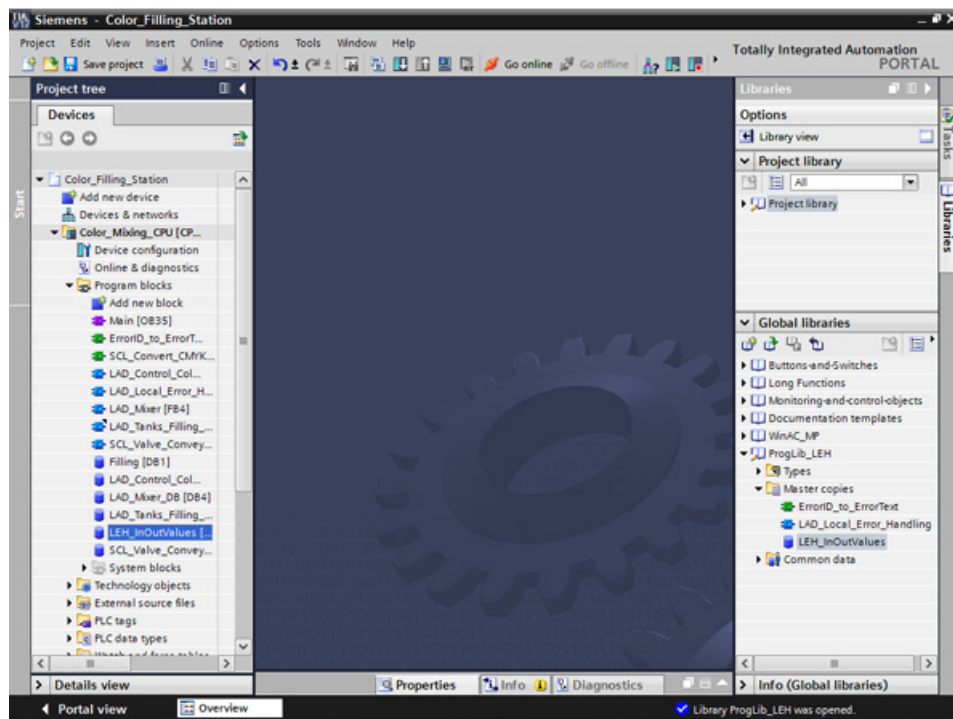
### 3.2.13.2 Loading blocks for local error handling

#### Introduction

To illustrate the local error handling, load the blocks of the "ProgLib\_LEH" library in the project. The blocks are used only to demonstrate the local error handling and are otherwise not used in the project.

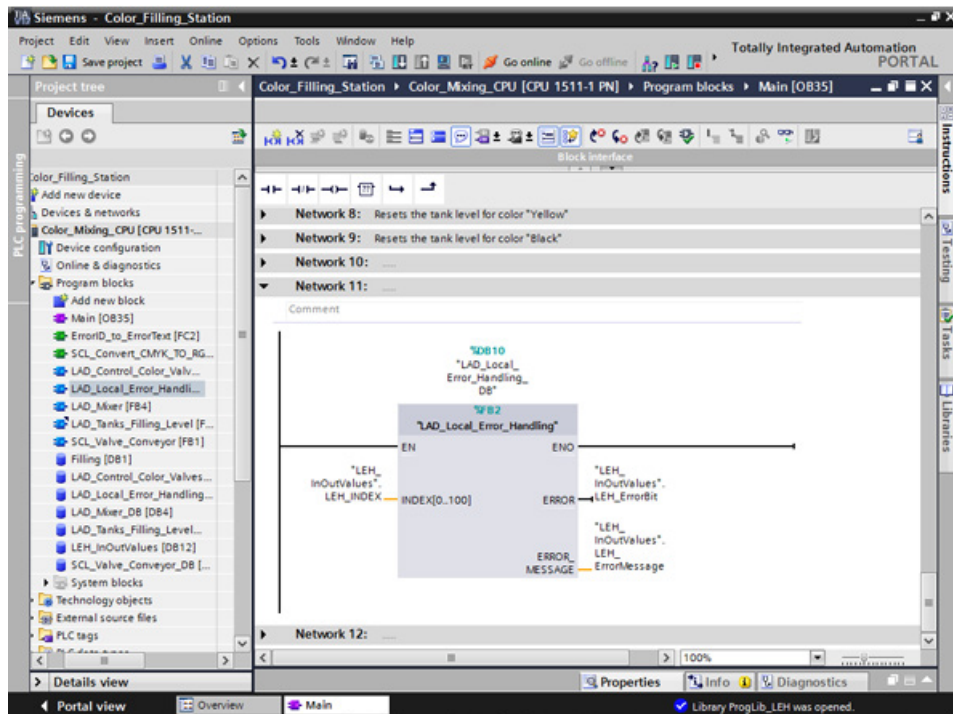
#### Procedure

1. Open the global library, "ProgLib\_LEH".
2. Copy the blocks from the master copies into the project.



3. Call the "LAD\_Local\_Error\_Handling" function block in an empty network of the "Main" block.

- Interconnect the parameters of the "LAD\_Local\_Error\_Handling" function block with tags of the "LEH\_InOutValues" data block.



- Connect to the CPU online.
- Compile and load the changes to the CPU.

## Result

Use the "LEH\_INDEX" tag at the "INDEX[0..100]" input parameter to trigger a programming error in the following. For example, if you set the input parameter to "101", an error at the output parameters is reported.

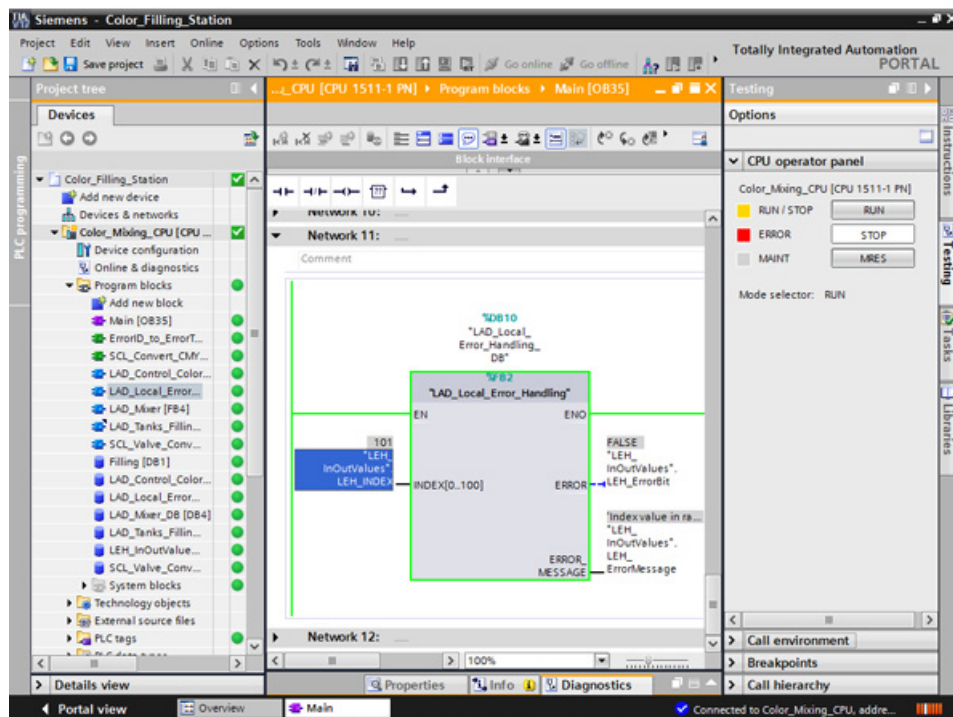
### 3.2.13.3 Generating errors without local error handling

#### Introduction

Perform the following steps to trigger a programming error without using the local error handling or creating a corresponding OB.

#### Procedure

1. Activate the "Monitor" function.
2. Set the value of the "LEH\_INDEX" tag to an invalid value, for example, "101". In the Testing dialog, the ERROR LED flashes briefly and the CPU goes from RUN to STOP.



3. Switch to the diagnostics buffer. The error and the error response is displayed in the diagnostics buffer.
4. Set the CPU back to RUN.

#### Result

The transition from STOP in RUN resets the "LEH\_INDEX" tag to the start value "0". This automatically solves the problem.

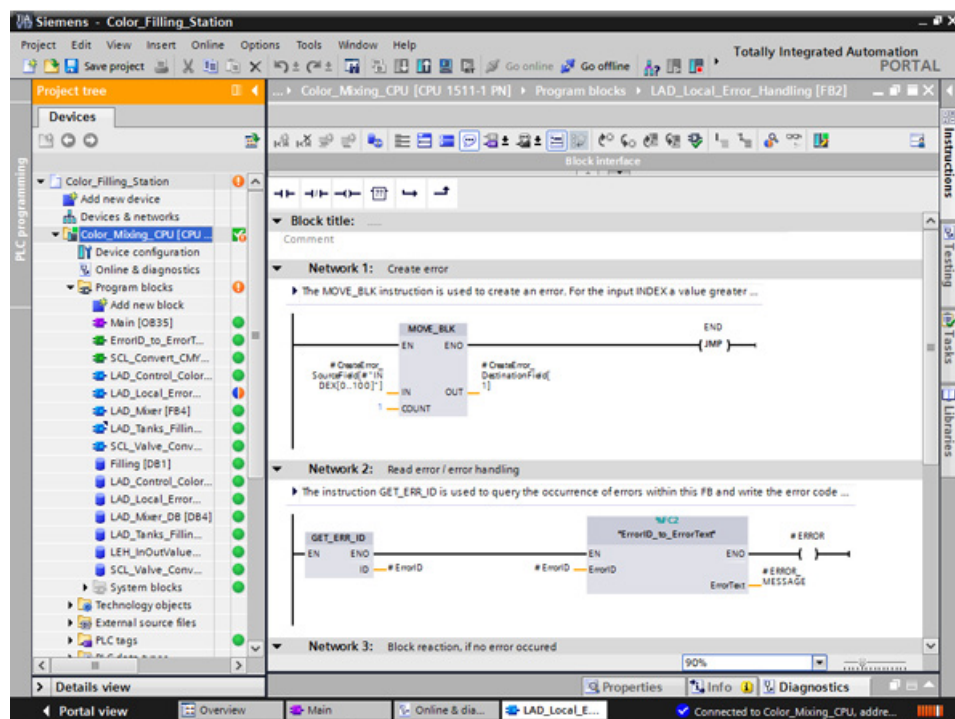
### 3.2.13.4 Generating errors with local error handling

#### Introduction

Perform the following steps to use "GET\_ERR\_ID" instruction and its ENO bit for the local error handling to respond to the error with an error message. This means the CPU remains in RUN mode.

#### Procedure

1. Open the "LAD\_Local\_Error\_Handling" function block.
2. Insert the "GET\_ERR\_ID" instruction in the second network and interconnect the "ID" output.

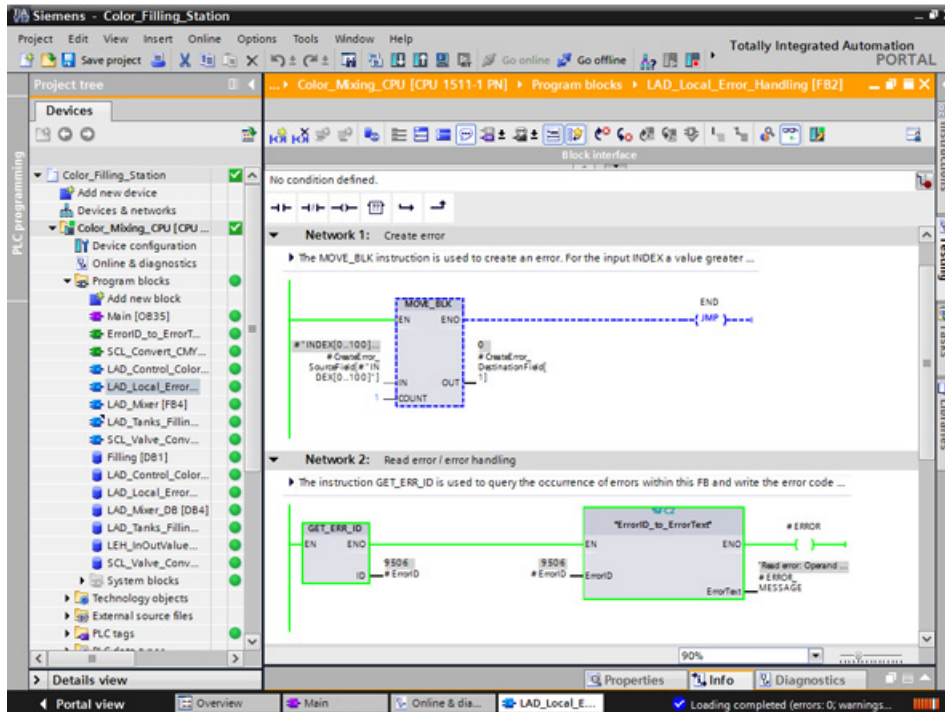


3. Call the "ErrorID\_to\_ErrorText" function from the project tree.
4. Interconnect the parameters of the "ErrorID\_to\_ErrorText" function so that they can convert the error code into an error message.
5. Load the changes to the CPU.
6. Trigger an error in the "Main" organization block by entering an invalid value, for example, "101". An error message is output at the "ERROR\_MESSAGE" parameter.



## Result

The error message is output as long as the error is not corrected. To correct the error, assign the "LEH\_INDEX" tag a valid value or restart the CPU.



## **3.3 Configure visualization**

### **3.3.1 Present sample project**

#### **Sample project for the application**

To configure the color mixing system with the TIA Portal, create the sample project "Color\_Filling\_Station". The following project components already exist for the sample project: The program blocks and tag tables of the CPU user program and a configured Comfort Panel with the necessary HMI screens, HMI tags and scripts.

In this section, we will explain the relationships between the individual project components of the sample project. You will carry out the necessary configuration steps yourself at a later point in time.

### **3.3.2 HMI configuration**

#### **3.3.2.1 Overview**

#### **Introduction of HMI configuration**

The supplied project includes the programmed CPU and the preconfigured HMI device in the "global library".

#### **HMI configuration**

In this section, we will introduce the HMI device and HMI configuration.

#### **Additional information**

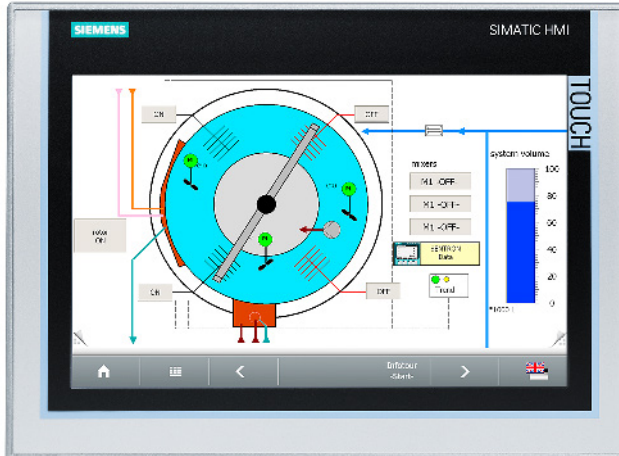
You can find detailed instructions on the HMI configuration on the Internet (<https://support.industry.siemens.com/cs/ww/en/ps/14678/man>); for this, search for "Getting Started".



### 3.3.2.2 SIMATIC HMI Comfort Panels

#### SIMATIC HMI Comfort Panels

The TP1200 Comfort HMI device from the Comfort Panel series is used to operate the color mixing system.



Comfort Panels are particularly suitable for challenging HMI tasks in PROFINET and PROFIBUS environments and are characterized by the following features:

- High-quality housing and numerous interfaces
- Industrial widescreen displays with large visualization area, optimum viewing angle stability and maximum brightness
- Installation either in horizontal or vertical format
- Exact diagnostics with system diagnostics viewer

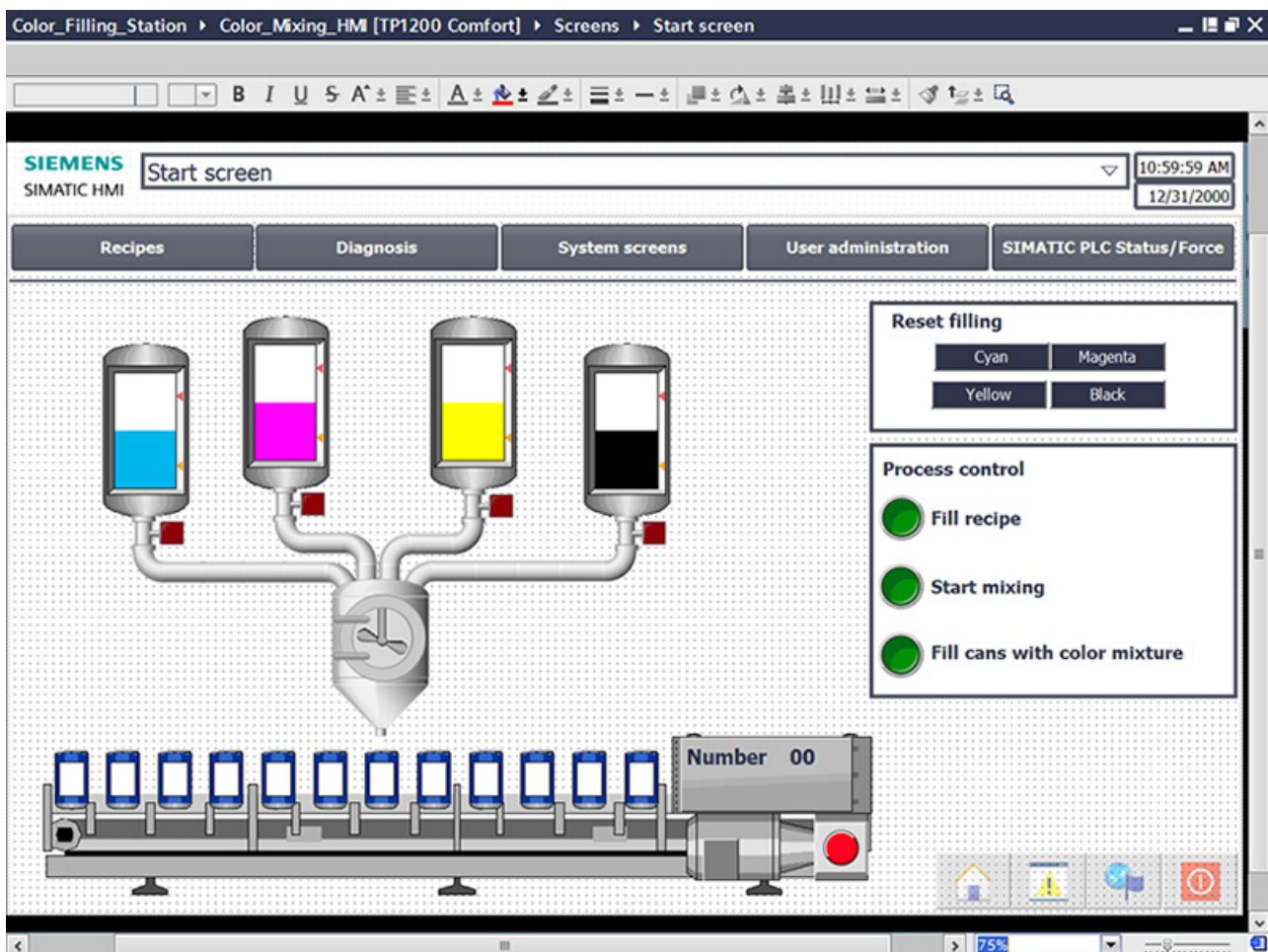
### 3.3.2.3 HMI screens

#### HMI screens

You use the screens loaded onto the respective HMI device to operate and monitor machines and plants in runtime.

You manage the screens in WinCC under "Screens" in the project navigation.

The start screen of the HMI device is used to visualize the color mixing system as well as the most important status information and key figures.



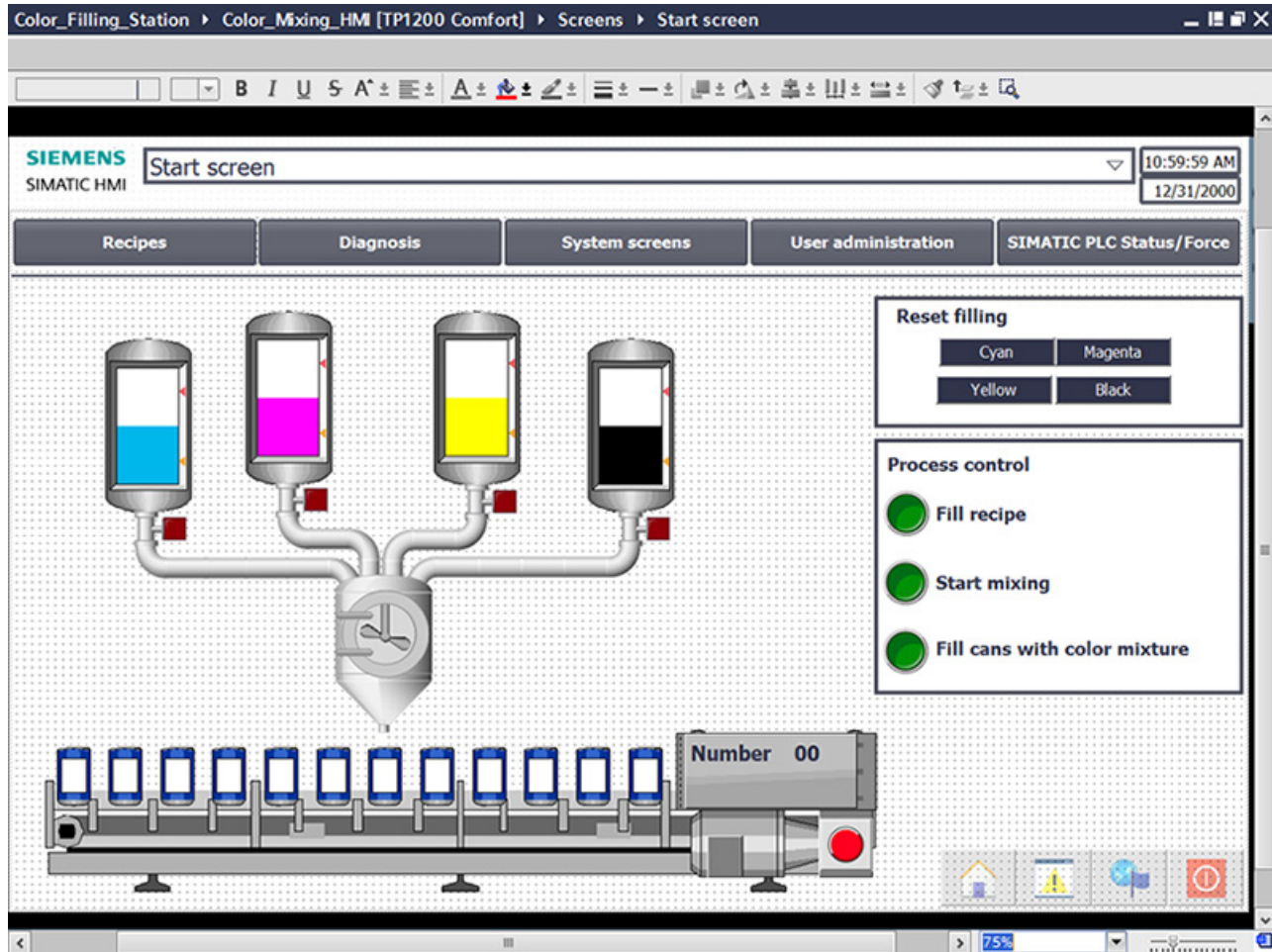
The color mixing system includes the following elements:

- A color reservoir for each print color with fill level display
- Mixer
- Feeder pipes to the mixer
- Conveyor belt with emergency stop switch

## 3.3.2.4 Additional control elements

## Additional control elements

The process steps "Mixing color" and "Filling color" are to be displayed as animations with dynamic visualization objects.



The start screen of the example project includes additional control objects:

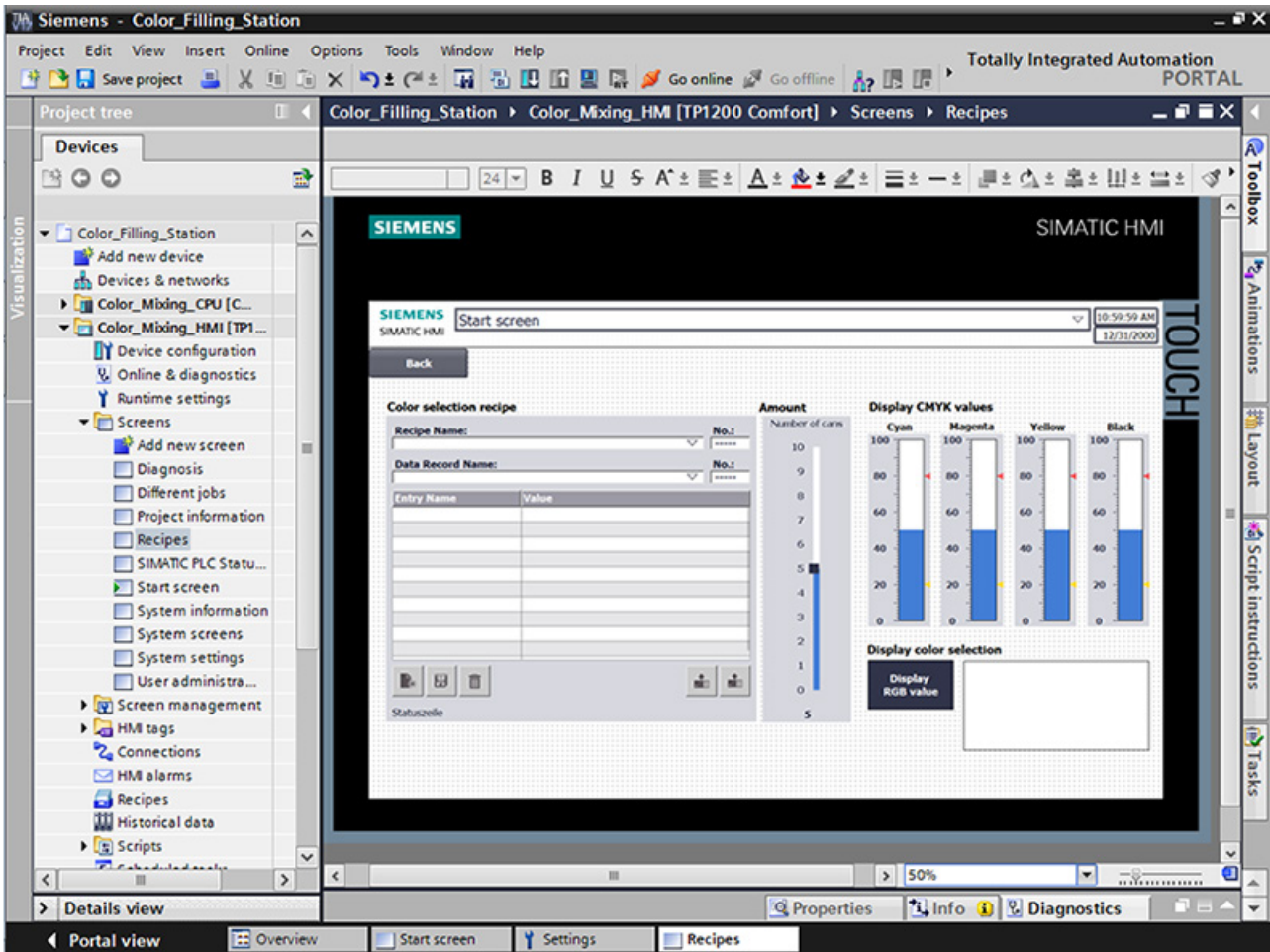
- Buttons for screen changes
- Buttons to reset the fill levels
- Buttons for operating and monitoring the system: Fill recipe, start mixing process, fill color mixture

### 3.3.2.5 Recipes

#### Recipes

A recipe contains related production parameters, such as mixing ratios.

The required mixing ratio can be transferred from the HMI device to the color mixing system in a single step, for example, to switch production from dark orange to signal yellow.



The color mixing system can produce the mixed colors "Orange", "Amber", "Green" and "Red".

A recipe data record is created for each color. The recipe data record includes the percentage of basic colors which result in the respective mixed color.

The recipe consists of relevant parameters and the recipe data records in which the mixing ratios for the individual shades of color are stored.



### 3.3.2.6 Archives

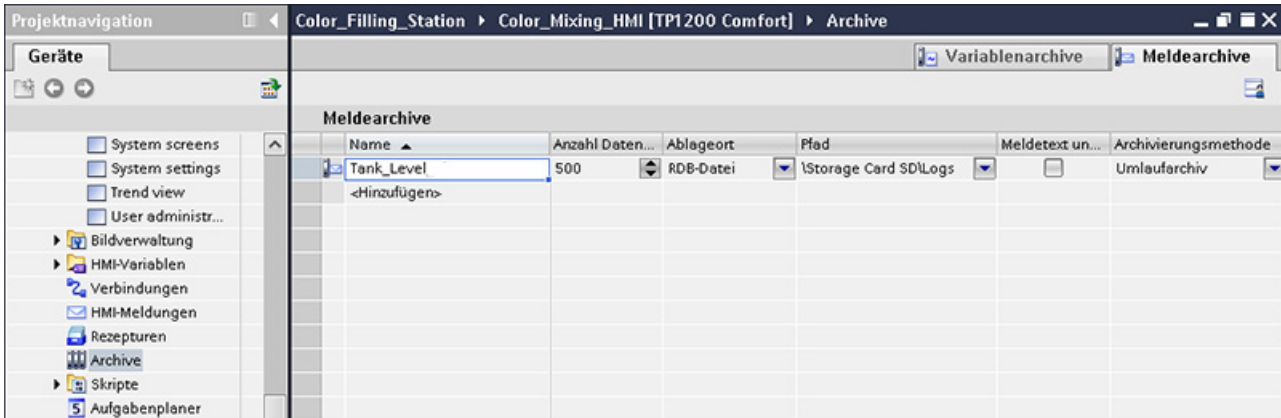
#### Archives

To record operational events of a system, the alarms and process values generated during production are saved to logs.

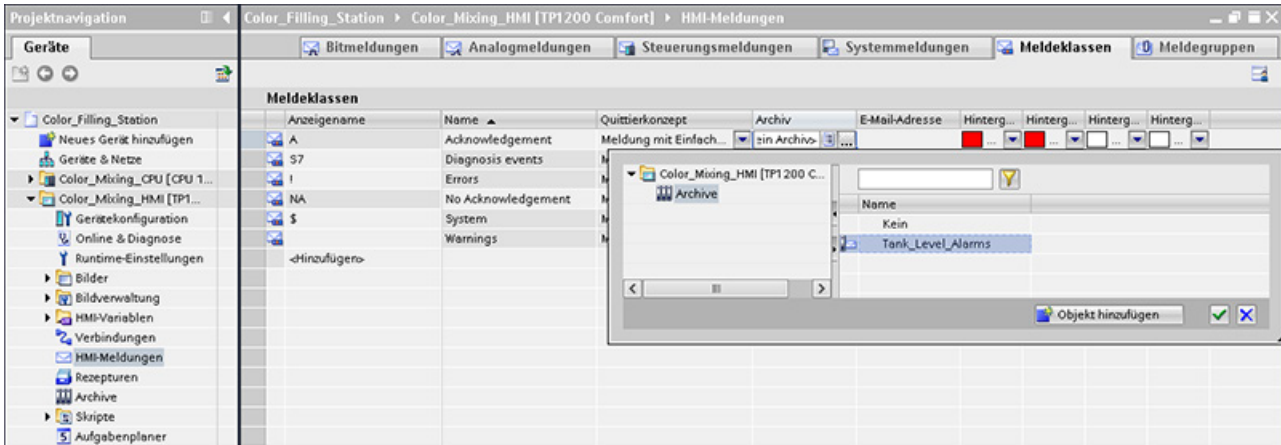
You can then evaluate the alarms and process data logs.

The fill levels of the color reservoirs are to be documented for the color mixing system.

You have configured the alarm log "Tank\_Level" for this purpose.



This log stores alarms for fill levels that were too low and fill levels that were too high during a shift.



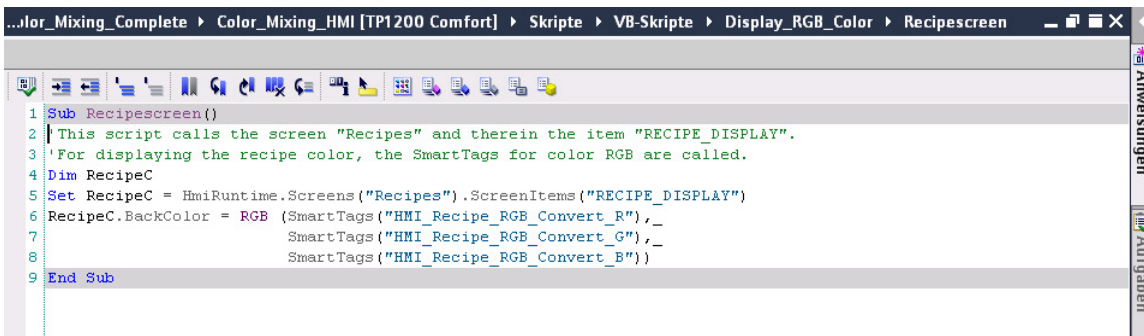
### 3.3.2.7 User-defined functions

#### Scripts

You use the user-defined functions to program additional functionality for the HMI device. WinCC offers a VBS programming interface to create user-defined functions.

The example project uses two user-defined functions to display the mixed print color on the monitor in different screens.

- "Recipescreen" displays a rectangle in the selected color in the "Recipes" screen.

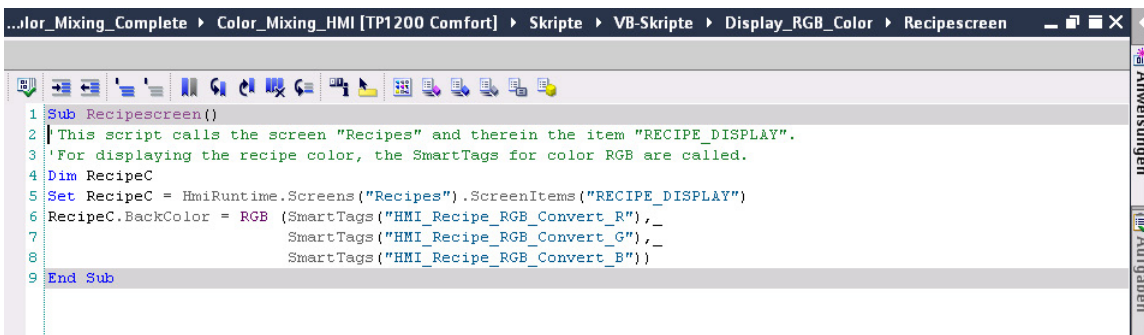


```
1 Sub Recipescreen()  
2 | This script calls the screen "Recipes" and therein the item "RECIPE_DISPLAY".  
3 | For displaying the recipe color, the SmartTags for color RGB are called.  
4 Dim RecipeC  
5 Set RecipeC = HmiRuntime.Screens("Recipes").ScreenItems("RECIPE_DISPLAY")  
6 RecipeC.BackColor = RGB (SmartTags("HMI_Recipe_RGB_Convert_R"), _  
7 SmartTags("HMI_Recipe_RGB_Convert_G"), _  
8 SmartTags("HMI_Recipe_RGB_Convert_B"))  
9 End Sub
```

- "Startscreen" displays the label of the filled cans in the currently mixed color in the system overview of the start screen.

The example project uses two user-defined functions to display the mixed print color on the monitor in different screens.

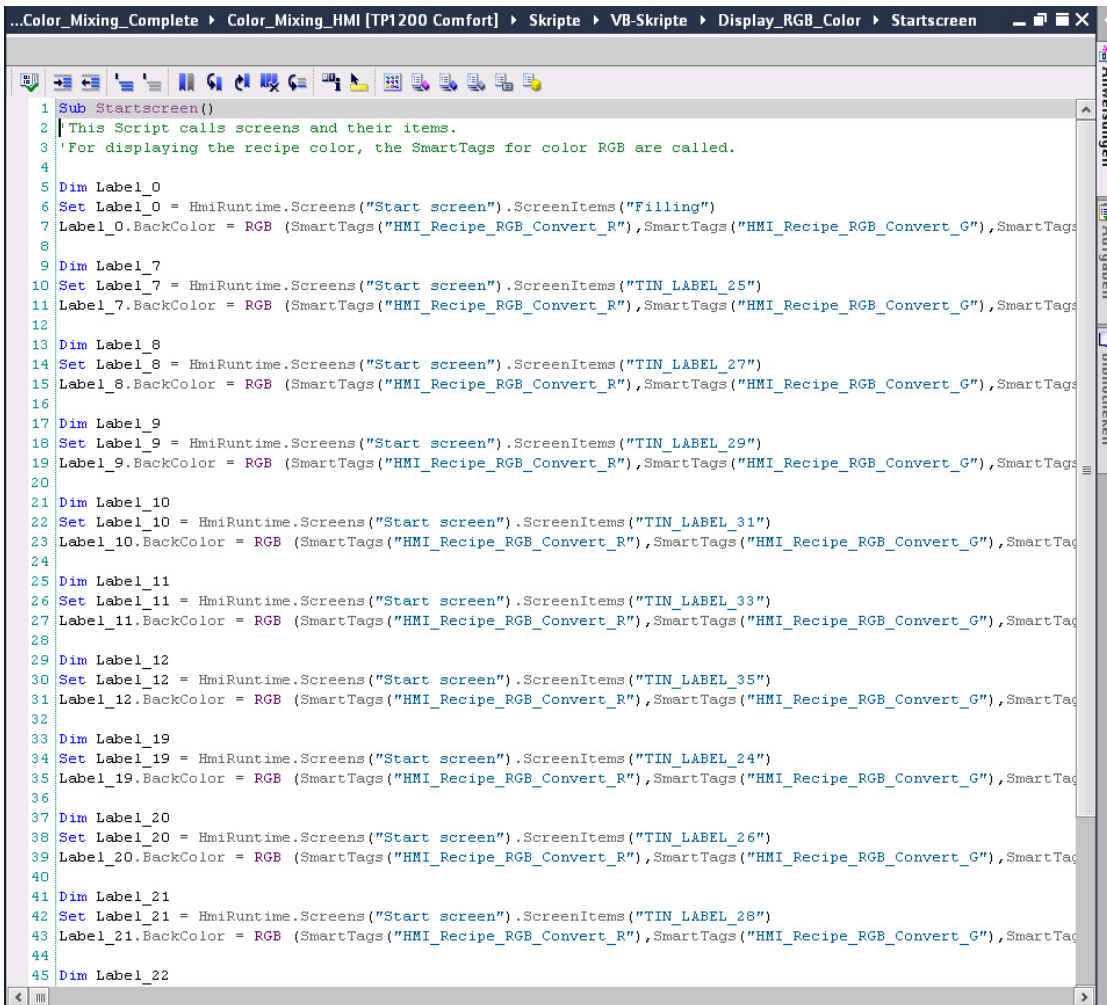
- "Recipescreen" displays a rectangle in the selected color in the "Recipes" screen.



```
1 Sub Recipescreen()  
2 | This script calls the screen "Recipes" and therein the item "RECIPE_DISPLAY".  
3 | For displaying the recipe color, the SmartTags for color RGB are called.  
4 Dim RecipeC  
5 Set RecipeC = HmiRuntime.Screens("Recipes").ScreenItems("RECIPE_DISPLAY")  
6 RecipeC.BackColor = RGB (SmartTags("HMI_Recipe_RGB_Convert_R"), _  
7 SmartTags("HMI_Recipe_RGB_Convert_G"), _  
8 SmartTags("HMI_Recipe_RGB_Convert_B"))  
9 End Sub
```

## 3.3 Configure visualization

- "Startscreen" displays the label of the filled cans in the currently mixed color in the system overview of the start screen.

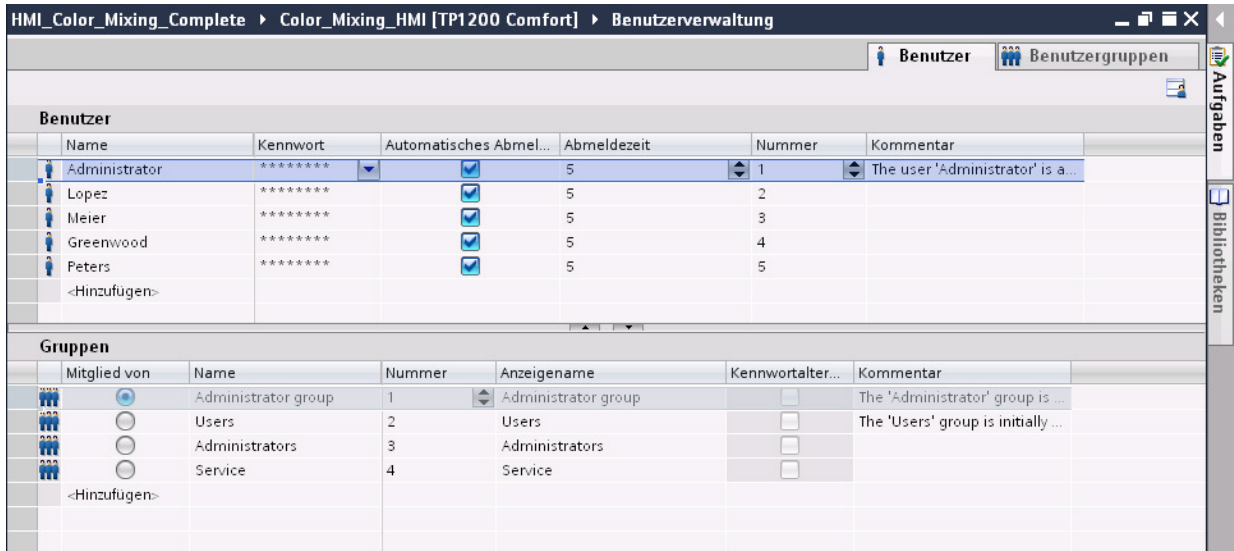


```
1 Sub Startscreen()  
2 | This Script calls screens and their items.  
3 | For displaying the recipe color, the SmartTags for color RGB are called.  
4  
5 Dim Label_0  
6 Set Label_0 = HmiRuntime.Screens("Start screen").ScreenItems("Filling")  
7 Label_0.BackColor = RGB (SmartTags("HMI_Recipe_RGB_Convert_R"),SmartTags("HMI_Recipe_RGB_Convert_G"),SmartTags  
8  
9 Dim Label_7  
10 Set Label_7 = HmiRuntime.Screens("Start screen").ScreenItems("TIN_LABEL_25")  
11 Label_7.BackColor = RGB (SmartTags("HMI_Recipe_RGB_Convert_R"),SmartTags("HMI_Recipe_RGB_Convert_G"),SmartTags  
12  
13 Dim Label_8  
14 Set Label_8 = HmiRuntime.Screens("Start screen").ScreenItems("TIN_LABEL_27")  
15 Label_8.BackColor = RGB (SmartTags("HMI_Recipe_RGB_Convert_R"),SmartTags("HMI_Recipe_RGB_Convert_G"),SmartTags  
16  
17 Dim Label_9  
18 Set Label_9 = HmiRuntime.Screens("Start screen").ScreenItems("TIN_LABEL_29")  
19 Label_9.BackColor = RGB (SmartTags("HMI_Recipe_RGB_Convert_R"),SmartTags("HMI_Recipe_RGB_Convert_G"),SmartTags  
20  
21 Dim Label_10  
22 Set Label_10 = HmiRuntime.Screens("Start screen").ScreenItems("TIN_LABEL_31")  
23 Label_10.BackColor = RGB (SmartTags("HMI_Recipe_RGB_Convert_R"),SmartTags("HMI_Recipe_RGB_Convert_G"),SmartTag  
24  
25 Dim Label_11  
26 Set Label_11 = HmiRuntime.Screens("Start screen").ScreenItems("TIN_LABEL_33")  
27 Label_11.BackColor = RGB (SmartTags("HMI_Recipe_RGB_Convert_R"),SmartTags("HMI_Recipe_RGB_Convert_G"),SmartTag  
28  
29 Dim Label_12  
30 Set Label_12 = HmiRuntime.Screens("Start screen").ScreenItems("TIN_LABEL_35")  
31 Label_12.BackColor = RGB (SmartTags("HMI_Recipe_RGB_Convert_R"),SmartTags("HMI_Recipe_RGB_Convert_G"),SmartTag  
32  
33 Dim Label_19  
34 Set Label_19 = HmiRuntime.Screens("Start screen").ScreenItems("TIN_LABEL_24")  
35 Label_19.BackColor = RGB (SmartTags("HMI_Recipe_RGB_Convert_R"),SmartTags("HMI_Recipe_RGB_Convert_G"),SmartTag  
36  
37 Dim Label_20  
38 Set Label_20 = HmiRuntime.Screens("Start screen").ScreenItems("TIN_LABEL_26")  
39 Label_20.BackColor = RGB (SmartTags("HMI_Recipe_RGB_Convert_R"),SmartTags("HMI_Recipe_RGB_Convert_G"),SmartTag  
40  
41 Dim Label_21  
42 Set Label_21 = HmiRuntime.Screens("Start screen").ScreenItems("TIN_LABEL_28")  
43 Label_21.BackColor = RGB (SmartTags("HMI_Recipe_RGB_Convert_R"),SmartTags("HMI_Recipe_RGB_Convert_G"),SmartTag  
44  
45 Dim Label_22
```

### 3.3.2.8 User Management

#### User Management

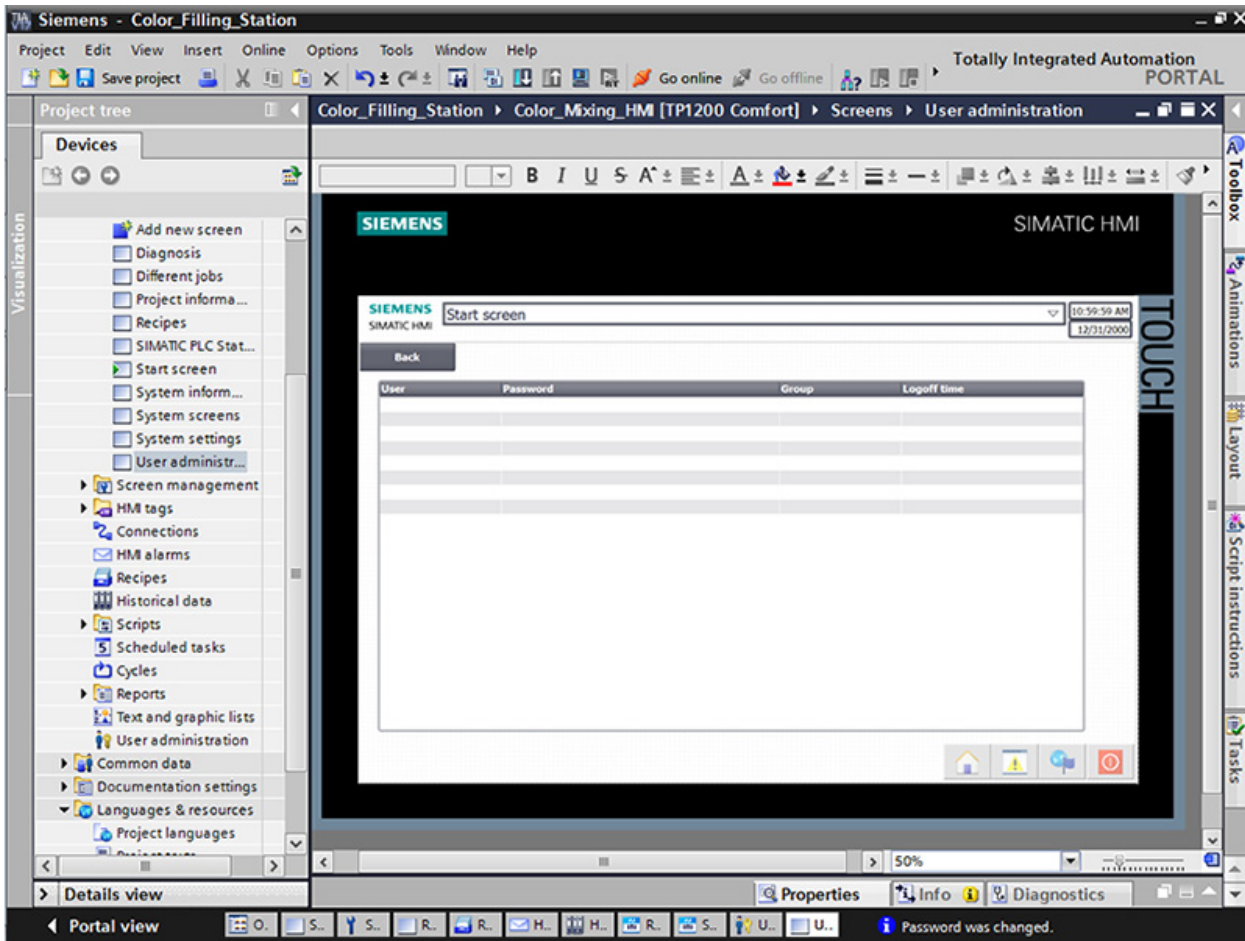
WinCC gives you the option to restrict safety-related operations to special user groups and thus protect data and functions from unauthorized access in Runtime.



The "User view" object offers management of users and passwords on the HMI device.



3.3 Configure visualization



Users with user management authorization have access to the full range of functions in the user view.

They can create and delete users and change their own password or that of other users.

### 3.3.2.9 Multilingualism

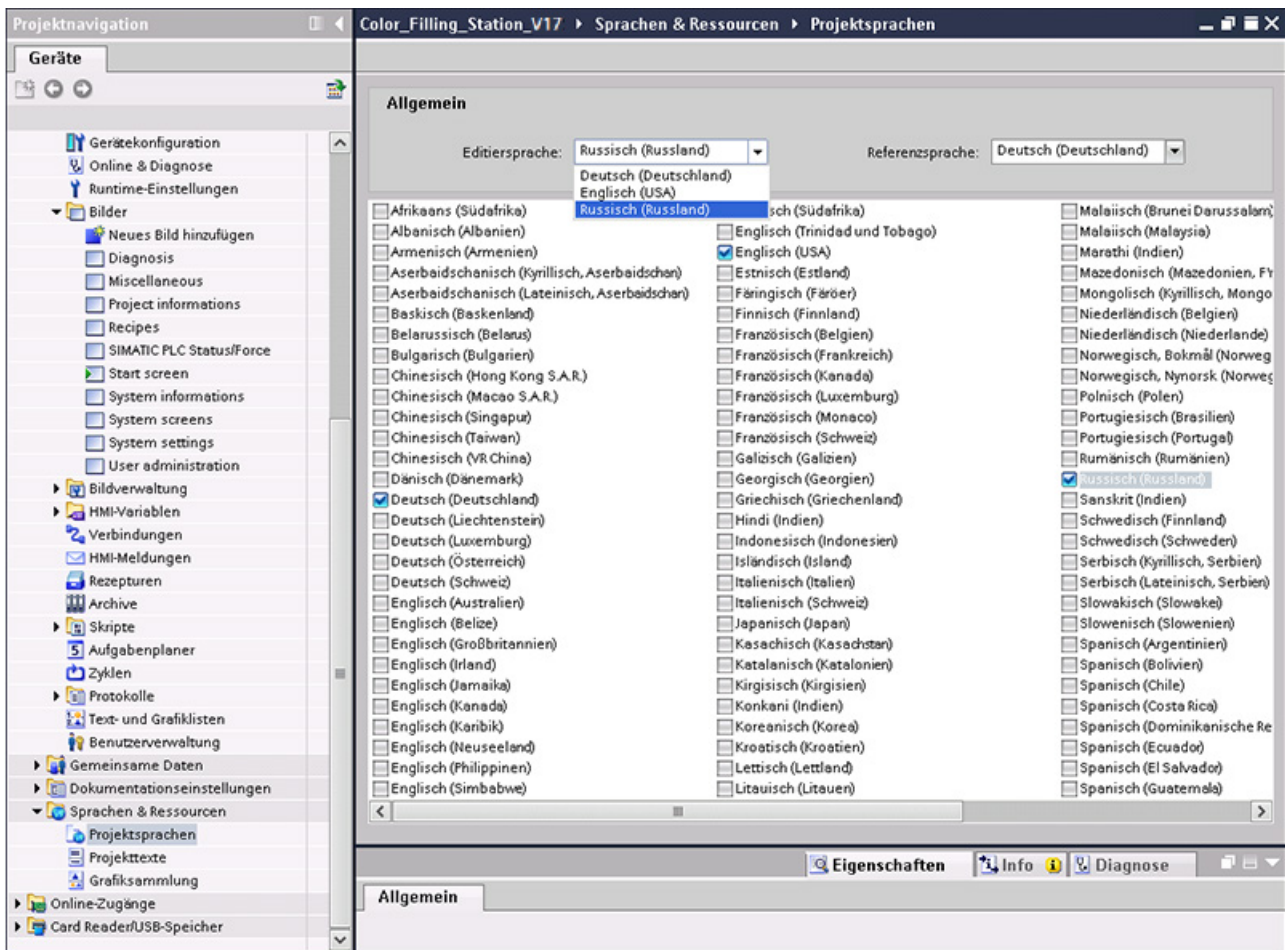
#### Multilingualism

WinCC supports multilingual user interfaces.

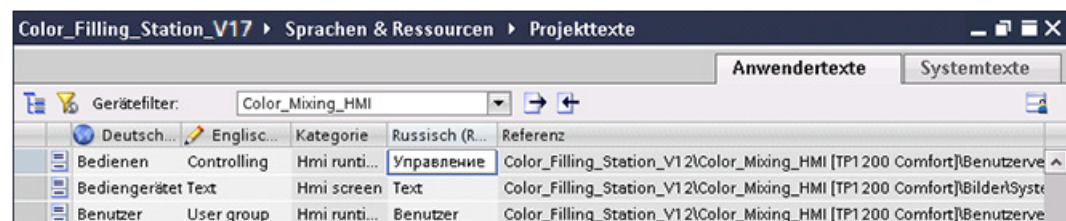
The color mixing system is operated in a new subsidiary in Russia.

A Russian user interface is required for maintenance and service technicians.

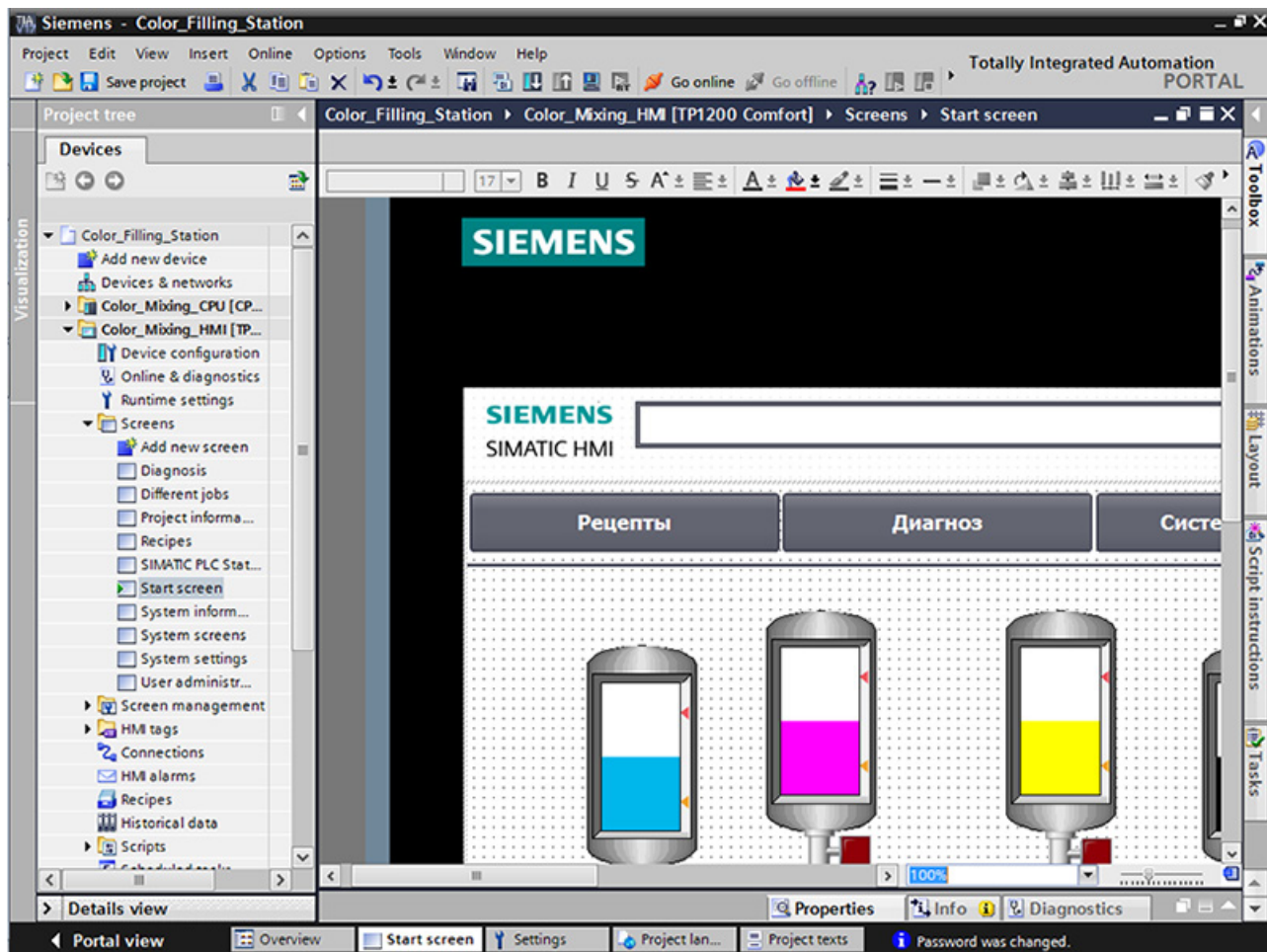
The example project has been expanded by another language for this purpose.



The texts are imported again after they have been exported and translated into Russian.



The Russian texts are displayed in Runtime in case of a language selection.



### 3.3.2.10 Reports

#### Reports

Reports are used to record events in a production process as a basis for product testing and quality control. Alarms and recipe data are output at regular intervals in the form of shift reports for this purpose.

A report has been created in WinCC for the "Tank\_Level" log with alarms for the fill level.

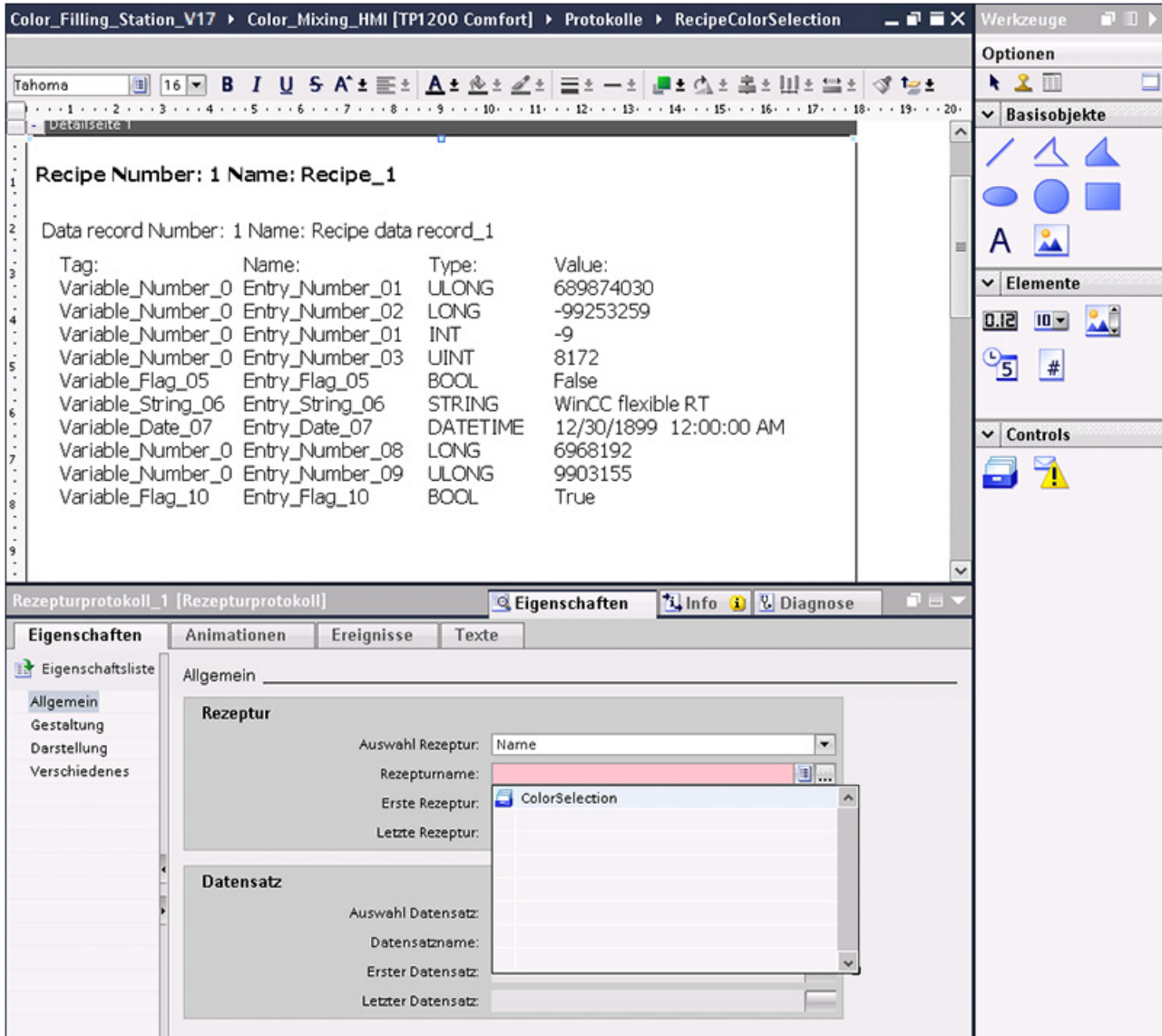
The screenshot shows the WinCC interface with the 'TankLevel' report configuration. The main window displays a table of report entries with the following data:

No.	Time	Status	Date	GR	PLC
0	12:00:00 PM	KGQ	1/1/1999	Class name	0 * Device
1	12:00:00 PM	KGQ	1/1/1999	Class name	1 * Device
2	12:00:00 PM	KGQ	1/1/1999	Class name	2 * Device
3	12:00:00 PM	KGQ	1/1/1999	Class name	3 * Device
4	12:00:00 PM	KGQ	1/1/1999	Class name	4 * Device

The 'Eigenschaften' (Properties) window is open, showing the 'Meldeklassen' (Alarm Classes) section. The 'Aktivieren' (Activate) column has checkboxes for 'Errors', 'Warnings', 'System', 'Diagnosis events', 'Acknowledgement', and 'No Acknowledgement'. The 'Acknowledgement' and 'No Acknowledgement' checkboxes are checked and circled in red.



Furthermore, a report for recipes has been created in this project.



The reports should be output on a daily basis to a printer which is connected to the HMI device.

The cyclical output was created with the help of the Scheduler.

### 3.3.3 Insert HMI device from libraries

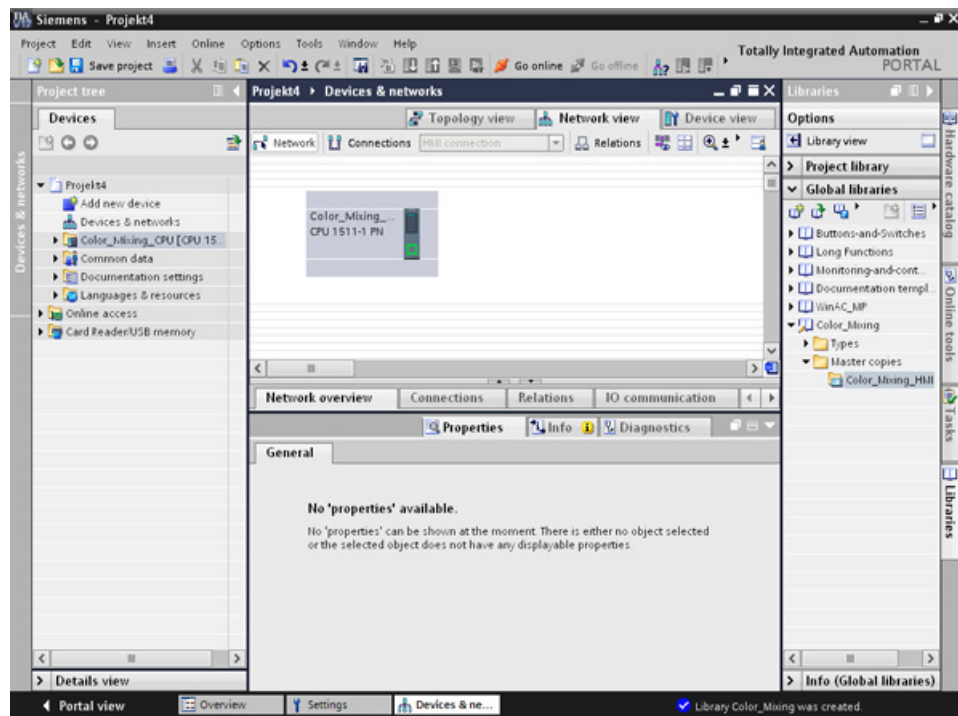
#### 3.3.3.1 Insert HMI device from library

##### Introduction

The global library includes a preconfigured HMI device.

##### Procedure

1. Open the global library.
2. Drag-and-drop the HMI device "Color\_Mixing\_HMI" into the "Devices & Networks" editor.



3. The mouse pointer changes into a crosshair with an object symbol attached.

##### Result

The preconfigured HMI device is created and can be connected to the CPU.

### 3.3.4 Configuring HMI connection

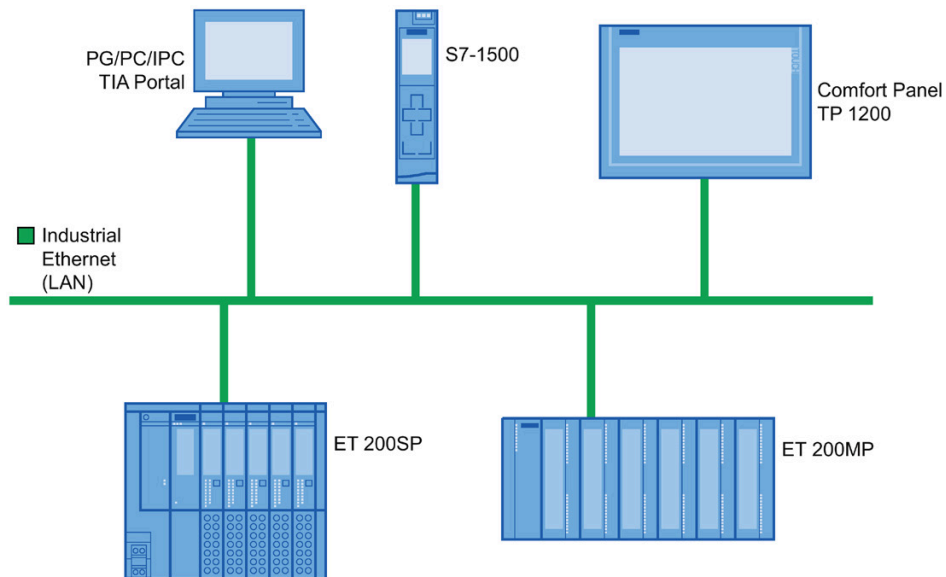
#### 3.3.4.1 Communication between devices

##### Communication

The data exchange between devices is referred to as communication.

The devices can be interconnected directly or via a network.

The interconnected devices in communication are referred to as communication partners.



Data transferred between the communication partners may serve different purposes:

- Display processes
- Operate processes
- Output alarms
- Archive process values and alarms
- Document process values and alarms
- Administer process parameters and machine parameters

##### Basic information for all communication

The basis for all types of communication is a network configuration. In a network configuration, you specify the connection that exists between the configured devices.

With the network configuration, you also ensure the necessary prerequisites for communication, in other words:

- Every device in a network is assigned a unique address.
- The devices carry out communication with consistent transmission characteristics.

### 3.3.4.2 Configuring HMI connection

#### Introduction

You configure an HMI connection between the Comfort Panel TP1200 and the CPU via PROFINET in the "Devices & Networks" editor.

The available communication partners in the project are displayed graphically in the network view.

 **CAUTION**

**Communication via Ethernet**

In Ethernet-based communication, the end user is responsible for the security of his data network.

Targeted attacks can overload the device and interfere with proper functioning.

#### Requirements

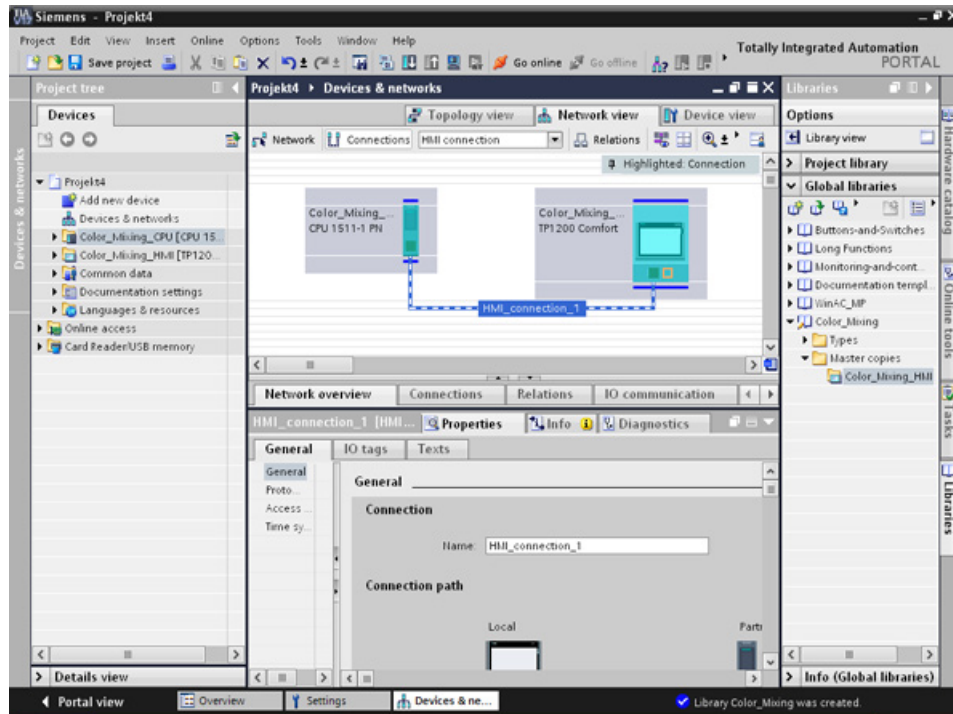
The following communication partners are created in the "Devices & Networks" editor:

- HMI device: SIMATIC Comfort Panel
- CPU: SIMATIC S7-1500



## Procedure

1. Click the "Connections" button and select "HMI connection" for the connection type.  
The devices available for connection are highlighted in color.
2. Click the PROFINET interface of the CPU and drag-and-drop a connection to the PROFINET interface of the HMI device.



3. Click the communication partners in the "Network view" and change the PROFINET parameters in the Inspector window according to the requirements of your project.

## Note

The created HMI connection is also shown in the tabular area of the editor in the "Connections" tab. You check the connection parameters in the table.

You can change the local name for the connection only in the table.

## Result

You have created a connection between an HMI device and the CPU.

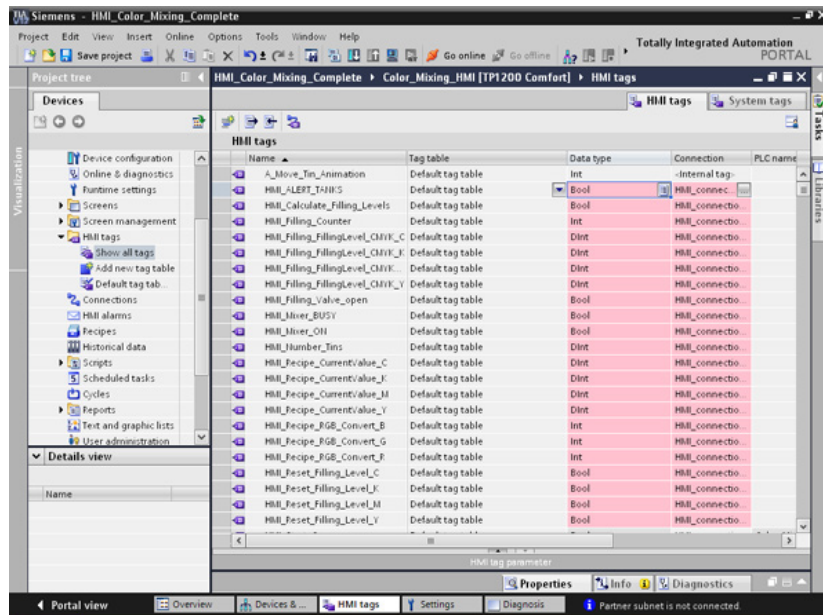
### **3.3.4.3 Connecting HMI tags**

#### **Introduction**

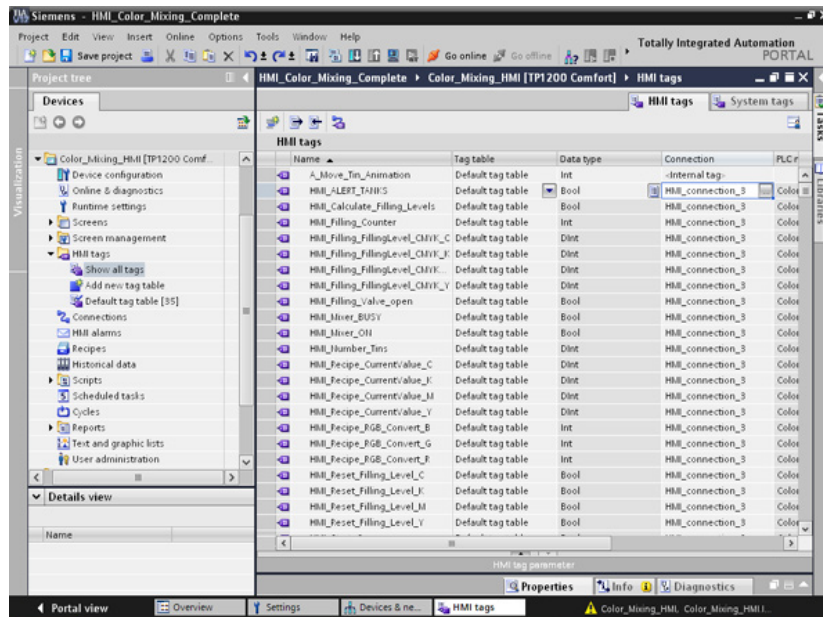
Once you have created the connection of CPU and HMI device, connect the tags of the two devices.

## Procedure

1. Open the HMI tag editor.



2. Select the HMI connection you have just configured in the "Connections" column.
3. Repeat this procedure for all entries highlighted in red.



## Result

An HMI connection had already been created for tags already configured in the CPU and HMI device.

You have restored this HMI connection.

## **3.3.5 Configuring system diagnostics**

### **3.3.5.1 System diagnostics basics**

#### **Introduction**

You use system diagnostics to detect problems and errors in any part of your plant. WinCC has two display and operating elements for quick error localization.

#### **System diagnostics view**

The alarm view shows the status of a CPU while the system diagnostics view gives you an overview of all devices available in your system: You navigate directly to the cause of the error and to the relevant device. You have access to all devices supporting diagnostics you have configured in the "Devices & networks" editor.

#### **System diagnostics window**

The system diagnostics window is an operating and display element that you can only use in the global screen.

The functions of the system diagnostics window are no different than those of the system diagnostics view. Because the system diagnostics window is configured in the global screen, you can, for example, also specify if the object is closable in Runtime.



## Diagnostic buffer view

The current data from the diagnostic buffer are shown in the diagnostic buffer view.

N	Date	Time	Event
1	1/11/2012	10:49:59 PM	Communication initiated request: WARM RESTART - CPU changes from STARTUP to RUN ...
2	1/11/2012	10:49:59 PM	Communication initiated request: WARM RESTART - CPU changes from STOP to STARTUP...
3	1/11/2012	10:49:56 PM	Communication initiated request: STOP - CPU changes from RUN to STOP mode
4	1/11/2012	7:32:36 AM	Communication initiated request: WARM RESTART - CPU changes from STARTUP to RUN ...
5	1/11/2012	7:32:36 AM	Communication initiated request: WARM RESTART - CPU changes from STOP to STARTUP...
6	1/11/2012	7:32:34 AM	Communication initiated request: STOP - CPU changes from RUN to STOP mode
7	1/11/2012	7:30:53 AM	Communication initiated request: WARM RESTART - CPU changes from STARTUP to RUN ...
8	1/11/2012	7:30:53 AM	Communication initiated request: WARM RESTART - CPU changes from STOP to STARTUP...
9	1/11/2012	7:21:03 AM	Communication initiated request: STOP - CPU changes from RUN to STOP mode
...	1/11/2012	4:33:55 AM	Communication initiated request: WARM RESTART - CPU changes from STARTUP to RUN ...
...	1/11/2012	4:33:55 AM	Communication initiated request: WARM RESTART - CPU changes from STOP to STARTUP...
...	1/11/2012	4:22:31 AM	Follow-on operating mode change - CPU changes from STOP (initialization) to STOP mode
...	1/11/2012	4:22:31 AM	Power on - CPU changes from NO POWER to STOP (initialization) mode
...	1/11/2012	4:21:59 AM	Power off - CPU changes from STOP to NO POWER mode
...	1/11/2012	4:19:51 AM	System initiated request: STOP - CPU changes from STOP (firmware update) to STOP mode
...	1/11/2012	4:19:51 AM	Device firmware update finished - CPU changes to STOP mode, new startup inhibit set: .
...	1/11/2012	4:19:51 AM	CPU firmware update: Completed successfully - Device firmware update sequence contin...
...	1/11/2012	4:19:03 AM	System initiated request: Firmware update - CPU changes from STOP (initialization) to ST...

## Detail view

The detail view gives detailed information about the selected device and any pending errors. Check whether the data is correct in the detail view. You cannot sort error texts in the detail view.

S7-1500 station_1 \ DI32	
> Status	
> Name	DI32
> Betriebszustand	
> Baugruppenträger	0
> Steckplatz	3
> Typ	DI32
> Bestellnummer	6ES7 521-1BL00-0AB0
> Adresse	259*
> Anlagenbezeichnung	
> Positionskennung	
> Hersteller ID	SIEMENS AG
> Hardware Version	97
> Profil ID	
> spezifische Profildaten	0003
> I&M Datenversion	1.1
> Fehlertext	Drahtbruch
	Hilfe: Kontrollieren Sie den Zustand der Verbindungsleitungen.






### Matrix view

The matrix view is only available for master systems. The matrix view shows the status of the subdevices of the master system.

- In PROFIBUS, the numbers assigned by Profibus are used as identification (DP station number).
- The IO devices are numbered consecutively from 1 in PROFINET.



### Navigation buttons

Button	Function
	Opens the child devices or the detail view if there are no child devices.
	Opens the parent device or the device view if there is no parent device.
	Opens the device view.
	Opens the diagnostic buffer view. Only visible in the device view.
	Refreshes the view. Configured softkey, for example F2

### **3.3.5.3 Configuring the system diagnostic view**

#### **Introduction**

You add a system diagnostics view to your project to get an overview of all devices available in your plant.

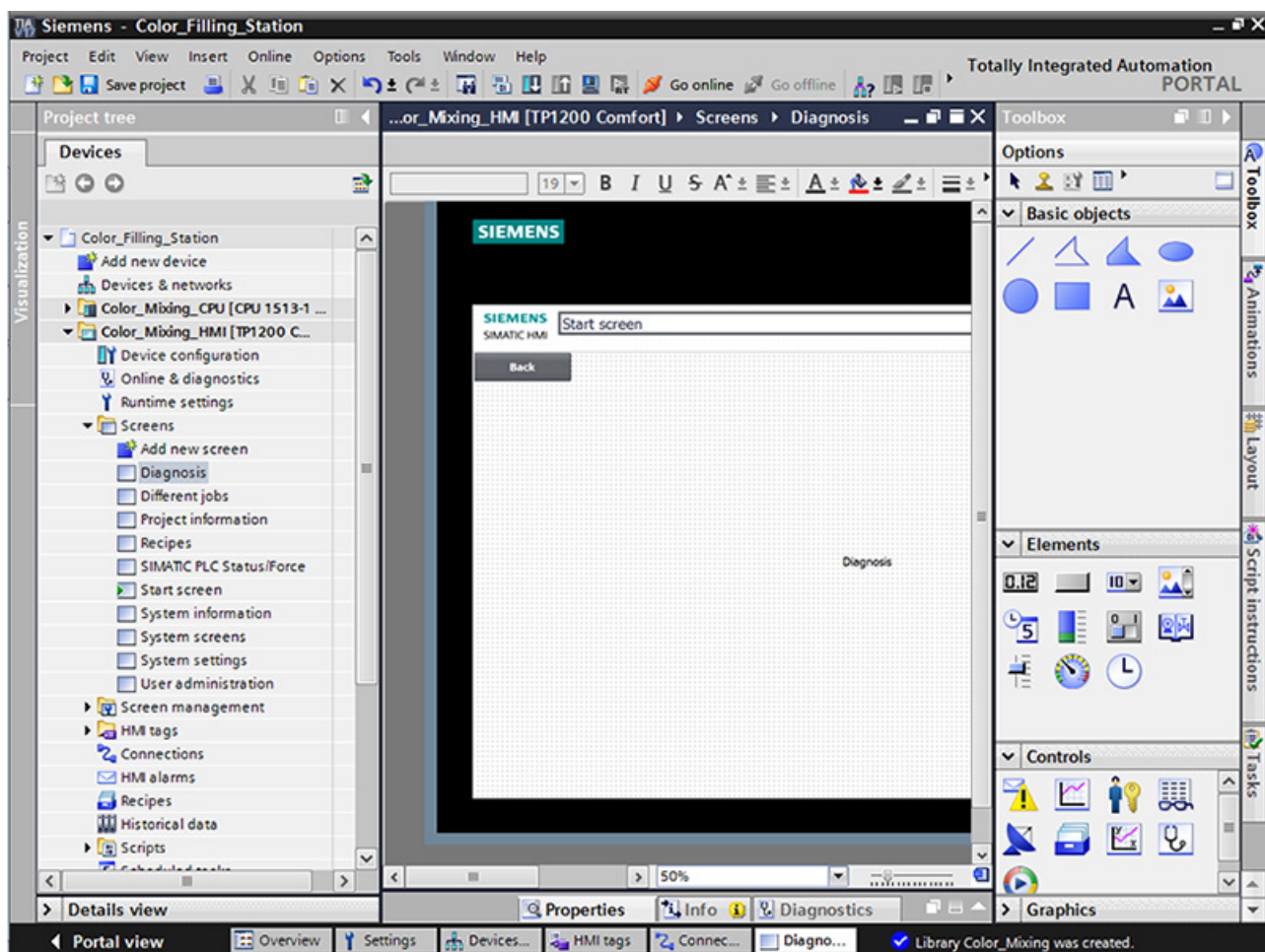
#### **Requirements**

- CPU has been created.
- The Inspector window is open.

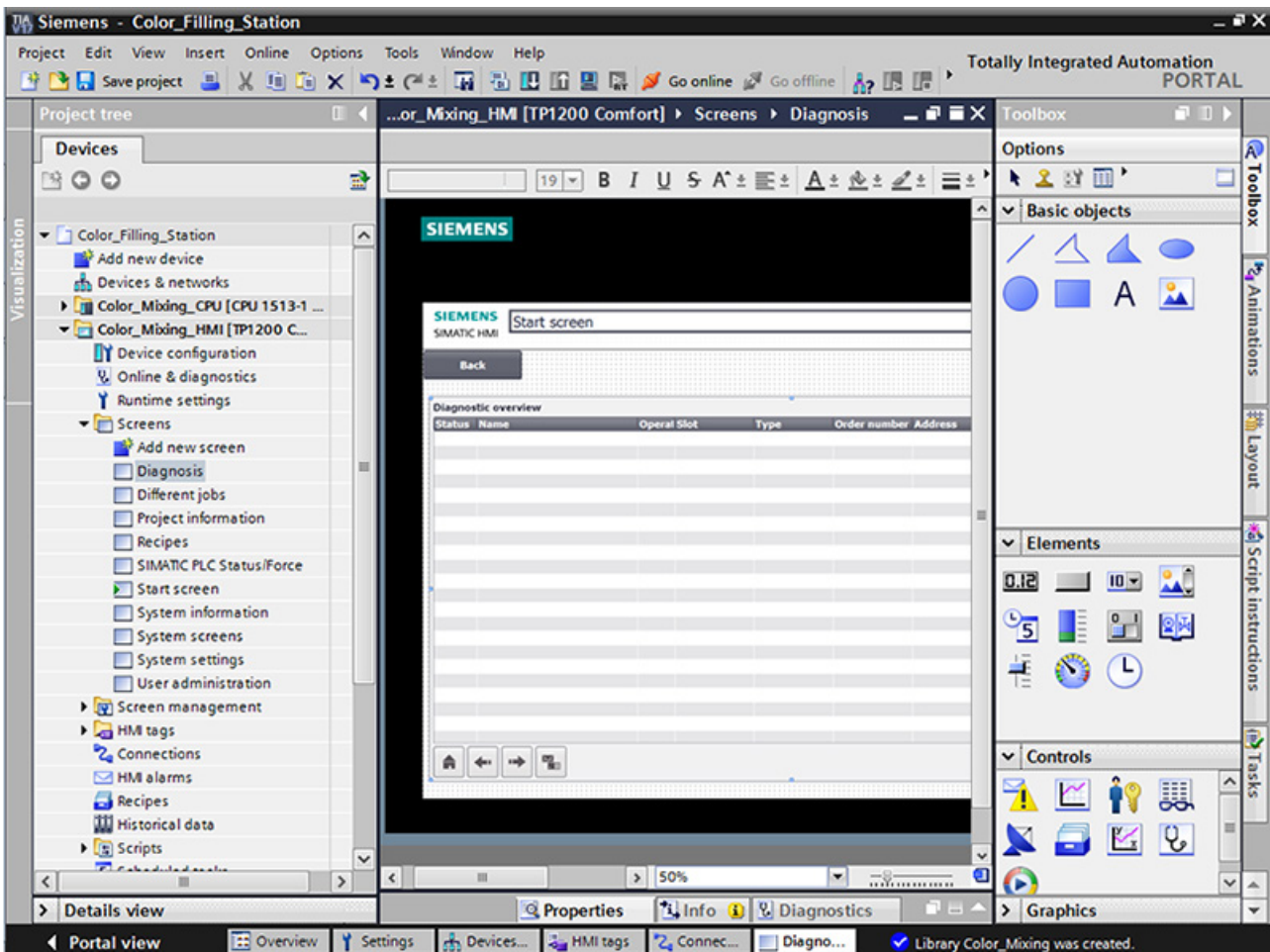


## Procedure

1. Double-click the "Diagnostics" HMI screen.



2. Double-click the "System diagnostics view" object in the "Tools" task card. The object is added to the screen.



3. Select "Properties > Properties > Columns > Devices/Detail view" in the Inspector window.
4. Enable the columns that you require in the device view for Runtime, for example, State, Name, Slot.
5. Enable the columns that you require in the detail view for Runtime, for example, State, Name, Higher level designation.
6. Select the columns that you require in the diagnostic buffer view, e.g. Status, Name, Rack.
7. If necessary, adapt the headers to the columns.
8. Enable "Properties > Properties > Layout > Column settings > Columns moveable" to move the columns in Runtime.
9. You can change the column headers under "Properties > Properties > Column headers", if necessary.

## Result

The system diagnostics view has been added to the "Diagnostics" screen.

Error messages for the entire plant are now displayed in the system diagnostics view in Runtime.

## 3.3.6 Simulating an HMI device

### 3.3.6.1 Simulation basics

#### Introduction

You can use the simulator to test the performance of your configuration on the configuration PC. This allows you to quickly locate any logical configuration errors before productive operation.

You can start the simulator as follows:

- In the shortcut menu of the HMI device or in a screen: "Start simulation"
- Menu command "Online > Simulation > [Start|With tag simulator|With script debugger]"
- Under "Visualization > Simulate device" in the portal view.

#### Requirement

The simulation/runtime component is installed on the configuration PC.

#### Field of application

You can use the simulator to test the following functions of the HMI system, for example:

- Checking limit levels and alarm outputs
- Consistency of interrupts
- Configured interrupt simulation
- Configured warnings
- Configured error messages
- Check of status displays

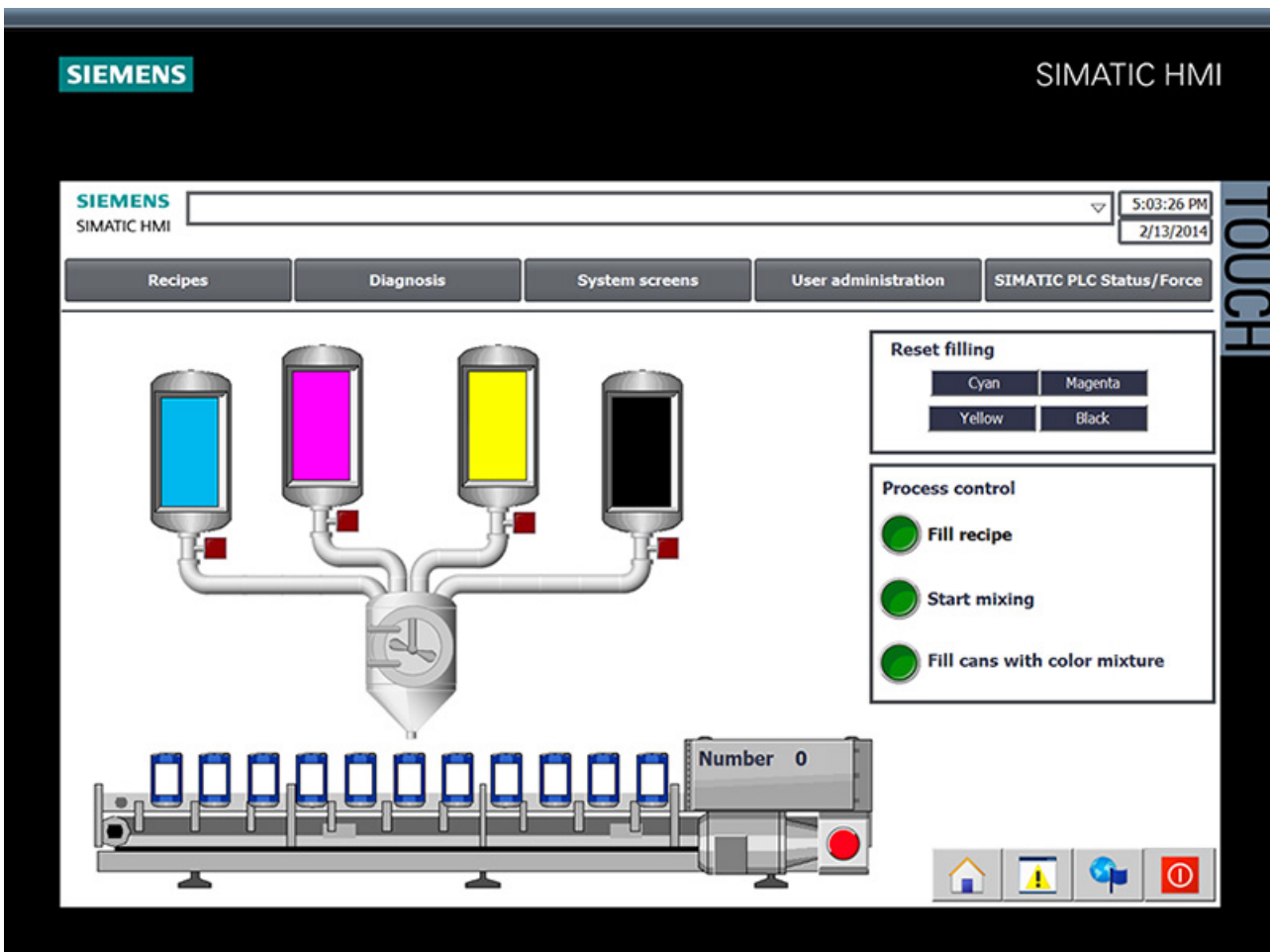
### 3.3.6.2 Operating the panel in simulation

#### Introduction

You simulate the HMI project on your computer.

#### Procedure

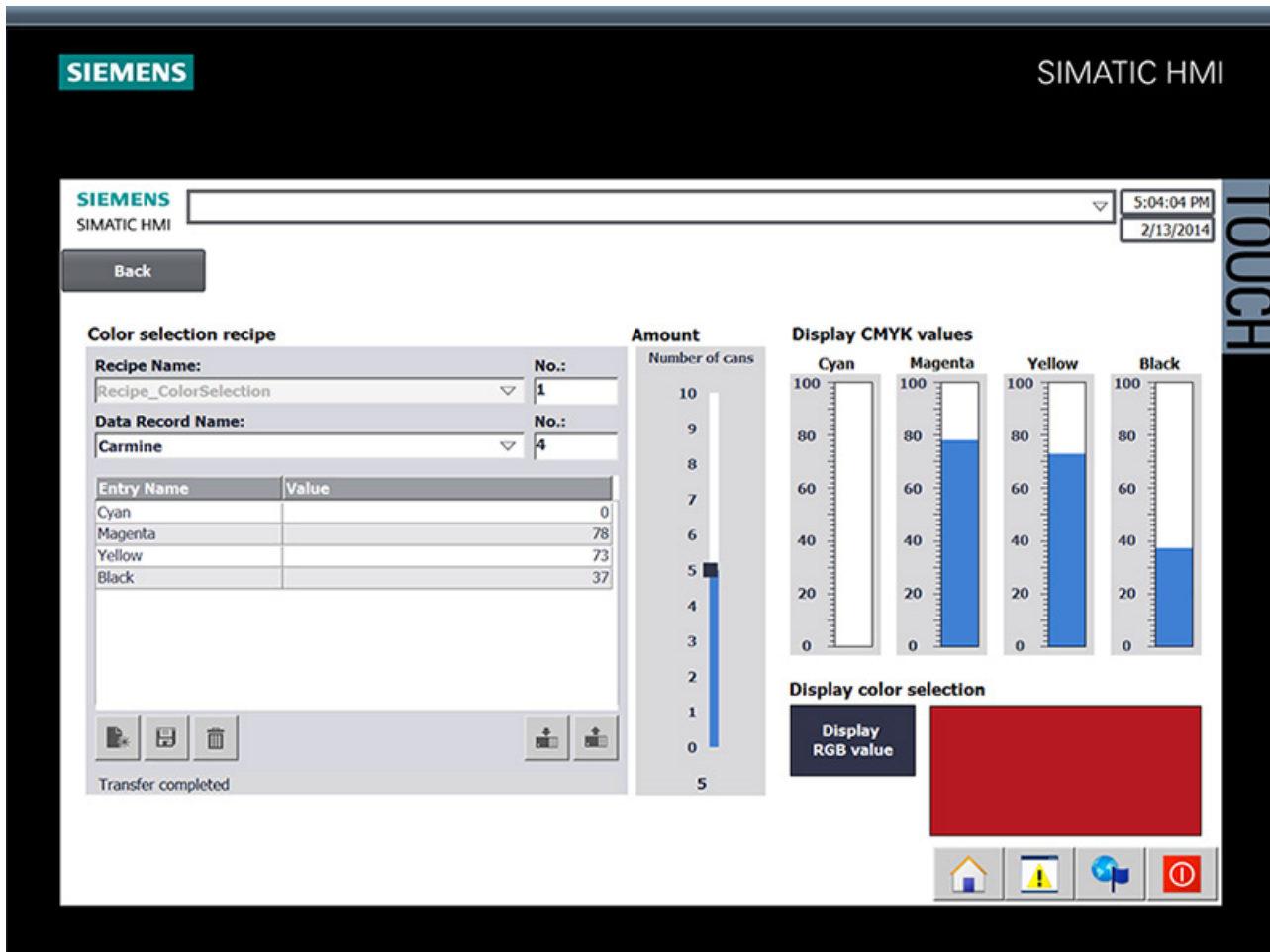
1. Start the simulation of the HMI device.



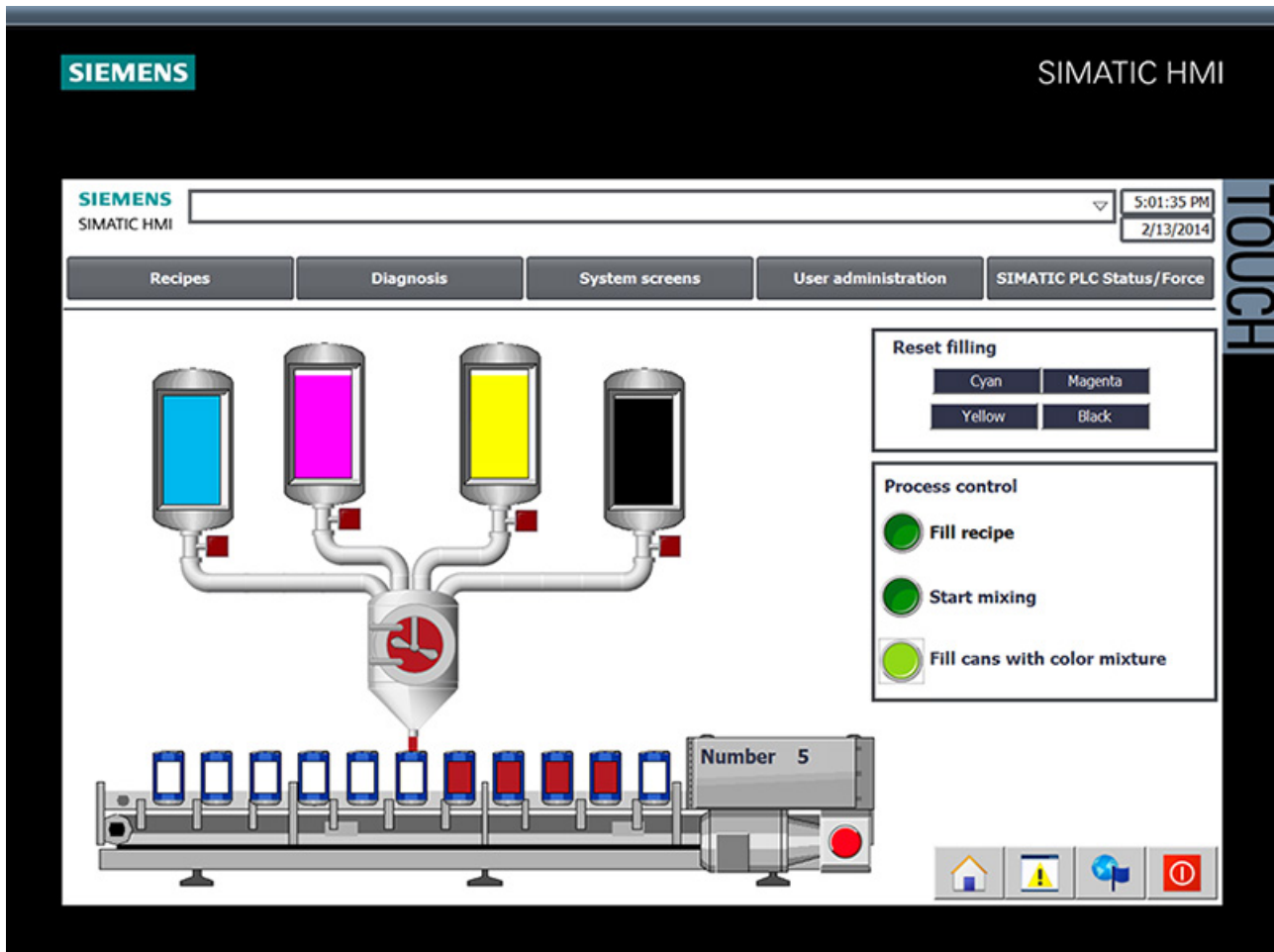
A connection to the CPU is established and the color mixing system is displayed in the simulation.

2. Open the "Recipes" screen and select a color.

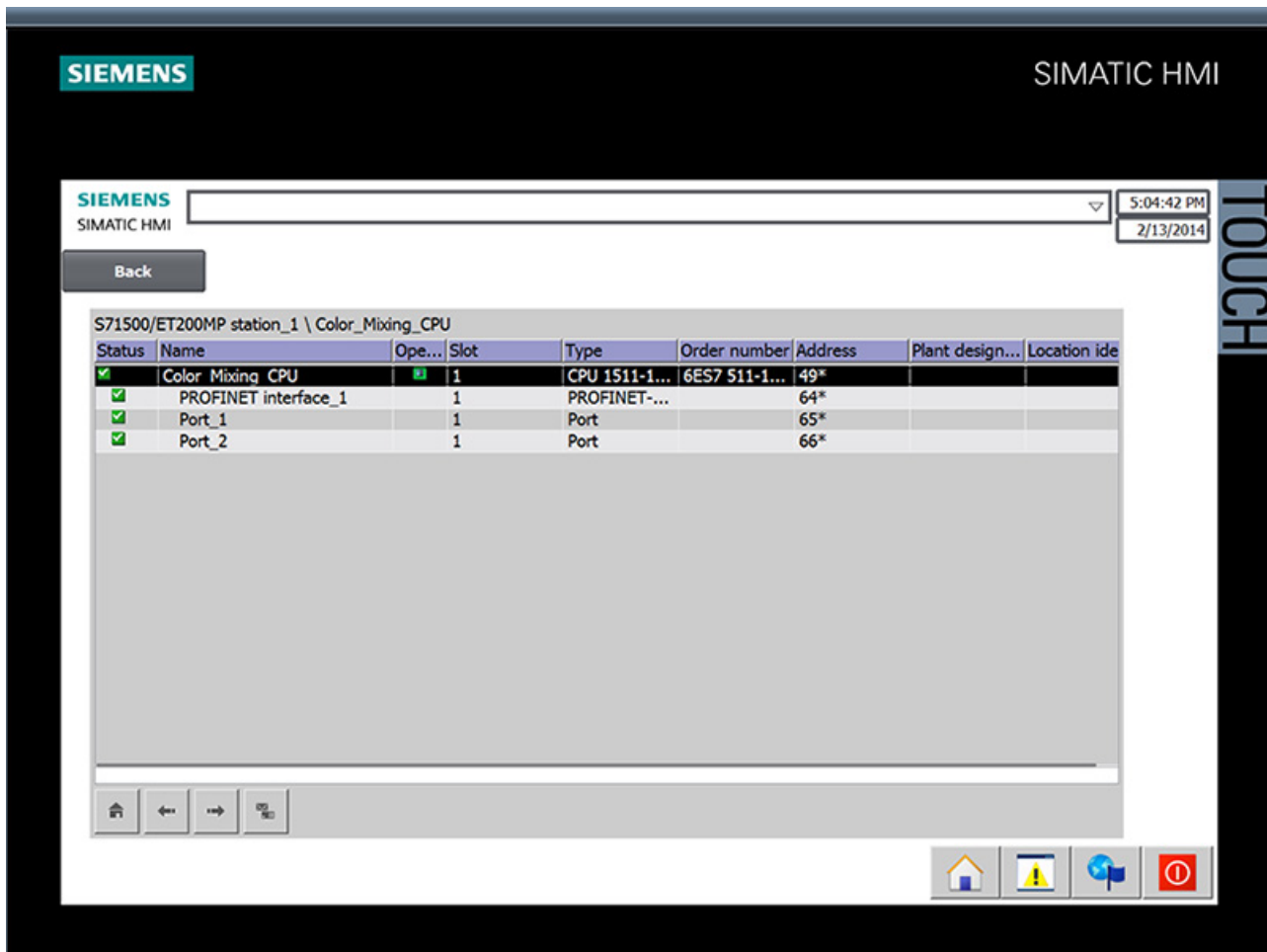
3. Specify the number of cans and view the selected color.



4. Go back to the start screen and start production.



5. You can query the current CPU status in the "Diagnostics" screen.





## 3.4 Loading the project into the programming device

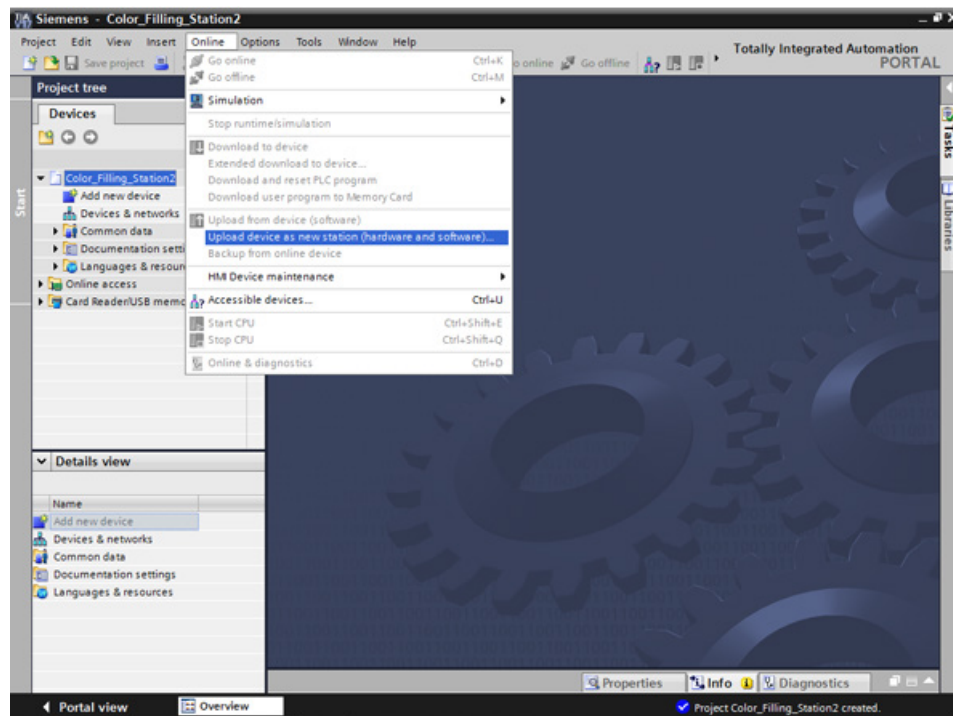
### 3.4.1 Load CPU to project

#### Introduction

You can create a new station including the actual values from the hardware configuration and the user program.

#### Procedure

1. Open the dialog for loading from the CPU.

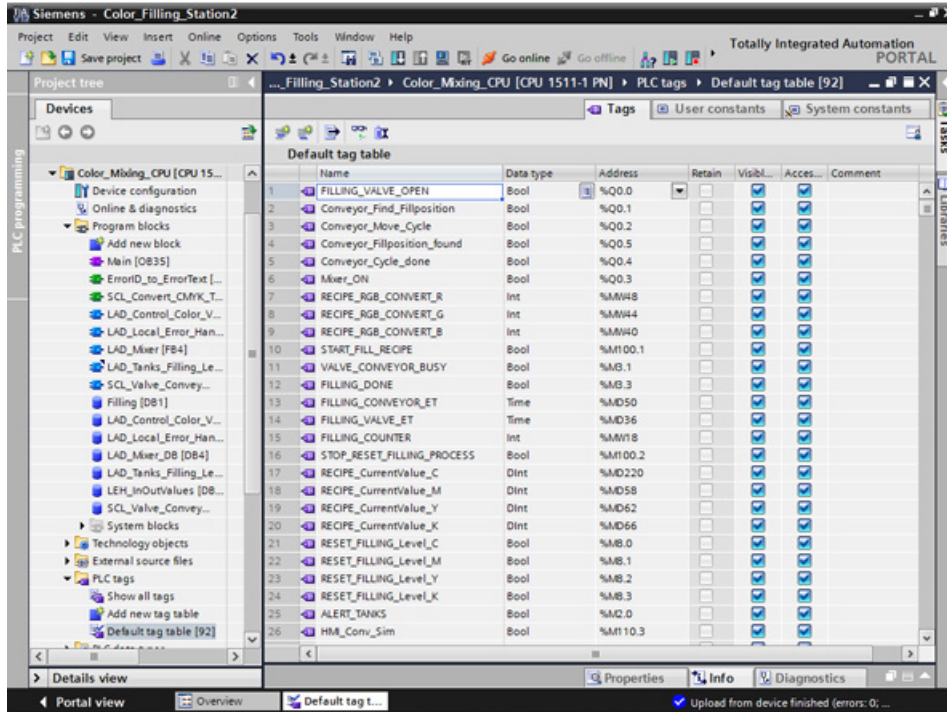


2. Select the interface with which the programming device is connected to the CPU. The search for accessible nodes starts automatically.
3. Load the CPU in the project.



## Result

The hardware and software configuration of the CPU are loaded into the project. The project now contains, for example, program blocks and tags.



## 3.5 Team engineering via Inter Project Engineering

### 3.5.1 Basics of "Inter Project Engineering"

#### Introduction

In this section, you will learn about the benefits of team engineering and how to create the required CPU data for an HMI project engineer. As an HMI project engineer, you will learn how to use this CPU data in your project.

#### Distributed configuration

You can use "Inter Project Engineering" to develop the user program and user interface in parallel at different locations. The HMI project engineer requires no CPU user program. There is no need for a STEP 7 installation.

Only tags, blocks, messages and address information of the CPU interfaces are ultimately relevant for the connection of an HMI device to a CPU. The programmer can conveniently export this data to an IPE file, which is imported into the project by the HMI developer. Updates are possible at any time by transferring a new IPE file.

The data are consistent after loading to the CPU and HMI device. The connections to the CPU created in the HMI configuration remain up-to-date.

---

#### Note

You can integrate the new Basic Panels 2nd Generation and Comfort Panels to STEP 7 projects as of V5.4 SP3 with an import into the TIA Portal.

---

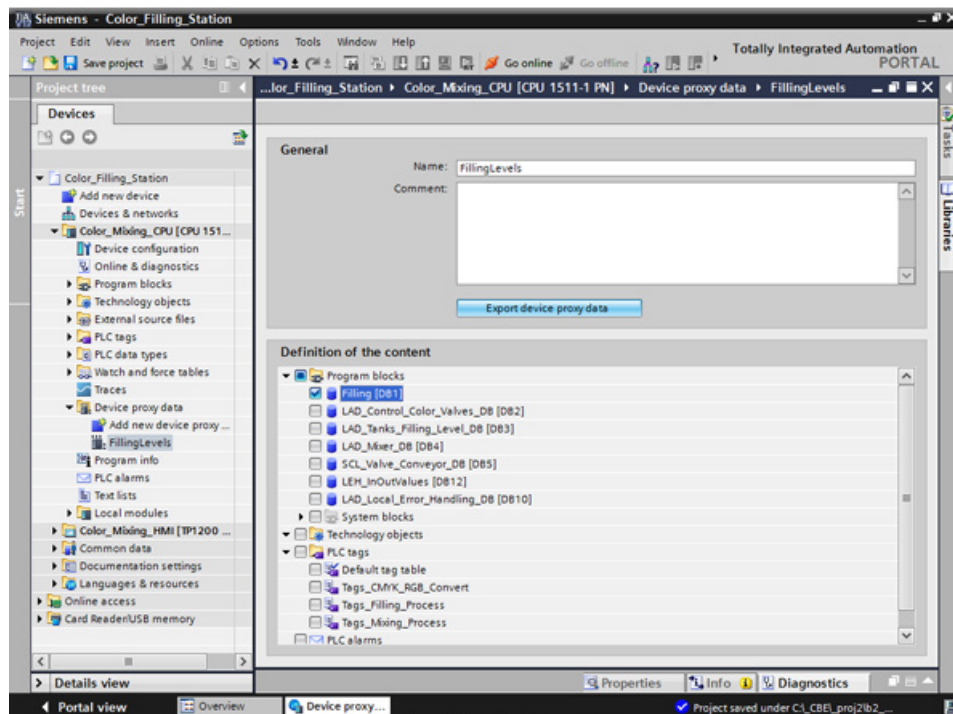
## 3.5.2 Creating an IPE file

### Introduction

You want to use a compact HMI device to display of fill levels directly at the paint mixing plant. You hire an engineering firm for the visualization and the provide the required CPU data as an IPE file.

### Procedure

1. Add new proxy data for the CPU.
2. Enter a name and select the required CPU data.



3. Export the proxy data.

### Result

The IPE file is created. You can send the IPE file, for example, as e-mail to the engineering firm.

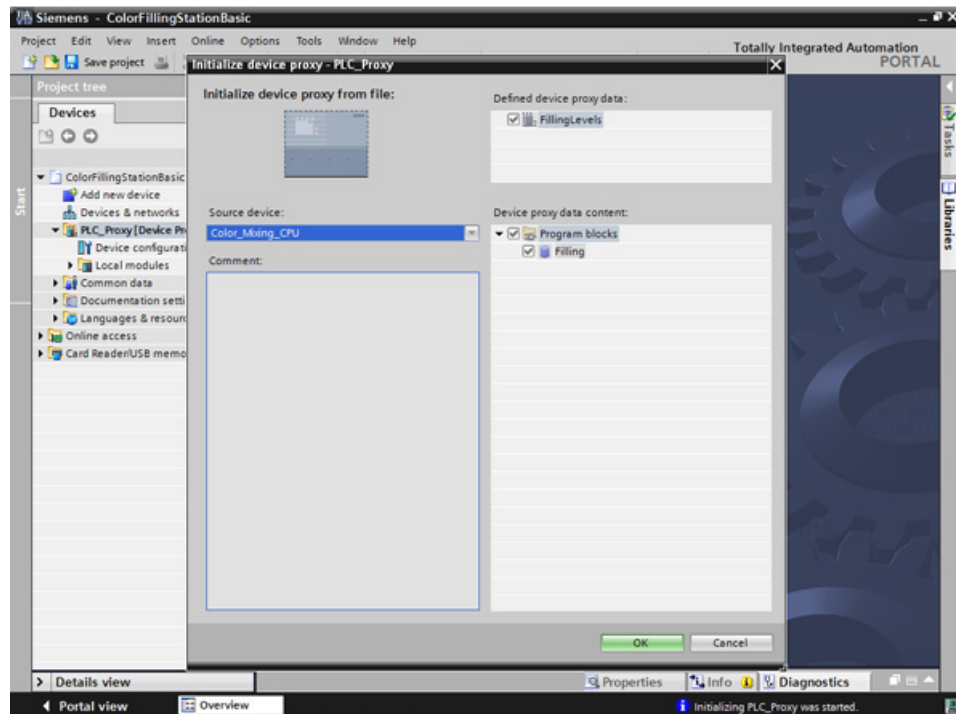
### 3.5.3 Importing an IPE file

#### Introduction

In the engineering office, the project engineer creates a device proxy in a new project and initializes it with the CPU data from the IPE file. The project engineer repeats the initialization for each update of the IPE file.

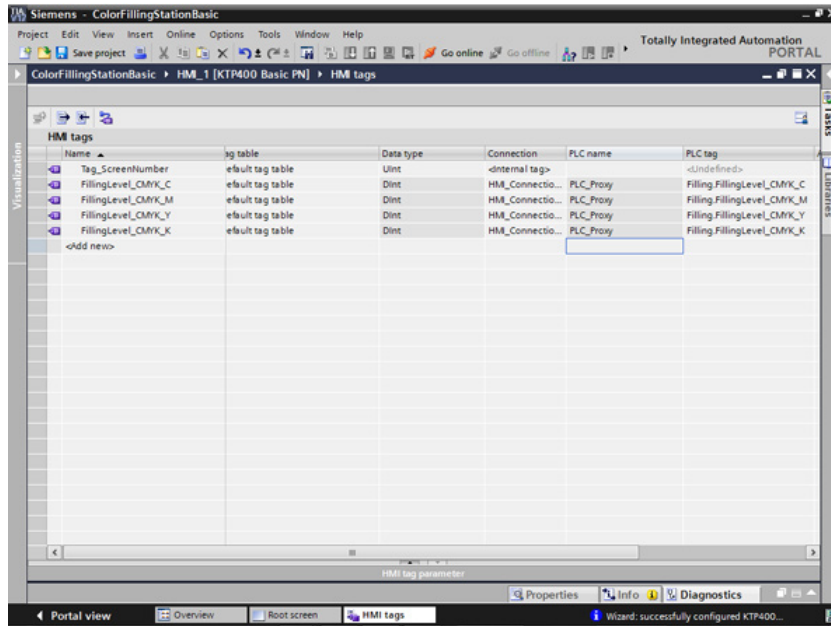
#### Procedure

1. Create the device proxy for the CPU in a new project and initialize it.

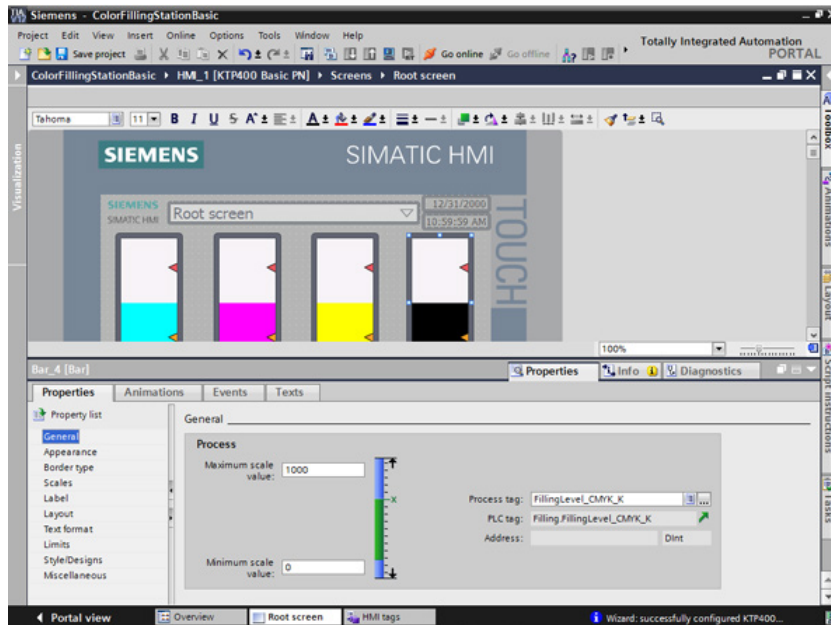


2. Use the Device Wizard to insert a Basic Panel.
3. Create the HMI tag for the fill level of the "Cyan" color and select the PLC tag.

4. Create the other HMI tags in the same way.



5. Configure a bar graph to display the fill level of the "Cyan" color.
6. Create a bar graph for the other fill levels in the same way.



7. Compile the project.

## Result

The project can now be loaded to the HMI device from the commissioning engineer. The communication with the CPU is up-to-date thanks to the CPU data from the IPE file.

# Security

## 4.1 Overview of the protective functions of the CPU

### Introduction

This chapter describes the following functions for protecting the S7-1500 automation system against unauthorized access:

- Access protection
- Know-how protection
- Copy protection
- Protection by locking the CPU

### Further measures for protecting the CPU

The following measures additionally increase the protection against unauthorized accesses to functions and data of the S7-1500 CPU from external sources and via the network:

- Deactivation of the Web server
- Deactivation of the time synchronization via an NTP Server
- Deactivation of the PUT/GET communication

When the Web server is used, you protect your S7-1500 automation system against unauthorized access by setting password-protected access rights for specific users in the user management.

## 4.2 Using the display to configure additional access protection

### Introduction

On the display of an S7-1500, you can block access to a password-protected CPU (local lock). The access lock is only in effect, when the operating mode switch is in the RUN position. The access lock applies independently of password protection, i.e. if someone accesses the CPU via a connected programming device and has entered the correct password, access to the CPU is still blocked.

The access block can be set separately for each access level on the display, so that, for example, read access is allowed locally, but write access is not allowed locally.

### Procedure

If an access level with a password is configured in STEP 7, access can be blocked using the display.

Proceed as follows to set the local access protection for an S7-1500 CPU on the display:

1. On the display, select Settings > Protection menu.
2. Confirm the selection using "OK", and configure for each access level, whether access at the RUN mode selector is allowed or not:

Allow: Access to the CPU is possible, provided the corresponding password in STEP 7 is entered.

Disabled in RUN: When the operating mode switch is in the RUN position, no more users with privileges for this access level can log in to the CPU, even if they know the password. In STOP mode, access is possible with password entry.

### Access protection for the display

A password can be configured for the display in STEP 7 in the properties of the CPU so that the local access protection is protected by a local password.

## 4.3 Know-how protection

You can use know-how protection to protect one or more blocks of the OB, FB, FC type and global data blocks in your program from unauthorized access. You can enter a password in order to restrict access to a block. The password protection prevents the block from being read or changed without authorization.

Without the password only the following data concerning the block can be read:

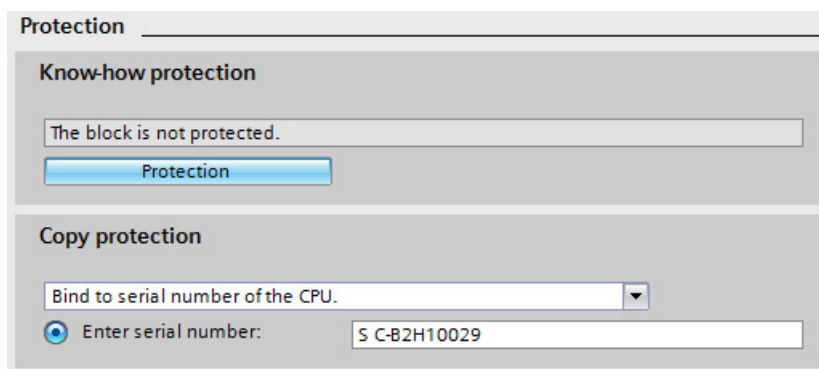
- Block title, comments and block properties
- Block parameters (INPUT, OUTPUT, IN, OUT, RETURN)
- Call structure of the program
- Global tags without information on the point of use

Further actions that can be carried out with a know-how protected block:

- Copying and deleting
- Calling in a program
- Online/offline comparison
- Load

### Setting up block know-how protection

1. Open the properties of the respective block.
2. Select the "Protection" option under "General".

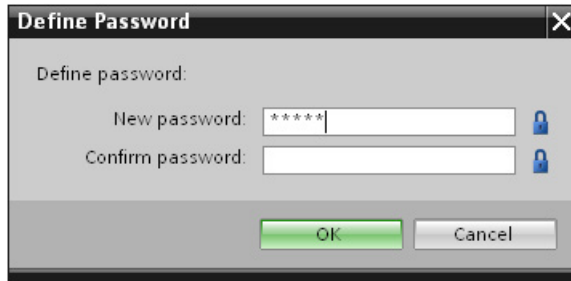


3. Click the "Protection" button to display the "Know-how protection" dialog.





4. Click the "Define" button to open the "Define password" dialog.



5. Enter the new password in the "New password" field. Enter the same password in the "Confirm password" field.
6. Click "OK" to confirm your entry.
7. Close the "Know-how protection" dialog by clicking "OK".

Result: The blocks selected will be know-how-protected. Know-how protected blocks are marked with a lock in the project tree. The password entered applies to all blocks selected.

### Opening know-how protected blocks

1. Double-click the block to open the "Access protection" dialog.
2. Enter the password for the know-how protected block.
3. Click "OK" to confirm your entry.

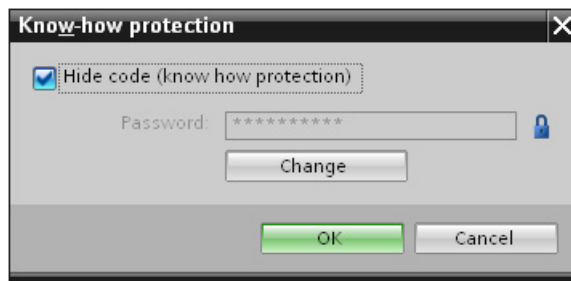
Result: The know-how-protected block will open.

Once you have opened the block, you can edit the program code and the block interface of the block till you close the block or the TIA Portal. The password must be entered again the next time the block is opened. If you close the "Access protection" dialog with "Cancel", the block will open but the block code will not be displayed and you will not be able to edit the block.

The know-how protection of the block is not removed if, for example, you copy the block or add it to a library. The copies will also be know-how-protected.

## Removing block know-how protection

1. Select the block from which you want to remove know-how protection. The protected block may not be open in the program editor.
2. In the "Edit" menu, select the "Know-how protection" command to open the "Know-how protection" dialog.
3. Deactivate the "Hide code (Know-how protection)" check box.



4. Enter the password.



5. Click "OK" to confirm your entry.

Result: Know-how protection will be removed from the block selected.

## 4.4 Copy protection

Copy protection allows you to bind the program or the blocks to a specific SIMATIC memory card or CPU. Through the linking of the serial number of a SIMATIC memory card or of a CPU the use of this program or of this block is only possible in combination with a specific SIMATIC memory card or CPU. With this function a program or block can be sent electronically (e.g. by e-mail) or by shipping a memory module.

When you set up such a copy protection for a block, also assign know-how-protection to this block. Without know-how protection, anyone can reset the copy protection. You must, however, set up copy protection first as the copy protection settings are read-only if the block is already know-how-protected.

### Setting up copy protection

1. Open the properties of the respective block.
2. Select the "Protection" option under "General".

3. In the "Copy protection" area, select either the "Bind to serial number of the CPU" entry or the "Bind to serial number of the memory card" entry from the drop-down list.

4. Enter the serial number of the CPU or the SIMATIC memory card.

5. You can now set up the know-how protection for the block in the "Know-how protection" area.

---

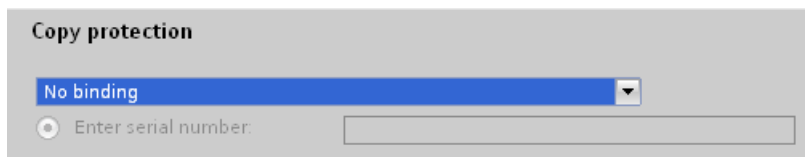
#### Note

If you download a copy protected block to a device that does not match the specified serial number, the entire download operation will be rejected. This means that blocks without copy protection will also not be downloaded.

---

## Removing copy protection

1. Remove any existing know-how protection.
2. Open the properties of the respective block.
3. Select the "Protection" option under "General".
4. In the "Copy protection" area, select the "No binding" entry from the drop-down list.

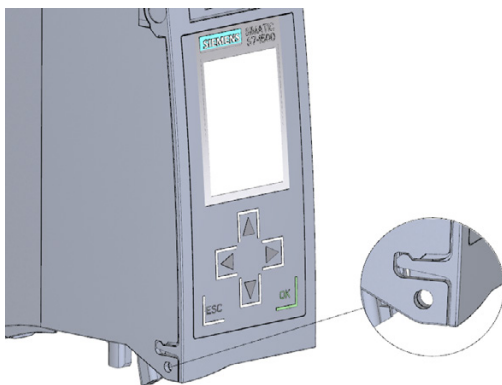


## 4.5 Protection by locking the CPU

Protect your CPU from unauthorized access using a sufficiently secured front cover.

Using the latch on the CPU cover, you have the following options:

- Affix a seal
- Secure the front cover with a lock (shackle diameter: 3 mm)



## 4.6 Configuring access protection for the CPU

### Introduction

The CPU offers four access levels, in order to limit access to specific functions.

By setting up the access levels and the passwords for a CPU, you limit the functions and memory areas that are accessible without entering a password. The individual access levels as well as the entry of their associated passwords are specified in the object properties of the CPU.

### Access levels of the CPU

Access levels	Access restrictions
Complete access (no protection)	The hardware configuration and the blocks can be read and changed by all users.
Read access	With this access level, read-only access to the hardware configuration and the blocks is possible without entering a password, which means you can download hardware configuration and blocks to the programming device. In addition, HMI access and access to diagnostics data is possible. Without entering the password, you cannot load any blocks or hardware configuration into the CPU. Additionally, the following are <b>not</b> possible without the password: Test functions which write, changing the operating mode (RUN/STOP), and firmware update (online).
HMI access	With this access level only HMI access and access to diagnostics data is possible without entering the password. Without entering the password, you can neither load blocks and hardware configuration into the CPU, nor load blocks and hardware configuration from the CPU into the programming device. Additionally, the following are <b>not</b> possible without the password: Test functions which write, changing the operating mode (RUN/STOP), and firmware update (online).
No access (complete protection)	When the CPU is completely protected, no read or write access to the hardware configuration and the blocks is possible. HMI access is also not possible. The server function for PUT/GET communication is disabled in this access level (cannot be changed). Authentication with the password will again provide you full access to the CPU.

Each access level allows unrestricted access to certain functions without entering a password, e.g. identification using the "Accessible devices" function.

The CPU's default setting is "No restriction" and "No password protection". In order to protect access to a CPU, you must edit the properties of the CPU and set up a password.

Communication between the CPUs (via the communication functions in the blocks) is not restricted by the protection level of the CPU, unless PUT/GET communication is deactivated.

Entry of the right password allows access to all the functions that are allowed in the corresponding level.

### Note

#### Configuring an access level does not replace know-how protection

Configuring access levels prevents unauthorized changes to the CPU, by restricting download privileges. However, blocks on the SIMATIC memory card are not write- or read-protected. Use know-how protection to protect the code of blocks on the SIMATIC memory card.

## Parameterizing the procedure at access levels

To configure the access levels of an S7-1500 CPU, follow these steps:

1. Open the properties of the S7-1500 CPU in the Inspector window.
2. Open the "Protection" entry in the area navigation.

A table with the possible access levels appears in the Inspector window.

Protection level	Access			Access permission	
	HMI	Read	Write	Password	Confirmation
<input checked="" type="radio"/> Full access (no protection)	✓	✓	✓		
<input type="radio"/> Read access	✓	✓			
<input type="radio"/> HMI access	✓				
<input type="radio"/> No access (complete protection)					

3. Activate the desired protection level in the first column of the table. The green checkmarks in the columns to the right of the respective access level show you which operations are still available without entering the password.
4. In the "Password" column, specify a password for the selected access level. In the "Confirmation" column, enter the selected password again to protect against incorrect entries.

Ensure that the password is sufficiently secure, in other words, that it does not follow a pattern that can be recognized by a machine!

You must enter a password in the first row ("Full access" access level). This enables unrestricted access to the CPU for those who know the password, regardless of the selected protection level.

5. Assign additional passwords as needed to other access levels if the selected access level allows you to do so.
6. Download the hardware configuration to the CPU, so that the access level will take effect.

### **Behavior of a password-protected CPU during operation**

The CPU protection takes effect after the settings are downloaded in the CPU.

Before an online function is executed, the necessary permission is checked and, if necessary, the user is prompted to enter a password. The functions protected by a password can only be executed by one programming device/PC at any one time. Another programming device/PC cannot log on.

Access authorization to the protected data is in effect for the duration of the online connection, or until the access authorization is manually rescinded with "Online > Delete access rights".

Access to a password-protected CPU in the RUN mode can be limited locally in the display so that access with a password is also not possible.

## 4.7 Configuring protection of the HMI connection

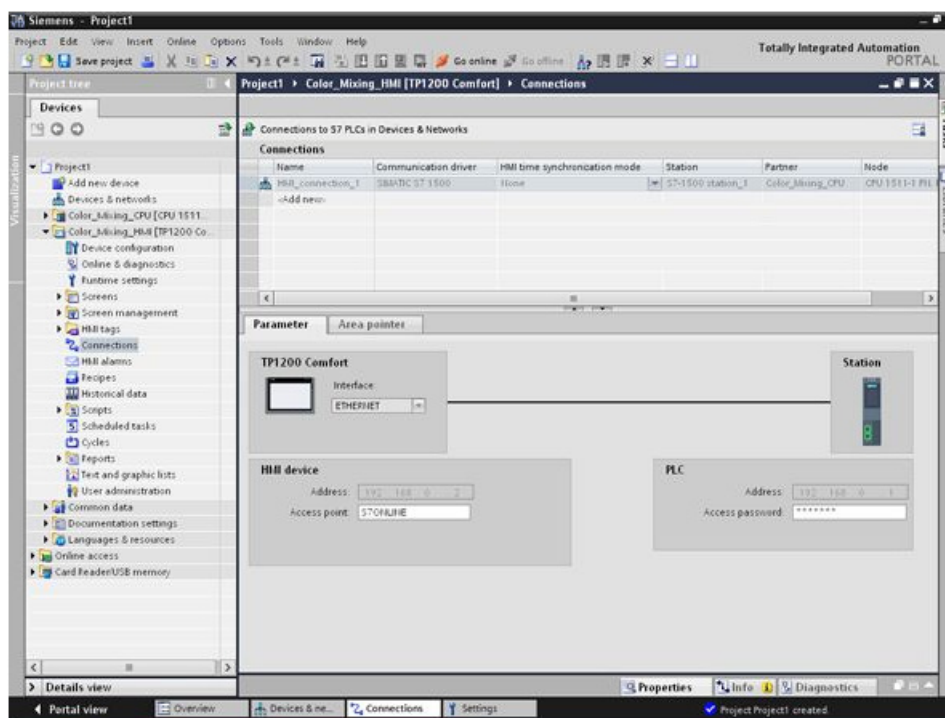
### Introduction

If the protection level "Complete protection" was set for the CPU, the HMI device can only access the CPU with the password stored there.

This function is only available with HMI devices from SIEMENS.

### Procedure

1. Open the "Connections" editor in the project tree.
2. Select the integrated connection.
3. Enter the password for the CPU in the "Password" area.



### Result

The HMI device can now communicate and exchange data with the CPU.



# S7-1500 Motion Control

## 5.1 Introduction

The Getting Started S7-1500 Motion Control guides you through the implementation of the motor-supported parts of the "Color mixing station" automation task. It will familiarize you with the basic functions of S7-1500 Motion Control. You configure the drive interfaces with general station description files (GSD), create technology objects and program these using instructions in your user program.

This Getting Started is based on the Getting Started S7-1500. A corresponding sample project is included.

The supplied sample project includes blocks and objects that have already been prepared. You can amend and edit these blocks and objects according to the specifications of this Getting Started for practice. You can also copy the prepared blocks and objects from the global library into the project and use them.

The Getting Started includes both comprehensive step-by-step instructions and the faster procedure of copying the objects from the library.

### Task

The agitator and the conveyor of a color mixing station are to be operated with S7-1500 Motion Control.

The "Color\_Filling\_Station" project is to be expanded by the axes "Conveyor" (conveyor belt) and "Mixer" (agitator) for this purpose. The basic control logic for the axes is already available in the user program. The axes are integrated into the existing user program by two interface blocks. These blocks are to execute the instructions for the respective axes and provide feedback to the user program.

## 5.2 Preparations

### 5.2.1 Requirements

#### Hardware requirements

You need the following hardware components to run the Getting Started:

- 1 × CPU 1511-1 PN (6ES7511-1AK02-0AB0)
- 1 × S7-1500 load current supply PM 70W 120/230VAC (6EP1332-4BA00)
- 1 × mounting rail (6ES7590-1AB60-0AA0)
- 1 × digital input module DI 16x24VDC SRC BA (6ES7521-1BH50-0AA0)
- 1 × digital output module DQ 16x24V DC/0.5A HF (6ES7522-1BH01-0AB0)
- 2 × front connectors (6ES7592-1AM00-0XB0)
- 1 × SIMATIC Memory Card with at least 4 MB (e.g. 6ES7954-8LC03-0AA0)
- 1 × Ethernet cable

You can configure and program the Getting Started without any additional hardware.

To commission the configured Motion Control-specific system parts, however, you need two drives in addition to the hardware requirements listed above. The drives must be configured and commissioned beforehand.

The drives must each support one of the following PROFIdrive frames:

- PROFIdrive frame 3 for positioning of the conveyor
- PROFIdrive frame 1 for speed control of the agitator

The following drives were selected for the Getting Started as an example:

- 1 × SINAMICS S120 CU310-2 PN V4.5 drive (6SL3 040-1LA01-0AA0)
- 1 × drive SINAMICS G120 CU240E-2 PN(-F) V4.5 (6SL3244-0BB13-1FA0)

#### Software requirements

You need the following software components to run the Getting Started:

- SIMATIC STEP 7 Professional V17
- SIMATIC WinCC Advanced V17 oder SIMATIC WinCC Professional V17

## Sample project

To run the Getting Started, you need the following project files which you can download as zip file from the Siemens website (<https://support.industry.siemens.com/cs/ww/en/view/78027451>), chapter "S7-1500 Motion Control" > "Project download":

- Sample project "Color\_Filling\_Station"
- Library "MotionLib\_Color\_Filling\_Station"

## See also

Link to the Getting Started contents overview (<https://support.industry.siemens.com/cs/ww/en/view/78027451>)

### 5.2.2 Basic procedure

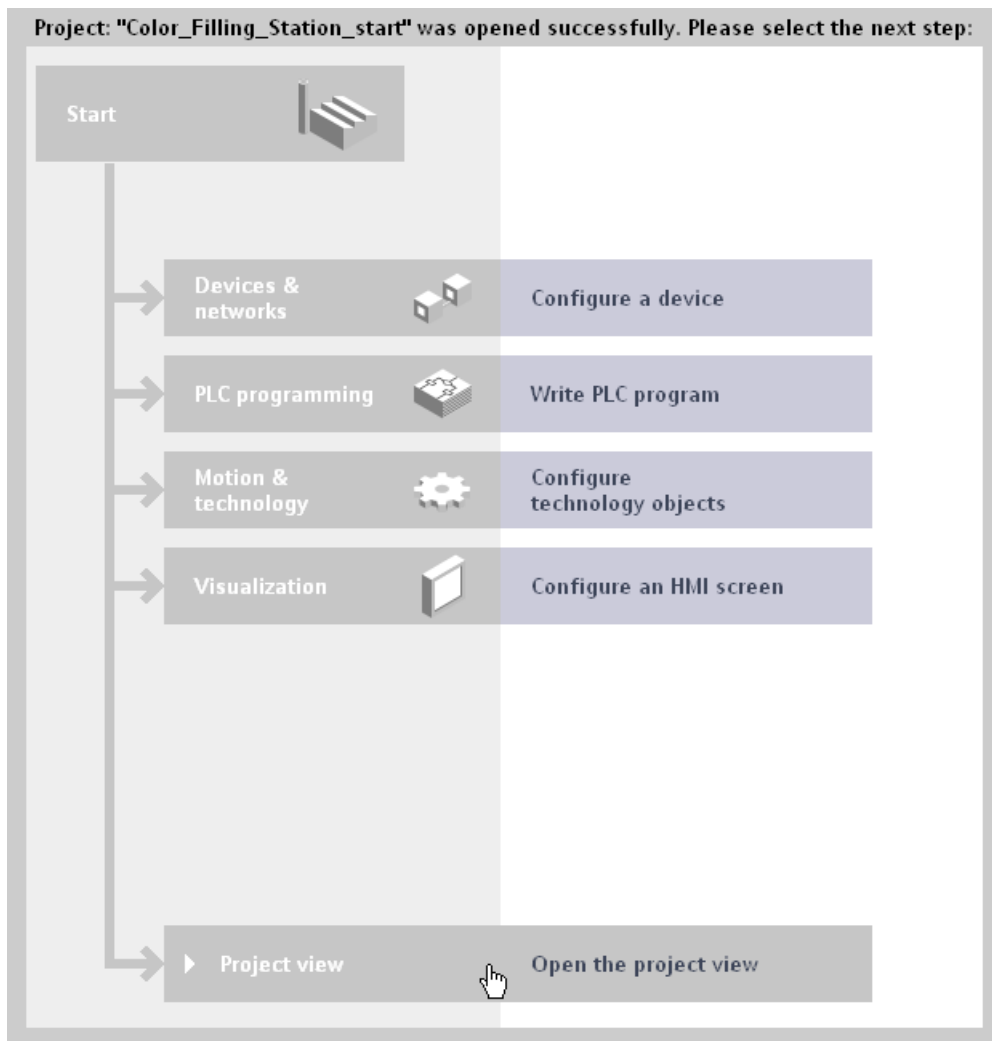
Starting with the sample project "Color\_Filling\_Station", configure the drives and technology objects with the TIA Portal. Then you create a STEP 7 user program that allows you to control the movement of the drives.

The procedure is divided into the following basic steps:

- Configuring drives
- Creating technology objects
- Programming the PLC
- Testing a function

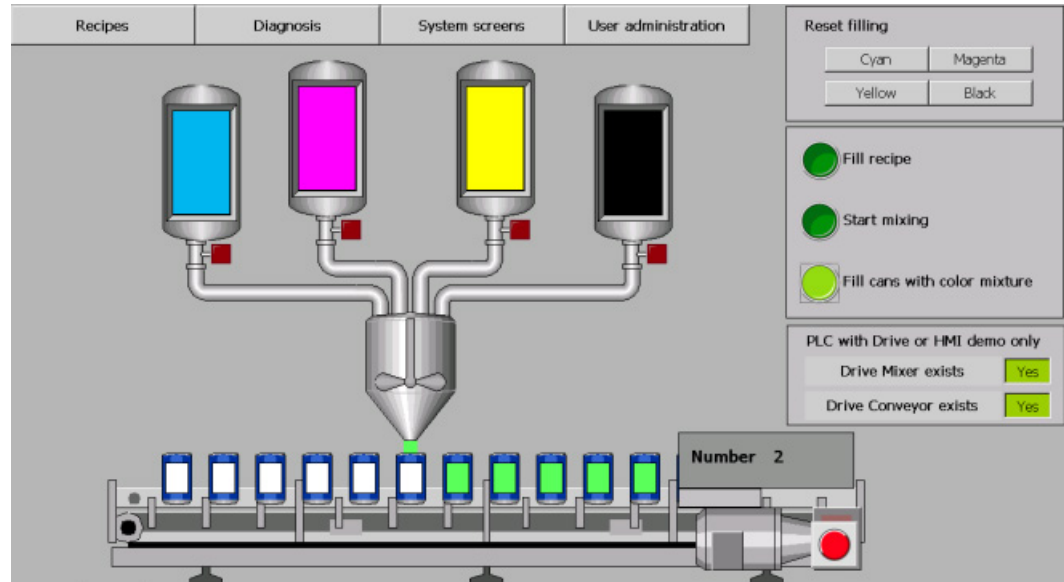
### 5.2.3 Opening a project

The included sample project "Color\_Filling\_Station" is the starting point for further action. Open the project in the project view.



## 5.2.4 Swapping the start screen

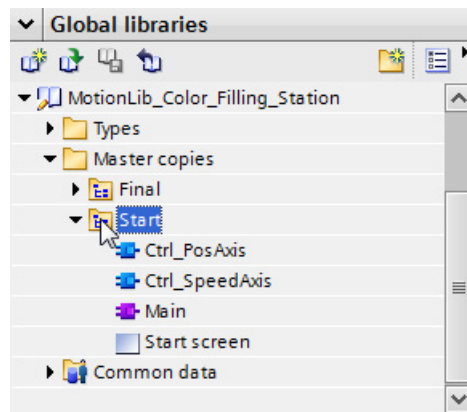
The visualization of the project has been expanded for the Getting Started S7-1500 Motion Control. To use this expansion, you simply need to swap the start screen of the visualization.



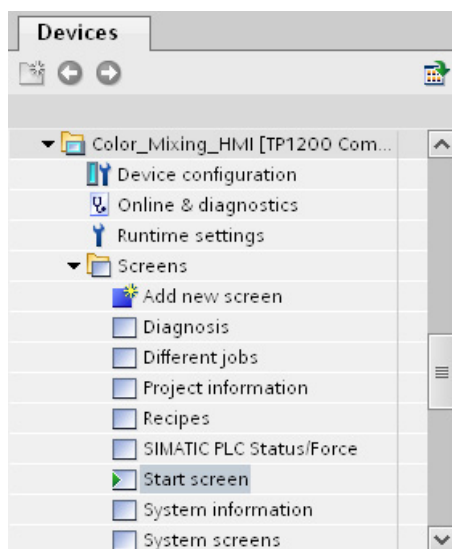
### Procedure

To swap the start screen of the visualization, follow these steps:

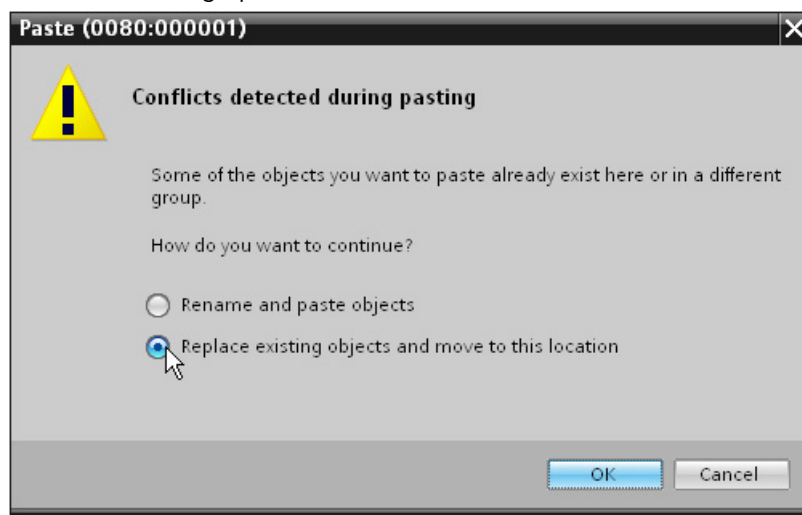
1. Open the global library "MotionLib\_Color\_Filling\_Station".
2. Open the folder "Master copies > Start" in the "MotionLib\_Color\_Filling\_Station" global library.



3. Drag the "Start screen" to the respective screen in the "Color\_Mixing\_HMI > Screens" folder.



The "Paste" dialog opens.

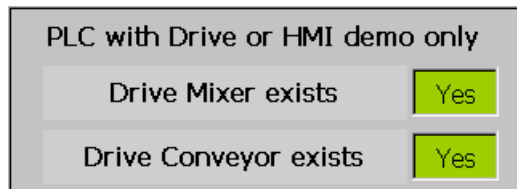


4. Select the option "Replace existing objects and move to this location".
5. Click "OK" to confirm.

## Expanding the start screen

The start screen has been expanded by the area "PLC with drive or HMI demo only" with the following buttons:

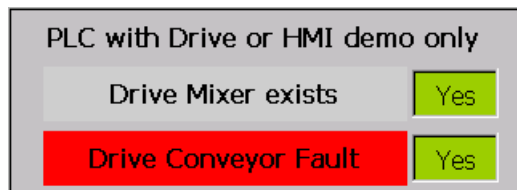
- **Drive Mixer exists: Yes/No**
- **Drive Conveyor exists: Yes/No**



If you want to test the color mixing station with the drives, switch the buttons to "Yes". The program is switched to use of the actual drives. The program uses the position of the drive for positioning the cans.

If you want to test the color mixing station without the drives, switch the buttons to "No". The program uses a calculated position for positioning the cans.

If an error occurs at the drive or the technology object, the display text is changed and highlighted in red.



You acknowledge errors with the following stop/reset button after the cause of the error has been removed:



### 5.2.5 Pasting blocks

The following program blocks have been prepared to simplify the step sequences and avoid repetitions:

- Organization block "Main"
- Function block "Ctrl\_PosAxis"
- Function block "Ctrl\_SpeedAxis"

You only have to copy these program blocks from the global library into the project.

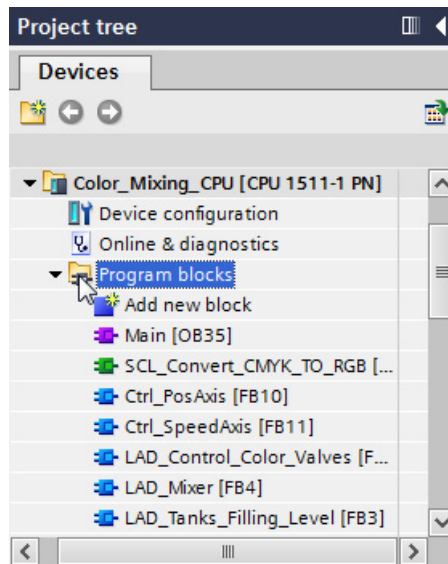
#### Procedure

To copy the program blocks into the project, follow these steps:

1. Open the "Color\_Mixing\_CPU > Program blocks" folder in the project tree.
2. Open the folder "Master copies > Start" in the "MotionLib\_Color\_Filling\_Station" global library.
3. Drag-and-drop the "Main" organization block to the corresponding block in the "Program blocks" folder.

The "Paste" dialog opens.

4. Select the option "Replace existing objects and move to this location".
5. Click "OK" to confirm.
6. Drag-and-drop the "Ctrl\_PosAxis" and "Ctrl\_SpeedAxis" function blocks to the "Program blocks" folder.



7. Compile the CPU software.



## 5.3 Configuring drives

### 5.3.1 Configuring SINAMICS S120

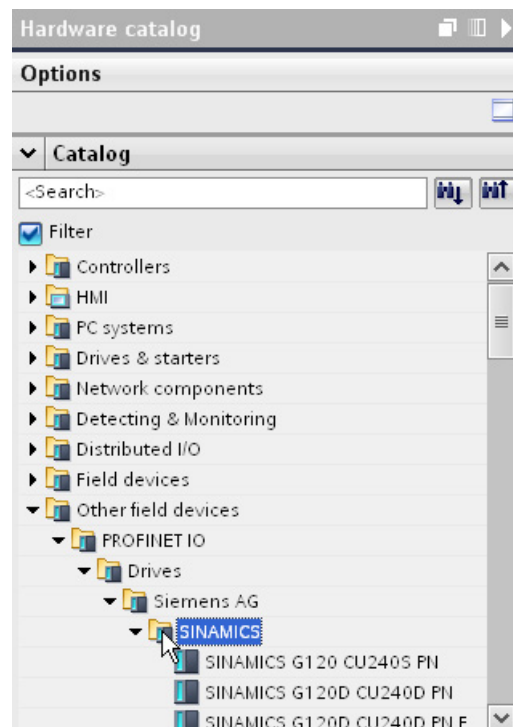
The SINAMICS S120 drive is used in the color mixing station for moving the conveyor. The drive is integrated via a general station description file (GSD).

In the following example, the IRT communication is also prepared with a send clock of 2 ms. IRT is isochronous real-time communication between the CPU and the drive.

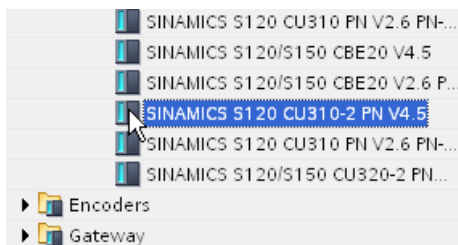
#### Creating a drive

To create the drive in the network view, follow these steps:

1. Open the network view of TIA Portal.
2. In the "Hardware catalog" task card, open the folder "Other field devices > PROFINET IO > Drives > Siemens AG > SINAMICS".

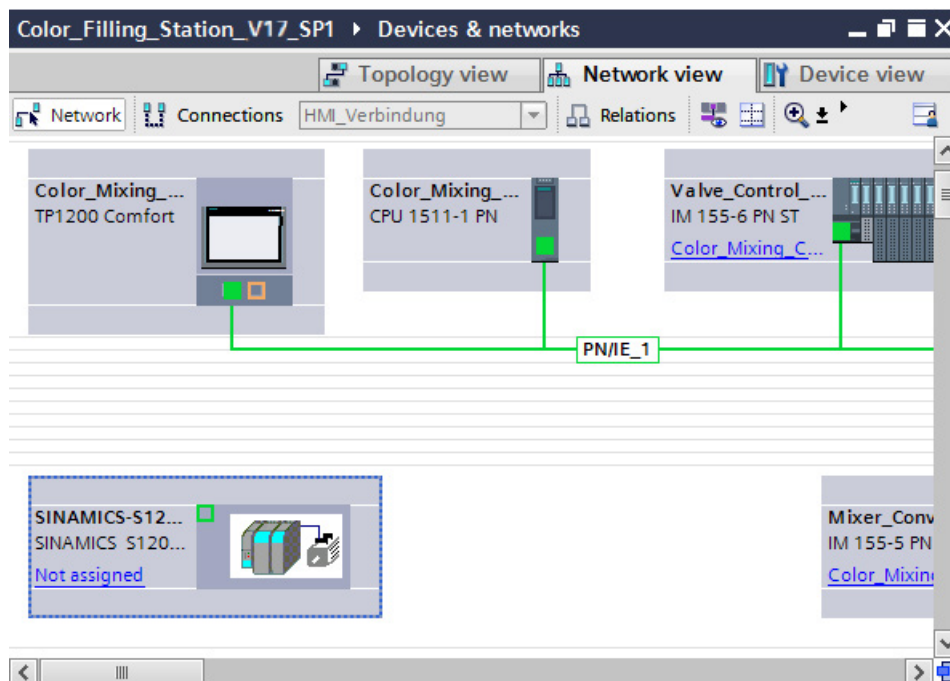


3. Select the drive "SINAMICS S120 CU310-2 PN V4.5".



4. Drag-and-drop the drive into the network view.

The drive is inserted in the network view.



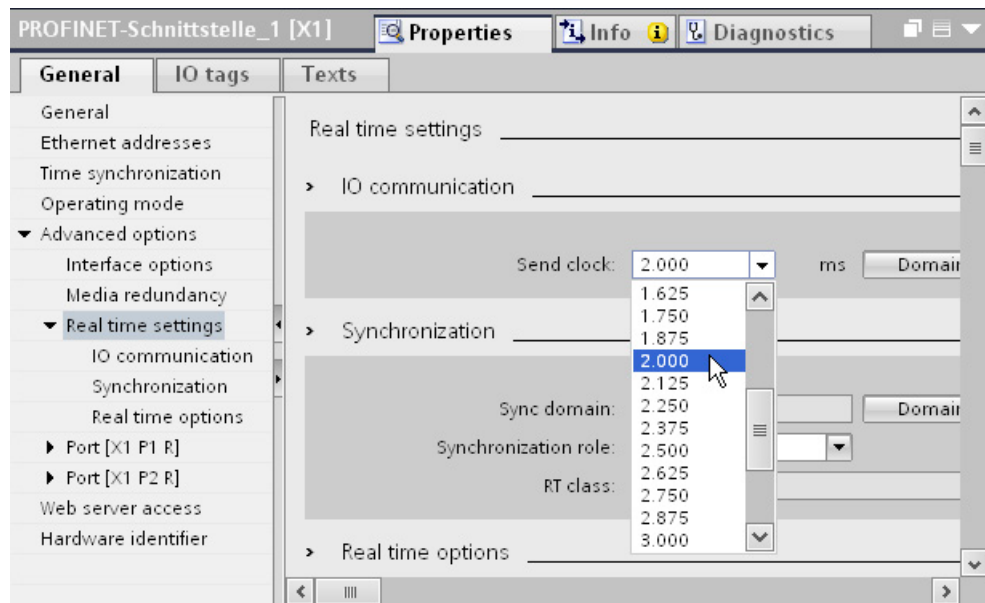
## Connecting the drive

To connect the drive to the CPU, follow these steps:

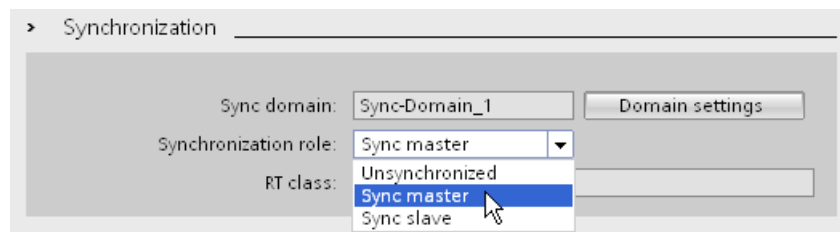
1. Select the PROFINET interface of the CPU.



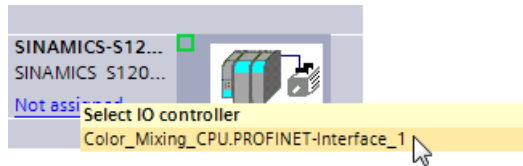
2. In the Inspector window, open the tab "Properties > General > Advanced options > Real-time settings".
3. Select the value 2 ms in the "IO communication" area of the "Send clock" drop-down list.



4. Select "Sync master" in the "Synchronization" area of the "Synchronization role" drop-down list.

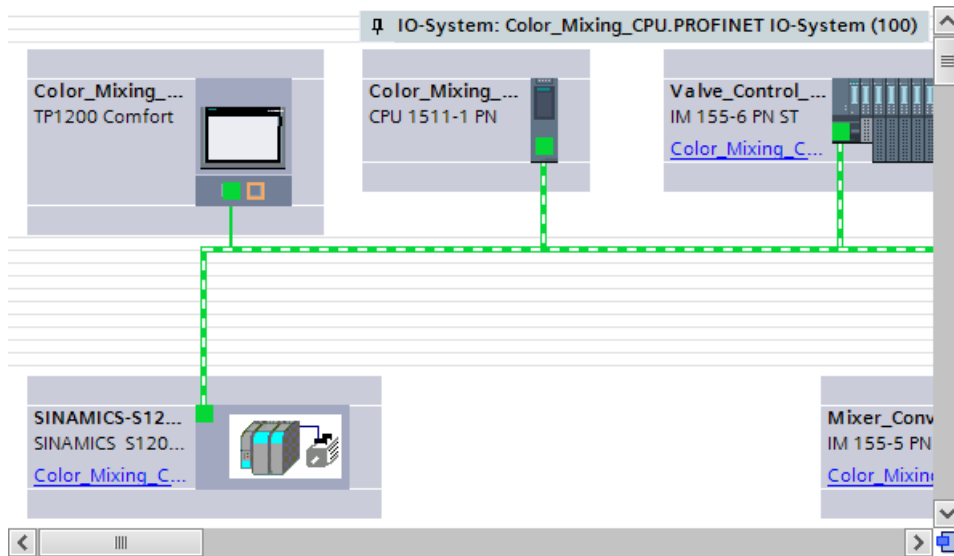


- 5. Assign the CPU as IO controller to the drive.

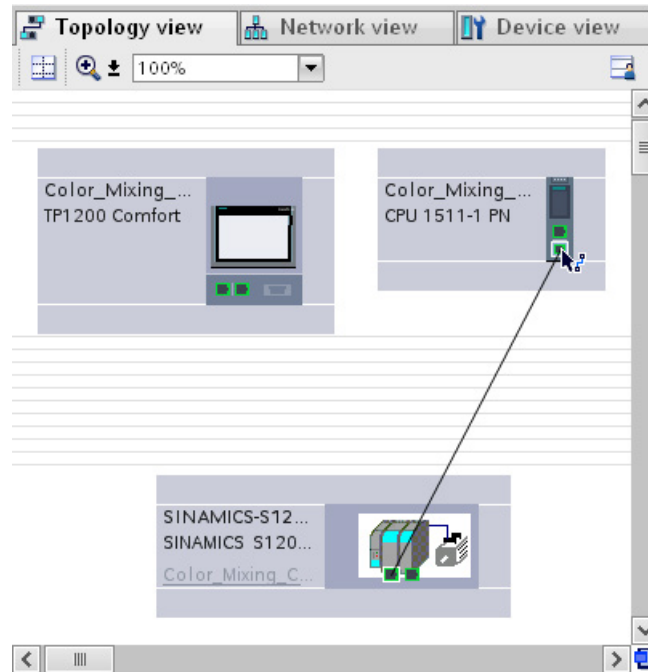


A PROFINET IO system is automatically created.

The drive is assigned to the PROFINET IO system as an IO device. The CPU is assigned to the drive as an IO controller.



6. Open the topology view.
7. Hold down the mouse button and draw a connection between a PROFINET port of the drive and a PROFINET port of the CPU.



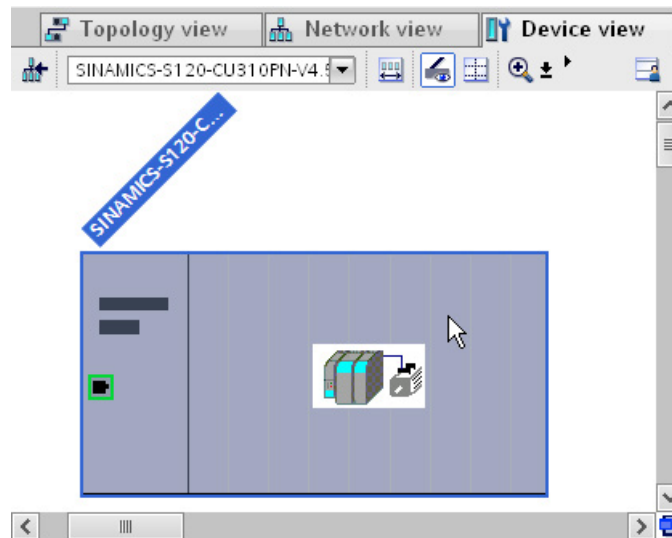
The ports of the PROFINET interfaces are interconnected. The port interconnection is necessary to use IRT when communicating with the drive. The bandwidth required for IRT communication can be more precisely determined with the port interconnection. This usually leads to better performance.

## Assigning a standard frame

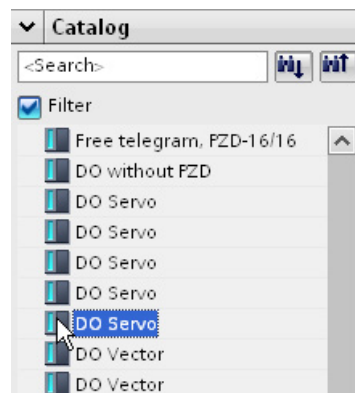
To assign a standard frame to the drive, proceed as follows:

1. Double-click on the SINAMICS S120 drive in the topology view.

The device view of the drive opens.



2. In the hardware catalog, select the drive object "DO Servo".



The TIA Portal automatically checks whether you can add the selected object. If you can assign the object to the drive, a blue marker is shown in the device view.

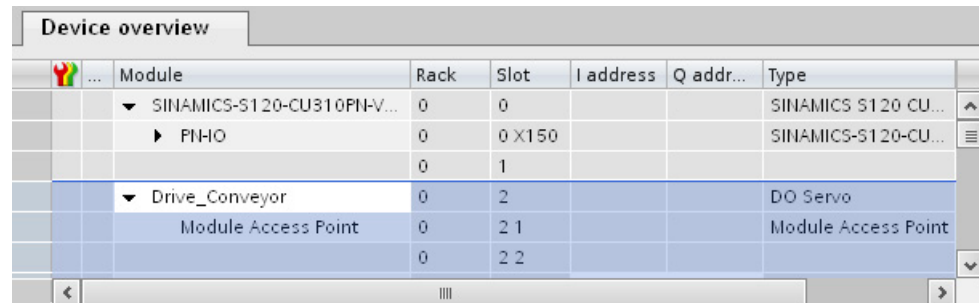
Device overview						
...	Module	Rack	Slot	I address	Q addr...	Type
	▼ SINAMICS-S120-CU310PN-V...	0	0			SINAMICS S120 CU...
	▶ PHHO	0	0 X150			SINAMICS-S120-CU...
		0	1			
		0	2			
		0	3			
		0	4			
		0	5			
		0	6			
		0	7			
		0	8			

3. Double-click the selected drive object.

The "DO Servo" drive object is assigned to the drive and appears in the device overview.

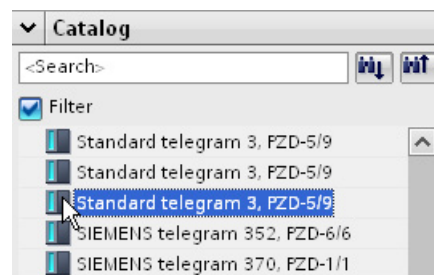
4. Assign "Drive\_Conveyor" as the name of the drive object in the drive overview.

A unique name makes it easier to assign the drive object to the technology object later.



Module	Rack	Slot	I address	Q addr...	Type
▼ SINAMICS-S120-CU310PIN-V...	0	0			SINAMICS S120 CU...
▶ PNH0	0	0 X150			SINAMICS-S120-CU...
	0	1			
▼ Drive_Conveyor	0	2			DO Servo
Module Access Point	0	2 1			Module Access Point
	0	2 2			

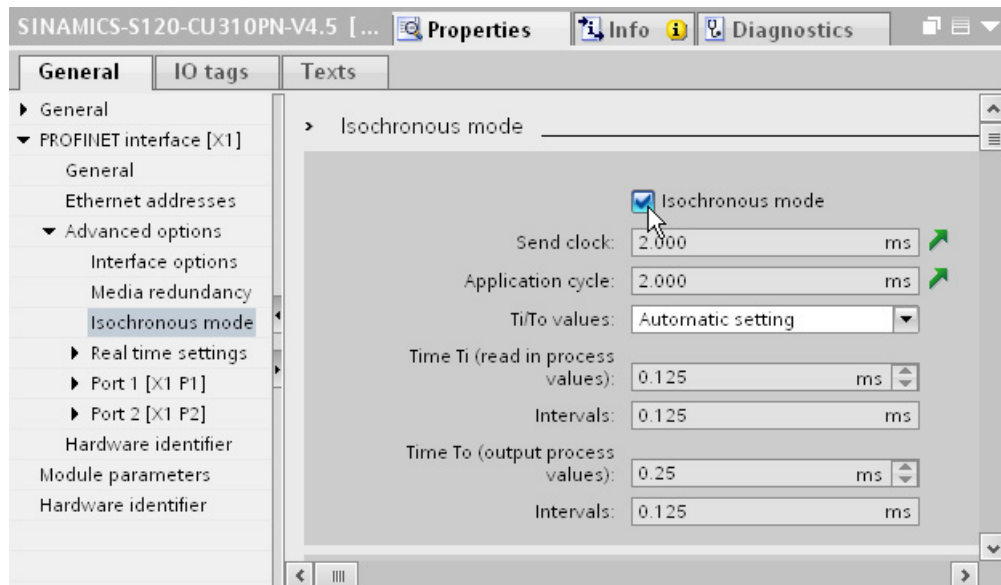
5. In the hardware catalog, double-click on the "Standard telegram 3, PZD-5/9" for PROFIdrive.



Frame 3 is assigned to the "DO Servo" drive object and appears in the device overview.

6. Select the drive in the device view.
7. In the Inspector window, open the tab "Properties > PROFINET Interface [X1] > Advanced options > Isochronous mode".

8. Select the "Isochronous mode" check box.



9. In the "Detail overview" list, select the "Isochronous mode" check box for the list item "GSD\_Device\_1/Standard frame 3".

**Detail overview**

Name	Slot/s...	Isochronous mode
Drive_Conveyor/Mo...	2/1	<input type="checkbox"/>
GSD-Geraet_1/Sta...	0/3	<input checked="" type="checkbox"/>

The bus times are automatically calculated and displayed with the settings for isochronous mode.



## 5.3.2 Configuring SINAMICS G120

The SINAMICS G120 drive is used in the color mixing station for the speed control of the agitator.

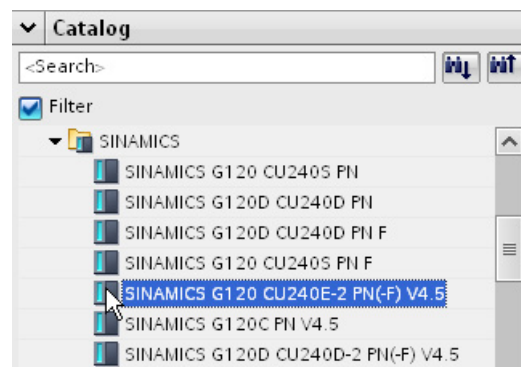
The drive is integrated via a general station description file (GSD). The SINAMICS G120 is also part of the commissioning software SINAMICS Startdrive.

In the following example, the drive is added to the existing PROFINET IO system. Simple cyclical communication between the CPU and the drive is sufficient for this application. IRT communication is not required.

### Creating a drive

To create the drive in the network view, follow these steps:

1. Open the network view of TIA Portal.
2. In the "Hardware catalog" task card, open the folder "Other field devices > PROFINET IO > Drives > Siemens > SINAMICS".
3. Select the drive "SINAMICS G120 CU240E-2 PN(-F) V4.5".

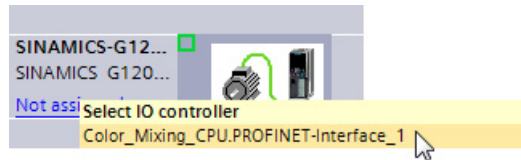


4. Drag-and-drop the drive into the network view.  
The drive is inserted in the network view.

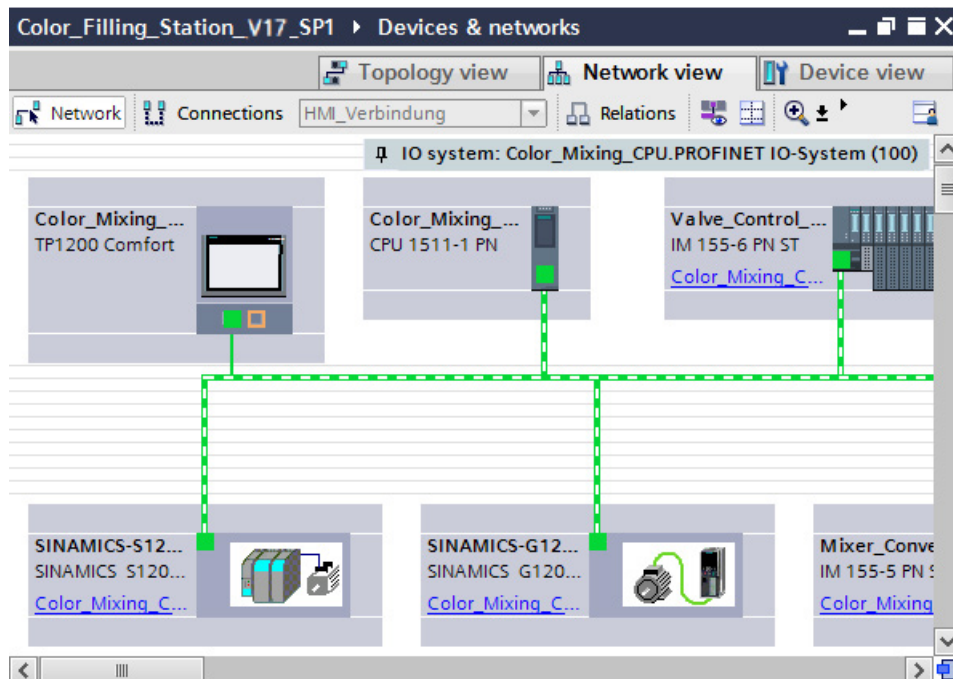
## Connecting the drive to the network

To connect the drive to the CPU, follow these steps:

1. Assign the CPU as IO controller to the drive.



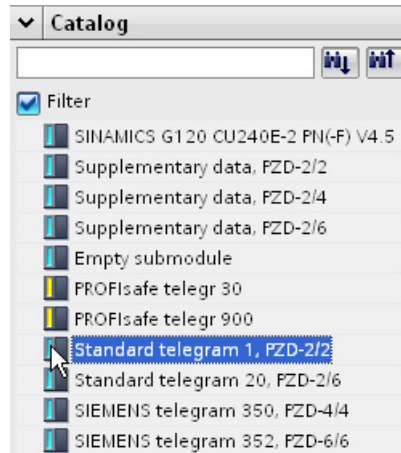
The drive is assigned to the PROFINET IO system as an IO device. The CPU is assigned to the drive as an IO controller.



## Inserting a standard frame

To assign a standard frame to the drive, proceed as follows:

1. Double-click on the SINAMICS G120 drive in the network view.  
The device view of the drive opens.
2. Assign "Drive\_Mixer" as the name of the drive object in the device overview.  
A unique name makes it easier to assign the drive object to the technology object later.
3. In the hardware catalog, double-click on the "Standard telegram 1, PZD-2/2".



Standard frame 1 is assigned to the drive and appears in the device overview.

Device overview						
...	Module	Rack	Slot	I address	Q addr...	Type
▼	SINAMICS-G120-CU240E-V4.5	0	0			SINAMICS G120 CU...
▶	PHO	0	0 X150			SINAMICS-G120-CU...
▼	Drive_Mixer	0	1			Drive
	Module Access Point	0	1 1			Module Access Point
		0	1 2			
	Standard Telegramm 1, ...	0	1 3	20...23	12...15	Standard telegram...
		0	1 4			

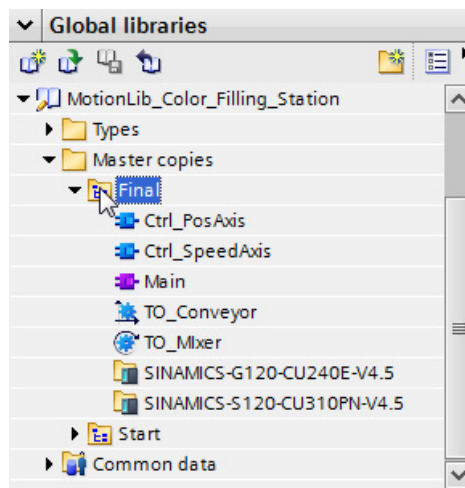
### 5.3.3 Using prepared drives

You can shorten the described procedure by inserting the drives from the global library into the project by means of the prepared device description files and connecting them accordingly.

#### Inserting drives

To insert the drives from the global library into the project, follow these steps:

1. Open the network view of TIA Portal.
2. Open the folder "Master copies > Final" in the "MotionLib\_Color\_Filling\_Station" global library.



3. Drag-and-drop the drives "SINAMICS-S120-CU310PN-V4.5" and "SINAMICS-G120-CU240E-V4.5" into the network view.

## Connecting drives

To connect the drives to the CPU, follow these steps:

1. Select the PROFINET interface of the CPU.
2. In the Inspector window, open the tab "Properties > General > Advanced options > Real-time settings".
3. Select the value 2 ms in the "IO communication" area of the "Send clock" drop-down list.
4. Select "Sync master" from the "Synchronization role" drop-down list.
5. Assign the CPU as IO-Controller to the drives.
6. Open the topology view.
7. Hold down the mouse button and draw a connection between a PROFINET port of the SINAMICS S120 drive and a PROFINET-Port of the CPU.
8. Select the drive in the topology view.
9. In the Inspector window, open the tab "Properties > PROFINET Interface [X1] > Advanced options > Isochronous mode".
10. Select the "Isochronous mode" check box.
11. In the "Detail overview" list, select the "Isochronous mode" check box for the list item "GSD\_Device\_1/Standard frame 3".

## 5.4 Creating technology objects

### 5.4.1 Configuring the positioning axis

The positioning axis technology object enables the position-controlled positioning of a drive. You can assign positioning jobs for the conveyor with Motion Control instructions in your user program.

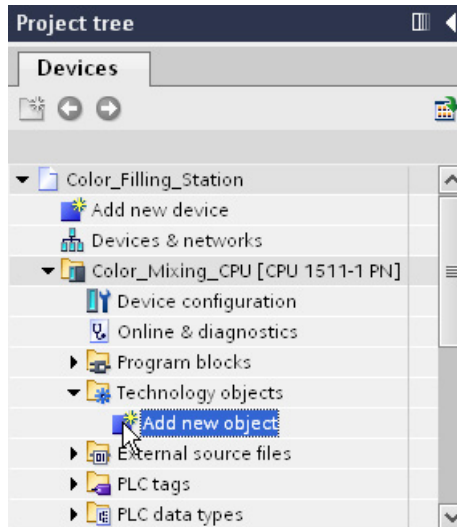
In the following example, you create a positioning axis technology object and assign it to the previously configured SINAMICS S120 drive. You assign the drive interface with PROFIdrive frame 3. The properties of the drive interface are set automatically with this step.

The organization blocks "MC-Servo" and "MC-Interpolator" are automatically created in the project when you create the technology object.

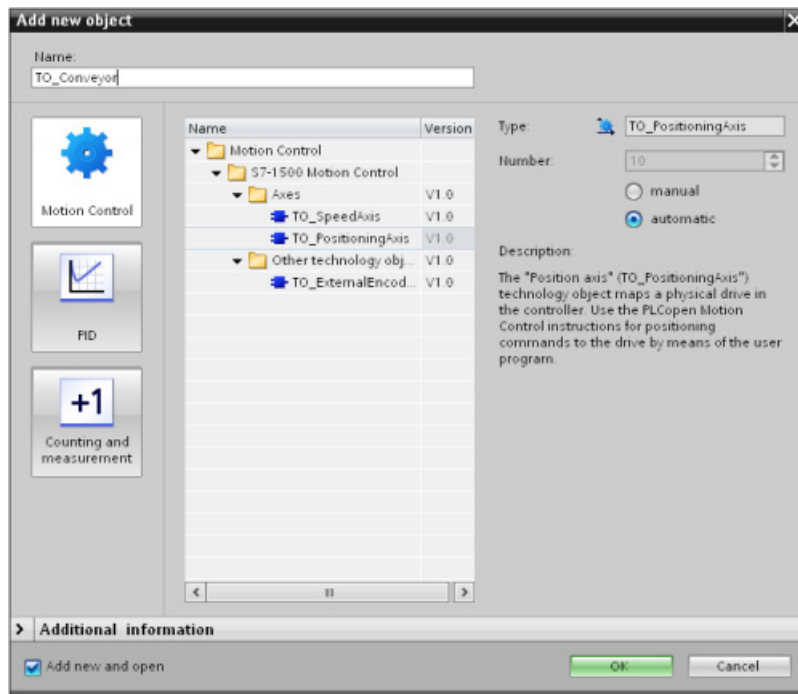
## Creating a positioning axis

To create the positioning axis, follow these steps:

1. Open the "Color\_Mixing\_CPU > Technology objects" folder in the project tree.
2. Double-click the "Add new object" command.



The "Add new object" dialog box opens.



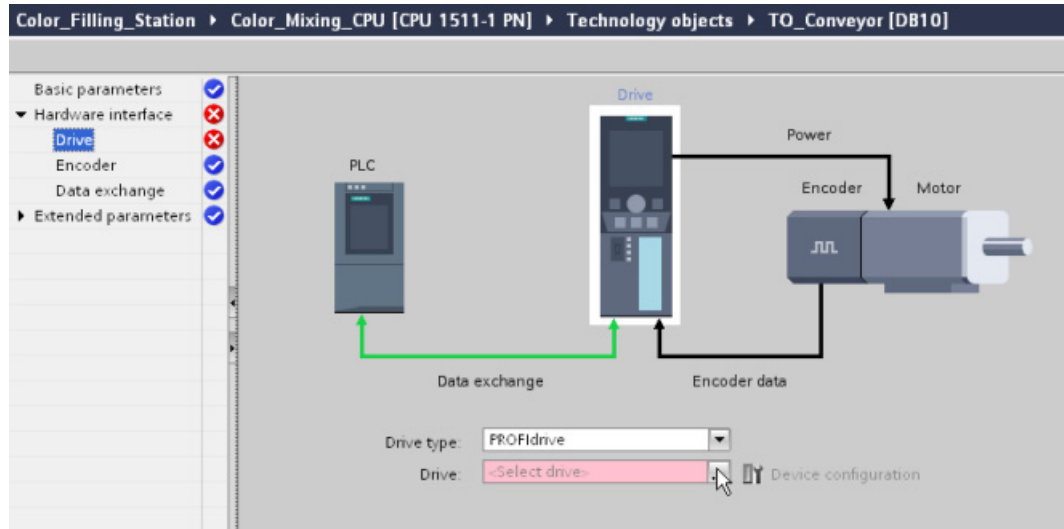
3. Select the "TO\_PositioningAxis" object.
4. Enter "TO\_Conveyor" as the name of the technology object.
5. Confirm your entry with "OK".

The technology object is created and opened in the project view.

## Assigning a drive

To assign a drive to the positioning axis and configure the necessary parameters, proceed as follows:

1. Open the configuration window "Hardware interface > Drive".



2. In the "Drive" list, select the entry "Drive\_Conveyor" of the PROFINET drive SINAMICS S120.



The positioning axis is connected to the drive. The process image of the drive is assigned to the process image partition "TPA OB Servo".

3. Open the configuration window "Hardware interface > Data exchange".
4. Check the settings for the data exchange. If necessary, adapt the settings to match those of the drive and the encoder.



## Configuring Modulo

To configure the Modulo settings, follow these steps:

1. Open the configuration window "Basic parameters".
2. Select the "Enable Modulo" check box.
3. Enter the value 0.0 mm in the "Start value" box.
4. Enter the value 54.0 mm in the "Modulo length" box.



The screenshot shows a configuration window titled "Modulo". It contains the following elements:

- A checked checkbox labeled "Enable modulo".
- A text input field labeled "Start value:" containing the value "0.0" and a unit selector "mm".
- A text input field labeled "Length:" containing the value "54.0" and a unit selector "mm".

You can use the default values for the extended parameters in this example.

### 5.4.2 Configuring a speed axis

The speed axis technology object enables you to specify the speed for a drive. You can assign motion jobs for the agitator with Motion Control instructions in your user program.

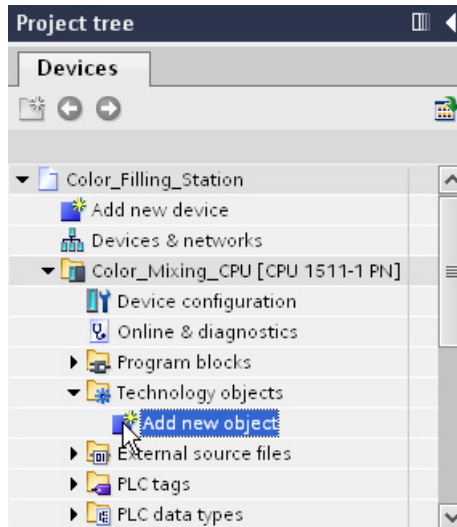
In the following example, you create a speed axis technology object and assign it to the previously configured SINAMICS G120 drive. You assign the drive interface with PROFIdrive frame 1. The properties of the drive interface are set automatically with this step.

The organization blocks "MC-Servo" and "MC-Interpolator" were already created when you created the positioning axis.

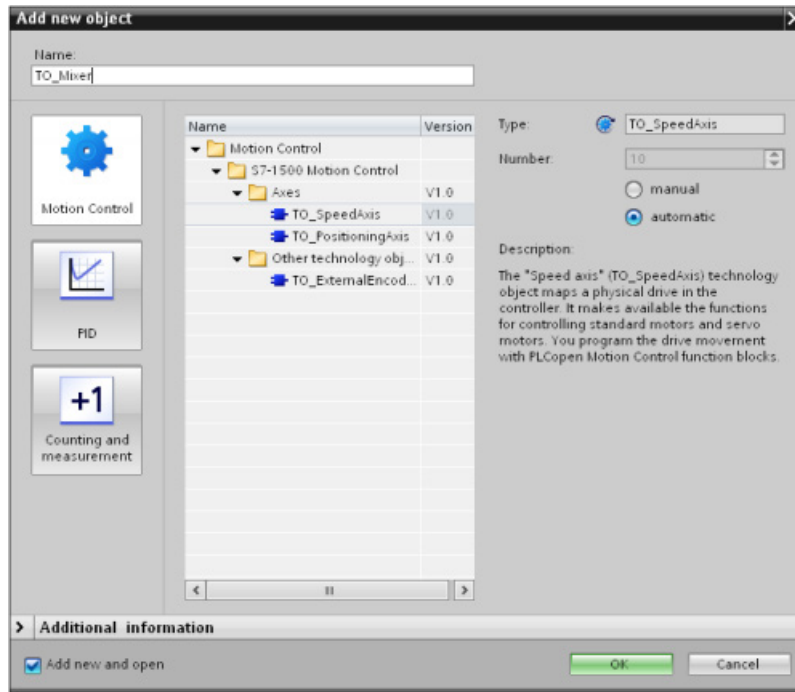
## Creating a speed axis

To create the speed axis, follow these steps:

1. Open the "Color\_Mixing\_CPU > Technology objects" folder in the project tree.
2. Double-click the "Add new object" command.



The "Add new object" dialog box opens.



3. Select the "TO\_SpeedAxis" object.
4. Enter "TO\_Mixer" as the name of the technology object.
5. Confirm your entry with "OK".

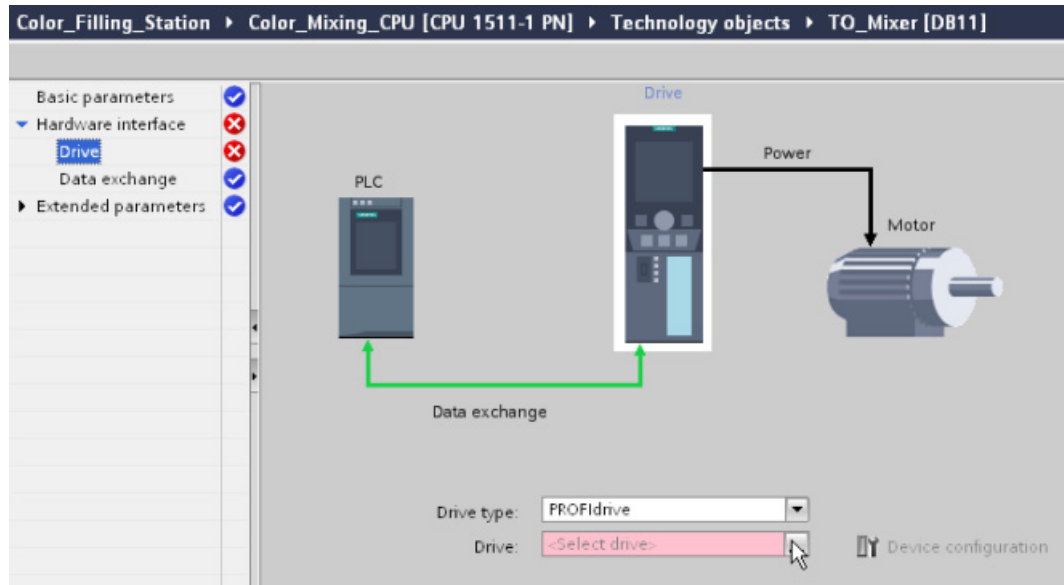
The technology object is created and opened in the project view.

## Assigning a drive

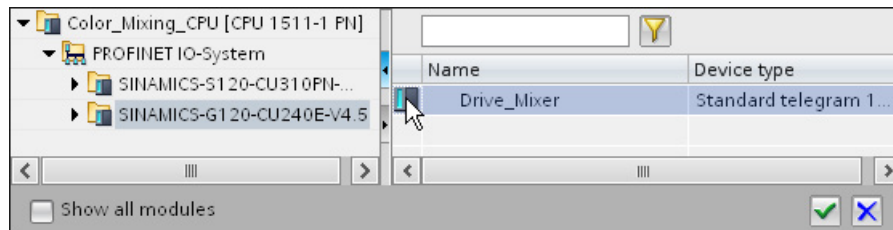
For this example, you can use the default values of the configuration. You only need to assign the corresponding drive to the speed axis.

To assign a drive to the speed axis, proceed as follows:

1. Open the configuration window "Hardware interface > Drive".



2. In the "Drive" list, select the entry "Drive\_Mixer" of the PROFINET drive SINAMICS G120.



The speed axis is connected to the drive. The process image of the drive is assigned to the process image partition "TPA OB Servo".

3. Open the configuration window "Hardware interface > Data exchange".
4. Check the reference speed. If necessary, adapt the reference speed to match the drive setting.

## Compiling a project

Compile the project to check the configuration. The compilation should take place without errors.

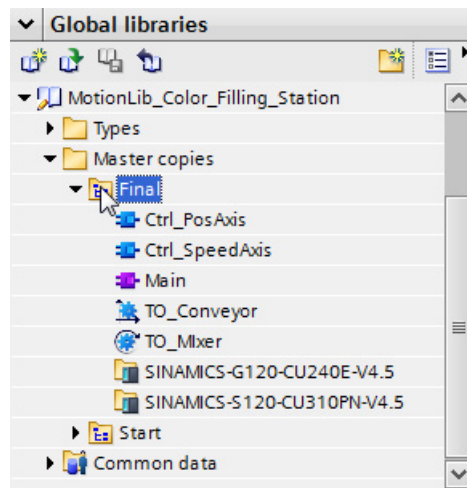
### 5.4.3 Using prepared technology objects

You can shorten the described procedure by inserting the prepared technology objects from the global library into the project and assigning the corresponding drives.

#### Inserting technology objects

To insert the technology objects from the global library into the project, follow these steps:

1. Open the folder "Master copies > Final" in the "MotionLib\_Color\_Filling\_Station" global library.



2. Drag-and-drop the "TO\_Mixer" and "TO\_Conveyor" technology objects to the "Color\_Mixing\_CPU > Technology objects" folder.
3. Open the configuration window "Hardware interface > Drive" for each technology object.
4. In the "Drive" list, select the entry "Drive\_Conveyor" of the PROFINET drive SINAMICS S120 for the positioning axis.
5. In the "Drive" list, select the entry "Drive\_Mixer" of the PROFINET drive SINAMICS G120 for the speed axis.
6. Check the settings for the data exchange. If necessary, adapt the settings to match those of the drives.
7. Compile the project to check the configuration.

The compilation should take place without errors.

## 5.5 Programming the PLC

### 5.5.1 Structure of the user program

In this example, the user program of the corresponding axis is created in a separate function block.

The required Motion Control instructions are inserted as instances into the function blocks. The function blocks are then called in the cyclic user program.

### 5.5.2 Adding Motion Control instructions

#### 5.5.2.1 Adding Motion Control instructions for a positioning axis

Insert the following Motion Control instructions in the "Ctrl\_PosAxis" function block for the user program to control the positioning axis:

- MC\_Power
- MC\_Reset

The following Motion Control instructions have already been inserted and interconnected with the corresponding tags:

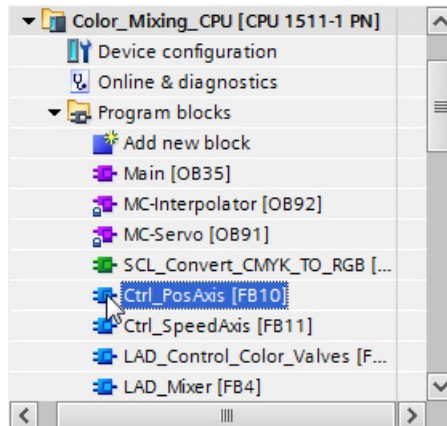
- MC\_Home
- MC\_MoveRelative
- MC\_MoveAbsolute

A network for transfer of error messages is also already in place.

## Opening function blocks

To open the function block, follow these steps:

1. Open the "Color\_Mixing\_CPU > Program blocks" folder in the project tree.



2. Double-click the function block "Ctrl\_PosAxis".

The function block opens.

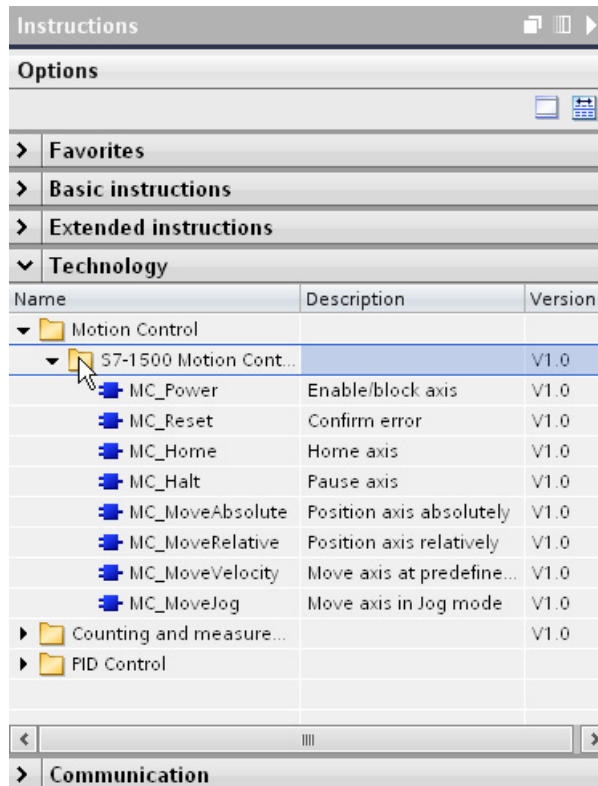
The following tags have been created in the "Interface" area:

- "Axis" - transfer variable for the technology object when the function block is called in the cyclic user program
- "Turn\_On\_Drive" - for switching on the drive with the "MC\_Power" instruction
- "Move\_Can\_Cycle" - for starting the axis movement with the "MC\_MoveRelative" instruction
- "Goto\_Fillpos" - for starting the axis movement with the "MC\_MoveAbsolute" instruction
- "Reset" - for error acknowledgment
- "Can\_Cycle\_Done" - feedback of the "MC\_Relative" instruction
- "Fillpos\_Reached" - feedback of the "MC\_MoveAbsolute" instruction
- "Error" - error message
- "temp\_Status\_power" - further interconnection of the feedback of the "MC\_Power" instruction to the "MC\_Home" instruction
- "temp\_Position" - reading out the actual position to use for visualization

## Inserting MC\_Power

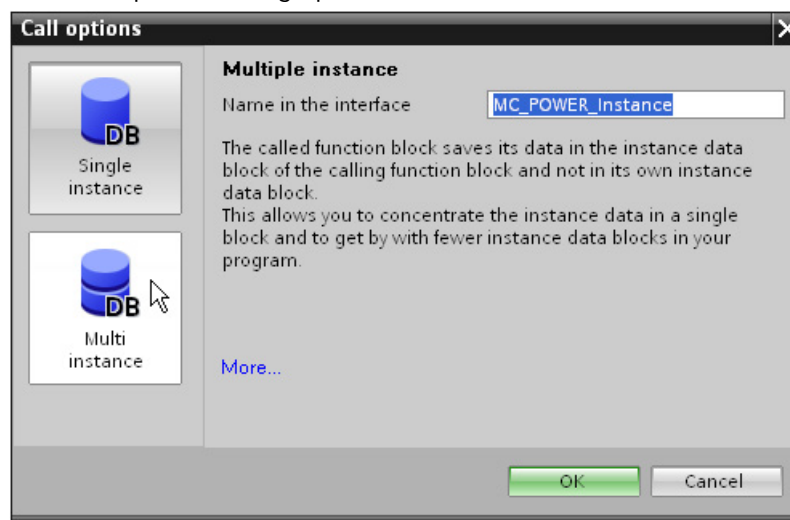
To insert the Motion Control instruction "MC\_Power" in the function block, follow these steps:

1. In the "Instructions" task card, open the "Technology > Motion Control > S7-1500 Motion Control" folder.



2. Insert the Motion Control instruction "MC\_Power" into network 1 using drag-and-drop.

The "Call options" dialog opens.



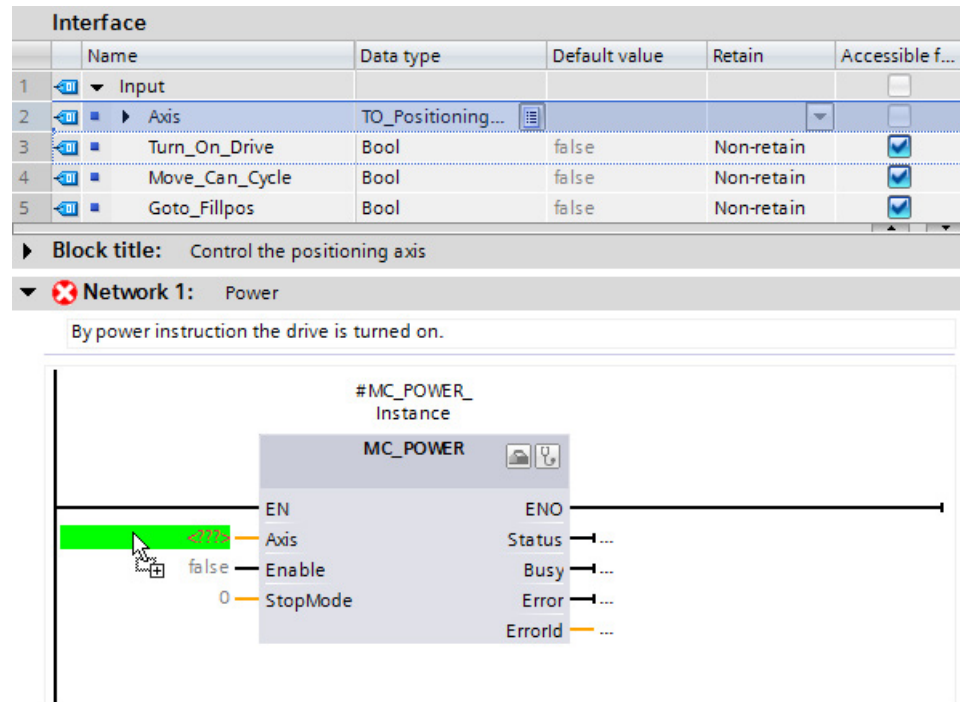
3. Click the "Multi instance" button.



- Confirm your entry with "OK".

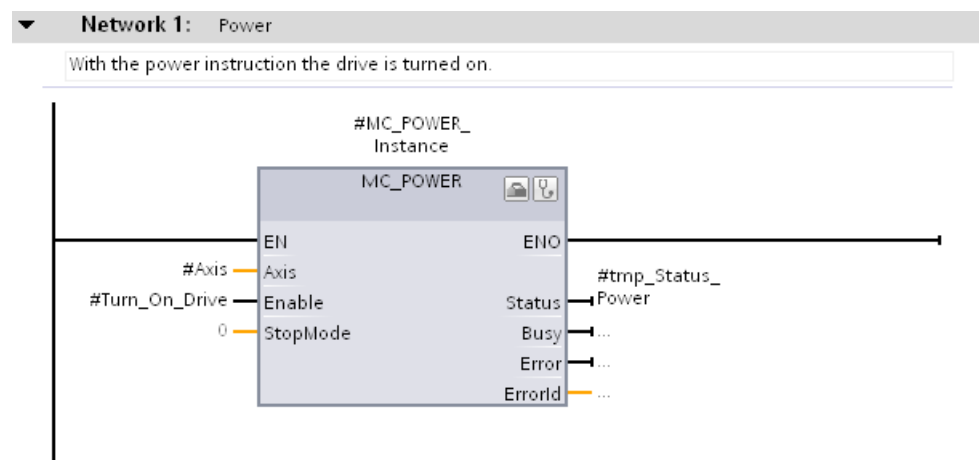
The Motion Control instruction "MC\_Power" is added to network 1.

- Drag-and-drop the "Axis" tag from the "Interface" area to the "Axis" parameter of the Motion Control instruction.



- Drag-and-drop the "Turn\_On\_Drive" tag to the "Enable" parameter of the Motion Control instruction.

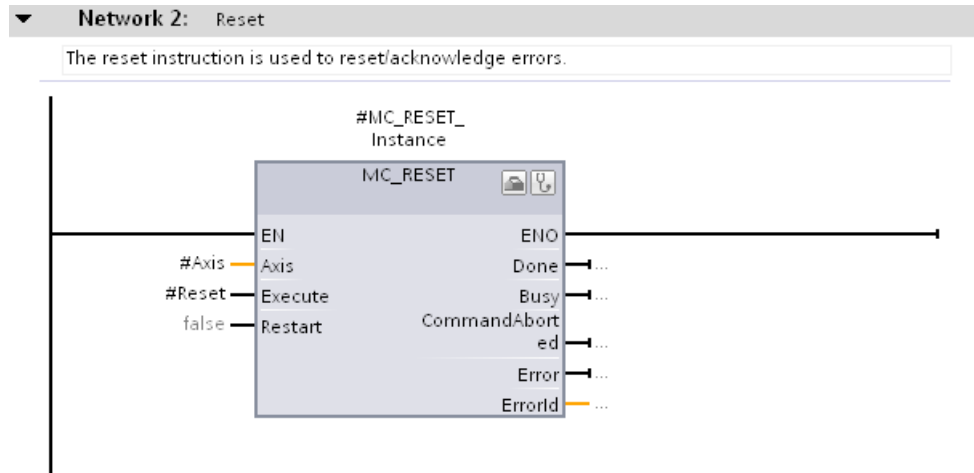
- Drag-and-drop the "tmp\_Status\_power" tag to the "Status" parameter of the Motion Control instruction.



### Inserting MC\_Reset

To insert the Motion Control instruction "MC\_Reset" in the function block, follow these steps:

1. Insert the Motion Control instruction "MC\_Reset" into an empty network according to steps 1 to 5 of the operating instruction "Insert MC\_Power".
2. Drag-and-drop the "Reset" tag to the "Execute" parameter of the Motion Control instruction.



### 5.5.2.2 Adding Motion Control instructions for a speed axis

Insert the following Motion Control instruction in the "Ctrl\_SpeedAxis" function block for the user program to control the speed axis:

- MC\_Power
- MC\_Reset

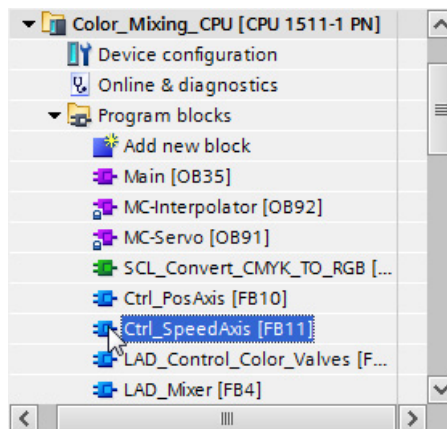
The Motion Control instruction "MC\_MoveVelocity" has already been inserted and linked with the corresponding tags.

A network for transfer of error messages is also already in place.

### Opening function blocks

To open the function block, follow these steps:

1. Open the "Color\_Mixing\_CPU > Program blocks" folder in the project tree.



2. Double-click the function block "Ctrl\_SpeedAxis".

The function block opens.

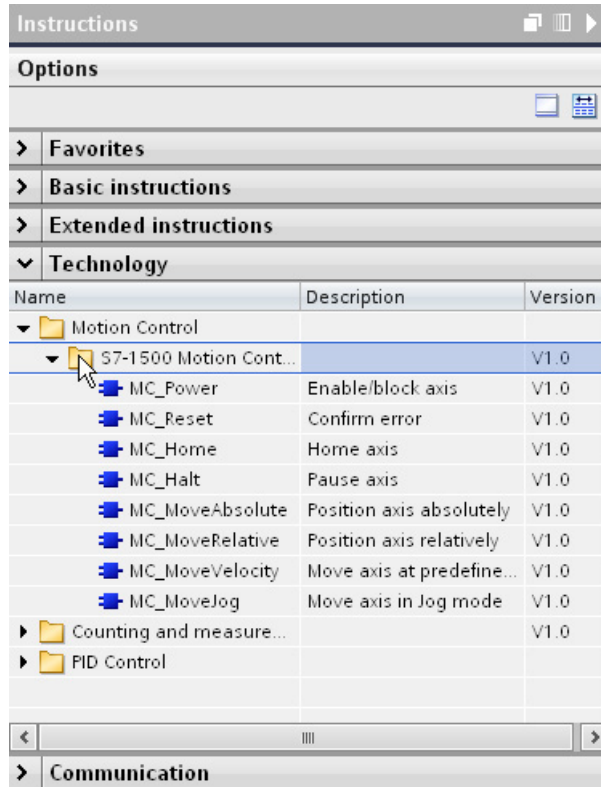
The following tags have been created in the "Interface" area:

- "Axis" - transfer variable for the technology object when the function block is called in the cyclic user program
- "Start\_Mixing" - for switching on the drive with the "MC\_Power" instruction
- "Reset" - for error acknowledgment
- "Error" - error message
- "temp\_Status\_power" - further interconnection of the feedback of the "MC\_Power" instruction to the "MC\_MoveVelocity" instruction

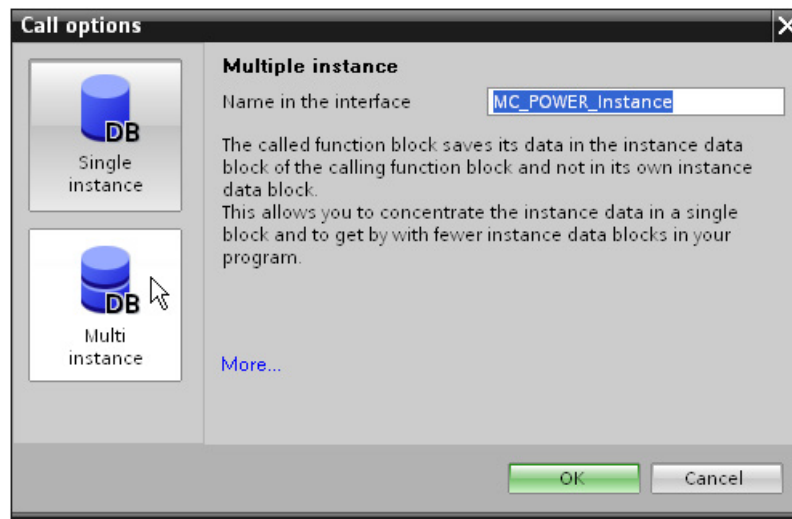
### Inserting MC\_Power

To insert the Motion Control instruction "MC\_Power" in the function block, follow these steps:

1. In the "Instructions" task card, open the "Technology > Motion Control > S7-1500 Motion Control" folder.



2. Insert the Motion Control instruction "MC\_Power" into network 1 using drag-and-drop. The "Call options" dialog opens.

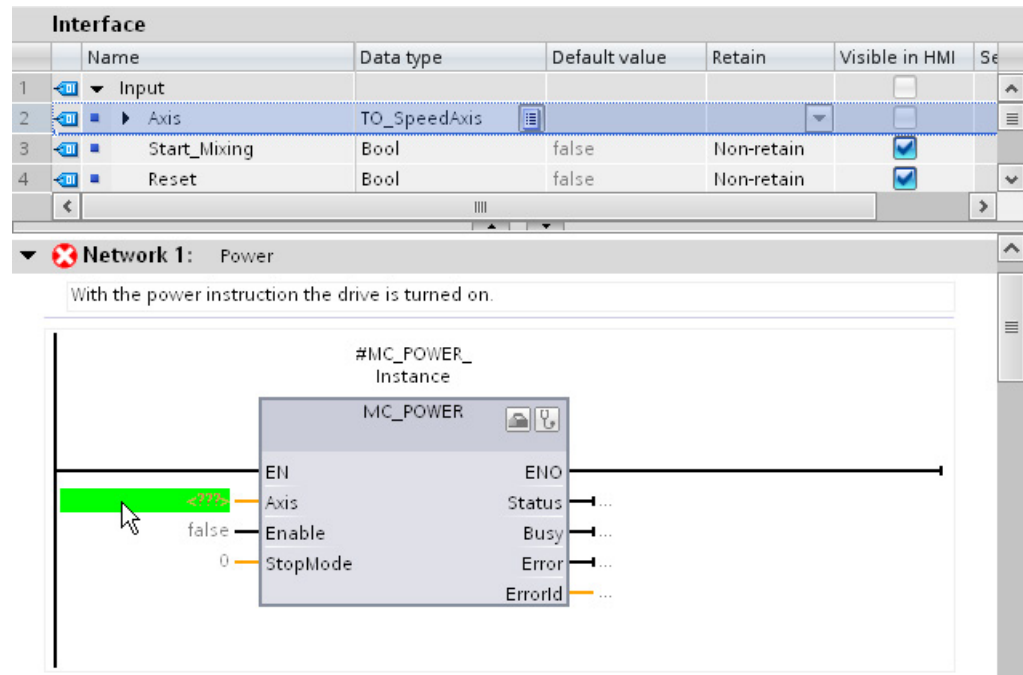


3. Click the "Multi instance" button.

4. Confirm your entry with "OK".

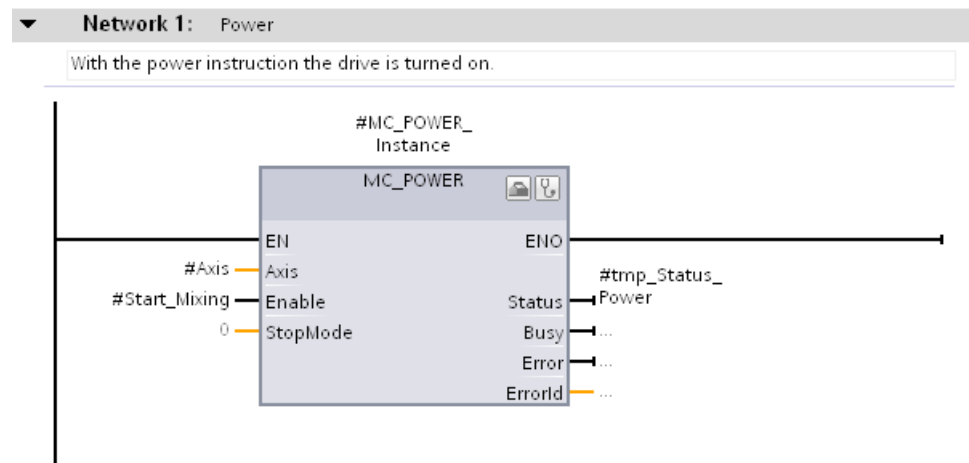
The Motion Control instruction "MC\_Power" is added to network 1.

5. Drag-and-drop the "Axis" tag from the "Interface" area to the "Axis" parameter of the Motion Control instruction.



6. Drag-and-drop the "Start\_Mixing" tag to the "Enable" parameter of the Motion Control instruction.

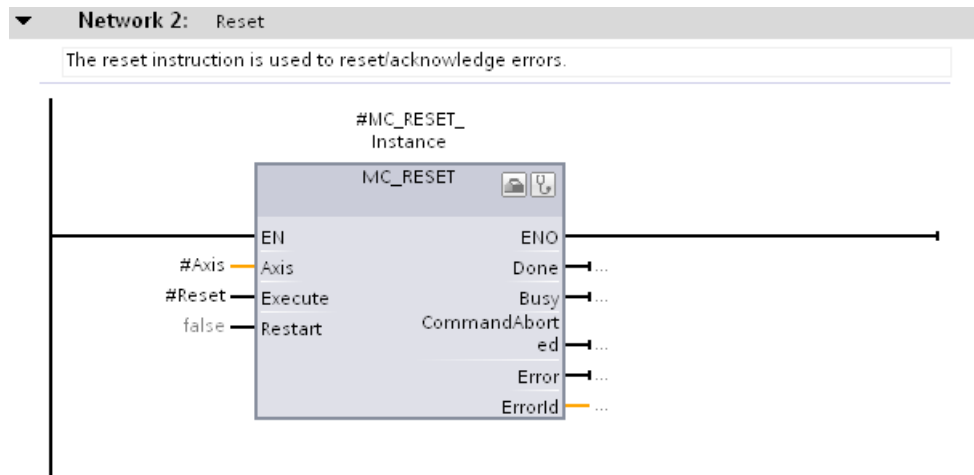
7. Drag-and-drop the "temp\_Status\_power" tag to the "Status" parameter of the Motion Control instruction.



### Inserting MC\_Reset

To insert the Motion Control instruction "MC\_Reset" in the function block, follow these steps:

1. Insert the Motion Control instruction "MC\_Reset" into an empty network according to steps 1 to 5 of the operating instruction "Insert MC\_Power".
2. Drag-and-drop the "Reset" tag to the "Execute" parameter of the Motion Control instruction.



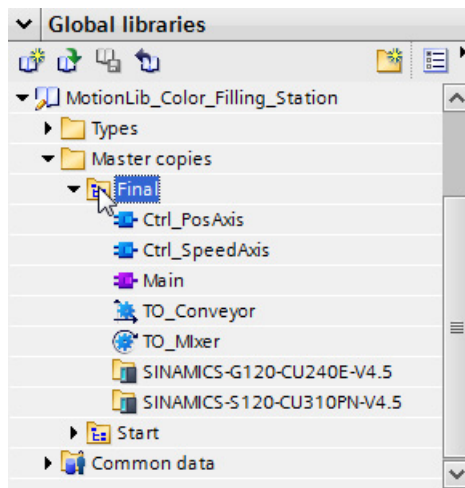
### 5.5.2.3 Using prepared function blocks

You can shorten the described procedure by inserting the prepared function blocks from the global library into the project and interconnecting the corresponding tags.

#### Inserting function blocks

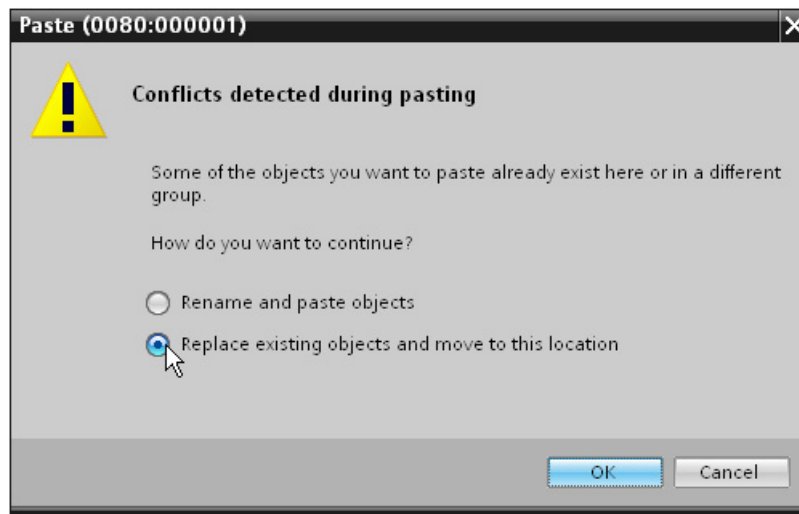
To insert the function blocks from the global library into the project, follow these steps:

1. Open the folder "Master copies > Final" in the "MotionLib\_Color\_Filling\_Station" global library.



2. Drag-and-drop both the "Ctrl\_SpeedAxis" and "Ctrl\_PosAxis" function blocks to the function blocks with the same name in the "Color\_Mixing\_CPU > Program blocks" folder.

The "Paste" dialog opens.



3. Select the option "Replace existing objects and move to this location".
4. Click "OK" to confirm.

### 5.5.3 Integrating Motion Control instructions in the cyclic user program

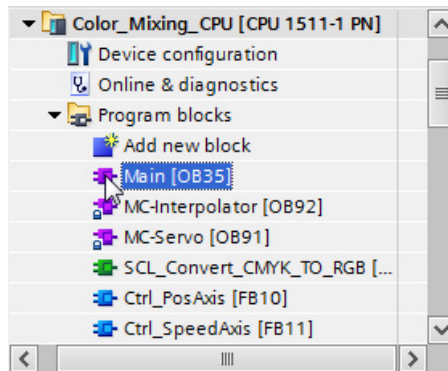
#### 5.5.3.1 Integrating Motion Control instructions for a positioning axis

The Motion Control instructions created to control the positioning axis are integrated in the cyclic user program through the corresponding function block. The function block is inserted into the "Main [OB35]" organization block and interconnected to the appropriate parameters.

#### Inserting a function block

To insert the function block in the cyclic user program, follow these steps:

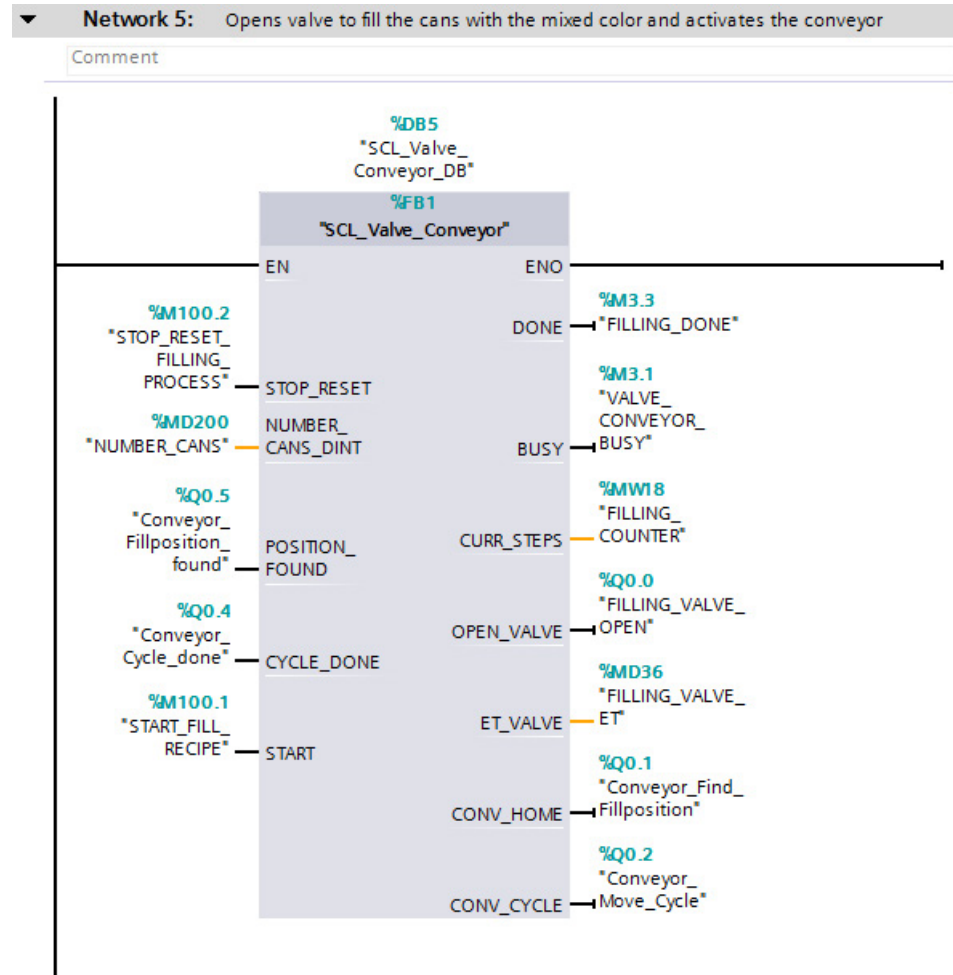
5. Open the "Color\_Mixing\_CPU > Program blocks" folder in the project tree.



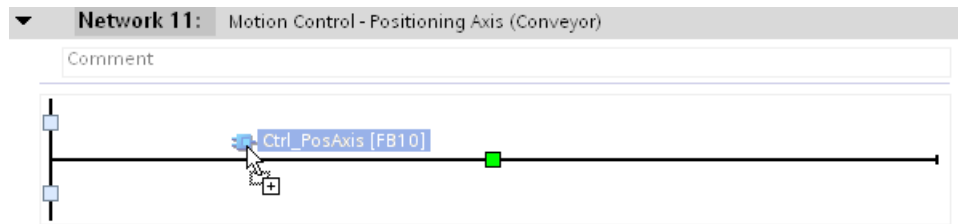


1. Open the organization block "Main [OB35]".

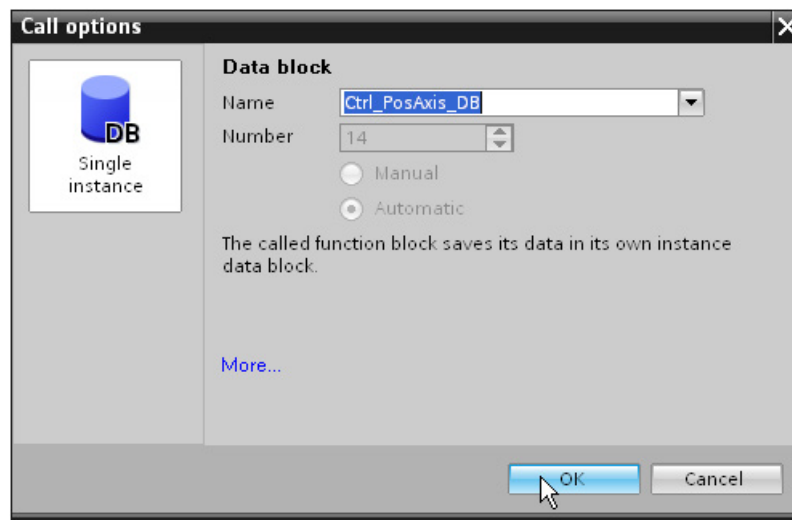
The control logic of the conveyor is located in network 5 (SCL\_Valve\_Conveyor).



2. Drag-and-drop the "Ctrl\_PosAxis" function block from the project tree into network 11 "Motion Control - Positioning Axis (Conveyor)".



The "Call options" dialog opens.

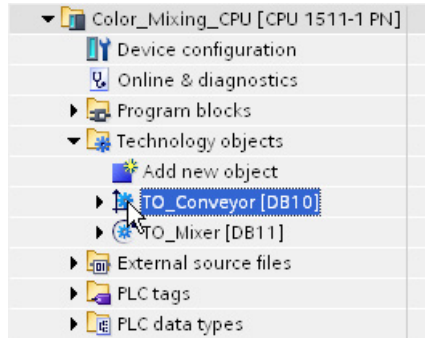


3. Click "OK" to confirm.

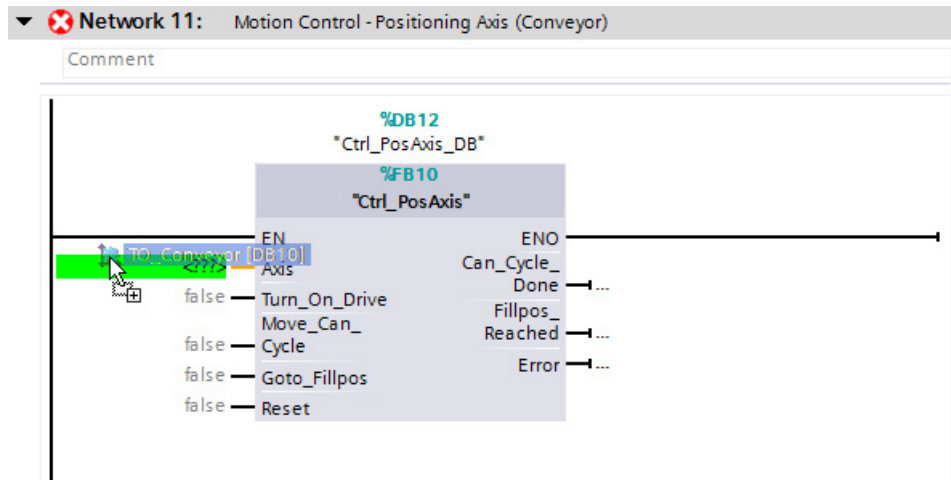
## Interconnecting the function block

To interconnect the function block with the appropriate parameters, follow these steps:

1. Open the "Color\_Mixing\_CPU > Technology objects" folder in the project tree.

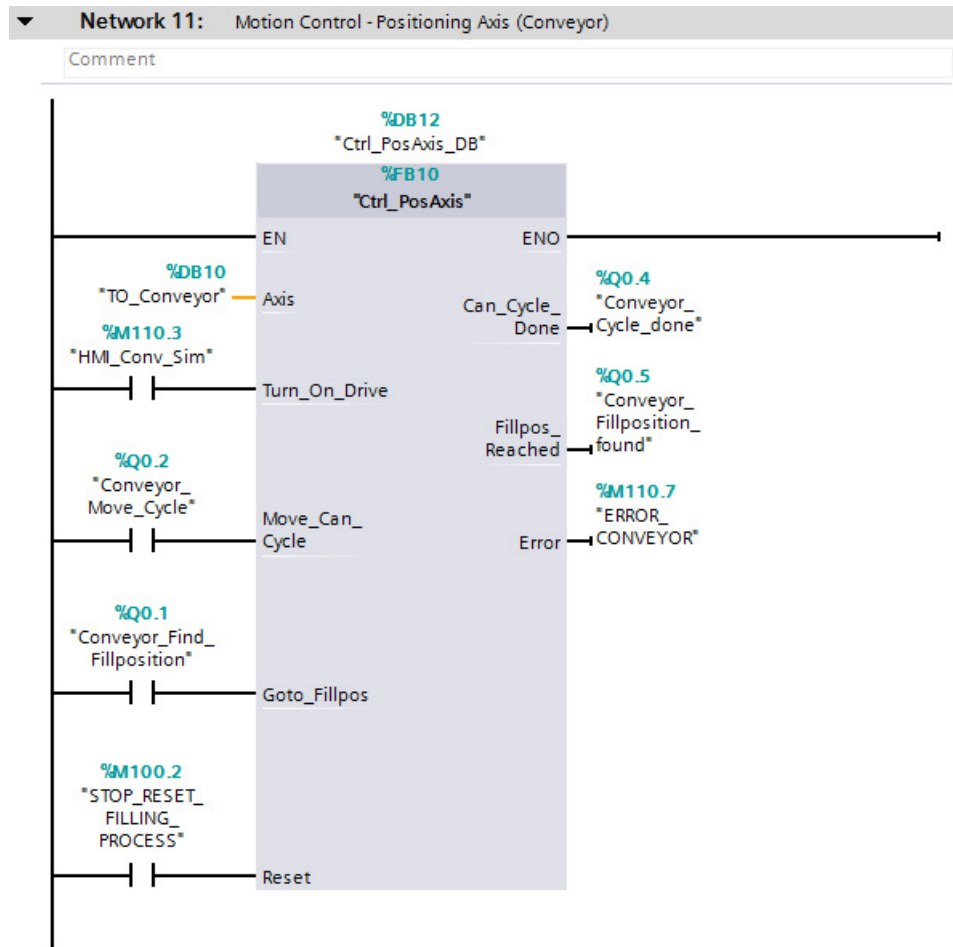


2. Drag-and-drop the technology object "TO\_Conveyor" to the "Axis\_Name" parameter in the project tree .



3. At the "Turn\_On\_Drive" parameter, select the "HMI\_Conv\_Sim" tag.
4. Keep the <Ctrl> key pressed and drag-and-drop the "Conveyor\_Move\_Cycle" tag (network 5) to the "Move\_Can\_Cycle" parameter.
5. Keep the <Ctrl> key pressed and drag-and-drop the "Conveyor\_Find\_Fillposition" tag (network 5) to the "Goto\_Fillpos" parameter.
6. Keep the <Ctrl> key pressed and drag-and-drop the "STOP\_RESET\_Filling\_Process" tag (network 5) to the "Reset" parameter.
7. Keep the <Ctrl> key pressed and drag-and-drop the "Conveyor\_Cycle\_Done" tag (network 5) to the "Can\_Cycle\_Done" parameter.

8. Keep the <Ctrl> key pressed and drag-and-drop the "Conveyor\_Fillposition\_found" tag (network 5) to the "Fillpos\_Reached" parameter.
9. At the "Error" parameter, select the "ERROR\_Conveyor" tag.



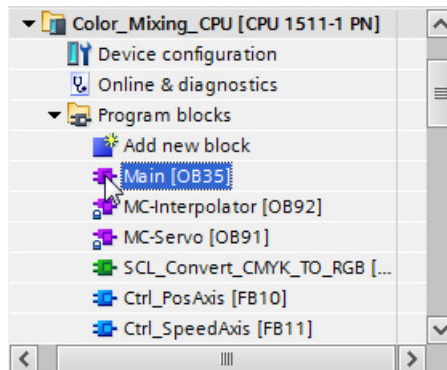
### 5.5.3.2 Integrating Motion Control instructions for a speed axis

The Motion Control instructions used to control the speed axis are integrated in the cyclic user program through the corresponding function block. The function block is inserted into the "Main [OB35]" organization block and interconnected to the appropriate parameters.

#### Inserting a function block

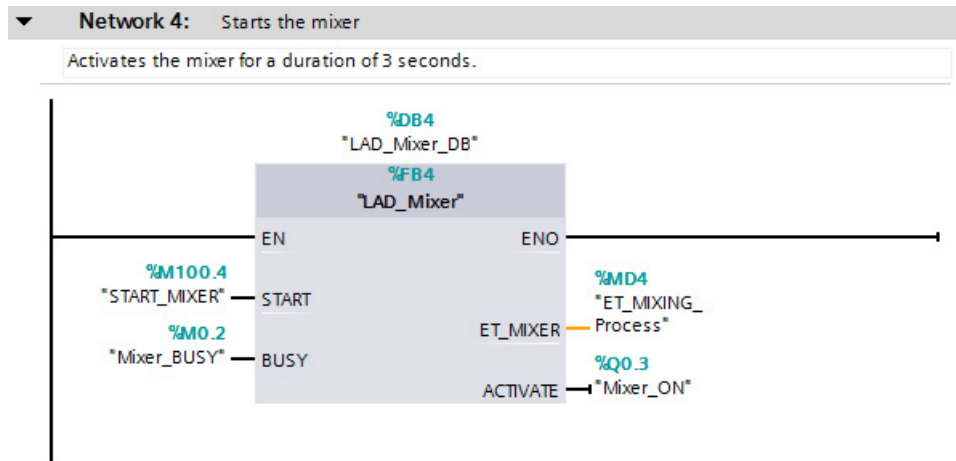
To insert the function block in the cyclic user program, follow these steps:

1. Open the "Color\_Mixing\_CPU > Program blocks" folder in the project tree.

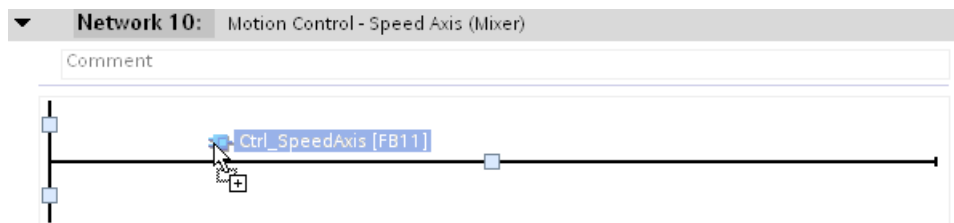


2. Open the organization block "Main [OB35]".

The control logic of the agitator (LAD\_Mixer) is located in network 4.

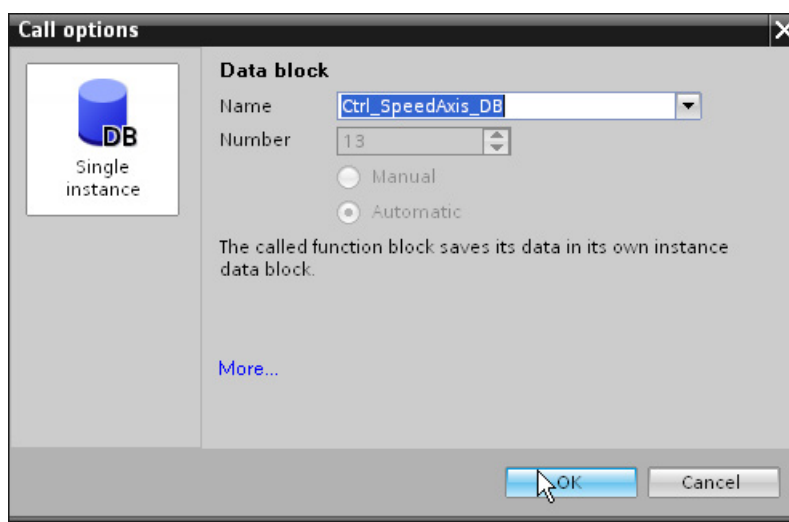


3. Drag-and-drop the "Ctrl\_SpeedAxis" function block from the project tree into network 10 "Motion Control Speed Axis (Mixer)".



The "Call options" dialog opens.

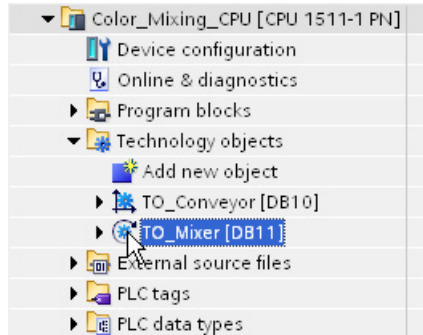
4. Click "OK" to confirm.



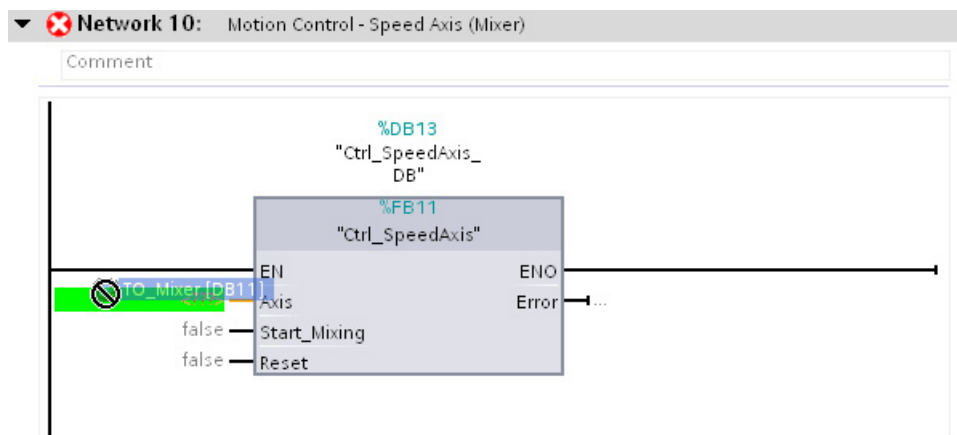
## Interconnecting the function block

To interconnect the function block with the appropriate parameters, follow these steps:

1. Open the "Color\_Mixing\_CPU > Technology objects" folder in the project tree.

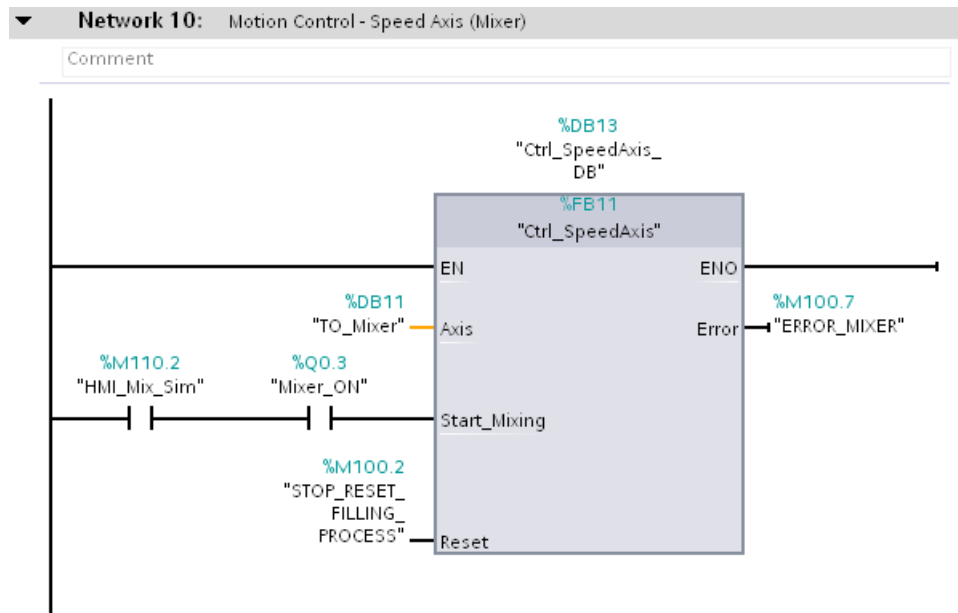


2. Drag-and-drop the technology object "TO\_Mixer" to the "Axis\_Name" parameter in the project tree .



3. Keep the <Ctrl> key pressed and drag-and-drop the "Mixer\_ON" tag (network 4) to the "Start\_Mixing" parameter.
4. Add an AND logic operation with the "HMI\_Mix\_Sim" tag to the "Mixer\_ON" tag at the "Start\_Mixing" parameter.

5. Keep the <Ctrl> key pressed and drag-and-drop the "STOP\_RESET\_Filling\_Process" tag (network 5) to the "Reset" parameter.



6. At the "Error" parameter, select the "ERROR\_MIXER" tag.



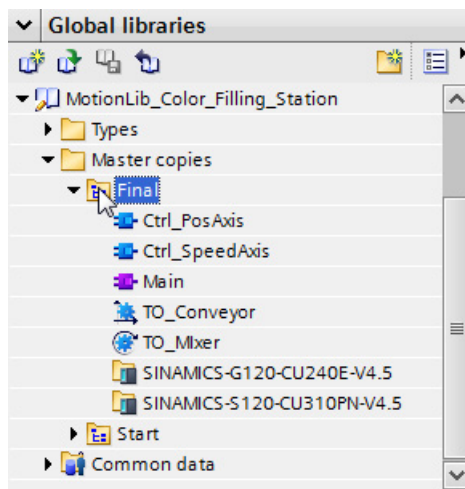
### 5.5.3.3 Using the prepared OB35

You can shorten the described procedure by inserting the prepared "Main [OB35]" organization block from the global library into the project.

#### "Inserting "Main [OB35]"

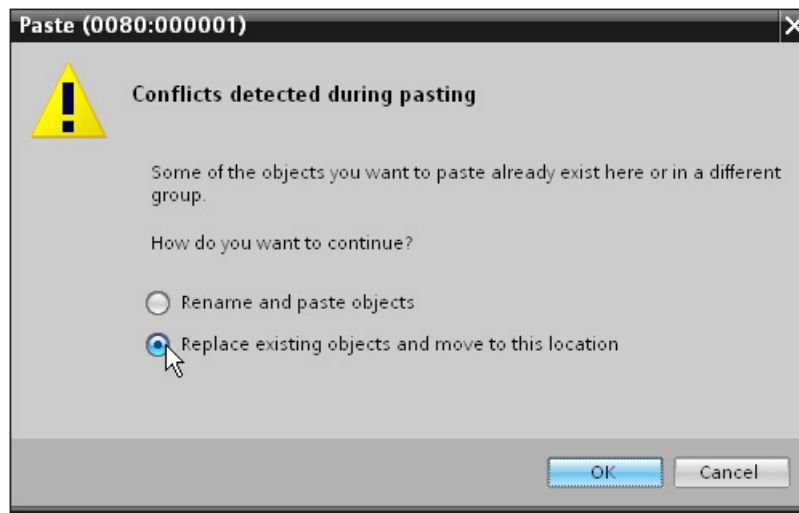
To insert the "Main [OB35]" organization block from the global library into the project, follow these steps:

1. Open the folder "Master copies > Final" in the "MotionLib\_Color\_Filling\_Station" global library.



2. Drag-and-drop the "Main" organization block to the corresponding organization block in the "Color\_Mixing\_CPU > Program blocks" folder.

The "Paste" dialog opens.



3. Select the option "Replace existing objects and move to this location".
4. Click "OK" to confirm.

### 5.5.4 Compiling and loading the project

Compile the project and download it to the CPU to test the functions of the axes in the next step.

#### Procedure

To compile the project, follow these steps:

1. Compile the hardware and software of all devices in the project.
2. Load all devices.

## 5.6 Testing a function

### 5.6.1 "Positioning axis" axis control panel

You use the axis control panel and tuning for the function test and commissioning of the positioning axis to position the conveyor belt.

The procedure is divided into the following actions:

- Enabling the technology object
- Checking the direction of rotation
- Checking the distance evaluation
- Checking the velocity evaluation
- Optimizing gain
- Applying gain to project
- Evaluating errors

#### Requirement

- The drive has been commissioned and is ready for operation.
- The drive has been assigned a PROFINET device name.

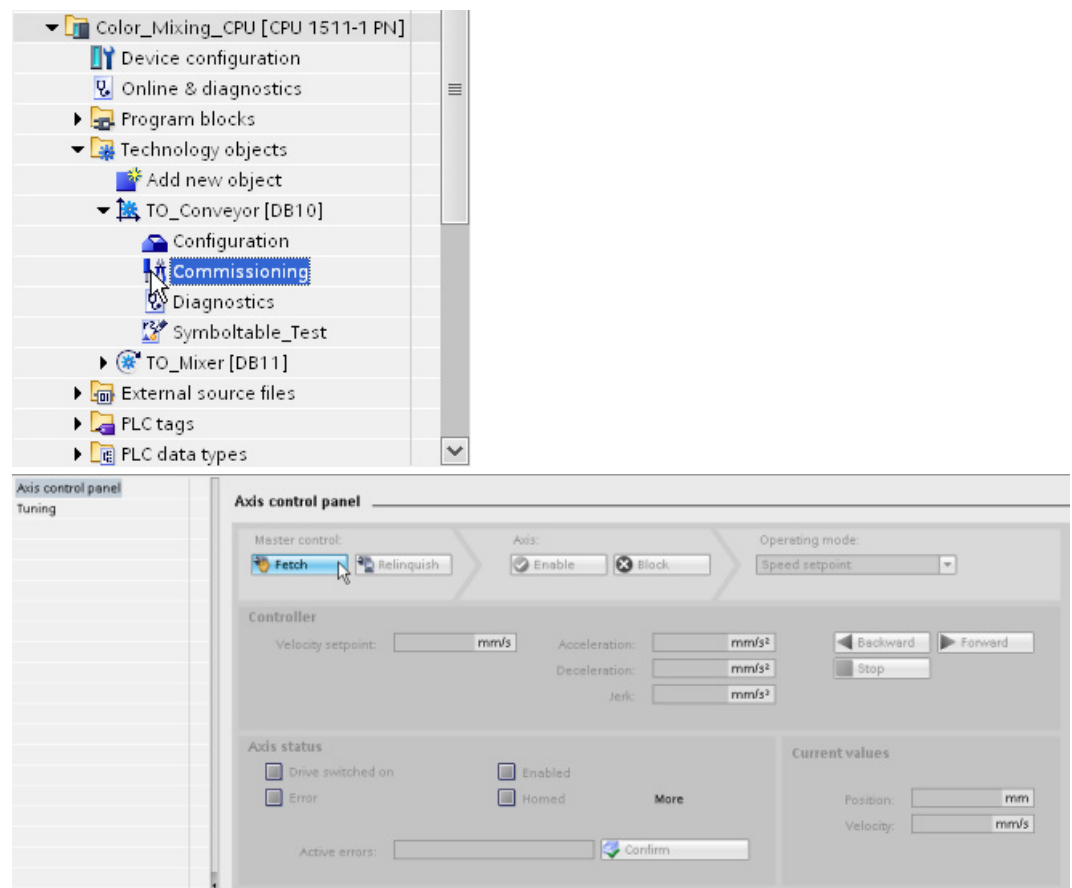
## Enabling the technology object

To enable the positioning axis with the axis control panel, follow these steps:

1. Open the "Color\_Mixing\_CPU > Technology objects" folder in the project tree.
2. Open the commissioning window "Configuration > Extended parameters > Control loop" of the technology object "TO\_Conveyor".
3. Enter the value 0.0 in the "Gain (Kv factor)" box.

You can compare the actual value and the setpoint of the velocity directly with each other if there is no gain of the control loop. Optimize the gain once you have checked the velocity evaluation and apply it to your project.

4. Compile the CPU software.
5. Load the CPU.
6. Open the commissioning window of the technology object "TO\_Conveyor".



- To create an online connection and take over the master control of the technology object, click the "Fetch" button.

A warning message is displayed.

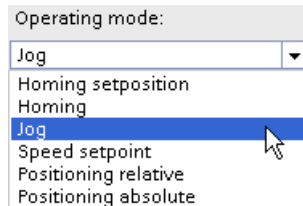


- Confirm with "Yes".
- To enable the technology object, click the "Enable" button.  
The drive is turned on and the speed setpoint zero is output.

### Checking the direction of rotation

To move the axis with the axis control panel and check the direction of rotation, follow these steps:

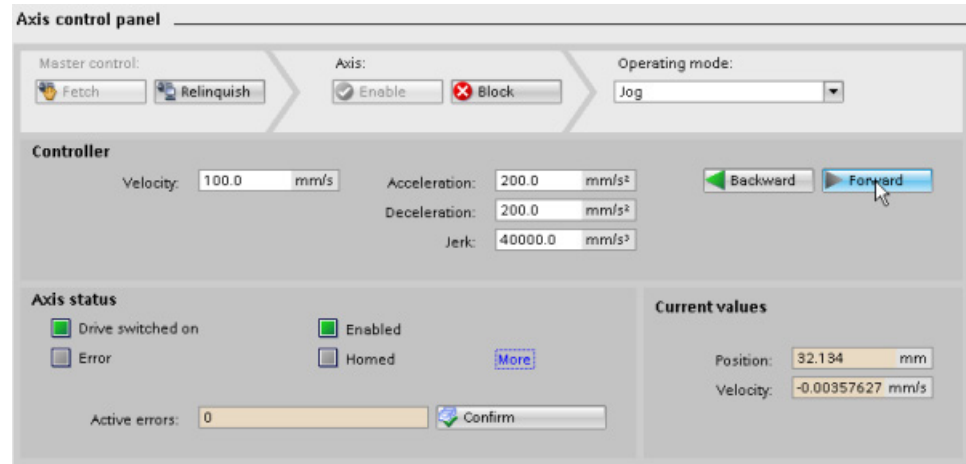
- Select the "Jog" mode in the axis control panel.



- Enter a velocity setpoint as well as the required values for acceleration, deceleration and jerk.

- Click the "Forward" or "Backward" button to execute a movement in the positive or negative direction.

The motion is executed as long as you press the mouse button.



- Check whether the actual direction of rotation is the same as the specified direction of rotation.

If necessary, you can invert the direction of rotation under "Technology object > Configuration > Hardware interface > Data exchange".

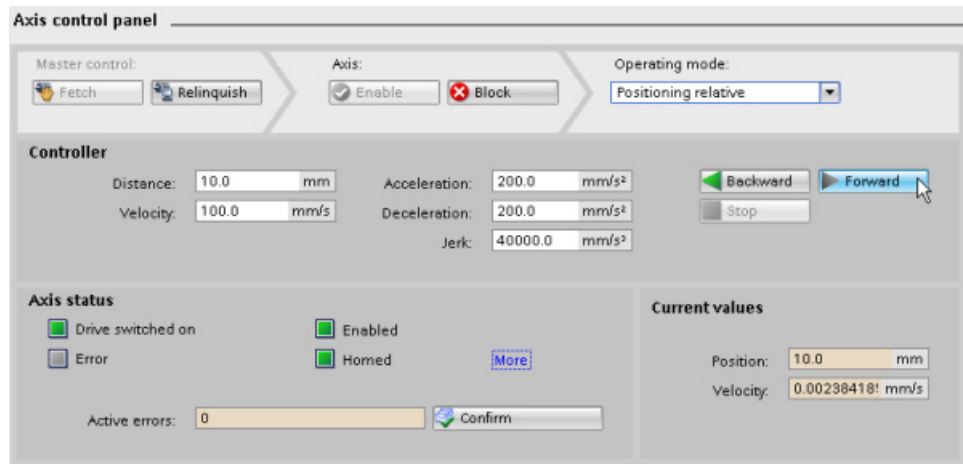
- Check the actual speed in the drive or use an external tachometer to measure the speed directly at the motor shaft.

If the actual speed deviates significantly from the specified setpoint, check the reference speed and the parameter assignment of the encoder.

## Checking the distance evaluation

To move the axis with the axis control panel and check the distance evaluation, follow these steps:

1. Mark the current mechanical position of the axis.
2. Select the "Homing setposition" mode in the axis control panel.
3. Enter a defined value (e.g. 0.0 mm) in the "Position" box.
4. Click the "Position setpoint" button to set the actual position to the value specified in "Position".
5. Select "Positioning relative" mode.
6. Enter only small values for the distance and the velocity (e.g. 10 mm and 100 mm/s).
7. To move the axis in the positive direction, click the "Forward" button.



8. Use the selection on the axis to check whether the current position displayed matches the actually traveled distance.

If the displayed current position deviates from the actually traveled distance, check the configuration of the encoder and the gearbox ("Technology object > Configuration > Extended parameters > Mechanics").

## Checking the velocity evaluation

To move the axis with the axis control panel and check the velocity, follow these steps:

1. Select the "Speed setpoint" mode in the axis control panel.
2. Enter a small value (e.g. 100 mm/s) for the velocity setpoint.
3. To move the axis in the positive direction, click the "Forward" button.
4. Check whether the displayed current velocity matches the velocity setpoint.

If the displayed current velocity deviates significantly from the velocity setpoint, adjust the reference speed (Technology object > Hardware interface > Data exchange).

5. Click the "Stop" button to stop the movement.

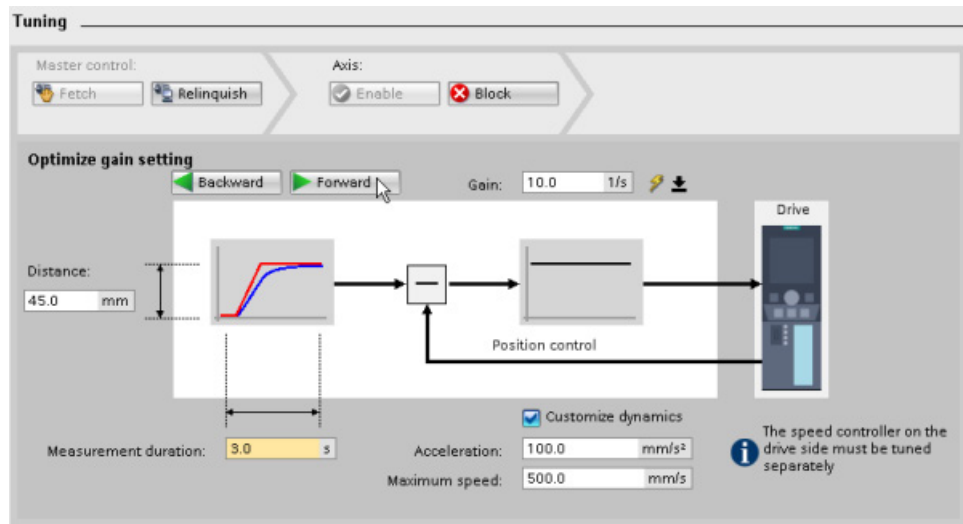


## Optimizing gain

The "Tuning" function supports you in determining the optimum gain (Kv factor) for the position control of the axis. For this purpose, the axis velocity profile is recorded by means of the Trace function for the duration of a configurable positioning movement. Then you can evaluate the recording and adapt the gain accordingly.

To determine the gain factor with the tuning, follow these steps:

1. Open the tuning in the commissioning window.



2. If necessary, configure values for the distance, duration, and dynamics of a test step.
3. Enter a start value for the gain. Start with a low value.



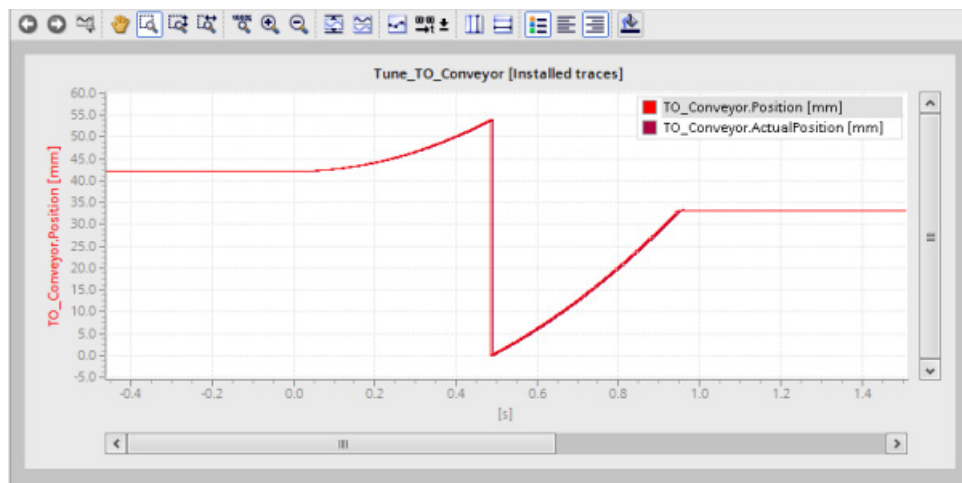
- Click the "Forward" or "Backward" button to start a test step for the tuning in positive or negative direction.

A setpoint is output for the specified duration according to the specified distance. The axis moves by the specified distance. A trace recording of the motion (setpoint and actual values) is created automatically.

- Evaluate the curve of the trace recording. Adapt the gain incrementally. Click the "Forward" or "Backward" button after each value that you input. This applies the value and starts a new movement and trace recording each time.

When adjusting the gain, pay attention to the following properties of the curve:

- The curve shows a brief compensation time.
- The curve does not show any motion reversal of the actual value.
- When approaching the setpoint, no overshoot occurs.
- The curve shows stable overall behavior (oscillation-free curve).



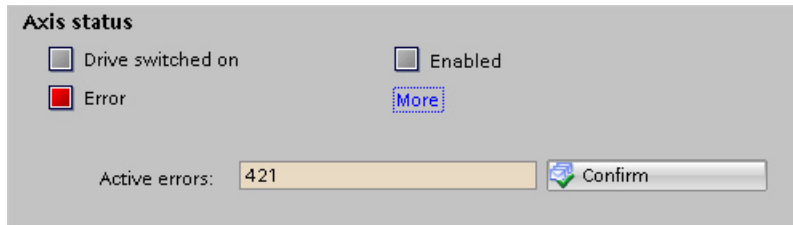
## Applying gain to project

To apply the determined gain (Kv) to your project, follow these steps:

- Click on the ⚡ icon next to the "Gain" box.  
A drop-down list is displayed.
- Enter the value of the determined gain in the "Project start value" box of the drop-down list.
- To disable the technology object, click the "Block" button.
- Click the "Relinquish" button to return master control to your user program.
- Compile the CPU software.
- Load the CPU.

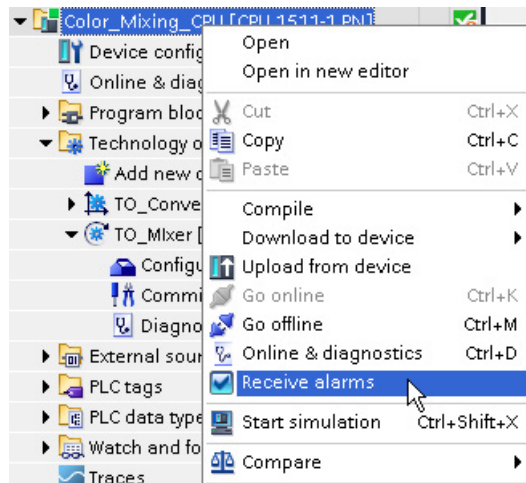
## Evaluating errors

If an error occurs during control of the technology object with the axis control panel, this is indicated by the "Error" status.



To evaluate an error, follow these steps:

1. Select the "Receive alarms" check box in the shortcut menu of the CPU to display the error messages in the alarm display.



2. To display the error bits of the technology object, click the "More" button.
3. Evaluate any pending errors and remove the cause of the error.
4. Click the "Confirm" button in the axis control panel to acknowledge an error.

---

### Note

#### Adapt following error limits

If error messages are repeatedly displayed from following error monitoring, temporarily adapt the following error limits.

---

## 5.6.2 "Speed axis" axis control panel

You use the axis control panel for the function test and commissioning of the speed axis to control the agitator.

The procedure is divided into the following actions:

- Enabling the technology object
- Moving and checking the axis
- Evaluating errors

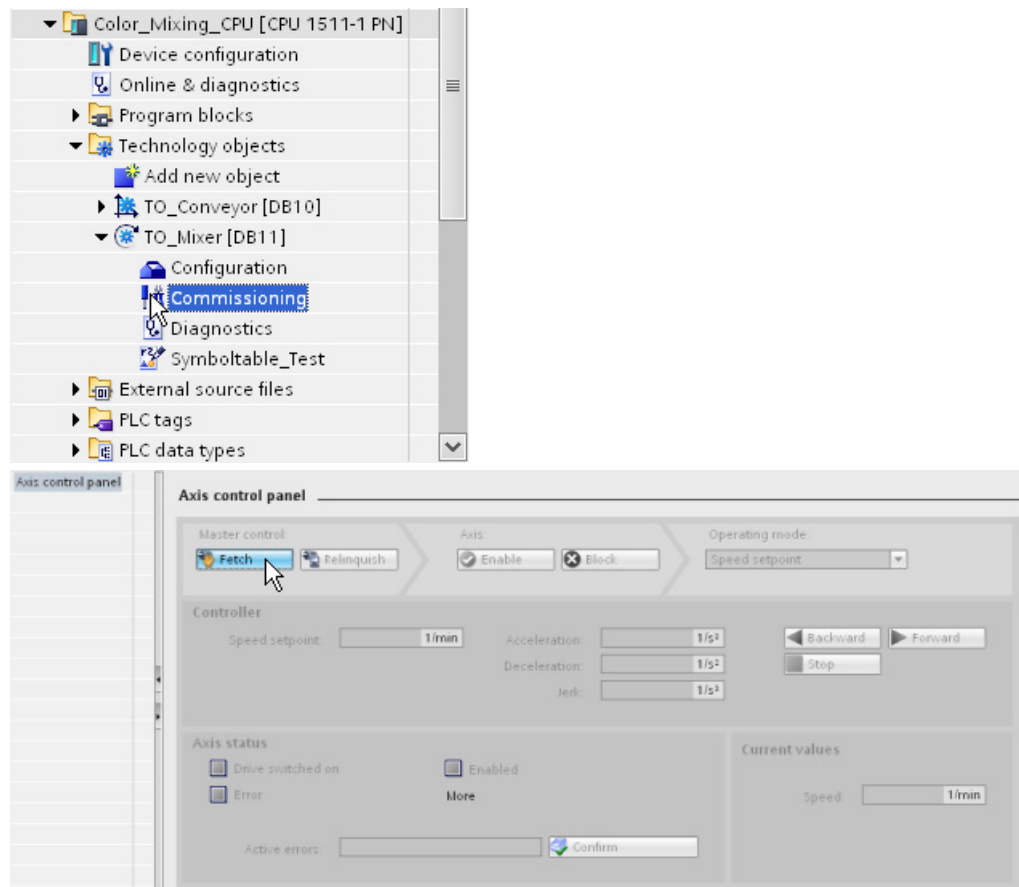
### Requirement

- The drive has been commissioned and is ready for operation.
- The drive has been assigned a PROFINET device name.

### Enabling the technology object

To enable the speed axis with the axis control panel, follow these steps:

1. Open the "Color\_Mixing\_CPU > Technology objects" folder in the project tree.
2. Open the commissioning window of the technology object "TO\_Mixer".



- To create an online connection and take over the master control of the technology object, click the "Fetch" button.

A warning message is displayed.



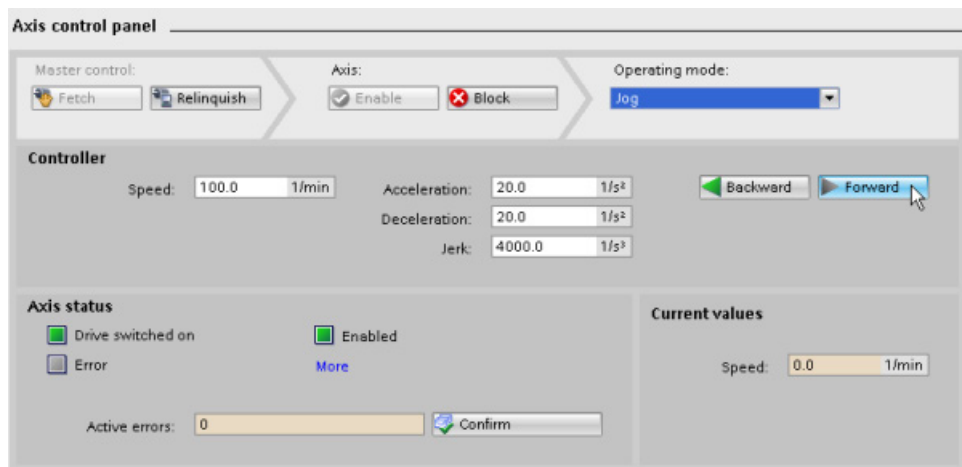
- Confirm with "Yes".
- To enable the technology object, click the "Enable" button.  
The drive is turned on and the speed setpoint zero is output.

## Moving and checking the axis

To move the axis with the axis control panel and to control the movement, follow these steps:

- Select the "Jog" mode in the axis control panel.
- Enter a speed setpoint as well as the required values for acceleration, deceleration and jerk.
- Click the "Forward" or "Backward" button to execute a movement in the positive or negative direction.

The motion is executed as long as you press the mouse button.



- Check whether the actual direction of rotation is the same as the specified direction of rotation.

If necessary, you can invert the direction of rotation under "Technology object > Configuration > Hardware interface > Data exchange".

5. Check the actual speed in the drive or use an external tachometer to measure the speed directly at the motor shaft.  
If the actual speed deviates significantly from the specified setpoint, check the reference speed and the parameter assignment of the encoder.
6. To disable the technology object, click the "Block" button.
7. Click the "Relinquish" button to return master control to your user program.

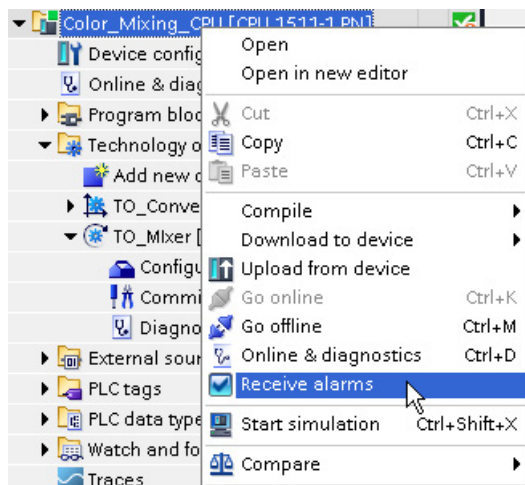
## Evaluating errors

If an error occurs during control of the technology object with the axis control panel, this is indicated by the "Error" status.



To evaluate an error, follow these steps:

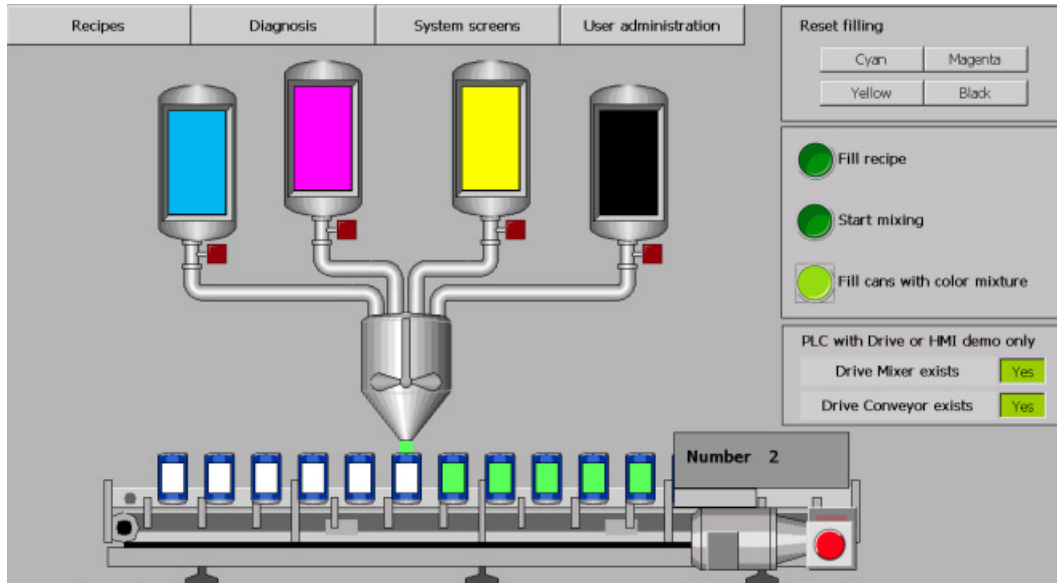
1. Select the "Receive alarms" check box in the shortcut menu of the CPU to display the error messages in the alarm display.



2. To display the error bits of the technology object, click the "More" button.
3. Evaluate any pending errors and remove the cause of the error.
4. Click the "Confirm" button in the axis control panel to acknowledge an error.

### 5.6.3 Testing the function in the start screen

You test the function of your user program by using the start screen of the visualization.



#### Procedure

To test the function of your user program, follow these steps:

1. Save the project.
2. Download the program to the CPU.
3. Start the CPU.
4. Press the "Drive Mixer exists" button in the start screen of the project.
5. Press the "Start mixing" button in the start screen of the project.

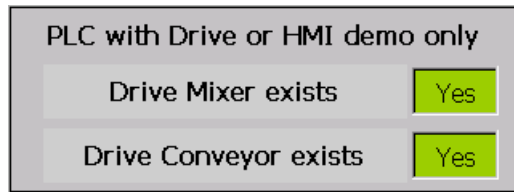
The drive of the agitator is switched on by the Motion Control instruction "MC\_Power".

When the technology object is enabled ("MC\_Power.Status" = TRUE), a job is triggered with the Motion Control instruction "MC\_MoveVelocity". The "MC\_MoveVelocity" instruction specifies the configured speed (100 rpm). The technology object calculates speed setpoints and sends them to the drive. The drive executes the received setpoints and turns the agitator.

The release by the control program is revoked after 3 seconds. The drive is stopped and switched off.

You can change the speed by defining the "Velocity" parameter at the "MC\_MoveVelocity" instruction. You can also adjust the dynamics in the configuration of the technology object under "Extended parameters > Dynamic default values".

6. Press the "Drive Conveyor exists" button in the start screen of the project.



The drive of the conveyor is switched on by the Motion Control instruction "MC\_Power".

With the enable "MC\_Power.Status" = TRUE , a job is triggered with the Motion Control instruction "MC\_Home". The current position value is applied as valid position. This references the technology object.

7. Press the "Fill cans with color mixture" button in the start screen of the project.



The can below the fill nozzle is filled and then moved forward by one can. This procedure is repeated until the set number of cans has been filled (see figure "Recipes").

## 5.7 Additional information

### Diagnosics/troubleshooting

Incorrect operation, incorrect wiring or inconsistent parameter assignment can cause errors.

The S7-1500 Motion Control function manual includes descriptions of how to diagnose such errors and alarms.

### Service & Support on the Internet

In addition to our range of documentation, you can also make use of our comprehensive online knowledge base on the Internet (<https://support.industry.siemens.com/cs/de/en/>).

There you will find:

- The newsletter that provides you with latest information relating to your products.
- The right documents for you, using the Service & Support search engine.
- A forum in which users and specialists worldwide exchange their know-how.
- Your local contact partner for Automation & Drives, using our contact partner database.
- Information about on-site services, repairs, spare parts.



# SIMATIC Safety V17

## Preface

### 6.1.1 Security information

Siemens provides products and solutions with industrial security functions that support the secure operation of plants, systems, machines and networks.

In order to protect plants, systems, machines and networks against cyber threats, it is necessary to implement – and continuously maintain – a holistic, state-of-the-art industrial security concept. Siemens' products and solutions constitute one element of such a concept.

Customers are responsible for preventing unauthorized access to their plants, systems, machines and networks. Such systems, machines and components should only be connected to an enterprise network or the internet if and to the extent such a connection is necessary and only when appropriate security measures (e.g. firewalls and/or network segmentation) are in place.

For additional information on industrial security measures that may be implemented, please visit (<https://www.siemens.com/industrialsecurity>).

Siemens' products and solutions undergo continuous development to make them more secure. Siemens strongly recommends that product updates are applied as soon as they are available and that the latest product versions are used. Use of product versions that are no longer supported, and failure to apply the latest updates may increase customers' exposure to cyber threats.

To stay informed about product updates, subscribe to the Siemens Industrial Security RSS Feed visit (<https://www.siemens.com/industrialsecurity>).

## 6.2 Introduction to example

### 6.2.1 Example Structure and Task Definition

#### Introduction

These instructions will guide you step-by-step through a specific example for configuring and programming with *STEP 7 Safety Advanced V17*.

You will become acquainted with the basic functions and special features of *STEP 7 Safety Advanced V17*. It should take one or two hours to work through this example, depending on your experience.

In order to understand these Getting Started instructions, you need general knowledge of automation engineering. You also need to be familiar with *STEP 7 Professional V17*.

#### WARNING

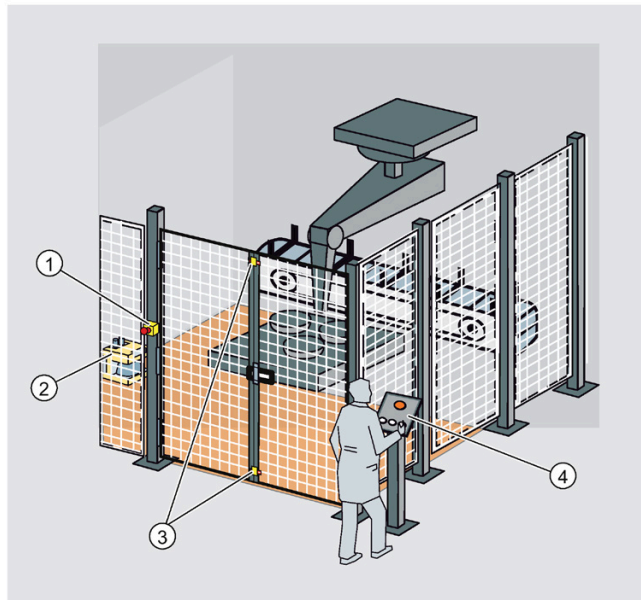
**There is a risk of personal injury or damage to property.**

As a component in plants and systems, the S7-1500 is subject to specific standards and regulations depending on the area of application. Please note the applicable safety and accident prevention regulations, e.g. IEC 602041 (General Requirements for Safety of Machinery).

The example in these Getting Started instructions serves as an introduction to configuring and programming *STEP 7 Safety Advanced V17*. It does not lead to actual live operation in every case. Before you do this, we strongly recommended that you read the latest version of the "SIMATIC Safety - Configuring and Programming (<https://support.industry.siemens.com/cs/ww/en/view/54110126>)" manual and the manuals of the F-modules in use. The warnings and other notices contained in these manuals must be heeded at all times even if they are not repeated in this Getting Started.

Serious injury and damage to machines and equipment may result if these regulations are ignored.

## Production cell with access protection



- ① Emergency stop
- ② Laser scanner
- ③ Safety door
- ④ Control panel with start and acknowledgement pushbuttons

The entry to the production area is monitored with a laser scanner. The service area is secured by a safety door.

Entering the production area or opening the safety door results in a stop or shutdown of the production cell similar to an emergency stop.

The system can only be started when the emergency stop is cancelled, the safety door is closed, and the laser scanner detects no one in the protected area. On-site user acknowledgment is required to restart production after the emergency stop has been activated, the safety door has been opened or the protected area has reacted.

## 6.2.2 Procedure

The example in these Getting Started instructions consists of the following chapters:

### Configuring

You configure:

- An ET 200SP fail-safe digital input module for connecting an emergency stop switch, the connection of the position switches for monitoring a safety door, and the connection of the laser scanner for monitoring the entry area.
- An ET 200SP fail-safe digital output module for connecting a motor.
- An ET 200SP fail-safe digital input module for user acknowledgment, feedback circuit, and start.

The configuration is described in the section "Configuring (Page 219)".

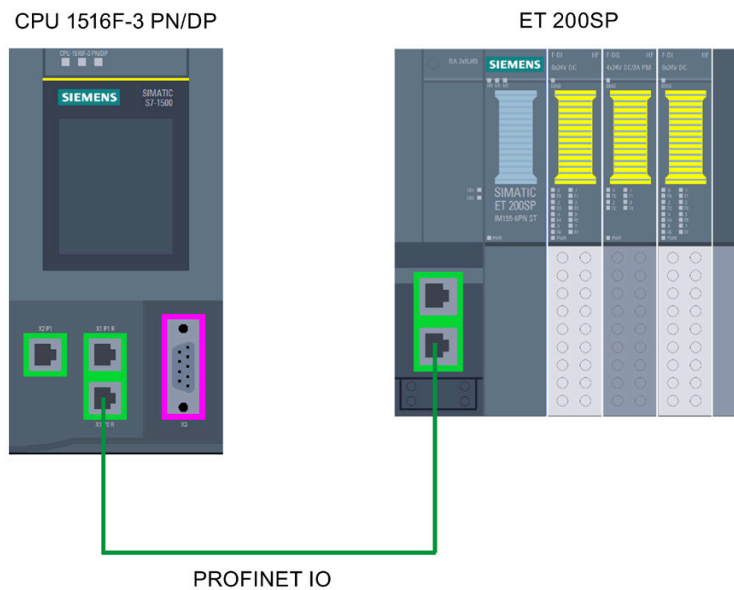
### Programming

Once the configuration is successfully completed, you can program your safety program.

In the Getting Started, a fail-safe block is programmed with an emergency stop, a safety door function, a feedback circuit (as restart protection when there is an incorrect contactor), and a user acknowledgment for reintegration. On completion, the safety program is compiled and downloaded to the F-CPU.

The programming is described in the section "Programming (Page 237)".

### Installation on PROFINET IO



### Wiring overview for ET 200SP

The Getting Started is based on the following wiring:

The laser scanner is connected to an external sensor supply.

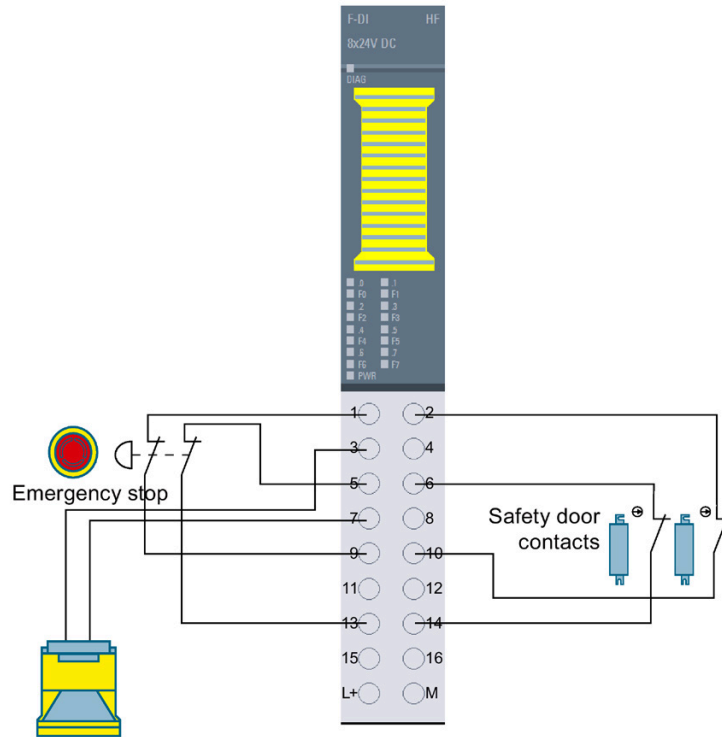


Figure 6-1 Wiring of the F-DI 8x24V DC HF

6.2 Introduction to example

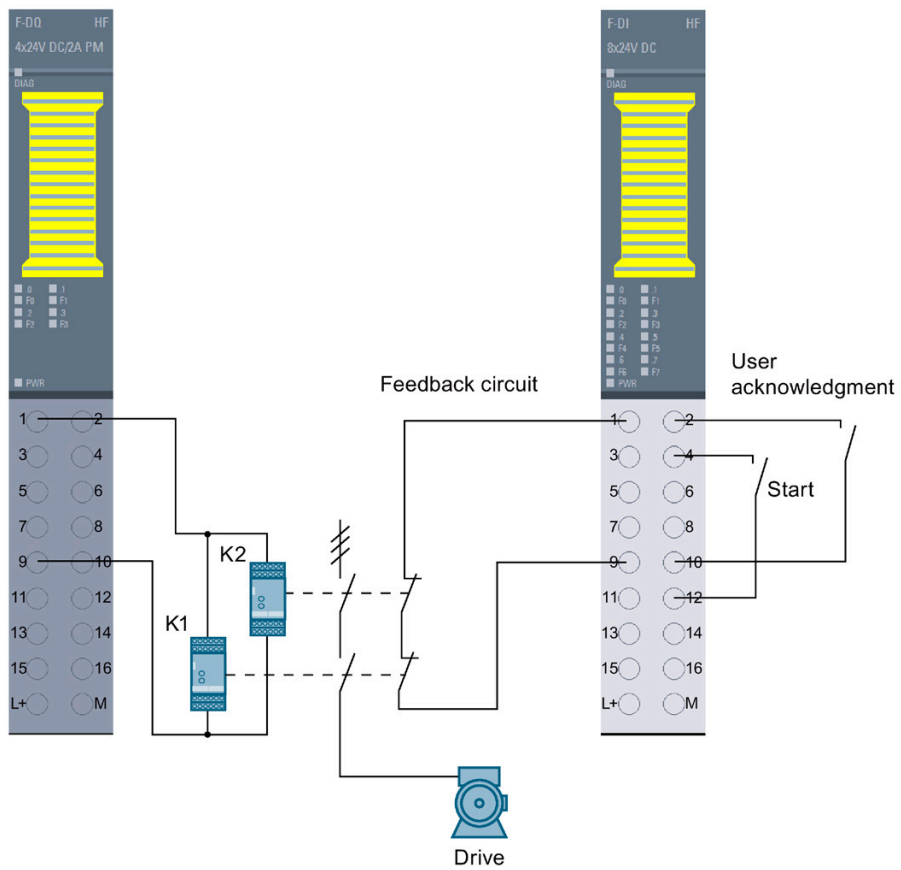


Figure 6-2 Wiring of the F-DQ 4x24V DC/2A PM HF and the F-DI 8x24V DC HF

## 6.2.3 Requirements for configuring and programming

### Requirements for the example

You need the following hardware for the Getting Started:

- CPU 1516F-3 PN/DP (6ES7516-3FN02-0AB0)
- SIMATIC Memory Card with at least 4 MB (e.g. 6ES7954-8LC03-0AA0)
- S7-1500 load current supply PM 70W 120/230VAC (6EP1332-4BA00)
- An ET 200SP distributed I/O system consisting of:
  - Interface module IM 155-6 PN ST incl. server module (6ES7155-6AU01-0BN0)
  - 1 BaseUnit BU15-P16+A0+2D (6ES7193-6BP00-0DA0)
  - 2 BaseUnits BU15-P16+A0+2B (6ES7193-6BP00-0BA0)
  - 2 fail-safe digital input modules F-DI 8x24VDC HF (6ES7136-6BA00-0CA0)
  - Fail-safe digital output module F-DQ 4x24VDC/2A PM HF (6ES7136-6DB00-0CA0)
- Emergency stop
- Position switch for monitoring a safety door
- Laser scanner
- 2 contactors which control a drive

The following software packages must be installed on your programming device/PC:

- *STEP 7 Professional V17*
- *STEP 7 Safety Advanced V17*

The programming device or PC must be connected to the F-CPU via the PROFINET interface.

Information on installation and wiring is available in the system manuals ET 200SP distributed I/O system (<https://support.industry.siemens.com/cs/ww/en/view/58649293>) and S7-1500 Automation System (<https://support.industry.siemens.com/cs/ww/en/view/59191792>).

## 6.2.4 Additional information

Detailed information on the hardware in use is available at:


- CPU 1516F-3 PN/DP
  - SIMATIC S7-1500 CPU 1516-3 PN/DP  
(<https://support.industry.siemens.com/cs/de/en/view/109767407>)
  - SIMATIC S7-1200/S7-1500 F-CPU's  
(<https://support.industry.siemens.com/cs/de/en/view/109478599>)
- S7-1500 load current supply PM 70W 120/230VAC  
(<https://support.industry.siemens.com/cs/ww/en/view/68036174>)
- Interface module IM 155-6 PN ST  
(<https://support.industry.siemens.com/cs/ww/en/view/59768173>)
- BaseUnits (<https://support.industry.siemens.com/cs/ww/en/view/59753521>)
- fail-safe digital input module F-DI 8x24VDC HF  
(<https://support.industry.siemens.com/cs/ww/en/view/78589499>)
- fail-safe digital output module F-DQ 4x24VDC/2A PM HF  
(<https://support.industry.siemens.com/cs/ww/en/view/78645789>)
- Server module (<https://support.industry.siemens.com/cs/ww/en/view/63257531>)



## 6.3 Configuring

### 6.3.1 Introduction

#### Introduction

 <b>WARNING</b>
<p><b>There is a risk of personal injury or damage to property.</b></p> <p>You may come into contact with live electrical wires connected to the mains power supply.</p> <p>Only wire the S7-1500 and ET 200SP when they are disconnected from the mains power supply.</p>

The installation and wiring of the CPU 1516F-3 PN/DP is described in the System Manual S7-1500 Automation System (<https://support.industry.siemens.com/cs/ww/en/view/59191792>) manual.

A detailed Getting Started which explains the installation of a CPU S7-1500 is available on the Internet (<https://support.industry.siemens.com/cs/de/en/view/78027451>).

#### Configuring the Hardware

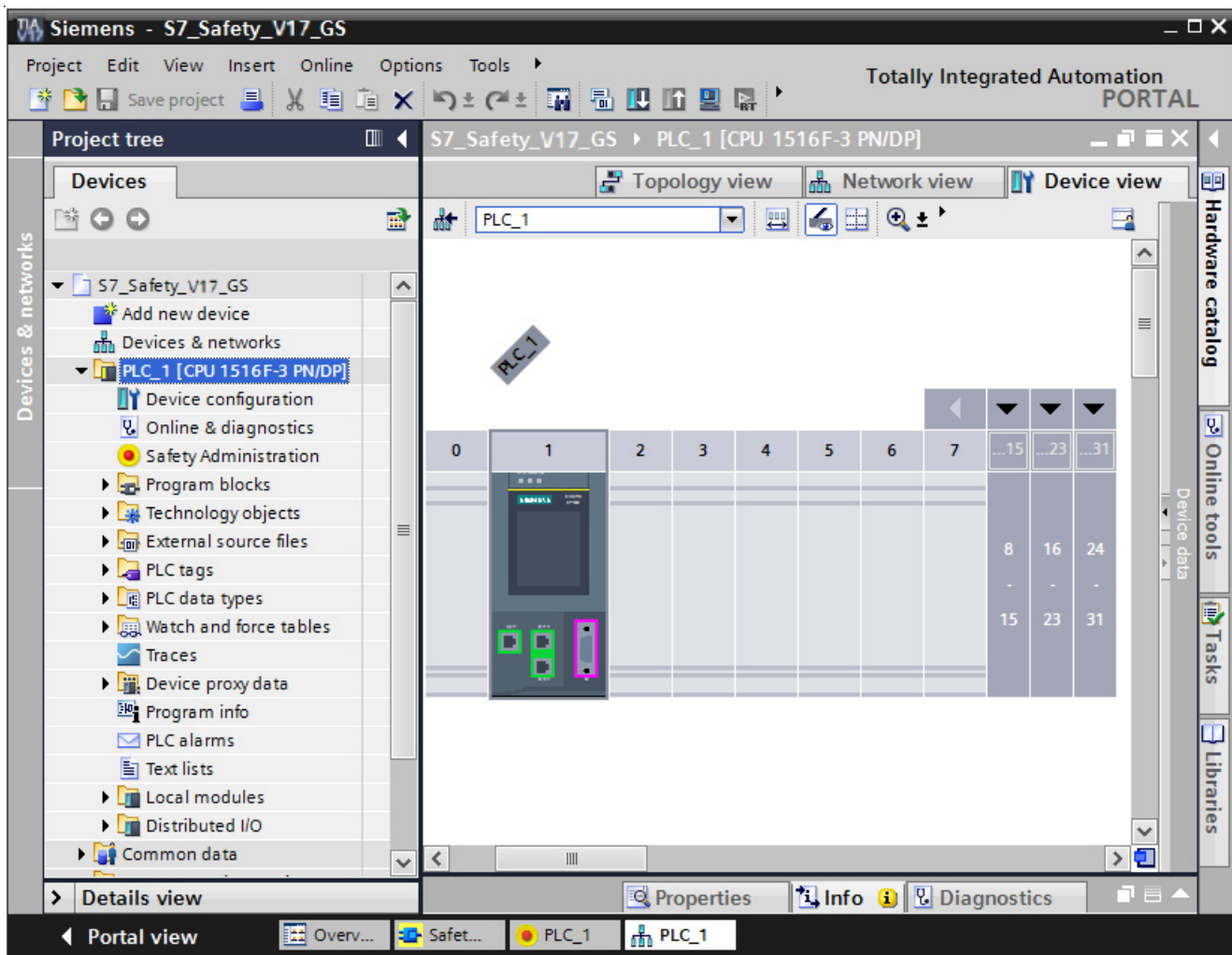
You configure in *STEP 7 Professional V17* with *STEP 7 Safety V17*:

- A CPU 1516F-3 PN/DP
- An ET 200SP distributed I/O system consisting of:
  - One interface module IM155-6 PN ST as of firmware V1.1
  - Two ET 200SP fail-safe digital input modules for connecting an emergency stop switch, position switches for monitoring a safety door, the laser scanner, the user acknowledgment, the feedback circuit and start
  - One ET 200SP fail-safe digital output module for connecting a motor

### 6.3.2 Step 1: Configuring the CPU 1516F-3 PN/DP

#### Introduction

In this step, you create a new project, add an F-CPU, and assign parameters for it.



## Procedure

1. In the portal view of *STEP 7 Professional V17*, create a new project with the name "S7\_Safety\_V17\_GS"
2. Add a CPU 1516F-3 PN/DP in the "Devices & networks" portal with "Add new device".  
**Result:** The device view with the CPU 1516F-3 PN/DP opens.
3. In the "Properties" of the F-CPU change to the "Fail-Safe" area.  
Here, you can change the following parameters or accept the default setting:
  - "Basis for PROFIsafe addresses"
  - "Default F-monitoring time for F-I/O of this interface"
4. Leave the default values unchanged for this example.

## Result

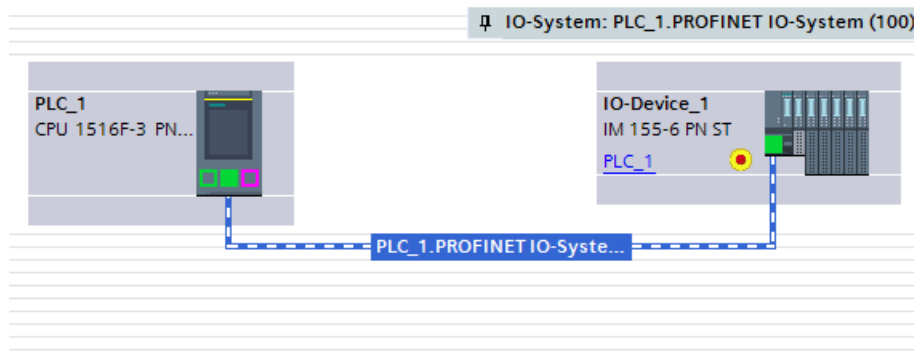
The new project has been created and the F-CPU has been configured.

### 6.3.3 Step 2: Configuring an ET 200SP distributed I/O system on a PROFINET subnet

#### Introduction

In this step, you configure the interface module for the ET 200SP distributed I/O system and a PROFINET subnet.

#### Procedure



1. In the project view, click the "Network view" tab.
2. In the hardware catalog, enter "IM155-6 PN" in the search field, and start the search with [Enter].
3. In the search results, click an interface module with article number 6ES7155-6AU01-0BN0
4. In the "Information" palette select the firmware version which matches the interface module in the "Version" drop-down list.
5. Drag the interface module with article number 6ES7155-6AU01-0BN0 from the "Catalog" pane to the graphic area of the network view.
6. Starting at the PROFINET interface of the IM155-6 PN ST hold down the mouse button and drag a PROFINET connection to the green PROFINET interface of the F-CPU on the right.

**Result:** A PROFINET subnet between the F-CPU and the IM155-6 PN ST is created automatically. The IP addresses and the PROFINET device names are also assigned automatically.

#### Result

The configuration of the interface module for the ET 200SP distributed I/O system and the PROFINET subnet is now complete.

### 6.3.4 Step 3: Configuring an F-DI module for connecting an emergency stop switch, position switches, and the laser scanner

#### Introduction

In this step, you configure a fail-safe digital input module for connecting the emergency stop, the position switches for monitoring the safety door, and the laser scanner for monitoring the entry area.

You will find detailed information on the parameters in the online help.

#### Procedure

1. In the graphic area of the network view, double-click on the IM155-6 PN ST.  
**Result:** The IM155-6 PN ST is opened in the device view.
2. In the device view of the ET 200SP, add an F-DI 8x24VDC HF fail-safe digital input module by dragging it from the hardware catalog to slot 1.
3. Select the "I/O addresses" area in the "Properties" tab.  
Leave the "Start address" parameter set to 0 for this example.
4. Change to the "F-parameter" area. Here, you can change the following parameters, if necessary, or apply the default settings:
  - "F-monitoring time"
  - "F-destination address"
  - "Behavior after channel fault"
  - "F-I/O DB-number"Leave the settings unchanged for the F-parameters for this example.
5. Switch to the "DI parameter" area.  
Disable the "Short-circuit test" parameter of the sensor supply for channels 2 and 6.

6. In this example, a two-channel emergency stop is to be connected to channels 0 and 4. Make the settings under "Channel parameter" as shown in the figure below. The equivalent 1oo2 evaluation then takes place in the F-module.

The screenshot displays the configuration interface for SIMATIC Safety, showing settings for a two-channel emergency stop connected to channels 0 and 4. The interface is organized into three main sections:

- Channel 0, 4:** This section contains global settings for the two-channel evaluation:
  - Sensor evaluation: 1oo2 evaluation, equivalent
  - Discrepancy behavior: Supply value 0
  - Discrepancy time: 500 ms
  - Reintegration after discrepancy error: Test 0-Signal not necessary
- Channel 0:** This section shows the configuration for the first channel:
  - Activated:
  - Sensor supply: Sensor supply 0
  - Input delay: 3,2 ms
  - Chatter monitoring:
  - Number of signal changes: 5
  - Monitoring window: 2 sec
- Channel 4:** This section shows the configuration for the second channel:
  - Activated:
  - Sensor supply: Sensor supply 4
  - Input delay: 3,2 ms
  - Chatter monitoring:
  - Number of signal changes: 5
  - Monitoring window: 2 sec

7. In this example, the position switches for monitoring a two-channel safety door will be connected to channels 1 and 5.

Make the settings under "Channel parameter" as shown in the figure below.

The screenshot displays the configuration interface for SIMATIC Safety V17, showing the settings for two channels (Channel 1 and Channel 5) under a parent group "Channel 1, 5".

**Channel 1, 5 (Parent Group):**

- Sensor evaluation: 1oo1 evaluation
- Discrepancy behavior: Supply value 0
- Discrepancy time: 5 ms
- Reintegration after discrepancy error: Test 0-Signal not necessary

**Channel 1 (Child Group):**

- Activated:
- Sensor supply: Sensor supply 1
- Input delay: 3,2 ms
- Chatter monitoring:
- Number of signal changes: 5
- Monitoring window: 2 sec

**Channel 5 (Child Group):**

- Activated:
- Sensor supply: Sensor supply 5
- Input delay: 3,2 ms
- Chatter monitoring:
- Number of signal changes: 5
- Monitoring window: 2 sec

8. In this example, the laser scanner for monitoring the accessible entry area is to be connected to channels 2 and 6.

Make the settings under "Channel parameter" as shown in the figure below.

The screenshot displays the configuration interface for SIMATIC Safety, showing settings for three channel groups:

- Channel 2, 6:**
  - Sensor evaluation: 1oo2 evaluation, equivalent
  - Discrepancy behavior: Supply value 0
  - Discrepancy time: 500 ms
  - Reintegration after discrepancy error: Test 0-Signal not necessary
- Channel 2:**
  - Activated:
  - Sensor supply: Sensor supply 2
  - Input delay: 3,2 ms
  - Chatter monitoring:
  - Number of signal changes: 5
  - Monitoring window: 2 sec
- Channel 6:**
  - Activated:
  - Sensor supply: Sensor supply 6
  - Input delay: 3,2 ms
  - Chatter monitoring:
  - Number of signal changes: 5
  - Monitoring window: 2 sec

9. Disable the unused DI channels 3 and 7 by clearing the "Activated" check box.

## Result

The configuration of the fail-safe DI module is now complete.



### 6.3.5 Step 4: Configuring an F-DQ module for connecting a motor

#### Introduction

In this step, you configure a fail-safe digital output module for indirect connection of a motor to channel 0 via two contactors.

You will find detailed information on the parameters in the online help.

#### Procedure

1. In the device view of the ET 200SP, add an F-DQ 4x24VDC/2A PM HF fail-safe digital output module by dragging it from the hardware catalog to slot 2.
2. Select the "I/O addresses" area in the "Properties" tab.  
Leave the "Start address" parameter set to 6 for this example.
3. Change to the "F-parameter" area. Here, you can change the following parameters or accept the default settings:
  - "F-destination address"
  - "F-monitoring time"
  - "Behavior after channel fault"
  - "F-I/O DB-number"

Leave the settings unchanged for the F-parameters for this example.

4. Change to the "DO parameter" area.

Make the settings under "Channel parameter" as shown in the figure below.

The screenshot shows the 'DO parameter' configuration window. At the top, there is a section for 'Maximum test period' with a value of 1000 and a unit dropdown set to 'sec'. Below this is the 'Channel 0' section. It contains several parameters:
 

- 'Activated': A checked checkbox.
- 'Max. readback time dark test': A text input field with '1.0' and a unit dropdown set to 'ms'.
- 'Max. readback time switch on test': A text input field with '0.6' and a unit dropdown set to 'ms'.
- 'Activated light test': An unchecked checkbox.
- 'Diagnosis: Wire break': A checked checkbox.

 Each parameter has a yellow warning icon to its right.

5. Disable the unused DQ channels 1, 2 and 3 by clearing the "Activated" check box.

#### Result

The configuration of the fail-safe DQ module is now complete.

### 6.3.6 Step 5: Configuring an F-DI module for user acknowledgement, feedback circuit and start

#### Introduction

In this step, you configure a fail-safe digital input module for connection of a user acknowledgement, a feedback circuit and start.

You will find detailed information on the parameters in the online help.

#### Procedure

1. In the device view of the ET 200SP, add an F-DI 8x24VDC HF fail-safe digital input module by dragging it from the hardware catalog to slot 3.
2. Select the "I/O addresses" area in the "Properties" tab.  
Leave the "Start address" parameter set to 11 for this example.
3. Change to the "F-parameter" area. Here, you can change the following parameters, if necessary, or apply the default settings:
  - "F-monitoring time"
  - "F-destination address"
  - "Behavior after channel fault"
  - "F-I/O DB-number"Leave the settings unchanged for the F-parameters for this example.
4. Switch to the "DI parameter" area.  
Disable the "Short-circuit test" parameter of the sensor supply for channels 3 and 7.

5. In this example, a feedback circuit is to be connected to channel 0.

Make the settings under "Channel parameter" as shown in the figure below.

The screenshot shows the configuration interface for SIMATIC Safety. It is divided into two sections:

- Channel 0, 4:** This section is currently expanded. It contains the following settings:
  - Sensor evaluation: 1oo1 evaluation
  - Discrepancy behavior: Supply value 0
  - Discrepancy time: 5 ms
  - Reintegration after discrepancy error: Test 0-Signal not necessary
- Channel 0:** This section is collapsed. It contains the following settings:
  - Activated:  (checked)
  - Sensor supply: Sensor supply 0
  - Input delay: 3,2 ms
  - Chatter monitoring:  (unchecked)
  - Number of signal changes: 5
  - Monitoring window: 2 sec

Disable channel 4 by clearing the "Activated" check box.

6. In this example, the user acknowledgment is to be connected to channel 1.

Make the settings under "Channel parameter" as shown in the figure below.

The screenshot shows the configuration interface for SIMATIC Safety. It is divided into two sections:

- Channel 1, 5:** This section is currently expanded. It contains the following settings:
  - Sensor evaluation: 1oo1 evaluation
  - Discrepancy behavior: Supply value 0
  - Discrepancy time: 5 ms
  - Reintegration after discrepancy error: Test 0-Signal not necessary
- Channel 1:** This section is collapsed. It contains the following settings:
  - Activated:  (checked)
  - Sensor supply: Sensor supply 1
  - Input delay: 3,2 ms
  - Chatter monitoring:  (unchecked)
  - Number of signal changes: 5
  - Monitoring window: 2 sec

Disable channel 5 by clearing the "Activated" check box.

7. In this example, start is to be connected to channel 3.

Make the settings under "Channel parameter" as shown in the figure below.

The screenshot displays two configuration panels for SIMATIC Safety. The top panel is titled "Channel 3, 7" and contains the following settings:

- Sensor evaluation: 1oo1 evaluation
- Discrepancy behavior: Supply value 0
- Discrepancy time: 5 ms
- Reintegration after discrepancy error: Test 0-Signal not necessary

The bottom panel is titled "Channel 3" and contains the following settings:

- Activated
- Sensor supply: Sensor supply 3
- Input delay: 3,2 ms
- Chatter monitoring
- Number of signal changes: 5
- Monitoring window: 2 sec

Disable channel 7 by clearing the "Activated" check box.

8. Disable the unused DI channels 2 and 6 by clearing the "Activated" check box.

9. Add a server module by dragging it from the hardware catalog to slot 4.

## Result

The configuration of the fail-safe DI module and the ET 200SP is now complete.

## 6.3.7 Step 6: Download hardware configuration

### Introduction

In this step, you download the hardware configuration to the F-CPU.

### Procedure

1. Select the F-CPU in the project tree.
2. In the shortcut menu for the F-CPU, select "Download to device > Hardware configuration". If an online connection to the F-CPU does not yet exist, you are prompted to establish this connection.

**Result:** The hardware configuration is compiled. The "Load preview" dialog is displayed afterward.

3. Click the "Load" button.

**Result:** The hardware configuration is downloaded.

4. The "Load results" dialog is then displayed, if applicable. Next click the "Finish" button.

### 6.3.8 Step 7: Assign device name

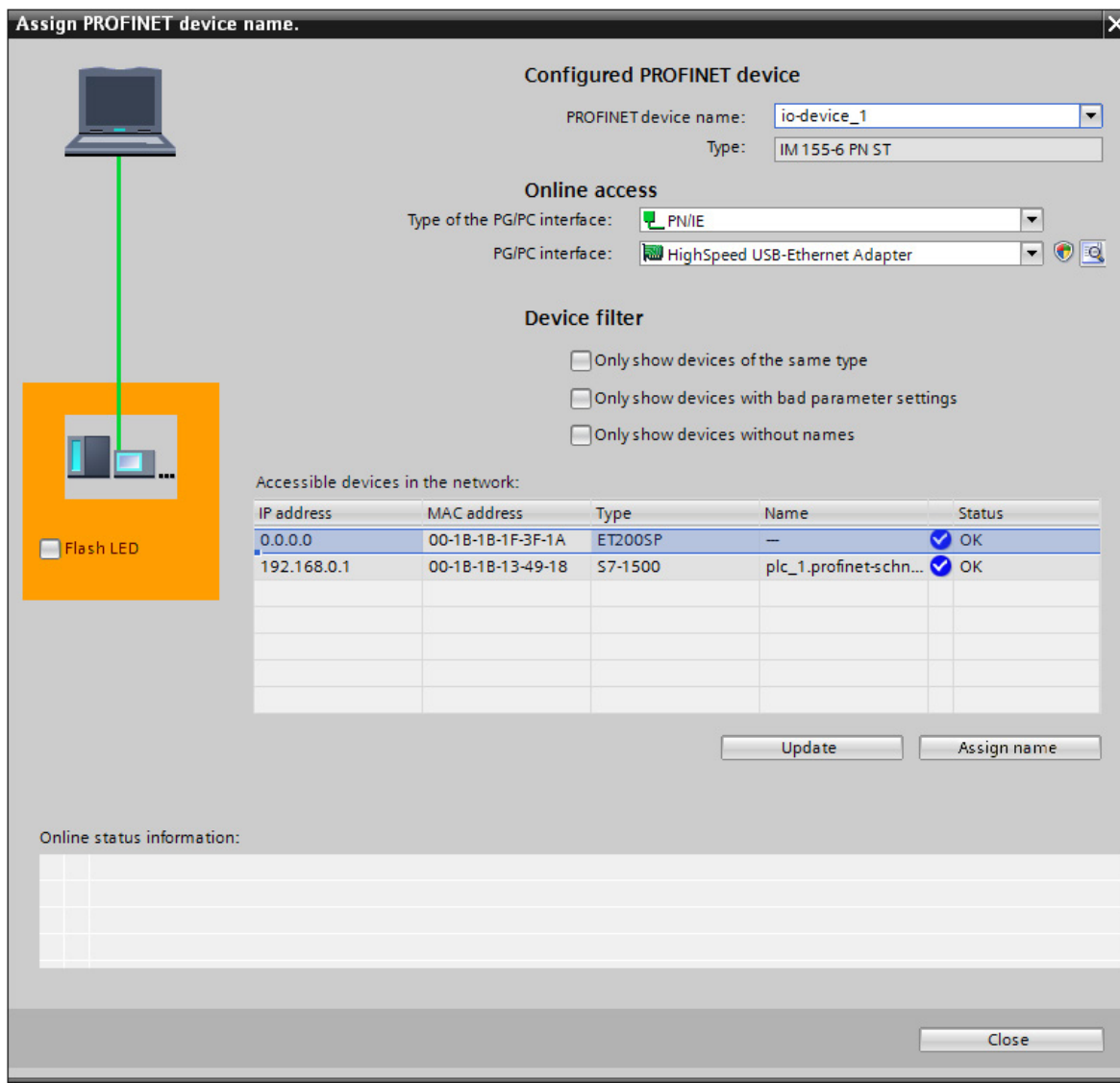
#### Introduction

In this step, you assign a valid PROFINET device name to the interface module. The device name of the F-CPU has already been assigned during the download of the hardware configuration.

The PROFINET device names are created automatically by *STEP 7 Professional V17*; you only need to assign them.

For additional information on PROFINET IO, refer to the "PROFINET with STEP 7 V17 (<https://support.industry.siemens.com/cs/ww/en/view/49948856>)" function manual.

You will find additional information on configuring PROFINET IO in the online help for *STEP 7 Professional V17* under "Configurations for PROFINET IO".



## Procedure

1. Change to the Network view.
2. Select "PN/IE\_1" subnet in the network view.
3. Select "Assign device name" from the shortcut menu.  
**Result:** The "Assign PROFINET device name" dialog opens.
4. Select "io\_device\_1" for the ET200SP in the "PROFINET device name" drop-down list.
5. Select the entry with the type "ET 200SP" in the "Accessible participants in the network" table.
6. You can use the "Flash LED" button to identify the device.
7. Click the "Assign name" button.

## Result

You have successfully assigned the device name of the IM155-6 PN ST.

### 6.3.9 Step 8: Assign PROFIsafe addresses

#### Introduction

Each F-I/O is uniquely addressed by its PROFIsafe address. The PROFIsafe address consists of an F-source address and an F-destination address.

The uniqueness of the PROFIsafe address is ensured by the combination of F-source address and F-destination address.

The PROFIsafe address must be unique network-wide and CPU-wide (system-wide) for each ET 200SP F-module. This is the case if the following two conditions are met:

- The F-source address ("Basis for PROFIsafe addresses" parameter) of the F-CPU is unique network-wide.
- The F-destination address of the F-module is unique CPU-wide.

Detailed information on PROFIsafe addresses is available in the manual "SIMATIC Safety - Configuring and Programming

(<https://support.industry.siemens.com/cs/ww/en/view/54110126>)" in the section "Configuring".

Detailed information on the assignment of PROFIsafe addresses is available in the manual "SIMATIC Safety - Configuring and Programming

(<https://support.industry.siemens.com/cs/ww/en/view/54110126>)" in the section "Assigning the F-destination address for ET 200SP fail-safe modules".

#### Rules for address assignment

The ET200SP is an F-module of the PROFIsafe address type 2. Keep to the following rules when assigning the PROFIsafe addresses:

#### WARNING

F-I/O of PROFIsafe address type 2 is uniquely addressed using a combination of F-source address ("Basis for PROFIsafe addresses of the assigned F-CPU" parameter) and F-destination address.

The combination of F-source address and F-destination address for each F-I/O must be unique network-wide\* and CPU-wide\*\* (system-wide). In addition, the F-destination address must not be occupied by F-I/O of PROFIsafe address type 1.

To ensure that addresses are unique across F-CPU's for supported configurations, you need to ensure that the "Basis for PROFIsafe addresses" parameter of all F-CPU's is unique network-wide\*. This is achieved by having different settings for the "Basis for PROFIsafe addresses" parameter of the F-CPU's.

\* A network consists of one or more subnets. "Network-wide" means beyond the boundaries of the subnet(s). In PROFIBUS, a network includes all nodes accessible via PROFIBUS DP. In PROFINET IO, a network includes all nodes accessible via RT\_Class\_1/2/3 (Ethernet/WLAN/Bluetooth, Layer 2) and if applicable RT\_Class\_UDP (IP, Layer 3).



\*\* "CPU-wide" means all F-I/Os assigned to an F-CPU: central F-I/O of this F-CPU as well as F-I/Os for which the F-CPU is DP master/IO controller. An F-I/O that is addressed using I-slave-slave communication is assigned to the F-CPU of the I-slave and not to the F-CPU of the DP master / IO controller.

## Procedure

For this example, leave the PROFIsafe addresses assigned automatically by the F-system. The module is ready for the assignment of the F-destination address when all LEDs are lit red. Assign the PROFIsafe addresses as follows:

1. Select the ET 200SP in the network view.
2. Select "Assign F-destination address" from the shortcut menu.
3. Under "Identification", select the method to be used for identifying the F-modules.

- "by LED flashing"

This is the default setting. During identification, the STATUS LEDs of the F-modules to be identified flash.

- "by serial number"

If you cannot see the F-modules clearly, you can identify them by the serial number of the interface module.

---

### Note

The displayed serial number may have a year number added to it compared with the serial number printed on the interface module. The serial numbers are nevertheless identical.

---

4. In the "Assign" column, select all the F-modules to which you want to assign the F-destination address.  
If you select the interface module in the "Assign" column, all F-modules of the station are selected.
5. Click the "Identification" button. Check whether the status LEDs for the F-modules whose F-destination address you want to assign are flashing green.  
If you identify using the serial number, compare the displayed serial number with the serial number of the interface module.
6. Confirm successfully identified F-modules in the "Confirm" column of the table.
7. Use the "Assign F-destination" button to assign the F-destination addresses to the F-modules.  
You must confirm the "Confirm Assignment" dialog within 60 seconds.
8. Close the dialog.

## Result

You have successfully assigned the PROFIsafe addresses.

### 6.3.10 Summary: Configuring the Hardware

#### Summary

So far, you have configured the following according to the task definition for the example:

- An F-CPU 1516F-3 PN/DP
- An ET 200SP with:
  - Interface module IM 155-6 PN ST
  - An ET 200SP fail-safe digital input module for connecting an emergency stop, position switches for monitoring a safety door, and the laser scanner for monitoring the accessible production area.
    - Start addresses of the input and output addresses: both 0
    - Channels 0 and 4 for emergency stop
    - Channels 1 and 5 for safety door position switch
    - Channels 2 and 6 for the laser scanner
  - One ET 200SP fail-safe digital output module for connecting a motor
    - Start addresses of the input and output addresses: both 6
    - Channel 0 for indirect switching of motor via two contactors
  - An ET 200SP fail-safe digital input module for user acknowledgment, feedback circuit and start.
    - Start addresses of the input and output addresses: both 11
    - Channel 0 for feedback circuit
    - Channel 1 for user acknowledgment
    - Channel 3 for start
  - Server module

You can now continue with programming the safety program.

## 6.4 Programming

### 6.4.1 Introduction

#### Introduction

In this example, you create a fail-safe block (F-FB). In F-FB you program:

- the safety door function,
- the emergency stop function (safety circuit for shutdown with emergency stop, if the safety door is open, or the protected area monitored by the laser scanner is entered),
- a feedback monitoring circuit (as automatic restart protection in case of a faulty contactor)
- the user acknowledgment for reintegration

You then compile the safety program and download it to the F-CPU.

### 6.4.2 The structure of the safety program

#### Introduction

When you insert the F-CPU, an F-runtime group and the associated F-OB with main safety block (F-FB with instance DB) are created by default.

The first step in programming your safety program is the main safety block. When this block is compiled, additional instructions are added that call the remaining F-blocks of the safety program.

## The structure of a safety program

For structuring purposes, a safety program consists of one or two F-runtime groups.

The main safety block is called in the F-CPU by the F-OB assigned to the F-runtime group.

From the main safety block you can call additional F-FBs and F-FCs with safety functions.

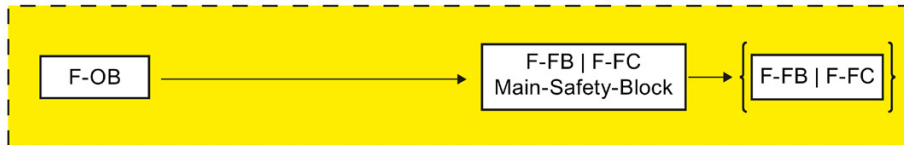


Figure 6-3 F-runtime group

This example consists of one F-runtime group. The F-FB "Safety\_Interlock" is called from the main safety block of the F-runtime group.

Detailed information on F-runtime groups is available in the manual "SIMATIC Safety - Configuring and Programming

(<https://support.industry.siemens.com/cs/ww/en/view/54110126>)" in the section "Defining F- Runtime Groups".

Detailed information on the safety program is available in the manual "SIMATIC Safety - Configuring and Programming

(<https://support.industry.siemens.com/cs/ww/en/view/54110126>)" in the section "Program structure of the safety program (S7-1500)".

## Programming

A safety program consists of F-blocks that you create using the FBD or LAD programming language and F-blocks that are automatically added.

In this example, you program using the FBD programming language. Note the following differences compared to programming a standard user program:

- Preconnection of the enable input EN or evaluation of the enable output ENO is not possible.
- Limitations in the instructions
- Limitations in the used data types and operand areas
- Fail-safe signals are shown in yellow in the *LAD/FBD Editor*.

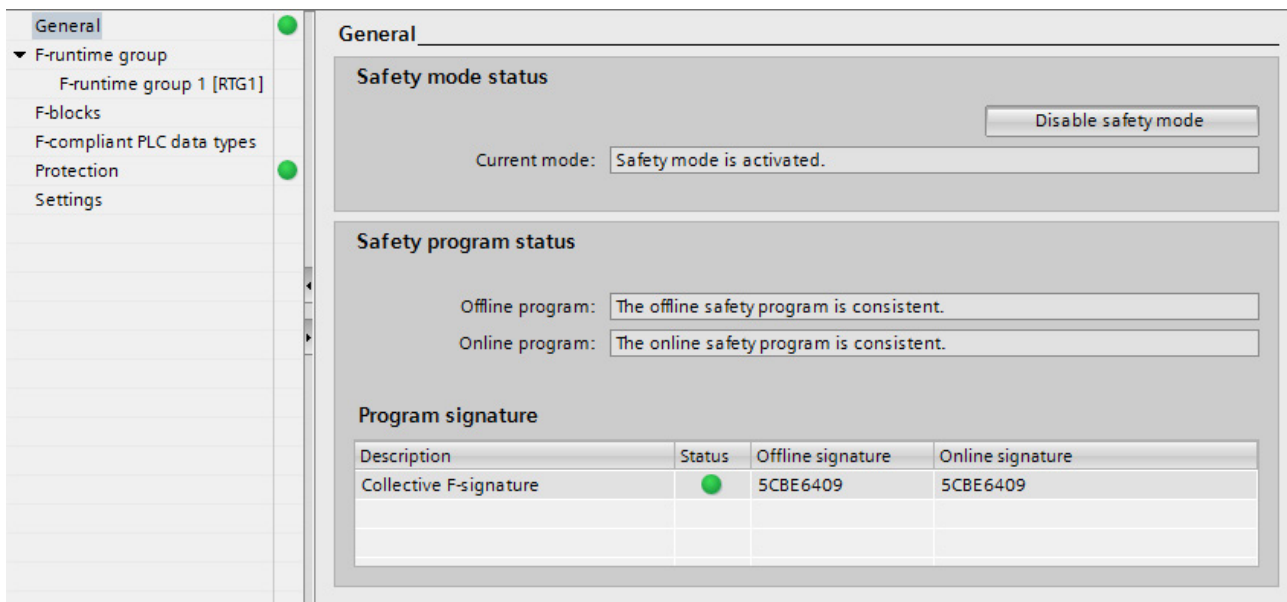
(refer to the manual "SIMATIC Safety - Configuring and Programming

(<https://support.industry.siemens.com/cs/ww/en/view/54110126>)" in the section "Overview of Programming").

### 6.4.3 The Safety Administration Editor

The *Safety Administration Editor* supports you in the following tasks:

- Displaying the status of the safety program
- Displaying the F-collective signature
- Displaying the status of the safety mode
- Creating/organizing F-runtime groups
- Displaying information on the F-blocks
- Displaying information about F-compliant PLC data types
- Specifying/changing access protection
- Specifying/changing general settings for the safety program



Detailed information on the Safety Administration Editor is available in the manual "SIMATIC Safety - Configuring and Programming (<https://support.industry.siemens.com/cs/ww/en/view/54110126>)".

## 6.4.4 Step 9: Specifying the centralized settings for the safety program

### Introduction

In this step, you learn how to call the *Safety Administration Editor*.

### Opening the *Safety Administration Editor*

1. In the project tree of the F-CPU, double-click on "Safety Administration".

**Result:**The *Safety Administration Editor* opens.

You make central settings for the safety program in the *Safety Administration Editor*.

2. In the area navigation of the *Safety Administration Editors*, switch to "F-runtime group".

The automatically created F-runtime group and the F-OB with the associated main safety block are displayed when you create the F-CPU.

Leave the preset blocks for this example.

3. In the area navigation of the *Safety Administration Editor*, switch to "Settings".

Here you can make the settings for the safety program.

Keep the preassigned settings for this example.

For additional information on the *Safety Administration Editor*, refer to the "SIMATIC Safety - Configuring and Programming

(<https://support.industry.siemens.com/cs/ww/en/view/54110126>)" manual in the section "Safety Administration Editor".

## 6.4.5 Step 10: Create PLC tag table

### Introduction

You create symbolic names for each input and output of the F-I/O in the PLC tag table.

	Name	Data type	Address
1	ESTOP	Bool	%I0.0
2	Safety_Door_SW1	Bool	%I0.1
3	Laserscanner	Bool	%I0.2
4	Safety_Door_SW2	Bool	%I0.5
5	Safety_Door_SW1_VS	Bool	%I1.1
6	Safety_Door_SW2_VS	Bool	%I1.5
7	Motor_VS	Bool	%I6.0
8	Feedback	Bool	%I11.0
9	Quit	Bool	%I11.1
10	START	Bool	%I11.3
11	Motor	Bool	%Q6.0

### Specifying inputs and outputs for the safety program

1. Create a new PLC tag table in the F-CPU project navigation under "PLC tags" with "Add new tag table".
2. Rename the new PLC tag table with the [F2] key as "Safety Program".
3. Double-click the PLC tag table to open it.
4. Assign symbolic names for the inputs and outputs as shown in the figure above.

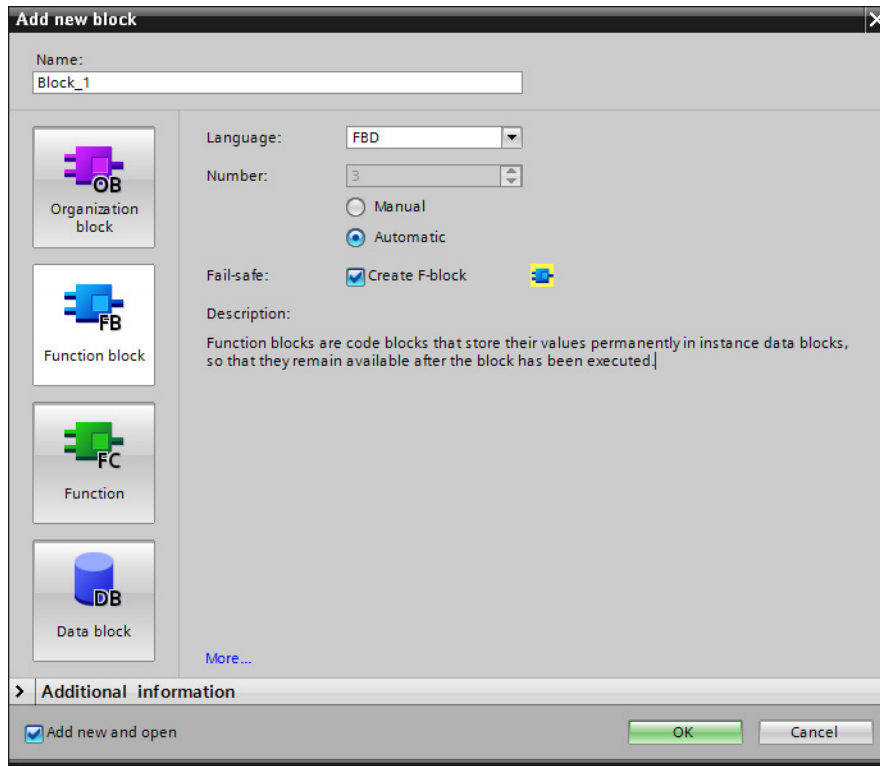
### Result

In the next steps of the Getting Started, you can now assign symbolic names to the inputs and outputs of the instructions simply using drag-and-drop from the detail view of the "Safety Program" PLC tag table.

## 6.4.6 Step 11: Creating an F-FB

### Introduction

In this step, you create an F-FB in which you program the safety functions for this example in the next steps.



### Procedure

1. Go to the "Program blocks" folder of the F-CPU and double-click "Add new block".  
**Result:** The "Add new block" dialog opens.
2. Under "Name" enter "Safety\_Interlock" for the name of the F-FB.
3. Click the "Function block" button on the left.
4. Select the "Create F-block" option.
5. Choose "FBD" as the language for the F-FB.
6. Close the dialog box with "OK".

### Result

The F-FB "Safety\_Interlock" is created in the "Program blocks" folder and opens automatically in the *LAD/FBD Editor*.

You can now continue with programming the safety functions in the next step.

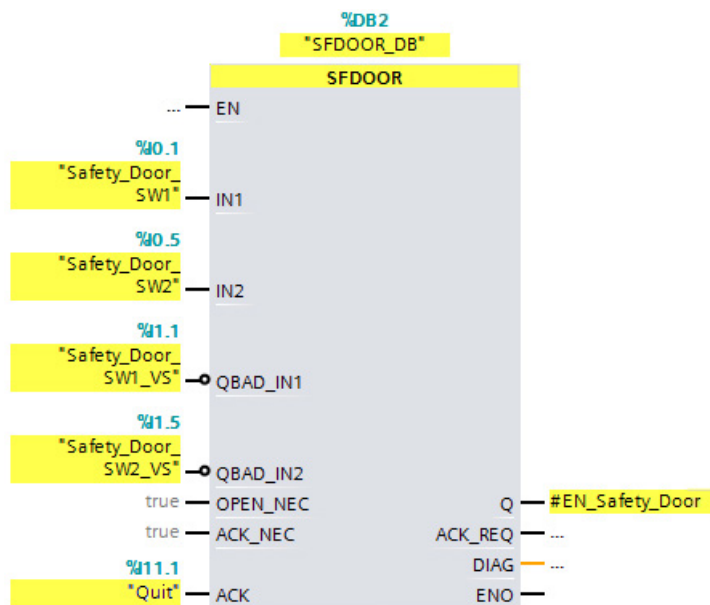


## 6.4.7 Step 12: Programming the safety door function

### Introduction

In this step, you program the safety door function for this example.

You will find detailed information on the "SFDOOR" instruction in the online help.



### Procedure

1. Create the following static tag of data type BOOL in the interface of the "Safety\_Interlock" F-FB:

Name	Data type
EN_Safety_Door	Bool

2. Insert the "SFDOOR" instruction from the "Safety functions" subfolder of the "Instructions" task card into network 1.
3. Click "OK" to confirm the "Call options" dialog.
4. Supply the inputs and outputs as shown in the figure above.
5. Insert one "Invert RLO" each at the inputs QBAD\_IN1" and "QBAD\_IN2" from the "Instructions" task card.

### Result

The programming of the safety door function is now complete.

## 6.4.8 Step 13: Programming the emergency stop function

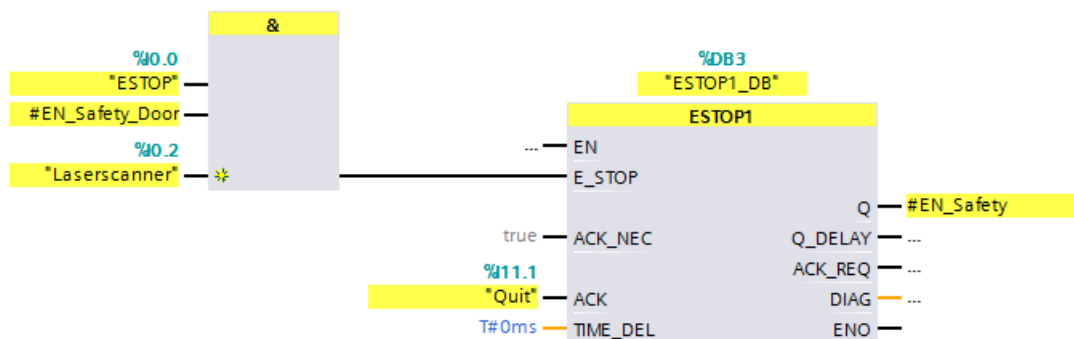
### Introduction

In this step, you program the emergency stop function for this example.

An emergency stop is executed:

- when the emergency stop is activated
- when the safety doors open
- when the protected area of the laser scanner is entered

You will find detailed information on the "ESTOP1" instruction in the online help.



### Procedure

1. Create the following static tag of data type BOOL in the interface of the "Safety\_Interlock" F-FB:

Name	Data type
EN_Safety	Bool

2. Insert the "AND logic operation" instruction from the "Bit logic operations" subfolder of the "Instructions" task card in network 2.
3. Insert a third input at the "AND logic operation" instruction by clicking on the yellow star.
4. Supply the inputs of the instruction as shown in the figure above.
5. Insert the "ESTOP1" instruction from the "Safety functions" subfolder of the "Instructions" task card into network 2.
6. Click "OK" to confirm the "Call options" dialog.
7. Supply the inputs and outputs of the instruction as shown in the figure above.
8. Connect the output of the "AND logic operation" instruction to the "E\_STOP" input of the "ESTOP1" instruction.

### Result

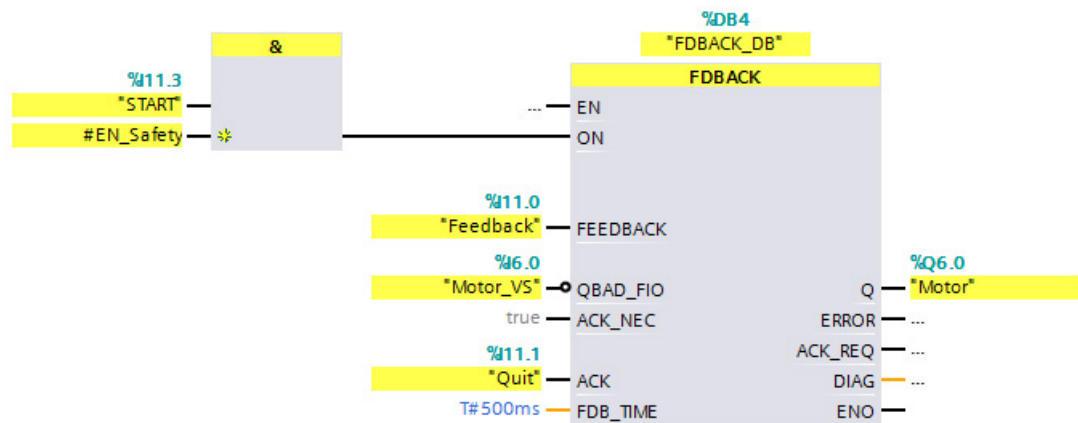
The programming of the emergency stop function is now complete.

## 6.4.9 Step 14: Programming the feedback monitoring

### Introduction

In this step, you program the feedback circuit monitoring for this example.

You will find detailed information on the "FDBACK" instruction in the online help.



### Procedure

1. Insert the "AND logic operation" instruction from the "Bit logic operations" subfolder of the "Instructions" task card in network 3.
2. Supply the inputs of the instruction as shown in the figure above.
3. Insert the "FDBACK" instruction from the "Safety functions" subfolder of the "Instructions" task card into network 3.
4. Click "OK" to confirm the "Call options" dialog.
5. Supply the inputs and outputs of the instruction as shown in the figure above.
6. Insert a "Negate RLO" at the "QBAD\_FIO" input from the "Instructions" task card.
7. Connect the output of the "AND logic operation" instruction to the "ON" input of the "FDBACK" instruction.

### Result

The programming of the feedback monitoring is now complete.

## 6.4.10 Step 15: Programming the User Acknowledgment for Reintegration of the F-I/O

### Introduction

In this step, you program the user acknowledgement for reintegration of the F-I/O for this example.

In your safety program, you must provide a user acknowledgment for the reintegration for the F-I/O. In this example, this is the "Quit" input.

You can use the ACK\_GL instruction to reintegrate all F-I/O of an F-runtime group.

### Note

A user acknowledgment with a positive edge at the ACK\_GL instruction is required for a reintegration of the F-I/O (i.e., for switching from fail-safe values (0) to process data) after a fault is corrected:

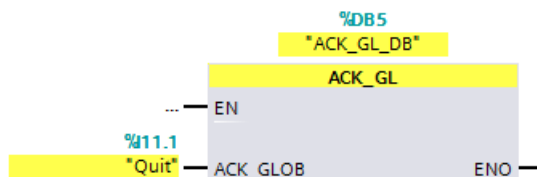
- After every communication error
- After F-I/O faults or channel faults when parameter ACK\_NEC = true in the F-I/O DBs

You will find detailed information on the "ACK\_GL" instruction in the online help.

If you use the option of user acknowledgment by means of an acknowledgment key and there is a communication error / F-I/O fault / channel fault on the F-module to which the acknowledgment key is connected (F-DI 8x24VDC HF on slot 3), acknowledgment for reintegration of this F-module is not possible.

This "block" can only be removed by a STOP/RUN transition of the F-CPU.

Consequently, it is recommended that you also provide for an acknowledgment by means of an HMI system in order to acknowledge reintegration of the F-module to which the acknowledgment key is connected. You can find more information on this in the manual "SIMATIC Safety - Configuring and Programming (<https://support.industry.siemens.com/cs/ww/en/view/54110126>)", section "Implementing User Acknowledgment in the Safety Program of the F-CPU of a DP Master or IO Controller").



## Procedure

1. Insert the "ACK\_GL" instruction from the "Safety functions" subfolder of the "Instructions" task card into network 4.
2. Click "OK" to confirm the "Call options" dialog.
3. Supply the input as shown in the figure above.

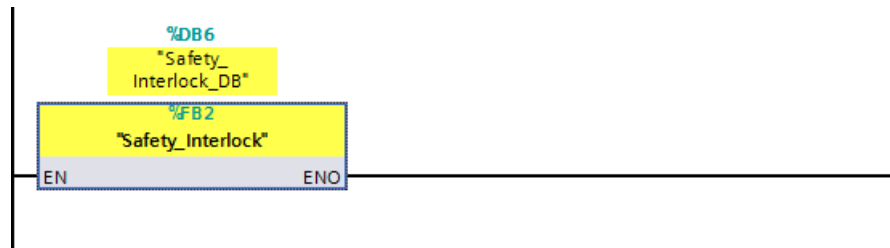
## Result

The programming of the user acknowledgment is now complete.

### 6.4.11 Step 16: Programming of the main safety block

#### Introduction

In this step, you program the main safety block for this example. The main safety block was created automatically when the F-CPU was created.



## Procedure

1. Double-click in the project navigation to open the main safety block "Main\_Safety".
2. Use drag-and-drop to insert the F-FB "Safety\_Interlock" into network 1 of the main safety block.
3. Click "OK" to confirm the "Call options" dialog.

## Result

The F-FB "Safety\_Interlock" is now called cyclically in the main safety block.

You have now programmed the functionality according to the task definition of the example. You can now proceed with the next steps to compile the safety program and then download the safety program along with the hardware configuration to the F-CPU.

## 6.4.12 Step 17: Compiling the safety program

### Introduction

In this step, you compile the safety program.

A consistency check is performed on the execution-relevant F-blocks when the safety program is compiled, that is, the safety program is checked for errors. Any error messages are output in the "Compile" tab. After a successful consistency check, the additionally required F-system blocks are generated automatically and added to the F-runtime group to generate an executable safety program.

### Procedure

1. Select the F-CPU in the project tree.
2. In the shortcut menu for the F-CPU, select "Compile > Software (only changes)".

**Result:** The safety program is compiled.

### Result

After successful compilation, you always have a consistent safety program that is ready for approval.

## 6.4.13 Step 18: Download the safety program to the F-CPU

### Introduction

In this step, you download the safety program to the F-CPU.

### Procedure

1. Select the F-CPU in the project tree.
2. In the shortcut menu for the F-CPU, select "Download to device > Software (only changes)".  
If an online connection to the F-CPU does not yet exist, you are prompted to establish this connection.

**Result:** The "Load preview" dialog is displayed.

3. Click the "Load" button.

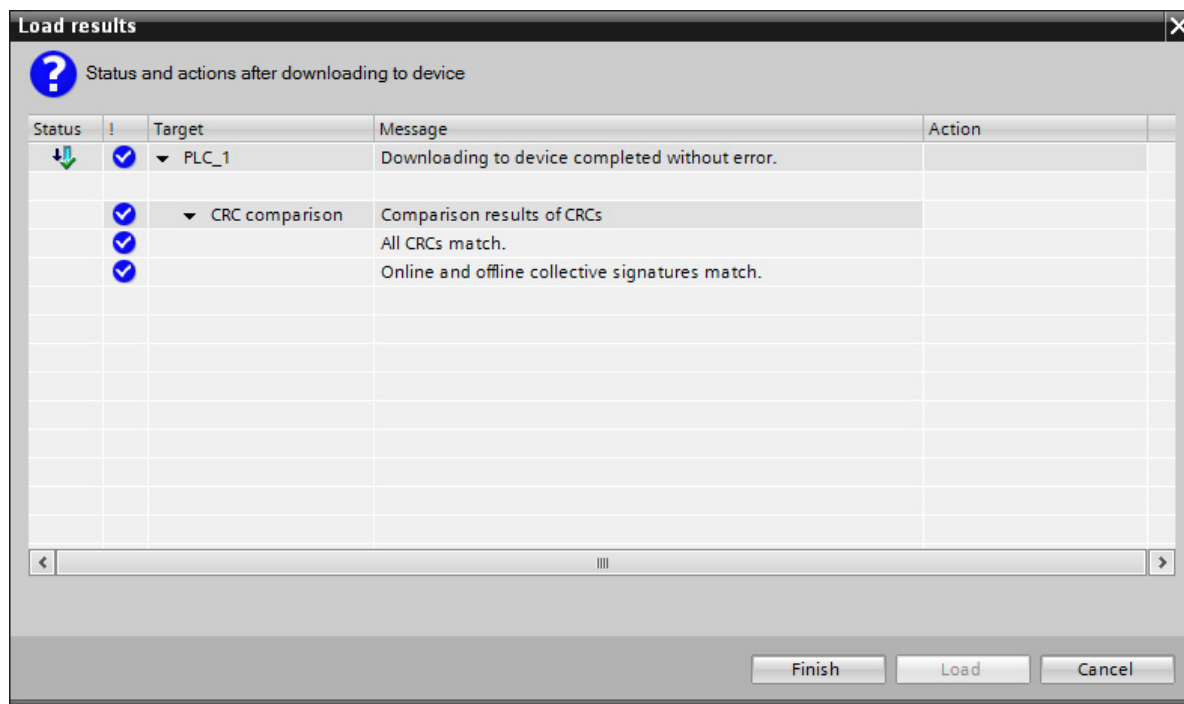
---

### Note

To download the entire safety program, the F-CPU must be in STOP mode.

---

**Result:** The safety program is downloaded and the "Load results" dialog is displayed.



4. Check in the dialog if the F-collective signatures are identical online and offline.  
If so, the download operation was successful. If not, repeat the download operation.

5. Click the "Finish" button.
6. Switch the F-CPU from STOP to RUN mode.

The display of the F-CPU shows you the current status of the safety mode in the "Overview > Fail-safe" menu item.



---

**Note**

Once a safety program has been created, you need to perform a full function test according to your automation task (see SIMATIC Safety - Configuring and Programming (<https://support.industry.siemens.com/cs/ww/en/view/54110126>) manual).

---

**Result**

You have now finished creating the safety program according to the task definition of the example.

In the following appendix, we show you how easy it is to set up access protection for your safety program and the F-CPU.



## 6.5 Setting up access protection

### Introduction

It is essential to provide access protection in production mode for the access to the SIMATIC Safety F-system.

No access protection is initially necessary for test purposes, commissioning, etc. This means you can execute all offline and online actions without access protection, that is, without password prompt.

In this step, you set up access protection for the safety program and the F-CPU.

For additional information, refer to the "Access protection" chapter in the SIMATIC Safety - Configuring and Programming (<https://support.industry.siemens.com/cs/ww/en/view/54110126>) manual.

### Procedure

To set up access protection for productive operation, follow these steps:

1. In the area navigation of the *Safety Administration Editors*, switch to "Access protection".
2. Under "Offline safety program protection" click "Set up". Enter the password in the dialog that appears, and enter it again to confirm.
3. Close the dialog with "OK".

**Result:** You have set up access protection for the safety program.

4. Under "F-CPU access protection", click the link "Go to the "Protection" area of the F-CPU".

**Result:** You switch to the device view of the F-CPU.

5. Under "Protection", select the "Full access (no protection)" option. Under "Access permission", enter a password. Then enter the password again to confirm.

Select the access level for the PLC.

Access level	Access				Access permission
	HMI	Read	Write	Fail-safe	Password
<input checked="" type="radio"/> Full access incl. fail-safe (no protection)	✓	✓	✓	✓	*****
<input type="radio"/> Full access (no protection)	✓	✓	✓		
<input type="radio"/> Read access	✓	✓			
<input type="radio"/> HMI access	✓				
<input type="radio"/> No access (complete protection)					

6. Select the F-CPU in the project tree.
7. In the shortcut menu for the F-CPU, select "Download to device > Hardware and software (only changes)".

**Result:** You have set up access protection for the F-CPU and the safety program.

## Result

You can only make changes to the safety program offline on your programming device/PC if you enter the password from Step 2.

You cannot overwrite the safety program in the F-CPU until you enter the password from Step 4.

As the next step, acceptance testing of the system may be necessary for productive operation. For additional information regarding acceptance testing, refer to the "System Acceptance Test" chapter in the "SIMATIC Safety - Configuring and Programming (<https://support.industry.siemens.com/cs/ww/en/view/54110126>)" manual.

## 6.6 Typical configuring and programming errors and their causes

### Errors, causes and remedy measures

Type	Error	Possible causes
Configuration error	F-blocks cannot be downloaded to the F-CPU.	F-CPU parameter "F-activation" is not activated in the "Fail-Safe" area.
Configuration error	DIAG-LED flashes red and all channel error LEDs on the F-module are lit	The PROFIsafe address does not match the PROFIsafe address in the hardware configuration.
Configuration error	The DIAG-LED on the F-module flashes red and TIMEOUT error in the DIAG byte of the F-I/O DB	Monitoring time of the F-module $\leq$ max. cycle time of the F-runtime group.
Configuration error	The DIAG-LED on the F-module flashes red and CRC error in the DIAG byte of the F-I/O DB	<ul style="list-style-type: none"> <li>Loaded safety program is not consistent with the loaded hardware configuration.</li> <li>Safety program is inconsistent.</li> <li>PIQ/PII of the F-module is being overwritten by the standard user program.</li> </ul>
Configuration error	The DIAG-LED on the F-DI module flashes red, channel error LED is lit and F-module signals discrepancy	<p>Sensor connection does not match parameter assignment, for example:</p> <ul style="list-style-type: none"> <li>Only one switching contact is connected to a channel with 1oo2 evaluation</li> <li>A sensor with nonequivalent contacts is connected to a channel set to "two-channel equivalent".</li> </ul>
Programming error	F-CPU changes to STOP	<ul style="list-style-type: none"> <li>Main safety block is being called more than once in the cyclic program.</li> <li>Operands of F-DBs are being written to in the standard user program.</li> <li>Undeclared TEMP variables are being used in the safety program.</li> <li>Bit memory that is modified during processing of the main safety block, e.g., clock memory, is being read-accessed in the safety program.</li> <li>Overflow with mathematical functions.</li> <li>F-monitoring time set too low. The F-monitoring time must be greater than the max. runtime (see section "Monitoring and response times" in the manual "SIMATIC Safety - Configuring and Programming (<a href="https://support.industry.siemens.com/cs/ww/en/view/54110126">https://support.industry.siemens.com/cs/ww/en/view/54110126</a>)").</li> </ul>

# Index

## P

- Protection, 137, 140, 141, 142
  - Behavior of a password-protected CPU, 144
  - Copy protection, 140
  - Know-how protection, 137
  - mechanical locking, 141

## S

- System diagnostics, 115
  - Detail view, 116
  - Device view, 116
  - Diagnostic buffer view, 116
  - Matrix view, 118
  - System diagnostics view, 115
  - System diagnostics window, 115
- System diagnostics view, 115
- System diagnostics window, 115